STIBO SYSTEMS
MASTER DATA MANAGEMENT

# SOLUTION ENABLEMENT

## PMDM for Automotive

## Reference Guide

Release 2024.1- 7.0.59 (March 2024)

# Table of Contents


**Table of Contents** ..... **2**

**Automotive Reference Guide Introduction** **13**

**Supported Versions and Formats** ..... **14**

**Application Editor Screen** ..... **16**

Prerequisites ..... 17

Configuring Application Editor Screen Properties ..... 17

Node List Properties ..... 21

Display Modes ..... 22

Actions ..... 24

Create Application Action ..... 25

Delete Application Action ..... 26

Bulk Applications Updates ..... 26

Searching for the Missing Application Records ..... 27

Create Application Action Component ..... 27

Configuring the Create Application Action Component ..... 28

Creating applications using Create Application Action ..... 33

Faceted Node Picker Dialog ..... 36

Faceted Node Picker Dialog Interface Overview ..... 36

Using The Faceted Node Picker dialog ..... 38

Configuring Faceted Node Picker Dialog Properties ..... 39

Prerequisites ..... 39

Configuration ..... 39

Improving Part Type Accuracy ..... 44

Configuring the Attribute Search Field ..... 45

Configuring the Hierarchy Search Field ..... 47

Configuring the Name Search Field ..... 49

Bulk Applications Updates Component ..... 51

Configuring the Bulk Applications Updates Component ..... 52

Using Business Action(s) for Bulk Update (s) Action ..... 62

Node Picker Parameter Component ..... 63

Configuring the Node Picker Parameter Component ..... 64

Faceted Node Picker Parameter Component ..... 71

Configuring the Faceted Node Picker Parameter Component ..... 73

**Application Manager** ..... **81**

Variations Across Standards ..... 83

Intelligent Search Interface ..... 84

Using Intelligent Search Interface ..... 87

Using Search Boxes ..... 91

Searching for Criteria ..... 91

Selecting Valid Criteria ..... 91

Removing Criteria ..... 95

Minimum Search Criteria ..... 95

Search Box Synchronization ..... 97

Using Make/Model Search Box ..... 98

Using Year Search Box ..... 103

Using Options Search Boxes ..... 104

Using Part Type Search Box .....107
Using Part Search Box .....108
Saved Search Tool .....111
Saving a Search .....112
Renaming a Saved Search .....113
Deleting a Saved Search .....114
Missing Application Coverage Functionality .....115
Improving Part Type Accuracy .....117
Application Coverage Report .....118
Application Coverage Report Results .118
Running an Application Coverage Report .....119
Adding the Application Coverage Report to a Background Processes Screen .....122
Confusing Counterman Solution .....124
Adding and Configuring Confusing Counterman Report .....125
Prerequisites .....125
Adding / Configuring Confusing Counterman Report Component ....125
Using Confusing Counterman Solution128
Running a Confusing Counterman Report .....128
Multiple Parts For Same Vehicle Report Component .....131
Results Table and Toolbar .....135
Results Table Default Columns .....141
ACES Application Qualifier Table Header .....142
Using Application Qualifier Value Editor to Add or Remove References to Qualifiers .....144
Application Approval Status Table Header .....147
Application Assembly Value Table Header .....151
Application Comment Table Header .. 157
Using Application Comment Value Editor to Add or Remove References to Comments .....162
Application Condition Header Components .....164
Application Criteria Blocks Table Header .....171
Using Application Criteria Blocks Value Editor .....176
Application Set Assembly Table Header .....179
Suggested Part Number Table Header .....185
Controlling Display of Conditions in Application Manager .....192
Creating Applications .....193
Using Multi-Select .....195
Automotive Validation Path Functionality .. 196
Validation Path Functionality and Search Box Criteria .....197
Validation Path Functionality and Options on Applications .....199
Automatic Population of Options on Applications .....199
Options Within the Value Editor ..... 199
Validation Path Functionality and 201

Missing Applications .....
Validation Path Functionality and Advanced Filtering .....202
Error Handling .....203
Workbench Configurations Related to Validation Path Functionality ..... 205
Vehicle Configurations .....206
Base Vehicle Applications .....208
Options on Applications .....210
Options / Conditions Stored as Attributes .....211
Options / Conditions Stored as References .....212
Configuring Application Manager .....214
Own Model Considerations .....214
Configuring an Application Manager Screen in Web UI .....215
Configuring Make/Model Search Box ....227
Configuring Attribute Search Box .....229
Configuring Options Group Search Box .231
Configuring Part Type Search Box .....237
Editing the Part Type Search Box Label .....237
Configuring Year Search Box .....240
Configuring Part Search Box .....241
Editing the Part Search Box Parameters .....242
**Application Suggestion Solution .....246**
Adding New Applications Manually ..... 249
Adding Applications from Suggestions .....252
Removing Existing Applications ..... 255
Removing Application from Suggestions List .....257
Configuring the Part Application Editor Component .....260
Configuring the Application Suggestions Component .....264
Configuration .....264
**Competitor and OE Numbers Solution ...274**
Adding Part Number References Manually 277
Removing Existing Part Number References .....279
Removing Part Number Reference Suggestions .....281
Creating and Referencing New Competitor and OE Part Numbers .....283
Normalization / Harmonization of Part Numbers ..... 286
Configuring the Part Number References Component .....288
Adding Part Number References from Suggestions .....293
Configuring the Part Number Suggestions Component .....295
Configuring the Advanced Part Number References Component .....302
Configuring the Advanced Part Number Suggestions Component .....307
Configuring the Data Model for Application Competitor or OE Part Numbers Table Header ..... 318
Configuring the Application Competitor or OE Part Numbers Table Header Component .....324

Configuring the Metadata Attribute References Component ....329

Automotive Business Rule Plugins ....334

Business Action: Application Suggestions ....336

Configuring Business Action for Application Suggestions ....338

Business Action: Change Assembly ....344

Configuring the Change Assembly Business Action in Workbench ....346

Configuring the Change Assembly Business Action in Web UI ....348

Business Action: Change Part Type ....355

Configuring the Change Part Type Business Action in Workbench ....358

Configuring the Change Part Type Business Action in Web UI ....361

Business Action: Change Part ....368

Configuring the Change Part Business Action in Workbench ....371

Configuring the Change Part Business Action in Web UI ....374

Business Action: Copy Application to Other Assembly ....381

Configuring the Copy Application to Other Assembly Business Action in Workbench ....383

Configuring the Copy Application to Other Assembly Business Action in Web UI ....385

Business Action: Copy Application to Other Part ....393

Configuring the Copy Application to Other Part Business Action in Workbench ....397

Configuring the Copy Application to Other Part Business Action in Web UI ....399

Business Action: Copy Applications to Related Parts ....406

Part Relation Reference Type Configuration ....406

Steps for Using the Business Action in Web UI ....407

Configuring the Copy Applications to Related Parts Business Action in Workbench ....409

Configuring the Reference Type ....409

Configuring the Business Action ....409

Configuring the Copy Applications to Related Parts Business Action in Web UI ....411

Business Action: Number Suggestions ....413

Configuring Business Action for Number Suggestions ....415

Business Action: Move ACES Applications for PIES Part ....422

Business Action: Set Condition Links on Part Types ....423

Business Action: Set Import Status Attributes ....424

Configuring Change Flags for New and/or Updated Objects ....425

Use Case Appendix ....429

Use Cases - Tracking Newly Created Objects ....430

Use Cases - Tracking Deleted Objects ....434

Use Cases - Tracking Changed Objects ....437

Use Cases - Clearing Delete Status Flag ..... 441

Business Action: Sync ACES Applications Between PIES Parts ..... 444

Business Action: Validate Application Conditions and Start in Workflow ..... 445

Configuring Business Action for Applications with Invalid Configuration 446

Business Action: Validate PIES Item ..... 450

Configuring Business Action to Handle Invalid PIES Items ..... 451

Business Condition: Check Path for Missing Application ..... 454

Improving Search Result Accuracy ..... 454

Configuring the Check Path for Missing Application Business Condition in Workbench ..... 456

Business Condition: TecDoc Reference Start Condition ..... 461

Prerequisites ..... 462

Configuring 'TecDoc Reference Start Condition' Business Condition ..... 462

Business Condition: TecDoc Supplier Start Condition ..... 466

Prerequisites ..... 466

Configuring Business Condition ..... 467

Business Condition: Validate Application on Import ..... 471

**Country Handling Solution ..... 475**

Using Country Handling ..... 476

Using Country Handling within Application Manager ..... 476

Using Country Handling within Node 478

Editor .....

Using Country Handling Value Editor to Add or Edit Country Handling Data ..... 483

Configuring Automotive - Country Model Component Model ..... 488

Configuring Application Country Handling Table Header ..... 498

Configuring Country Handling Value Component ..... 501

**Data Onboarding ..... 504**

Configuring Data Onboarding Solution ..... 505

Data Onboarding Solution Initial Setup ..... 506

Configuring the Execute Mapper Configuration Business Action ..... 511

Configuring Web UI for Data Onboarding Solution ..... 513

Prerequisites ..... 513

Onboarding Mappings Details Screen ..... 513

Mapper Configurations - Global Navigation Panel Component ..... 513

Onboarding Mappings Details Screen ..... 515

Configuration Prerequisites ..... 515

Configuration Process ..... 515

Implementing the Onboarding Mappings Details Screen ..... 521

Configuring the Global Navigation Panel ..... 522

Configuring the Mapper Configurations to Display Onboarding Mappings Details Screen ..... 524

Configuring Mapper Configuration Setup Entity ..... 529

Prerequisites ....529
Creating Mapper Configuration Setup Entity in Workbench ....529
Creating Mapper Configuration Setup Entity in Web UI ....530
Configuring Mapper Configuration Setup Entity in Web UI ....532
Target Hierarchy Editor Dialog ....537
Prerequisites ....539
Configuring the Target Hierarchy Editor Dialog ....539
Using the Onboarding Mappings Details Screen ....547
Creating Mapping Plugins ....547
Managing Mapping Plugins ....549
Deleting an Existing Mapping Plugin 549
Executing Mapper Configuration Setup Entity ....551
Prerequisites ....551
Executing Through Mapper Action Button ....551
Executing Through Stacked Mapper Action Button ....556
Executing by Onboarding Comparison Screen ....561
Executing by a Background Process ....562
Configuring the Onboarding state to execute Mapper Configuration setup entity in the Workflow ....564
Monitoring the automated Onboarding process ....572
Executing by a Business Rule ....574
Mapping Plugins ....576
Application Mapping Plugin ....576
Attribute Mapping Plugin ....593
Automatic Mapping Plugin ....606
Children of Type Mapping Plugin ....608
Concatenator Mapping Plugin ....615
Object to Object Mapping Plugin ....626
Mapping Validation Path Functionality 639
Business Action Mapping Plugin ....641
Attribute Transformation in Mapper Configuration Setup Entity ....645
Creating Attribute Transformations Through Mapping Guide Window ....646
Creating Attribute Transformations Through Target Hierarchy Editor Window ....651
Modifying Transformations on the Transformation Overview Dialog ....657
Add a transformation ....659
Test an attribute transformation ....660
Delete a transformation ....660
Change the order of the transformations ....661
Edit an existing transformation ....662
Testing Attribute Transformations Configured in Mapping Plugins ....664
Mapping Plugins Use Case Overview ....666
Use Case Appendix ....667
Attribute Mapping Use Cases ....667
Object to Object Mapping Use Cases . 667

Use Cases - Copy Attribute Values from One Object to Another Object .... 668
Example One .... 668
Use Cases - Copy Data from One Object to Another Object .... 678
Example One .... 678
Example Two .... 689
Use Cases - Copy Data from One Object to Child Object of Another Object .... 699
Example One .... 699
Example Two .... 710
Use Cases - Copy Data from One Object to a Referenced Object of Another Object .... 723
Example One .... 723
Modifying Mapper Rows on the Onboarding Mapping Details Screen .... 734
Add a mapper row to the mapping plugin 735
Delete a mapper row in the mapping plugin .... 736
Disable a mapper row in the mapping plugin .... 736
Enable the disabled mapper row in the mapping plugin .... 736
Check the health of the mapper row .... 737
Add a description to the mapper row .... 740
Change the order of the mapper row in the mapping plugin .... 742
Onboarding Comparison Screen .... 743
Prerequisites .... 743
Onboarding Comparison Screen 743
Configuration ....
Configuration Prerequisites .... 744
Configuration Process .... 744
Configuring Access to an Onboarding Comparison Screen .... 749
Using Onboarding Comparison Screen Interface .... 751
Using Applications Stage .... 757
Using Attributes Stage .... 766
Using Objects Stage .... 773
Forward Screen Action Component .... 781
Using Forward Screen Action button in Search Screens .... 786
**Exporting Automotive Data .... 787**
AutoCare ACES Application Exporter .... 788
Step 1. Start ACES Application Exporter .... 788
Step 2. Select Objects .... 788
Step 3. Select Format .... 789
Step 5. Advanced .... 795
Example .... 800
Start Export from Configuration Plugin .... 804
AutoCare PIES Exporter .... 813
Step 1. Start AutoCare PIES Exporter 813
Step 2. Select Objects .... 813
Step 3. Select Format .... 814
Example .... 830
NAPA Application Exporter .... 839
**Importing Automotive Data .... 842**

Import Framework ....843
Import Process Overview ....843
Default Workflow States and Functions ....846
Validation State ....848
Validation State Common Parameters ....848
Validation State AutoCare PIES Specific Parameter ....850
Validation State TecDoc Specific Parameter ....851
Validation Service Function Details ....852
Validation State Results ....853
Conversion State ....855
AutoCare Conversion State Parameters ....855
NAPA Conversion State Parameters ....861
TecDoc Conversion State Parameters ....863
Conversion State Function Details ....866
Conversion State Results ....867
Configuring Application Year Range Functionality in Workbench ....868
Configuring Application Name to Include Year Range ....871
Delta Calculation State ....874
Delta Calculation State Common Parameters ....874
Delta Calculation State Function Details ....877
Delta Calculation State Results ....879
Error State ....880
Ready for Import State ....881
Rejected State ....882
Import State ....883
Import State Common Parameters ....883
Import State Function Details ....884
Import State Results ....885
Import Completed State ....886
Discard File State ....887
Modifying Import Framework ....888
Displaying Import Modifications in Web UI ....890
ID Structures in Importers ....891
Adding Reporting Extensions to Imports ....894
Validation Error Handling ....895
AutoCare Importers ....898
AutoCare ACES Importer ....899
Using AutoCare ACES Importer ....900
Importing ACES Files via Web UI ....901
Import Process Overview ....901
Using the AutoCare ACES Importer in Web UI ....902
Sample AutoCare ACES File Structure ....908
Header ....909
Body ....910
Footer ....912
ACES Import Validation Rules ....913
Configuring AutoCare ACES Importer ....915
Configuring an IIEP for AutoCare ....916

ACES Imports .....
Configuring a File Loading Widget for AutoCare ACES Imports .....922
AutoCare PIES Importer .....925
Using AutoCare PIES Importer .....926
Importing PIES Files via Web UI .....927
Import Process Overview .....927
Using the AutoCare PIES Importer in Web UI .....927
Sample AutoCare PIES File Structure .....934
Header .....940
Price Sheet Header Segment (PRCS) ..946
PIES Item Segment (ITEM) .....947
PIES Description Segment .....949
PIES Price Segment PRCE .....951
PIES Extended Product Info Segment (EXPI) .....952
Packaging Segment (PACK) .....956
Hazardous Material Package Segment (HAZM) .....957
Kits Segment (KITS) .....959
Interchange Segment (INTE) .....961
Digital Asset File Information Segment (ASST) .....962
Output .....964
PIES Import Validation Rules .....967
Configuring AutoCare PIES Importer .968
Configuring an IIEP for AutoCare PIES Imports .....969
Configuring a File Loading Widget for AutoCare PIES Imports .....975
AutoCare PAdb Importer .....979
Using AutoCare PAdb Importer .....980
Importing PAdb Files via Web UI ....981
Import Process Overview .....981
Using the AutoCare PAdb Importer in Web UI .....982
PAdb Import Validation Rules .....987
Configuring AutoCare PAdb Importer .989
Configuring an IIEP for AutoCare PAdb Imports .....990
Configuring a File Loading Widget for AutoCare PAdb Imports .....996
AutoCare VCdb Importer .....1000
Using AutoCare VCdb Importer .....1000
Importing VCdb Files via Web UI ..1001
Import Process Overview .....1001
Using the AutoCare VCdb Importer in Web UI .....1002
Sample AutoCare VCdb File Structure .....1008
VCdb Import Validation Rules .....1015
Using VCdb Change Handling Service .....1016
Configuring AutoCare VCdb Import Workflow to Run VCdb Change Handling .....1017
Configuring AutoCare VCdb Importer .....1025
Configuring an IIEP for AutoCare VCdb Imports .....1026

Configuring a File Loading Widget for AutoCare VCdb Imports .....1032
Qdb Import Validation Rules ..... 1036
Brand Table Import Validation Rules ...1037
PCdb Import Validation Rules ..... 1038
TecDoc Reference Data Importer ..... 1039
Using TecDoc Reference Data Importer .....1040
Importing Reference Data Files via Web UI ..... 1041
Import Process Overview .....1041
Using the TecDoc Reference Data Importer in Web UI .....1041
Reference Data Import Validation Rules .....1047
Error Handling for Reference Data Imports ..... 1048
Link Type (Reference Type) Not Valid for Generic Article .....1049
Missing Country Code When Importing Vehicle Country Specifications .....1054
Missing Object Type When Adding Additional Descriptions to Vehicle Model Series .....1059
Missing Object Type When Allocating Engine Number ..... 1063
Missing Object Type When Allocating German KBA Vehicle Number .....1067
Validation Error When Importing LOV Values ..... 1072
Configuring TecDoc Reference Data Importer .....1076
Configuring an IIEP for TecDoc Reference Data Imports .....1077
Configuring a File Loading Widget for Reference Data Imports .....1083
TecDoc Supplier Data Importer .....1086
Using TecDoc Supplier Data Importer .1087
Importing Supplier Data Files via Web UI ..... 1088
Import Process Overview .....1088
Using the TecDoc Supplier Data Importer in Web UI .....1088
Configuring TecDoc Supplier Data Importer .....1095
Configuring an IIEP for Supplier Data Imports ..... 1096
Configuring a File Loading Widget for Supplier Data Imports ..... 1102
**Reference Delegation Screen ..... 1105**
Prerequisites .....1105
Configuring Reference Delegation Screen .....1105
Use Case .....1109
Example One ..... 1110
Example Two ..... 1111
**Search by Card Screen .....1113**
Search by Card Screen Interface Overview ..... 1113
Search by Card Initial Configuration .....1116
Configuration Prerequisites .....1116
Configuration Process ..... 1116
Adding Search by Card to the Homepage ..... 1120

Configuring the Attribute Search Box .. 1124
Configuring the Hierarchy Search Box .1126
Configuring the Name Search Box ..... 1130
Using Search by Card Screen Interface ..1132
Using Search Cards .....1134
Searching for Criteria .....1135
Selecting Valid Criteria .....1135
Selecting Multiple Criteria .....1135
Removing Criteria .....1138
Minimum Search Criteria .....1138
Using Attribute Search Card .....1138
Using Hierarchy Search Card ..... 1139
Using Name Search Card .....1143
Using Search by Cards Screen with Data Onboarding Tasks ..... 1144
Prerequisites .....1144
Navigating to the Onboarding Comparison Screen ..... 1145
**STEP Path Component .....1146**
Prerequisites .....1147
Adding a STEP Path component to a Node Editor ..... 1147
Configuring the STEP Path component .. 1150
STEP Path Elements ..... 1151

# 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.

# 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

- **Brand Table Importer:** All three flat file formats are supported (original, with revision date, and with sub-brands and revision date with or without OEM)
- **PAdb Importer:** ASCII
- **PCdb Importer:** ASCII
- **Qdb Importer:** ASCII
- **VCdb Importer:** ASCII
- **ACES Exporter:** ACES versions 3.0, 3.2, 4.0, 4.1, and 4.2
- **ACES Importer:** ACES versions 3.0, 3.2, 4.0, 4.1, and 4.2
- **PIES Exporter:** PIES versions 6.5, 6.7, 7.0, 7.1, and 7.2
- **PIES Importer:** PIES versions 6.5, 6.7, 7.0, 7.1, and 7.2

## NAPA: The NAPA commercial license has been discontinued.

- **Application Exporter:** Format is not versioned
- **Application Importer:** Format is not versioned
- **Asset Reference Exporter:** 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 and TAF 2.6 using either the ZIP or 7z formats
- **Supplier Data Exporter:** TAF 2.4 and TAF 2.6
- **Supplier Data Importer:** TAF 2.4 and TAF 2.6 using either the ZIP or 7z formats

# Application Editor Screen

Running step 1 of Easy Setup for each standard (AutoCare / NAPA / TecDoc) will create two Application Editor Screens:

- One for when a Part is selected to display the applications belonging to that Part.
- One for when a Vehicle / Assembly is selected to display the applications belonging to that Vehicle / Assembly. This screen can also be used for searching missing application records for the selected vehicle.

Web UI users can view, edit, create, and delete application records within these screens. Users can also further configure results table in the screen to meet their requirements.

The Application Editor Screen require an object selection, so these screens need to be mapped for a particular object type or added on a tab within another screen that is mapped to the appropriate object. To map the Part's or Vehicle's screen, so that when you are on a part or vehicle, only that screen is displayed, refer to the **Mappings** topic within the **Main Properties Overview** section of **STEP Online Help**.

To add the Parts or Vehicles screen to an existing screen, use the Sub Screen Tab Page configuration, described in the **Tab Pages** topic in the **Web User Interfaces** documentation of **STEP Online Help**.

**Note:** AutoCare will be used as an example throughout this topic.

Each of the automatically configured Application Editor Screens for the different standards are described below:

## AutoCare

Each of the automatically configured screens for the AutoCare solution are described below:

- **AutoCare Application Editor Screen (Parts):** A screen for viewing / editing application records when a part is selected. This screen is mapped to the AC_PIESITEM object type.
- **AutoCare Application Editor Screen (Vehicles):** A screen for viewing / editing application records when a vehicle is selected. This screen is mapped to the AC_Basevehicle object type.

## NAPA

- **NAPA Application Editor Screen (Parts):** A screen for viewing / editing application records when a part is selected. This screen is mapped to the NAPA_Product object type.
- **NAPA Application Editor Screen (Vehicles):** A screen for viewing / editing application records when a vehicle is selected. This screen is mapped to the NAPA_Year object type.

## TecDoc

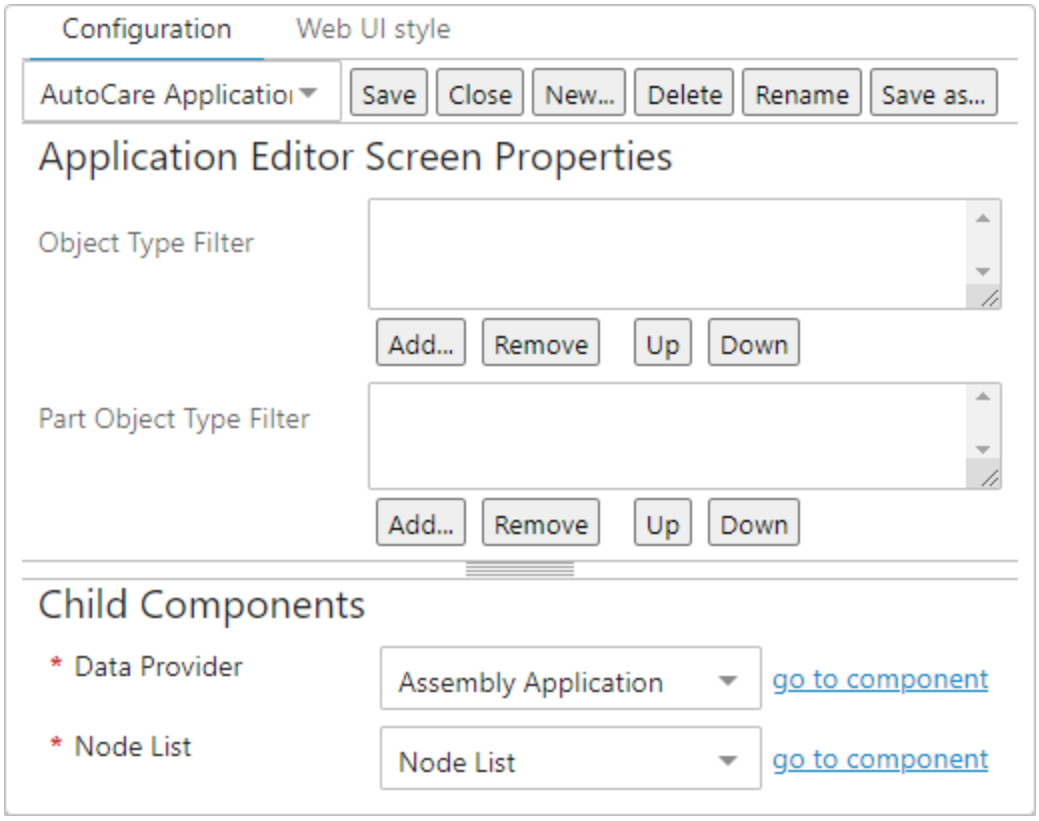
- **TecDoc Application Editor Screen (Articles):** A screen for viewing / editing application records / linkages when an article is selected. This screen is mapped to the TD_DS_SupplierArticle object type.
- **TecDoc Application Editor Screen (Assemblies):** A screen for viewing / editing application records / linkages when an assembly is selected. This screen is mapped to the TD_VehicleType(PC) object type.

## Prerequisites

It is expected that anyone configuring the Application Editor Screen (Parts / Vehicle) 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** topic in the **Web UI Getting Started** section of the **Web User Interfaces** documentation.

## Configuring Application Editor Screen Properties

The Application Editor Screen Properties includes the following parameters:




**Object Type Filter:** This parameter allows users to restrict the Application Editor results by an object type. If left blank, application records belonging to all object type will be displayed.

**Note:** Selecting the Filtered Assembly Application option within the Data Provider parameter requires adding at least one object type in the Object Type Filter parameter.

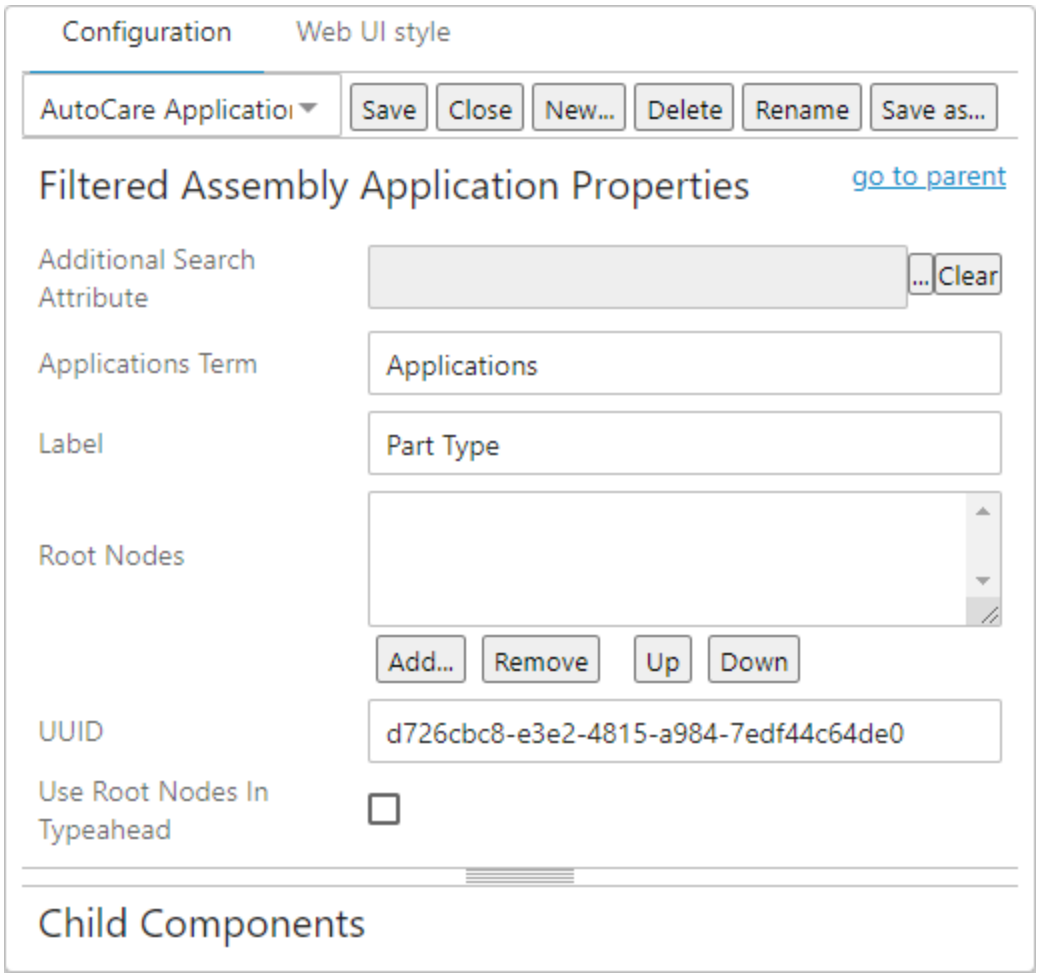
**Part Object Type Filter:** This parameter allows users to restrict the application records displayed in the Application Editor results based on the object type of the applicated part. If left blank, application records belonging to all part object types will be displayed.

**Data Provider:** This parameter is prepopulated either with the Part Application, Filtered Assembly Application, or Assembly Application option.

- The Part Application option is selected for the Application Editor Screen that is used to display the existing application records while selecting a part
- The Assembly Application option is selected for the Application Editor Screen that is used to display the existing application records while selecting a vehicle / assembly
- While the Filtered Assembly Application option is selected for the Application Editor Screen to display the existing application records, it can also search for missing application records for the selected vehicle / assembly.

  Further configuration may be required for the Filtered Assembly Application option, and no additional configuration is required for the other options selected within the Data Provider parameter.

  Select the and click on the 'go to component' link to configure Filtered Assembly Application Properties. Edit the following parameters as necessary.

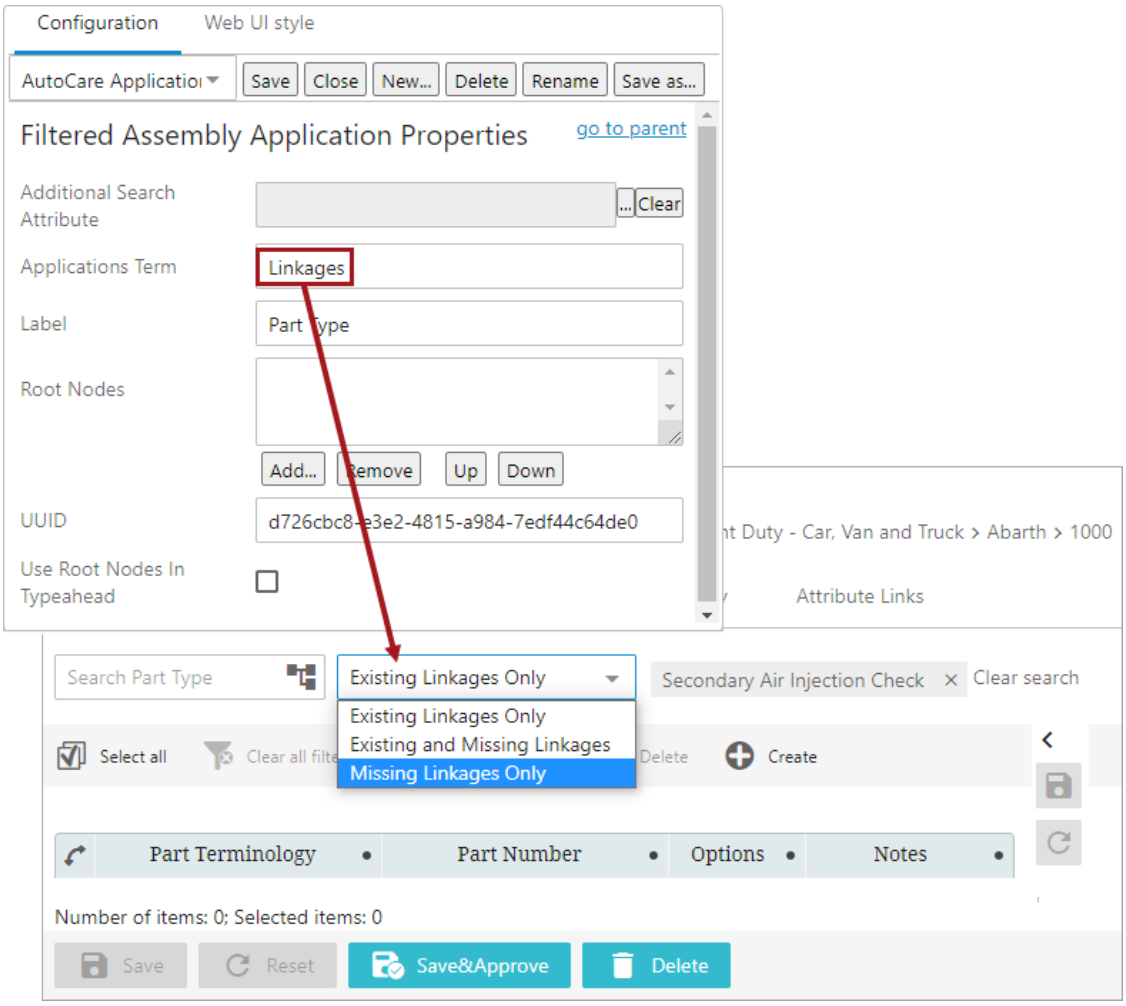



  - **Additional Search Attribute:** This optional field allows users to add attribute criteria to their search. All attribute validation base types (e.g., Text, Number, List Of Values) are allowed. Click the ellipsis button (...) to find and select the attributes. The selected attributes must be valid for the part type

object type, and the attribute value must be populated on the part type in order for it to display correctly in the typeahead search field.

- **Applications Term:** This parameter provides users the ability to change the terminology 'Applications' within the Existing and Missing Applications drop-down list. This parameter is defaulted to be populated with value 'Applications' as shown below.

  Below is an example of the Applications Term parameter populated with the value 'Linkages'.



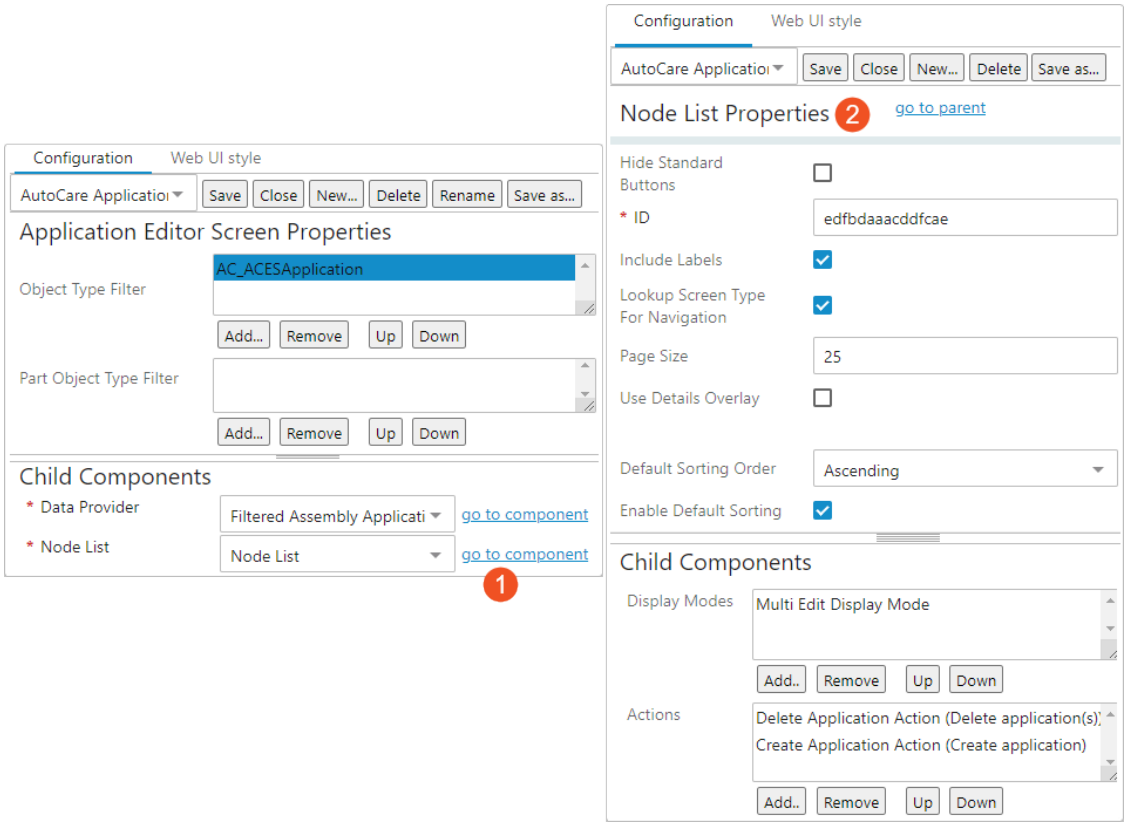

- **Label:** By default, the Label parameter is populated as 'Part Type.' Optionally edit the parameter to a more suitable label.

- **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).

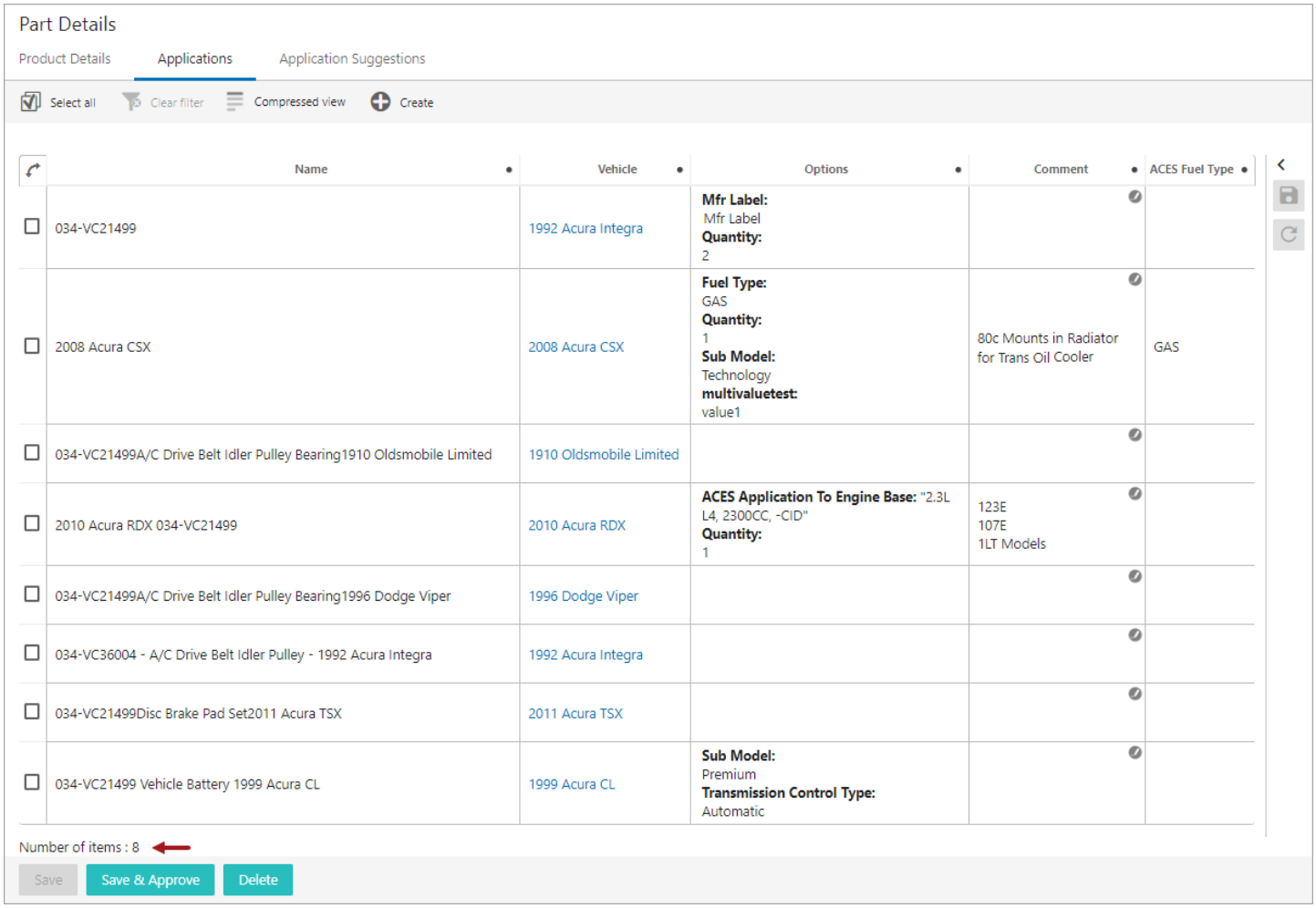
- **Use Root Nodes In Typeahead:** Selecting this parameter allows the display of suggestions in the typeahead field of only those objects that belong in the root node defined within the Root Nodes parameter. However, the user needs to be aware that the selection of this parameter would affect the performance of the display of suggestions in the typeahead field.

**Node List:** The Application Editor Screen Properties is pre-configured with a Node List in Multi Edit Display Mode along with the 'Delete Application Action' and 'Create Application Action' toolbar actions (as shown above).




1. Select the 'Node List' option in the dropdown and click on the 'go to component' link.
2. Configure the Node List component. For more information, refer to the **Node List Component** topic within the **Web User Interfaces** section of **STEP Online Help**.

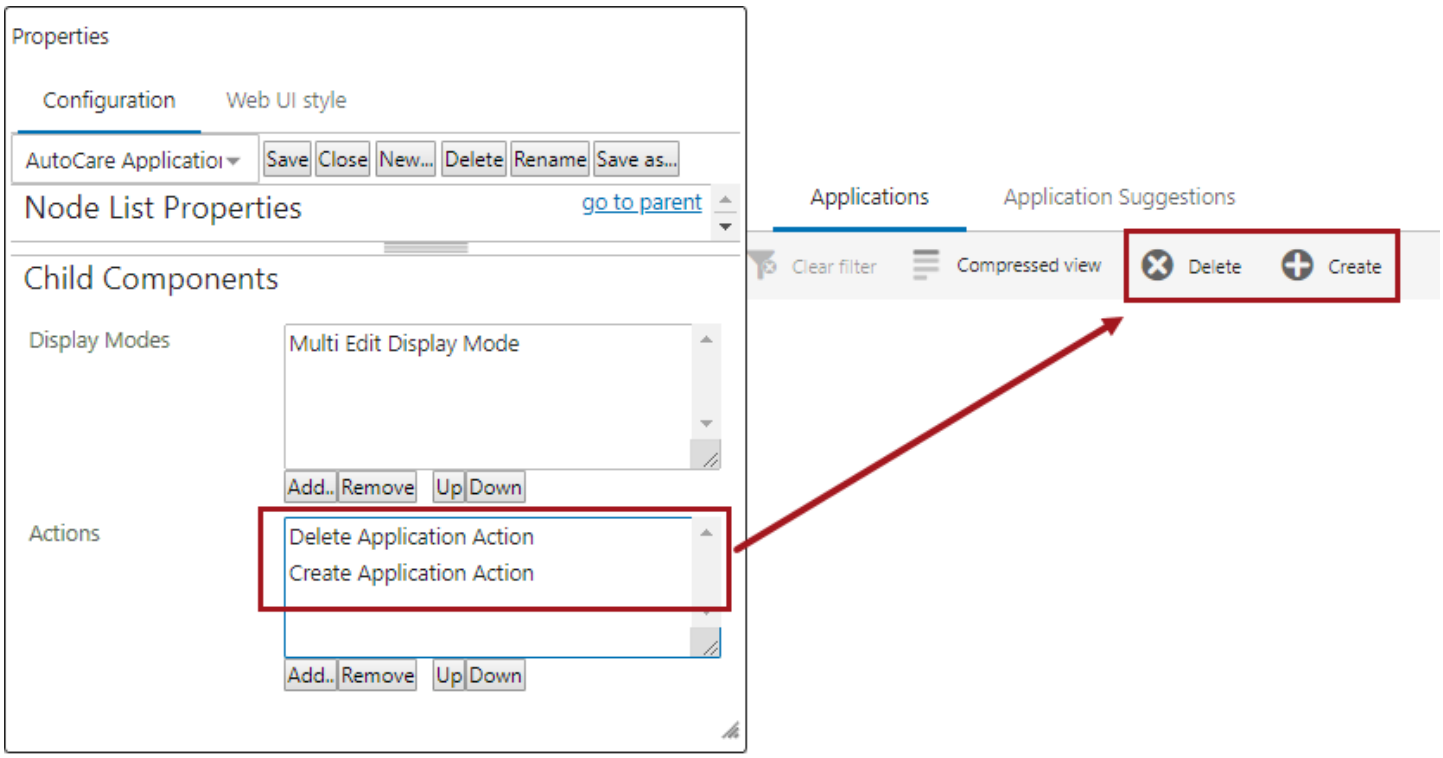
Since the Application Editor Screen is based on the Node List component and the Multi Edit Display Mode, sorting and filtering of the header component values are supported. This allows for multiple information of the application records to be shown in the same table with the column filtering capabilities. Also, the 'Number of Items' shown in the Application Editor Screen is displayed towards the bottom of the results table, allowing a user to review the total number of application records without having to scroll through the data. In the example screenshot shown below, the Application Editor Screen is configured as a tab page within the Node details screen and eight application records are displayed for the selected part.




# Node List Properties

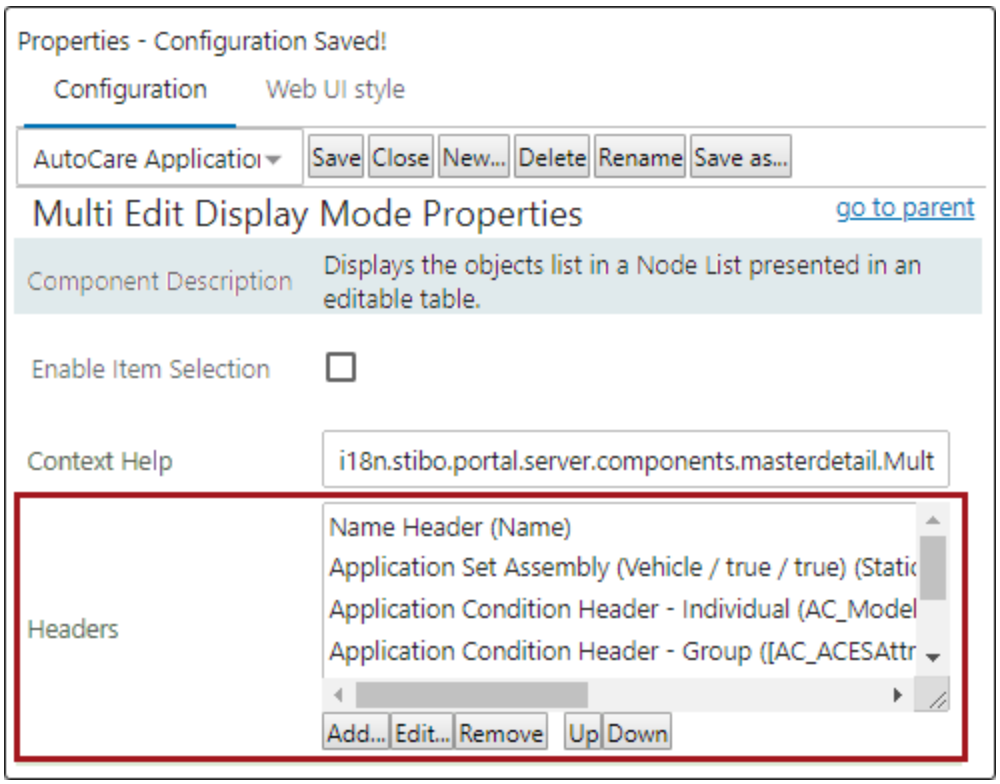
Only the significant information applicable to the Application Editor Screen is detailed below. Additional information about the Node List Properties can be found in the **Node List Component** section of the **Web User Interfaces** documentation.

The following are the parameters available within the Child Components section of the properties screen:




## Display Modes

The **Multi Edit Display Mode** comes pre-configured in this Node List. As the name implies, users can edit multiple application records and multiple attributes in the same screen and save the changes automatically (using the 'Use Immediate Save' parameter) or manually. For details, refer to the **Auto-Save in the Web UI** topic within the **Using Web UI** section of **STEP Online Help**. When the user double clicks the Multi Edit Display Mode component available within the Display Mode parameter, the Multi Edit Display Mode Properties dialog displays. The Multi Edit Display Mode Properties dialog allows to further configure the component based on the user requirements.




### Table Headers

The Headers parameter available within the Multi Edit Display Mode Properties allow users to add multiple table header components that gets listed as individual columns in the results table of the Application Editor Screen.

Below are some header examples. The bullet under each header explains what is displayed in that header column. Additional headers may be used (e.g., Name, Attribute Value, Attribute Value Group). When adding headers, you can read a description of each before adding them to the Headers field.

> **Note:** It is suggested that the **Application ID Header** be used in place of the ID Header to display the ID of the application records.

- **Application Condition Header - Individual** - Table header expanding to show all 'important' conditions for an application. For details, refer to the **Application Condition Header Components** topic.
- **Application Condition Header - Group** - Table header showing attributes that serves as conditions for an application. The attributes could be on the application object, the child object to the application or on the reference from the application object to the assembly object. For details, refer to the **Application Condition Header Components** topic.
- **ACES Application Qualifiers** - Table header that shows AutoCare application qualifiers. For details, refer to the **ACES Application Qualifier Table Header** topic
- **Application Approval Status** - Table header that shows application approval status.

- **Application Assembly Value** - Table header that shows the value of an attribute of an assembly linked to an application. For details, refer to the **Application Assembly Value Table Header** topic.
- **Application Asset Reference** - Table header that shows the asset reference for an application.
- **Application Comment** - Table header that shows comments.
- **Application Competitor or OE Part Numbers** - 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 Country Handling Table Header** - Component used to display and edit country handling data within the Application Editor Screen table.
- **Application ID** - Table header that shows the ID of an application.
- **Application Part Type Title Header** - Table header that shows the name of the part type that is linked to the applications in the table.
- **Application Part Type Value** - Table header that shows the value of an attribute of a part type that is linked to an application.
- **Application Set Assembly** - Table header that shows the name of the assemblies linked to the applications in the table. For details, refer to the **Application Set Assembly Table Header** topic.
- **Application Set Part** - Table header that shows the name of the parts linked to the applications in the table.
- **Application Suggestion Info** - Table header component used to display information for why the application record is being suggested.

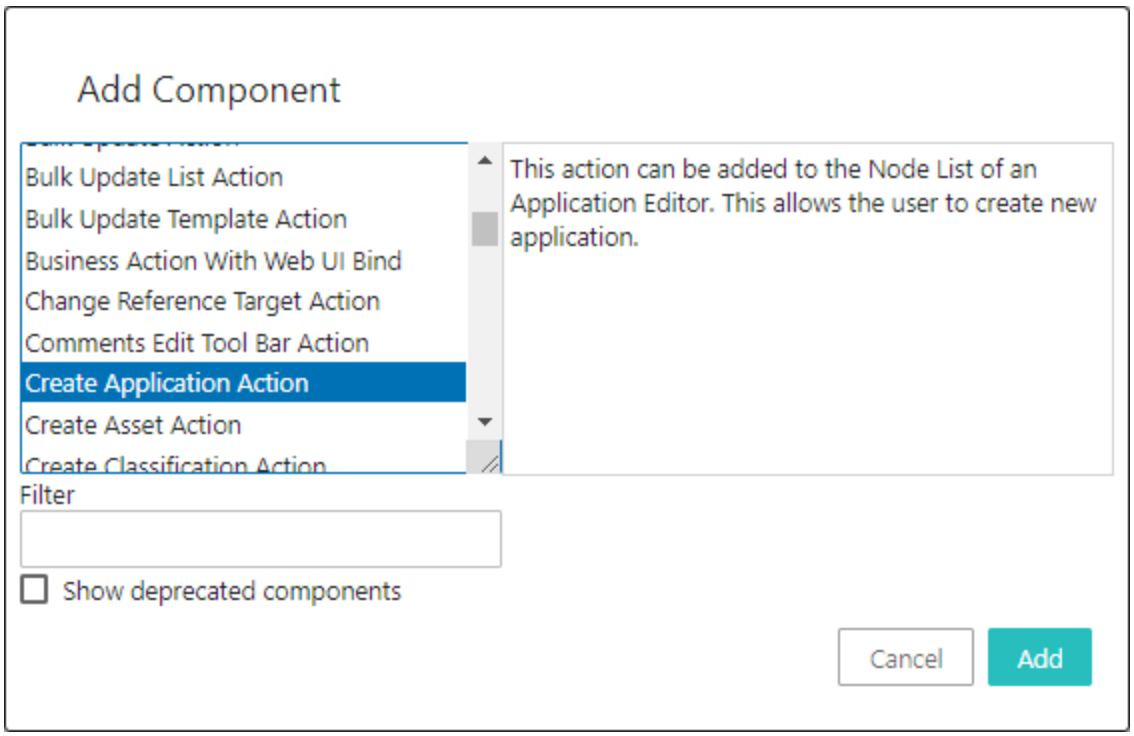
  While configuring the headers in Display Mode properties, a user can choose to set up the table sort order based on a specific header. By editing a particular header's properties, sorting order (ascending or descending) can be designated using the dropdown next to Table Sorting. Once the properties are saved, the table will sort itself based on the new header settings, and the Display Mode properties will be retained until configured differently.

  The Table Sorting designation can only apply to one header at a time. It has to be removed from one header before setting up on another. Additionally, if 'Enable Default Sorting' is enabled / selected in the Node List properties, selecting a header to sort by overwrites this setting.

## Actions

Many action buttons can be added to a screen to be used with a Node List. These are all added and configured within the **Actions** field on the Node List Properties screen. The **Create Application Action** and the **Delete Application Action** come pre-configured. These actions and the Bulk Application Updates action are described below. Other actions can be configured based on the user requirements.

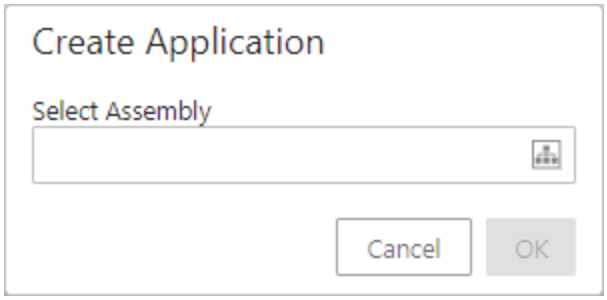
After clicking the **Add** button, the 'Add component' dialog is displayed. Selecting a component in the left area will display the description for that component in the right area.



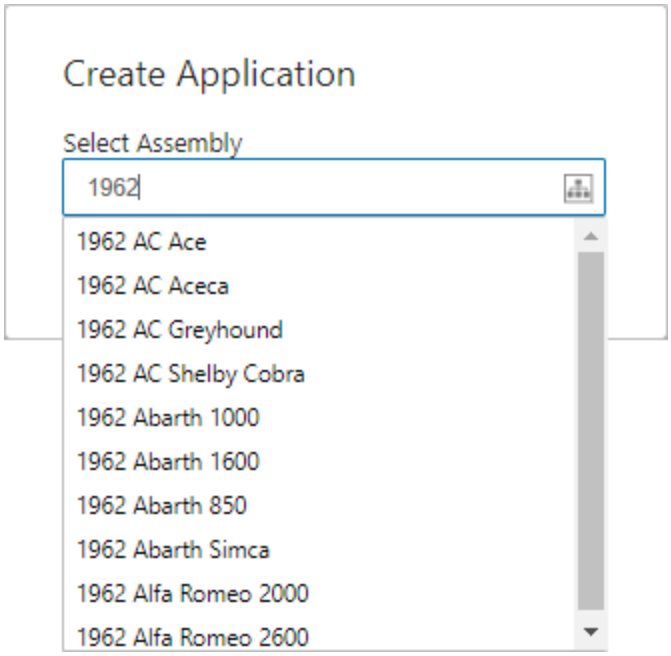

### Create Application Action

The **Create Application Action** button ( ) opens a 'Create Application' dialog with Select Assembly (or Select Part depending on the type of Application Editor Screen) option.






When a user clicks on a node picker icon (as shown in the screenshot above), they are presented with standard Browse and Search tabs to select an existing vehicle / assembly or part. The 'Select Assembly' and 'Select Part' value fields support typeahead functionality, which allows users to easily add a vehicle / assembly or part without having to click through various hierarchies and/or folders.




Additional functionality can be configured that allows users to set a name of the newly created application records with a unique name format, search for vehicles / assemblies or parts in an effort to only select below the defined hierarchy, or narrow the search result in the Faceted Search window to aid efficient vehicles / assemblies or parts selection. For more details regarding how to fully configure the properties for the Create Application Action component, refer to the **Create Application Action** section of the **Automotive Reference Guide** documentation.

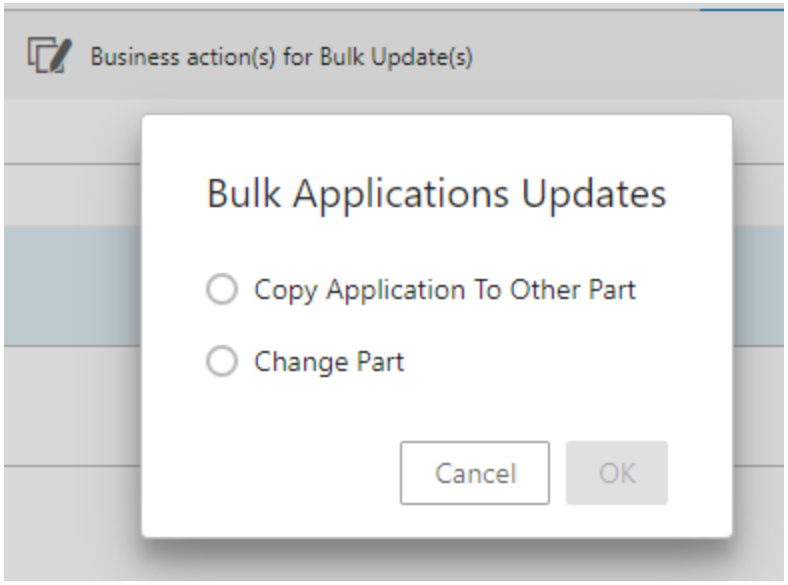
### Delete Application Action

The **Delete Application Action** ( ) is enabled when an application record is selected in the table. When clicked, it will delete the selected application record from the table without showing any warning dialog.



### Bulk Applications Updates

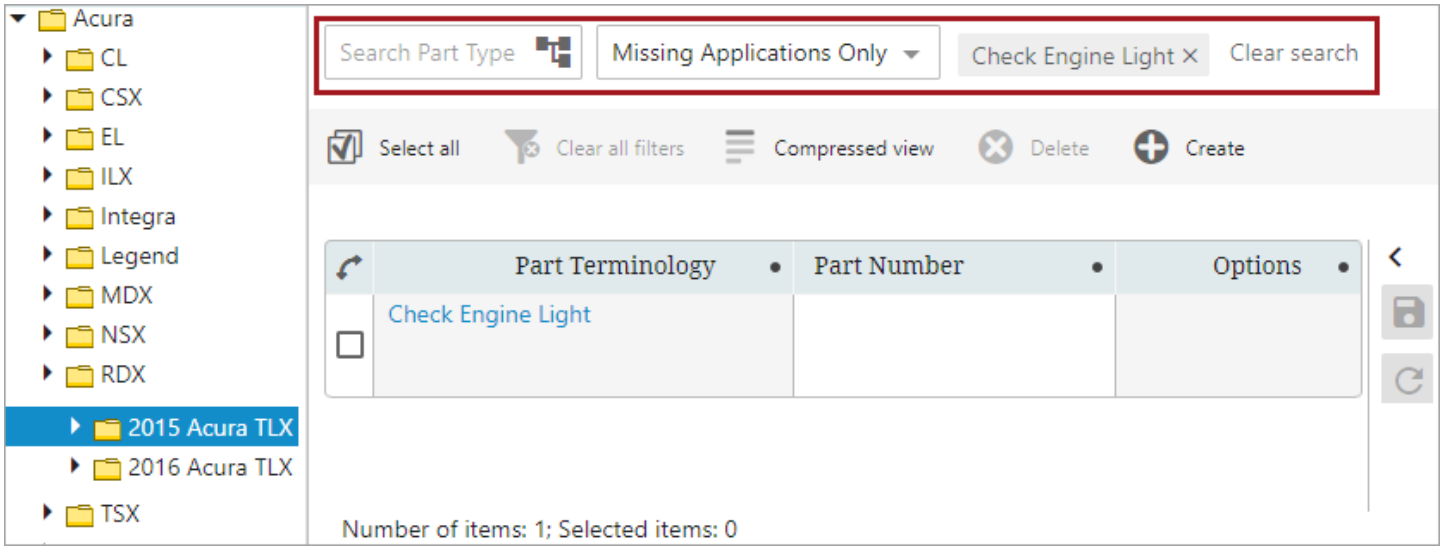
The **Business action(s) for Bulk Update(s)** button is enabled when one or more application records are selected in the table. When clicked, it will update multiple application records in a single operation. Users have the flexibility to add any number of bulk updates. If more than one bulk update is configured then the Bulk Applications Updates dialog will display (as shown below), prompting users to select the desired bulk update to be executed. For more details regarding how to fully configure the properties for the Create Application Action component, refer to the **Bulk Applications Updates** topic of the **Automotive Reference Guide** documentation.




## Searching for the Missing Application Records

The Application Editor Screen (Vehicle) can search for missing application records for the selected vehicle / assembly only if the Data Provider parameter is configured with the Filtered Assembly Application option.

Selecting the Filtered Assembly Application option displays search fields on top of the Application Editor Screen (as shown below). Users can enter the part type(s) to define their criteria to search for the missing application records.




## Create Application Action Component

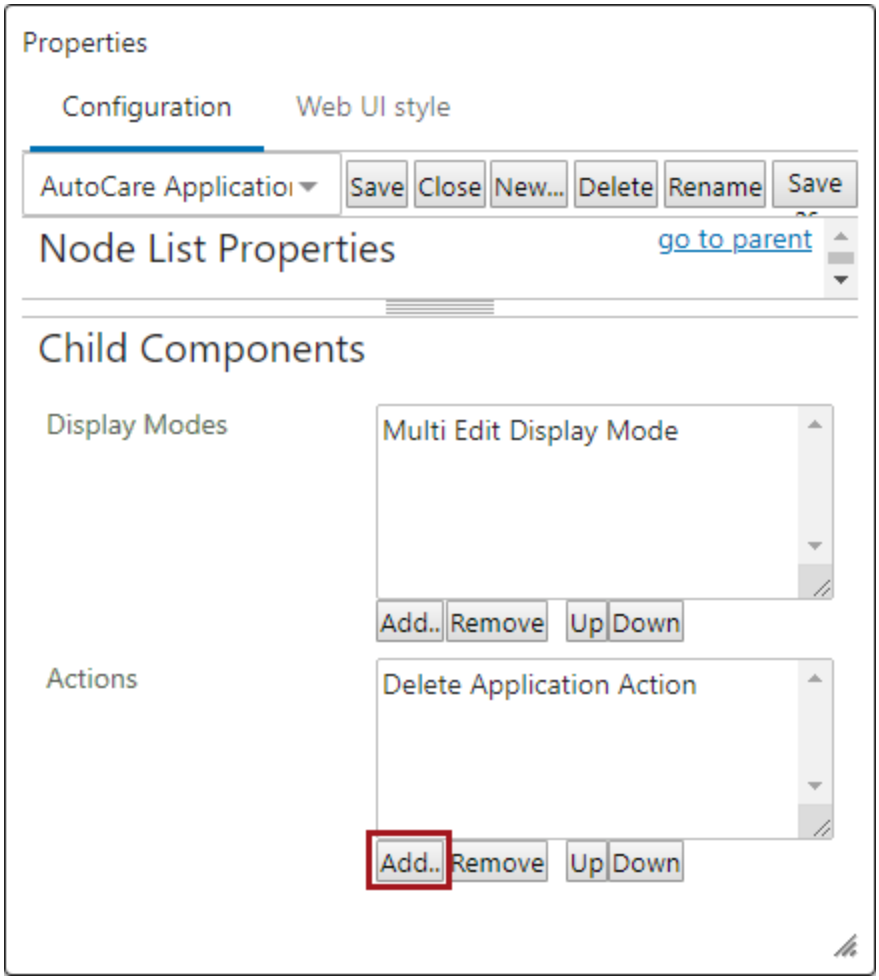
The Create Application Action component enables users to create application records for the selected vehicle / assembly or part. Meaning that this component applicates the selected part to the vehicle / assembly or vice versa. This component is available as a toolbar action for Node Lists, and one common usage is within the

Application Editor Screen.

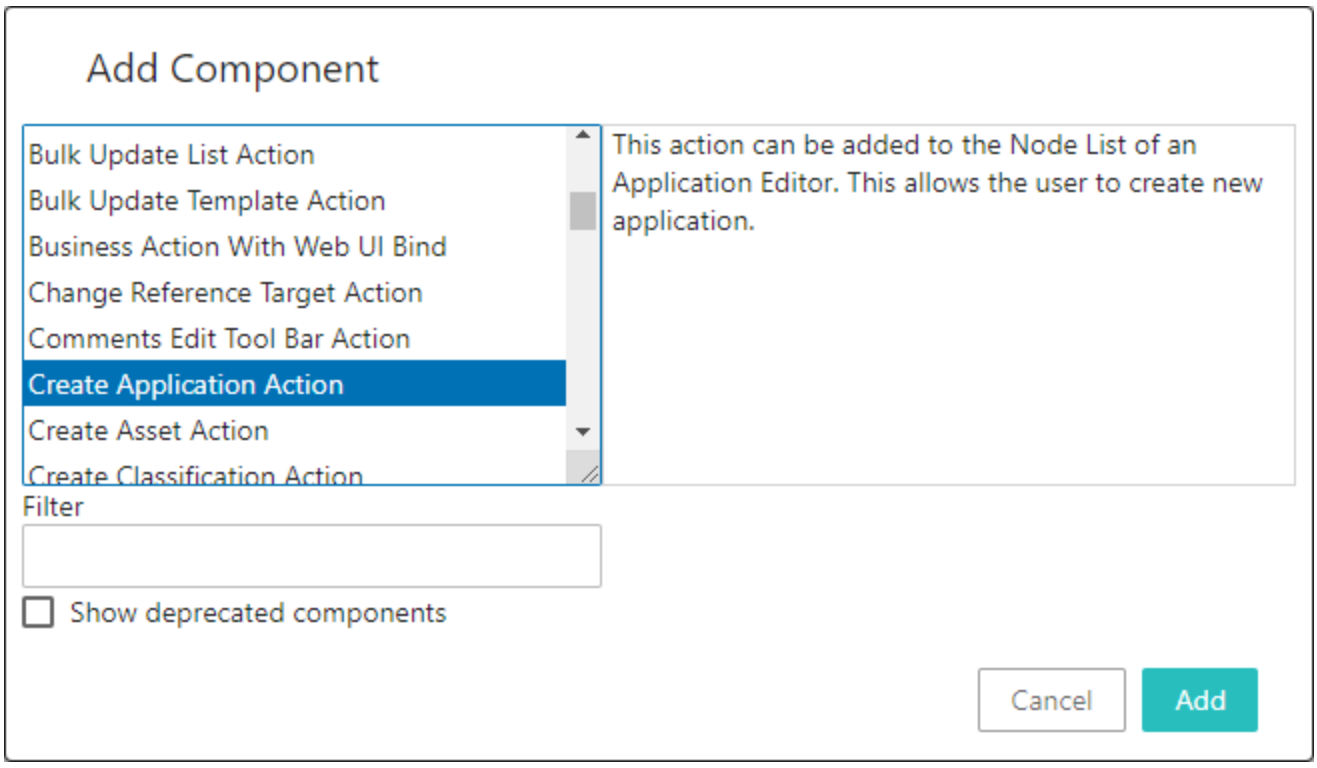
## Configuring the Create Application Action Component

When Easy Setup actions are used to create a standard data model, Create Application Action component is added in the Application Editor Screen by default. It is the ready-to-use component that allows users to create application records. Users can further configure this component to their requirements. Follow the steps described below to add and configure the Create Application Action component as a toolbar action on a Node List, or skip to step 3 to directly configure the available Create Application Action component.

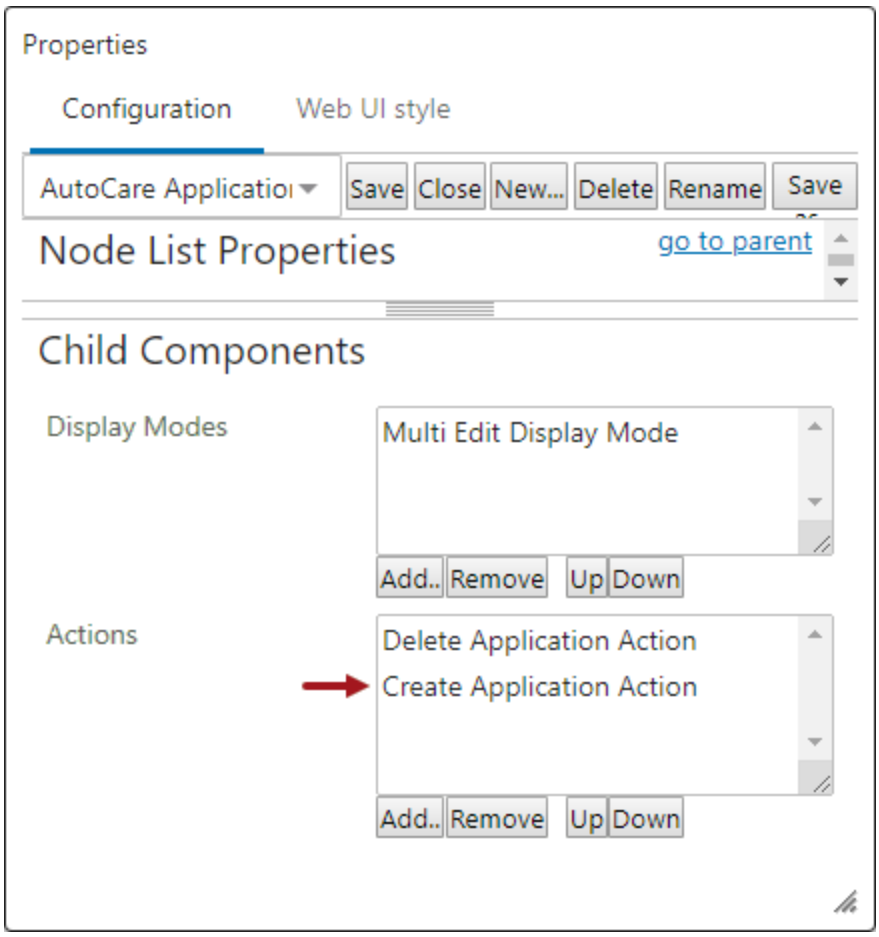
1. First, add the component. Under Child Components on the Node List Properties screen in the designer, click the **Add** button under the field for Actions parameter.



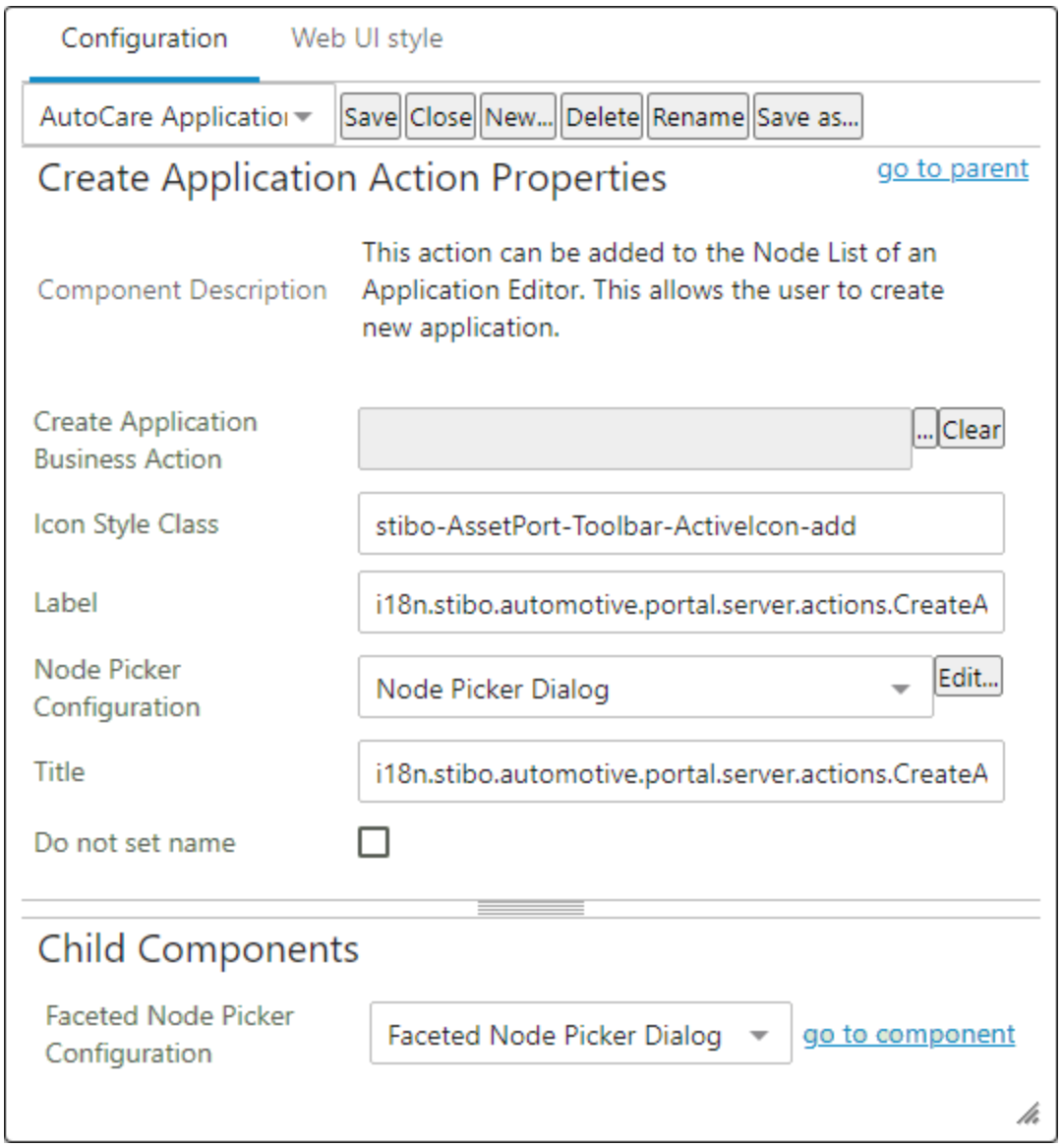

2. Then select **Create Application Action** from the component list. Click **Add**.




3. Once added to the list of actions for the Node List, double-click the **Create Application Action** component to configure.




4. In the Create Application Action Properties dialog, define the following optional parameters:




- **Create Application Business Action**: This parameter lets users define a business rule that will be executed on all new applications. One example is to configure a business action that sets the STEP object Name of the newly created application records. When using a business action to set the application record name, it is required that the 'Do not set name' parameter is selected so that the name is only set through the configured business action.
- **Icon Style Class**: This parameter allows a custom icon to be displayed on the action button.
- **Label**: To change the default button label, add the custom text to the Label field. If nothing is added, the Label default will display as 'Create.'

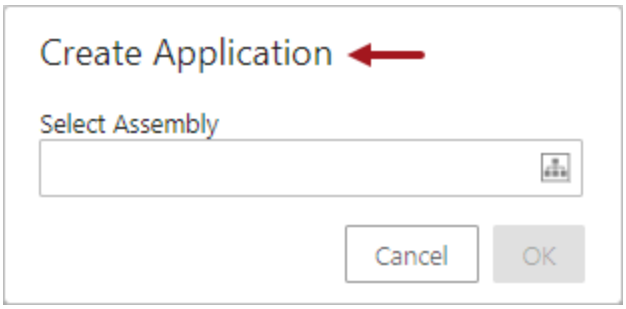
  This is how the button will display at the top of the Application Editor Screen:

- **Node Picker Configuration**: This parameter allows users to define root nodes, search tab pages, and a custom title. For details, refer to the **Node Picker Dialog** documentation within the **Web User Interfaces** section of **STEP Online Help**.

  **Note:** The settings made in this parameter are overwritten when the Faceted Node Picker Configuration parameter is populated with the Faceted Node Picker Dialog component.

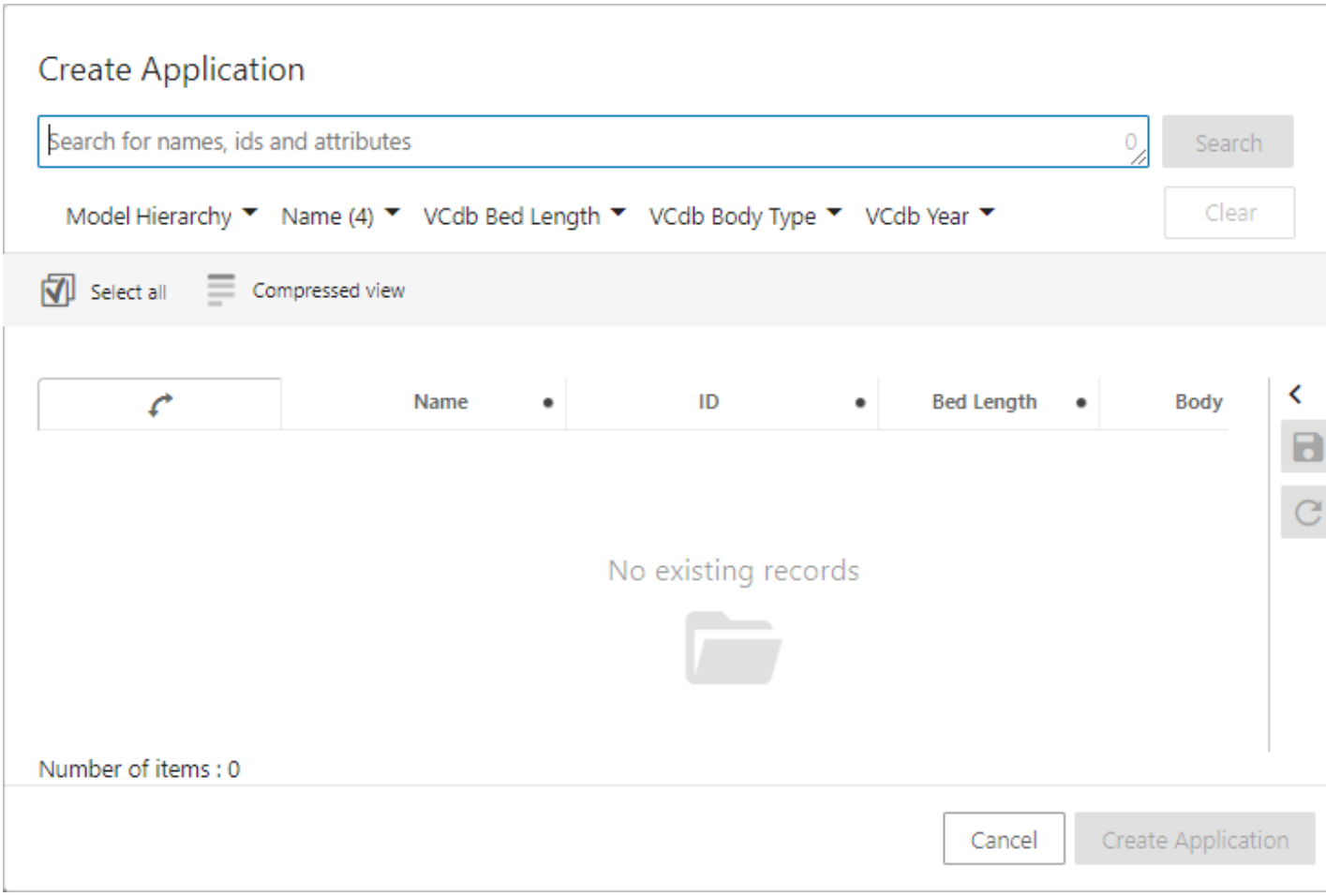
- **Title**: To change the default title of the dialog that appears when a user clicks on the Create Application Action button, add the custom text to the Title field. If nothing is added, the default title will display as 'Create Application.'

  This is how the title will display at the top of the dialog when a user clicks on the Create Application Action button:




- **Do not set name**: This parameter provide control over the naming convention of the application records that are created via the Create Application Action button. If unselected, the STEP name of the application record will be created using the default combination of Part Number, Part Type, Year, Make, and Model from the system. Select this parameter if the STEP Name of the newly created application record should be left empty, or if it will be set by the business action that is defined in the Create Application Business Action parameter.
- **Faceted Node Picker Configuration**: This parameter, when configured with Faceted Node Picker Dialog component, acts as a replacement for the regular Node Picker component where the user can search for the required objects by narrowing down the search result using facets. The faceted search fields allow users to search for vehicle or part objects based on the object Name, ID, or attribute values. Users can configure the faceted node picker to use all available attributes, IDs, or Names. For details, refer to the **Faceted Node Picker Dialog** documentation within this guide.

  Below is an example of how the Faceted Node Picker dialog will display when the user clicks on the Create Application Action button:



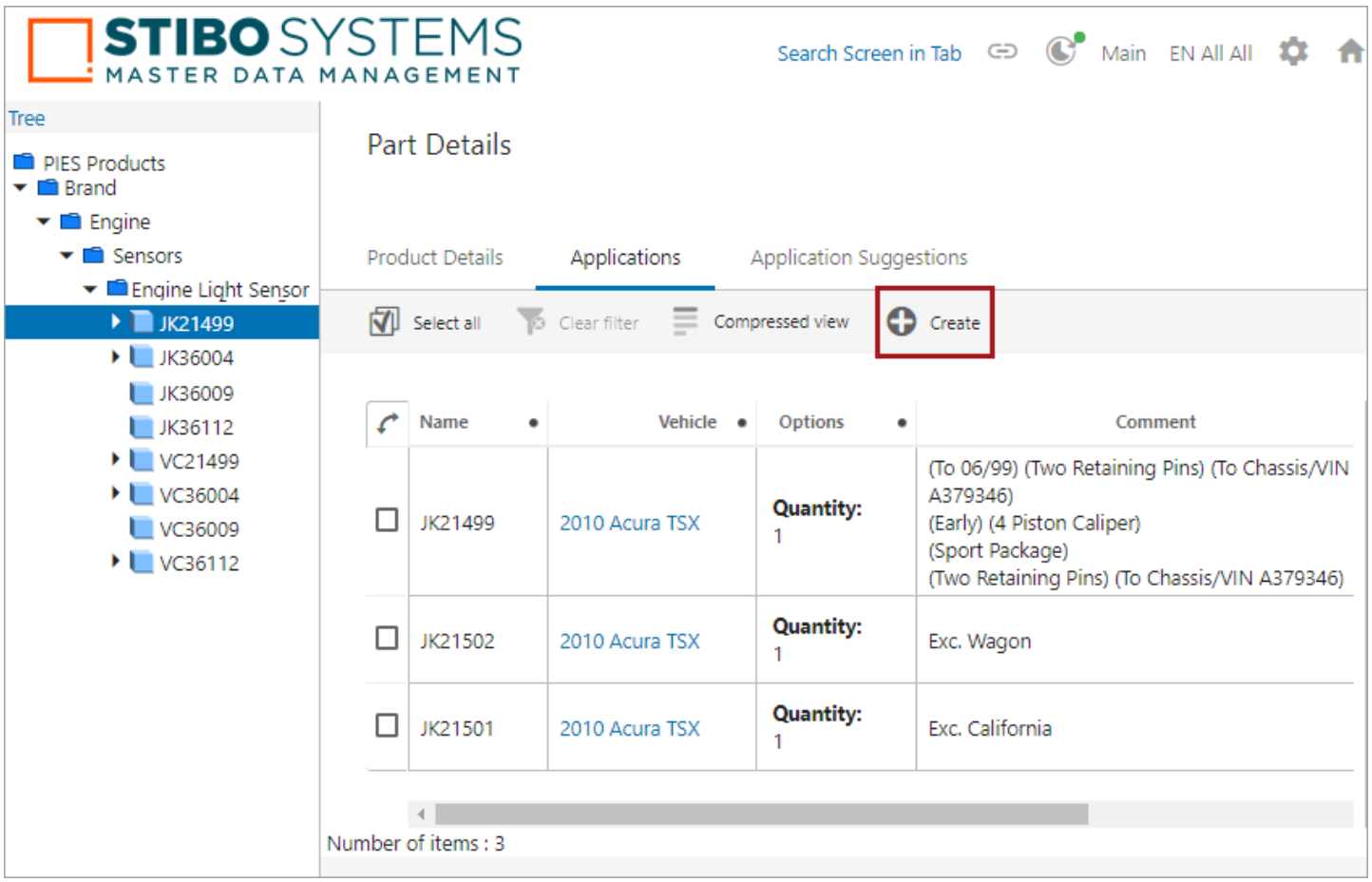

## Creating applications using Create Application Action

The central feature of the Application Editor Screen is the ability to create an application record on the selected vehicle / assembly or part (depending upon the type of Application Editor Screen). Follow the steps described below to create an application record for the selected part in a Application Editor Screen. The example in this document considers applicating the Part to the vehicle / assembly.

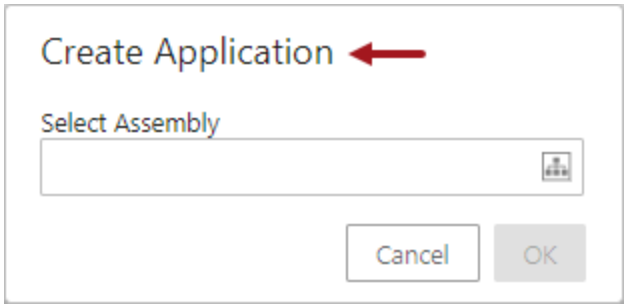
1. Select the part that application records should be created for.

   **Note:** Application Editor Screen must be configured to display either as the Main Screen or as a Sub Screen Tab Page when the part is selected.

   In this example, the part 'JK21499' is selected and the Application Editor Screen displays the existing application records belonging to that part.



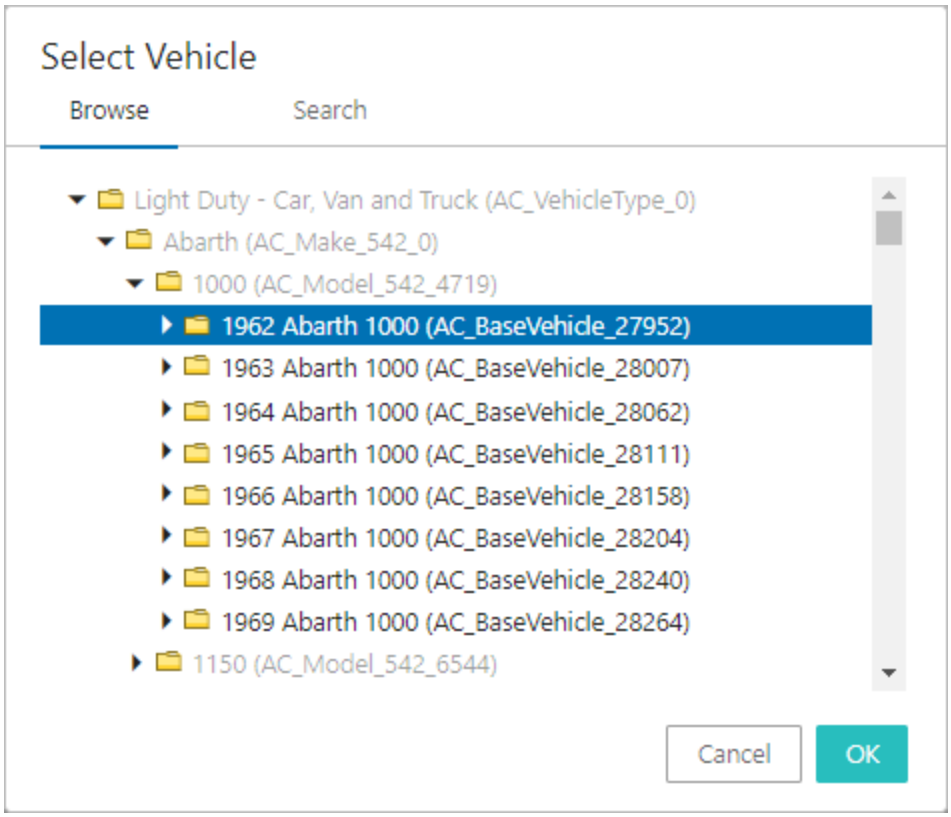

2. Click the **Create** button that is available in the toolbar of the screen and the Create Application dialog will be displayed as shown below:



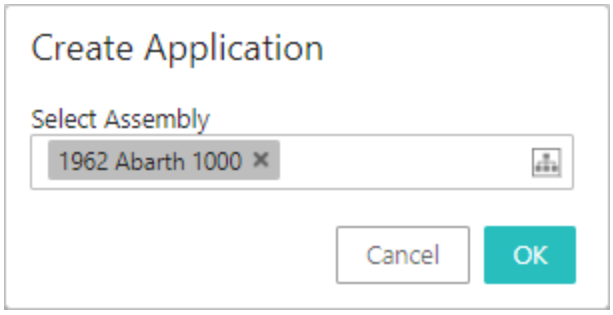

**Note:** In the above example, a regular Node Picker Dialog is displayed. However, the user can configure it to display the Faceted Node Picker Dialog instead of the regular node picker. For details on how to use the Faceted Node Picker Dialog, refer to the **Faceted Node Picker Dialog** documentation within this guide.

3. Click on the Node Picker icon and select the vehicle / assembly that should be applicated to the selected part either by navigating through the standard Browse tab or by searching through the Search tab. In the example

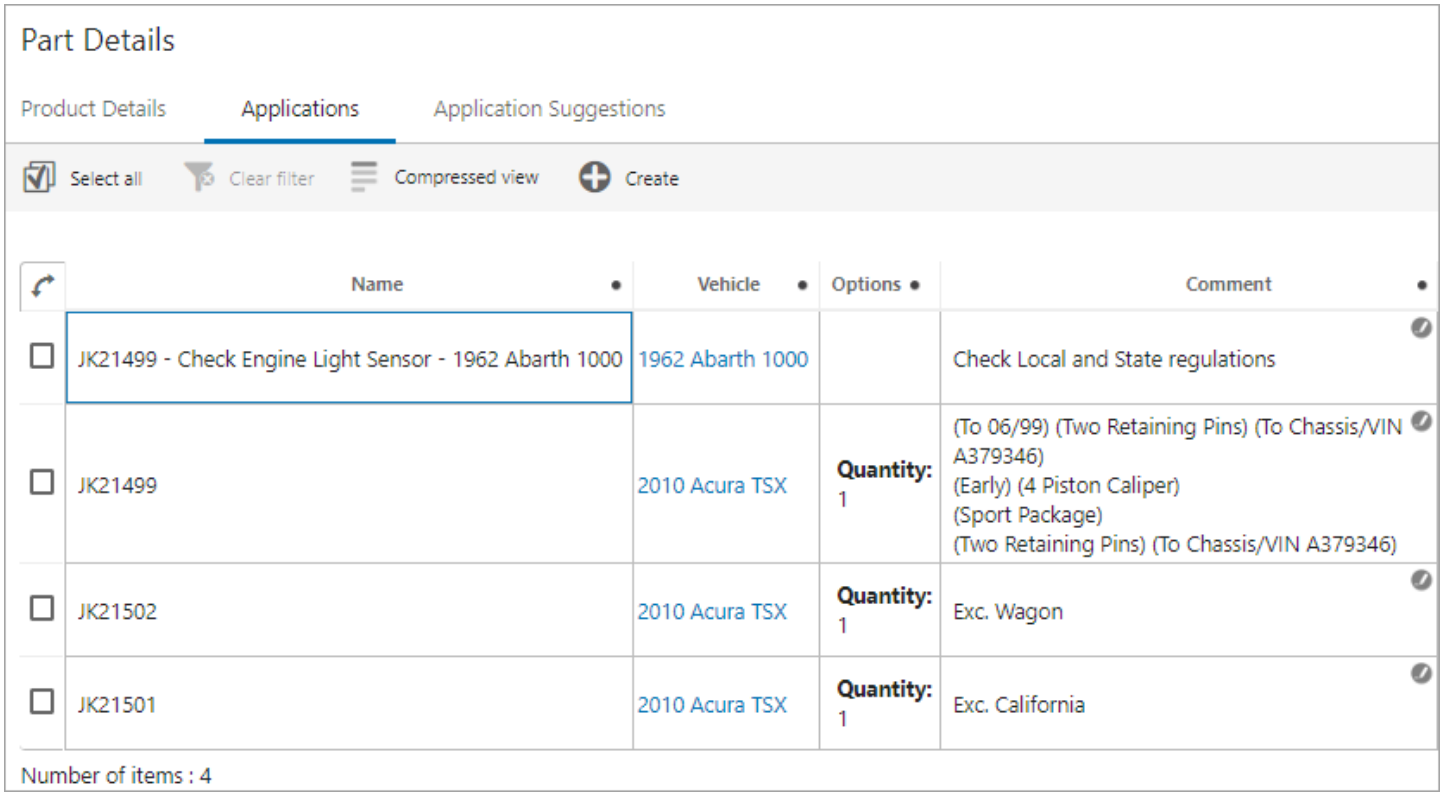
below, vehicle '1962 Abarth 1000' is selected. Click **OK** to close the Node Picker dialog.




4. Click **OK** to close the Create Application dialog.




The result is that the part 'JK21499' is applicated to vehicle '1962 Abarth 1000' and the application record 'JK21499 - Check Engine Light Sensor - 1962 Abarth 1000' is created and listed in the Application Editor Screen (as shown below).



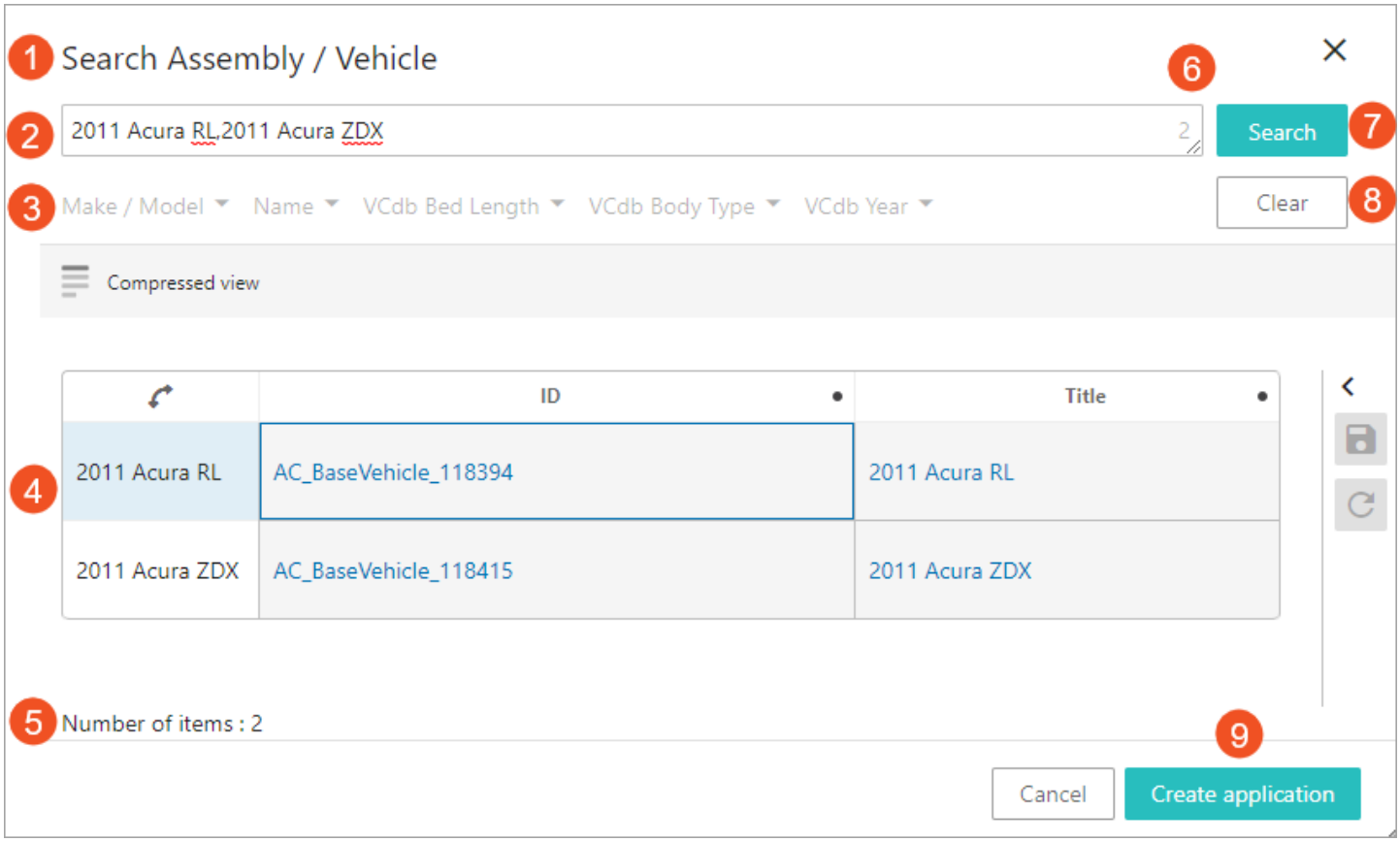

## Faceted Node Picker Dialog

The Faceted Node Picker Dialog is a replacement for the regular Node Picker dialog that is available to be used within the Create Application Action button that is configured in the Application Editor Screen or within the Faceted Node Picker Parameter component that is configured in the Bulk Applications Updates component. The faceted search field uses the sync between search cards functionality to make it possible for the user to narrow down the search result and only display relevant targets (assembly / vehicle or part). In Web UI Design mode, users can configure searches for vehicle / assembly or part objects based on IDs, Names, or attribute values.

The Faceted Node Picker dialog can be configured to define the most likely used filter groups for the search in the Web UI. As an example, a user has selected a part and needs to applicate that part to a vehicle. The user would possibly need to find the vehicle based on Make, Model, Year, or any attribute value that is valid for the vehicle object. For the ease of access, if a facet is configured and made available for each of these search criteria in the Faceted Node Picker dialog, then users can refine the list of objects based on one or more of the available filters. Additionally, a configurable free text search bar is also provided within the dialog so that the users can type in the IDs, Names, or attribute values to search on.

### Faceted Node Picker Dialog Interface Overview

The top section of the Faceted Node Picker dialog offers features related to the free text search field. The section below the free text search field is dedicated to search by using facets. Both of the types of searches that are offered in the dialog can benefit users in their own way. The Faceted Node Picker Dialog is composed of the following:




1. **Title:** Users can edit the Title in Design mode with any text string.
2. **Free text search field:** Search for objects (assembly / vehicle) by manually typing in the ID / Name / attribute value in the free text field. Users can type in multiple lists of Names and IDs separated by a comma. When the comma-separated search is executed, the search result will be displayed with the IDs, Names, or attribute values that match the input from the free text search field. Users can copy (IDs / names / attribute values) from an Excel spreadsheet and paste directly to the free text search field to search for results.
3. **Facets:** Use facets to narrow down the search result and only display relevant targets (assembly / vehicle or part). Users can configure the facets to use attributes, IDs, or Names. All of the configured facets are synchronized and the search criteria can be entered into the facets in any order, and the available search criteria will display in the typeahead dropdown based upon the criteria that have been entered within the other facets.
4. **Results table:** Displays the list of search results. A Node List component is used to display the results of selections made within the dialog. Each column label can be edited as needed by accessing the Faceted Node Picker Node List Properties Child Component Headers parameter.
5. **Count of search results:**The total count of the search result is displayed.
6. **Total number of search entries:** View the total number of object IDs / object names / attribute values that are typed in the free text field.

7. **Search button:** When clicked, the Search button will use the criteria that have been provided within the facets / free text field, and then executes the search to provide any results in the results table.
8. **Clear button:** Allows users to clear all search criteria, characters typed in the free text field, and results displayed in the results table. The Clear button remains disabled until at least one character is typed in the free text field or at least one criterion for one of the facets is populated. Once any unnecessary criteria are removed, the Search button must be used to update the results table.
9. **Create application button:** Create an application record for the selected assembly / vehicle by clicking 'Create application' button.

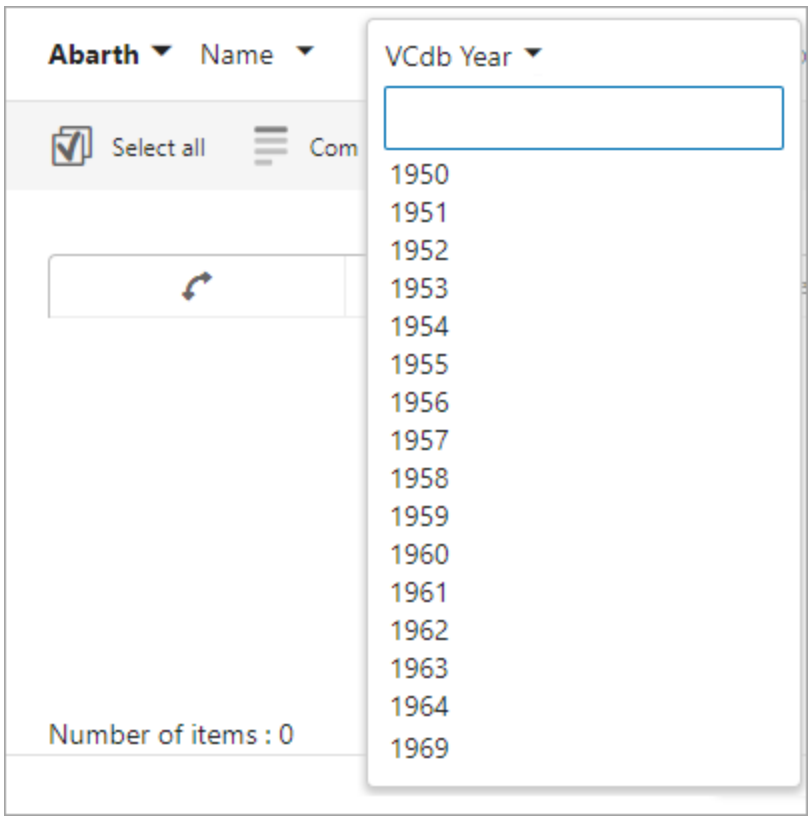
### Using The Faceted Node Picker dialog

The Faceted Node Picker dialog can be used to search in two ways. First, by using the free text field where users can type in multiple lists of Names, IDs, and/or attribute values separated by a comma. Second, by defining the search criteria using facets and narrowing down the search results. The free text field and facets cannot be used simultaneously. When the user types any character in free text field, the faceted search field is disabled (grayed out).

Users need to type in the exact characters of the Name, ID, or attribute value (not case sensitive) in the free text field. Failing to enter the exact characters will not find the expected results.

The Faceted Node Picker dialog can be configured to display any number of facets. Within each facet, a typeahead field is used to view and select valid search criteria. Because only valid criteria can be selected from each dropdown list, the Faceted Node Picker provides automated intelligence for each search. Because of the synchronization of the search boxes, valid criteria are considered upon each selection and within any of the displayed facets, no matter the order of entry.

For example, if the Make / Model facet contains a vehicle that was only produced from 1950-1969, then the dropdown within the VCdb Year facet will only display those available years for selection.




### Configuring Faceted Node Picker Dialog Properties

Faceted Node Picker dialog when configured provides a plethora of possibilities to define a vast range of search criteria which the user can use to narrow down the search results to meet various requirements.

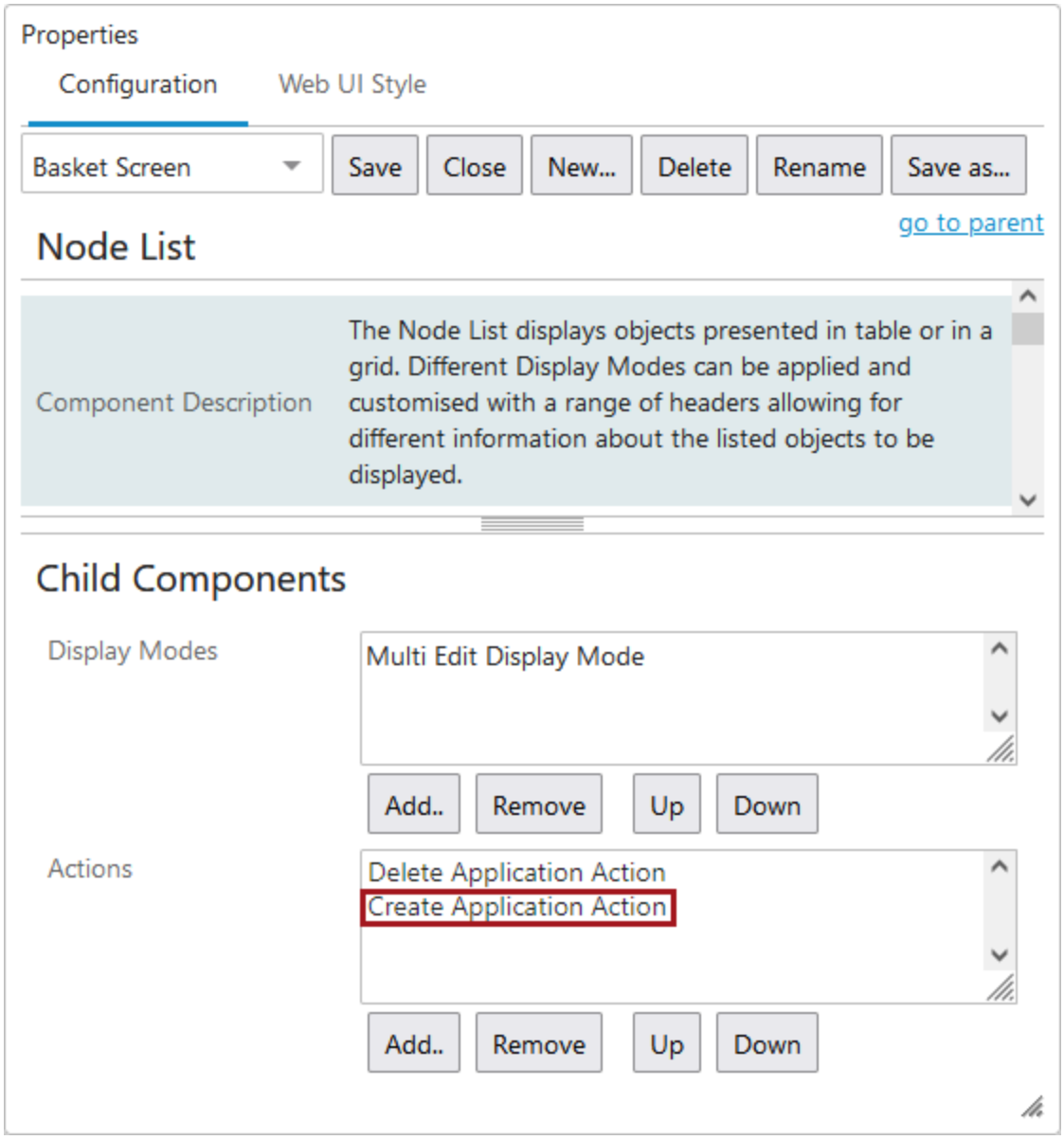
### Prerequisites

To configure this functionality, a user must already have the Application Editor Screen (Parts / Vehicles) with the Create Application Action button configured in it. For information on setting up the required screens and buttons, refer to the **Application Editor Screen** topic within this guide.

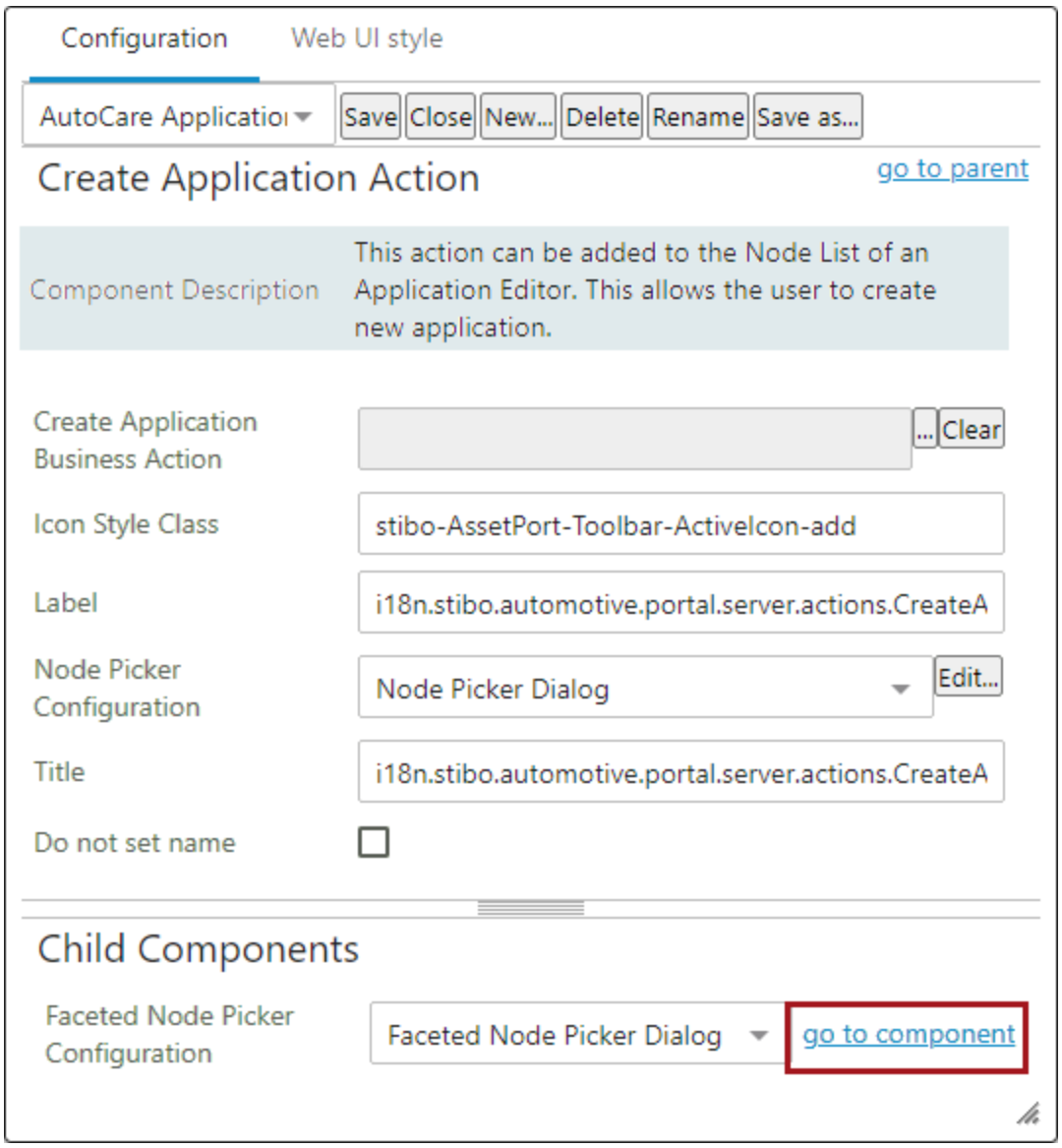
### Configuration

In the following example, the faceted node picker is configured on the Create Application Action button in an Application Editor Screen. The functionality of the Faceted Node Picker dialog differs based on the type of Application Editor Screen it is configured in. For the Application Editor Screen configured on the Part object, the Faceted Node Picker dialog is to be configured to search for an assembly / vehicle as target. And for the Application Editor Screen configured on the vehicle / assembly object, the Faceted Node Picker dialog must be configured to search for a Part object as target. The AutoCare standard will be used as an example throughout this document.

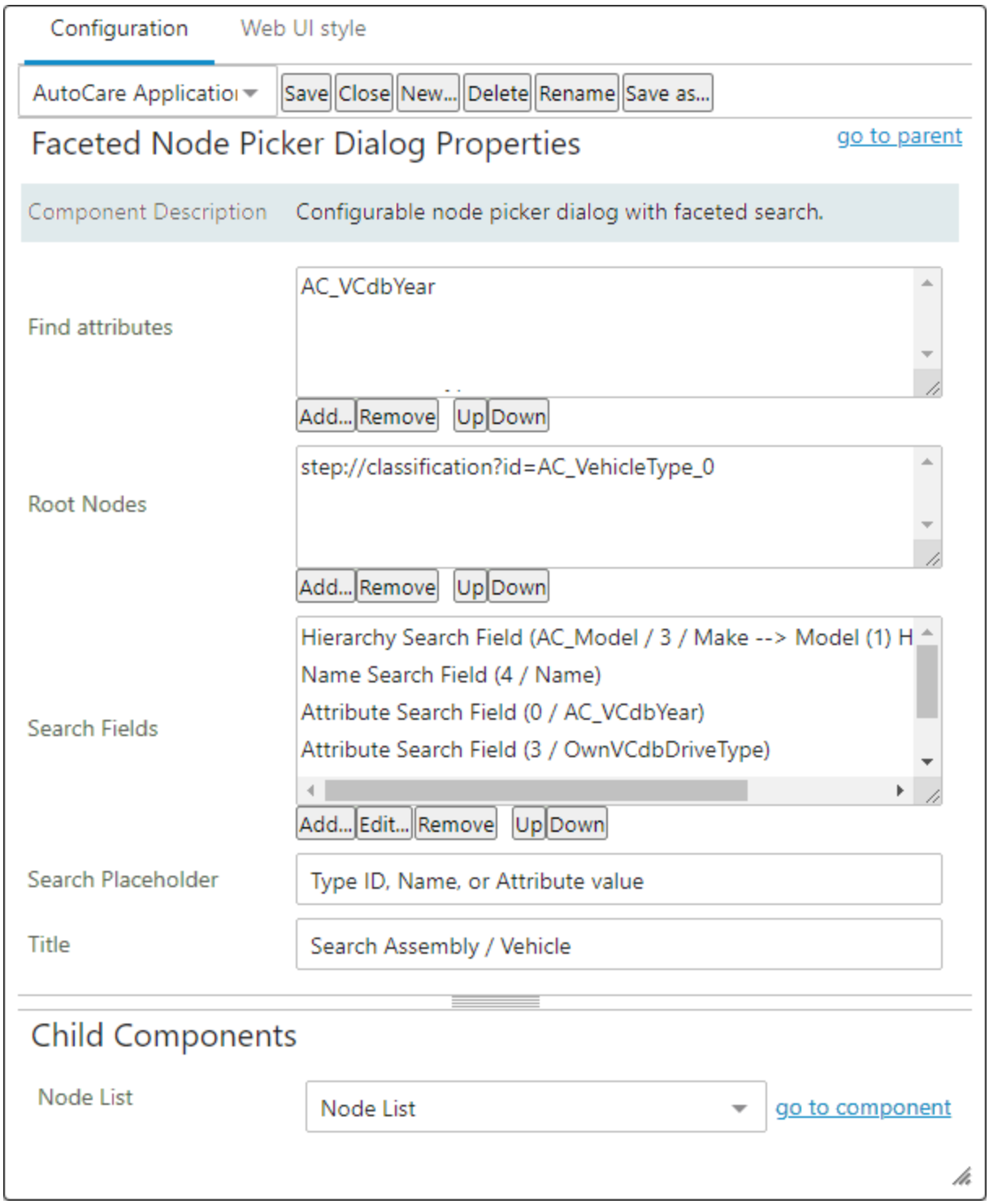
1. In the Web UI Designer mode for Application Editor Screen, double-click 'Create Application Action' in the Actions field of the Child Components section.




2. In the 'Create Application Action' properties, the 'Faceted Node Picker Configuration' option is available in the 'Child Components' section. Select the 'Faceted Node Picker Dialog' option from the list, then select 'go to component.'




3. Shown in the screenshot below is the Faceted Node Picker Dialog Properties. Populate the parameters that are listed below.

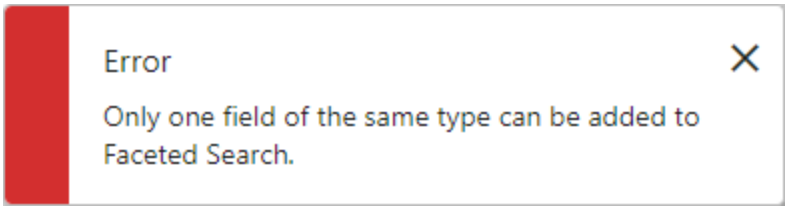



- **Find attributes:** This parameter allows multiple attributes to be added so that the vehicle / assembly (or part) can be searched based on their attribute values in the free text search field available within the Faceted Node Picker dialog. By default, the search criteria in the free text search field are defined to search based on the ID and Name of the object (vehicle / assembly or part). Adding the option to search based on the attribute values gives the user a broader criteria to search through the free text search field. Click the

**Add** button to find and select the required attributes. The selected attributes must be valid for the object type of the objects that are intended to be searched on. For example, if Base Vehicle objects should be displayed in the search results, then the attribute must be made valid for AC_BaseVehicle object type.

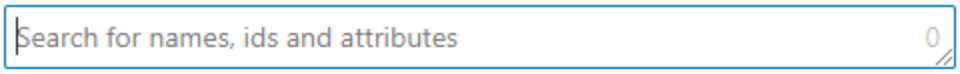
- **Root Nodes:** Allows for root nodes to be added to restrict the vehicle / assembly or part search within the Faceted Node Picker dialog. Click the **Add** button next to the parameter field to browse or search for the hierarchy to search below.
- **Search Fields:** This parameter allows adding multiple search fields that are displayed as facets within the Faceted Node Picker dialog. Of the available options, Name Search Field and Hierarchy Search Field components can be added only once and Attribute Search Field can be added multiple times. The order that the search fields are listed determines the order that they will be displayed within the Faceted Node Picker dialog. All of the search fields require additional configuration. Clicking the Add button next to the Search Fields parameter displays an Add Component dialog with the option to add one of the following search fields: Attribute Search Field, Hierarchy Search Field, and Name Search Field. Steps for configuring each search field can be found in their respective sections below:
  - Configuring Name Search Field
  - Configuring Hierarchy Search Field
  - Configuring Attribute Search Field

  Because Name Search Field and Hierarchy Search Field components can be only be configured once for each Faceted Node Picker dialog, if a second search field of the same type is added, an Error dialog will be displayed as shown below, and the Search Fields parameter will remain unchanged.



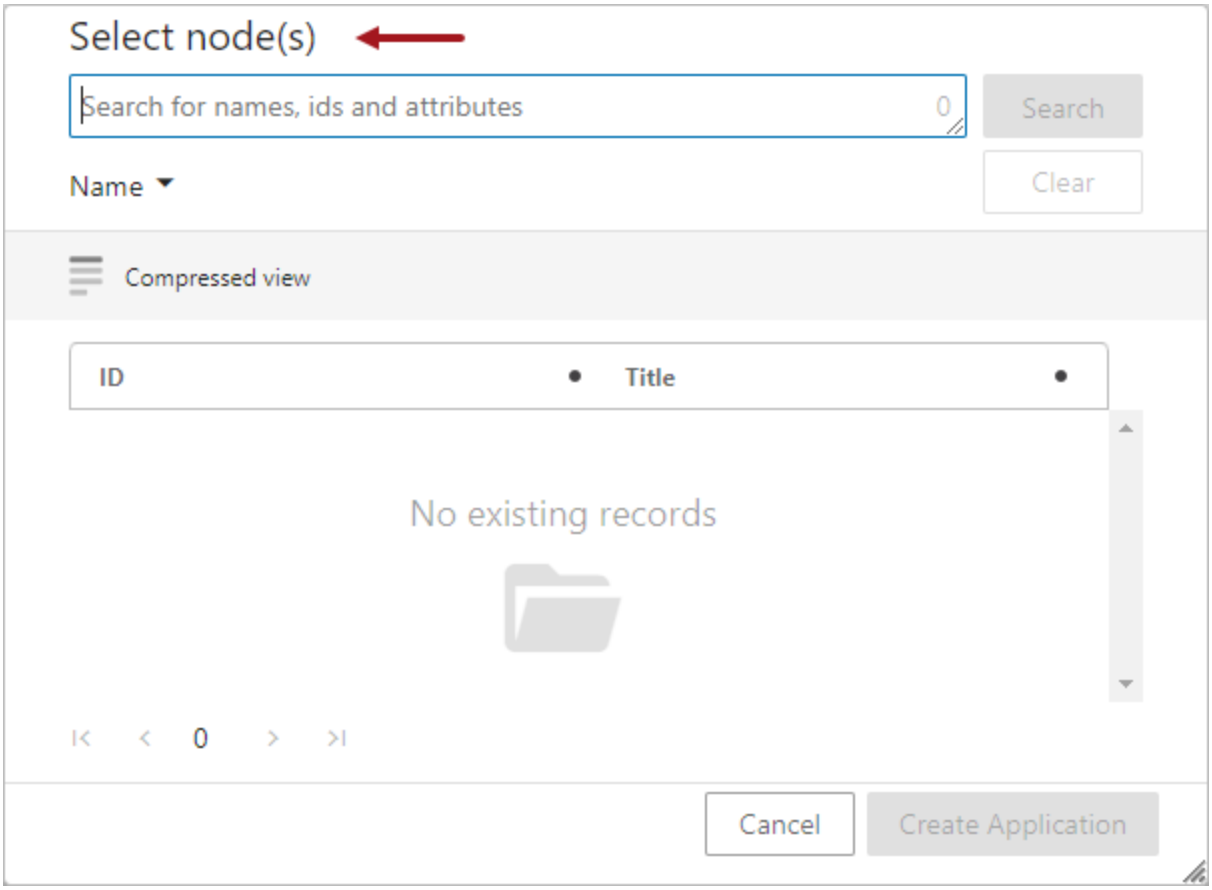

- **Search Placeholder:** This parameter defines the text to be displayed within the free text search field when no character is typed in the free text search field. To make changes to the default text in the free text search field, add the custom text to the Search Placeholder field. If nothing is added, the Search Placeholder default will display 'Search for names, ids, and attributes.'

  This is how the default free text search field will display at the top of the Faceted Node Picker dialog:




- **Title:** To make changes to the default title in the Faceted Node Picker dialog that appears when the user clicks on the Create Application Action button, add the custom text to the Title field. If nothing is added, the Title default will display as 'Select node(s).'

  This is how the Title displays at the top of the Faceted Node Picker dialog:




- **Node List:** The Faceted Node Picker Node Picker Dialog Properties is pre-configured with a Node List in Multi Edit Display Mode. Select the 'Node List' option in the dropdown and click on the 'go to component' to add more table header components in the Node List component. For more information, refer to the **Node List Component** topic within the **Web User Interfaces** section of **STEP Online Help**.

### Improving Part Type Accuracy

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 targets (assembly / vehicle or part) are displayed more accurately.

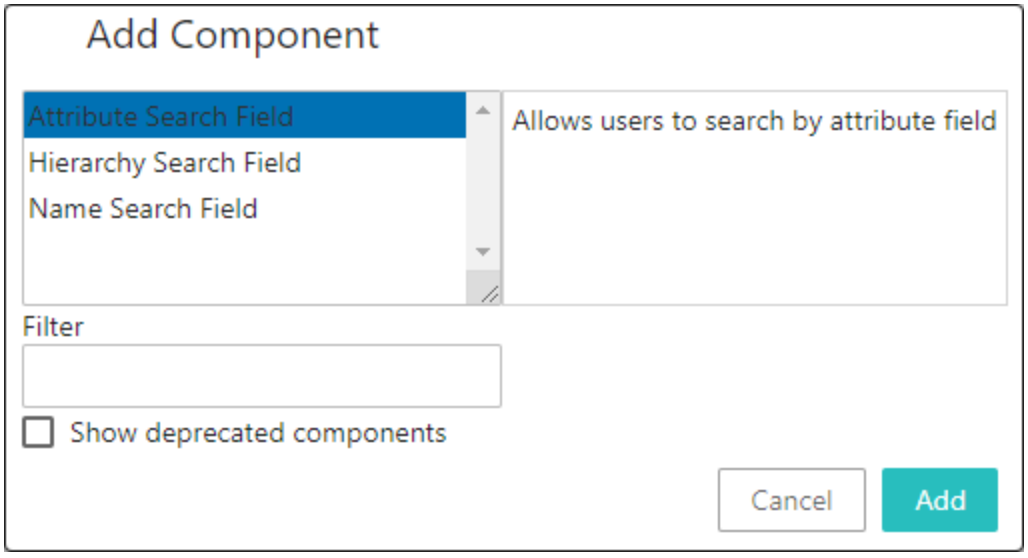
For example, when searching for a 2013 Audi A3 vehicle for a selected spark plug, the results table would list both Gas and Diesel engines, even though Diesel engines do not use spark plugs, and are invalid in the search results. However, when the validation path functionality is implemented in conjunction with the 'Missing Application Conditions' attribute and the 'Check path for missing application' business condition, the search results will display more accurately because the target search 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, refer to the **Automotive Validation Path Functionality** and the **Business Condition: Check Path for Missing Application** topics within this guide.

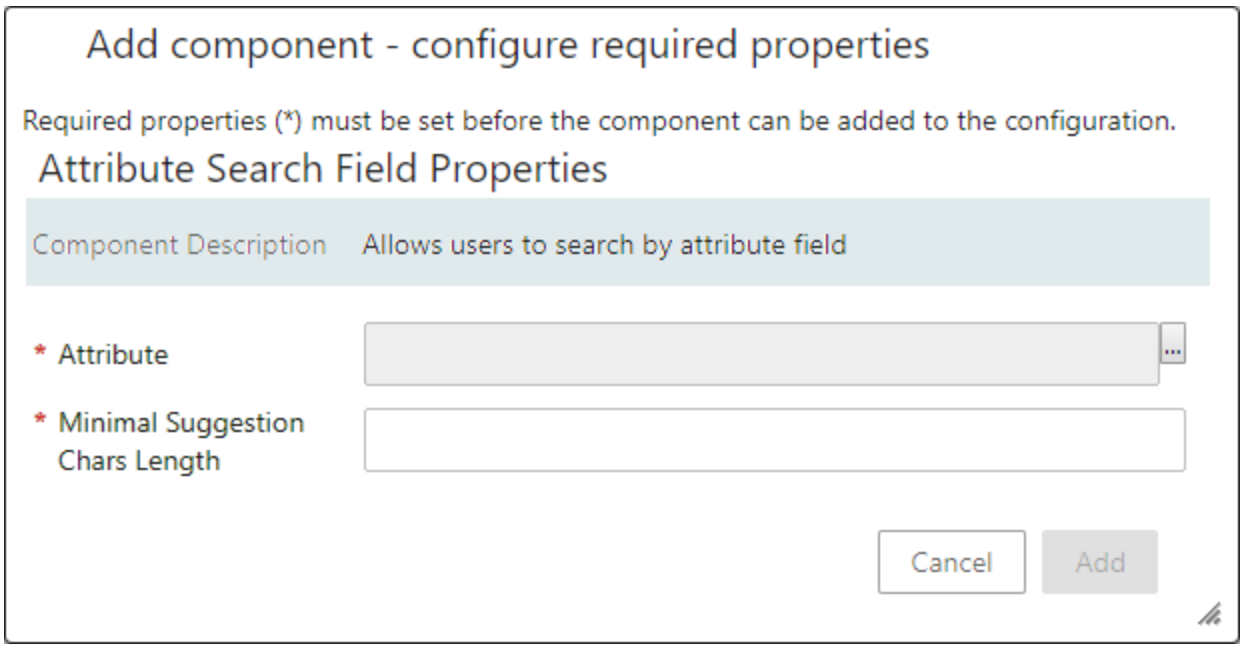
### Configuring the Attribute Search Field

The Attribute Search Field will search for attribute values that is valid for the vehicle / assembly or part for each standard dependent on which object type has been selected. For example, in the AutoCare standard, if a PIES Item object (Part) is selected, then the Attribute Search Field will by default search only on the attribute(s) that are valid for the Base Vehicle objects (Vehicle) to display (e.g. VCdb Year). Or in the TecDoc standard, if a Vehicle Type (PC) object (Vehicle) is selected, then the Attribute Search Field will by default search only on the attribute(s) that are valid for the Supplier Article objects (Part) to display (e.g. Country Inclusion, Housing Type, etc.). If a root node has been configured in the Root Nodes parameter, then the Attribute Search Field will only search below the configured root node(s). To add the Attribute Search Field component within the Search Fields parameter:

1. From the Faceted Node Picker Dialog Properties dialog, click the **Add** button next to the Search Fields parameter field, and the Add Component dialog will display.




2. Click **Attribute Search Field** from the list of components.
3. Click the **Add** button, and the Attribute Search Field Properties will display, as shown below.

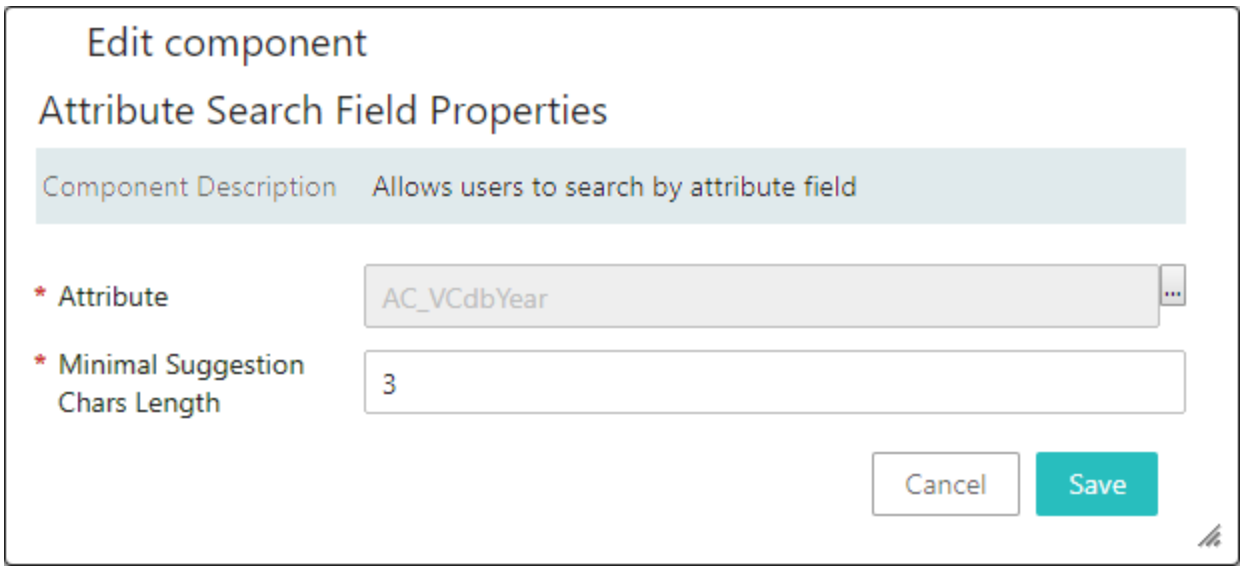



**Attributes:** This required field allows users to add an attribute criterion to their search. All attribute validation base types (e.g., Text, Number, List Of Values) are allowed. Click the ellipsis button (...) to find and select the required attribute. The selected attribute must be valid for the object type that is intended for search, and the attribute value must be populated on that object in order for it to display correctly. The Name of the attribute will be displayed in the Faceted Node Picker dialog.

**Minimal Suggestion Length:** This required field determines the minimum number of characters that needs to be typed in the typeahead field so that the associated facet starts suggesting the values in the dropdown matching with the typed characters.

4. Once the required parameters are populated, click the **Add** button to return to the Faceted Node Picker Dialog Properties and **Save** the configuration. Optionally add another search field.

   Below is an example of a configured Attribute Search Field component:



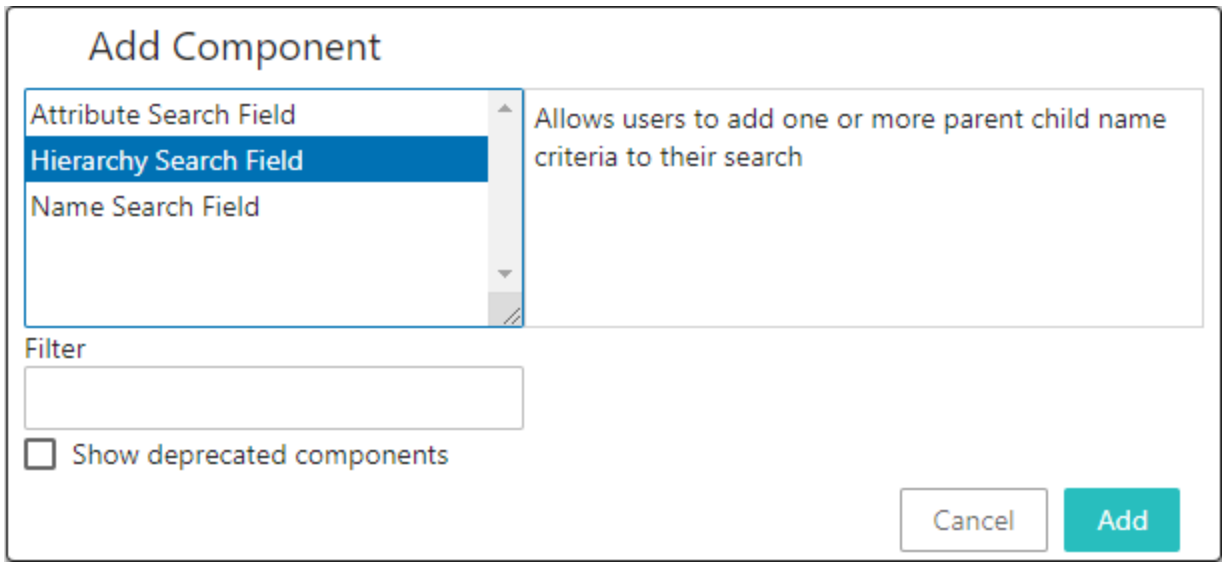

### Configuring the Hierarchy Search Field

The Hierarchy Search Field will search for parent objects of the base vehicle or part in order to narrow down the search for each standard dependent on which object type has been selected. For example, in the AutoCare standard, if an application record has to be created for the selected PIES Item object (Part), then the Hierarchy Search Field can be configured to search for the Make/Model to display all of the Base Vehicle objects (Vehicle) for the selected Make/Model. Or the Make/Model can be selected to further narrow down the list of Years that are available for that Make/Model.

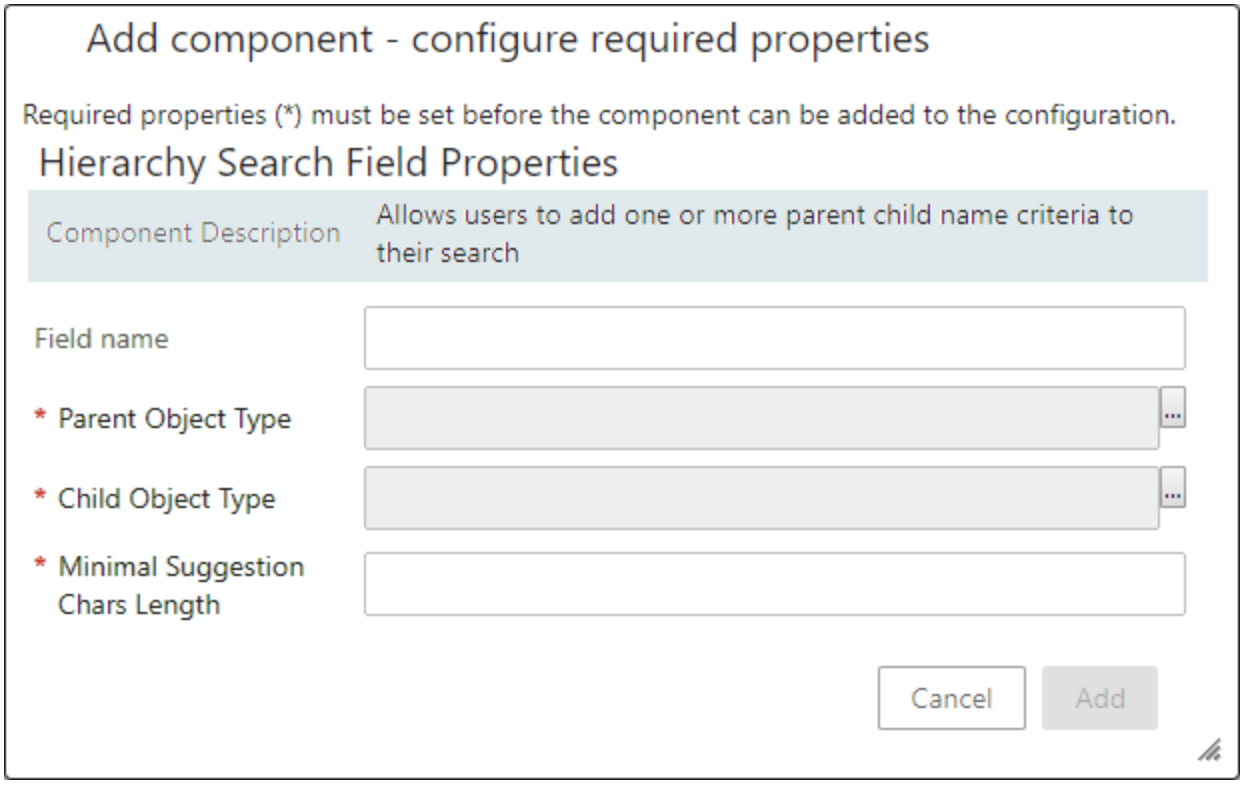
If a root node has been configured in the Root Nodes parameter, then the Hierarchy Search Field will only search below the configured root node(s). For example, the Light Duty Vehicle root node can be added so that the search will only be performed to include Light Duty vehicles but excluded all other vehicle types (Medium/Heavy Duty, Powersports, etc.).

To add and configure the Hierarchy Search Field component within the Search Fields parameter:

1. From the Faceted Node Picker Dialog Properties dialog, click the **Add** button next to the Search Fields parameter field and the Add Component dialog will display.




2. Click **Hierarchy Search Field** from the list of components.
3. Click the **Add** button, and the Hierarchy Search Field Properties will display three required fields and one optional field, as shown below.




**Field Name:** This optional field lets the user provide a label for the facet associated with the Hierarchy Search Field that best communicates to the user what criteria are available for selection within the facet. By default, the label is set to 'Hierarchy.'

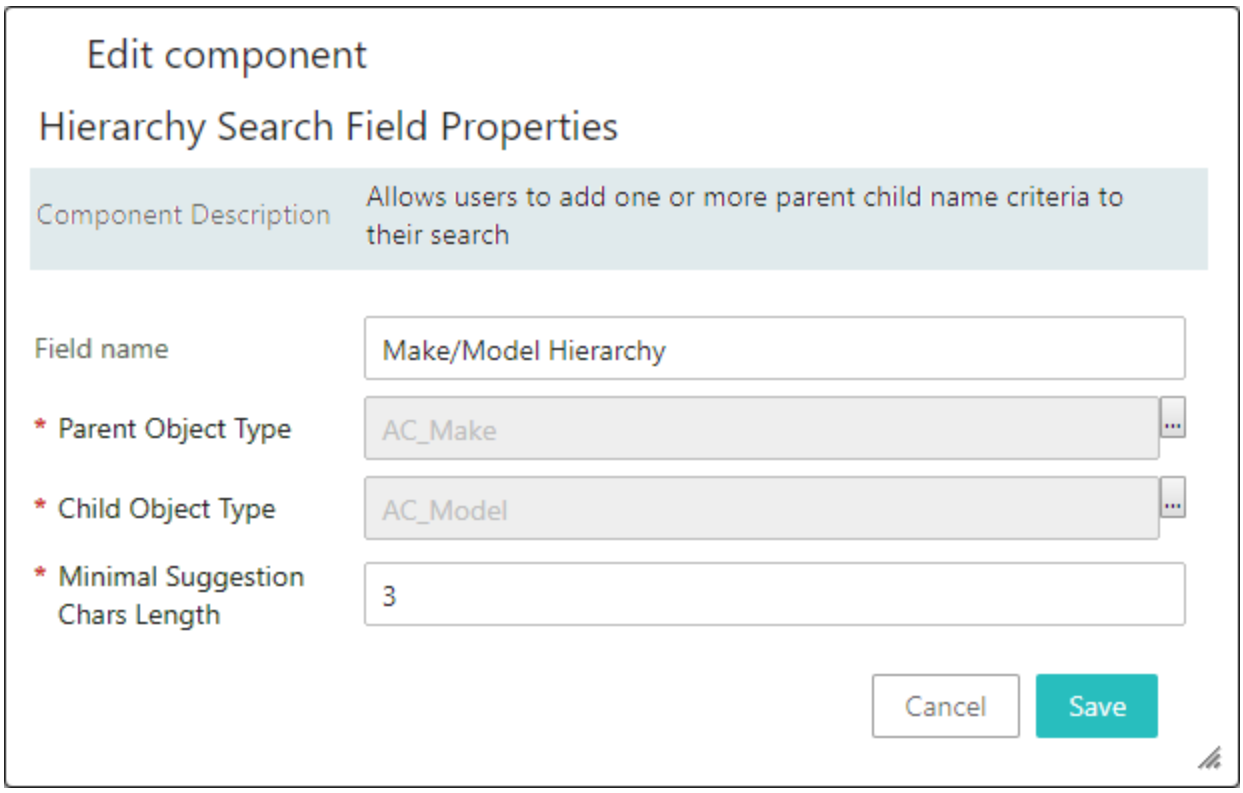
**Parent Object Type:** Determines the parent object type that a user can select within the Hierarchy facet. Click the ellipsis button (...) to find and select the object type for the objects that exist in the vehicle hierarchy or part hierarchy, and should be displayed for a user as a parent object to select within the facet dropdown. One such example of the parent object is the Make object of the vehicle.

**Child Object Type:** Determines the child object type that a user can select within the Hierarchy facet. Click the ellipsis button (...) to find and select the object type that is the child of the object type that has been defined in the Parent Object Type parameter, and should be displayed for a user as a child object to select within the facet dropdown. One such example of the child object is the Model object of the vehicle.

**Minimal Suggestion Length:** This required field determines the minimum number of characters that needs to be typed in the typeahead field so that the associated facet starts suggesting the object name in the dropdown matching with the typed characters.

4. Once the required parameters are populated, click the **Add** button to return to the Faceted Node Picker Dialog Properties and **Save** the configuration. Optionally add another search field.

   Below is an example of a configured Hierarchy Search Field component:

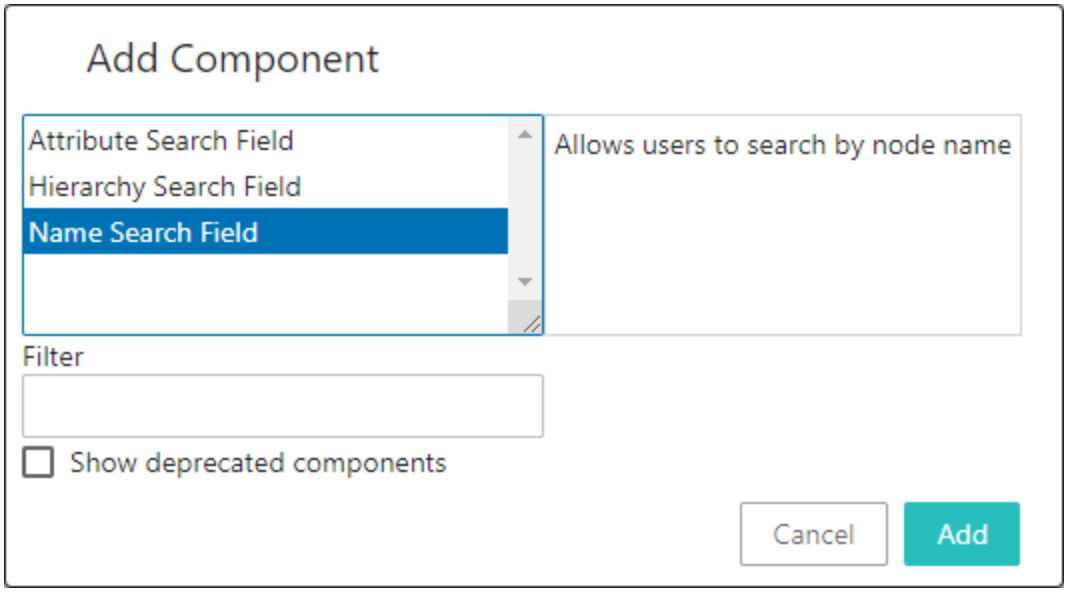



### Configuring the Name Search Field

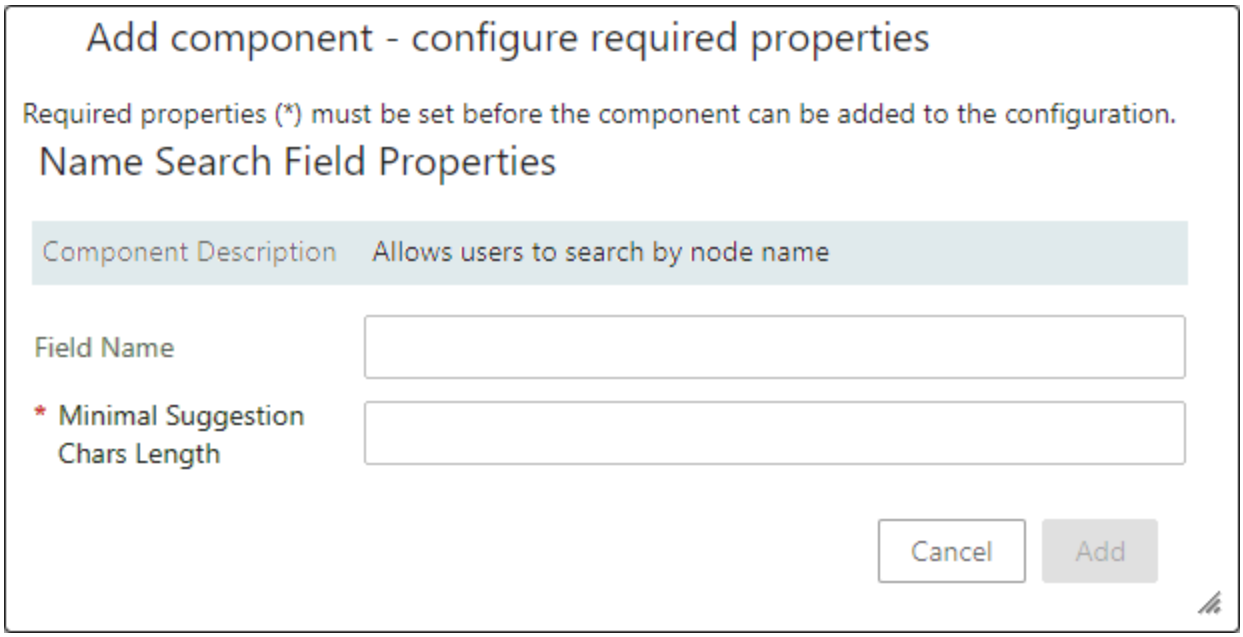
The Name Search Field will search for the vehicle / assembly or part to create an application for each standard dependent on which object type has been selected. For example, in the AutoCare standard, if a PIES Item object (Part) is selected, then the Name Search Field will by default search on the Name of the Base Vehicle objects (Vehicle) to display. Or in the TecDoc standard, if a Vehicle Type (PC) object (Vehicle) is selected, then the

Name Search Field will by default search on the Name of the Supplier Article objects (Part) to display. If a root node has been configured in the Root Nodes parameter, then the Name Search Field will only search below the configured root node(s). To add the Name Search Field component within the Search Fields parameter:

1. From the Faceted Node Picker Dialog Properties dialog, click the **Add** button next to the Search Fields parameter field, and the Add Component dialog will display.




2. Click **Name Search Field** from the list of components.
3. Click the **Add** button, and the Name Search Field Properties will display, as shown below.

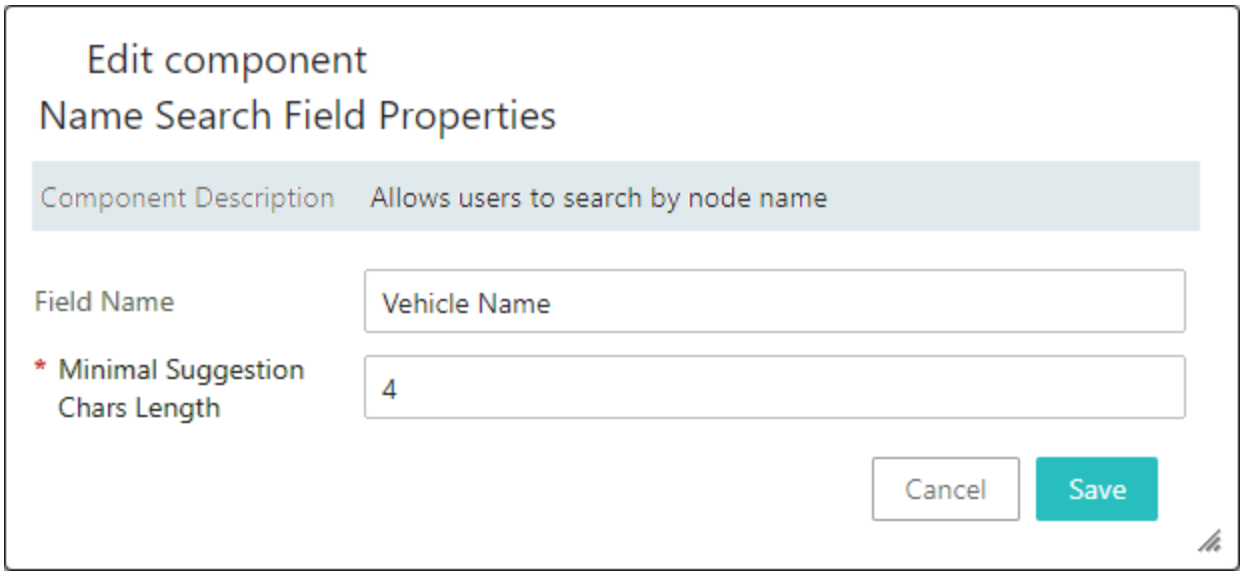



**Field Name:** This optional field allows the user to provide a label for the facet associated with the Name Search Field that best communicates to the user what criteria are available for selection within the facet. By default, the label is set to 'Name.'

**Minimal Suggestion Length:** This required field determines the minimum number of characters that needs to be typed in the typeahead field so that the associated facet starts suggesting the names in the dropdown matching with the typed characters.

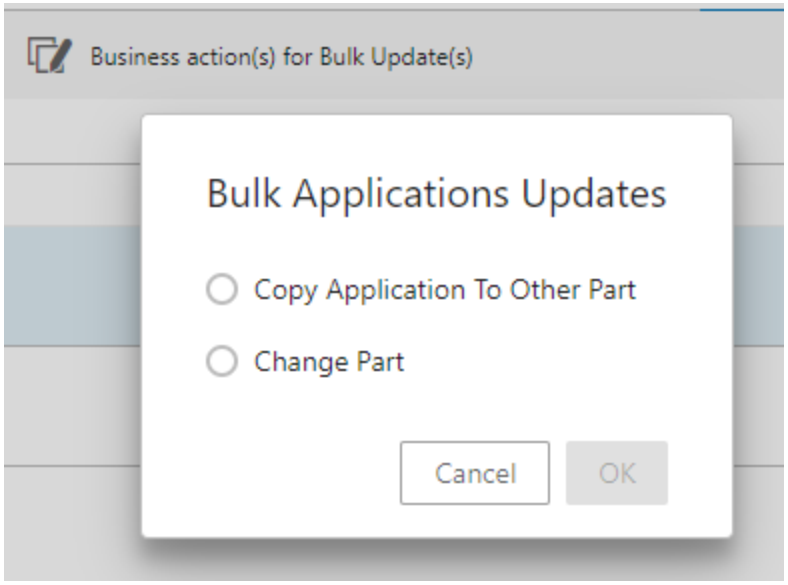
4. Once the component is populated, click the **Add** button to return to the Faceted Node Picker Dialog Properties and Save the configuration. Optionally add another search field.

   Below is an example of a configured Name Search Field component:




## Bulk Applications Updates Component

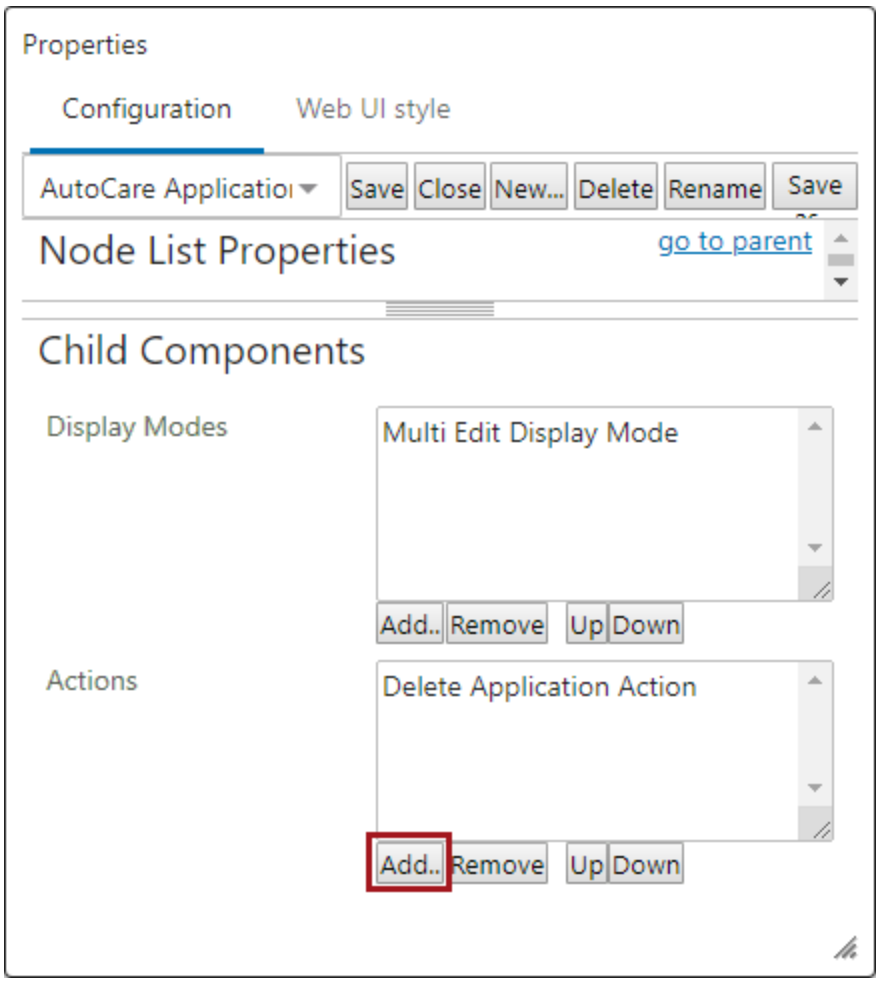
The Bulk Applications Updates component allows users to execute configured Business Action(s) on selected application records. This component is available as a toolbar action for Node Lists, and is commonly used within the Application Editor Screen and Application Manager Screen. One of the notable features of this component is that it allows users to add any number of Business Actions. If more than one Business Action is configured, then the Bulk Applications Updates dialog will display the options for the user to select the desired Business Action to be executed.



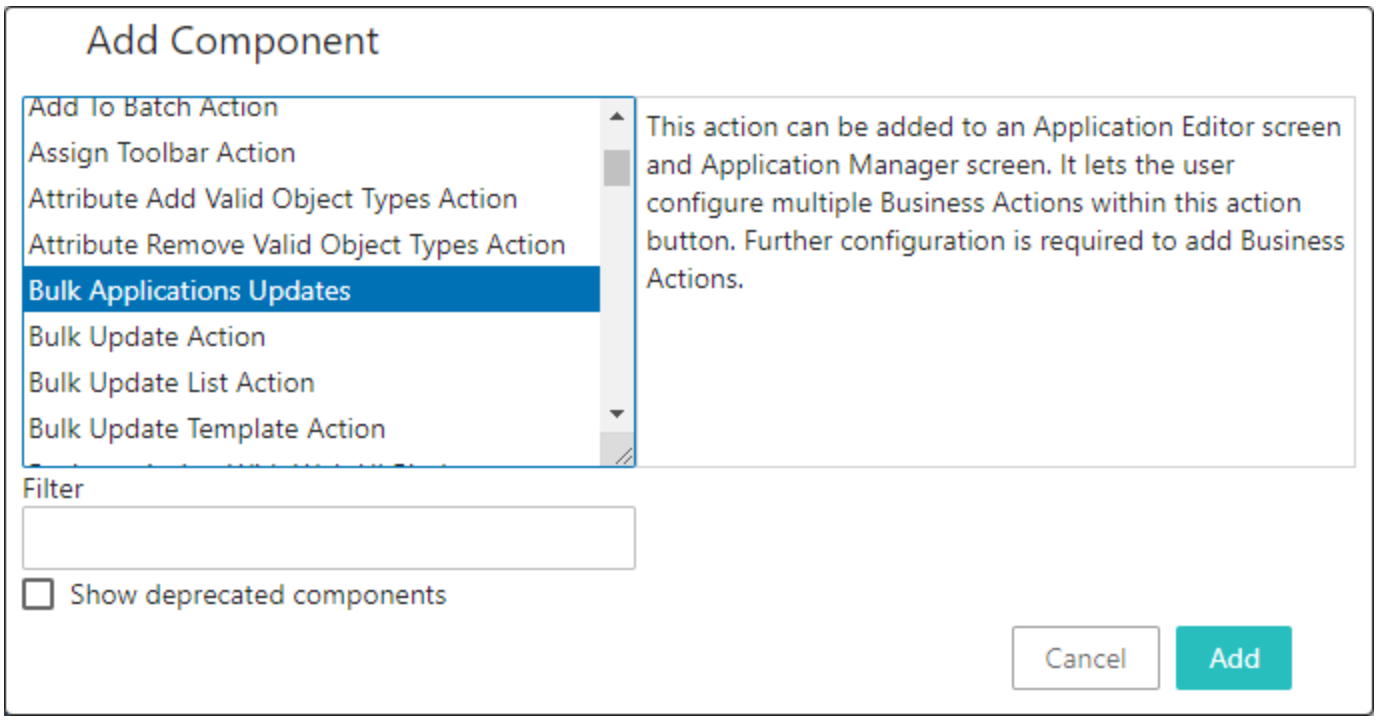

## Configuring the Bulk Applications Updates Component

In the Application Editor Screen and Application Manager Screen, users need to add and configure the Bulk Applications Updates component within the Actions field of the Node List. The following process details how to add and configure the Bulk Applications Updates component as a toolbar action in a Node List.

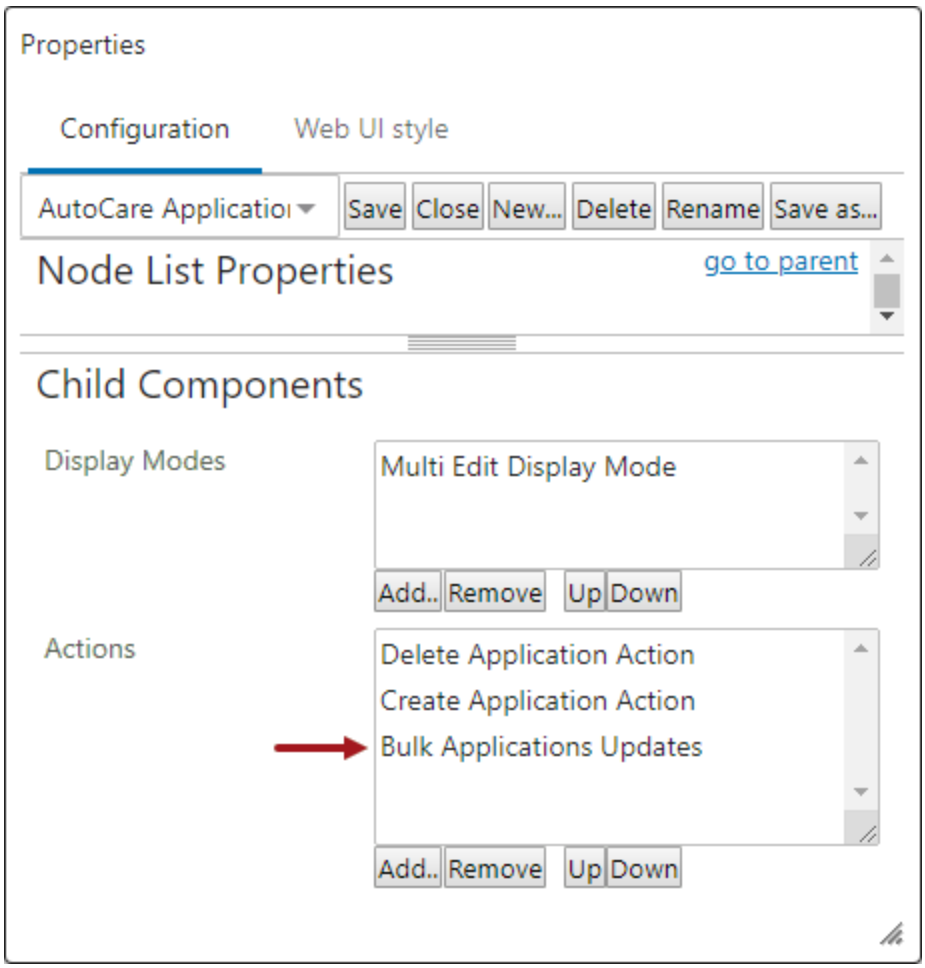
1. First, add the component. Under Child Components on the Node List Properties screen in the designer, click the **Add** button under the field for the Actions parameter.



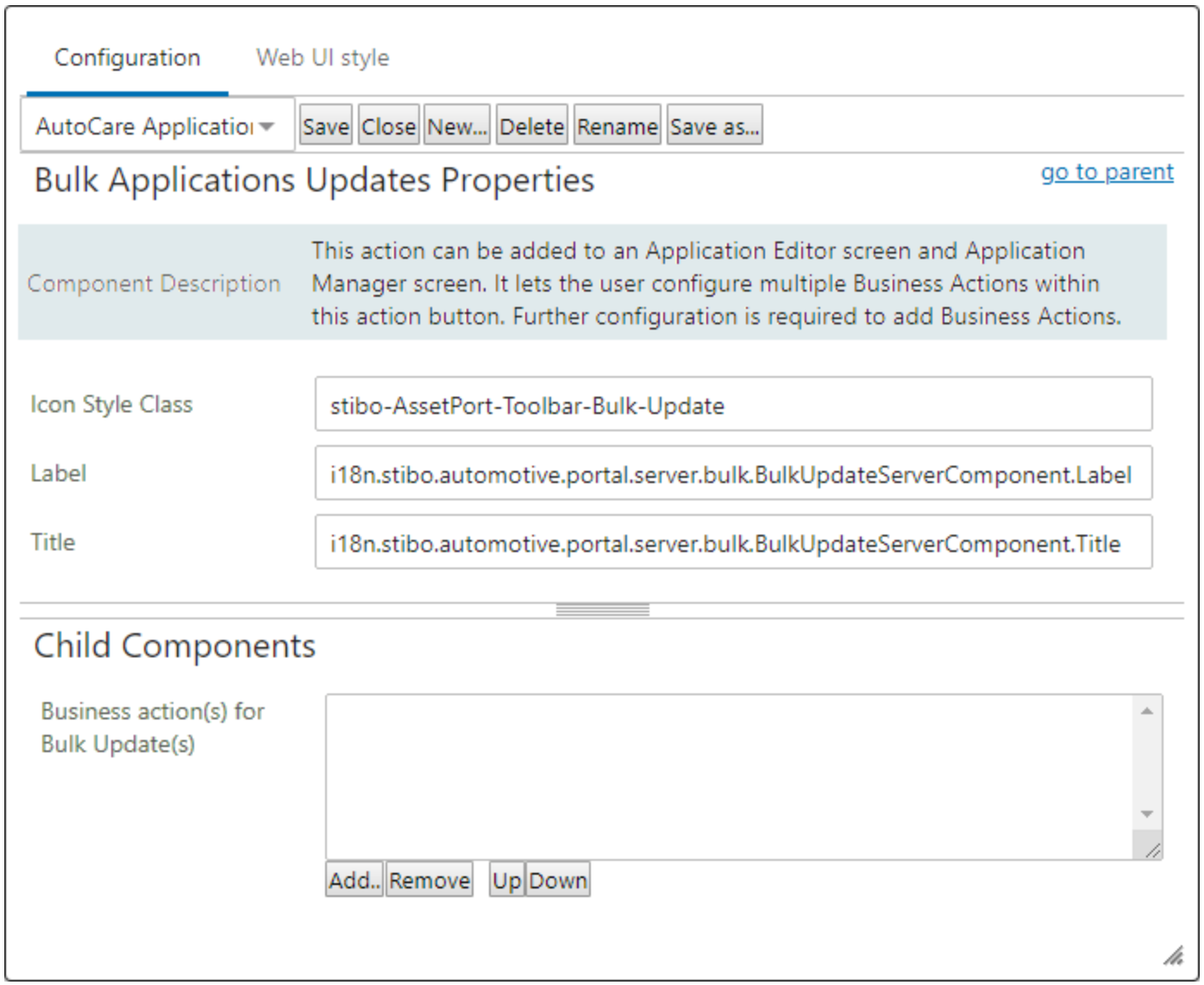

2. Then select **Bulk Applications Updates** from the component list. Click **Add**.




3. Once added to the list of actions for the Node List, double-click the **Bulk Applications Updates** component to configure.




4. In the Bulk Applications Updates Properties dialog, define the following parameters:



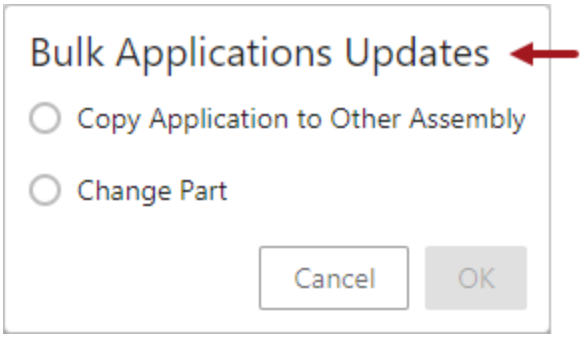

- **Icon Style Class**: This parameter allows a custom icon to be displayed on the action button.
- **Label**: To make changes to the default button label, add the custom text to the Label field. If nothing is added, the Label default will display as 'Business action(s) for Bulk Update(s).'

  This is how the button will display at the top of the Application Editor Screen:

- **Title**: To make changes to the default title of the dialog that appears when the user clicks on the Bulk Applications Updates button, add the custom text to the Title field. If nothing is added, the Title default will display as 'Bulk Applications Updates.'

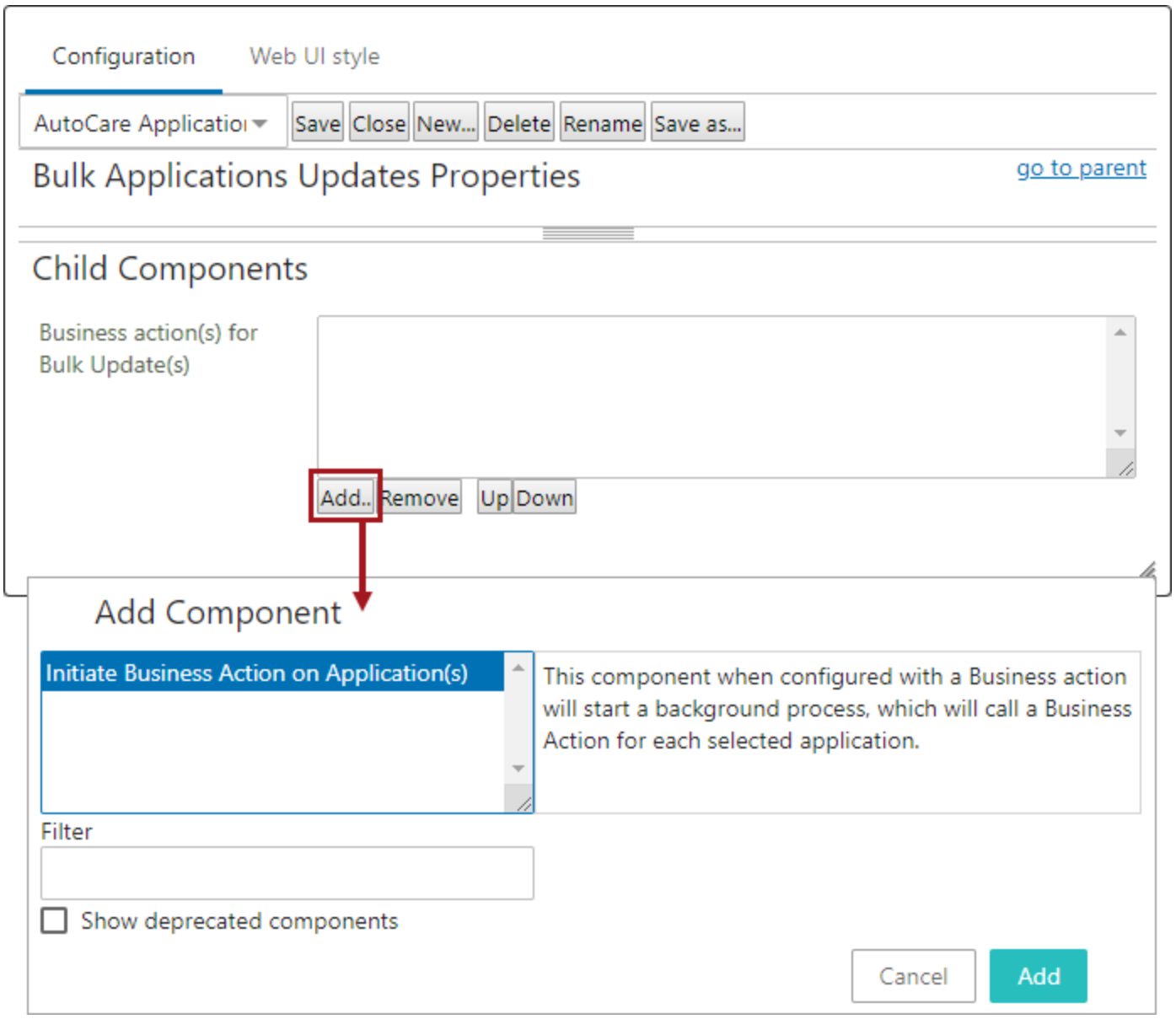
This is how the title will display at the top of the dialog when the user clicks on the Business action(s) for Bulk Update(s) button:



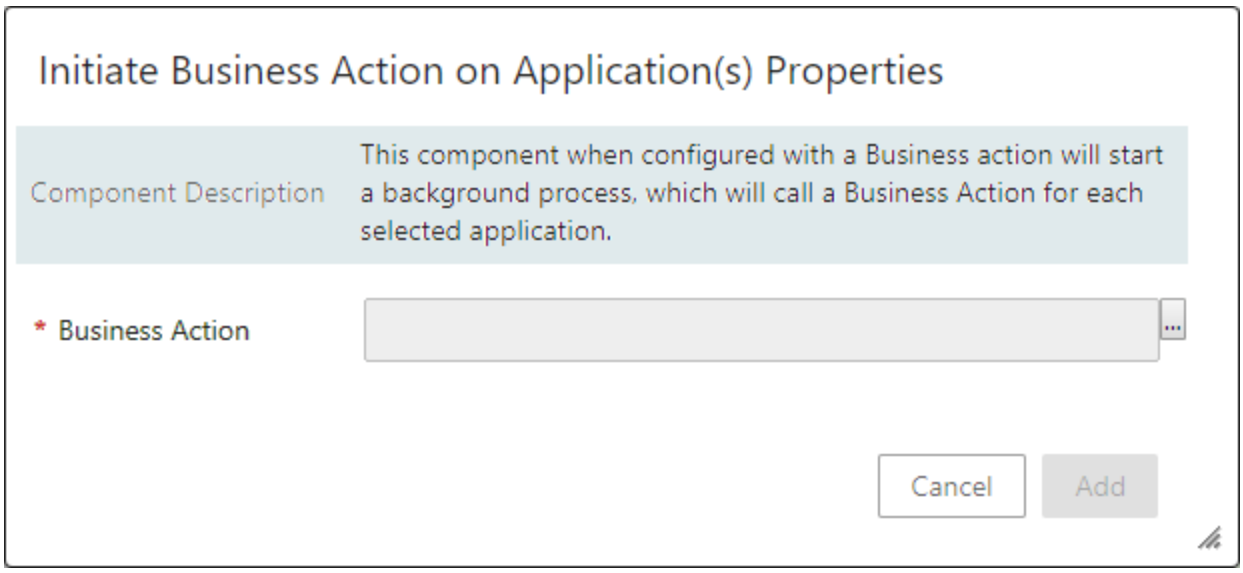

**Note:** The Bulk Applications Updates dialog will be displayed only when there is more than one Business Action configured within the component. This dialog prompts users to select the desired Business Action to be executed. **The presence of only one bulk update will not display this dialog and the configured sole Business Action gets executed (without any confirmation dialog) when the user clicks on the Business action(s) for Bulk Update(s) button.**

- **Business action(s) for Bulk Update(s)**: This parameter lets users add one or multiple Business Actions that gets displayed as radio buttons to select when the user clicks the action button in the screen.

5. Click the **Add** button under the field for **Business action(s) for Bulk Update(s)** parameter to add Business Action(s) that needs to be executed. Then the **Add Component** dialog will display.



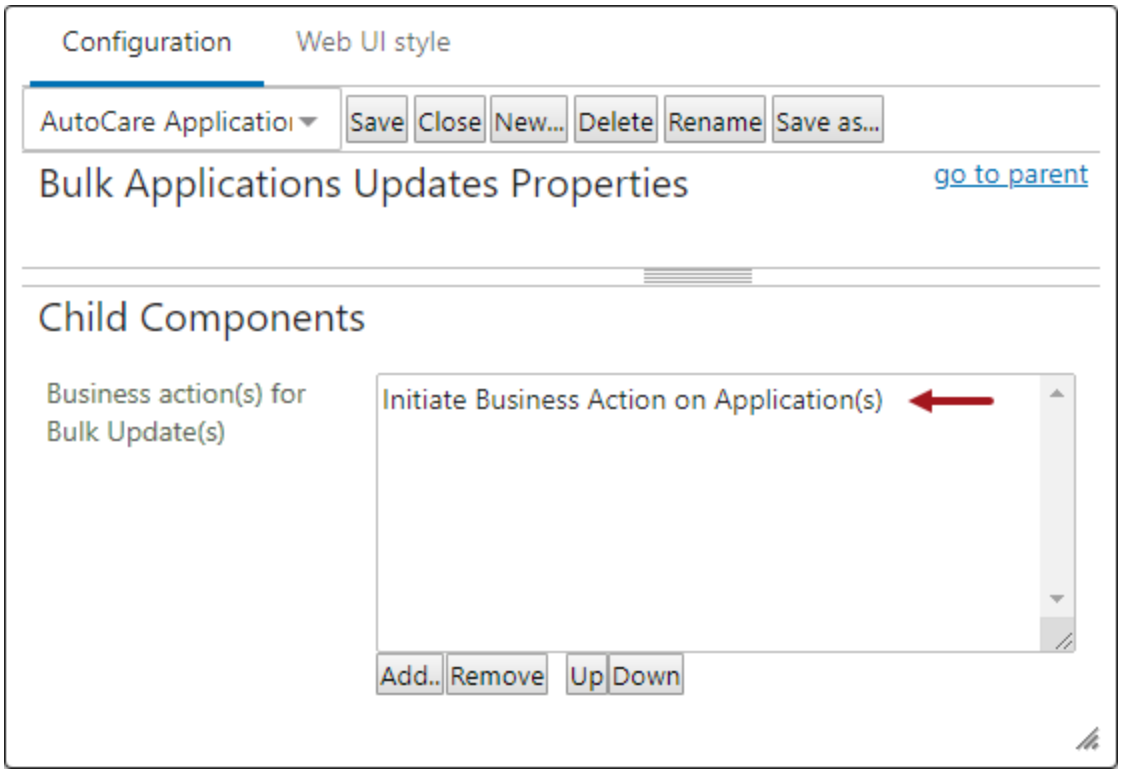

6. Select the **Initiate Business Action on Application(s)** component from the component list and then click **Add**, and the Initiate Business Action on Application(s) Properties will display as shown below.




7. In the Initiate Business Action on Application(s) Properties dialog, define the following parameter:

   **Business Action**: This parameter allows the user to configure a Business Action that is intended to be executed on the selected application records in the Application Editor Screen or Application Manager Screen. Click the ellipsis button (...) to find and select the required Business Action. Click the **OK** button once the Business Action has been selected.

8. Click the **Add** button and the Initiate Business Action on Application(s) component will be listed within the Business action(s) for Bulk Update(s) parameter as shown below.

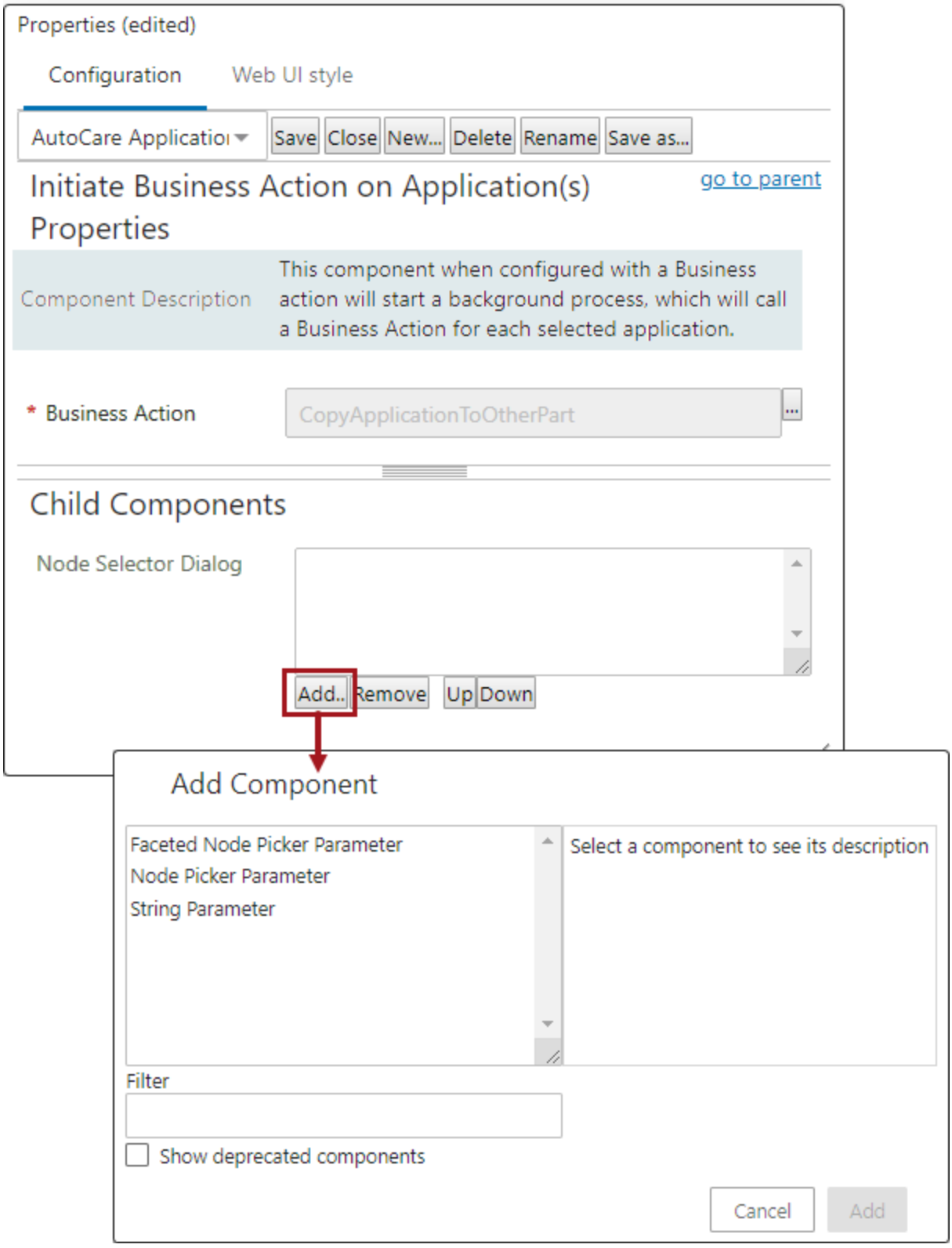


9. Double-click on the Initiate Business Action on Application(s) component listed within the Business action(s) for Bulk Update(s) parameter to define the type of node selector dialog that needs to be displayed if the business action configured within the Business Action parameter needs an object selection when executed.

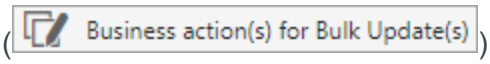
   If the configured business action requires an object selection, click the **Add** button available under the field for **Node Selector Dialog** parameter. Node Selector Dialog allows the user to determine how the nodes are selected.

   Configure the required Node Selector Dialog as detailed in the following topics:

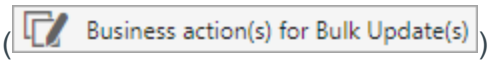
   - **Faceted Node Picker Parameter Dialog**
   - **Node Picker Parameter Dialog**
   - **String Parameter Dialog**


Properties (edited)
Configuration
Web UI style
AutoCare Applicatio
Save
Close
New...
Delete
Rename
Save as...
Initiate Business Action on Application(s) Properties
go to parent
Component Description
This component when configured with a Business action will start a background process, which will call a Business Action for each selected application.
* Business Action
CopyApplicationToOtherPart
Child Components
Node Selector Dialog
Add..
Remove
Up
Down
Add Component
Faceted Node Picker Parameter
Node Picker Parameter
String Parameter
Select a component to see its description
Filter
Show deprecated components
Cancel
Add

## Using Business Action(s) for Bulk Update(s) Action

The **Business action(s) for Bulk Update(s)** button ( 

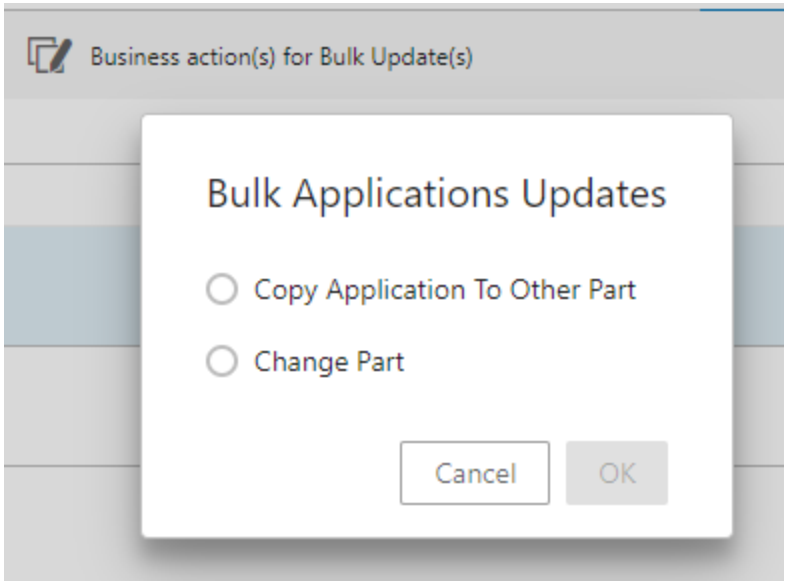
) is enabled when one or more application records are selected in the search results table. If more than one Business Actions are configured then the Bulk Applications Updates dialog will display, prompting users to select the desired Business Action to be executed.

1. Perform a search for applications.
2. Select the application records that are to be updated.
3. Click the **Business action(s) for Bulk Update(s)** button ( 

) that is available in the toolbar of the screen.

   If only one Business Action has been configured, then the bulk update will be done instantly without a dialog window being displayed. In this case, it is suggested that the Label of the button be changed to something that is more meaningful for the user to relay what the bulk update will do (for example: 'Change Part').

   If more than one Business Actions are configured, then a dialog window will be displayed with the available Business Actions for the user to select. Below is an example of two Business Actions that have been configured.




4. Select the required Business Action that is to be executed by clicking the radio button, and then click **OK**.

   If the Business Action requires an object selection, then the Node Picker Dialog will display. Select the required object and click **OK** to close the dialog.

   The 'Started background process' dialog (shown below) will be displayed at the top of the screen. Optionally, click the Background Process Link (BGP_109137) to view the detailed execution of the background process within the Background Process Details screen.

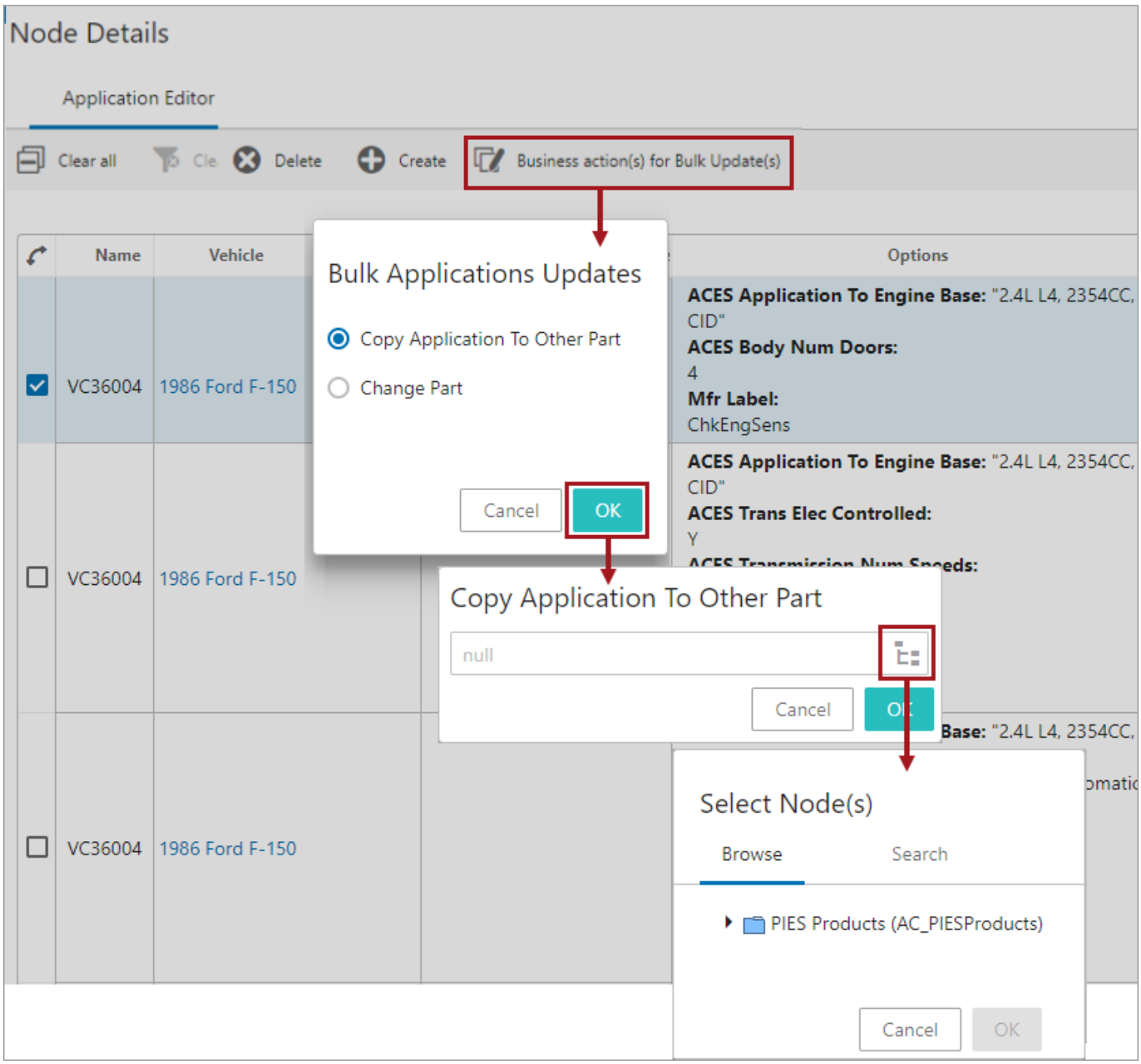
This action button can be configured to execute different Business Actions. The following use cases are contained within the **Automotive Business Rule Plugins** section

- **Business Action: Change Assembly**
- **Business Action: Change Part Type**
- **Business Action: Change Part**
- **Business Action: Copy Application to Other Assembly**
- **Business Action: Copy Application to Other Part**
- **Business Action: Copy Applications to Related Parts**

## Node Picker Parameter Component

The **Node Picker Parameter** component is configured within the Initiate Business Action on Application(s) Properties dialog. This component assists in executing the defined Business Action that requires an object selection. Node Picker Parameter component determines which objects are selectable through the Node Picker dialog.

Below is a screenshot representing the Node Picker Parameter component that gets displayed when the Business Action that needs an object selection is initiated for execution.



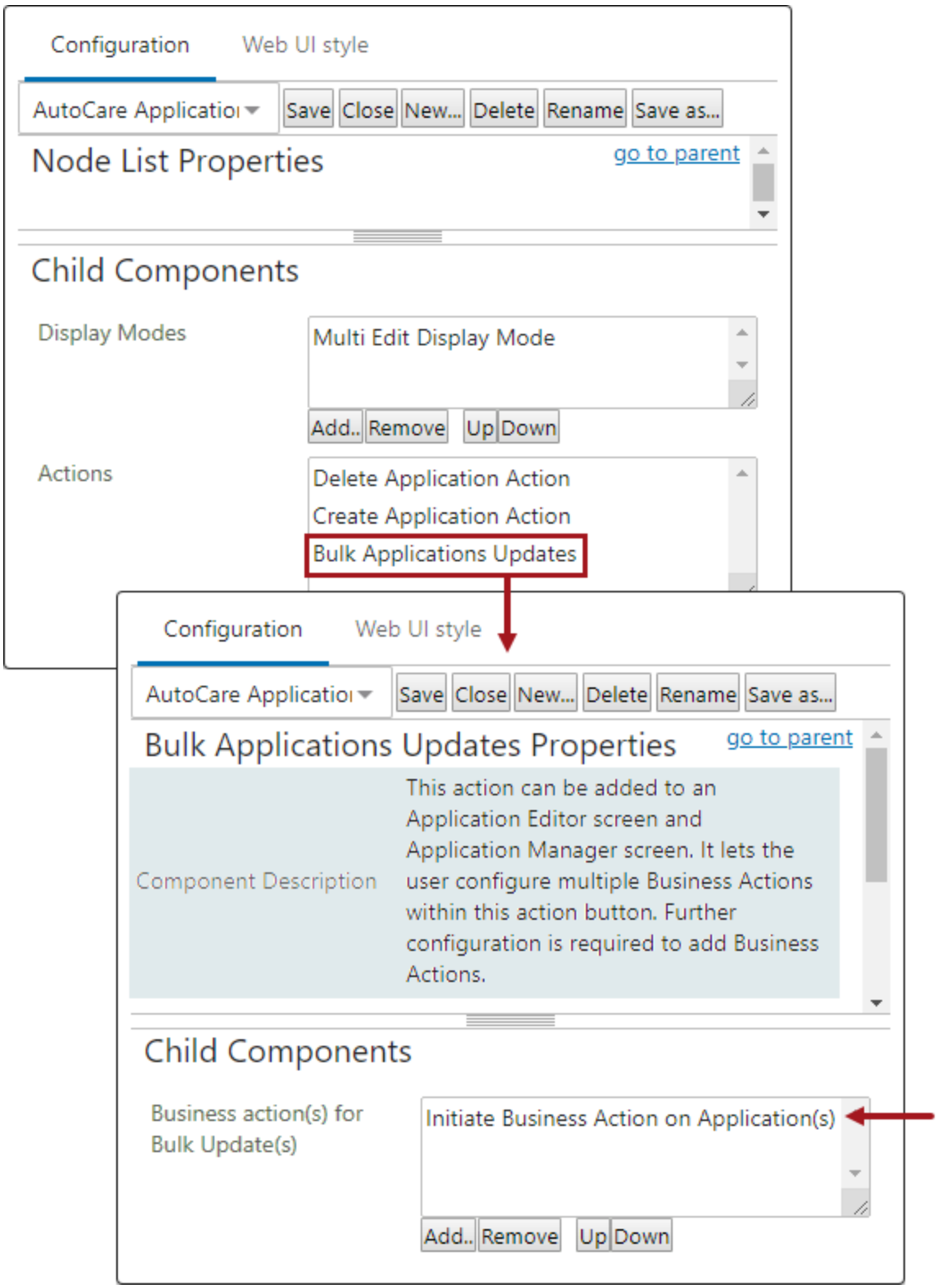

### Configuring the Node Picker Parameter Component

Follow the steps described below to add and configure the Node Picker Parameter component that gets displayed when the user executes the Business Action that needs an object selection via **Business action(s) for Bulk Update(s)** action button

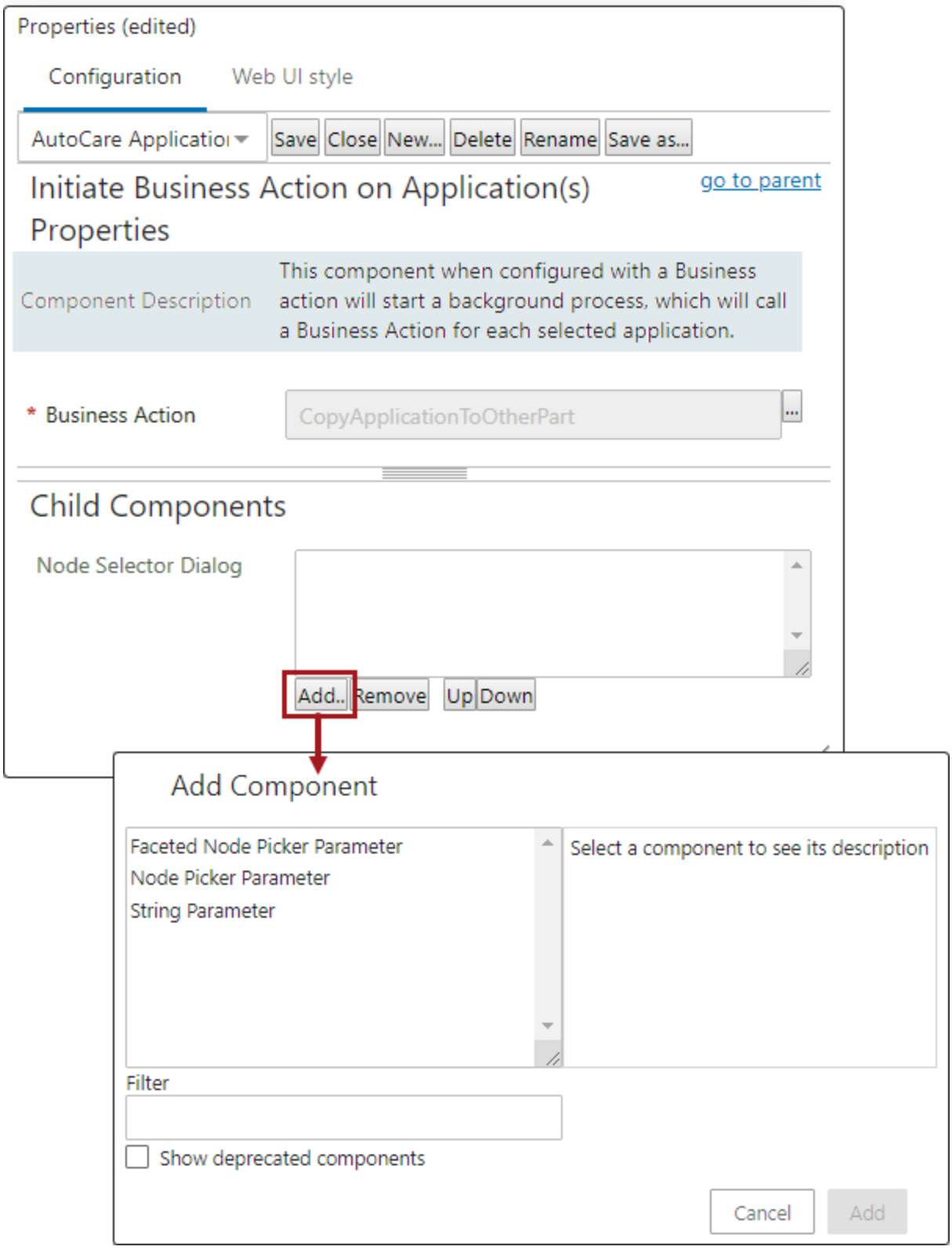
.

The example below shows a configuration of Node Picker Parameter component in an Application Editor Screen:

1. In the Web UI design mode, select the Bulk Applications Updates action button configured within the Node List component of the required screen (Application Editor Screen / Application Manager Screen).
2. Double click the **Initiate Business Action on Application(s)** component if it already exists, if not create a new one by clicking the **Add** button within the Business action(s) for Bulk Update(s) parameter. For more information, refer to **Bulk Applications Updates Component** topic.

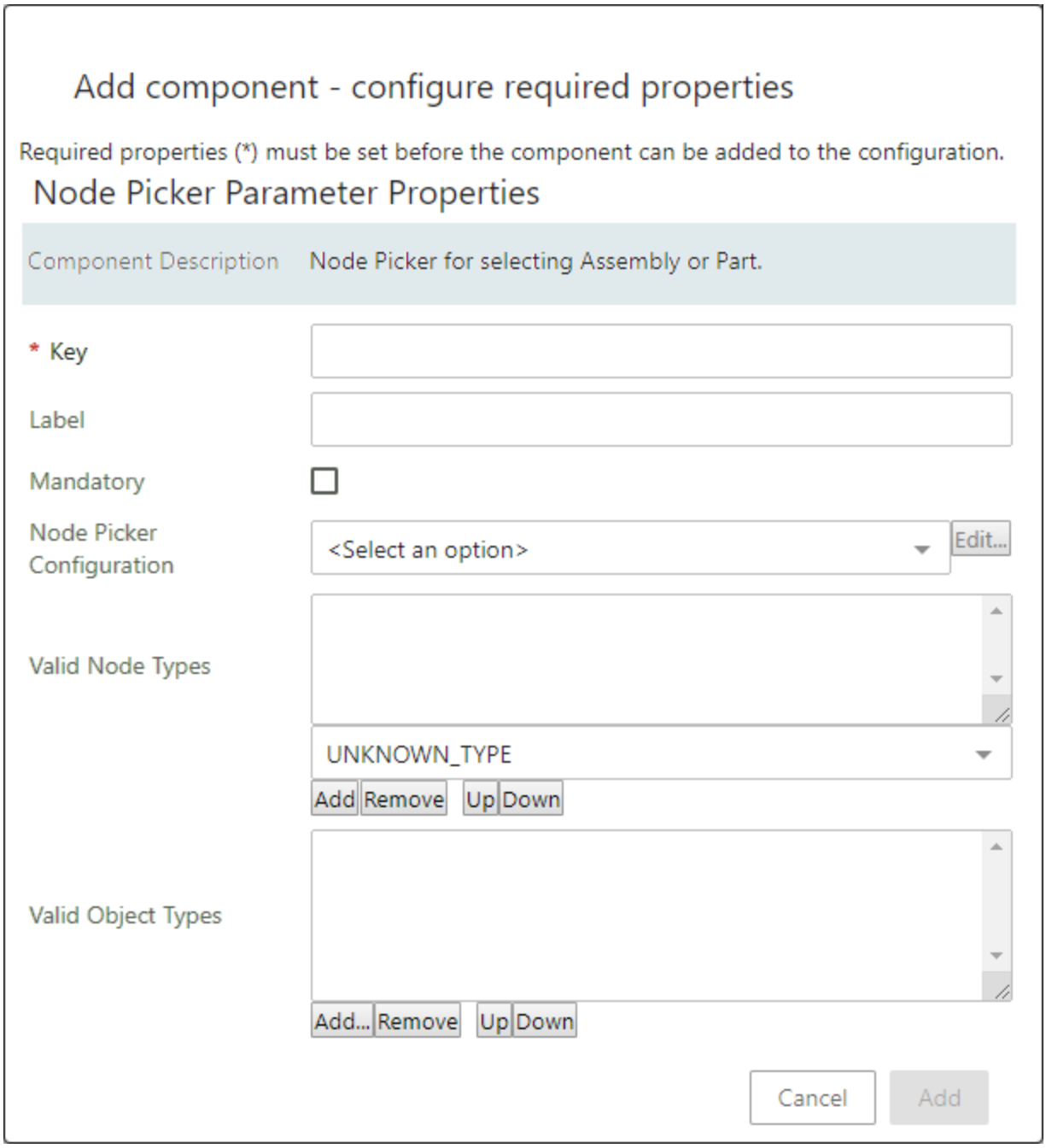



3. Under the Child Components section on the Initiate Business Action on Application(s) Properties, click the **Add** button under the field for Node Selector Dialog parameter. The Add Component dialog will display.




4. Select the Node Picker Parameter component and click **Add**, and the 'Add component - configure required properties' for the 'Node Picker Parameter Properties' dialog will display (as shown below). Define the

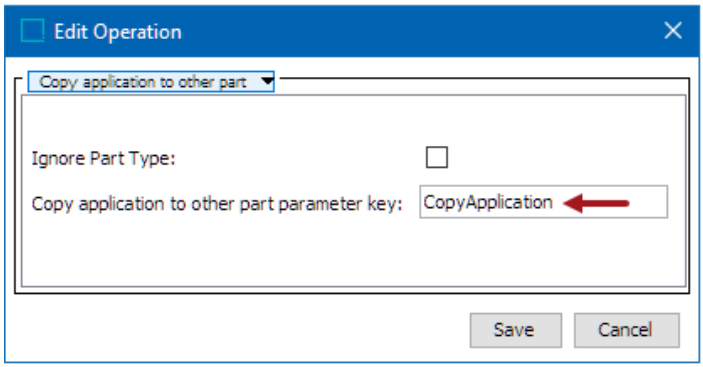
following parameters:

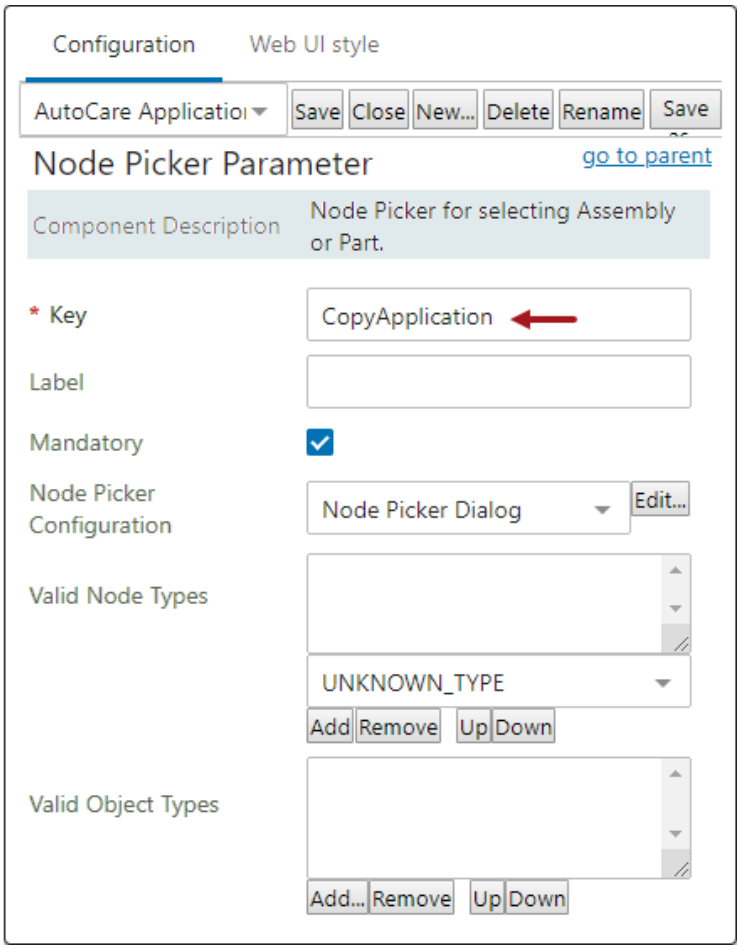



- **Key:** By default, this required parameter is blank. The value entered for this field is used to connect a business rule in the system to this Web UI component. Type in the exact characters (key) that is entered within the business rule configured within the Business Action parameter of Initiate Business Action on Application(s) Properties. Uniqueness is the only restriction for this key and it is case sensitive. For example, the Business Action 'Copy application to other part' has the value 'CopyApplication' populated within the 'Copy application to other part parameter key' parameter as shown in the image on the left. In

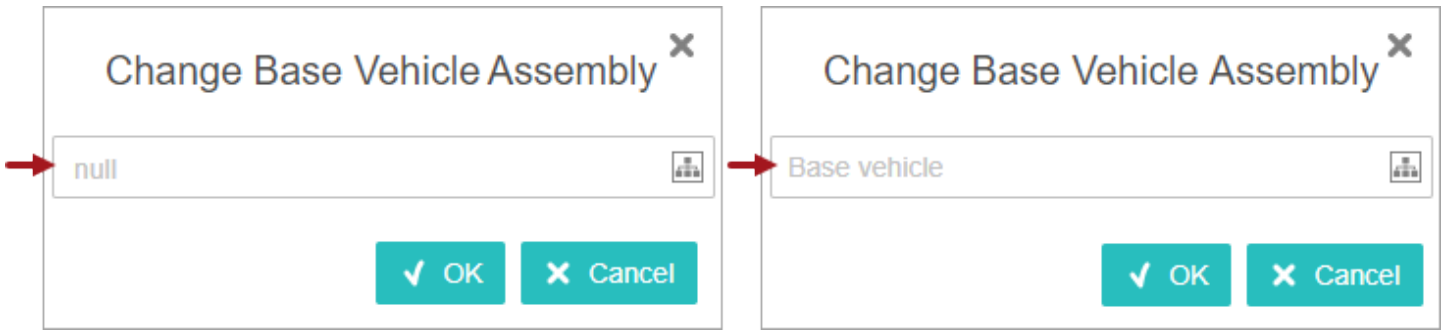
order to identify this business action, the Key parameter within the Node Picker Parameter Properties dialog has to be populated with the same value of 'CopyApplication' as shown within the image on the right.

**Important:** If the exact key is not entered, then the business rule will not properly function when executed.



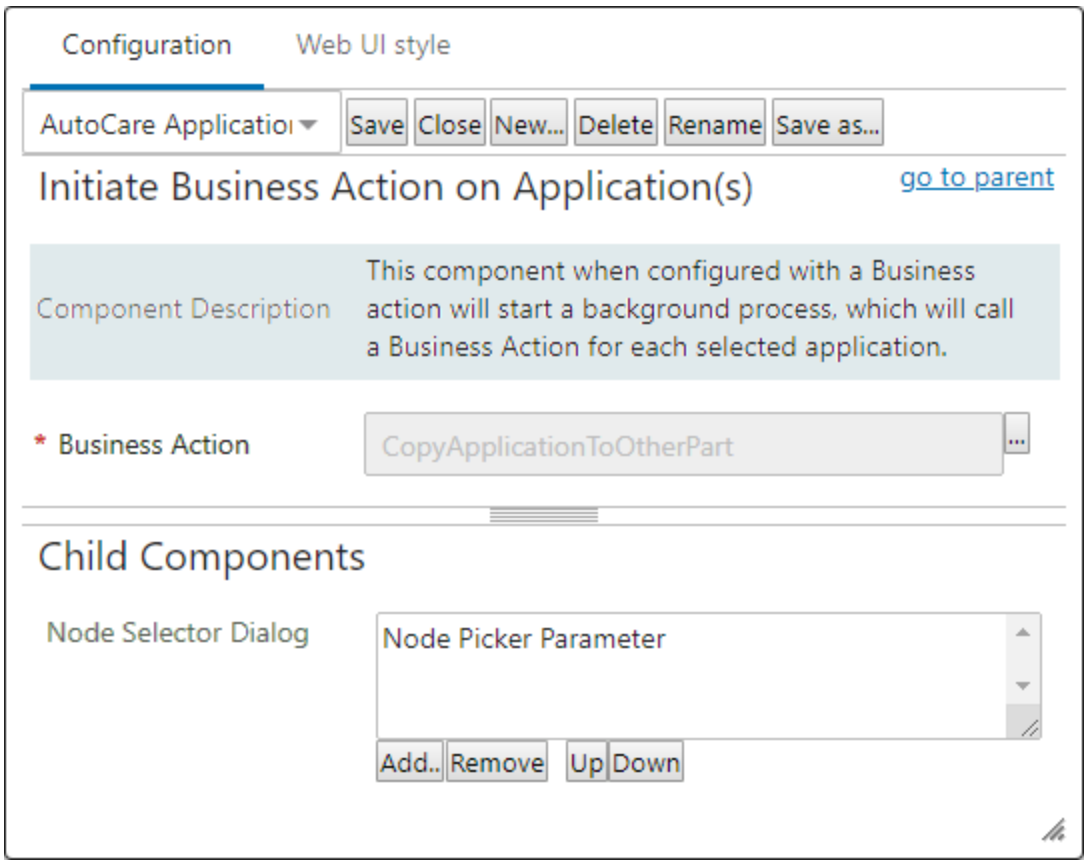




- **Label:** Within the Label parameter, enter the text that will prompt the user as to what they should select when using this business action. For example, when the Label parameter is blank the parameter within the dialog will display as 'null' as shown in the image on the left. If the Label parameter is populated with 'Base vehicle' the parameter within the dialog will display with 'Base vehicle' as shown within the image on the right.




- **Mandatory:** Enabling this parameter will require the user to select a Target node when the Business Action is executed. The **OK** button will not be activated unless a valid node is selected for the execution of the business action.
- **Node Picker Configuration:** In the dropdown next to the Node Picker Configuration parameter, select Node Picker Dialog option and click **Edit**. The 'Edit component' for the 'Node Picker Dialog Properties' dialog will display. Node Picker Dialog allows the user to define the nodes that are presented for selection in the dialog and also allows the user to determine how the nodes are selectable. For more information about the Node Picker Dialog, refer to the **Node Picker Dialog** topic within the **Web User Interfaces** section of **STEP Online Help**.
- **Valid Node Types:** By default, this parameter is populated with the value 'UNKNOWN_TYPE', meaning that all node types are valid for selection. Click the dropdown located beneath the Valid Node Types parameter to select the required node type, and then click the Add button so that the required node type is displayed within the Valid Node Types parameter.
- **Valid Object Types:** By default, this parameter is blank. Click the Add button beneath the Valid Object Types parameter and then select the desired valid object types. This ensures that only the defined object types are allowed for the selection.

5. Click the **Add** button, and the Initiate Business Action on Application(s) Properties dialog will display with the Node Picker Parameter component listed within the Node Selector Dialog parameter.

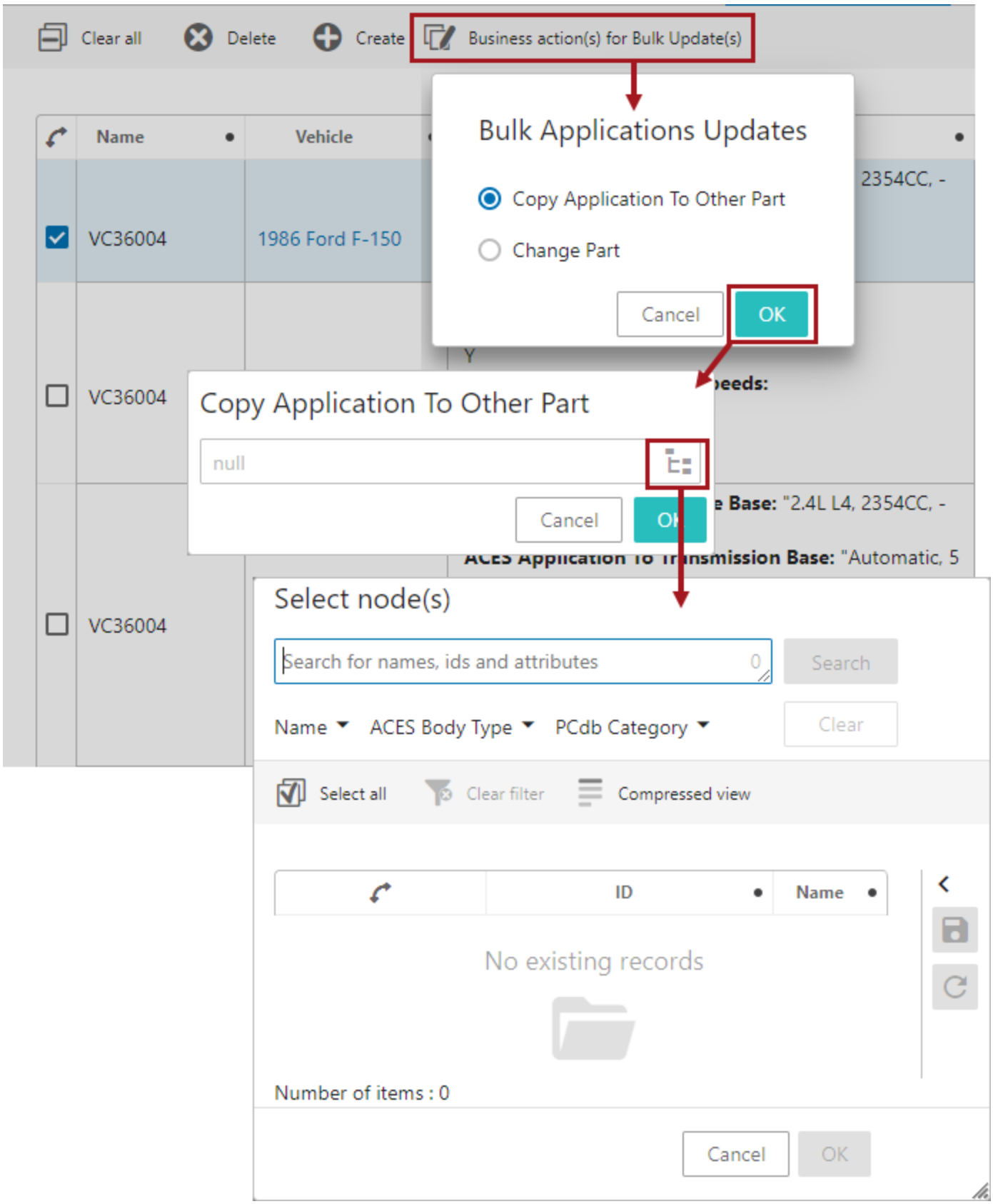



6. Click **Save** button to save the configuration.

## Faceted Node Picker Parameter Component

The Faceted Node Picker Parameter component is configured within the Initiate Business Action on Application (s) Properties dialog. This component assists in executing the defined Business Action that requires an object selection. Faceted Node Picker Parameter component determines that the objects are selectable through the Faceted Node Picker dialog.

Below is a screenshot representing the Faceted Node Picker Parameter component that gets displayed when the Business Action that needs an object selection is initiated for execution.


Clear all
Delete
Create
Business action(s) for Bulk Update(s)
Name
Vehicle
VC36004
1986 Ford F-150
2354CC, -
VC36004
VC36004
Bulk Applications Updates
Copy Application To Other Part
Change Part
Cancel
OK
Copy Application To Other Part
null
Cancel
Select node(s)
Search for names, ids and attributes
Search
Name
ACES Body Type
PCdb Category
Clear
Select all
Clear filter
Compressed view
ID
Name
No existing records
Number of items : 0
Cancel
OK

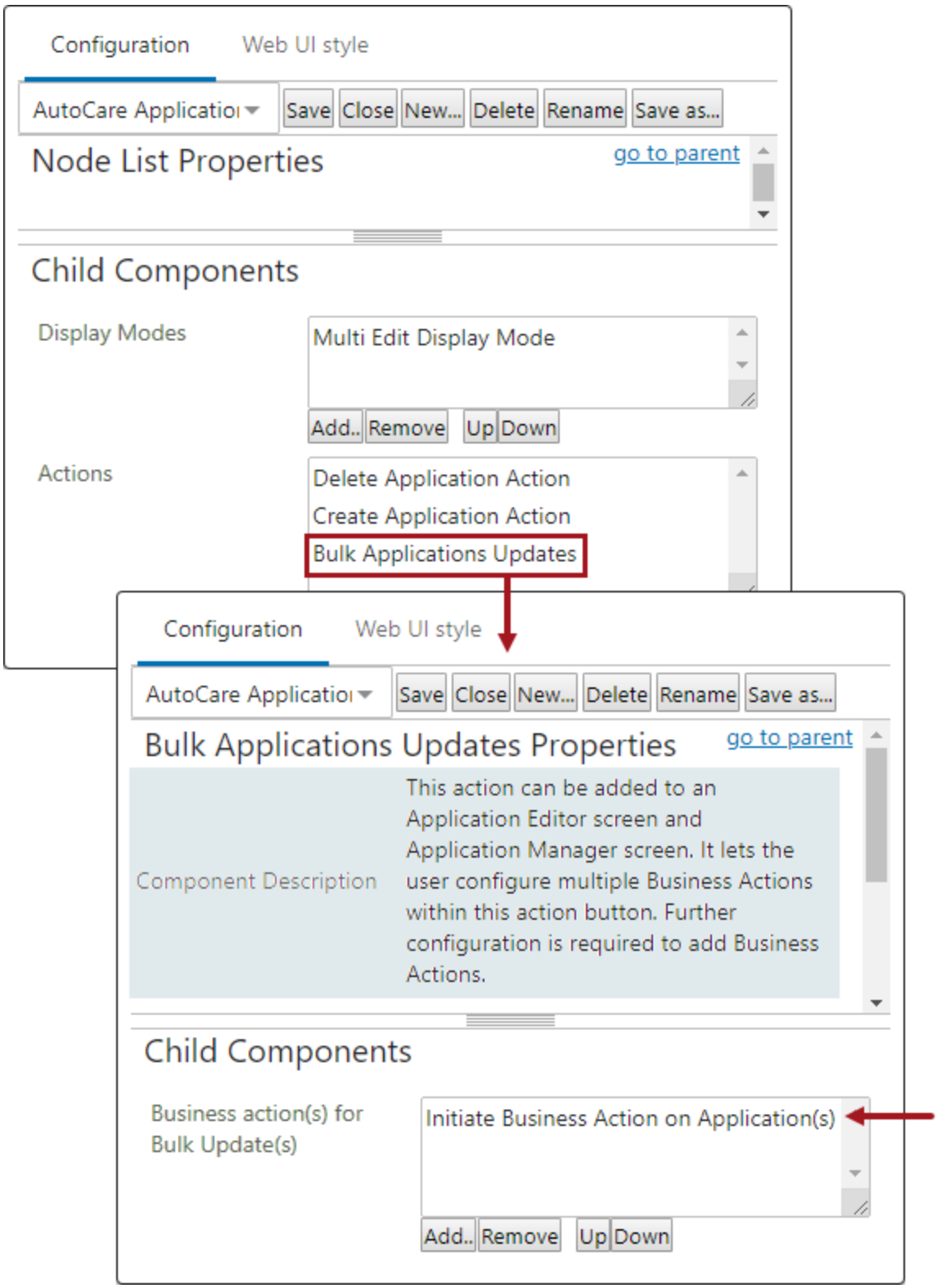
### Configuring the Faceted Node Picker Parameter Component

Follow the steps described below to add and configure the Faceted Node Picker Parameter component that gets displayed when the user executes the Business Action (that needs an object selection) via **Business action(s) for Bulk Update(s)** action button

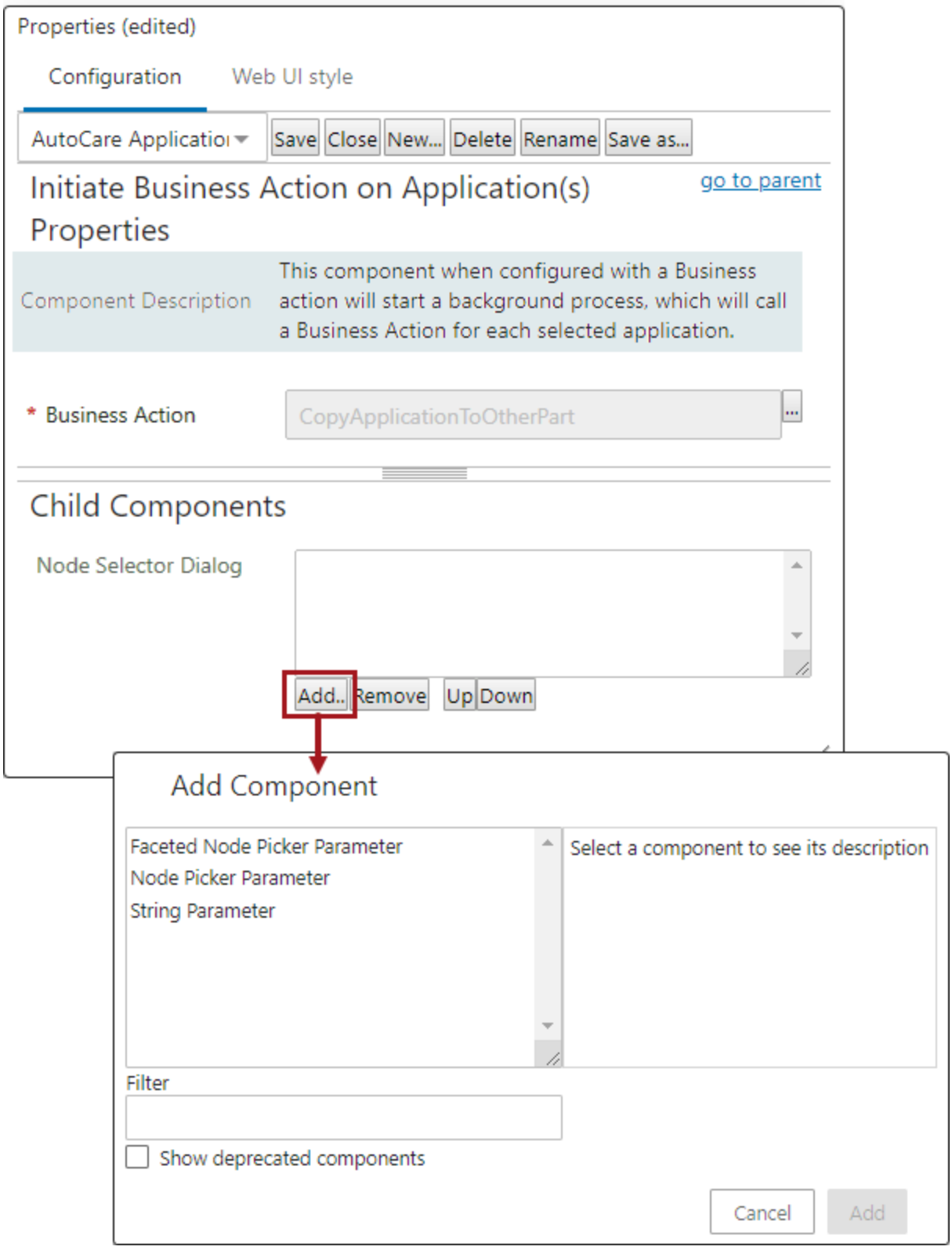
.

The example below shows a configuration of the Faceted Node Picker Parameter component in an Application Editor Screen:

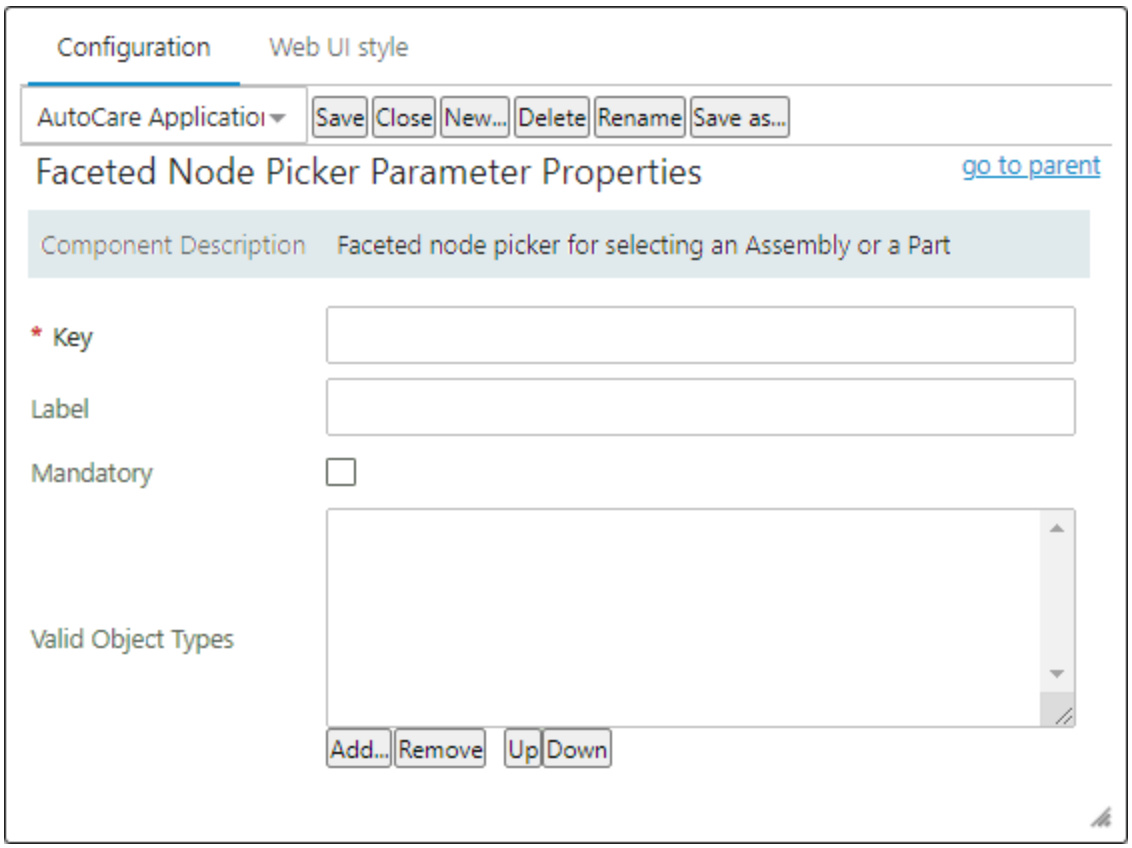
1. In the Web UI design mode, select the Bulk Applications Updates action button configured within the Node List component of the required screen (Application Editor Screen / Application Manager Screen).
2. Double click the Initiate Business Action on Application(s) component if it already exists, otherwise, create a new one by clicking the **Add** button within the Business action(s) for Bulk Update(s) parameter. For more information, refer to **Bulk Applications Updates Component** topic.




3. Under the Child Components section on the Initiate Business Action on Application(s) Properties, click the **Add** button under the field for Node Selector Dialog parameter. The Add Component dialog will display.

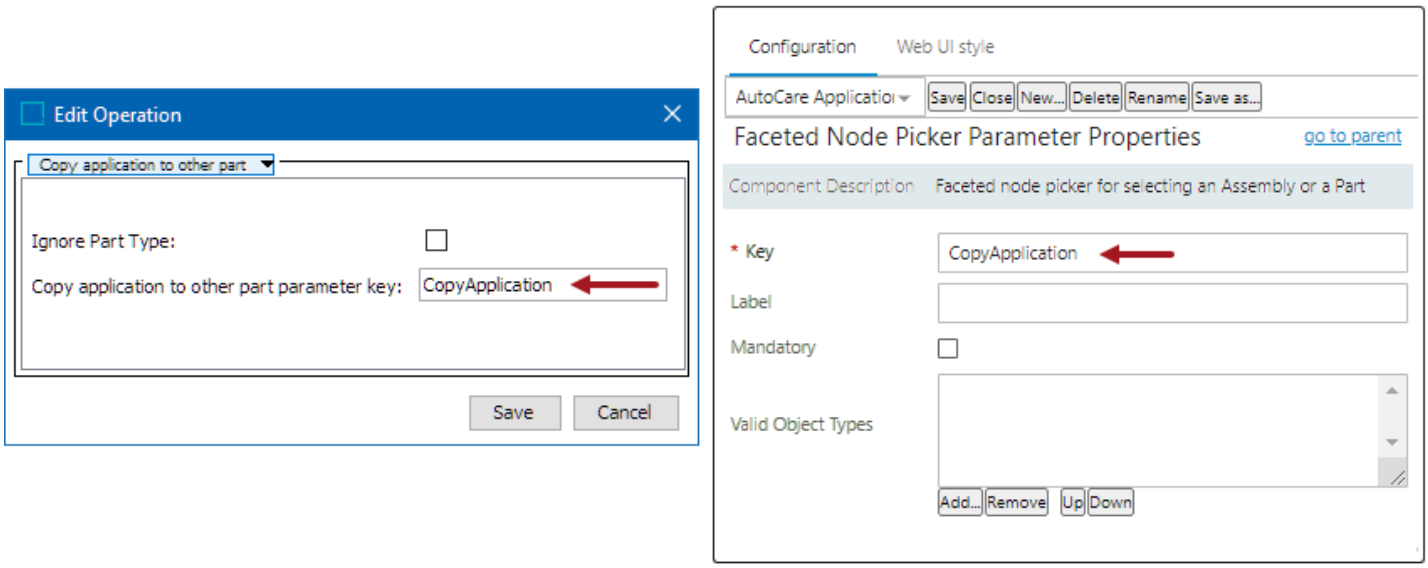



4. Select Faceted Node Picker Parameter component and click **Add**, and the Faceted Node Picker Parameter Properties dialog will display. Define the following parameters:

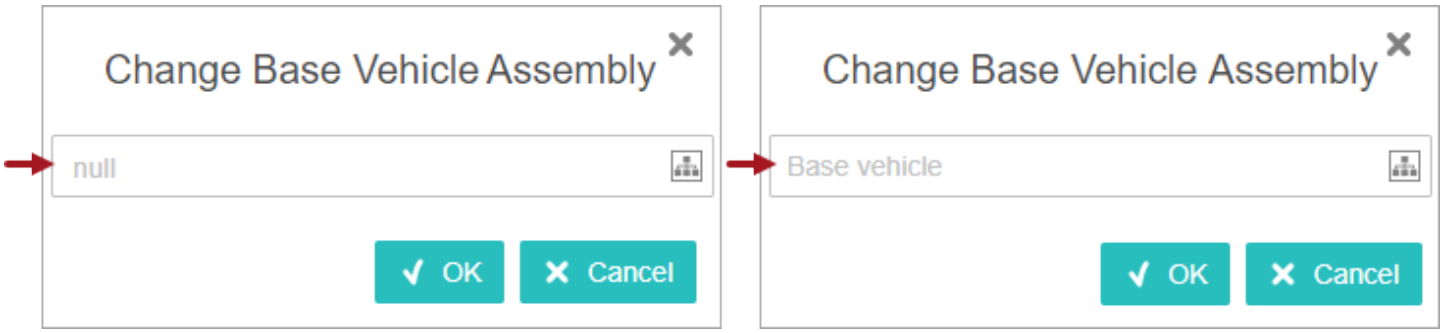



- **Key:** By default, this required parameter is blank. The value entered for this field is used to connect a business rule in the system to this Web UI component. Type in the exact characters (key) that is entered within the business rule configured within the Business Action parameter of Initiate Business Action on Application(s) Properties. Uniqueness is the only restriction for this key and it is case sensitive. For example, the business Action 'Copy application to other part' has the value 'CopyApplication' populated within the 'Copy application to other part parameter key' parameter as shown in the image on the left. In order to identify this business action, the Key parameter within the Faceted Node Picker Parameter Properties dialog has to be populated with the same value of 'CopyApplication' as shown within the image on the right.

**Important:** If this key is not entered exactly in both places, then the business rule will not properly function when executed.

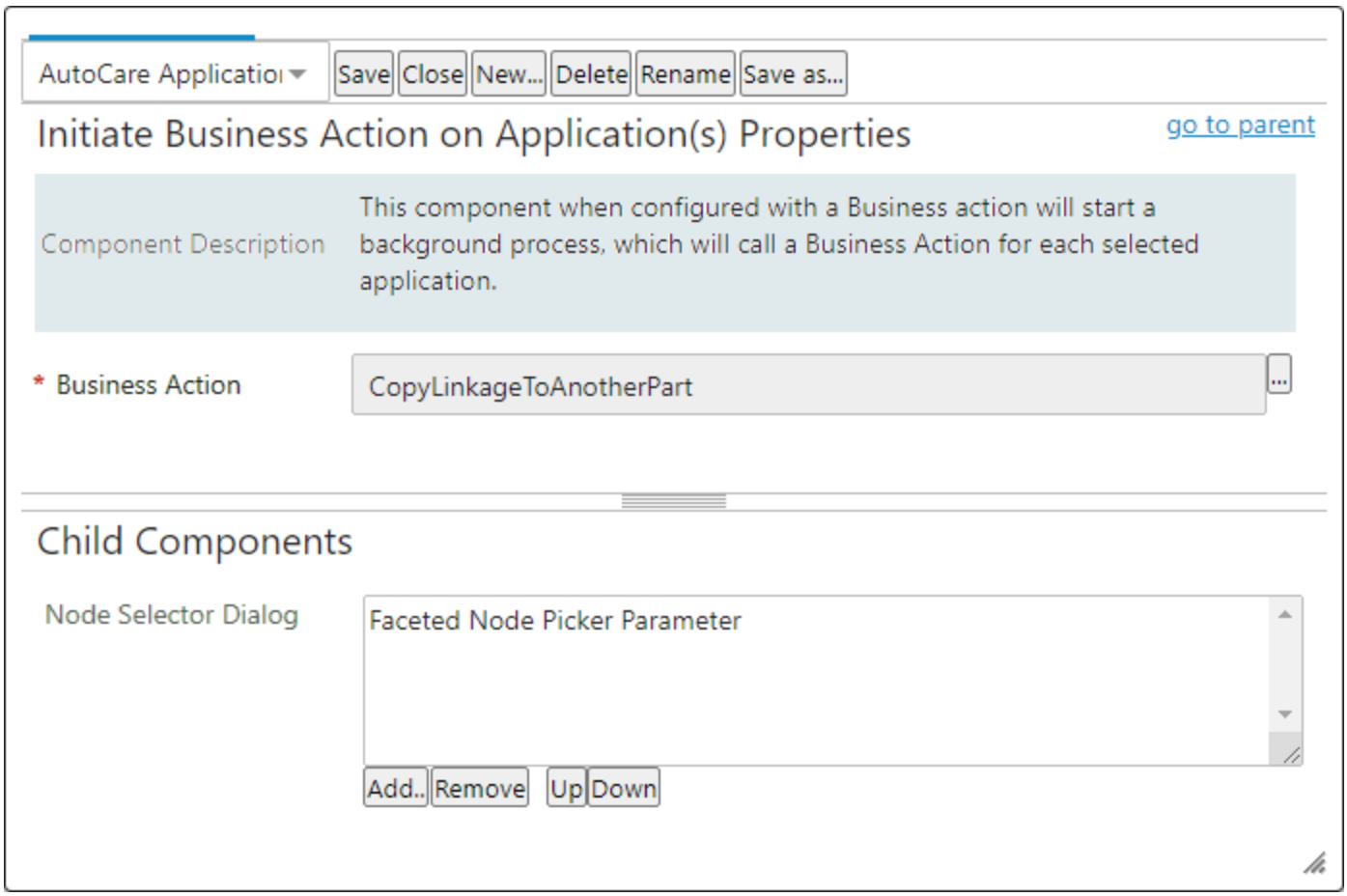



- **Label:** Within the Label parameter, enter text that will prompt the user as to what they should select when using this business action. For example, when the Label parameter is blank the parameter within the dialog will display as 'null' as shown in the image on the left. If the Label parameter is populated with 'Base vehicle' the parameter within the dialog will display with 'Base vehicle' as shown within the image on the right.

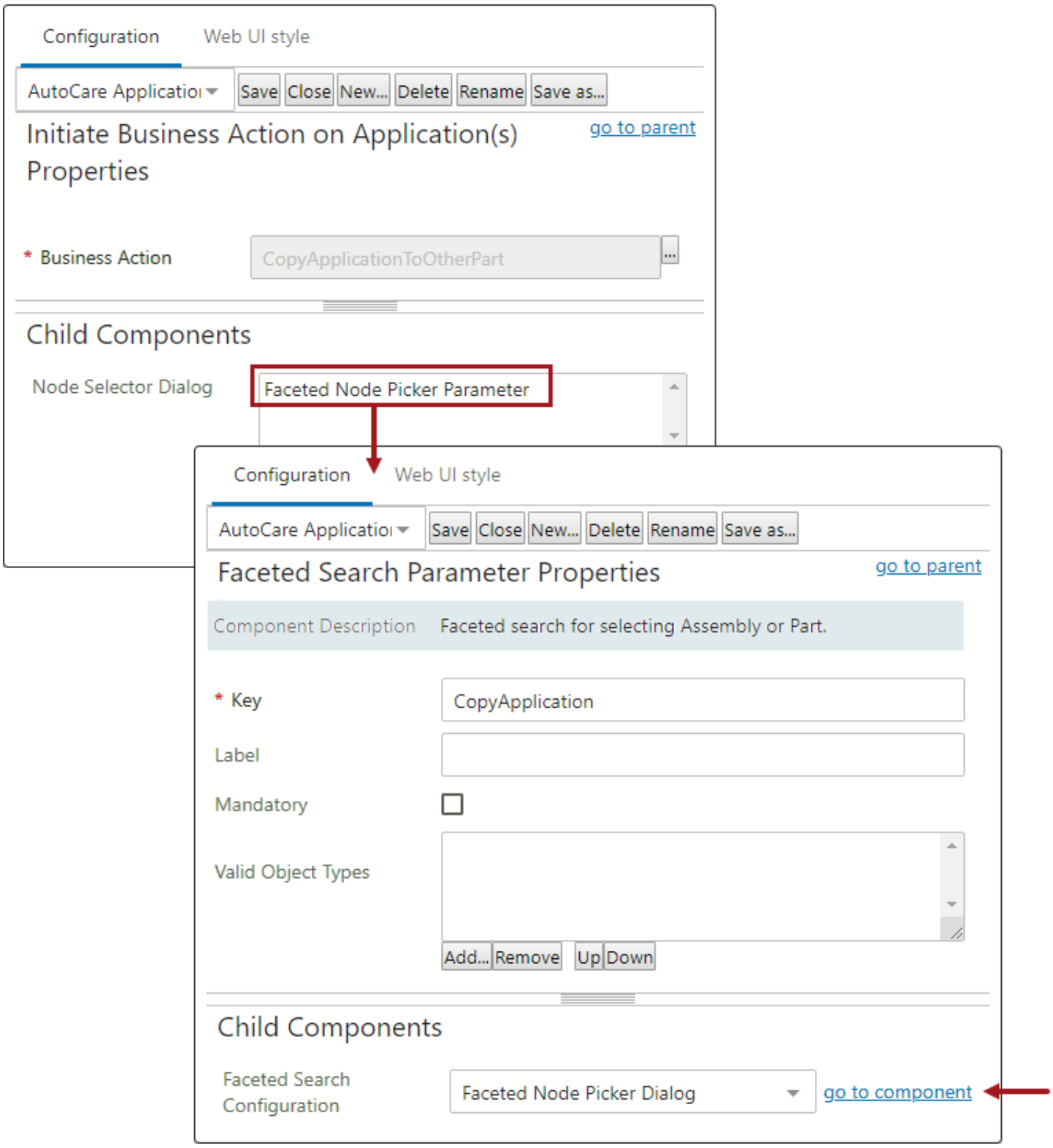



- **Mandatory:** Enabling this parameter will require the user to select a Target node when the Business Action is executed. The **OK** button will not be activated unless a valid node is selected for the execution of the business action.
- **Valid Object Types:** By default, this parameter is blank. Click the **Add** button beneath the Valid Object Types parameter and then select the desired valid object types. This ensures that only the defined object types are allowed for the selection.

5. Click the **Add** button, and the Initiate Business Action on Application(s) Properties dialog will display with the Faceted Node Picker Parameter component listed within the Node Selector Dialog parameter.




6. To further define a search criteria, double click the **Faceted Node Picker Parameter** component listed within the Node Selector Dialog parameter.




7. Under the Child Component section go to > Faceted Search Configuration > Faceted Node Picker Dialog > and click **go to component**. The 'Faceted Node Picker Dialog Properties' dialog will display. Faceted Node Picker Dialog allows the user to define a search criteria that are presented for selection. For more information

about the Faceted Node Picker Dialog, refer to the **Faceted Node Picker Dialog** topic within this section.

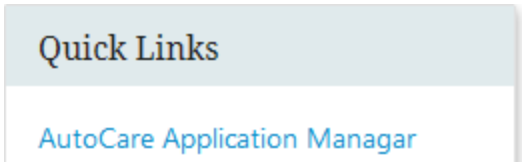
8. Click **Save** button to save the configuration.

# 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.



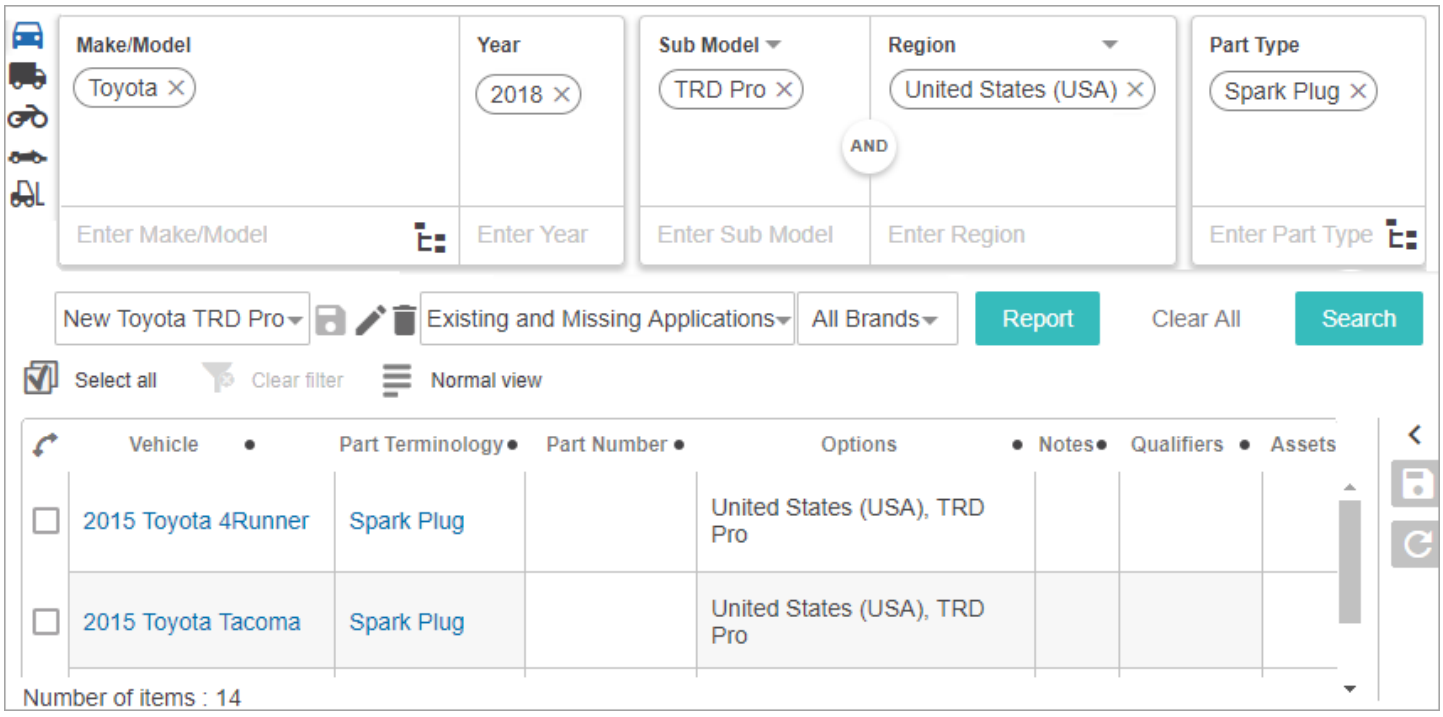

**Important:** An error will occur when trying to access an Application Manager screen if the Vehicle table classifications have not been properly configured. When this error displays, the user must log out and then back in to the Web UI. Once the Vehicle reference data is imported, the link to the Application 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.

**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, refer to 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.




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.

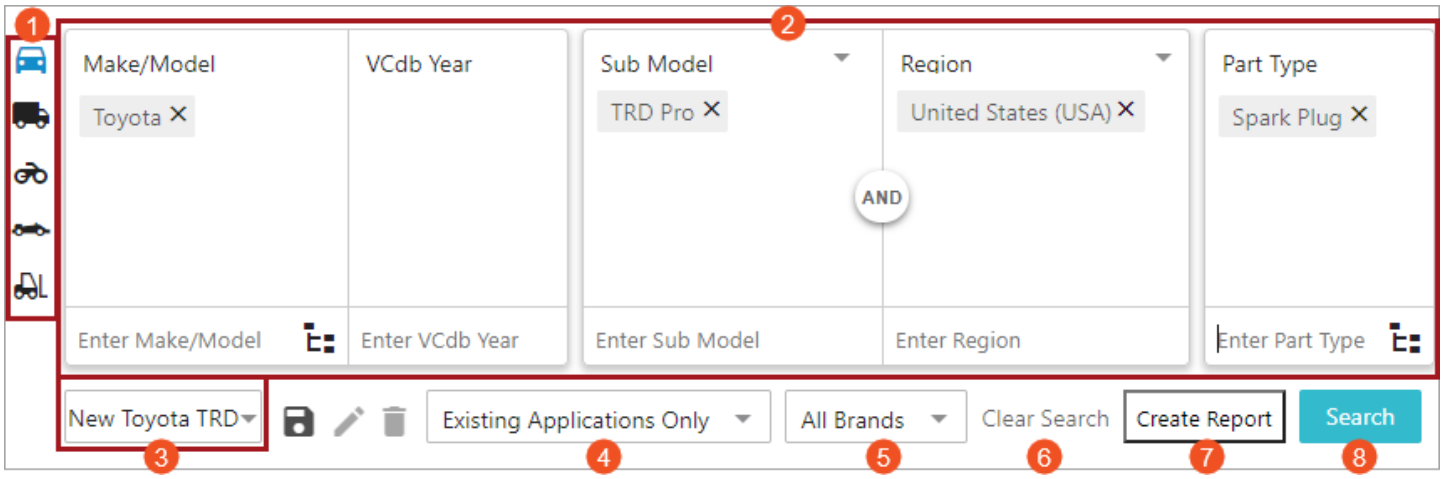
# Intelligent Search Interface

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, refer to 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:




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. Whenever there are more than five vehicle type icons, an ellipsis button will be shown below the fifth icon that allows users to view more Vehicle Type Search Panels. For a list of available icons, and more information about the Vehicle type icon, refer to the **Configuring Application Manager** topic within this guide.
2. **Search Boxes:** This search section can be made up of five available 'search box types' (Make/Model, Attribute, Options Group, Part, 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, refer to the **Using Intelligent Search Interface** 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, refer to 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, refer to 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, refer to 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').

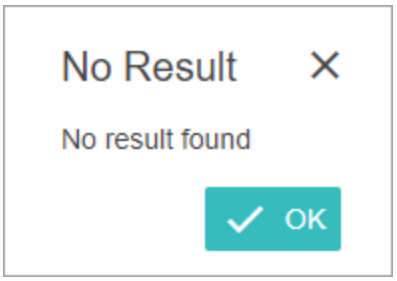
6. **Clear Search link:** Allows users to easily clear all search criteria, results, and filters applied to Results Table. The Clear All link remains disabled until at least one criterion for one of the search boxes is populated. When text is typed into a search box, but is not yet selected, clicking 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.
7. **Application Coverage Report button:** This action button has the ability to do the following functions:
   - Allows users to export an Application Coverage Report in an Excel spreadsheet (XLSX file type)
   - Allows users to export a report of invalid application records in an Excel spreadsheet (XLSX file type) and initiate the invalid application records into a workflow

   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, refer to the **Application Coverage Report** topic within this guide.

**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.

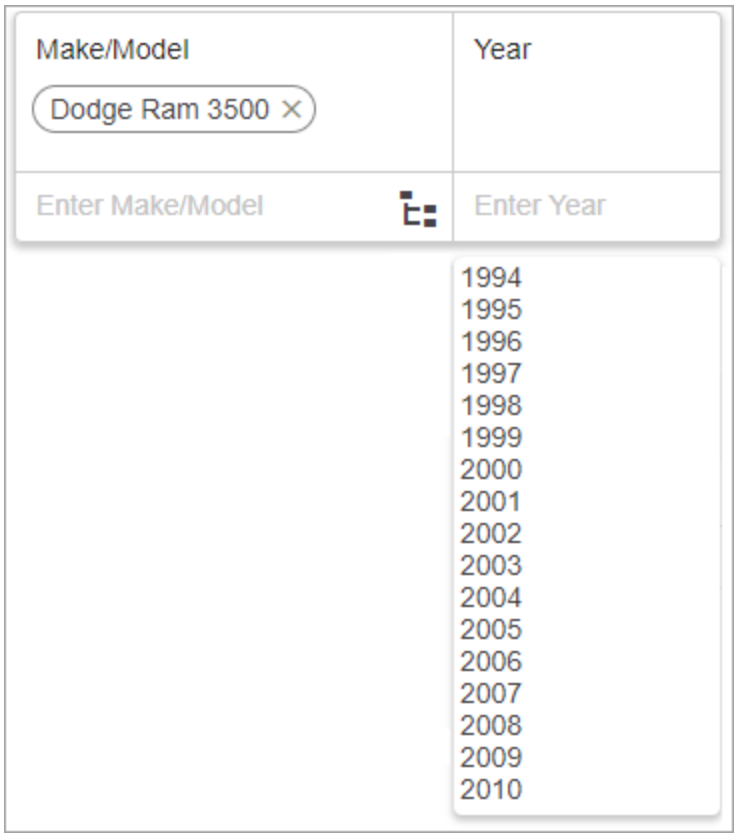


## Using Intelligent Search Interface

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.

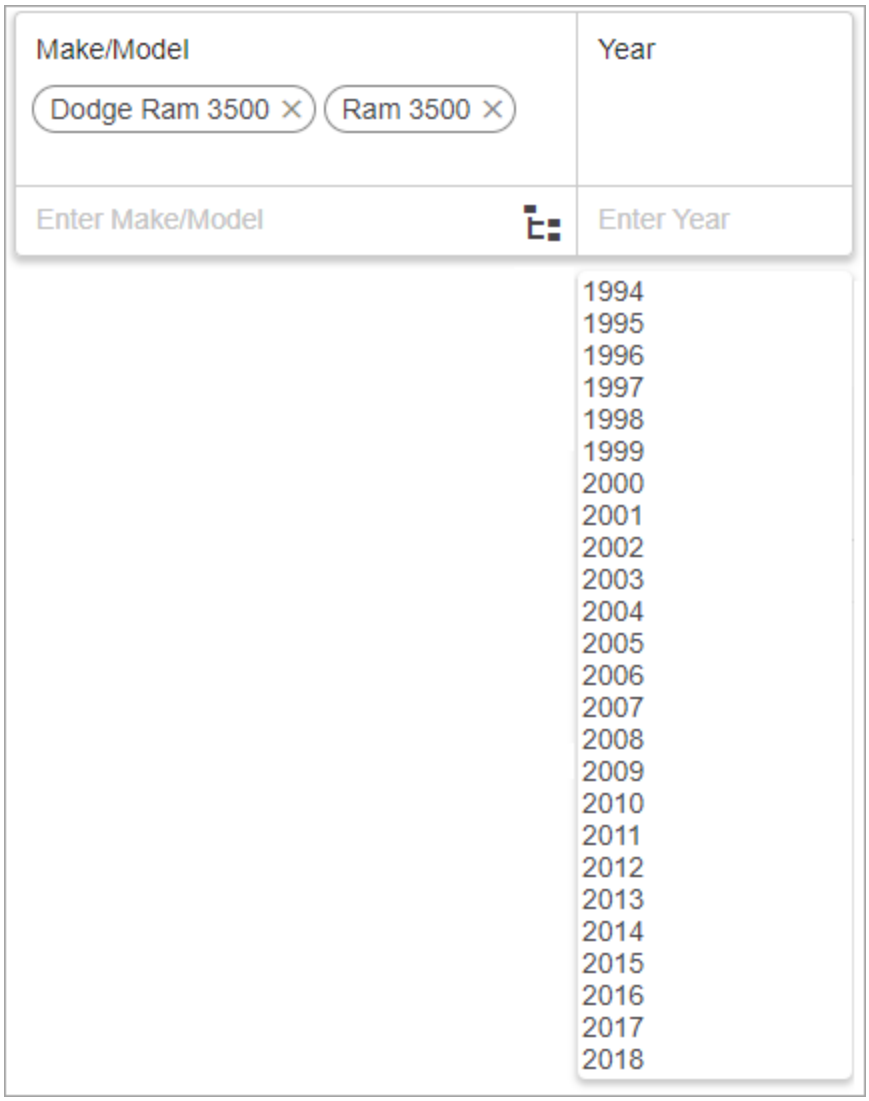
Within an Application Manager screen, the Vehicle Type Search Panel can be configured to display up to four search box types. Because the Options Group search box type allows for up to two Options search boxes, up to five search boxes can be displayed. Within each search box, a typeahead field is used 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).

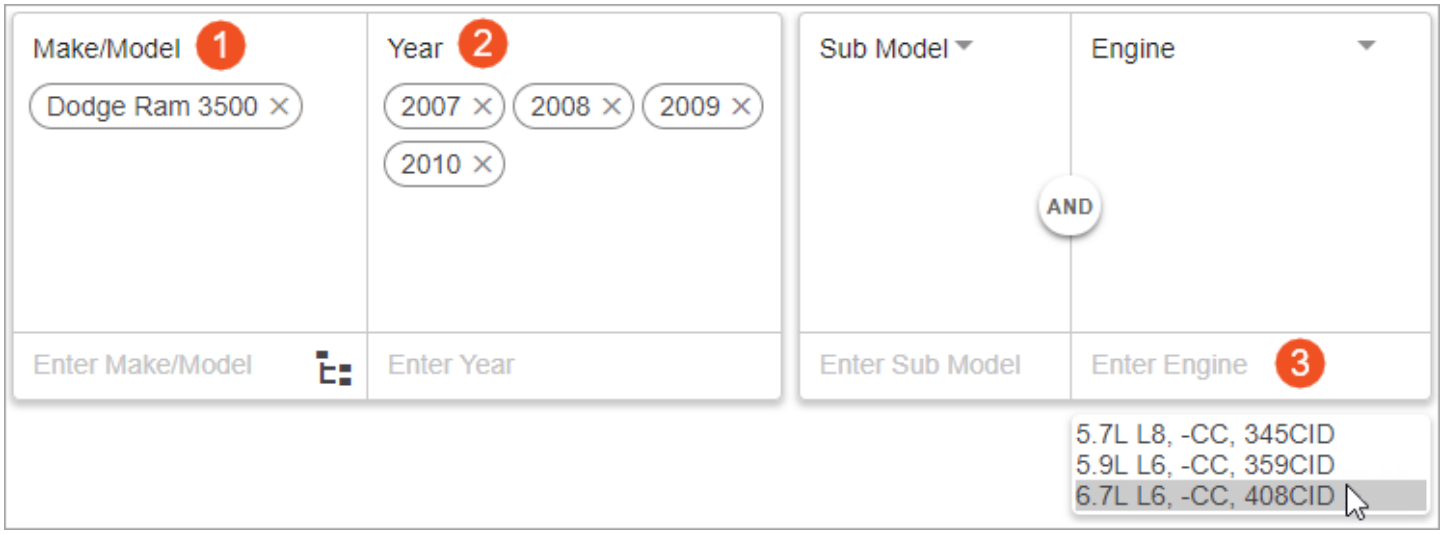


**Note:** The Vehicle Type Search Panel, and its search boxes, have many configuration options. If your screen, or search criteria dropdown options display differently than the examples within this section, contact your Web UI administrator. For more information on configuring an Application Manager, refer to the **Configuring Application Manager** topic within this guide. For information about the general layout and features of the Application Manager, refer to the **Application Manager** topic within this guide.

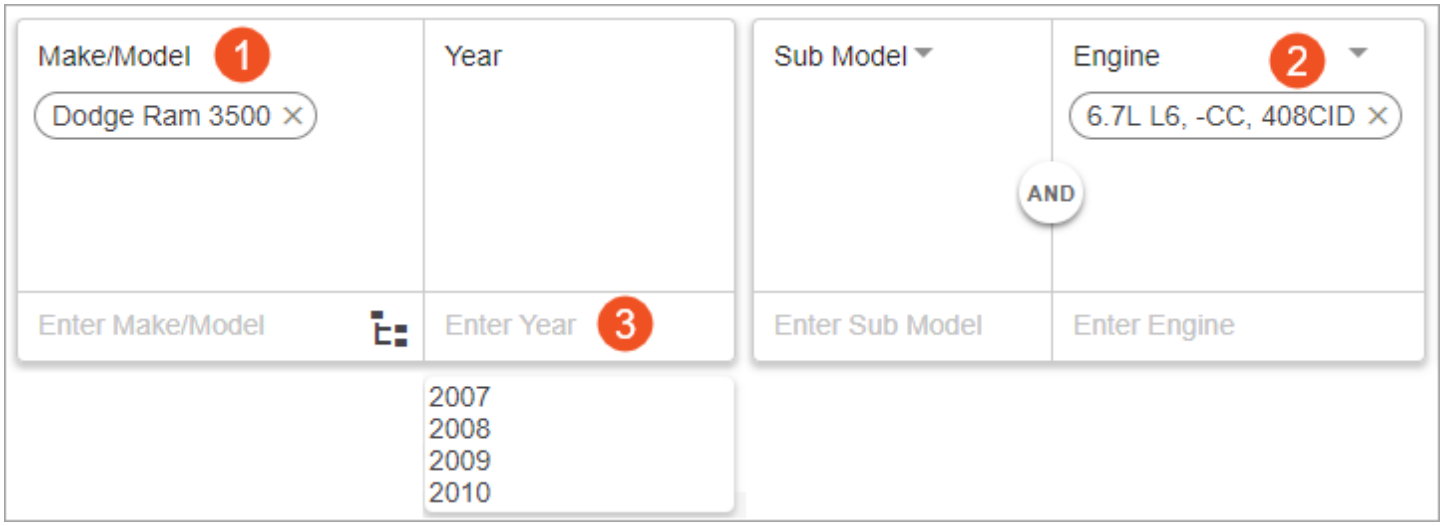
However, if the Ram 3500 is added to the Make/Model search box, then the available years display as 1994-2018 (as shown below).



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.




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 was manufactured within the selected vehicle (as shown below).




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 Part Search Box**

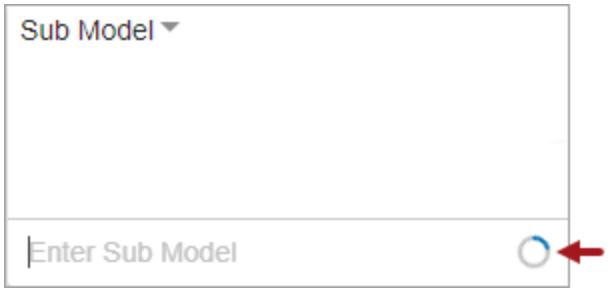
## Using Search Boxes

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

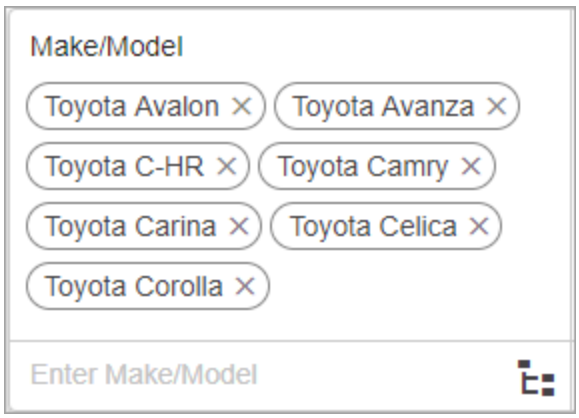
The cursor is activated for each search box either by clicking within the box itself or within the typeahead field at the bottom of the box. Text typed into the field is used to filter and then display only valid criteria within a dropdown list. Typing an asterisk (*) displays any available criteria 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.

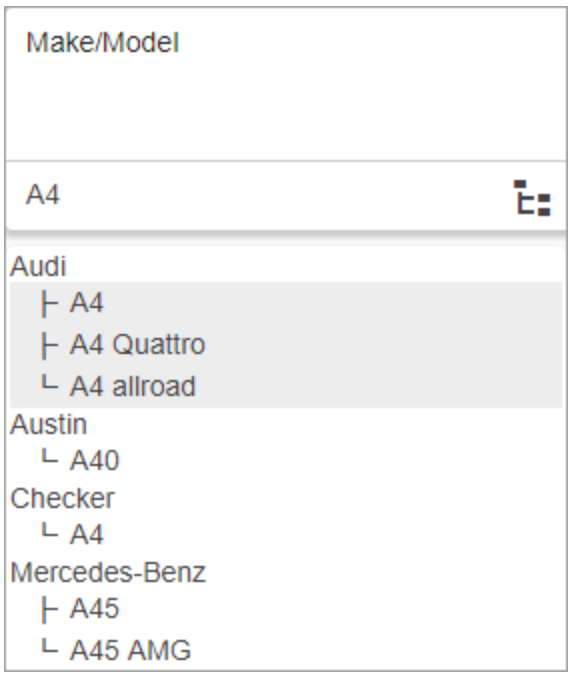
The Tab key on the keyboard can be used to move the cursor from left to right across different search boxes. Holding the keyboard Shift key down while pressing the Tab key will move the cursor from right to left.

**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.

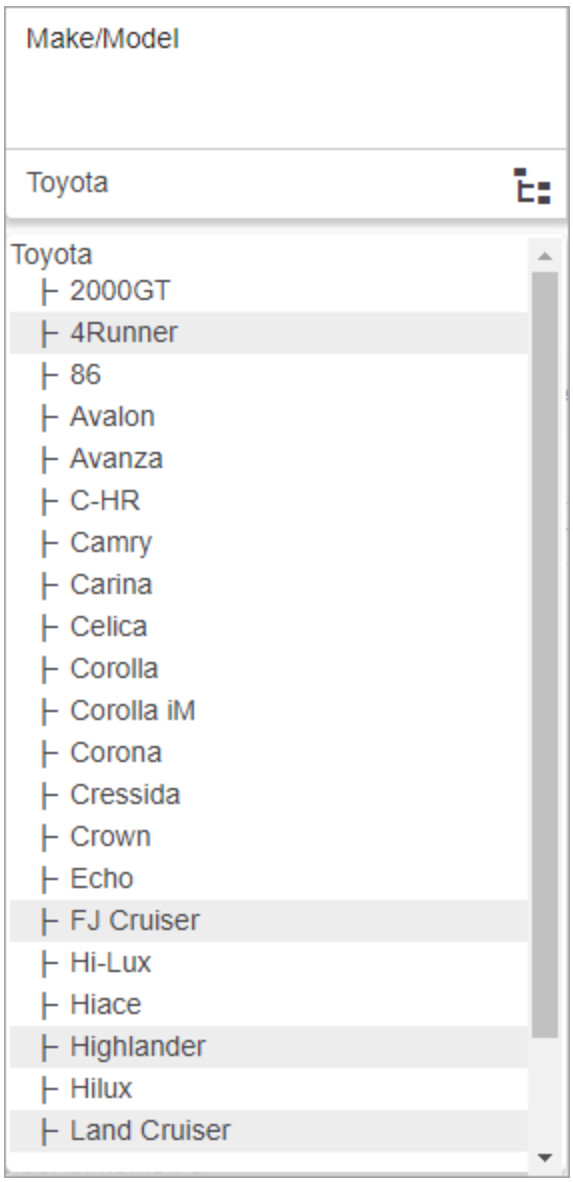
## Selecting Multiple Criteria

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.

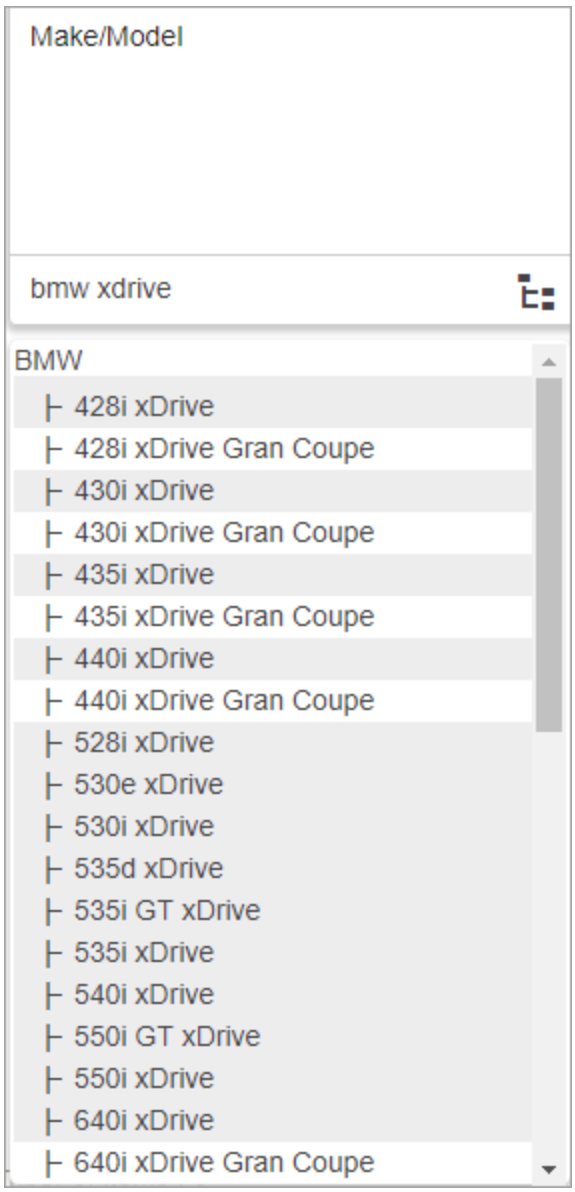
In the example below, after a list of valid criteria options displayed are within the dropdown list, the down arrow key was used to highlight the Audi A4 option, the Shift key was held down, and the down arrow key was used to highlight the two additional Audi model options.




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**

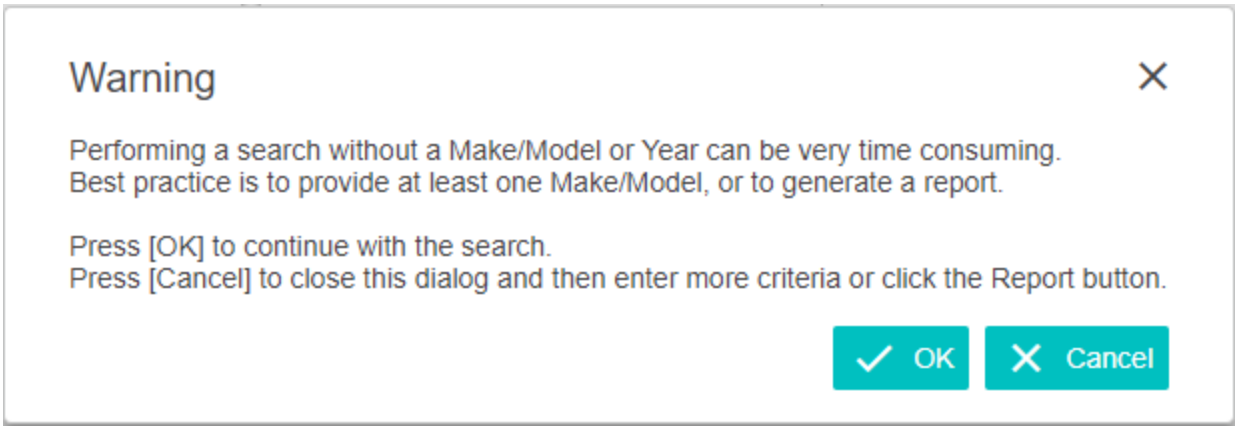
Criterion can be individually removed from a search box, or the Clear All link can be used to remove all selected criteria. To remove a criterion from a search box using the mouse, click the 'X' at the right of a criterion tile. Optionally when the Tab key on the keyboard is used to select a criterion, 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.

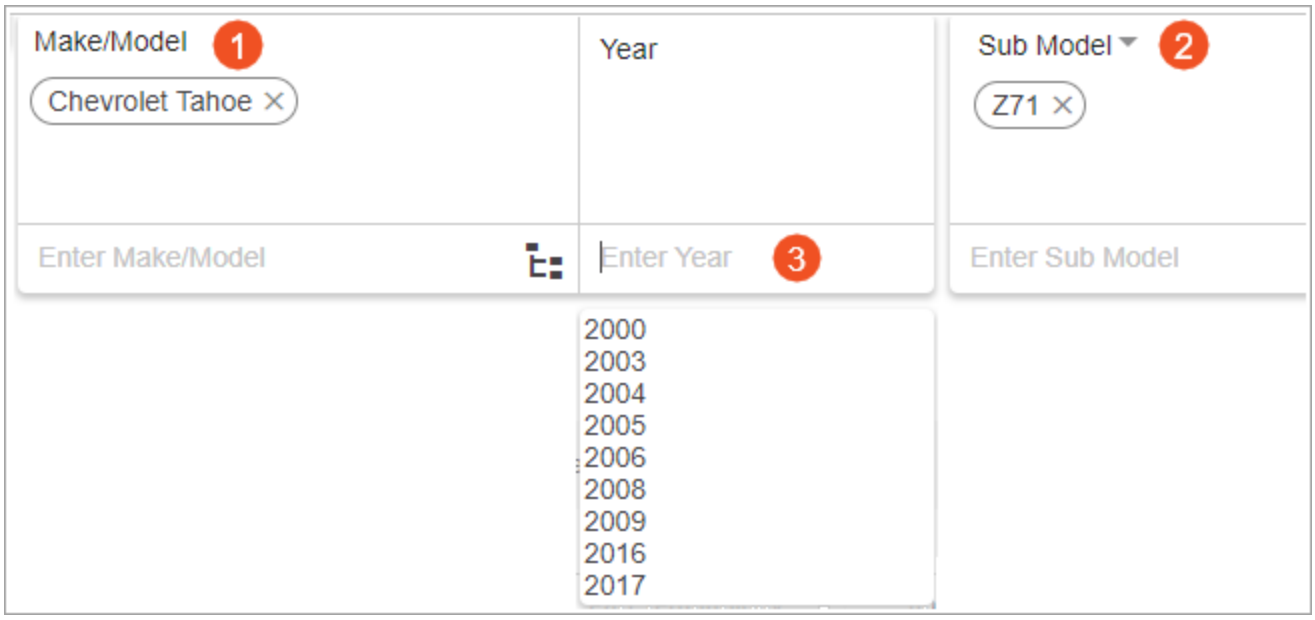
When the use of broad search criteria is necessary, recommended practice is to use the Application Coverage Report by clicking the Report button within the Application Manager. For more information about using the Report button, refer to the **Application Coverage Report** topic.

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.

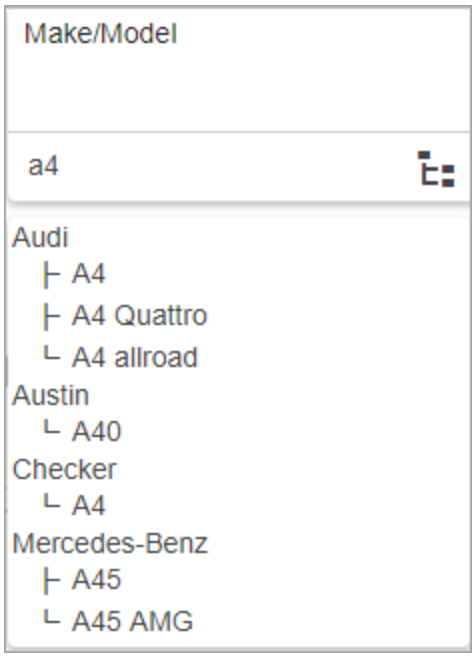



**Note:** To remove criteria that is displayed within a search box, click the 'X' to the right of the value, or use the Clear All link to clear all criteria within all search boxes.

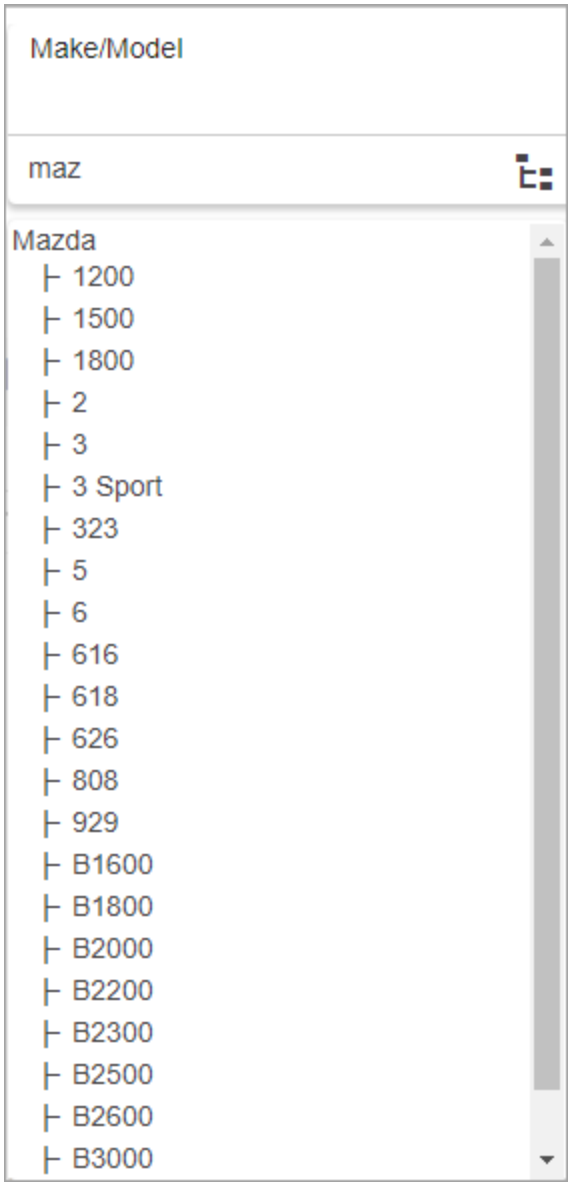
### Using Make/Model Search Box

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, refer to the **Using Search Boxes** topic within this guide.

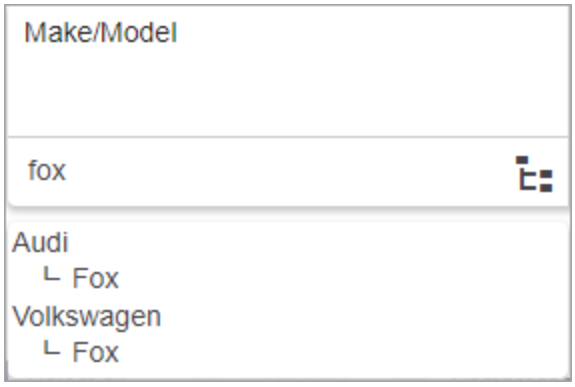
Within the Make/Model Search Box, the typeahead field can be used to search for make and/or model criteria. The dropdown list will display the criteria options alphabetically by Make, and then by Model. Models are always displayed below their respective Make. For example, if 'a4' is typed into the field, 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).



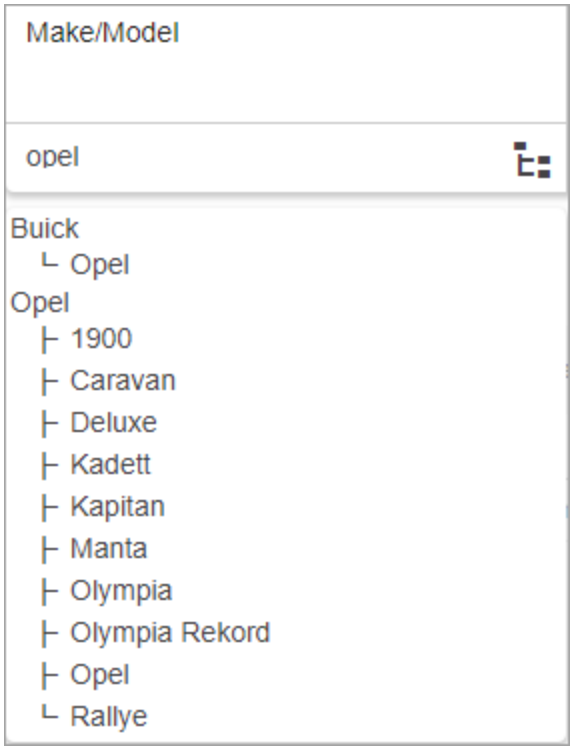
If the text typed in matches a make only, then all Models below that make will be displayed. For example, if 'maz' is typed into the field, then the dropdown list will display the Mazda make with all its available Models listed below.



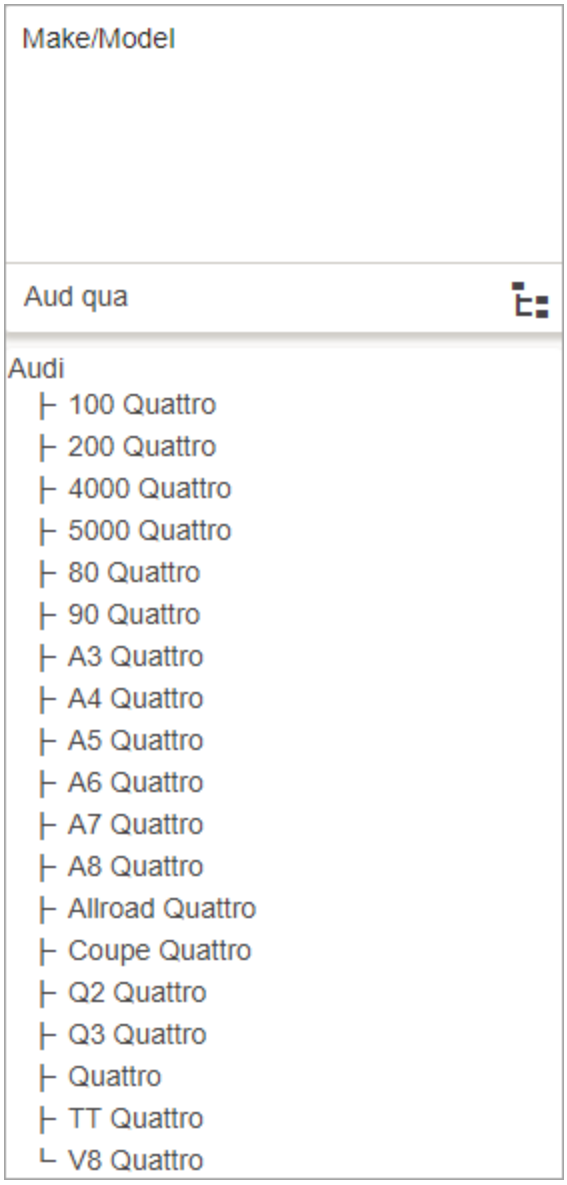
If the text typed in matches a model only, then the make for the matching model is displayed with only the model that matches the search string. For example, if 'fox' is typed into the field, then the dropdown list will display the Audi and Volkswagen makes with the 'Fox' Models listed below.



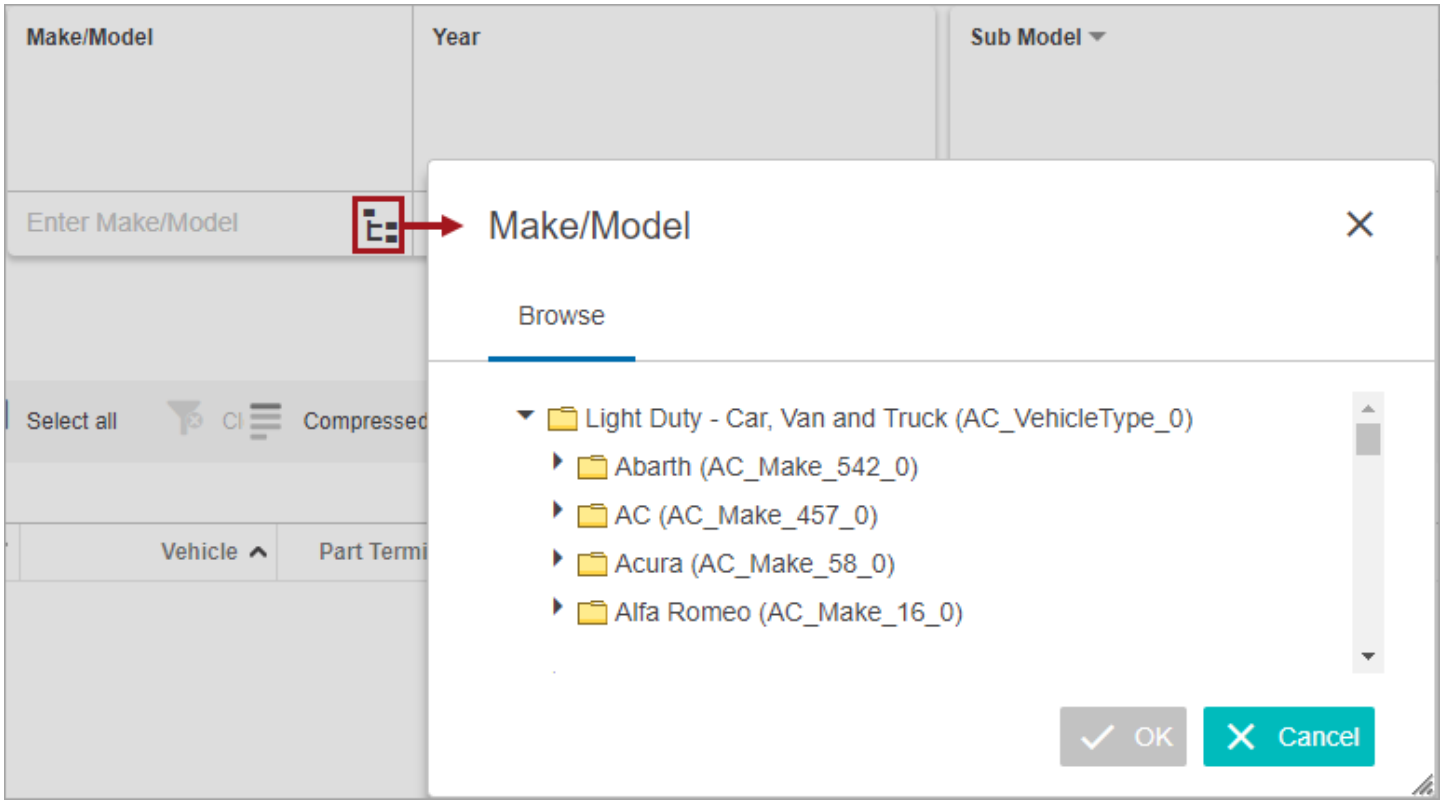

If the text typed in matches the make AND Model, then the matching makes and/or Models will display within the dropdown for selection. For example, if 'opel' is typed into the field, then the dropdown list will display the Buick make with the Opel model beneath, and then the Opel make will display with all of its available Models below.



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.'



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.

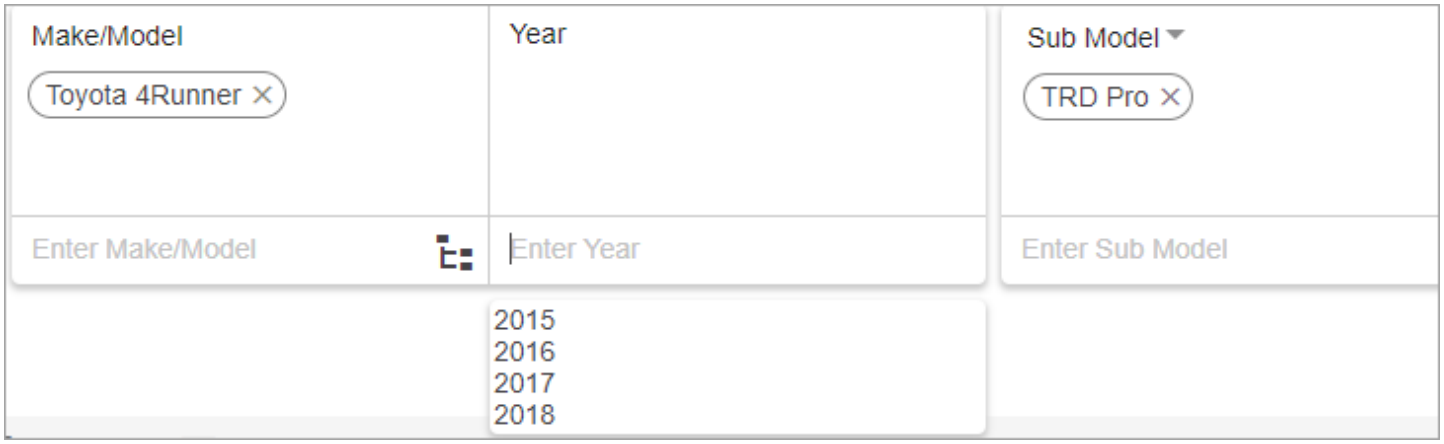



**Important:** Recommended practice is to provide at least one value for the Make/Model search box, otherwise longer than expected wait times can occur. When the use of broad search criteria is required, recommended practice is to use the Application Coverage Report by clicking the Report button within the Application Manager. For more information about running a search without the Make/Model search box populated, refer to 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, refer to 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).




**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.

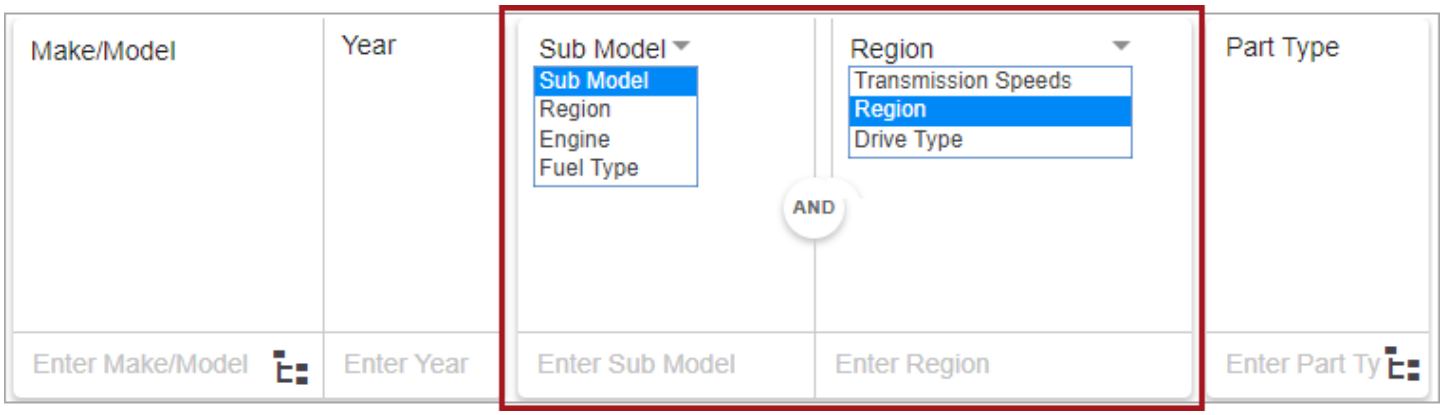
For more information about synchronized search boxes, refer to the **Search Box Synchronization** topic within the **Using Intelligent Search Interface** section of this guide.

### 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, refer to the **Using Search Boxes** topic within this guide.

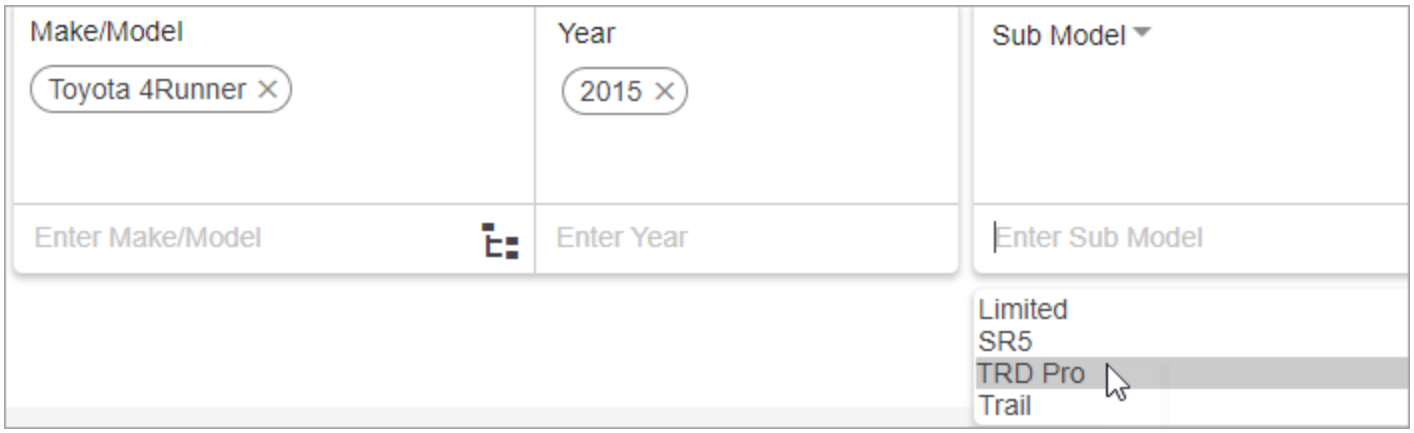
The Options search boxes can be configured to allow users to select up to two attribute and/or reference type options. The attribute and/or reference type option dropdown is located in the top left corner of the Options search box. Clicking the down arrow to the right of the Options search box label, 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).



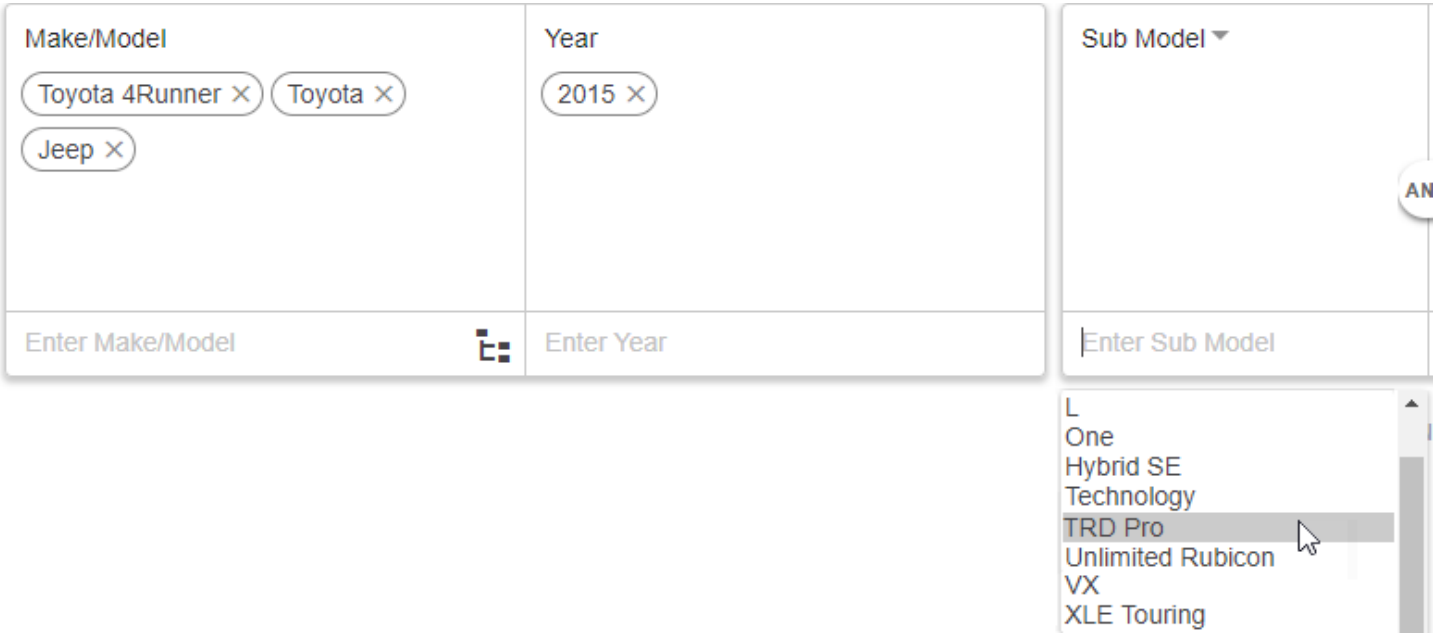

A Search Panel can have up to two Options search boxes. When two Options search box types are configured, an AND / OR toggle button displays between the two boxes. This toggle button allows users to determine if the search criteria within the two Options search box types should be inclusive or exclusive 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.

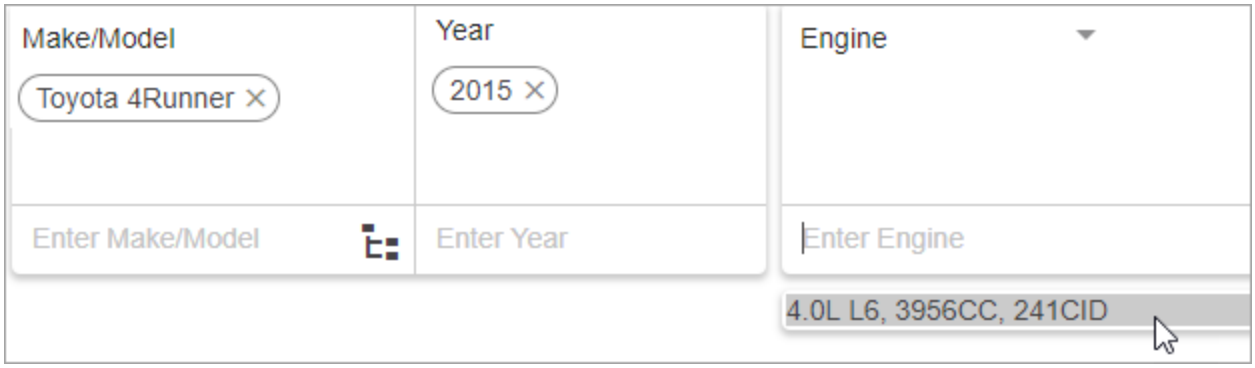
The Options search box typeahead field results are dependent upon selections made in the Make/Model and Year search boxes. In the example below, the Toyota 4Runner is added to the Make/Model search box, and the 2015 year is added to the Year search box, allowing the Sub Model search box dropdown to only 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.



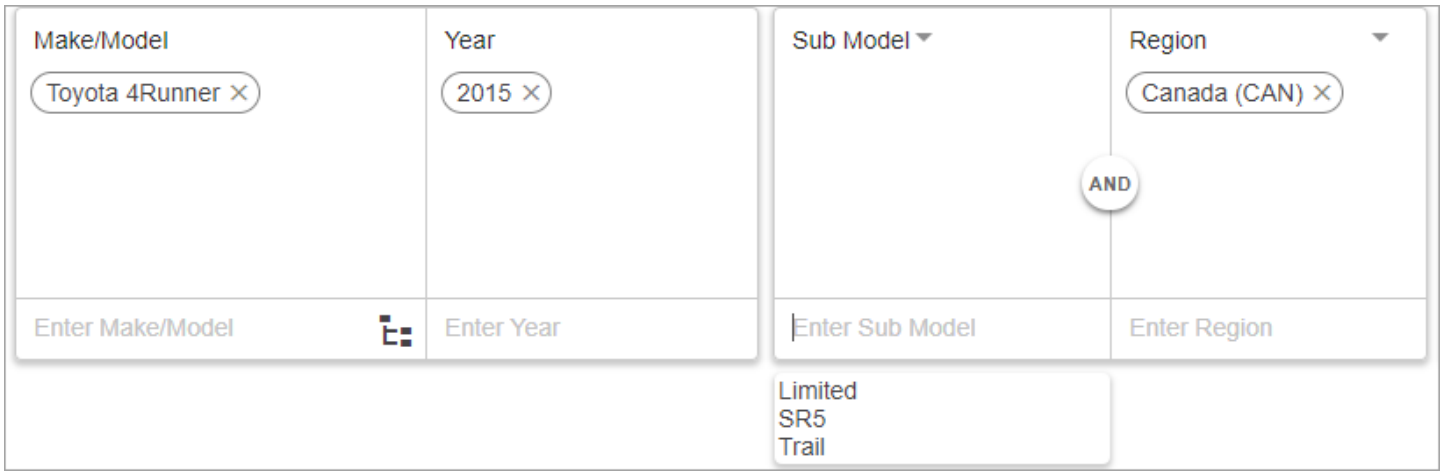

Adding additional makes or models to the Make/Model search box will expand the Options search box dropdown results. In the example below, the Toyota and Jeep makes have been added, and result in all Toyota, Jeep, and 4Runner sub models displaying within the Sub Model Options search box dropdown. (Though the long list of available sub models cannot be found 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.)



By changing the Option selection from Sub Model to Engine, the results dropdown only displays the one engine available for the 2015 4Runner.




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.



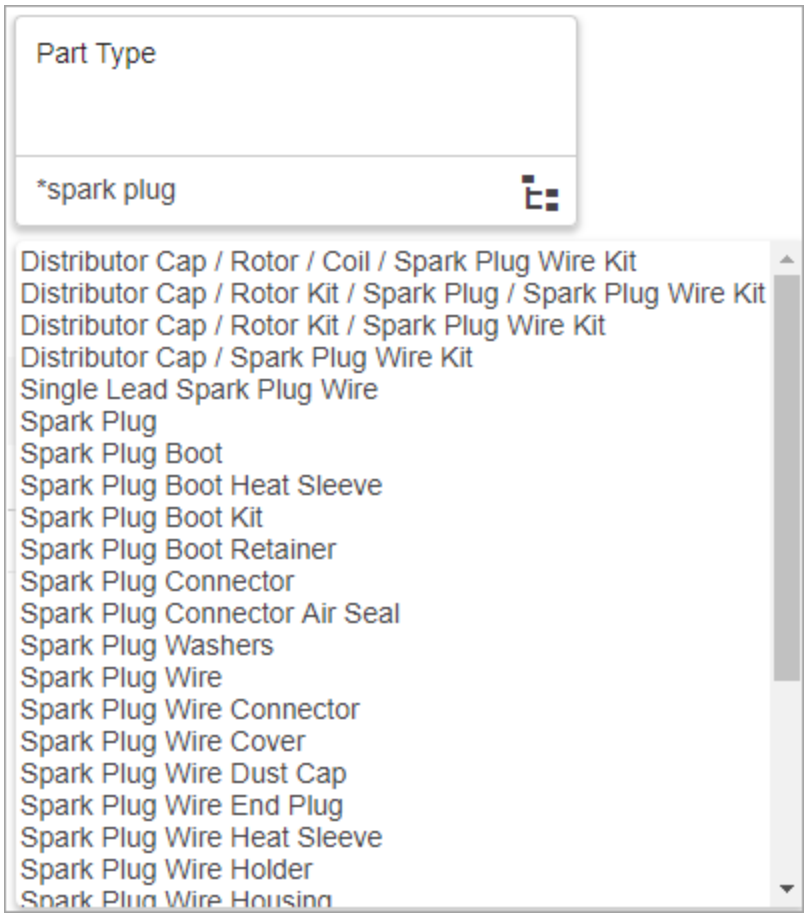

**Note:** The Options search boxes will not automatically display the suggested dropdown options until at least one value is added to the Make/Model search box.

Optionally, the Report button can be used to export these results to an Excel file. For more information, refer to the **Using Application Coverage Report** section of this guide.

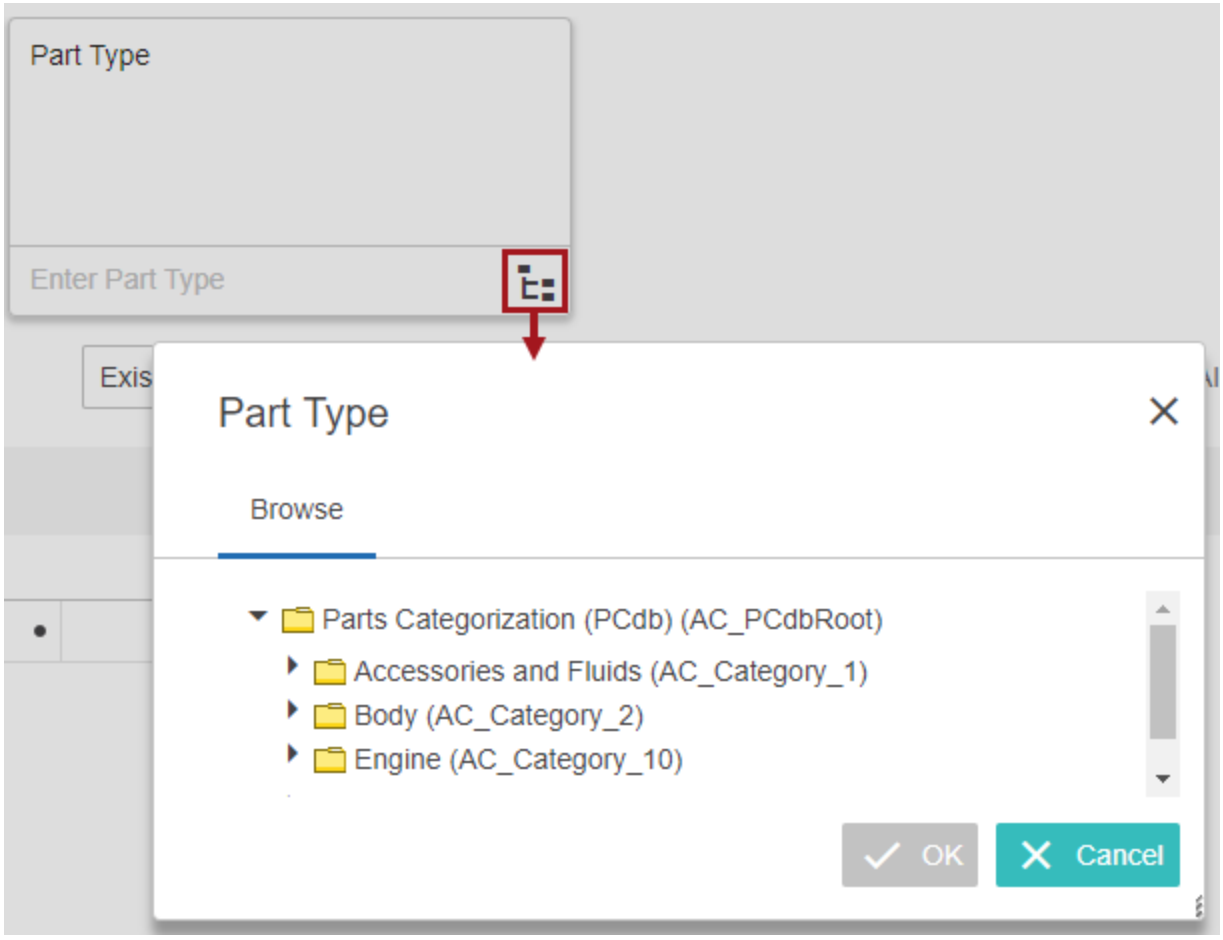
### Using Part Type Search Box

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, refer to 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.

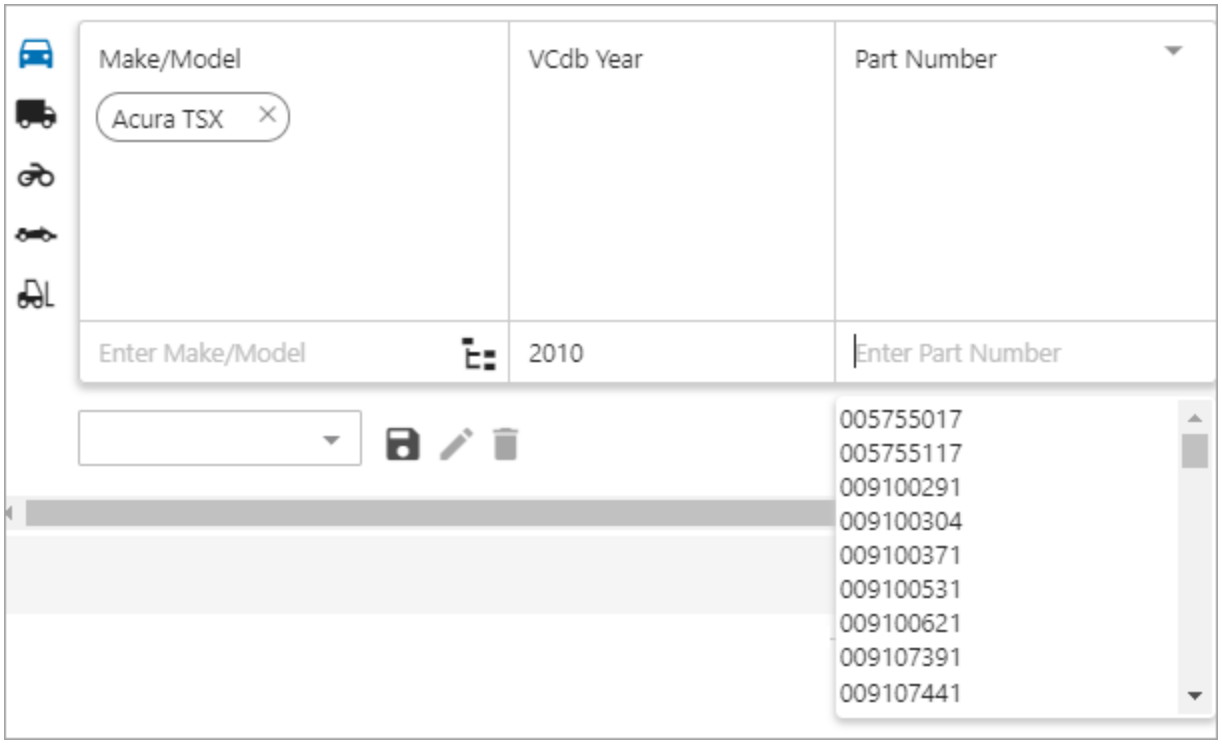
### Using Part Search Box

The Part ID, Name, or attributes belonging to the part can be selected as search criteria within an Application Manager. This section addresses functionality that is specific to the Part Search Box. For more information on the general functionality found within the search boxes, refer to the **Using Search Boxes** topic within this guide.

The Part ID / Name / attribute option dropdown is located in the top right corner of the Part Search Box. Clicking the down arrow to the right of the Part Search Box label, or clicking the label itself will display the available options within the dropdown. In the example below, a Part Search Box have been configured that displays the option 'Part Number' by default. Clicking the search box label displays a dropdown of available options (as shown below).

A Search Panel can have only one Part Search Box.

The Part Search Box typeahead field results are dependent upon selections made in the other search boxes. In the example below, the Acura TSX is added to the Make/Model search box, and the 2010 year is added to the Attribute search box, allowing the Part Search Box to only display options related to the 2010 TSX.




**Important:** The Part Search Box will only work when finding existing applications. An error dialog will be displayed if you have only a Part Number selected and try to select Existing and Missing Applications, or Missing Applications Only.

## 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.

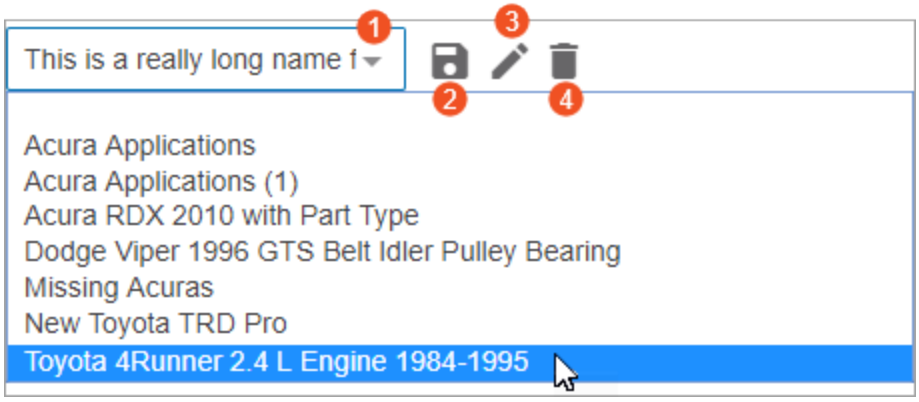
Once one or more search criteria selections are made, users can save the selections with a custom name. Using previously saved searches can be helpful when the same search criteria is needed repeatedly, or when using a basic set of saved search criteria aids users in more quickly building complex searches.

Saved searches are restricted by User login, Web UI, Vehicle Type Search Panel, and Screen ID. In other words, only the saved searches created with the login for the specific Web UI being used are displayed. 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.



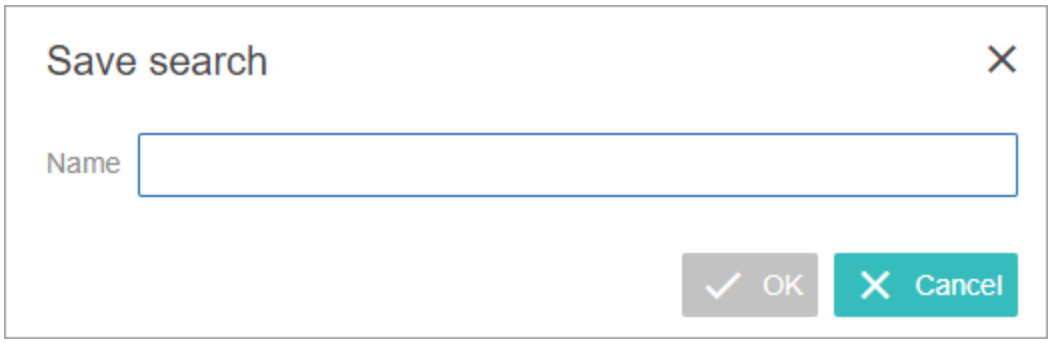
### Saving a Search

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.

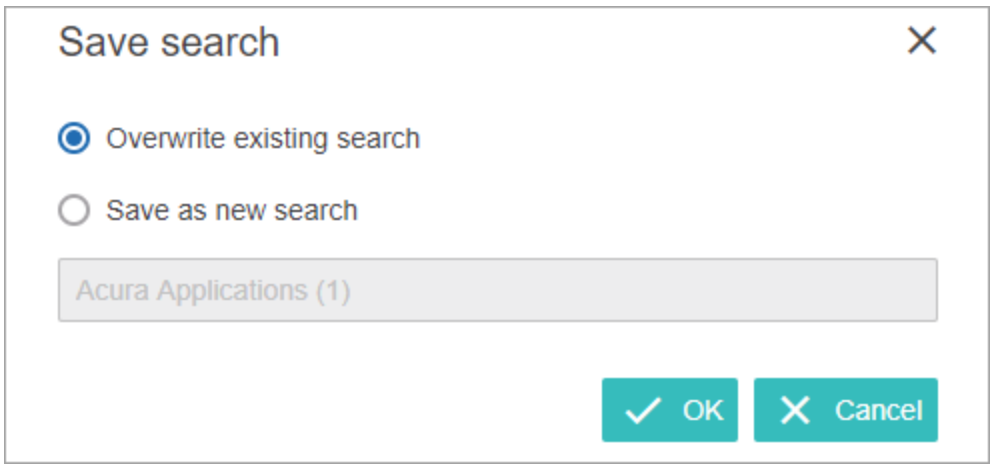
2. Within the Name parameter, type a name that best describes the saved search. The name can be created using any alphanumeric combination, but should be easy to identify for future use in the saved search dropdown.
3. Once at least one character is entered into the Name parameter, the OK button is enabled. 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.




**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.

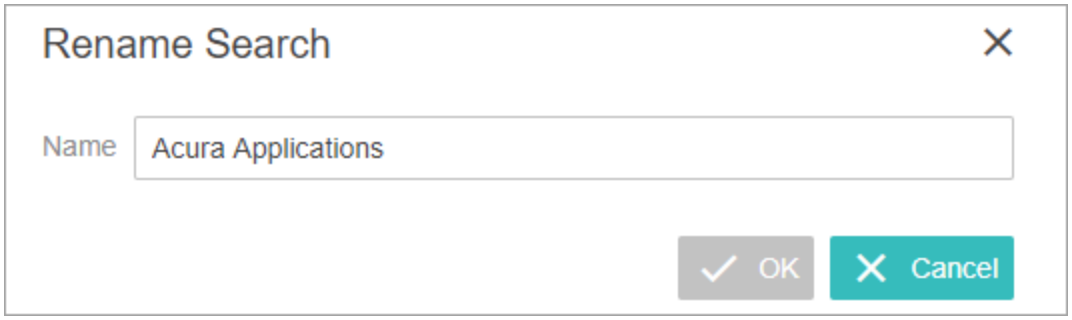
2. Once the desired option is selected and/or populated, click the OK button. The Save search dialog will close and the newly created saved search will display in the saved search dropdown.

**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.



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.

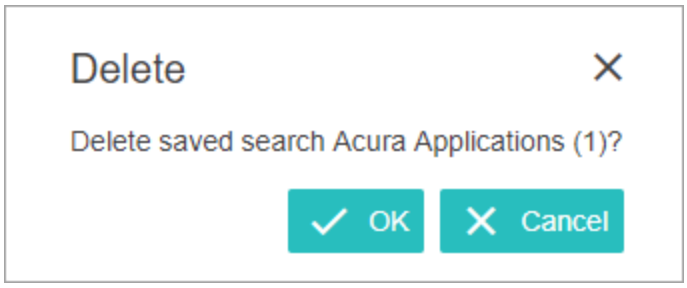
**Note:** The Cancel button can be clicked at any time. Doing so will close the dialog without saving any changes.

### 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).



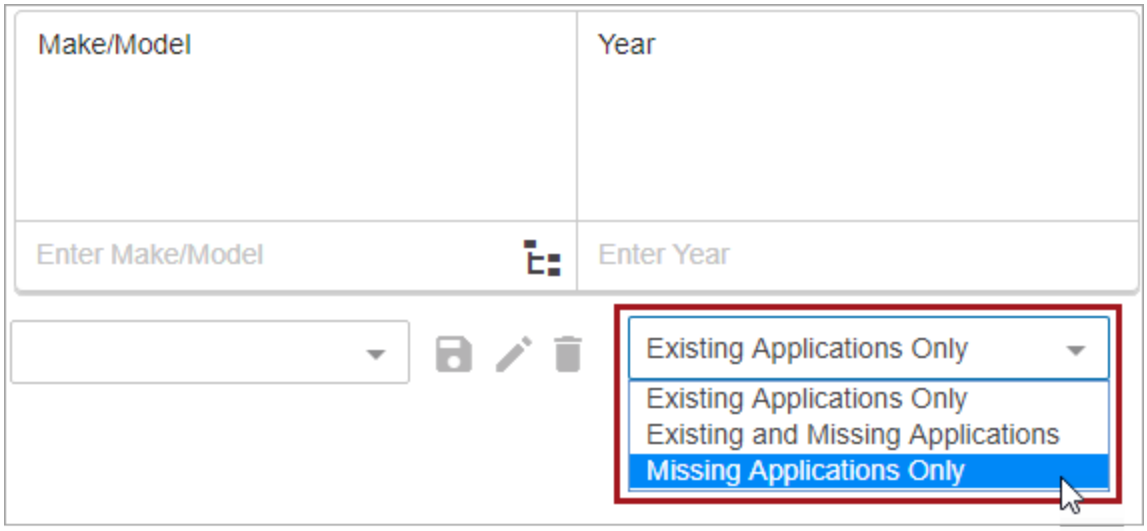

2. Click the OK button, the dialog will close, and the saved search will no longer display in the dropdown list.

**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




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, refer to 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, refer to 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, refer to 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 applicate 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, refer to 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 the they are not handled: Application comments (AC Comment), ACES Application Qualifiers, Comments, NAPA Notes, and Application Asset References.

### Running an Application Coverage Report

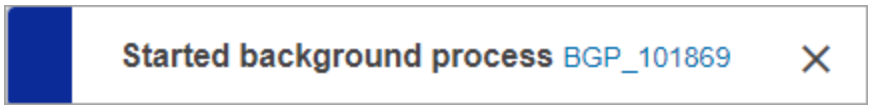
Users can choose to export existing applications and/or missing applications by selecting one of the following options in the dropdown to the left of the Report button:

- Existing Applications Only
- Existing and Missing Applications
- Missing Applications Only

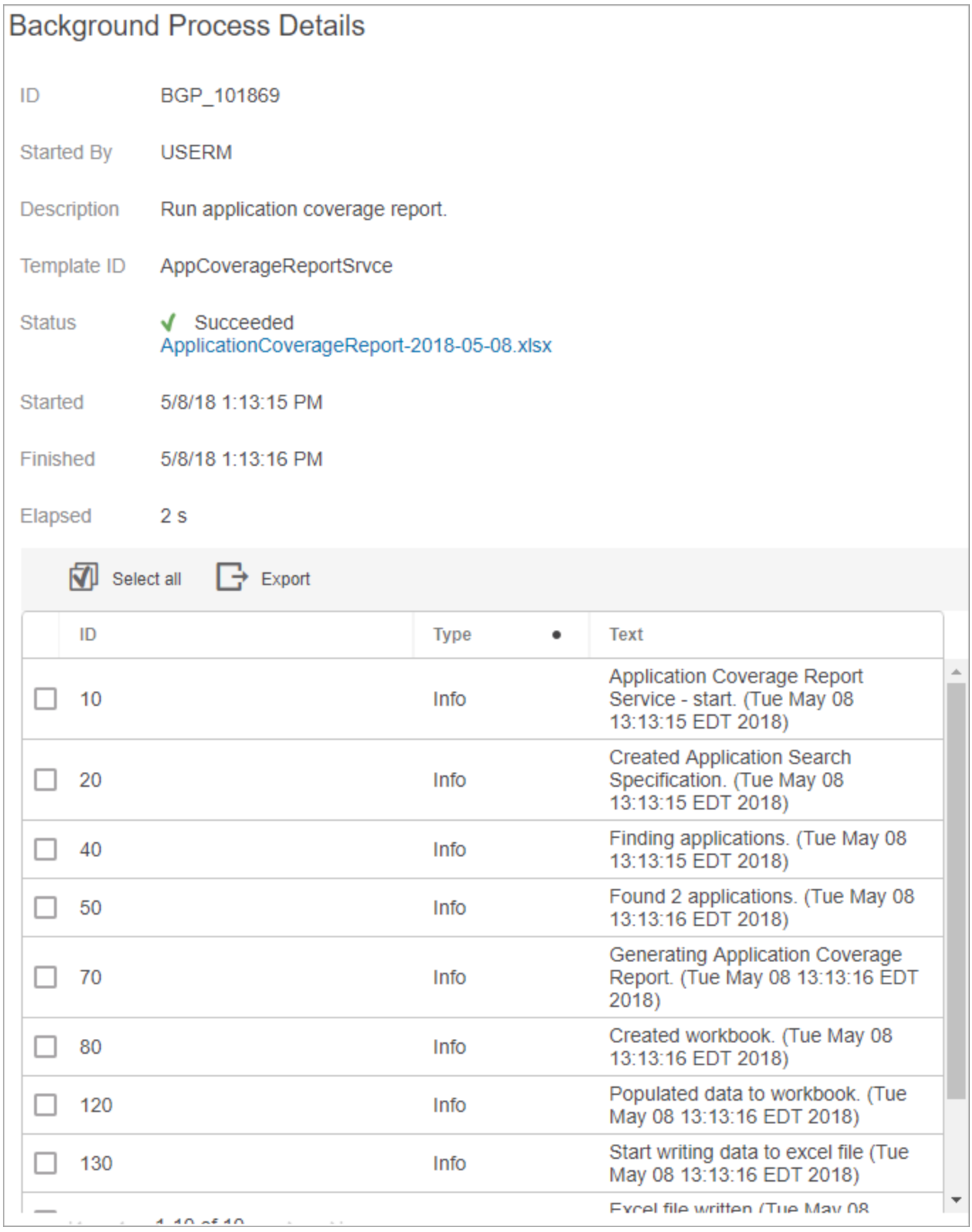
When the 'Existing and Missing Applications' or 'Missing Applications Only' options are selected, STEP performs specific actions known as missing application coverage functionality. For more information about missing applications, refer to the **Missing Application Coverage Functionality** topic within 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.'

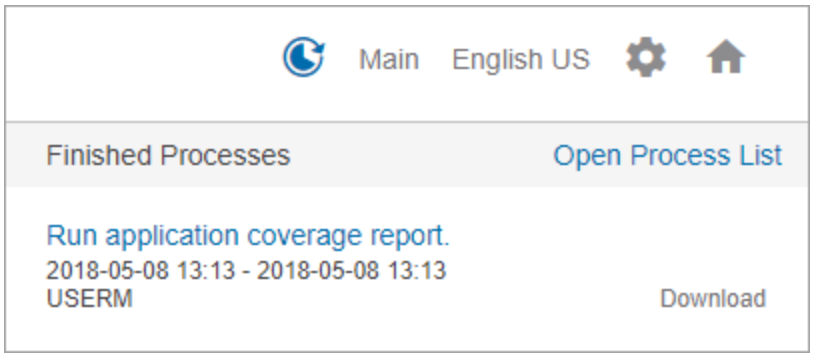



Within the notification, clicking the 'X' will close the dialog, whereas clicking the ID link will close the dialog and display the Background Process Details (as shown below).


Background Process Details
ID
BGP_101869
Started By
USERM
Description
Run application coverage report.
Template ID
AppCoverageReportSrvce
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
10
Info
Application Coverage Report Service - start. (Tue May 08 13:13:15 EDT 2018)
20
Info
Created Application Search Specification. (Tue May 08 13:13:15 EDT 2018)
40
Info
Finding applications. (Tue May 08 13:13:15 EDT 2018)
50
Info
Found 2 applications. (Tue May 08 13:13:16 EDT 2018)
70
Info
Generating Application Coverage Report. (Tue May 08 13:13:16 EDT 2018)
80
Info
Created workbook. (Tue May 08 13:13:16 EDT 2018)
120
Info
Populated data to workbook. (Tue May 08 13:13:16 EDT 2018)
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.'

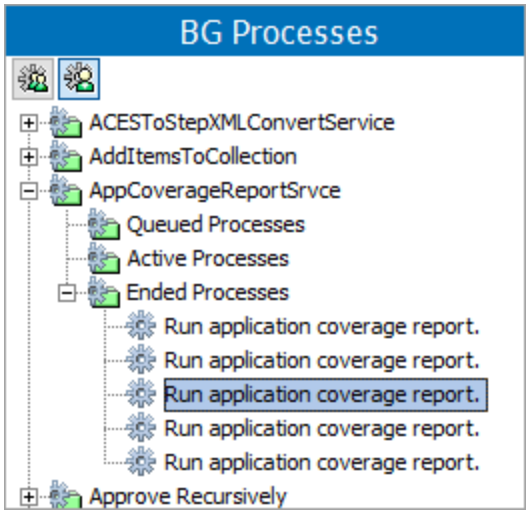
Users can easily track the progress of the report in the 'Recent background processes' side panel. In the example below, the 'Run application coverage report' link displays within the 'Finished Processes' of the side panel.




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 'Recent background processes' side panel, refer to the **Recent Background Processes Side Panel** topic within **STEP Online Help**.

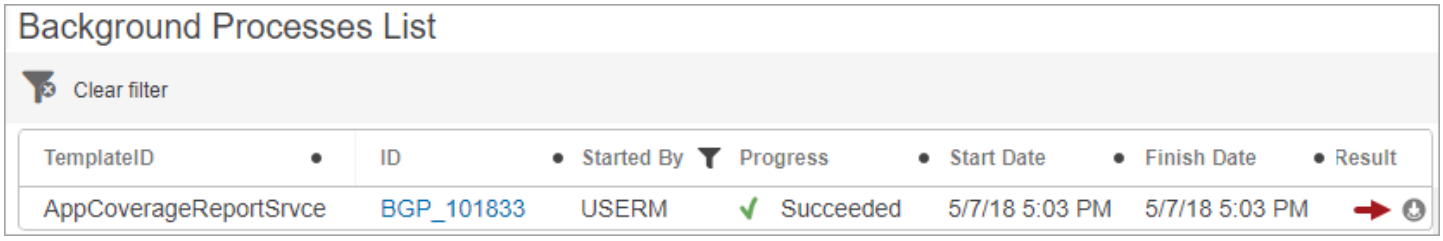
Additionally, the report can be downloaded from the workbench. To download the report from the workbench, go to the BG Processes tab > AppCoverageReprtSrvce > Ended Processes. Locate the desired process and click the save icon in the lower right corner of the workbench.

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

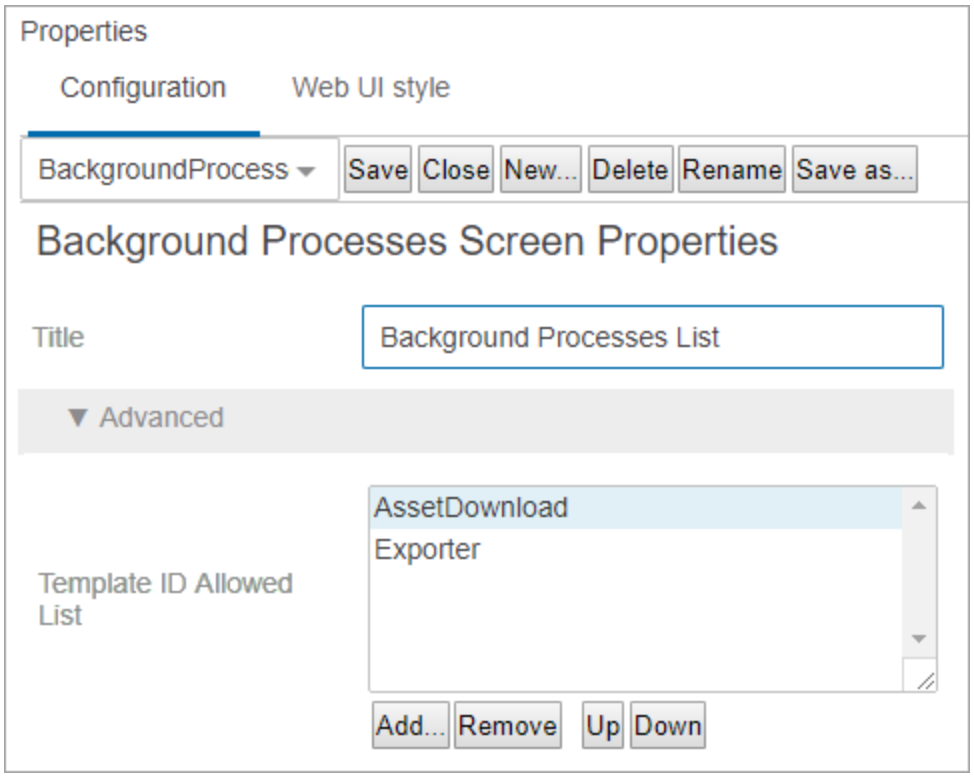
Optionally, the progress of a download for the Application Coverage Report can be configured to display within a Background Processes Screen. When configured, users can navigate to a Background Processes screen to view the following: TemplateID, ID, Started By, Progress, Start Date (and time), Finish Date (and time), Results download (as shown below).



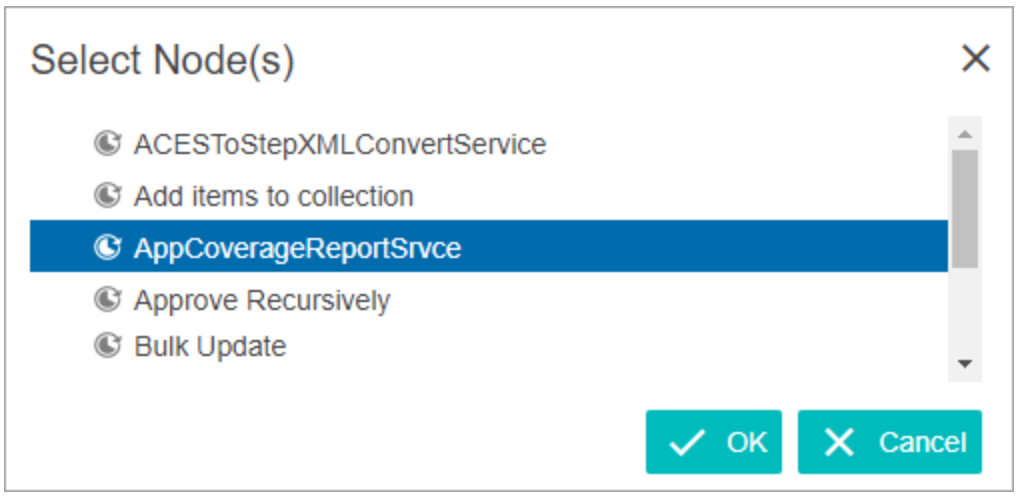

From this table, users can optionally click on the BGP ID and the Background Process Details screen will display, or click the Results download to download the report.

Below are the steps to add the Application Coverage Report to a Background Processes Screen.

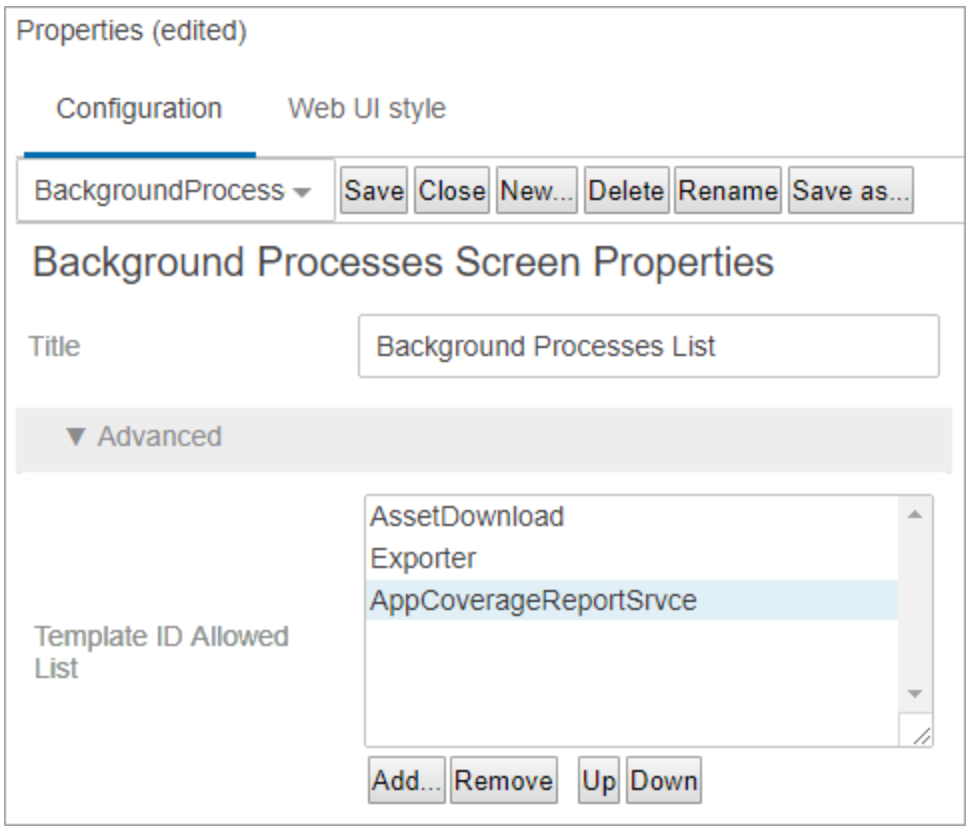
1. Within a Web UI, go to a Background Processes Screen, access the Designer, and the Background Processes Screen Properties will display as shown below.




2. Click the **Add** button below the 'Template ID Allowed List' parameter, and the Select Nodes list will display.
3. Select the **AppCoverageReportSrvce** node, as shown below.




4. Click the **OK** button, the dialog will close, and the 'Template ID Allowed List' parameter will display with the newly added AppCoverageReportSrvce (as shown below).




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

For more information about the Background Process List Screen, refer to the **Background Process List Screen** topic within **STEP Online Help**.

## Confusing Counterman Solution

Multiple application records may appear in the search result for defined search criteria in the Application Manager Screen. While some application records contain clear information, others may need more clarifying data. The application record must contain enough information to say whether the part is designed to fit a vehicle at all generic conditions or specific options / conditions.

The counterman must ensure that the part fits the vehicle and meets the customer's requirements. As the fitment data is stored as an application record (linkages), the confusing application records must be addressed correctly. Thus, a solution to address confusing application records is called the 'Confusing Counterman Solution.'

This solution is designed to improve the process of selling the correct part to the customer in the retail store. When a customer wants to buy a part for a vehicle, the counterman may have to ask questions about the vehicle to ensure that the right part is being sold. The questions are guided by the data populated in the application records. If multiple identical application records exist that would confuse the counterman, such application records are considered unclear / confusing application records.

For example, if an application record mentions that the part would fit the vehicle belonging to a specific SubModel, then the counterman must know the SubModel of the customer's car.

The implementation of this solution will:

- Limit the questions that a counterman would ask. For example, if a customer's car model only exists with one engine, the counterman is clear about this and need not have to ask what engine is in the customer's car.
- Clarify the counterman about the type of questions to ask the customer.
- Ensure that the counterman should have one precise result to suggest to the customer. The clear difference between the multiple results is displayed if there are multiple results.

The Confusing Counterman solution can generate a report of ambiguous application records when configured within the Application Manager screen. The report is called as confusing counterman report. The objective of generating the report is to identify the confusing application records and rectify the ambiguity in the application records.

This document includes the following related to the Confusing Counterman solution:

- The process of configuring the Confusing Counterman component in the Application Manager screen
- Using Confusing Counterman solution in the Application Manager Screen
- And some example use cases where the solution comes in handy in identifying the ambiguous application records.

## Adding and Configuring Confusing Counterman Report

The Confusing Counterman Report component can be added to the Application Manager Screen to generate a report of ambiguous application records that would confuse a counterman.

### Prerequisites

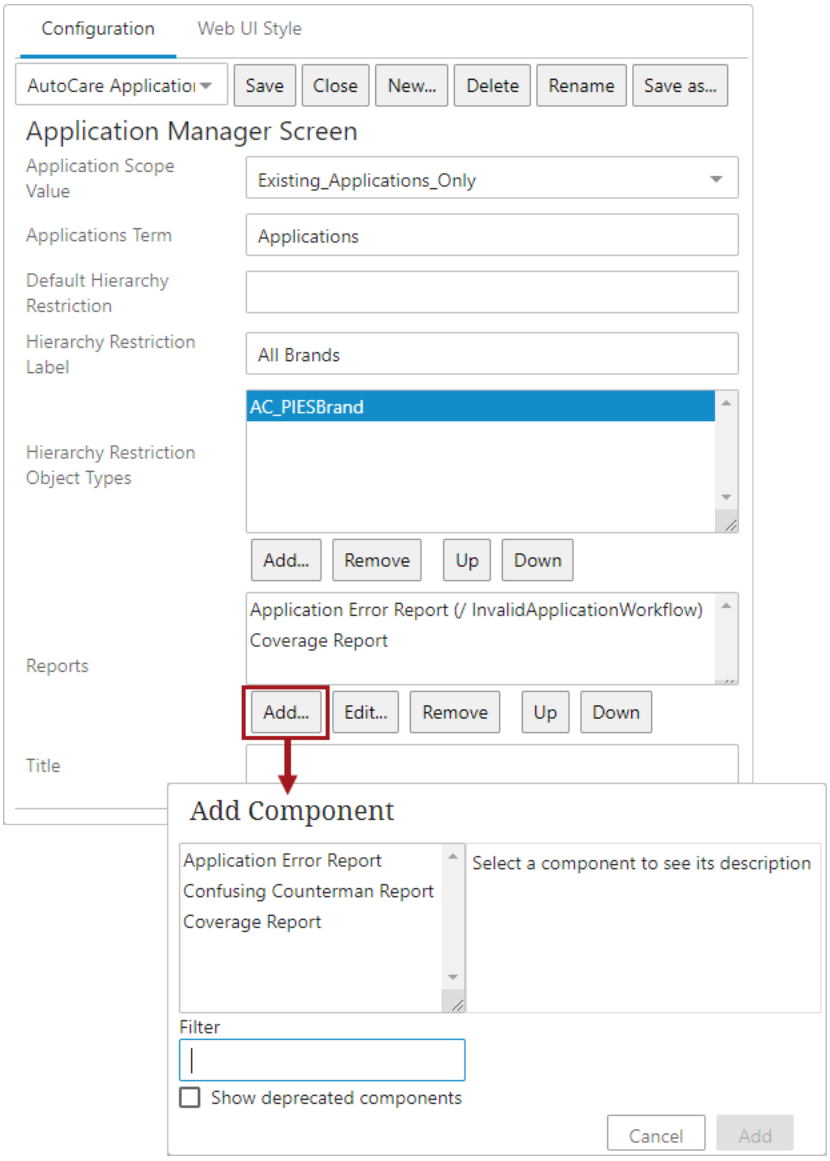
Since basic concepts for working with the designer are not covered in this section, a user configuring the Confusing Counterman Report component within a Web UI is expected to be familiar with the Web UI Designer. In addition, the user must have appropriate privileges to access the designer. For more information, refer to the **Designer Access** topic within the **Web User Interfaces** section of the **STEP Online Help**.

The user must have appropriate knowledge about the functionalities of the Application Manager Screen. For more information about the Application Manager Screen, refer to the **Application Manager** topic

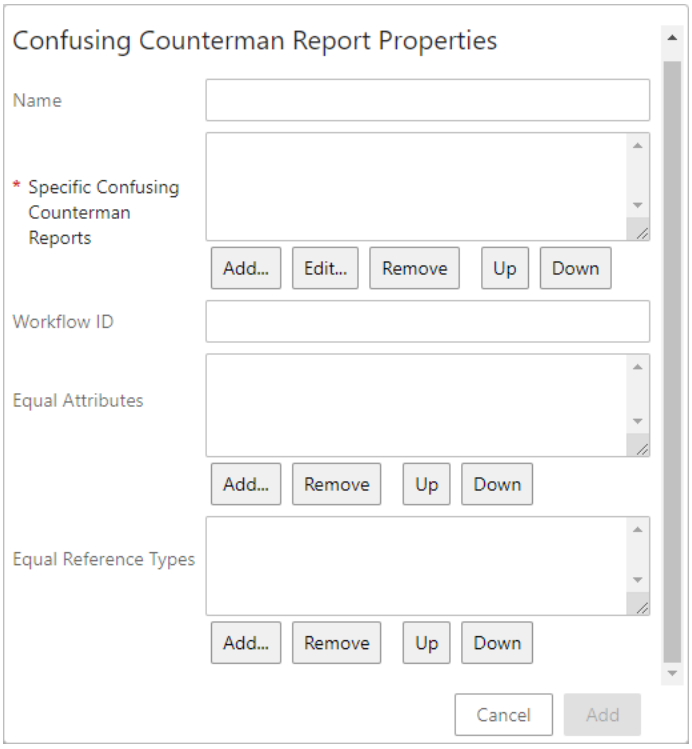
### Adding / Configuring Confusing Counterman Report Component

Steps for adding and configuring the Confusing Counterman Report Component for an Application Manager screen are below.

1. Access the Web UI Designer, where the Application Manager can be configured.
2. Click the **Add** button next to the Reports parameter, and the Add Component dialog will display.




3. Select the Confusing Counterman Report component, click the **Add** button, and the Confusing Counterman Report Properties dialog will display as shown below.




Within the Confusing Counterman Report Properties dialog, populate the following parameters:

- **Name:** Enter a text that needs to be displayed as a radio button when the user clicks the **Create Report** button in the Application Manager.
- **Specific Confusing Counterman Reports:** Within this parameter, the user can add one or more report components by clicking the **Add** button. It is mandatory to have at least one report component so that when clicking the **Create Report** button in the Application Manager, a report of confusing application records is generated. The report components within this parameter further define the nature of the report to be generated. Each report component will have a criterion defined based on the type of report generated. The following report components are available to configure:
    - **Multiple Parts For Same Vehicle Report:** This component helps in generating a report where multiple identical application records belonging to different parts are referenced to the same vehicle. For more information, refer to the **Multiple Parts For Same Vehicle Report Component** topic.
    - **Overlapping Applications Report:** This component helps in generating a report where multiple application records belonging to a part are referenced to the same vehicle. The report is generated if application records exist with unclear conditions / options defined in them.
- **Workflow ID:** Enter the workflow ID to which the confusing application records are to be initiated. The workflow should be able to handle confusing applications. The workflow defined in this parameter should be valid for the object type of the application record. The Invalid Application Workflow (ID = InvalidApplicationWorkflow) is entered for the example above.

- **Equal Attributes:** Add attributes valid to the application records that are to be exempted from being flagged as confusing if an ambiguity exists. Attributes added in this parameter will not be considered for comparison while analyzing the application records.
- **Equal Reference Types:** Add reference types valid to the application records that are to be exempted from being flagged as confusing if an ambiguity exists. Reference types added in this parameter will not be considered for comparison while analyzing the application records.

### Using Confusing Counterman Solution

The Confusing Counterman solution helps users generate a report of ambiguous application records in an Excel spreadsheet (XLSX file type). Since the Confusing Counterman Report is built to work with the Application Manager's intelligent search interface, it is extremely configurable, and provides users with on-demand flexibility to choose the report criteria that best meet their needs.

Before a Confusing Counterman 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 **Create Report** button allows users to export the Confusing Counterman Report.

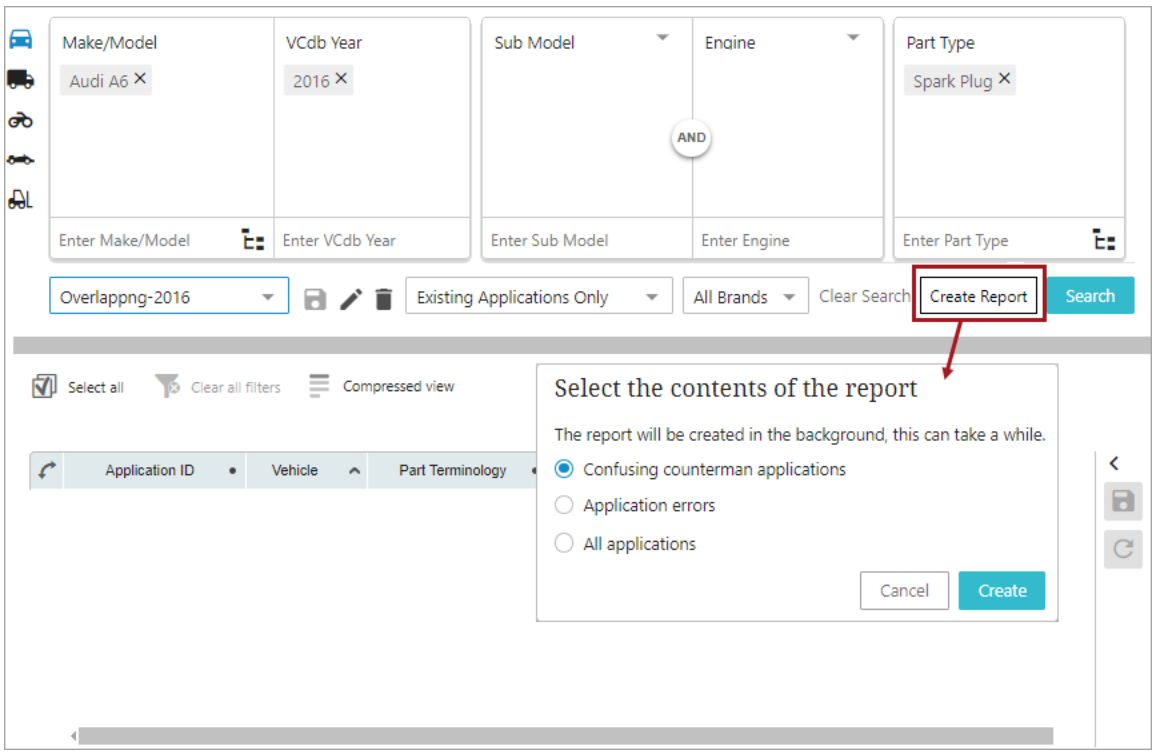
## Prerequisites

Before attempting to understand how the Confusing Counterman 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.

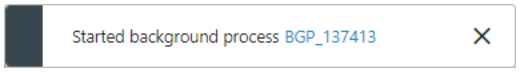
Configuration is required before the Confusing Counterman solution can be used. For more information, refer to the **Prerequisites** section of the **Adding and Configuring Confusing Counterman Report** topic.

#### Running a Confusing Counterman Report

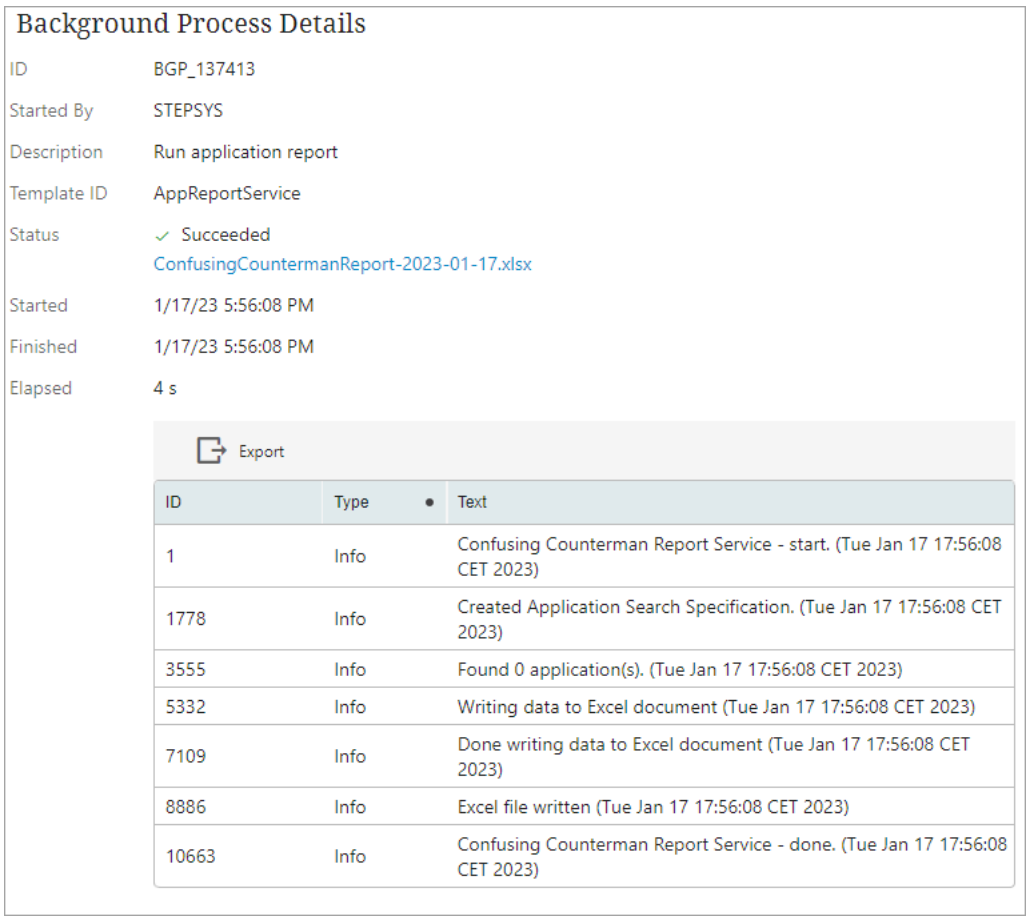
After at least one criterion has been added to a search box, the **Create Report** button will enable. Once enabled, the **Create Report** button can be clicked before or after clicking the Search button. Depending on the configuration, the **Create Report** action allows users to generate multiple types of reports. Clicking the **Create Report** action button displays the 'Select the contents of the report' dialog.




Selecting the 'Confusing counterman applications' option and clicking the **Create** button initiates a background process. A notification displays informing users of the linked background process ID, and that the process has started. The example below displays the background process notification with the ID link as 'BGP_137413.'

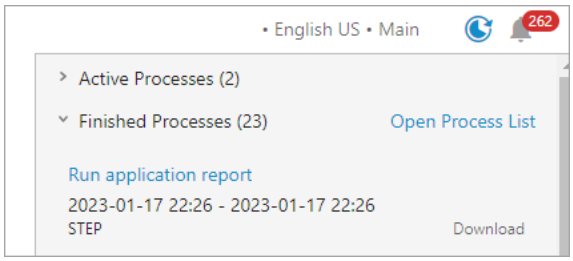



Within the notification, clicking the 'X' will close the dialog, whereas clicking the ID link will close the dialog and display the 'Background Process Details' (as shown below).




Once the Background Process Details screen Status parameter displays as 'Succeeded,' the download link for the Confusing Counterman Report will display. In the example above, the link displays as 'ConfusingCountermanReport-2023-01-17.xlsx.'

Users can easily track the progress of the report in the 'Recent background processes' side panel. In the example below, the 'Run application 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 report'), and the Background Process Details screen will display. Users can also click the 'Download' link to download the .XLSX Confusing Counterman Report. For more information on the 'Recent background processes' side panel, refer to the **Recent Background Processes Side Panel** topic within **STEP Online Help**.

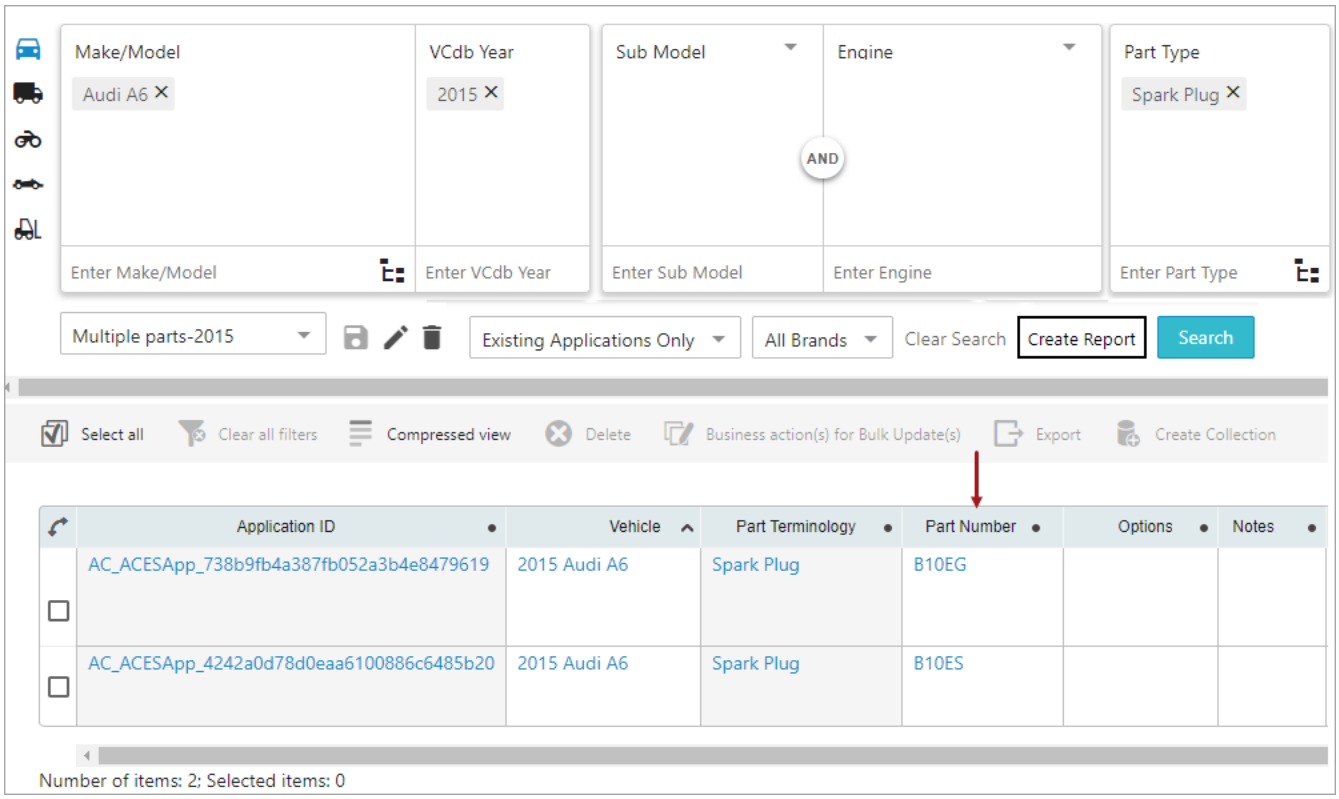
### Multiple Parts For Same Vehicle Report Component

The Multiple Parts For Same Vehicle Report component has defined criteria that generate a report of identical application records belonging to different parts referenced to the same vehicle. In addition, it also considers reporting the ambiguity persisting to options / conditions stored as attributes / reference types in the application records.

For ease of understanding, this document takes an example of two ACES application records, AC_ACESApp_a2a76b390df08b7bbdc2775f73419 and AC_ACESApp_3af9120b34ec36f3fc528c6d49420. These application records are displayed in the Application Manager Screen search result table based on the search performed, as shown in the screenshots below. The behavior of these application records under various use cases and how it could cause ambiguity is explained in detail below:

#### *Use case one*

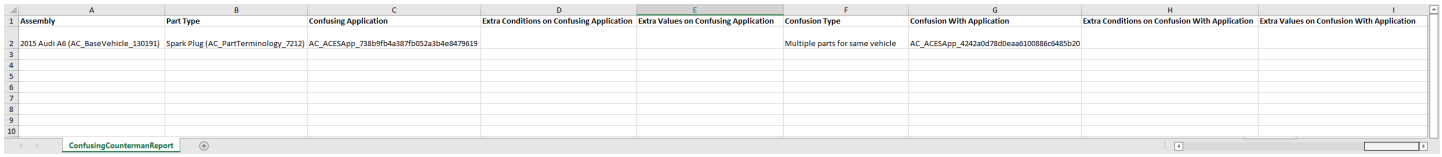
This simple use case displays two application records with identical values except for the part number.




The report will be generated as follows, and the same is shown in the screenshot below:

- The 'Confusion Type' column displays the reason for ambiguity which, in this case, is flagged by the Multiple Parts For Same Vehicle Report component.
- The 'Extra Conditions on Confusing Application' and 'Extra Conditions on Confusion With Application' columns are empty, as there are no options / conditions in the application records that would cause confusion.

- The 'Extra Values on Confusing Application' and 'Extra Values on Confusion With Application' columns are also empty, as no values are populated in the options / conditions.

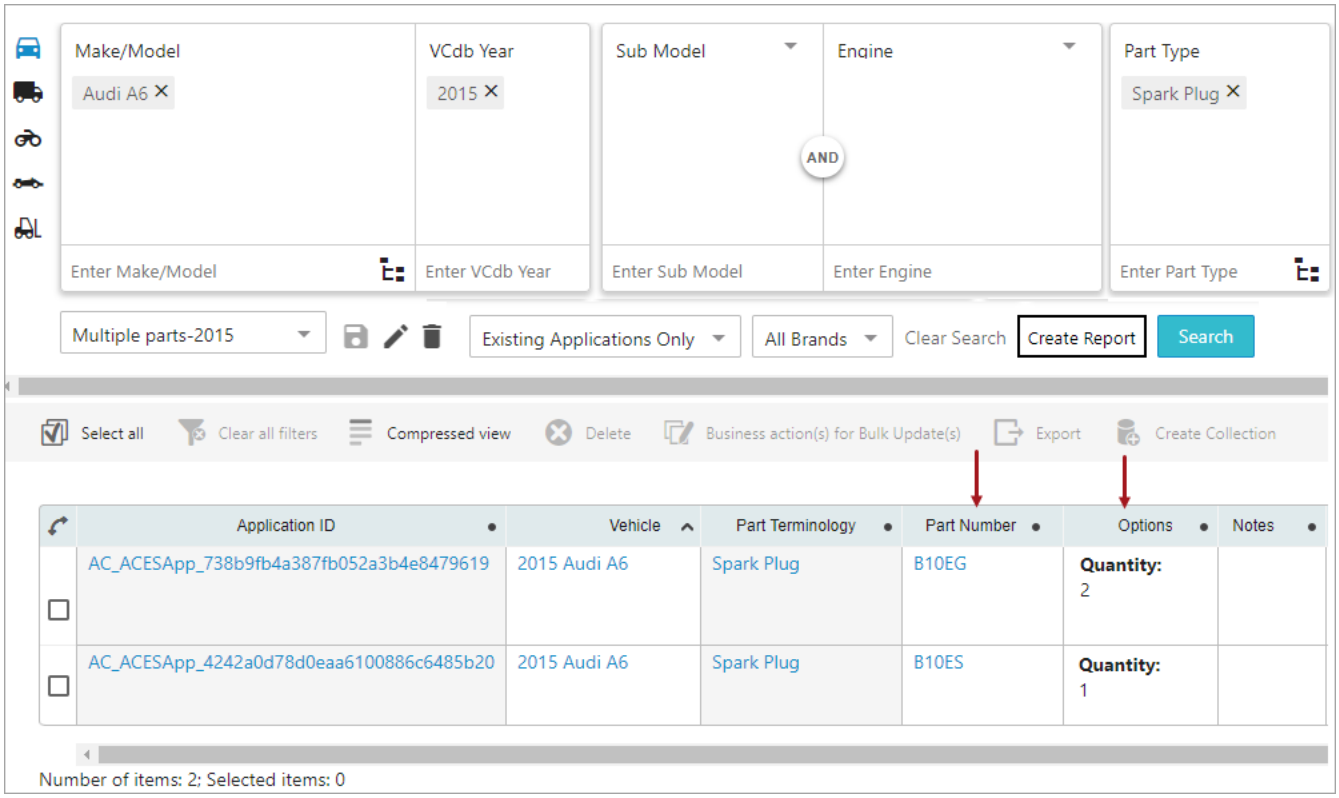


| Assembly | Part Type | Confusing Application | Extra Conditions on Confusing Application | Extra Values on Confusing Application | Confusion Type | Confusion With Application | Extra Conditions on Confusion With Application | Extra Values on Confusion With Application |
|---|---|---|---|---|---|---|---|---|
| 2015 Audi A6 (AC_BaseVehicle_130191) | Spark Plug (AC_PartTerminology_7212) | AC_ACESApp_738b9fb4a387fb052a3b4e8479619 | | | Multiple parts for same vehicle | AC_ACESApp_4242a0d78d0eaa6100886c6485b20 | | |

ConfusingCountermanReport

### *Use case two*

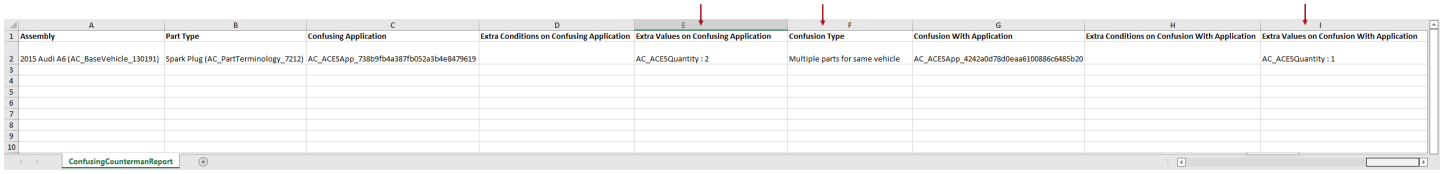
In this use case, two application records belong to different part numbers, which include options / conditions populated, as shown below. The options / condition attribute Quantity (ID = AC_ACESQuantity) has a unique value populated in both the application records.




The report will be generated as follows, and the same is shown in the screenshot below:

- The 'Confusion Type' column displays the reason for ambiguity which, in this case, is flagged by the Multiple Parts For Same Vehicle Report component.
- The 'Extra Conditions on Confusing Application' and 'Extra Conditions on Confusion With Application' columns are empty as there are no options / conditions in the application records that would cause confusion.

- The 'Extra Values on Confusing Application' and 'Extra Values on Confusion With Application' columns display options / conditions that have unique values (attributes values) populated in both the application records. In this example, the attribute Quantity (ID = AC_ACESQuantity) has been populated with value '2' in AC_ACESApp_a2a76b390df08b7bbdc2775f73419 application record and value '1' in AC_ACESApp_3af9120b34ec36f3fc528c6d49420 application record.

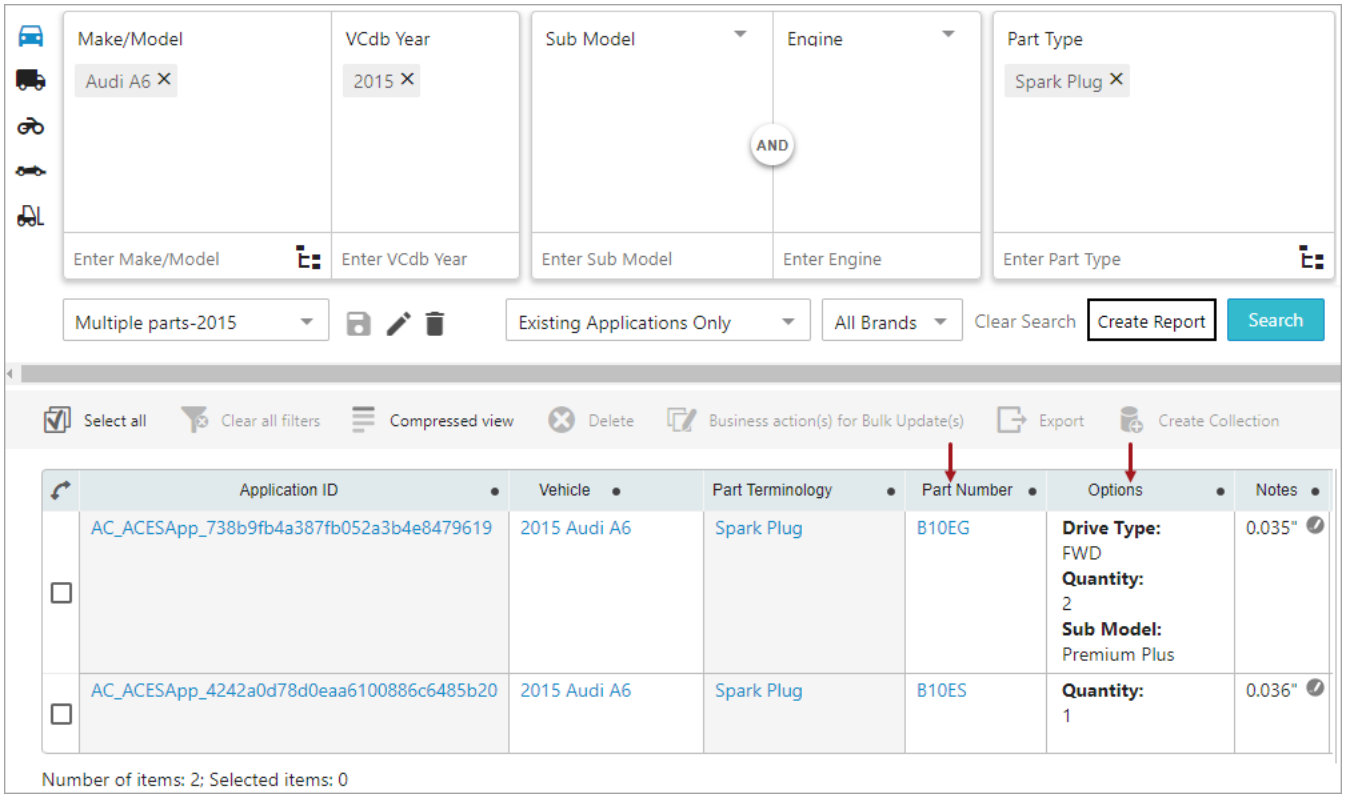


| Assembly | Part Type | Confusing Application | Extra Conditions on Confusing Application | Extra Values on Confusing Application | Confusion Type | Confusion With Application | Extra Conditions on Confusion With Application | Extra Values on Confusion With Application |
|---|---|---|---|---|---|---|---|---|
| 2015 Audi A6 (AC_BaseVehicle_130191) | Spark Plug (AC_PartTerminology_7212) | AC_ACESApp_738b9fb4a387fb052a3b4e8479619 | | AC_ACESQuantity : 2 | Multiple parts for same vehicle | AC_ACESApp_4242a0d78d0eaa6100886c6485b20 | | AC_ACESQuantity : 1 |

ConfusingCountermanReport

### *Use case three*

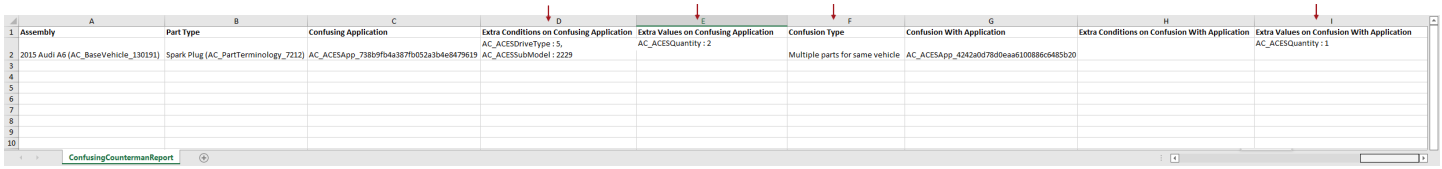
In this use case, two application records belong to different part numbers, which include options / conditions defined to it as shown below:




The report will be generated as follows, and the same is shown in the screenshot below:

- The 'Confusion Type' column displays the reason for ambiguity which, in this case, is flagged by the Multiple Parts For Same Vehicle Report component.

- The 'Extra Conditions on Confusing Application' and 'Extra Conditions on Confusion With Application' columns display options / conditions that lack clarity. This lack of transparency leads to further confusion in determining the correct application record. In the example above, the AC_ACESApp_a2a76b390df08b7bbdc2775f73419 application record has values populated in Drive Type (ID = AC_ACESDriveType) and Sub Model (ID = AC_ACESSubModel) condition attributes, and the AC_ACESApp_3af9120b34ec36f3fc528c6d49420 application record does not have any values defined in it. This lack of information in the second application record brings about much confusion.
- The 'Extra Values on Confusing Application' and 'Extra Values on Confusion With Application' columns display options / conditions that have unique values (attributes values) populated in both the application records. In this example, the attribute Quantity (ID = AC_ACESQuantity) has been populated with value '2' in AC_ACESApp_a2a76b390df08b7bbdc2775f73419 application record and value '1' in AC_ACESApp_3af9120b34ec36f3fc528c6d49420 application record.



| | A | B | C | D | E | F | G | H | I |
|---|---|---|---|---|---|---|---|---|---|
| 1 | Assembly | Part Type | Confusing Application | Extra Conditions on Confusing Application | Extra Values on Confusing Application | Confusion Type | Confusion With Application | Extra Conditions on Confusion With Application | Extra Values on Confusion With Application |
| 2 | 2015 Audi A6 (AC_BaseVehicle_130191) | Spark Plug (AC_PartTerminology_7212) | AC_ACESApp_738b9fb4a387fb052a3b4e8479619 | AC_ACESDriveType : 5,<br>AC_ACESSubModel : 2229 | AC_ACESQuantity : 2 | Multiple parts for same vehicle | AC_ACESApp_4242a0d78d0eaa6100886c6485b20 | | AC_ACESQuantity : 1 |
| 3 | | | | | | | | | |
| 4 | | | | | | | | | |
| 5 | | | | | | | | | |
| 6 | | | | | | | | | |
| 7 | | | | | | | | | |
| 8 | | | | | | | | | |
| 9 | | | | | | | | | |
| 10 | | | | | | | | | |

ConfusingCountermanReport

# Results Table and Toolbar

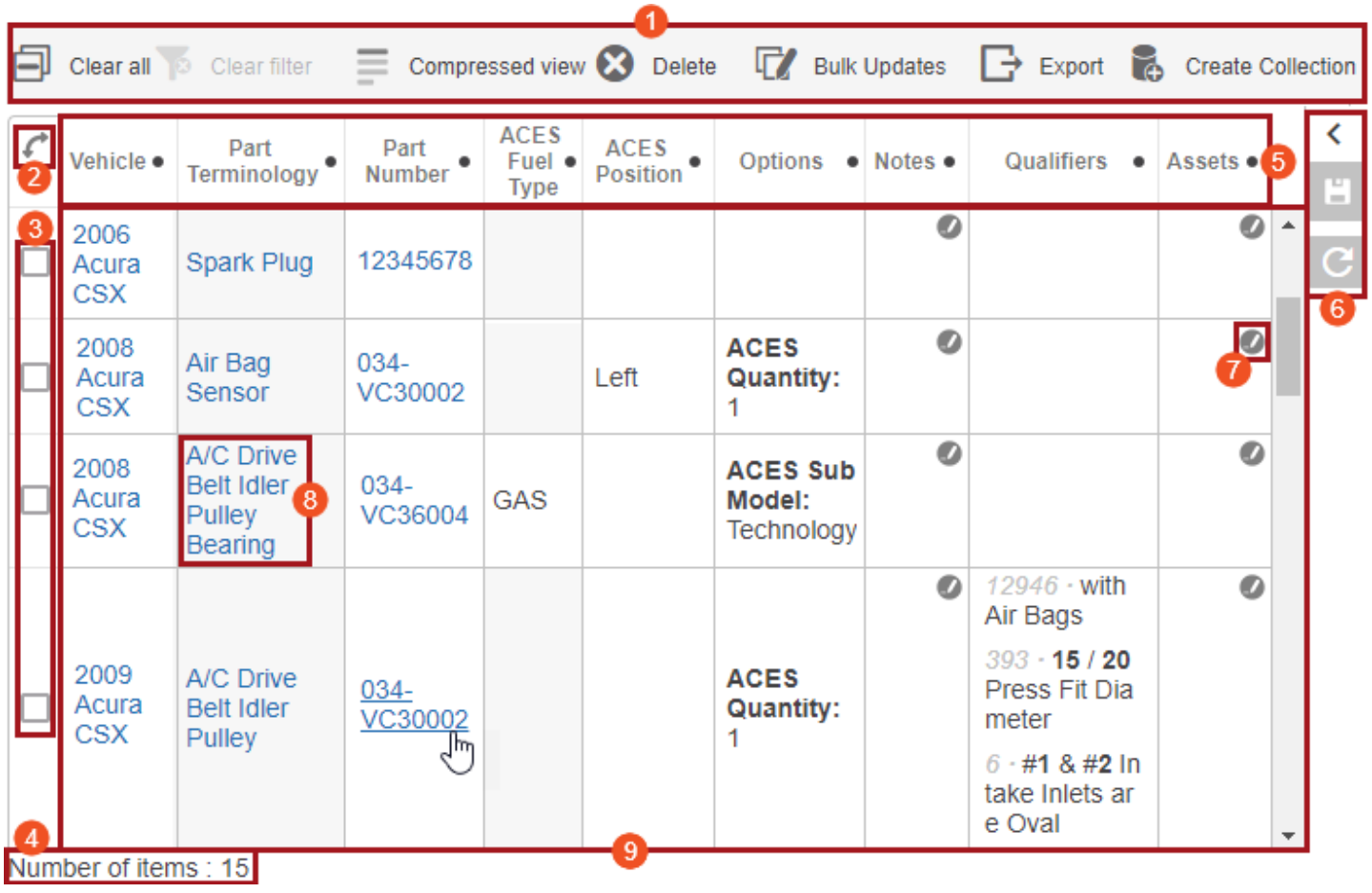
The bottom half of the Application Manager screen offers features related to the results table and its toolbar. The top half of the screen is dedicated to the intelligent search interface. For more information about the intelligent search interface, refer to the **Intelligent Search Interface** topic.

This section provides an overview of the results table and toolbar and addresses the following:

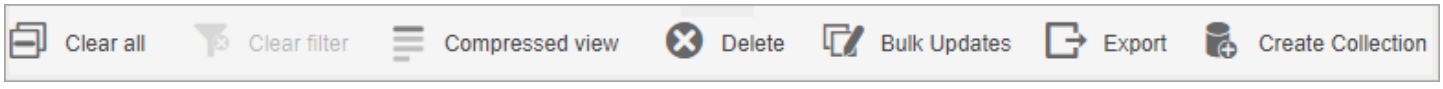
- Results Table Default Columns
- Controlling Display of Conditions in Application Manager
- Creating Applications

## Results Table and Toolbar Overview

The results table and toolbar of an Application Manager screen are composed of the following:




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, refer to 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, refer to 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 of every cells to be displayed as a single line of information in the Application Manager results table. This way it helps in accommodating more application records to be visible in the results table. 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, refer to 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, refer to the **Collections** topic within the **Getting Started Section** of the **STEP Online Help**.

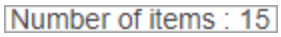
2. **Flip table icon**

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.

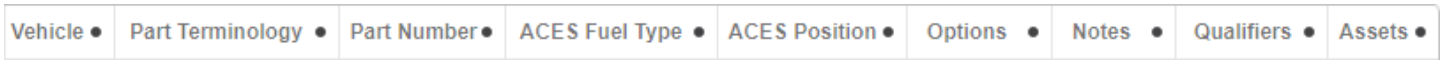
4. **Number of items**




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, refer to the **Application Coverage Report** section.

5. **Column Headers**




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.

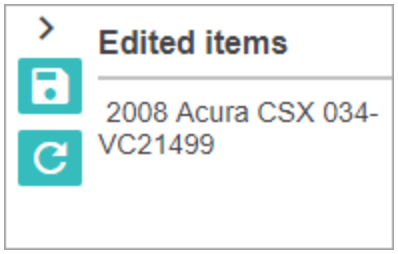
By default, the following components are configured within the Headers parameter for the Node List Properties Child Component of the Application Manager:

- 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, refer to 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, refer to 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 obvious 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**

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, refer to 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<br>*393* · **15 / 20** Press Fit Diameter<br>*6* · **#1 & #2** Intake 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, refer to 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, refer to 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 Table 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<br>*393* · **15 / 20** Press Fit Diameter<br>*6* · **#1 & #2** Intake 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.

### ACES Application Qualifier Table Header

Any free text information that is not covered by a vehicle attribute will be stored as individual qualifiers in the system. These qualifiers are imported through the Qdb file and are stored as classification objects in the system. The relation between the application records and the qualifiers is established using the 'ACES Application To Qualifier' Classification Reference Type (ID = AC_ACESApplicationToQualifier).

An application record can hold the relationship with multiple qualifiers to create a piece of compound information. Also, some qualifier texts will have placeholders (called parameters) that can be used to replace variable data found in a qualifier. This mechanism reduces the number of qualifiers stored in the system. For example, dates or serial numbers are often used to identify mid-year production changes ('From 2/20/98', 'To Chassis #92817820'). Rather than creating a new qualifier for every date or chassis number that might be needed, a template is used with a parameter in it. Instead of 'From 12/1/2002', for example, the entry would include just the static text ('From') and leave a placeholder for the actual date.

The ACES Application Qualifier table header component displays the AutoCare application qualifier within the AutoCare Application Manager Results Table and/or Application Editor Screen. The applicable qualifiers are listed in a column against each result. Further, users can double click on the cell to open up the Value Editor window and can add, delete, and override the qualifiers.

## Prerequisites

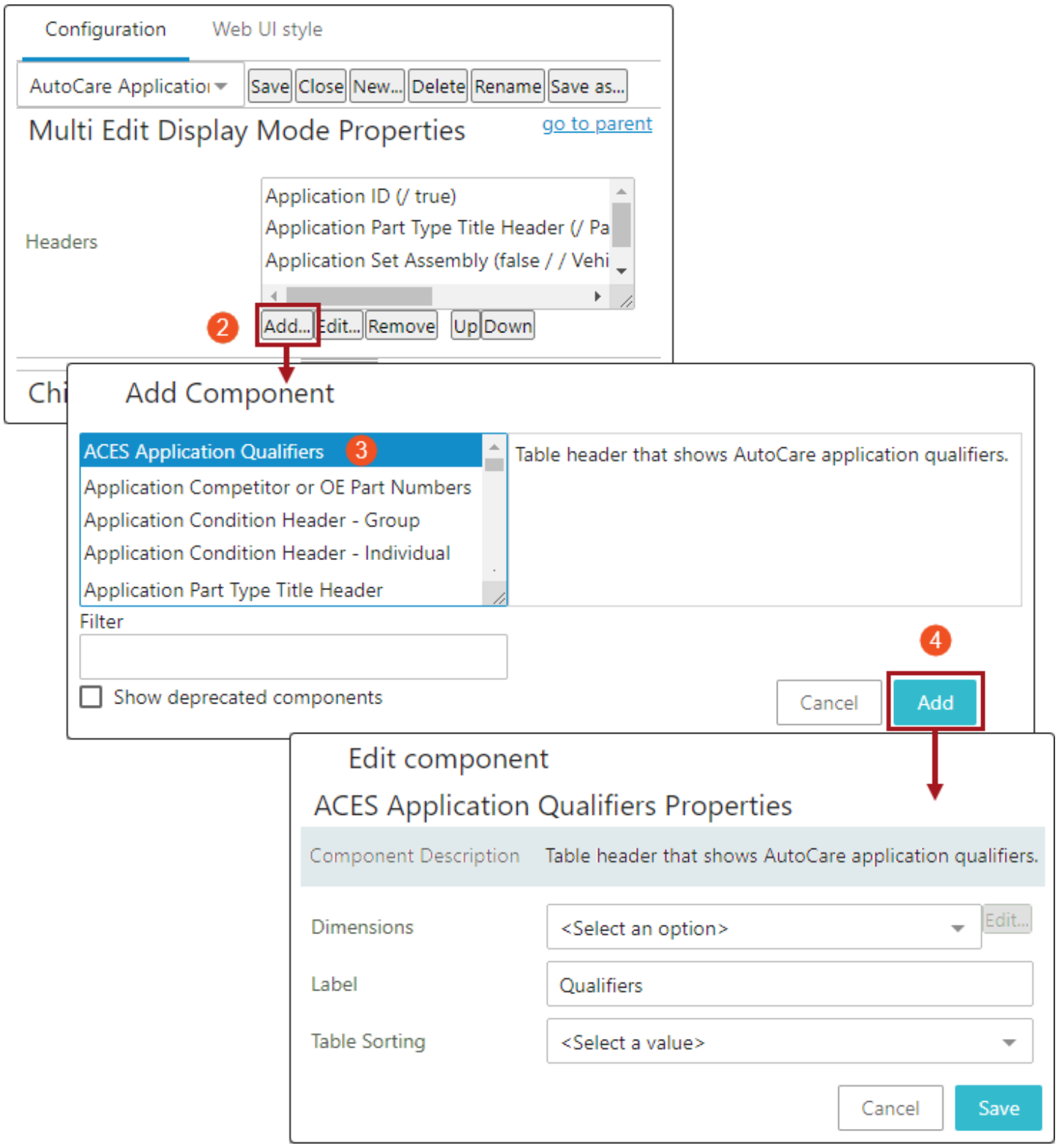
It is expected that anyone configuring the ACES Application Qualifier component is familiar with the Web UI Designer as basic concepts for working with the designer are not covered in this section. In addition, the user must have appropriate privileges to access the designer. Additional information can be found in the **Designer Access** section of the **Web User Interfaces** documentation.

For information about accessing and configuring the Node List component, refer to the **Node List Component** topic within the **Web User Interfaces** section of **STEP Online Help**.

## Configuration Steps

The ACES Application Qualifier component can be added within any of the display modes available in the Node list Properties for the selected Application Manager Screen (or Application Editor Screen). Below are the steps to configure the ACES Application Qualifier component.

1. Using the Web UI designer as shown in the image below, go to Node List Properties > Multi Edit Display Mode Headers parameter for the Application Manager Screen (or Application Editor Screen) that needs to display qualifiers.
2. Click the **Add** button for the Headers parameter, and the Add Component dialog will display.
3. Find and select the **ACES Application Qualifiers** component.
4. Click the **Add** button, and the ACES Application Qualifiers Properties dialog will display (as shown below).

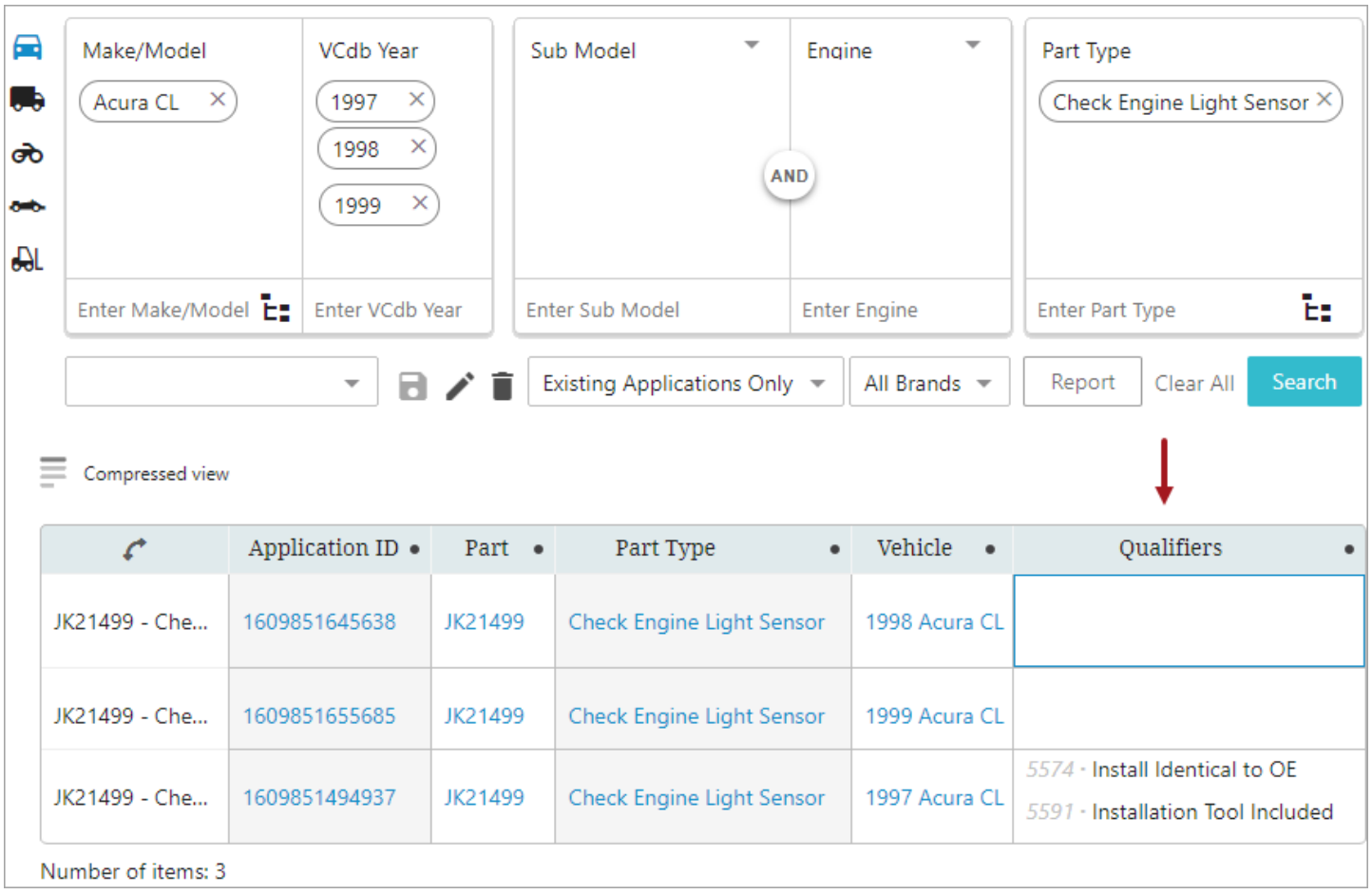



5. Optionally, populate the parameters listed below.

- **Dimensions:** By default, the parameter is blank. Optionally, to configure the column width and height, select 'Table Header Dimensions.'
- **Label:** By default, the parameter is populated with the value 'Qualifiers'. Optionally, enter the 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.

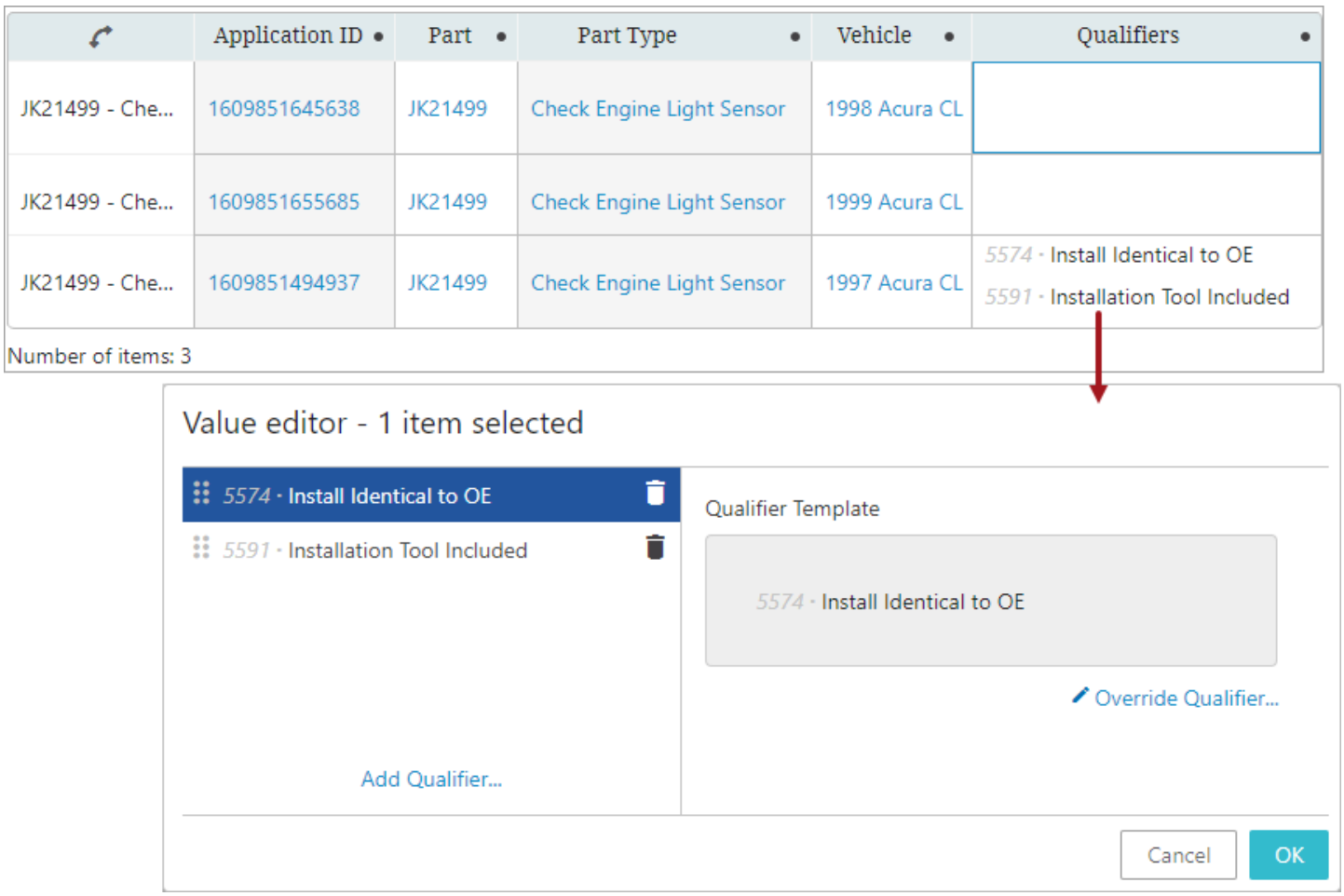
6. Click the **Save** button to save the newly added component to the Headers parameter.
7. Optionally, on the Node List Properties Headers parameter, use the **Up** and/or **Down** buttons to configure the order in which the newly added column should display within the results table.
8. Click the **Save** and **Close** buttons for the designer.

When a search is performed in the Application Manager Screen, the Application Manager results table will display the application records with the qualifiers (if any) of each application as shown below.



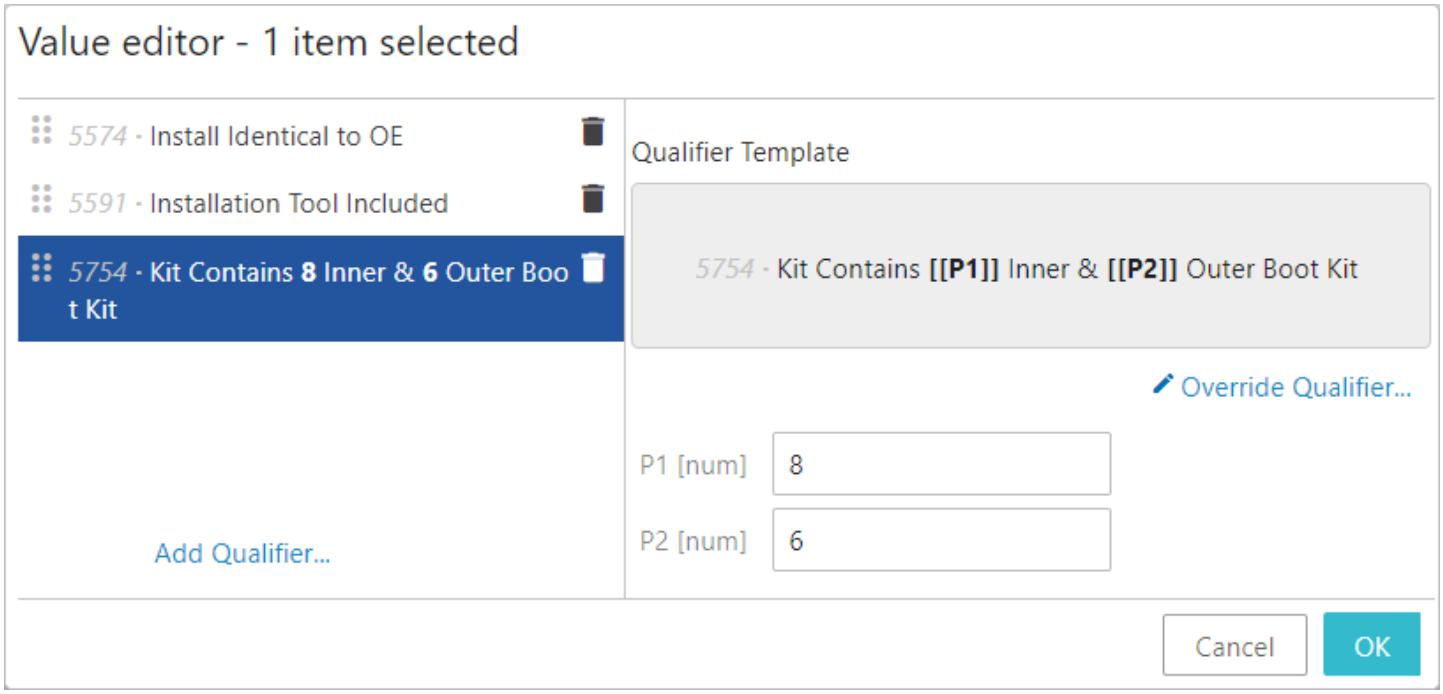

### Using Application Qualifier Value Editor to Add or Remove References to Qualifiers

To add or remove references to a qualifier from an application record via the 'ACES Application Qualifier' component, double click on the cell. A Value editor dialog will display as shown below.

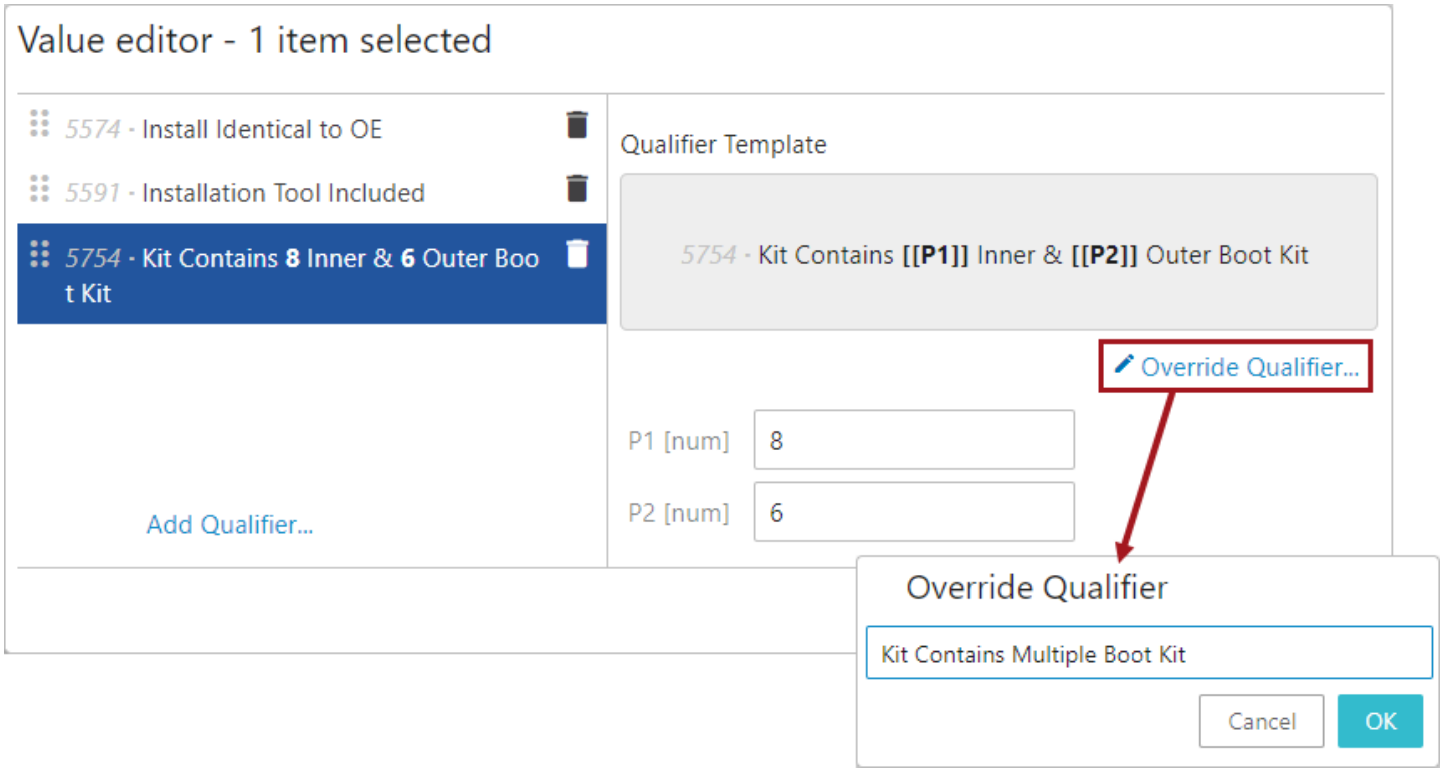



The Value editor dialog has two columns. The left column displays the qualifiers (along with the QualifierID) that are currently referenced to the application record. Clicking on a qualifier listed in the left column will display the template that the qualifier text is adopted to. If a qualifier comprises only static text, then the same text will be displayed in the left and right columns. If a qualifier text has a combination of static text and a parameter, then its template is displayed in the right column along with their parameter fields that allow populating values. Any values typed in the parameter fields will instantly be displayed in the qualifier text displayed in the left column.

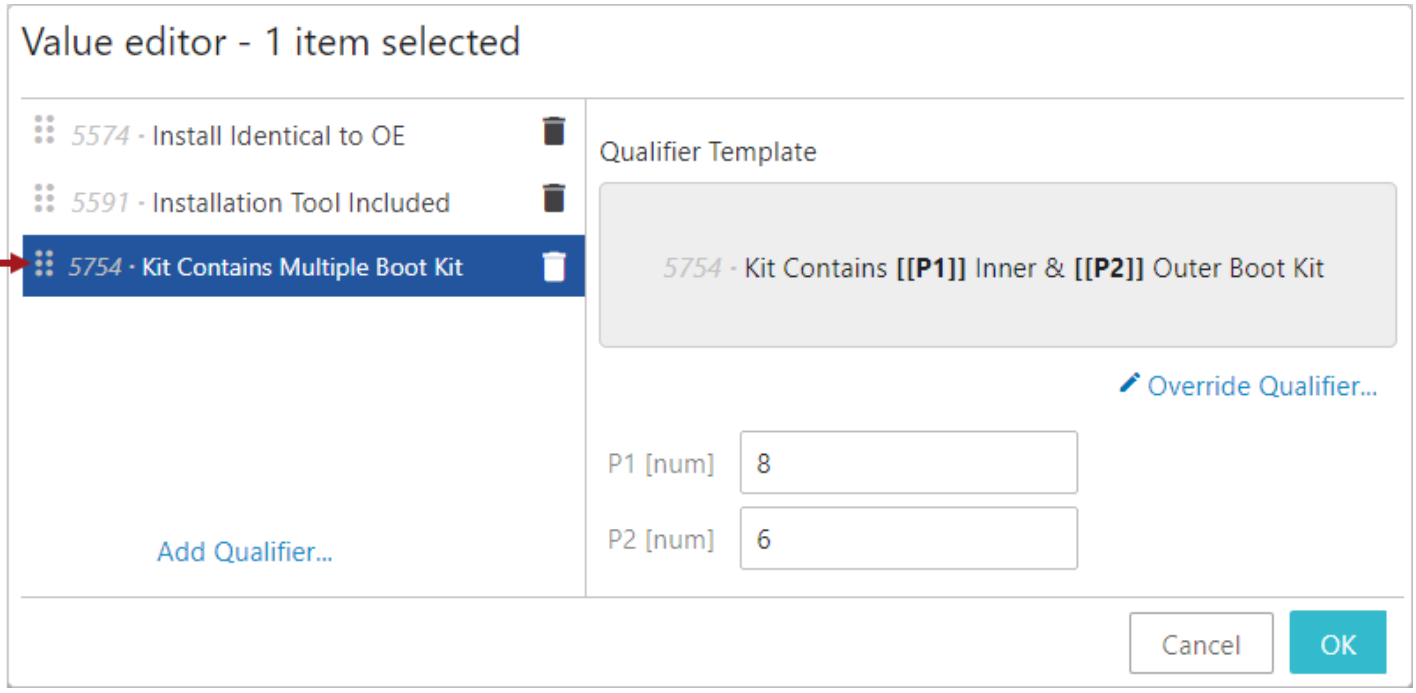
In the example screenshot below, the qualifier template 'Kit Contains [[P1]] Inner & [[P2]] Outer Boot Kit' has two parameters P1 and P2. Adding the values of numeric '8' and '6' in parameters P1 and P2, respectively, has lead to the display of qualifier text 'Kit Contains 8 Inner & 6 Outer Boot Kit' in the left column.




Clicking on the 'Override Qualifier' button that is available in the right column will open an 'Override Qualifier' dialog. Any text entered in the dialog will override the existing qualifier text (not the template). In the example screenshot below, the qualifier template 'Kit Contains [[P1]] Inner & [[P2]] Outer Boot Kit' is overwritten with a new text 'Kit Contains Multiple Boot Kit.'



The below image shows the overridden text 'Kit Contains Multiple Boot Kit' displayed as the qualifier text in the left column.




### Application Approval Status Table Header

When configured, the 'Application Approval Status' component can be used to display the approval status of the application records within an Application Manager Results Table and/or Application Editor Screen. The addition of this column will help filter the applications based on their approval status. Additionally, within a cell, an Information icon ( ) will be visible that provides detailed information on the approval status.

## Prerequisites

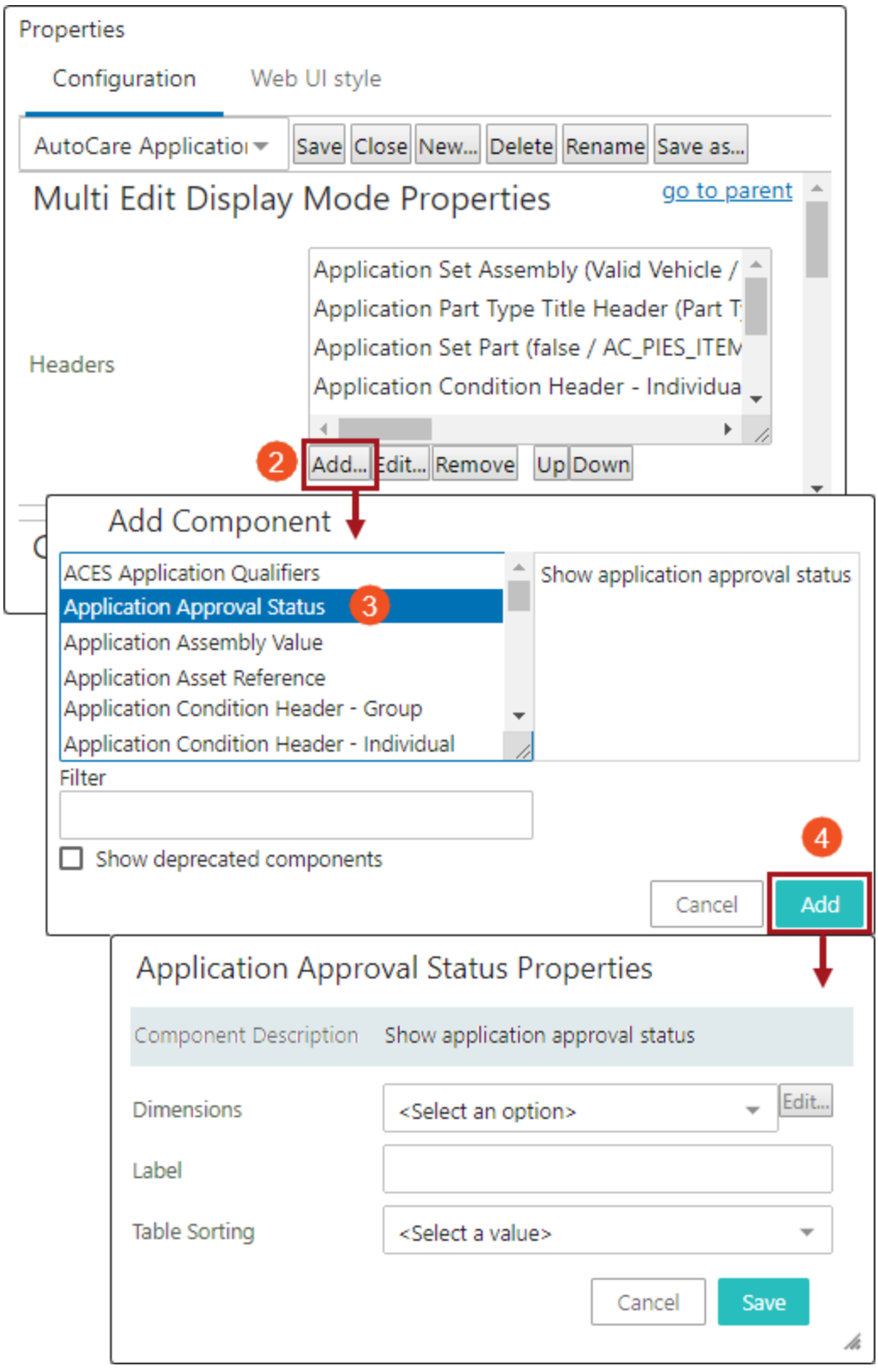
It is expected that anyone configuring the Application Approval Status component is familiar with the Web UI Designer as basic concepts for working with the designer are not covered in this section. In addition, the user must have appropriate privileges to access the designer. Additional information can be found in the **Designer Access** section of the **Web User Interfaces** documentation.

For information about accessing and configuring the Node List component, refer to the **Node List Component** topic within the **Web User Interfaces** section of **STEP Online Help**.

## Configuration Steps

The Application Approval Status component can be added within any of the display modes available in the Node list Properties for the selected Application Manager Screen (or Application Editor Screen). Below are the steps to configure the Application Approval Status component.

1. Using the Web UI designer as shown in the image below, go to Node List Properties > Multi Edit Display Mode Headers parameter for the Application Manager Screen (or Application Editor Screen) that needs to display application approval status.
2. Click the **Add** button for the Headers parameter, and the Add Component dialog will display.
3. Find and select the **Application Approval Status** component.
4. Click the **Add** button, and the Application Approval Status Properties dialog will display (as shown below).

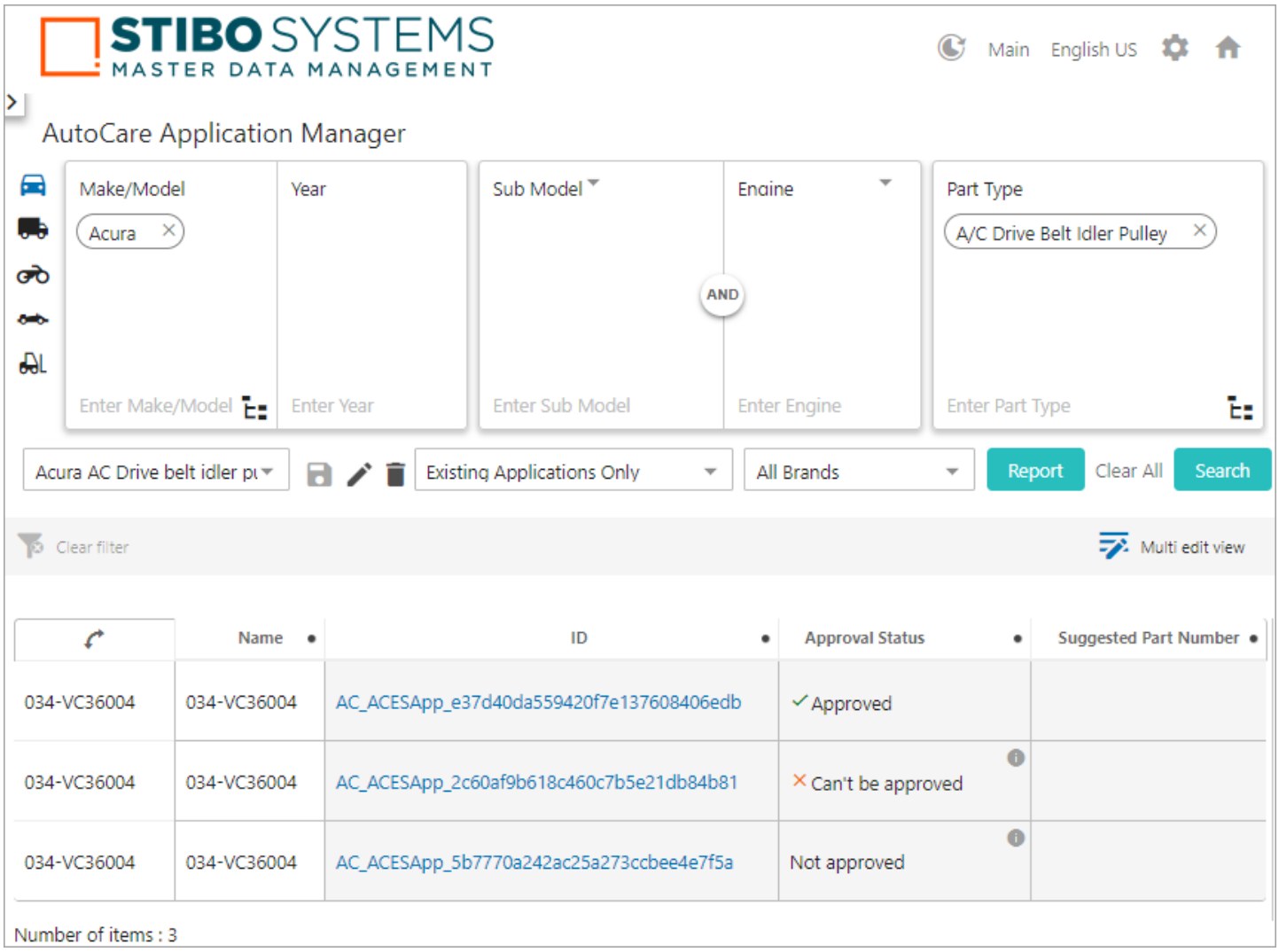



5. Optionally, populate the parameters listed below.

- **Dimensions:** By default, the parameter is blank. Optionally, to configure the column width and height, select 'Table Header Dimensions.
- **Label:** By default, the parameter is blank. But after clicking the **Add** button, the default column header label (Approval Status) will be stored. Optionally, enter the 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.

6. Click the **SAVE** button to save the newly added component to the Headers parameter.
7. 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.
8. Click the **Save** and **Close** buttons for the designer.

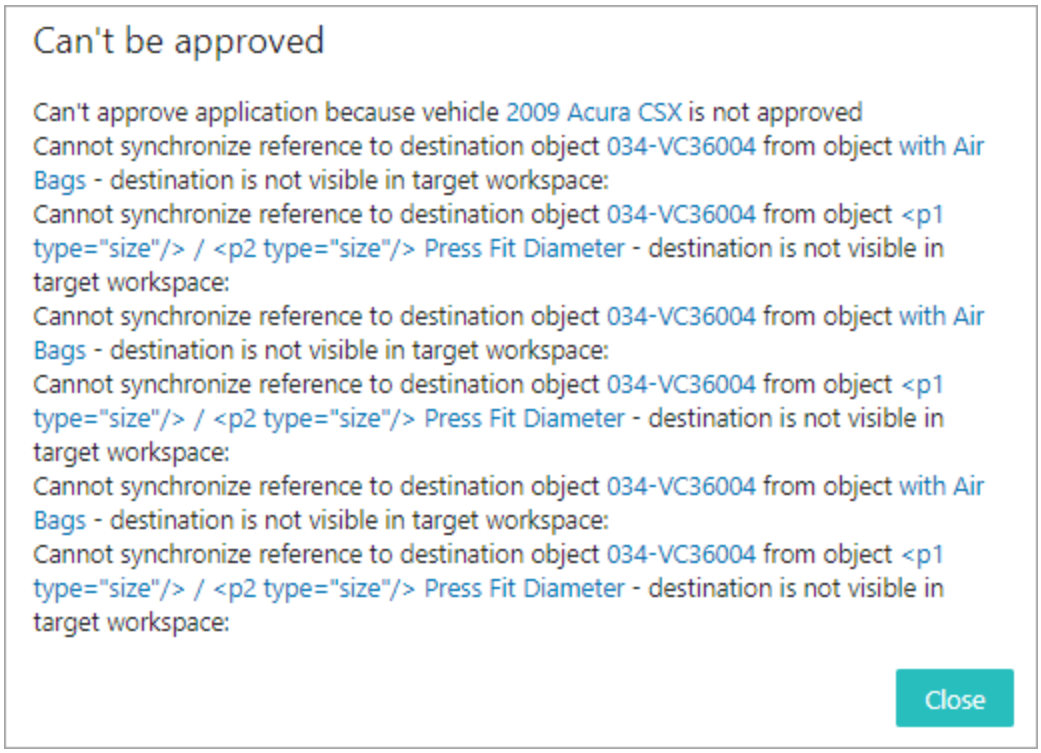
When a search is performed in the Application Manager Screen, the Application Manager results table will display the application records with the approval status of each application as shown below.




There are three Approval statuses that can be displayed:

- **Approved:** This status is displayed when the object and all of its parent nodes and referenced target objects are fully approved.
- **Can't be approved:** This status is displayed when the object has either never been approved, or that the object cannot be fully approved because there are parent nodes or referenced target objects that have not been fully approved yet.
- **Not approved:** This status is displayed when the target has been previously fully approved but now a change has been made in the object that requires it to be approved again (such as changing an attribute value).

And when users click on the Information icon ( ), the reason(s) for the application record for not being approved or cannot be approved is displayed as shown below.




### Application Assembly Value Table Header

When configured within an Application Manager Results Table, the 'Application Assembly Value' component can be used to display an attribute value of the assembly / vehicle linked to an application record. The attribute value is listed in a column against each result in the Application Manager screen. Users can click on the cell content (attribute value) to navigate to the screen that is mapped to display the assembly / vehicle object.

## Prerequisites

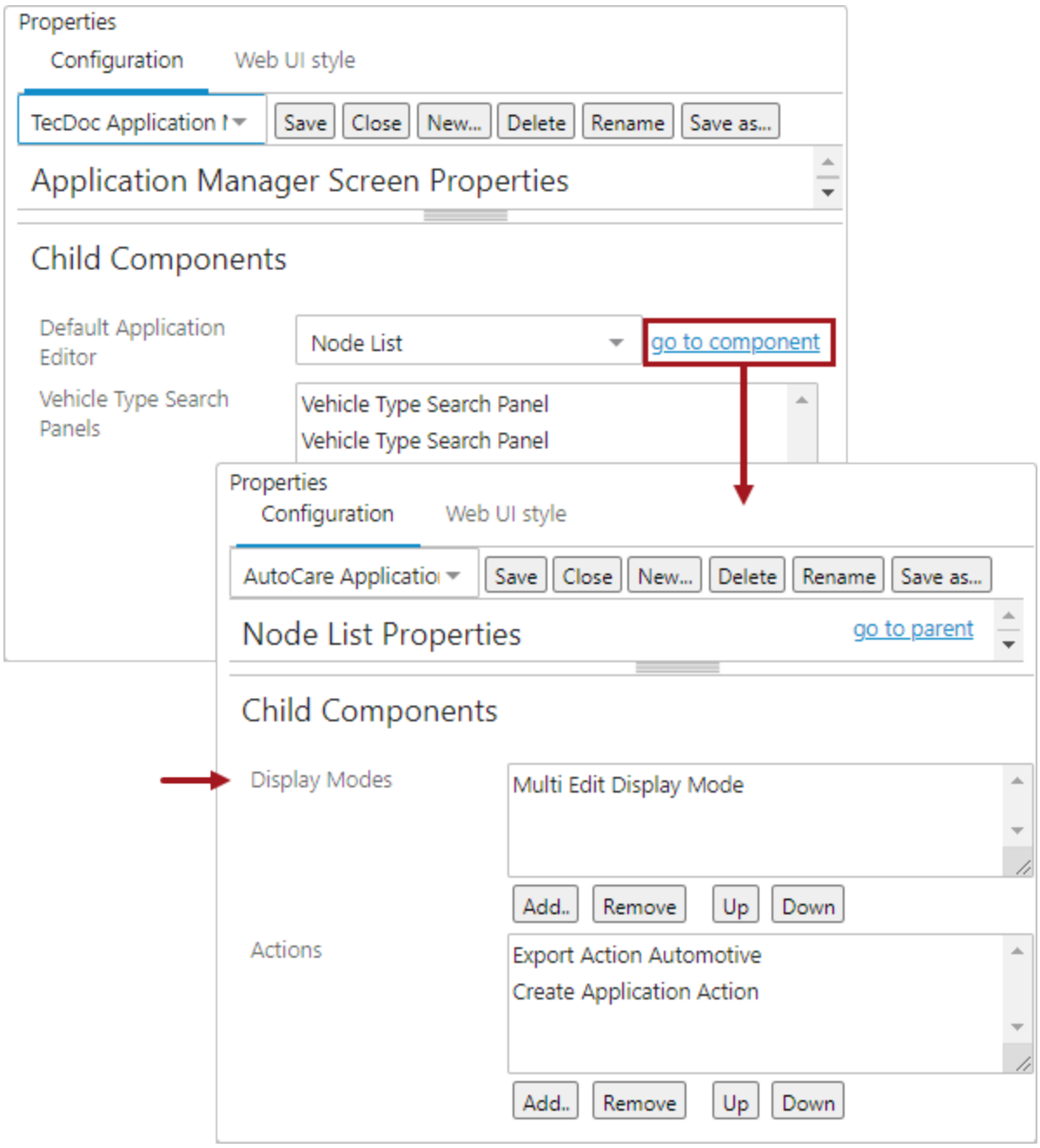
Anyone configuring the Application Assembly Value component is expected to be familiar with the Web UI Designer, as basic concepts for working with the designer are not covered in this section. In addition, the user must have appropriate privileges to access the designer. Additional information can be found in the **Designer Access** section of the **Web User Interfaces** documentation.

As this component is configured as a table header, users should be familiar with the process required to add table headers into the Node List component, as basic concepts for accessing and adding the table headers are not covered in this section. For information about accessing and configuring the Node List component, refer to the **Node List Component** topic within the **Web User Interfaces** section of **STEP Online Help**.

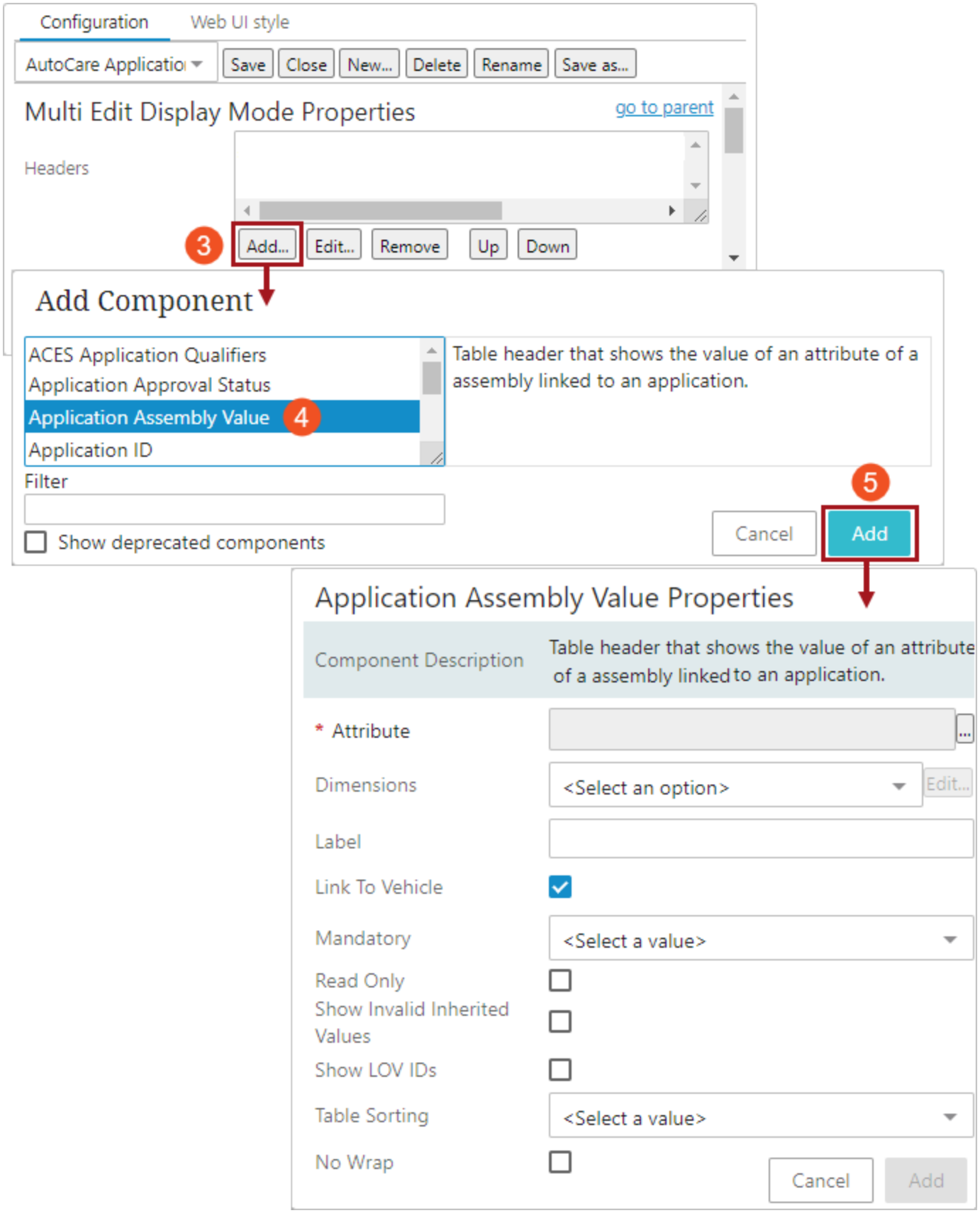
## Configuration Steps

The Application Assembly Value component is added within any of the display modes available in the Node list Properties for the selected Application Manager screen.Follow the steps below to configure the Application Assembly Value component.

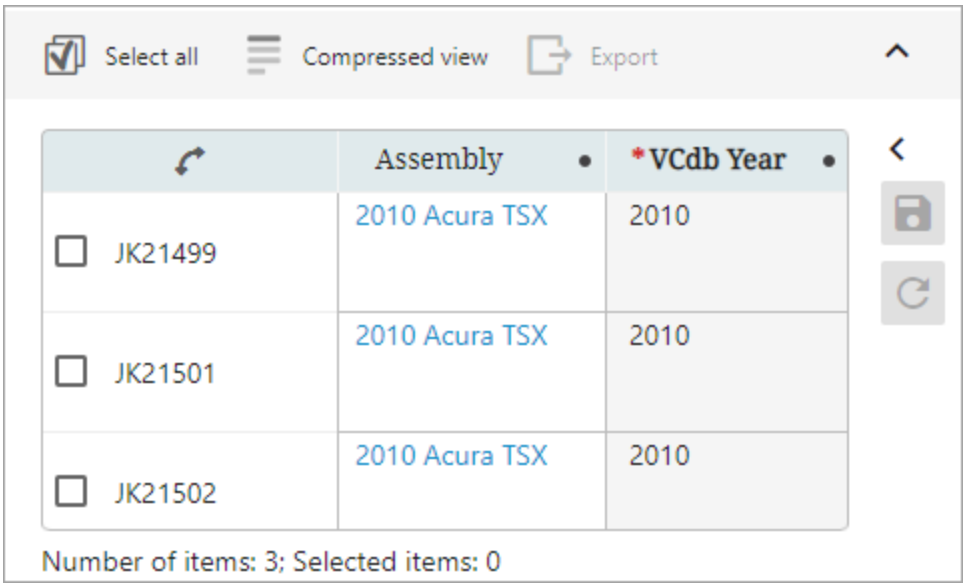
1. Using the Web UI designer as shown in the image below, go to Node List Properties > Display Modes parameter for the Application Manager Screen that needs to display attribute value.




2. Add a display mode (or edit the one that exists) by clicking the **Add** button available next to the Display Modes parameter.
3. Double click to enter the added (or available) display mode, click the **Add** button for the Headers parameter, and the Add Component dialog will display.
4. Find and select the **Application Assembly Value** component.
5. Click the **Add** button, and the Application Assembly Value Properties designer screen will display (as shown below).


Configuration
Web UI style
AutoCare Applicatio
Save
Close
New...
Delete
Rename
Save as...
Multi Edit Display Mode Properties
go to parent
Headers
3
Add...
Edit...
Remove
Up
Down
Add Component
ACES Application Qualifiers
Application Approval Status
Application Assembly Value
4
Application ID
Table header that shows the value of an attribute of a assembly linked to an application.
Filter
Show deprecated components
5
Cancel
Add
Application Assembly Value Properties
Component Description
Table header that shows the value of an attribute of a assembly linked to an application.
* Attribute
Dimensions
<Select an option>
Edit...
Label
Link To Vehicle
Mandatory
<Select a value>
Read Only
Show Invalid Inherited Values
Show LOV IDs
Table Sorting
<Select a value>
No Wrap
Cancel
Add

6. Populate the parameters listed below:
   - **Attribute:** This parameter must be populated with an attribute that is valid for the assembly / vehicle object.
   - **Dimensions:** By default, the parameter is blank. Optionally, to configure the column width and height, select 'Table Header Dimensions.'
   - **Label:** By default, the parameter is blank, but after the **Add** button is clicked, the default column header label (name of the attribute) will be stored. Optionally, enter the desired label to be displayed as the column header within the Application Manager result's table.
   - **Link to Vehicle:** This checkbox parameter allows users to choose if they want the attribute value to be displayed as a hyperlink or as plain text. Selecting the Link To Vehicle parameter will enable the values to display as a hyperlink. Clicking the hyperlink will navigate to the screen that is mapped to display the Assembly / Vehicle.
   - **Mandatory:** This parameter is used in Multi Edit Display mode to visually prompt the user to add a value for the defined 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 that a value must be added to the configured attribute. In the example below, an asterisk (*) displays to the left of the attribute 'VCdb Year.'

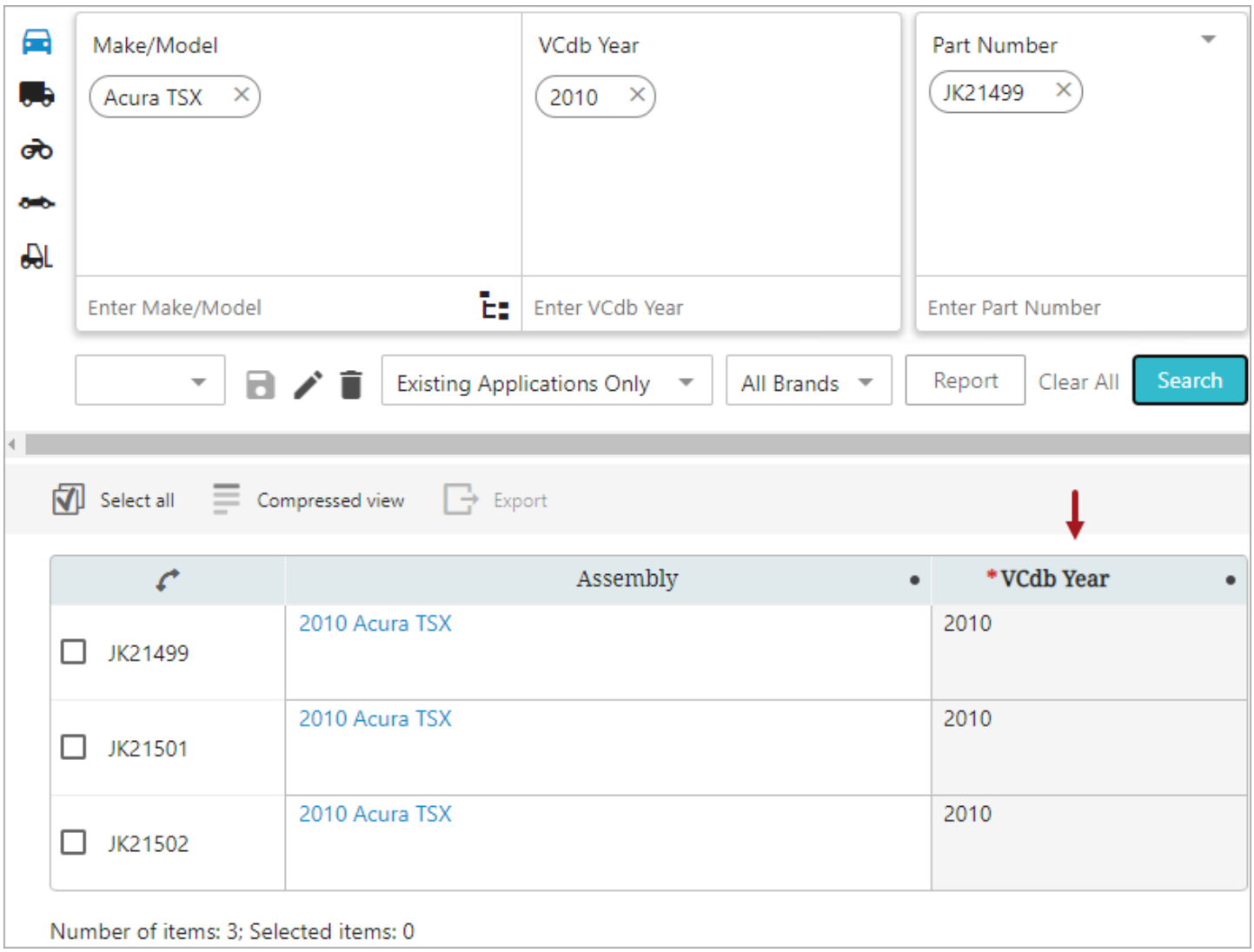



   - **Read Only:** Determines if the value field should be editable or read only.
     - If disabled (default), then users can edit the cell
     - If enabled, the attribute value on the screen will be locked so users cannot make edits
   - **Show Invalid Inherited Values:** Determines if the inherited attribute values need to be displayed even if the attribute is invalid to the assembly / vehicle.
     - If enabled (not default), and the attribute has values inherited from the parent, then the inherited attribute value will display against the application record.

- **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.
- **Table Sorting:** By default, the parameter is blank. Optionally, select from the dropdown values to specify the default sorting order of the column data.
- **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 attribute value breaks to a second line, allowing the display of the whole value.

7. Click the **Add** button (or **Save** button if editing an existing component) 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.

When a search is performed in the Application Manager screen, the Application Manager Results Table will display the application records with the value of the configured attribute as shown below.




### Application Comment Table Header

Any additional information of an application record not covered by a vehicle attribute or qualifier can be included as a free-form text comment / note. These individual application comments are stored as STEP objects in a separate hierarchy within the system. Usually, these comments would contain information that the manufacturer wants the catalog reader to know when ordering a part. An example of this type of comment might be an installation instruction such as 'Coil wire modification required. Splice in original connector.' An application record can be referenced to multiple comments in the system.

The application comments are identified by different names across different standards. The following are ways these comments are addressed in different standards:

- In the Auto Care standard, these comments are identified as Application Notes. In the ACES file, these comments are embedded within the <Note> tag. In the system, the relation between the application records and the Application Notes is established using the 'ACES Application To Application Note' Product Reference Type (ID = AC_ACESApplicationToApplicationNote).

- Similarly, the TecDoc standard addresses these comments as Text Blocks. In the system, the relation between the supplier linkages and the Text Blocks is established using the 'Supplier Linkage To Text Block' Product Reference Type (ID = TD_SupplierLinkageToTextBlock).

The 'Application Comment' table header component can be configured in the Web UI within the Application Manager results table and Application Editor Screen to display the application comments. When configured, the linked application comments are listed in a column against each result. Further, users can double click on the cell to open up the Value Editor window to add or delete a comment reference. This topic provides insight into configuring the Application Comment table header in the Web UI.

## Prerequisites

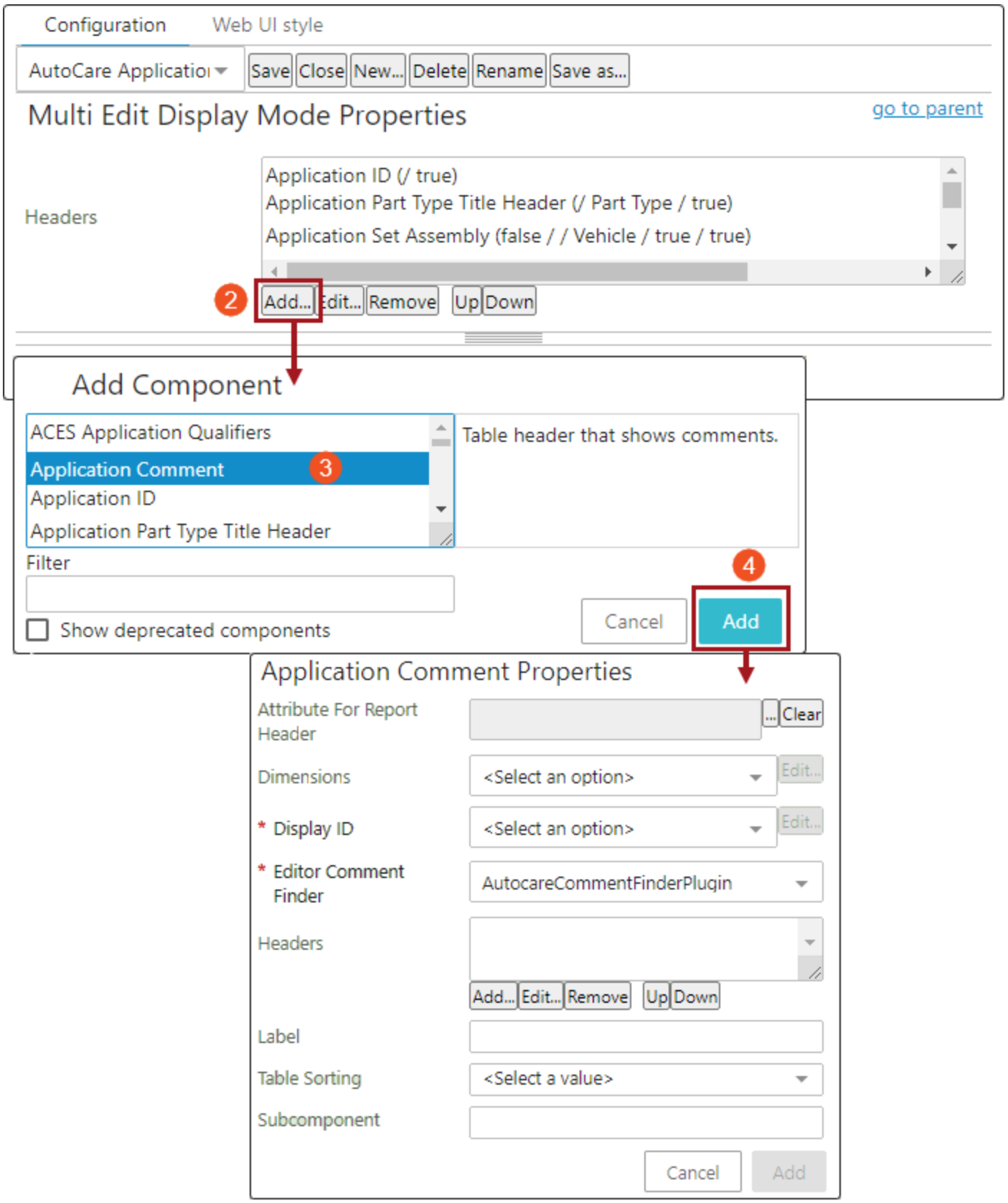
Anyone configuring the Application Comment component is expected to be familiar with the Web UI Designer as basic concepts for working with the designer are not covered in this section. In addition, the user must have appropriate privileges to access the designer. Additional information can be found in the **Designer Access** section of the **Web User Interfaces** documentation.

For information about accessing and configuring the Node List component, refer to the **Node List Component** topic within the **Web User Interfaces** section of **STEP Online Help**.

## Configuration Steps

The Application Comment component can be added within any of the display modes available in the Node list Properties for the selected Application Manager Screen (or Application Editor Screen). Below are the steps to configure the Application Comment component.

1. Using the Web UI designer as shown in the image below, go to Node List Properties > Multi Edit Display Mode Headers parameter for the Application Manager Screen (or Application Editor Screen) that needs to display application notes (comments).
2. Click the **Add** button for the Headers parameter, and the Add Component dialog will display.
3. Find and select the **Application Comment** component.
4. Click the **Add** button, and the Application Comment Properties dialog will display (as shown below).

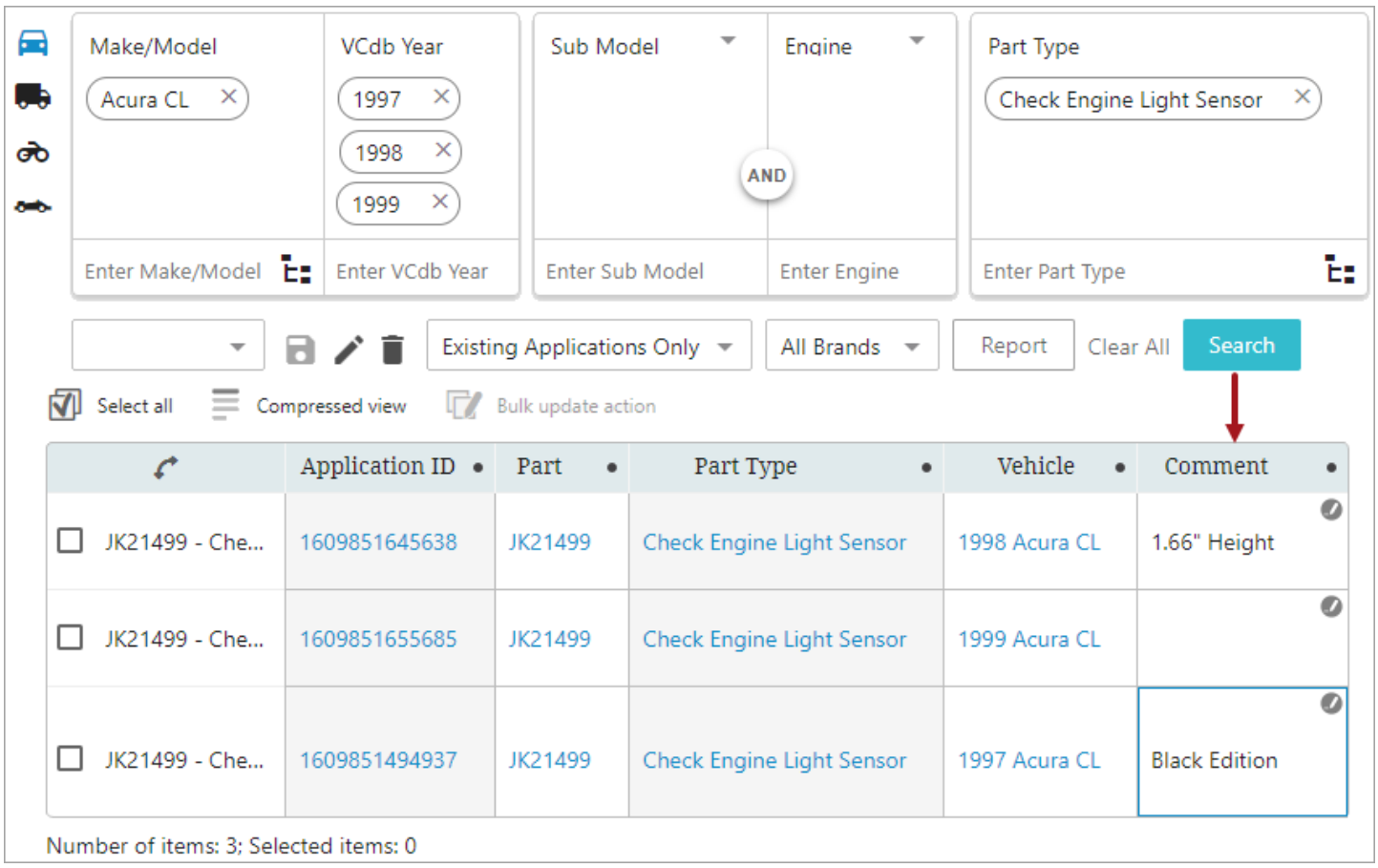

Configuration
Web UI style
AutoCare Applicatio
Save
Close
New...
Delete
Rename
Save as...
Multi Edit Display Mode Properties
go to parent
Headers
Application ID (/ true)
Application Part Type Title Header (/ Part Type / true)
Application Set Assembly (false / / Vehicle / true / true)
2
Add...
Edit...
Remove
Up
Down
Add Component
ACES Application Qualifiers
Application Comment
Application ID
Application Part Type Title Header
3
Table header that shows comments.
Filter
Show deprecated components
4
Cancel
Add
Application Comment Properties
Attribute For Report Header
Clear
Dimensions
<Select an option>
Edit...
* Display ID
<Select an option>
Edit...
* Editor Comment Finder
AutocareCommentFinderPluqin
Headers
Add...
Edit...
Remove
Up
Down
Label
Table Sorting
<Select a value>
Subcomponent
Cancel
Add

5. Populate the required parameters listed below:
   - **Display ID:** By default, this required parameter is blank. This parameter determines if the ID or Name of the object, or the attribute value of the application comment should be displayed within the table cell. The 'Node ID' option allows the ID of the application comment to be displayed within the cell, and the 'Node Name' option displays the name of the application comment. Selection of the 'Attribute Value' option further requires an attribute (valid for the application note object type) to be configured that allows the values of the configured attribute to be displayed within the cell.
   - **Editor Comment Finder:** This parameter defines the plugin that is to be used to find the application comment.
     - The Auto Care standard should select the 'AutocareCommentFinderPlugin' option.
     - The TecDoc standard should select the 'TecDocCommentFinderPlugin' option.

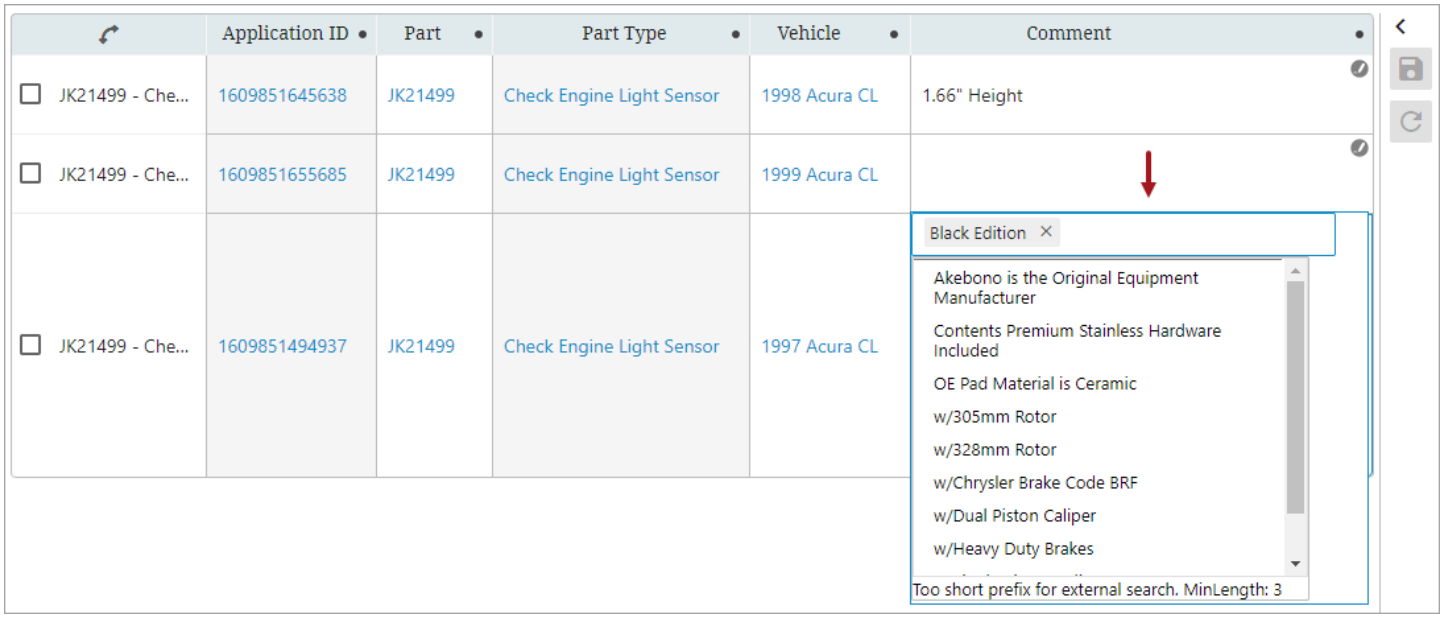
6. Optionally, populate the parameters listed below.
   - **Attribute For Report Header:** By default, the parameter is blank. This parameter allows users to define an attribute that should be used to retrieve the values to be used within the 'Comment' column header within the Application Coverage Report. If this parameter is left empty, then the default column header will use the value that is in the 'Label' field as the column header in the report and all referenced comments will be combined into one cell. If an attribute is defined for this parameter (for example, AC_ACESNoteType), then the values of this attribute will be used as the column header(s) and Notes with the same Note Type will be grouped within the same cell.
   - **Dimensions:** By default, the parameter is blank. Optionally, to configure the column width and height, select 'Table Header Dimensions.'
   - **Headers:** The recommended practice is to add at least the ID and the Name headers within this parameter. Table headers configured in this parameter display the application comment's associating information when the user accesses the Value Editor dialog by clicking on the Application Comment edit icon( ). Further, a column for reference metadata attribute (via Reference Metadata Value Header component) can also be added to access information like Display Sequence, Note Type Calculated, etc. (metadata attributes).
   - **Label:** By default, the parameter is populated with the value 'Comment.' Optionally, enter the desired label to be displayed as the column header within the Application Manager results table (or Application Editor Screen).
   - **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. Click the **SAVE** 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.

When a search is performed in the Application Manager Screen, the Application Manager results table will display the application records with the existing comments (if any) of each application, as shown below.

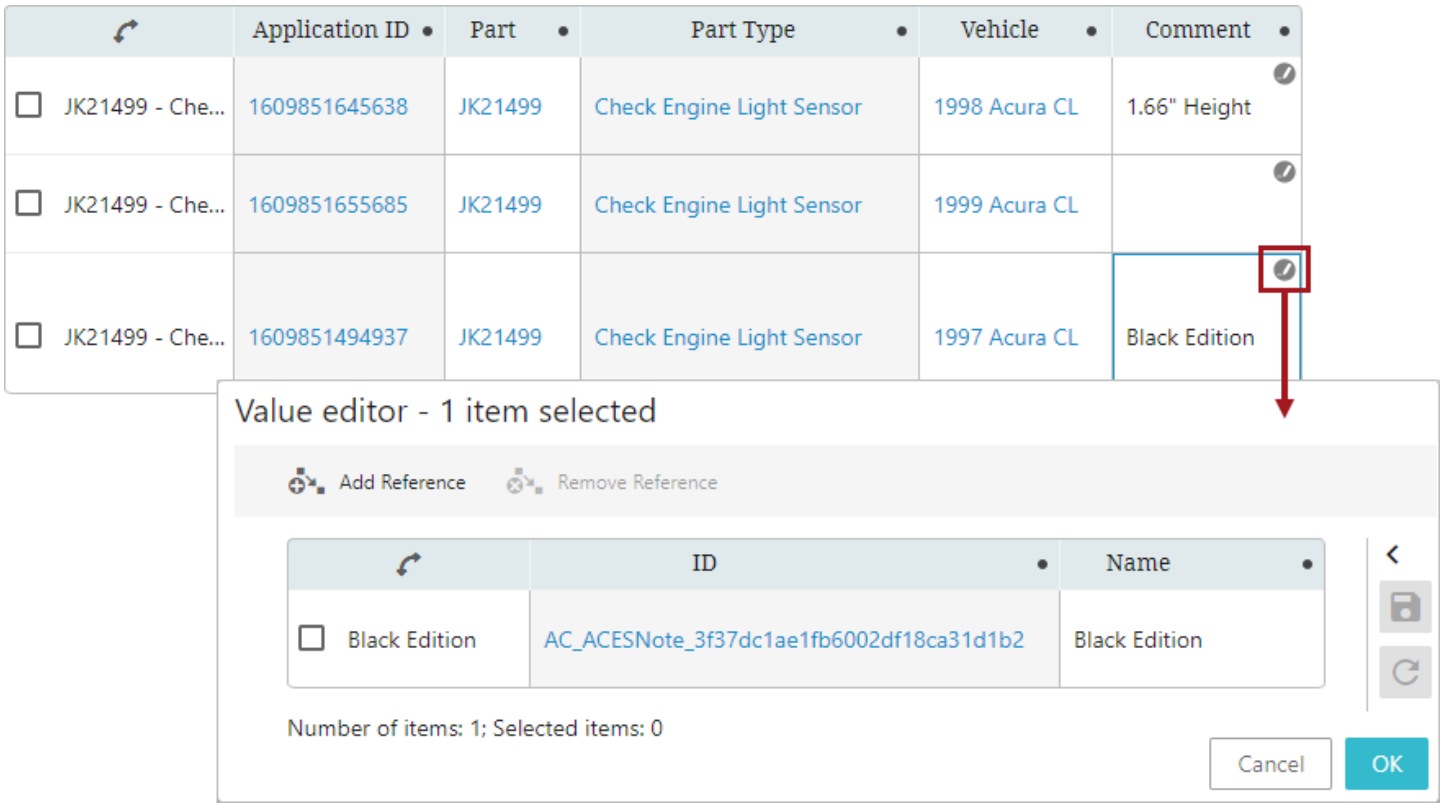



When users double click on the cell, a typeahead field will be displayed in the cell. Using the typeahead field, users can type in the first letters of any word within the comment name to quickly find the comment for selection (as shown below). Selecting any comment will add a reference to the selected comment. Otherwise, click the edit button ( ) to the right of the cell, and a Value editor dialog will display.

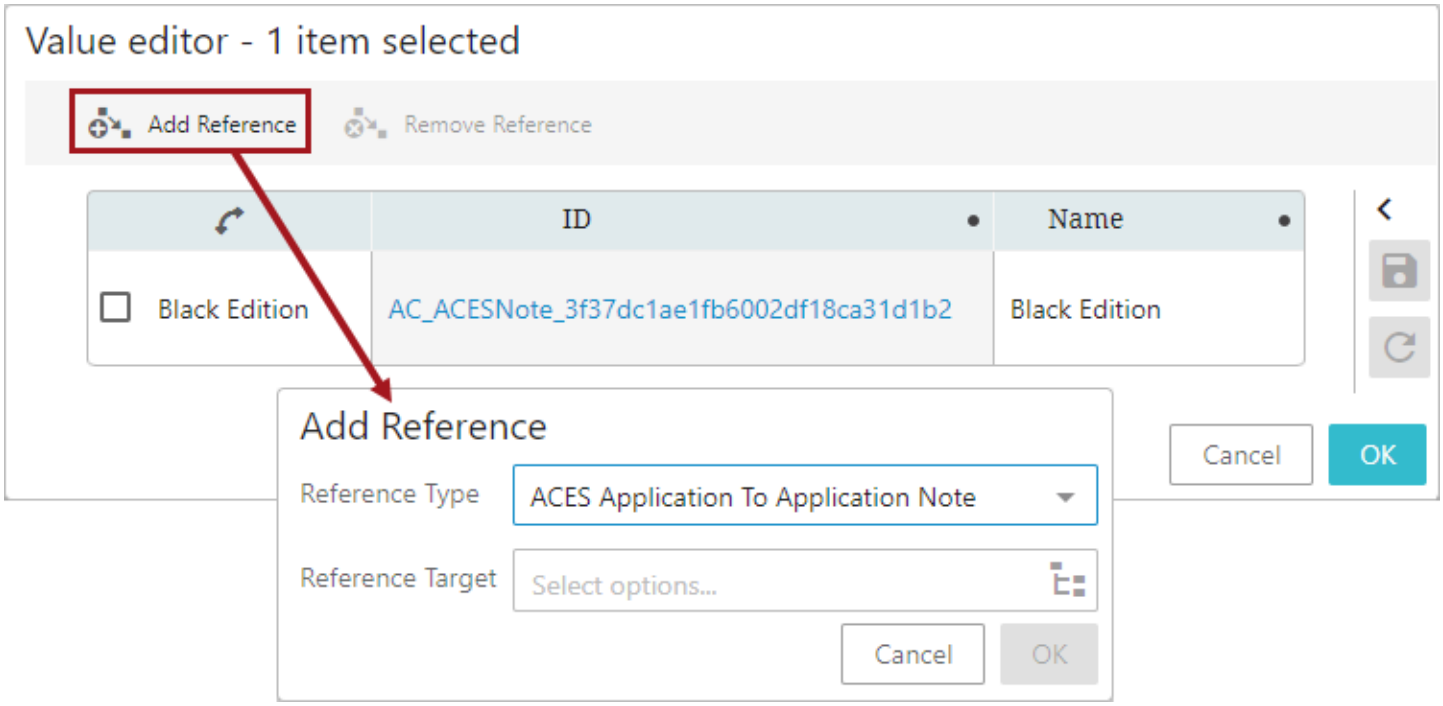


**Using Application Comment Value Editor to Add or Remove References to Comments**

To add or remove comment to/from an application via the 'Application Comment' component, click the edit button ( ) to the right of the cell, and a Value editor dialog will display as shown below.




Clicking the 'Add Reference' button will open the 'Add Reference' dialog. Users can establish the reference to the desired comment either by typing the name / ID in the Reference Target field. Alternately, users can use the node picker ( ) next to the Reference Target field to select the desired comment.




However, it is also possible to remove the reference of an existing comment by selecting the desired comment's checkbox and clicking on the 'Remove Reference' button.

Clicking the OK button closes the dialog and displays the referenced comments in the cell.

**Note:** The OK button is only enabled when there are changes to be saved.

### 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:

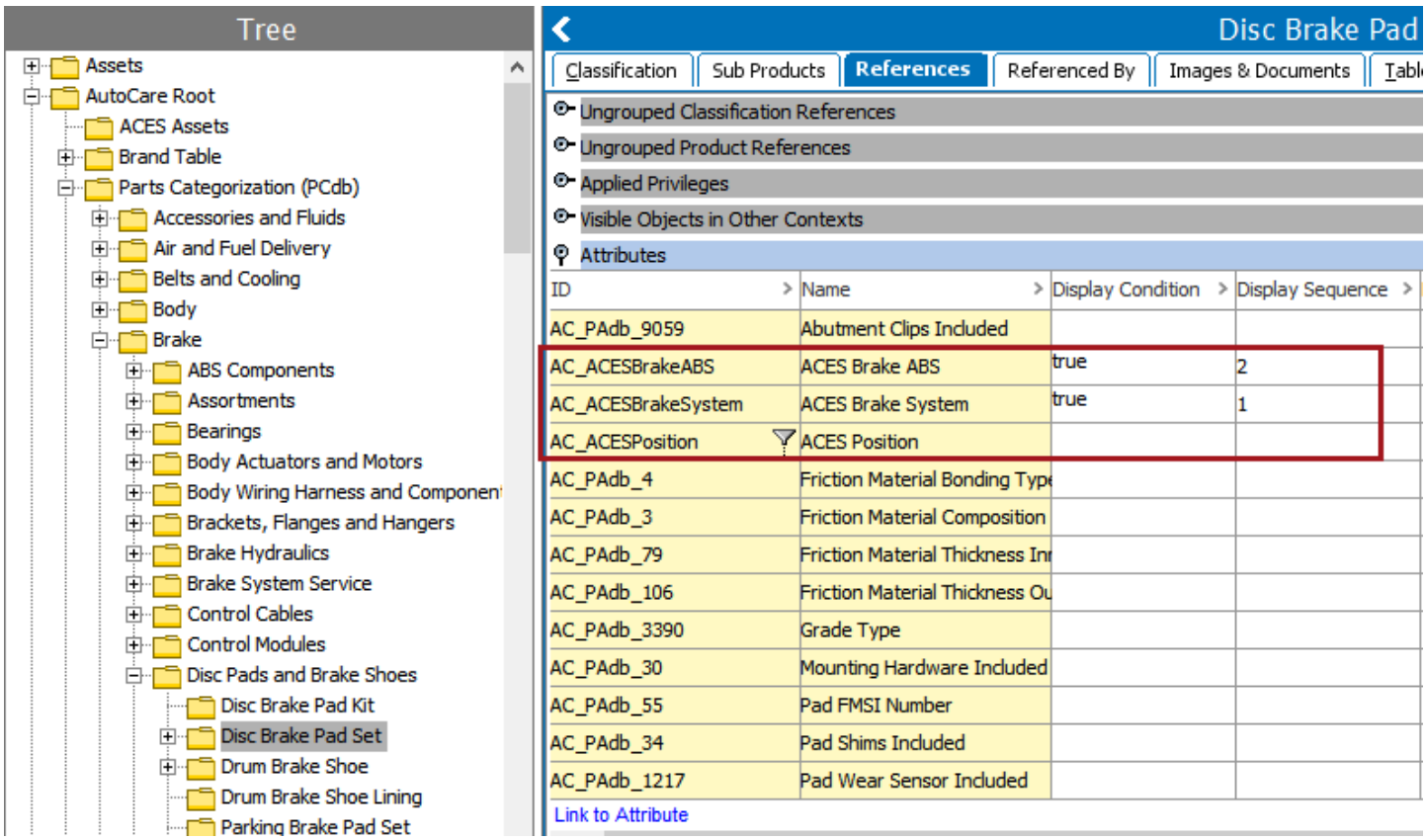
- **Application Condition Header - Individual**: Allows table headers to be expanded to display condition attributes and/or references in individual columns. Condition attributes can be displayed in their own column in the results table and/or Application Coverage Report when one of the following are 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.

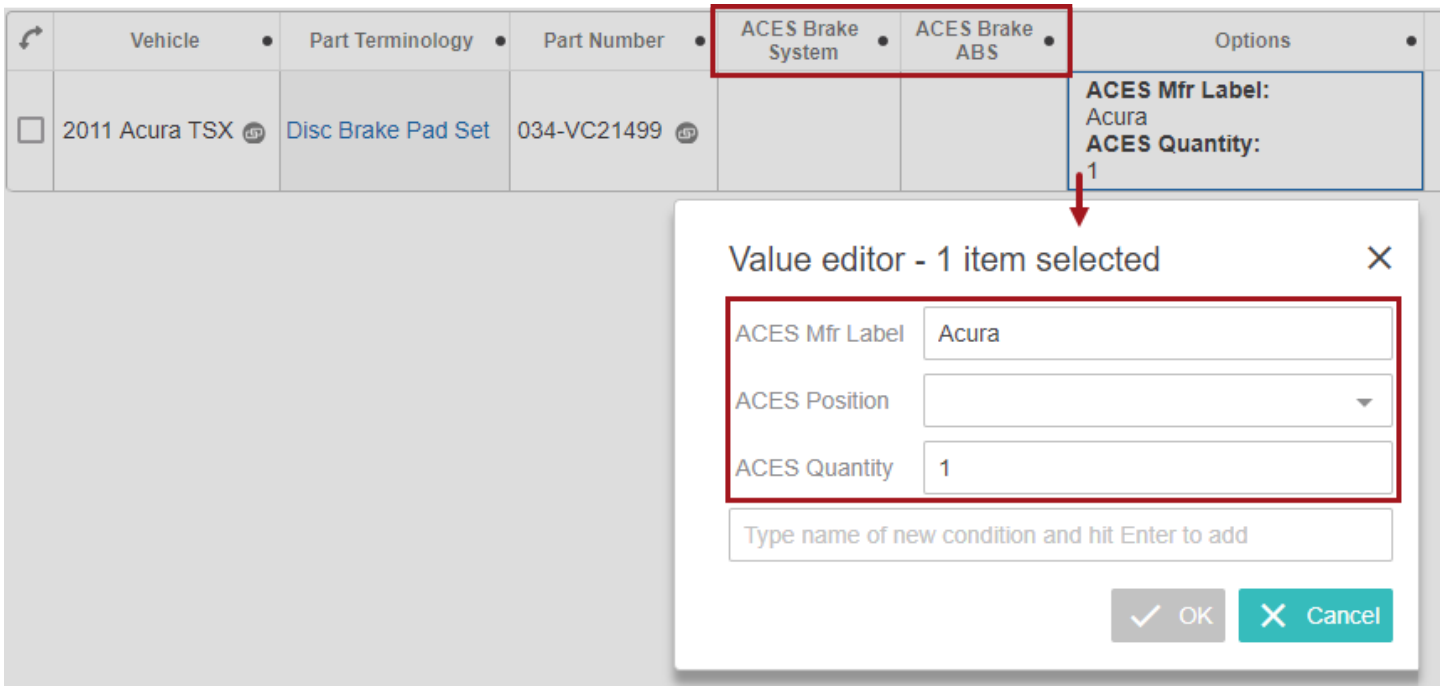
If the condition is modeled using an attribute, the condition link is handled via a simple attribute link to the part type classification.

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.



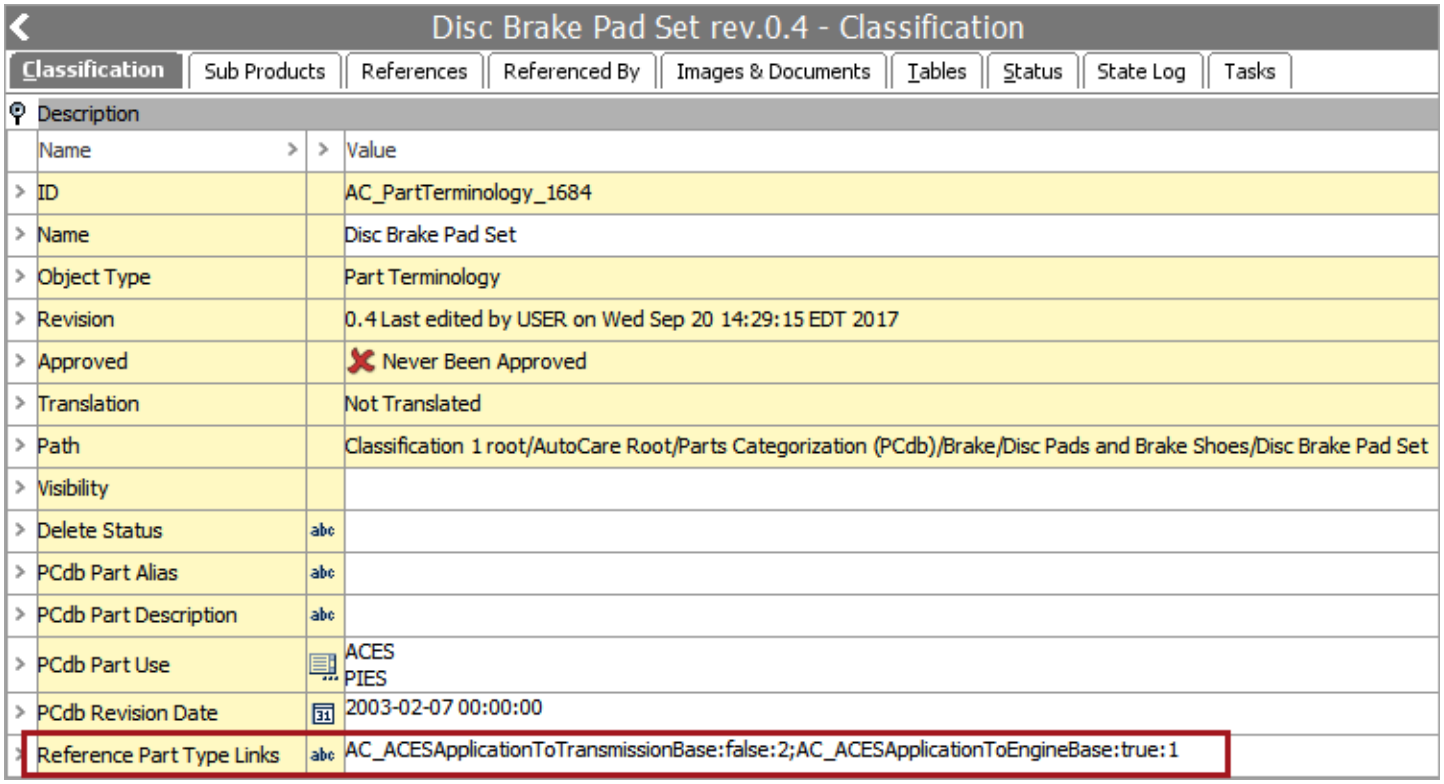

In the screenshot below, the **ACES Brake System** and **ACES Brake ABS** attributes are displayed as individual columns within the results table of an Application Manager. Whereas, the **ACES Mfr Label**, **ACES Position**, and **ACES Quantity** display within the Value editor.



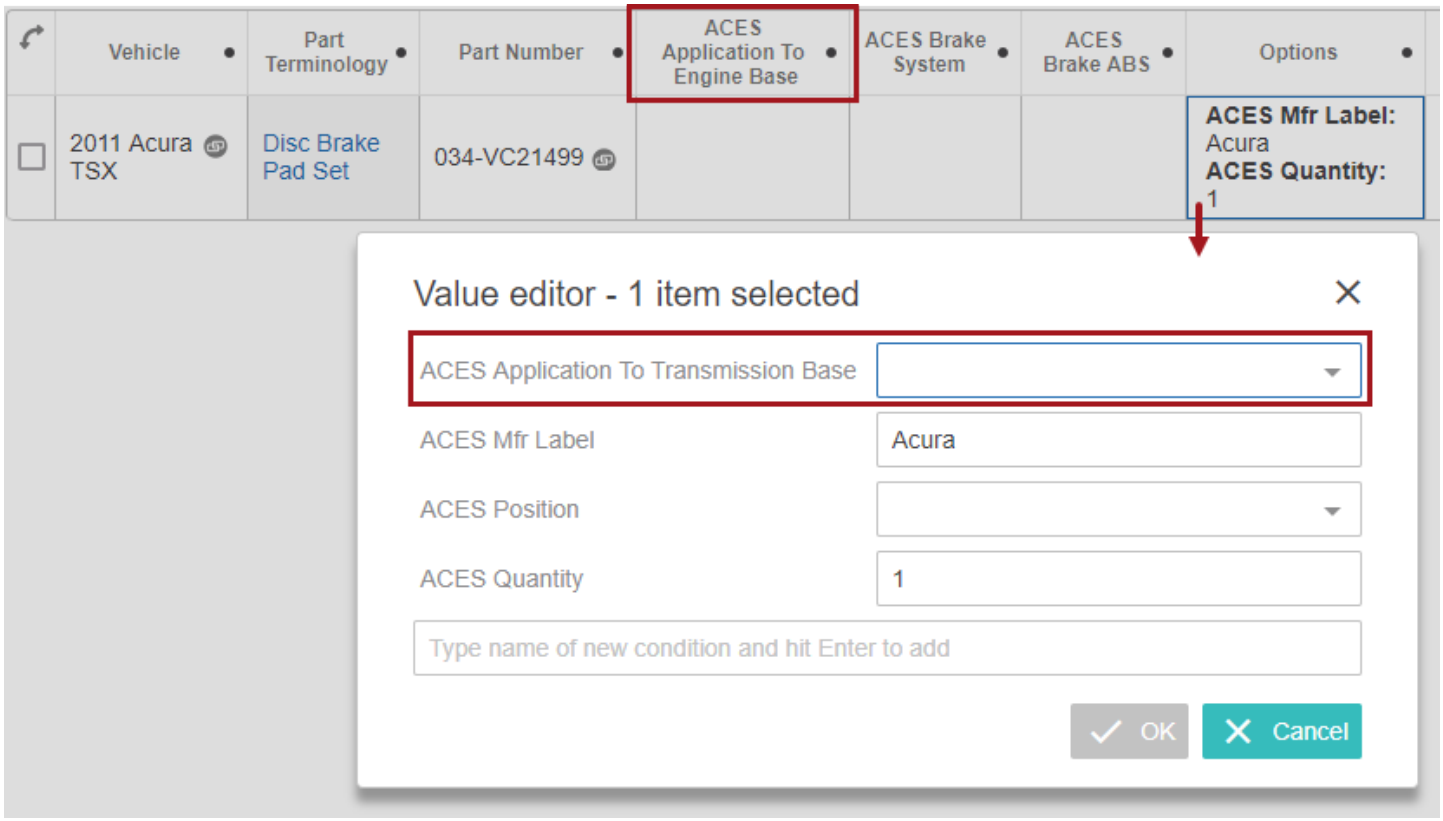
The behavior of the Value editor is the same, whether the condition is modeled using an attribute or a reference. However, on references, the condition link is modeled using the Reference Part Type Links metadata attribute (ID=ReferencePartTypeLinks). The Reference Part Type Links attribute is created 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:



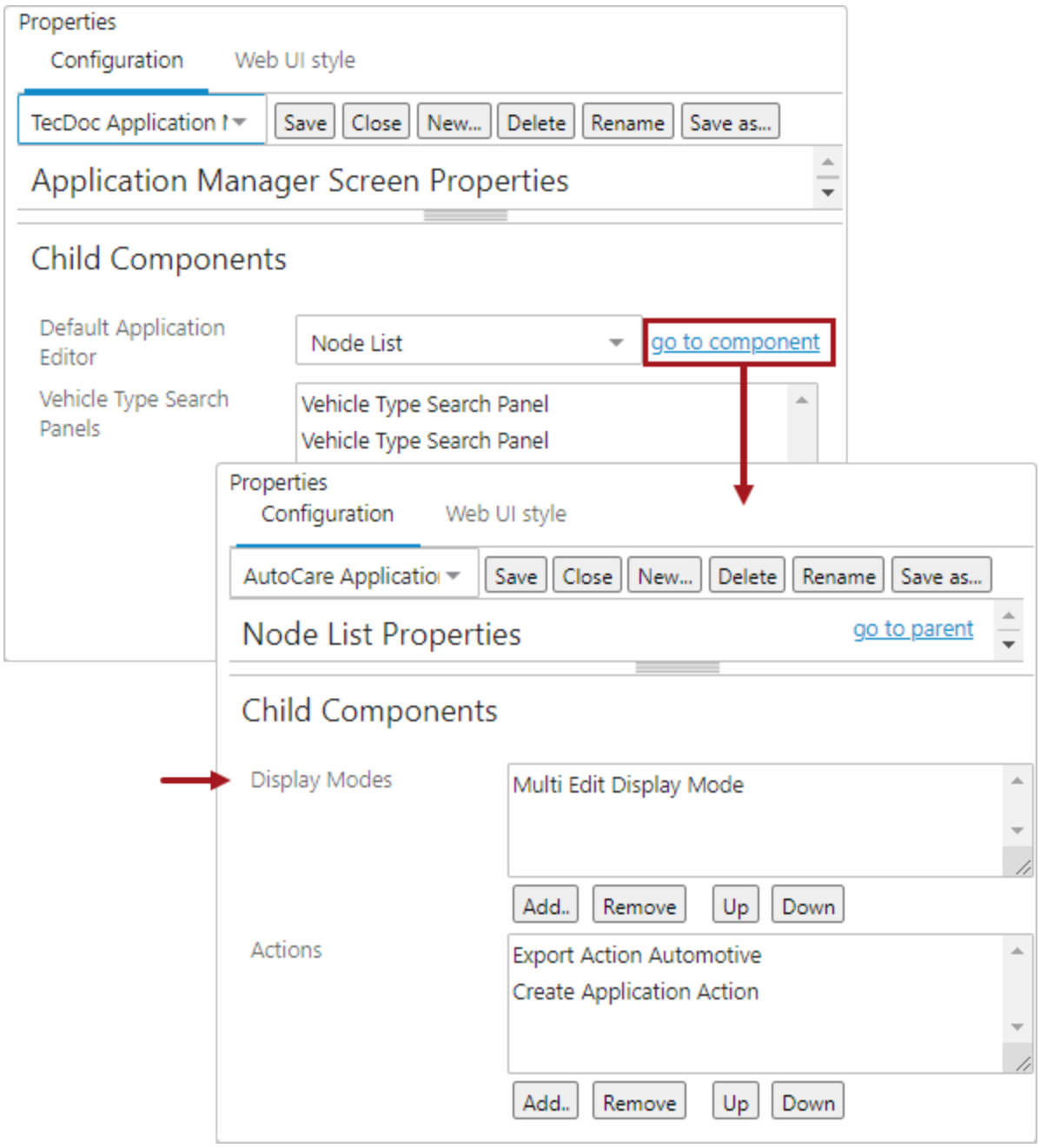

Since the transmission reference is set to false and the engine reference is set to true, the engine option is displayed within its own column in the Application Manager results table, and the transmission option is only displayed within the Value editor (as shown below).



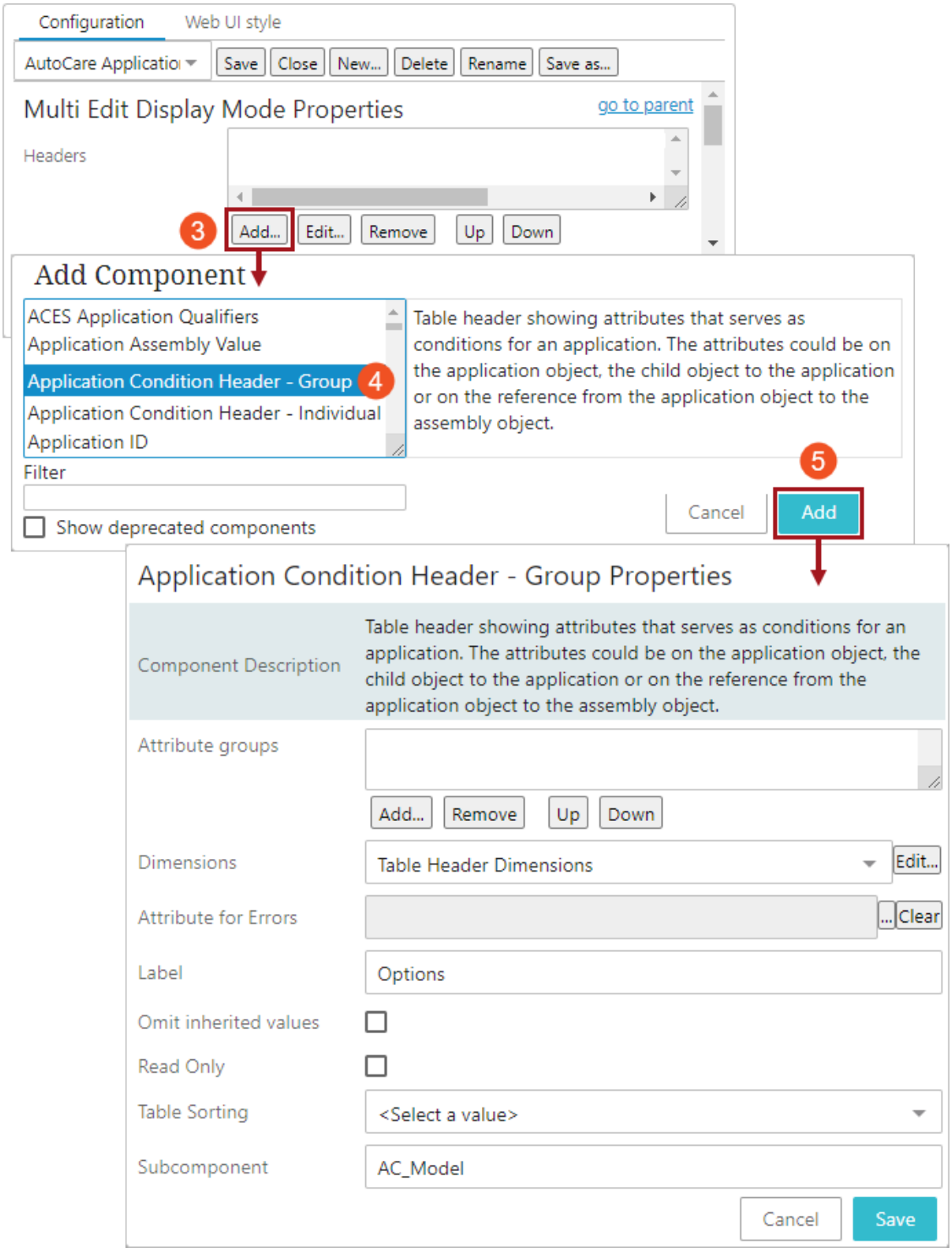

## Configuration Steps

The Application Condition Header - Group table header component is added within any of the display modes available in the Node List Properties for the selected Application Manager screen. Follow the steps below to configure the Application Condition Header - Group component.

1. Using the Web UI designer as shown in the image below, go to Node List Properties > Multi Edit Display Mode Headers parameter for the Application Manager Screen (or Application Editor Screen) that needs to display vehicle option data for applications.
2. Add a display mode (or edit the one that exists) by clicking the **Add** button available next to the Display Modes parameter.



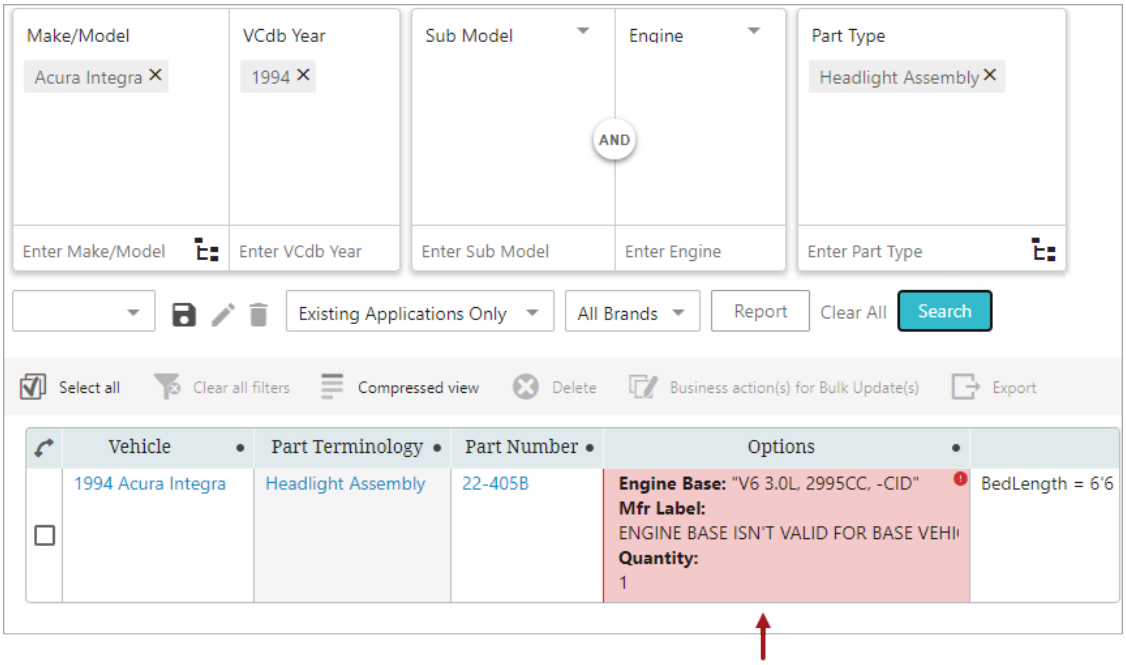

3. Click the **Add** button for the Headers parameter, and the Add Component dialog will display.
4. Find and select the **Application Condition Header - Group** component.
5. Click the **Add** button, and the Application Condition Header - Group Properties dialog will display (as shown below).




6. Populate the parameters listed below:
   - **Attribute Groups:** This parameter allows users to add multiple attribute groups. Adding attribute groups ensures that only the condition attributes and/or references from that group will be evaluated and are allowed to be added in the Condition Editor. The Easy Setup adds the following attribute groups within the parameter:

- The AutoCare standard adds the AutoCare ACES Attributes (ID = AC_ACESAttributes) attribute group within the parameter
- The TecDoc standard adds two attribute groups called Attributes for Supplier Data (ID = TD_ Attributes for Supplier Data) and Criteria Attributes (ID = TD_CriteriaAttributes) within the parameter

- **Dimensions:** By default, the parameter is blank. Optionally, to configure the column width and height, select 'Table Header Dimensions.'
- **Attribute for Errors:** This parameter can be populated with an attribute that stores an application record's vehicle validation error messages. Configuring the attribute within the parameter looks for any values present in the attribute for each application record displayed in the Application Manager Screen. If the attribute holds any vehicle validation error messages, then the cell will be highlighted red, and the error message value will be shown in the mouse over of the cell (as shown below). The 'AC_ACESErrors' attribute is generally used for the AutoCare standard to store the vehicle validation error messages.

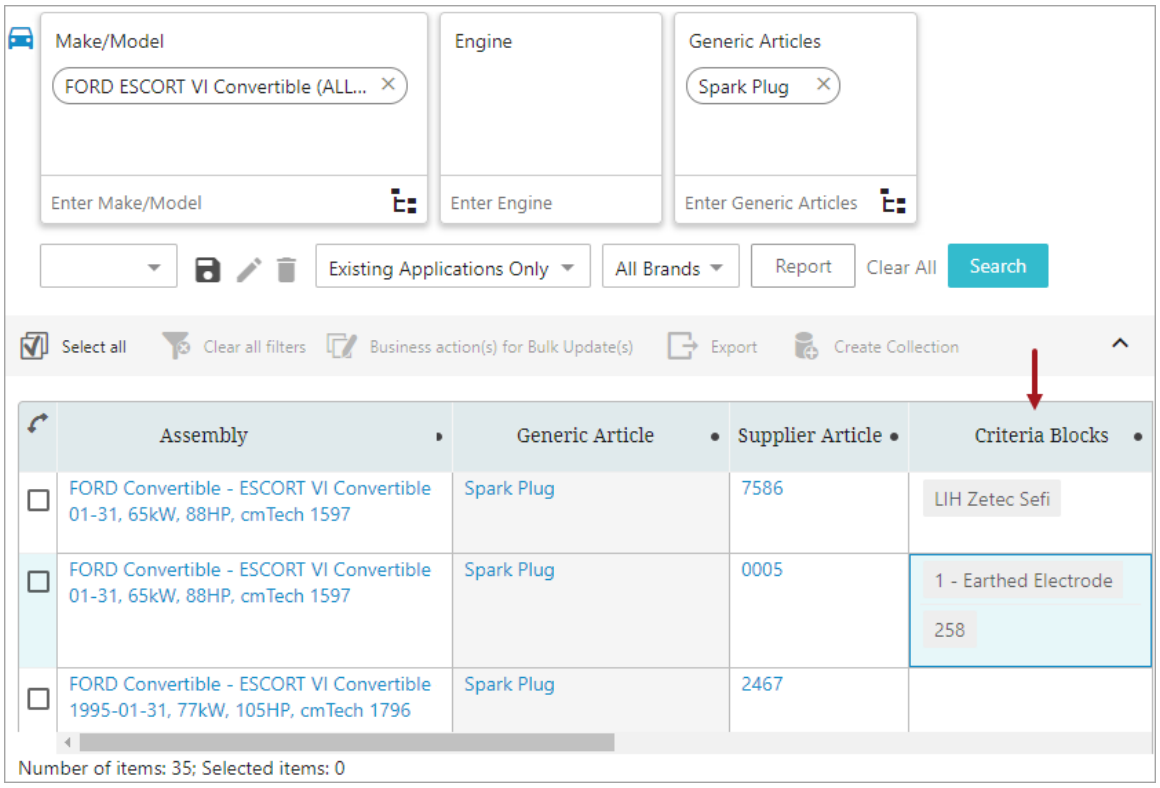



- **Label:** By default, the parameter is blank, but after the **Add** button is clicked, the default column header label (Vehicle) will be stored. Optionally, enter the desired label to be displayed as the column header within the Application Manager result's table.
- **Omit inherited values:** By default, the parameter is unselected. If selected, the grouped application condition header will not add attributes that are inherited from the part, and are not linked to the part type.
- **Read Only:** Determines if the value field should be editable or read only.
  - If disabled (default), then users can edit the cell.
  - If enabled, the cell will be locked so users cannot make edits.
- **Table Sorting:** By default, the display parameter is blank. Optionally, select from the dropdown values to specify the default sorting order of the column data.

- **Subcomponent:** This parameter helps to restrict the display of vehicle options from unwanted assembly / vehicle and allows only the assembly / vehicle of the required object type to be displayed in the typeahead field of the table header cell. Users should populate the ID of the *parent object type* of the required vehicle / assemblies, so that the vehicle options from the vehicle / assemblies of the object types that are not mentioned in this parameter will be excluded from being displayed. The assembly / vehicle object types are determined in the Automotive component model.

### Application Criteria Blocks Table Header

The Application Criteria Blocks component is used to handle criteria blocks in the Web UI. Users can add, delete, and edit the criteria blocks when configured within the TecDoc Application Manager Screen (and Application Editor Screen). Users can also rearrange the position order of the criteria blocks. Below is an example of the Application Manager Screen with the Application Criteria Blocks component configured. Of the three application records displayed below, the first has one criterion, the second has two criteria, and the third has no criteria defined.




There are different ways to handle criteria of application records (linkages). This topic deals with handling application records (linkages) using the criteria blocks in the system.

## Prerequisites

The system must be modeled to handle the criteria blocks. Additional information can be found in the **Criteria Block Handling** topic within the **Importing Automotive Data** section of the **Automotive Reference Guide**.

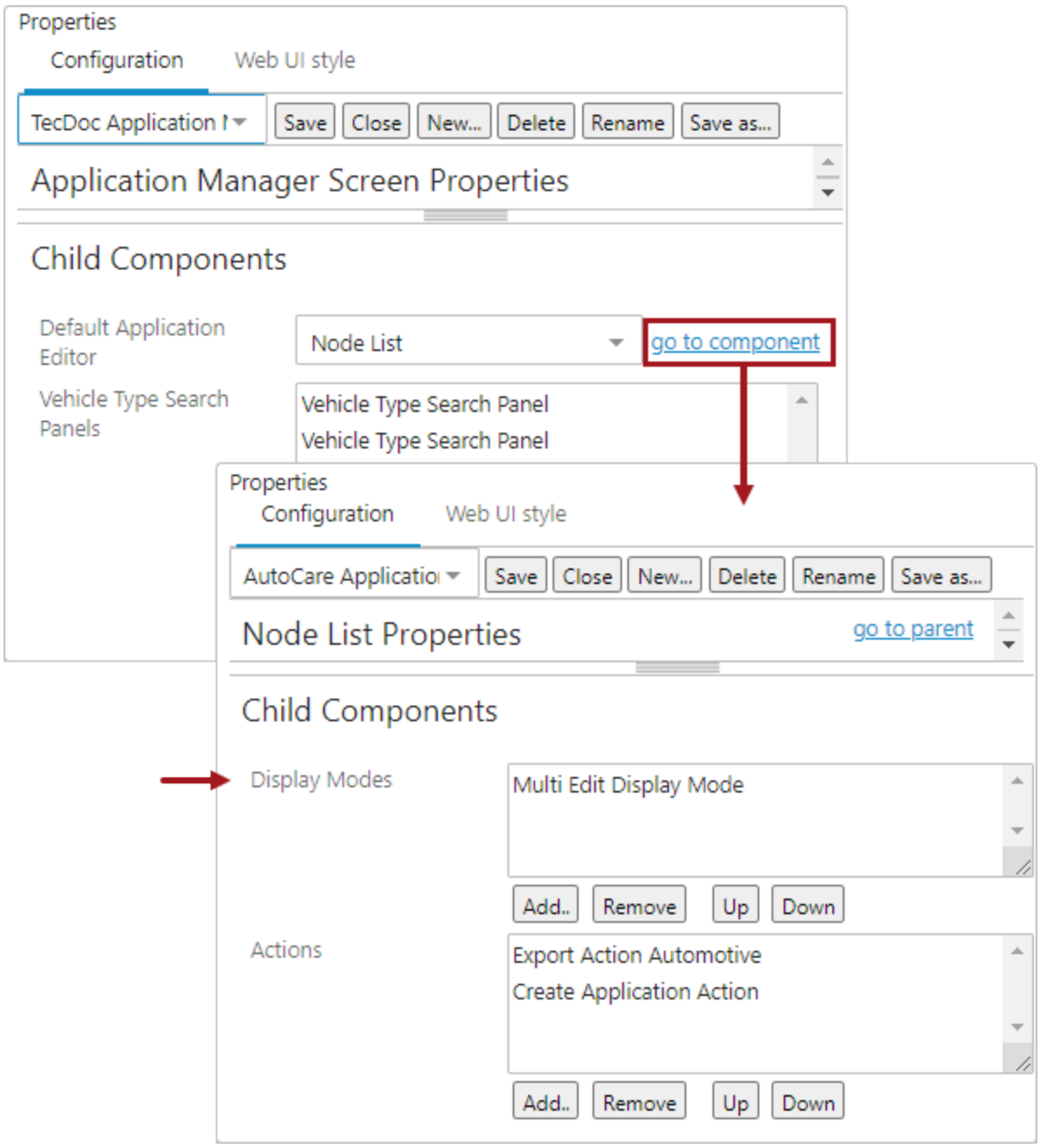
Anyone configuring the Application Criteria Blocks component is expected to be familiar with the Web UI Designer, as basic concepts for working with the designer are not covered in this section. In addition, the user must have appropriate privileges to access the designer. Additional information can be found in the **Designer Access** section of the **Web User Interfaces** documentation.

As this component is configured as a table header, users should be familiar with the process required to add table headers into the Node List component, as basic concepts for accessing and adding the table headers are not covered in this section. For information about accessing and configuring the Node List component, refer to the **Node List Component** topic within the **Web User Interfaces** section of **STEP Online Help**.

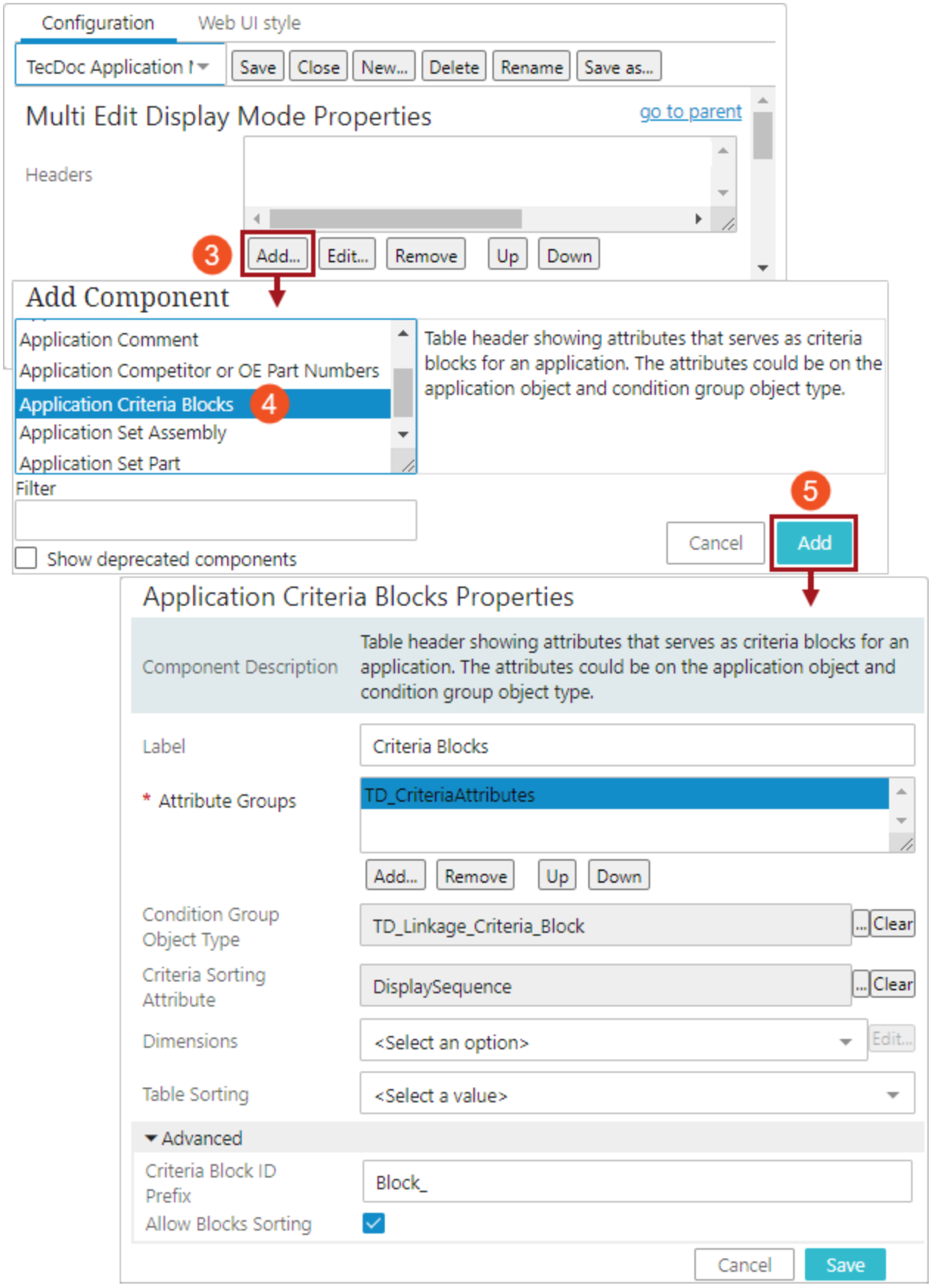
## Configuration Steps

The Application Criteria Blocks component can be added within any of the display modes available in the Node list Properties for the selected Application Manager screen. Follow the steps below to configure the Application Criteria Blocks component.

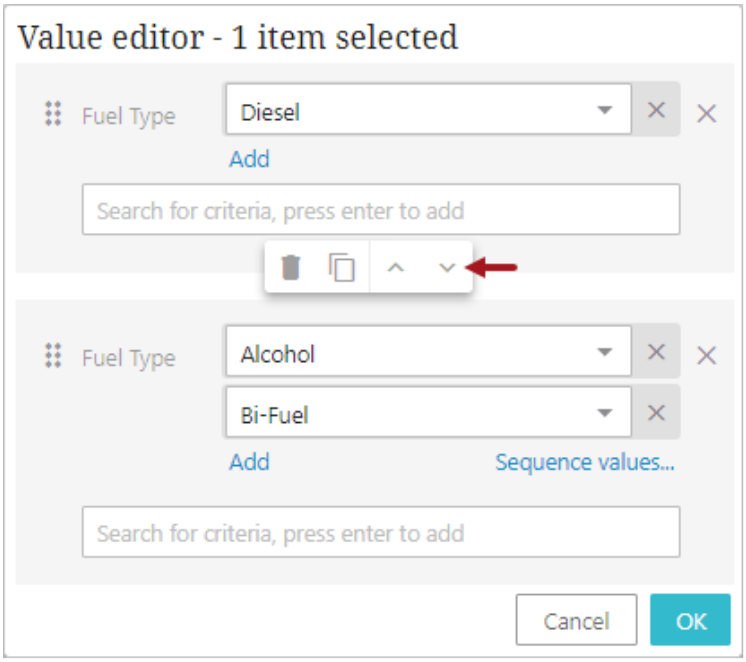
1. Using the Web UI designer as shown in the image below, go to Node List Properties > Display Modes parameter for the Application Manager Screen that needs to display attribute value.

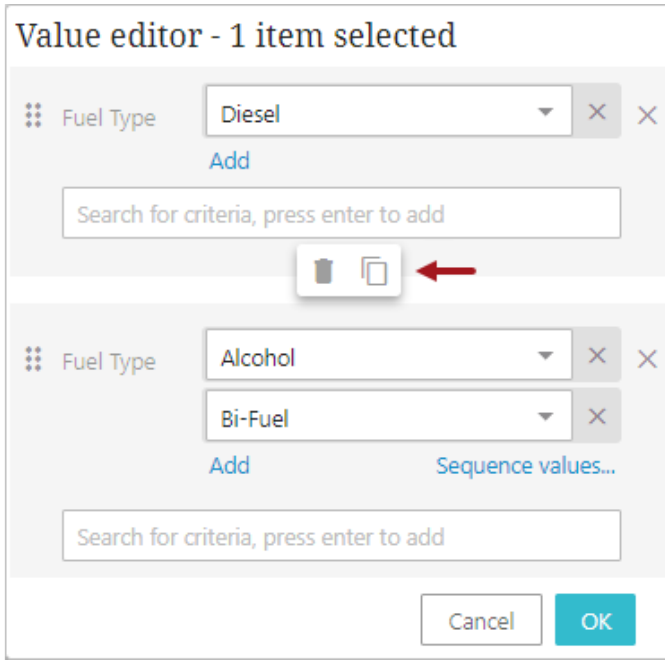



2. Add a display mode (or edit the one that exists) by clicking the **Add** button available next to the Display Modes parameter.
3. Double click to enter the added (or available) display mode, click the **Add** button for the Headers parameter, and the Add Component dialog will display.
4. Find and select the **Application Criteria Blocks** component.
5. Click the **Add** button, and the Application Criteria Blocks Properties designer screen will display (as shown below).


Configuration
Web UI style
TecDoc Application I
Save
Close
New...
Delete
Rename
Save as...
Multi Edit Display Mode Properties
go to parent
Headers
3
Add...
Edit...
Remove
Up
Down
Add Component
Application Comment
Application Competitor or OE Part Numbers
Application Criteria Blocks
4
Application Set Assembly
Application Set Part
Table header showing attributes that serves as criteria blocks for an application. The attributes could be on the application object and condition group object type.
Filter
Show deprecated components
5
Cancel
Add
Application Criteria Blocks Properties
Component Description
Table header showing attributes that serves as criteria blocks for an application. The attributes could be on the application object and condition group object type.
Label
Criteria Blocks
* Attribute Groups
TD_CriteriaAttributes
Add...
Remove
Up
Down
Condition Group Object Type
TD_Linkage_Criteria_Block
Clear
Criteria Sorting Attribute
DisplaySequence
Clear
Dimensions
<Select an option>
Edit...
Table Sorting
<Select a value>
Advanced
Criteria Block ID Prefix
Block_
Allow Blocks Sorting
Cancel
Save

6. Populate the parameters listed below:
   - **Label:** By default, the parameter is populated to display 'Criteria Blocks' as the column header. Optionally, enter the desired label to be displayed as the column header within the Application Manager result's table.
   - **Attribute Groups:** This parameter allows users to add multiple attribute groups. Select one or more attribute groups that hold the criteria attributes. All attributes in the selected group will be evaluated, and any that are populated on the selected application records and its criteria blocks will be displayed as the cell's content. Attributes that are not part of the selected group(s) cannot be added / will not be listed in the column. Further, defining new criteria attribute value using the 'Value editor' dialog will link the criteria attributes to the application records / criteria blocks.
   - **Condition Group Object Type:** This parameter must be configured with the object type of the criteria blocks. Leaving this parameter empty *will not* allow users to add a new criteria block in the cell.
   - **Criteria Sorting Attribute:** This parameter must be populated with the metadata attribute that determines the order of display of the attribute values within the cell. This metadata attribute only determines the order of display of criteria attributes and *not* the criteria blocks within the cell.
   - **Dimensions:** By default, the parameter is blank. Optionally, to configure the column width and height, select 'Table Header Dimensions.'
   - **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. Scroll down to the bottom of the designer window and expand the Advanced flipper to populate the parameters listed below:
   - **Criteria Block ID:** By default, this mandatory parameter is populated with the following prefix: 'Block_.' This prefix will be used to ID any criteria blocks created using this table header component. This must be a unique prefix. Optionally, if a different prefix is desired, edit the prefix to something more suitable but unique.
   - **Allow Blocks Sorting:** This checkbox determines if the rearranging of the criteria blocks within the 'Value editor' dialog should be allowed or not. Selecting the checkbox will display two rearranging arrow icons when the user hovers over the lower gray area of the criteria block (as shown on the left side of the image below). Deselecting the checkbox will no longer display the rearranging arrow icons (as shown on the right side of the image below).

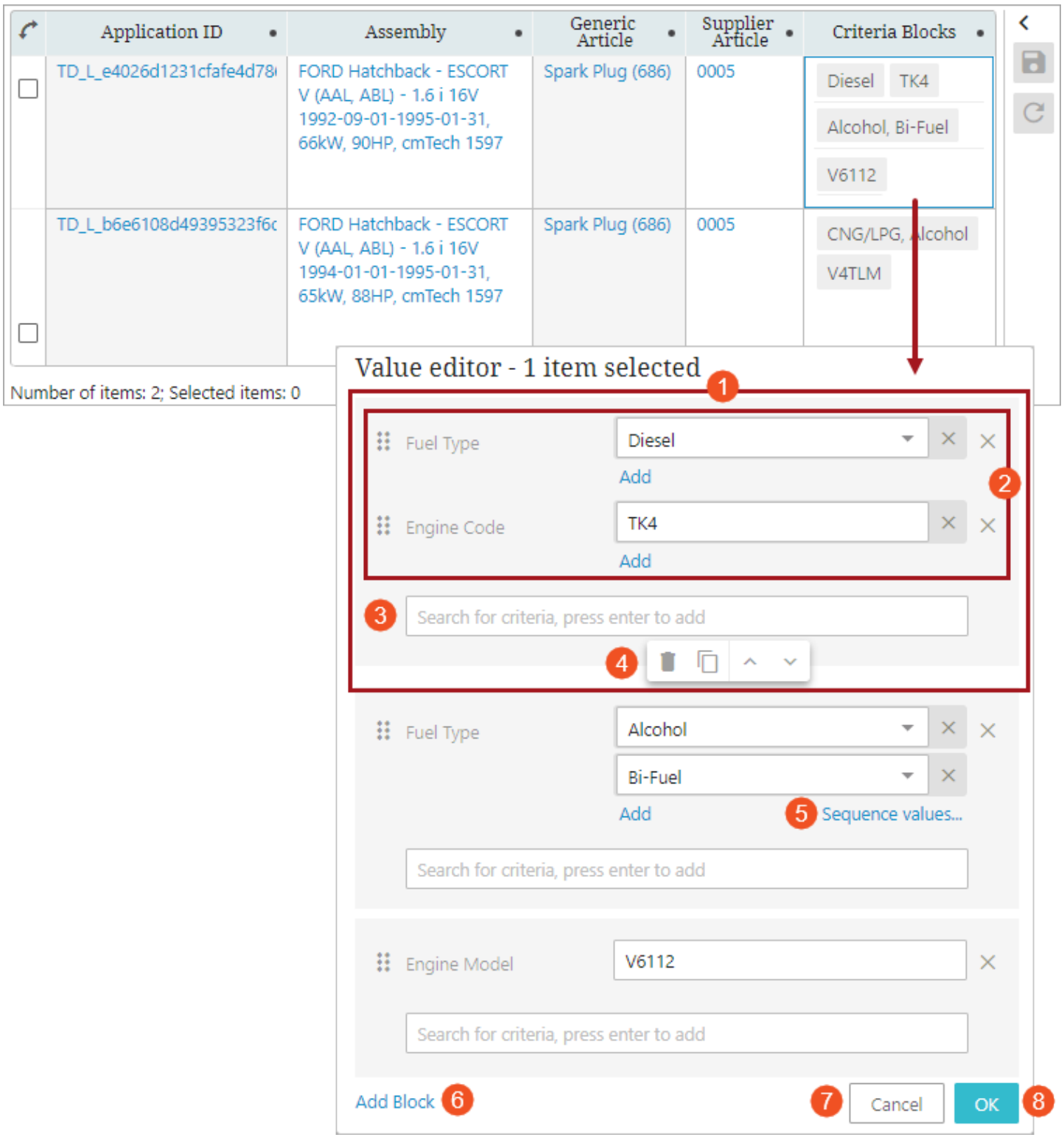



8. Click the **Add** button (or **Save** button if editing an existing component) to save the newly added component to the Headers parameter.
9. Optionally, on the Node List Properties Headers parameter, use the **Up** and/or **Down** buttons to configure the order in which the newly added column should display within the Results Table.
10. Click the **Save** and **Close** buttons for the designer.

#### Using Application Criteria Blocks Value Editor

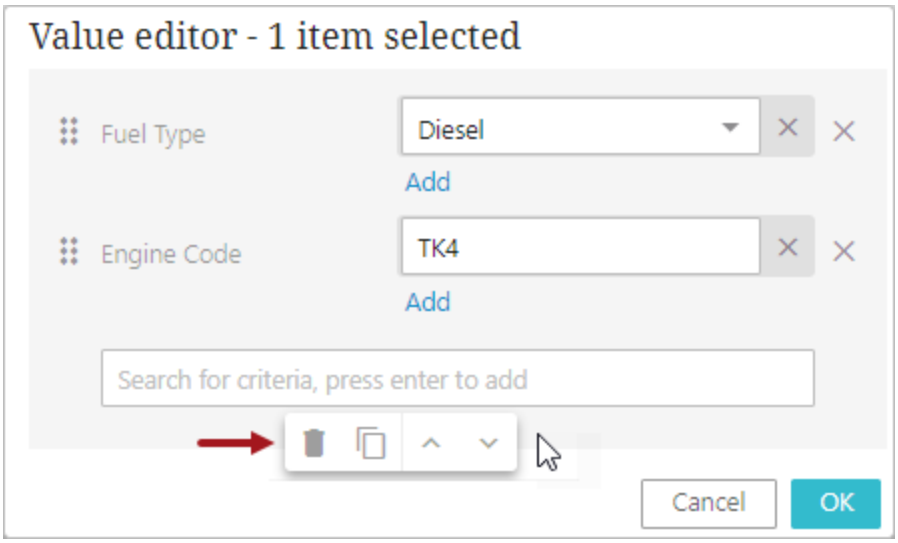
The cells of the configured Application Criteria Blocks column will display the value populated in the criteria attributes for each application record displayed within the Application Manager results table. Double click on the cell to access the value editor. Users can add, edit, or remove criteria attributes / criteria blocks for the selected application record in the value editor.




The value editor for the Application Block Criteria column is composed of the following:

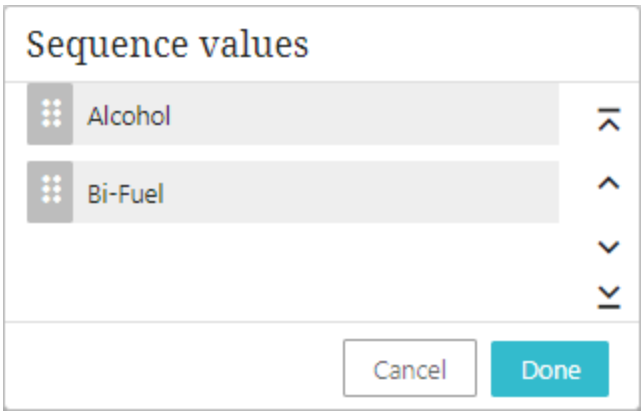
1. **Criteria block canvas:** Each criteria block stored as an individual child object under the application record is depicted as a gray canvas within the value editor. Each canvas representing criteria blocks comprises a set of criteria attribute(s).

2. **Criteria attribute:** Only those criteria attributes with values populated for the criteria block / application record is displayed in the value editor. Also, the criteria attributes that are not part of the attribute group(s) configured within the Attribute Groups parameter will not be listed in the canvas. Further, users can do the following on the displayed attributes:
   - Rearrange the display order for the attributes listed in the criteria blocks using the Move icon ( ). Users need to click and drag the Move icon ( ) to rearrange the order of display. Rearranging the order will change the display sequence value in the metadata attribute which is configured within the Criteria Sorting Attribute parameter.
   - Use the 'Add' link to add multiple values for the same criteria attribute. The 'Add' link is visible only if it is a multivalued attribute.
   - Use the Remove icon ( ) to remove the attribute values populated in the criteria attribute.
3. **Typeahead field:** This field allows users to add more criteria attributes for a criteria block. Typing text into this typeahead field will display a list of attributes that begin with the provided text. Adding a new attribute and entering a value to it will link the attribute to the criteria block. Attributes that are not part of the attribute group (s) configured within the Attribute Groups parameter of design mode cannot be added to the canvas.
4. **Additional icons:** Hovering over the lower gray area of the criteria block canvas will display an icon set (as shown below) consisting of four icons that allow the following:




   - **Delete icon ( ):** This icon deletes the criteria block object linked to the application record. The deleted criteria block will be moved to the Recycle Bin in the system.
   - **Duplicate icon ( ):** This icon allows users to duplicate the criteria block.
   - **Rearrange icons ( ):** This icon set consists of an Up and Down arrow that allows users to rearrange the order of display of criteria blocks. Rearranging the order will move the criteria block objects up / down in the Tree hierarchy and renames the criteria blocks based on their order of listings. So, for example, if a criteria block named 'Block_001' is moved to the second-order, then that criteria block will be renamed to 'Block_002,' and the other criteria block that was in the second-order before will be renamed to 'Block_001' as it is listed in the first order. These icons are visible only when the user selects the Allow Blocks Sorting option in the designer.

5. **Sequence Values...:** This link is displayed only when a criteria attribute has multiple values populated in it. Clicking the link will open the 'Sequence values' dialog (as shown below) that allows users to rearrange the order of listings of the attribute values. Users can click and drag the Move icon ( ) or click the Up and Down arrow to rearrange the order of listing.




6. **Add Block:** This link will be visible only when an object type is defined within the Condition Group Object Type parameter in the designer screen. Clicking this link will add an empty criteria block canvas in the value editor with a typeahead field prompting users to add a criteria attribute. If no criteria attribute is added, then a criteria block with no criteria attributes defined will be created in the system for the selected application record.
7. **Cancel button:** Click the Cancel button if the user needs to exit the value editor without saving the changes.
8. **OK button:** Click the OK button to save the changes made in the value editor.

**Important:** After making edits in the value editor for an application record, it is required to save the changes in the Application Manager Screen by clicking the Save button available in the right of the Application Manager result table. Failing to save the changes and navigating out of the Application Manager Screen may lose those changes.

### Application Set Assembly Table Header

When configured within an Application Manager Results Table, the 'Application Set Assembly' component can display the title of the assemblies / vehicles (and its attribute value) linked to the applications in the results table. The assembly / vehicle name is listed in a column against each result in the Application Manager screen. In addition, users can also change the assembly / vehicle referenced to the application record. Further, users can click on the cell content (assembly / vehicle name) to navigate to the screen that is mapped to display the assembly / vehicle object.

## Prerequisites

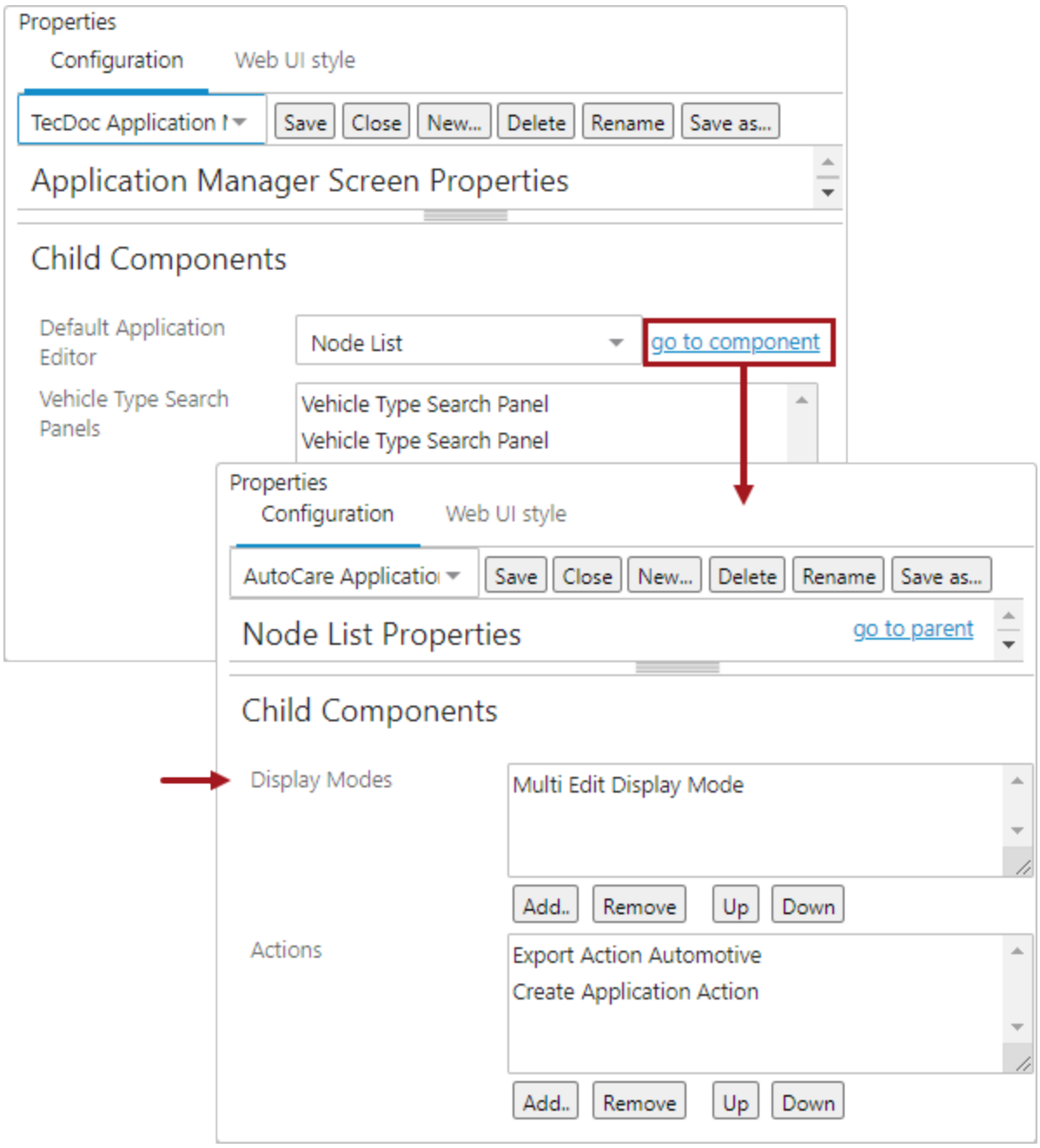
Anyone configuring the Application Set Assembly component is expected to be familiar with the Web UI Designer, as basic concepts for working with the designer are not covered in this section. In addition, the user must have appropriate privileges to access the designer. Additional information can be found in the **Designer Access** section of the **Web User Interfaces** documentation.

As this component is configured as a table header, users should be familiar with the process required to add table headers into the Node List component, as basic concepts for accessing and adding the table headers are not covered in this section. For information about accessing and configuring the Node List component, refer to the **Node List Component** topic within the **Web User Interfaces** section of **STEP Online Help**.

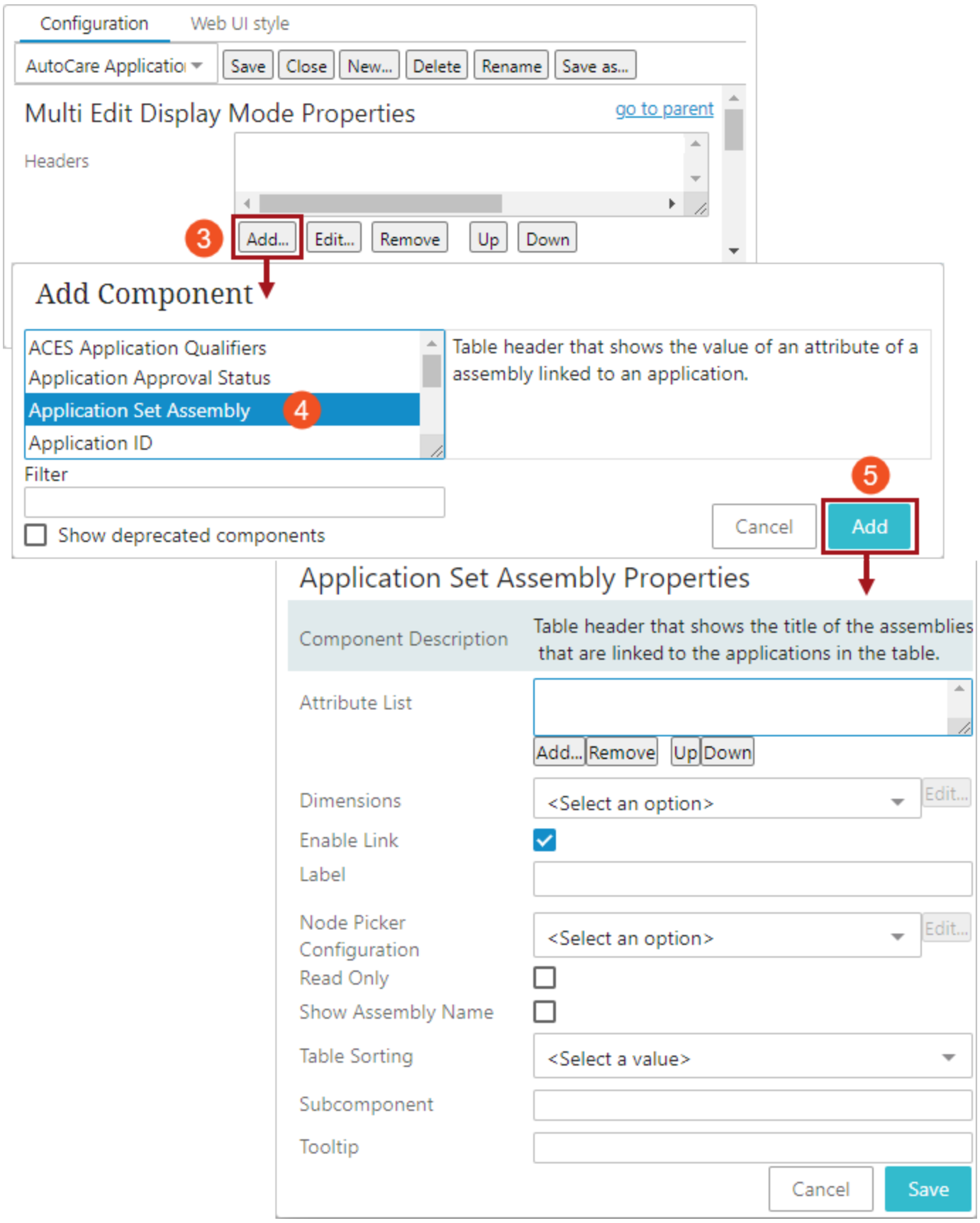
## Configuration Steps

The Application Set Assembly component is added within any of the display modes available in the Node list Properties for the selected Application Manager screen.Follow the steps below to configure the Application Set Assembly component.

1. Using the Web UI designer as shown in the image below, go to Node List Properties > Display Modes parameter for the Application Manager Screen that needs to display assembly / vehicle name.

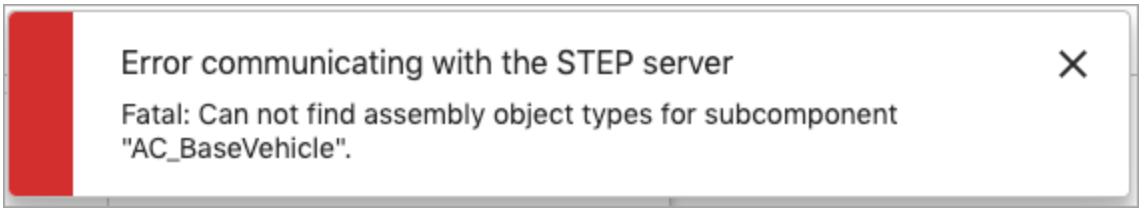



2. Add a display mode (or edit the one that exists) by clicking the **Add** button available next to the Display Modes parameter.
3. Double click to enter the added (or available) display mode, click the **Add** button for the Headers parameter, and the Add Component dialog will display.
4. Find and select the **Application Set Assembly** component.
5. Click the **Add** button, and the Application Set Assembly Properties designer screen will display (as shown below).


Configuration
Web UI style
AutoCare Applicatio
Save
Close
New...
Delete
Rename
Save as...
Multi Edit Display Mode Properties
go to parent
Headers
3
Add...
Edit...
Remove
Up
Down
Add Component
ACES Application Qualifiers
Application Approval Status
Application Set Assembly
4
Application ID
Table header that shows the value of an attribute of a assembly linked to an application.
Filter
Show deprecated components
5
Cancel
Add
Application Set Assembly Properties
Component Description
Table header that shows the title of the assemblies that are linked to the applications in the table.
Attribute List
Add...
Remove
Up
Down
Dimensions
<Select an option>
Edit...
Enable Link
Label
Node Picker Configuration
<Select an option>
Edit...
Read Only
Show Assembly Name
Table Sorting
<Select a value>
Subcomponent
Tooltip
Cancel
Save

6. Populate the parameters listed below:
   - **Attribute List:** This parameter can be populated with multiple attributes that are valid for the assembly / vehicle object. The attributes listed within this parameter will be displayed as the cell content along with the name of the assembly / vehicle.
   - **Dimensions:** By default, the parameter is blank. Optionally, to configure the column width and height, select 'Table Header Dimensions.'
   - **Enable Link:** This checkbox parameter allows users to choose if they want the cell value to be displayed as a hyperlink or as plain text. Selecting the Enable Link parameter will enable the values to display as a hyperlink. Clicking the hyperlink will navigate to the screen that is mapped to display the assembly / vehicle object.
   - **Label:** By default, the parameter is blank, but after the **Add** button is clicked, the default column header label (Vehicle) will be stored. Optionally, enter the desired label to be displayed as the column header within the Application Manager result's table.
   - **Node Picker Configuration:** In the dropdown next to the Node Picker Configuration parameter, select the Node Picker Dialog option and click **Edit**. The 'Edit component' for the 'Node Picker Dialog Properties' dialog will display. This parameter allows users to define root nodes, search tab pages, and a custom title. The Node Picker icon ( ) will be displayed within the cell next to the type ahead field. For details, refer to the **Node Picker Dialog** documentation within the **Web User Interfaces** section of **STEP Online Help**.
   - **Read Only:** Determines if the value field should be editable or read only.
     - If disabled (default), then users can edit the cell.
     - If enabled, the assembly / vehicle name on the screen will be locked so users cannot make edits.
   - **Show Assembly Name:** Determines if the cell has to display the assembly / vehicle name as its content.
     - If enabled (default), the assembly / vehicle name will be displayed ahead of the attribute value.
   - **Table Sorting:** By default, the display parameter is blank. Optionally, select from the dropdown values to specify the default sorting order of the column data.
   - **Subcomponent:** This parameter helps to restrict the display of unwanted assembly / vehicle and allows only the assembly / vehicle of the required object type to be displayed in the typeahead field of the table header cell. Users should populate the ID of the *parent object type* of the required vehicle / assemblies, so that the vehicle / assemblies of the object types that are not mentioned in this parameter will be excluded from being displayed. The assembly / vehicle object types are determined in the Automotive component model.

     When the user types in the assembly / vehicle of the object type ID that is not included in the Subcomponent parameter, then an error message will be displayed as shown below.

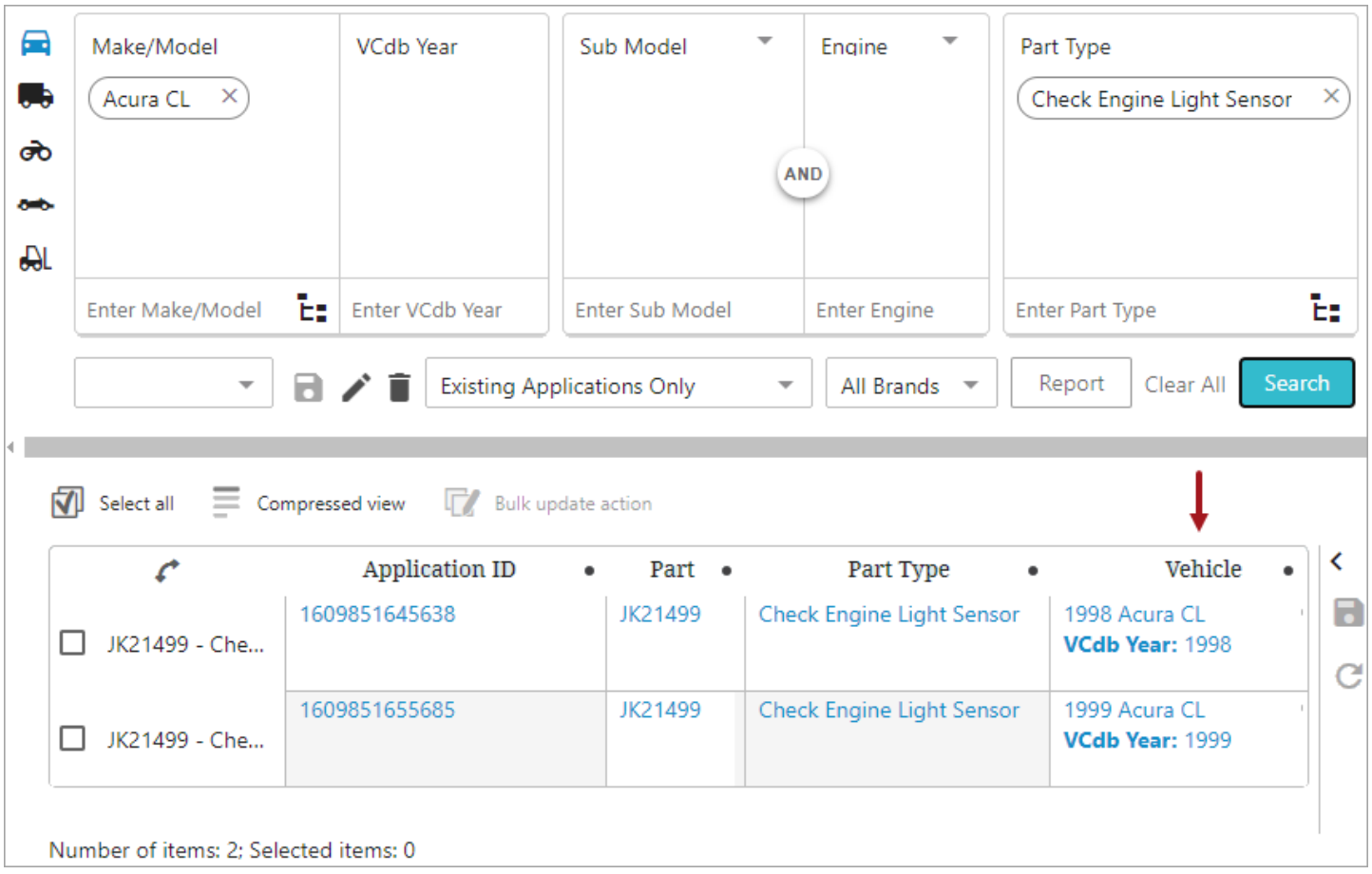


**Note:** The above method of filtering the assembly / vehicle only works when the user uses the typeahead field to access the assembly / vehicle. The filtering does not applies if the users chooses to access the assembly / vehicle via the Node Picker icon ( ).

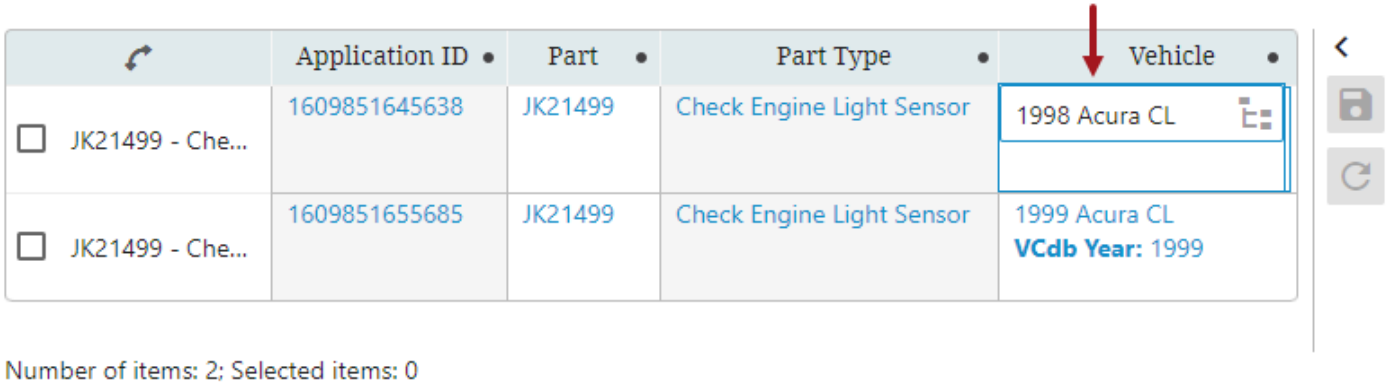
- **Tooltip:** By default, the parameter is blank. Optionally, enter what the tooltip has to say when the cursor is placed over the cell content.

7. Click the **Add** button (or **Save** button if editing an existing component) 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.

When a search is performed in the Application Manager screen, the Application Manager Results Table will display the application records with the assembly / vehicle and the configured attribute value as shown below.




When users click on the cell, a Node Picker icon ( ) beside the typeahead field will be displayed, as shown below. Further, users can either click on the Node Picker icon ( ) or use the typeahead field to edit the assembly / vehicle.




### Suggested Part Number Table Header

When configured within an Application Manager results table, the 'Suggested Part Number' component can be used to display the potential parts fitting the missing application. The suggested parts are listed in a column against each result in the Application Manager screen. Users can double click on the cell to open up the Value Editor window and can add the suggested parts to the missing application.

## Prerequisites

It is expected that anyone configuring the Suggested Part Number 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** documentation.

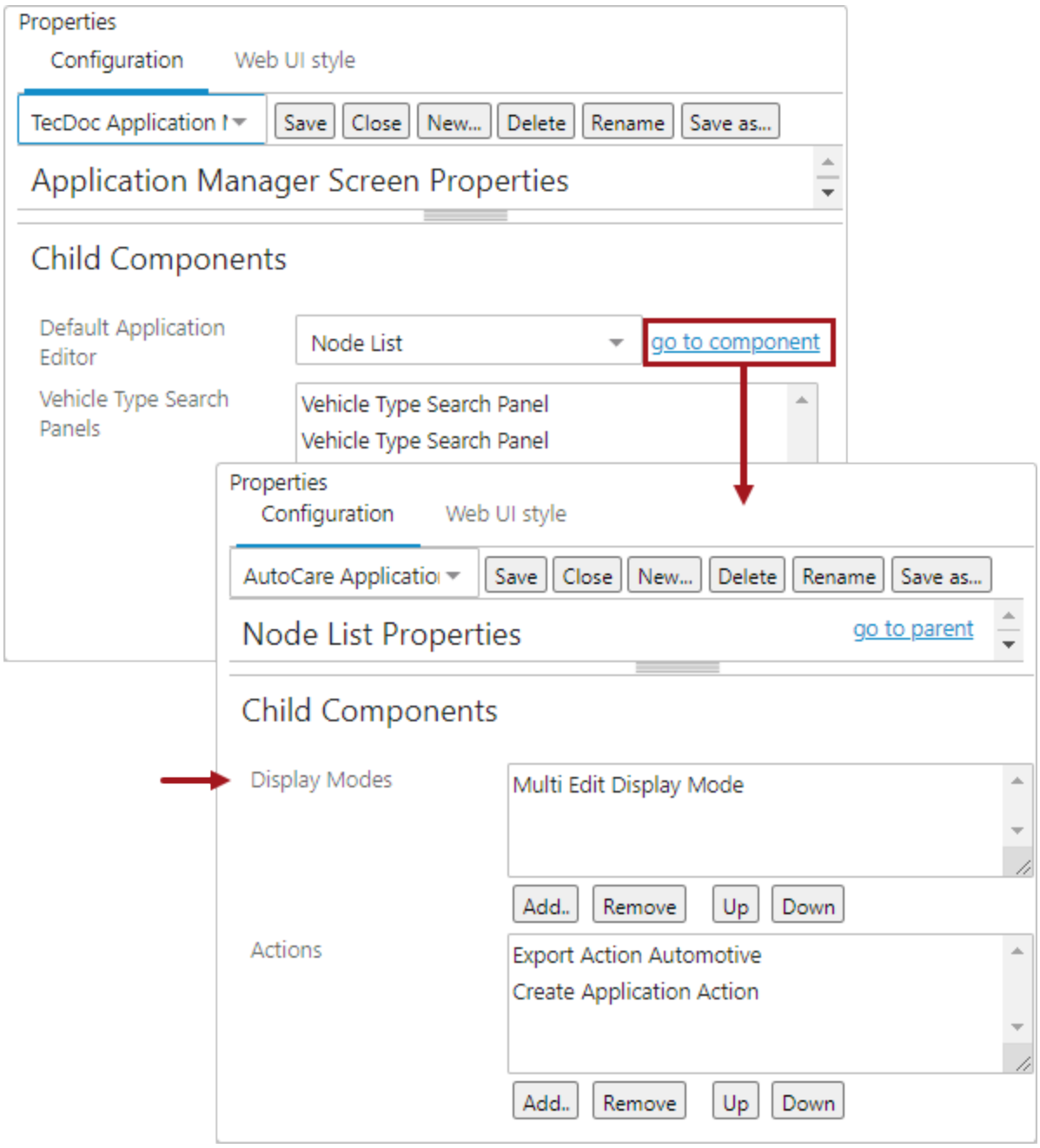
For information about accessing and configuring the Node List component, refer to the **Node List Component** topic within the **Web User Interfaces** section of **STEP Online Help**.

For information about how the Application Manager results table displays the missing application for a search criteria, refer to the **Missing Application Coverage Functionality** topic within the **Application Manager** section of this guide.

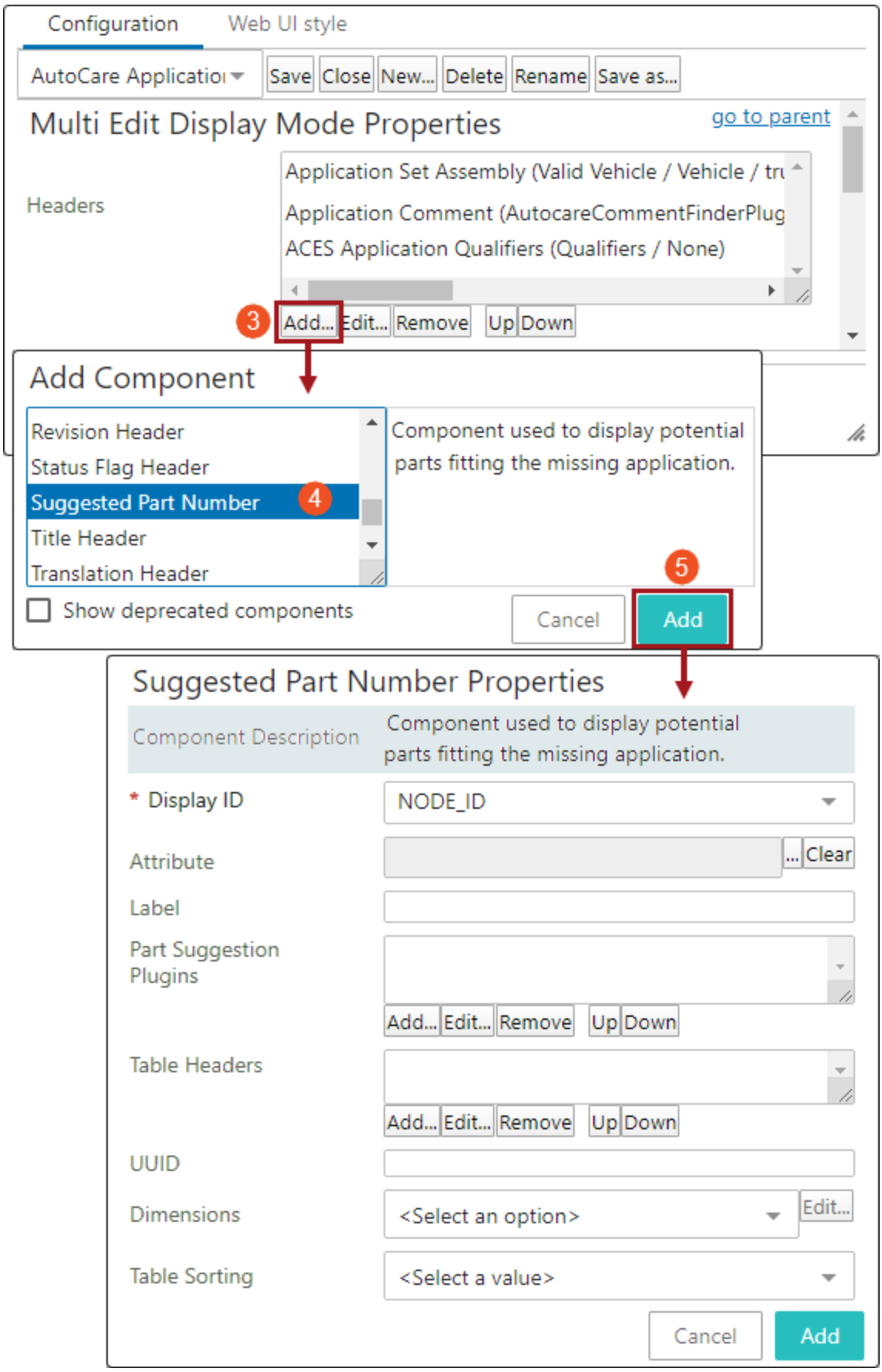
## Configuration Steps

The Suggested Part Number component is added within any of the display modes available in the Node List Properties for the selected Application Manager screen.Follow the steps below to configure the Suggested Part Number component.

1. Using the Web UI designer as shown in the image below, go to Node List Properties > Display Modes parameter for the Application Manager Screen that needs to display potential part number(s).




2. Add a display mode (or edit the one that exists) by clicking the **Add** button available next to the Display Modes parameter.
3. Double click to enter the added (or available) display mode, click the **Add** button for the Headers parameter, and the Add Component dialog will display.
4. Find and select the **Suggested Part Number** component.
5. Click the **Add** button, and the Suggested Part Number Properties designer screen will display (as shown below).


Configuration
Web UI style
AutoCare Applicatio
Save
Close
New...
Delete
Rename
Save as...
Multi Edit Display Mode Properties
go to parent
Headers
Application Set Assembly (Valid Vehicle / Vehicle / tru
Application Comment (AutocareCommentFinderPlug
ACES Application Qualifiers (Qualifiers / None)
3
Add...
Edit...
Remove
Up
Down
Add Component
Revision Header
Status Flag Header
Suggested Part Number
4
Title Header
Translation Header
Component used to display potential parts fitting the missing application.
5
Show deprecated components
Cancel
Add
Suggested Part Number Properties
Component Description
Component used to display potential parts fitting the missing application.
* Display ID
NODE_ID
Attribute
...
Clear
Label
Part Suggestion Plugins
Add...
Edit...
Remove
Up
Down
Table Headers
Add...
Edit...
Remove
Up
Down
UUID
Dimensions
<Select an option>
Edit...
Table Sorting
<Select a value>
Cancel
Add

6. Populate the parameters listed below:
   - **Display ID:** Determines if the ID, Name, or attribute value of the suggested part number to be displayed within the table cell. By default, the parameter uses the 'NODE_ID' value that allows the ID of the suggested part number to display within the cell. Optionally, the 'NAME_VALUE' or 'ATTRIBUTE_VALUE' value can be selected using the dropdown. While the 'NAME_VALUE' displays the Name of the suggested part number, the 'ATTRIBUTE_VALUE' allows the value of the attribute selected within the 'Attribute' parameter to display within the cell.
   - **Attribute:** This parameter must be populated with an attribute that is valid for the suggested part number.

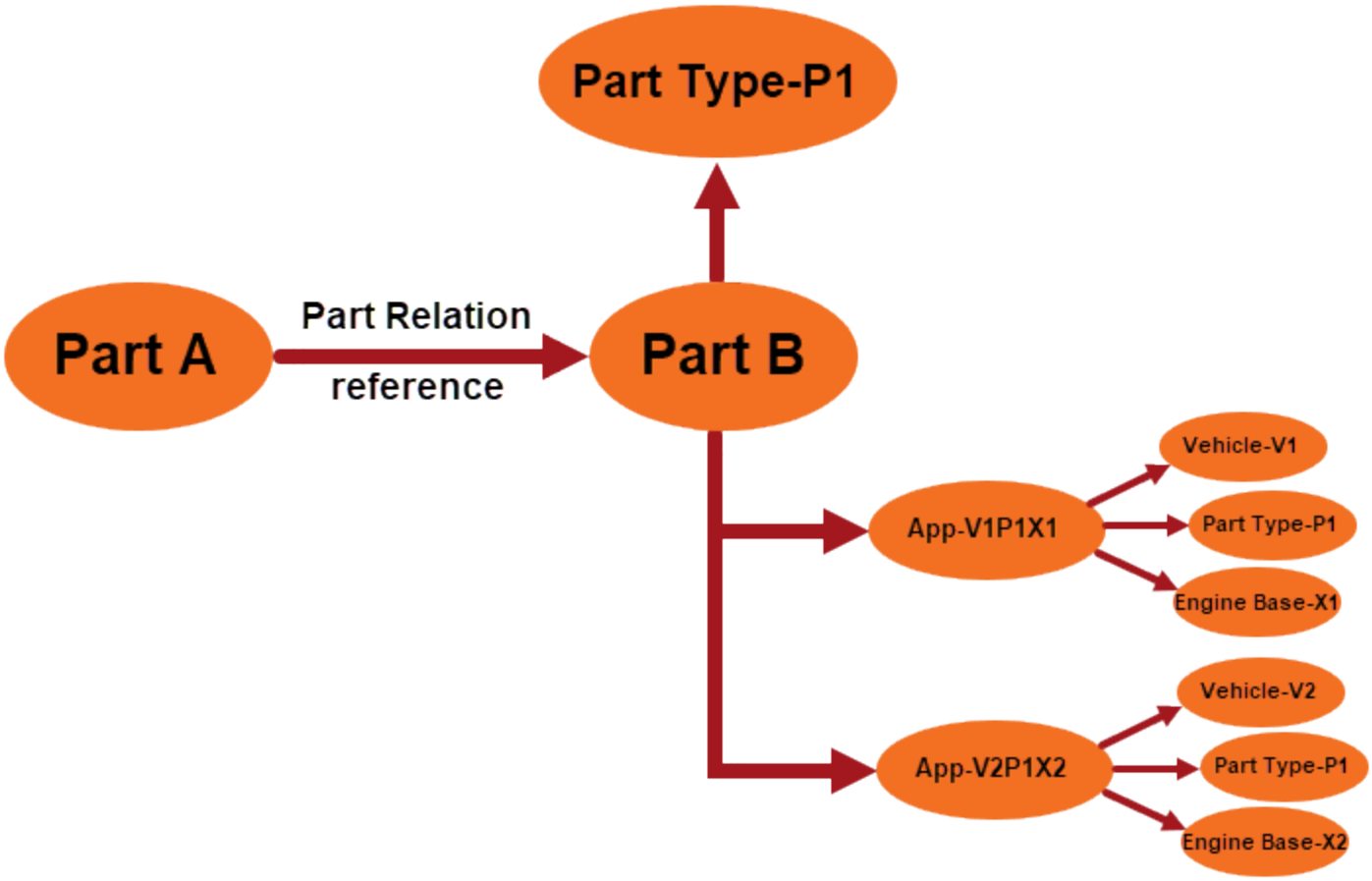
     **Note:** The Attribute parameter should only be populated when the 'ATTRIBUTE_VALUE' value is selected for Display ID. Otherwise, the defined attribute value will not be displayed in the cell.

   - **Label:** By default, the parameter populated with 'Suggested Part Number.' Optionally, enter the desired label to be displayed as the column header within the Application Manager result's table.
   - **Part Suggestion Plugins:** Currently, this parameter only allows the 'Referenced Parts Suggestion Plugin' to be configured. This plugin suggests the part number that is referencing the current part which in turn has similar application records defined in the Application Manager search criteria. Clicking the Add button will directly display the Referenced Parts Suggestion Plugin Properties designer screen where the user has to add at least one reference type. The configured reference types must be the one that establishes a relation between the suggesting part number and the current part number. When the Application Manager performs a search for missing application records, the 'Referenced Part Suggestion Plugin' configured in the Suggested Part Number table header looks for the part that is already applicated to the defined vehicle and the part type. Thereafter, it follows through the configured reference to find the source object of that part to display it as a suggestion for the missing application records.

     Below is an illustration of how the parts are suggested using this plugin.

     To understand it better, it is necessary that the user knows about how the missing applications are calculated and displayed in the search results. For more information refer to the **Missing Application Coverage Functionality** topic within the **Application Manager** section of this guide.

     In the example chart below, App-V1P1X1 and App-V2P1X2 are the existing application records for Part B (notice that the Part B is applicated to Part Type-P1 and also applicated to Vehicle V1 and V2).




When the user searches for missing application records based on the vehicle (V1) and the part type (P1), the Part Suggestion Plugin looks for the part (in this case it will be Part B) that is already applicated to the defined vehicle (V1) and the part type (P1), and then follows to find the source of the reference (Part Relation) to display Part A as a suggestion to all missing application records.

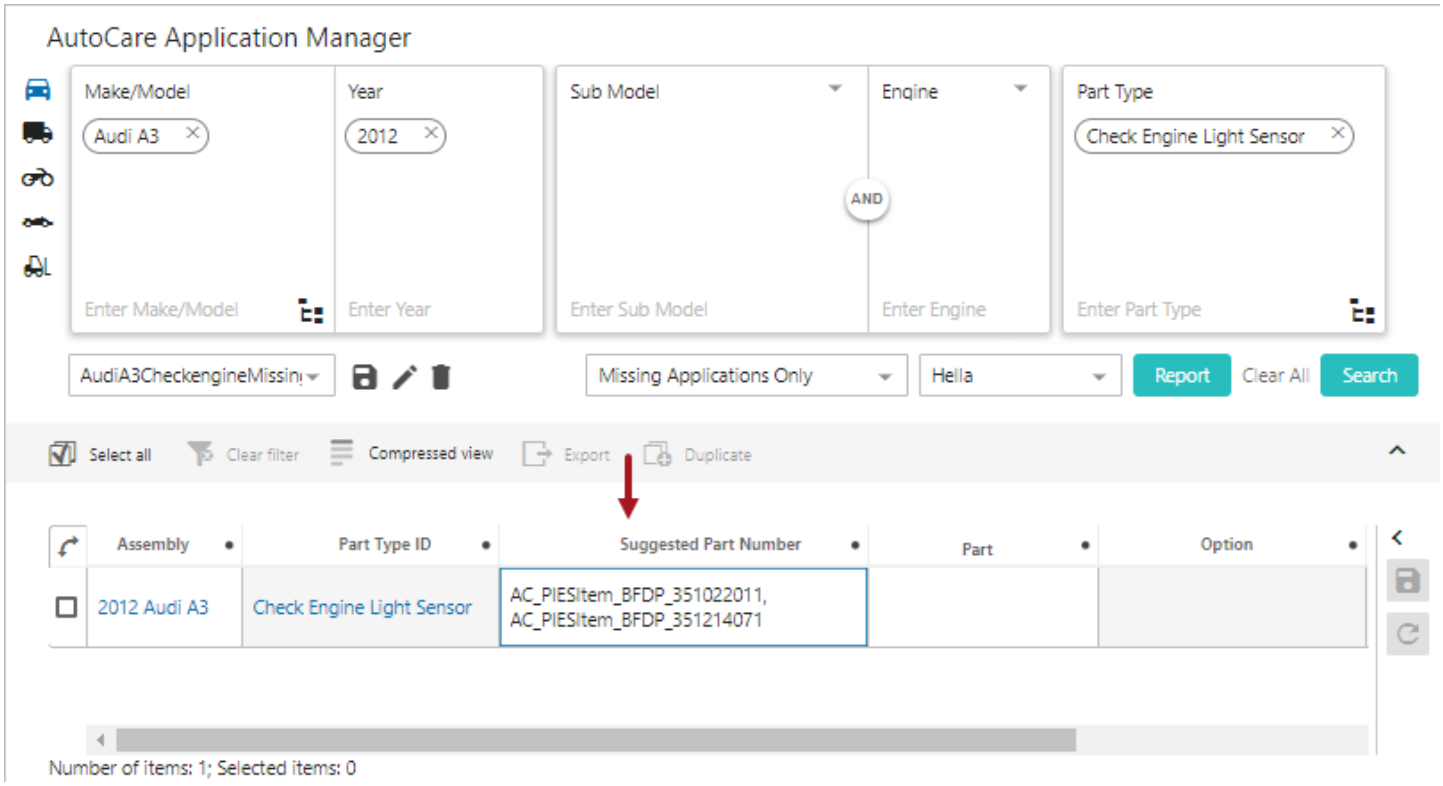
In this scenario, since the Part A is referenced to Part B by the reference type (Part Relation) configured within the Part Suggestion Plugins, Part A will be displayed as suggested part number.

- **Table Headers:** By default, no table headers are listed in this parameter. It is required to add at least the ID and the Name headers within this parameter. Table headers configured in this parameter display the associating information of the suggested part number when the user accesses the Value Editor dialog by double-clicking the suggested part number cell.
- **Dimensions:** By default, the parameter is blank. Optionally, to configure column width and height, select 'Table Header Dimensions.'
- **Table Sorting:** By default, the parameter is blank. Optionally, select from the dropdown values to specify the default sorting order of the column data.

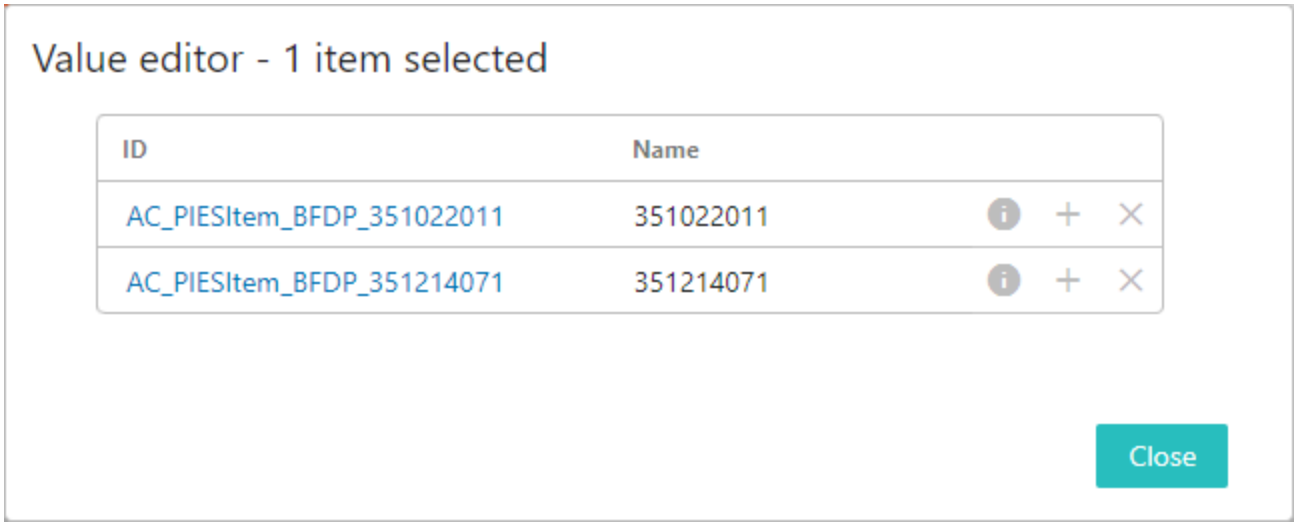
7. Click the **Add** button (or **Save** button if editing an existing component) to save the newly added component to the Headers parameter.

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.

When a search is performed in the Application Manager screen, the Application Manager results table will display the missing application records with the suggested part number(s) for application as shown below.



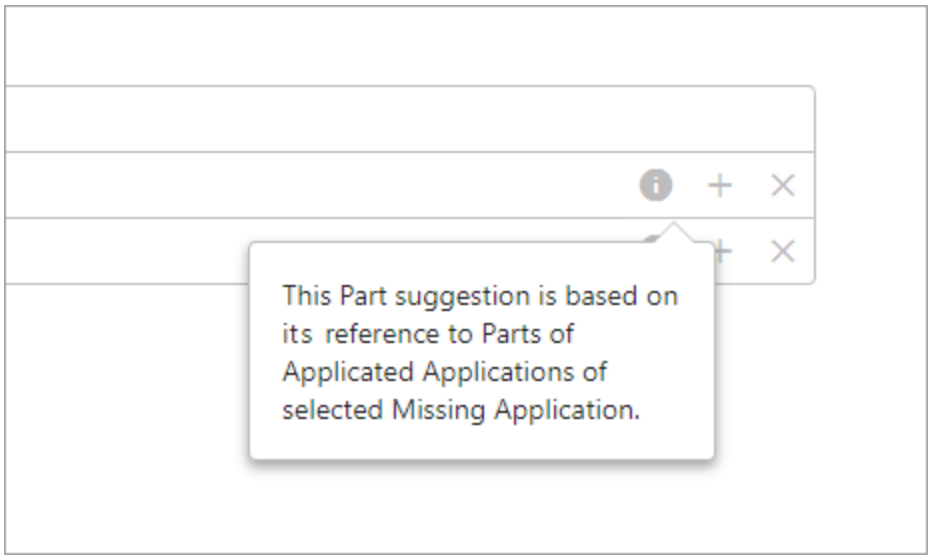

When users click on the cell, the Value Editor window listing all the suggested part numbers will be displayed, as shown below.



Clicking on the add button (+) that is available alongside each part number will applicate the missing application to the suggested part number.

Clicking the remove button (×) that is available alongside each part number will remove the suggested part number from the list.

Clicking the info button ( ) opens a dialog explaining the reason for suggesting that part number.



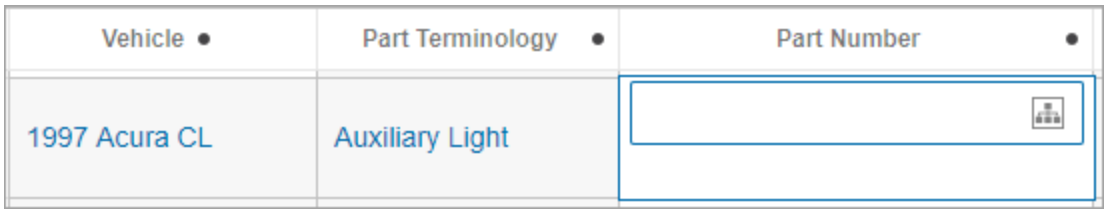
## Controlling Display of Conditions in Application Manager

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, refer to the **Application Condition Header Component** topic.

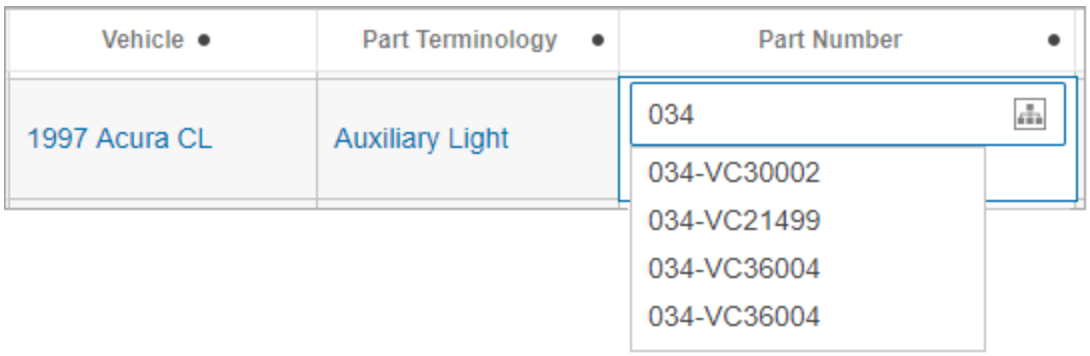
## Creating Applications

Creating an application for a vehicle can be done within the results table.

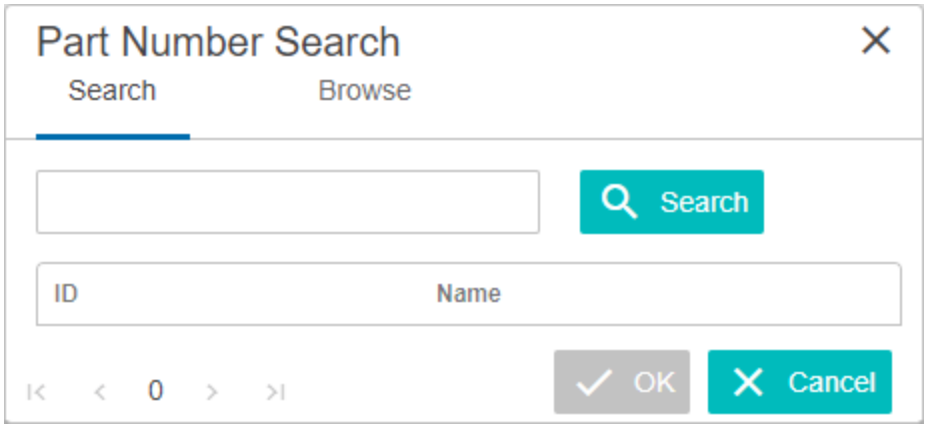
Once a vehicle is listed within the results table, a user can create an application by clicking into the Part Number cell, and providing a part number for the vehicle. In the example below the Part Number cell was blank, upon double clicking within the cell, the gray borders of the cell change to blue, indicating it is active and ready to be edited, and a node selector icon displays.



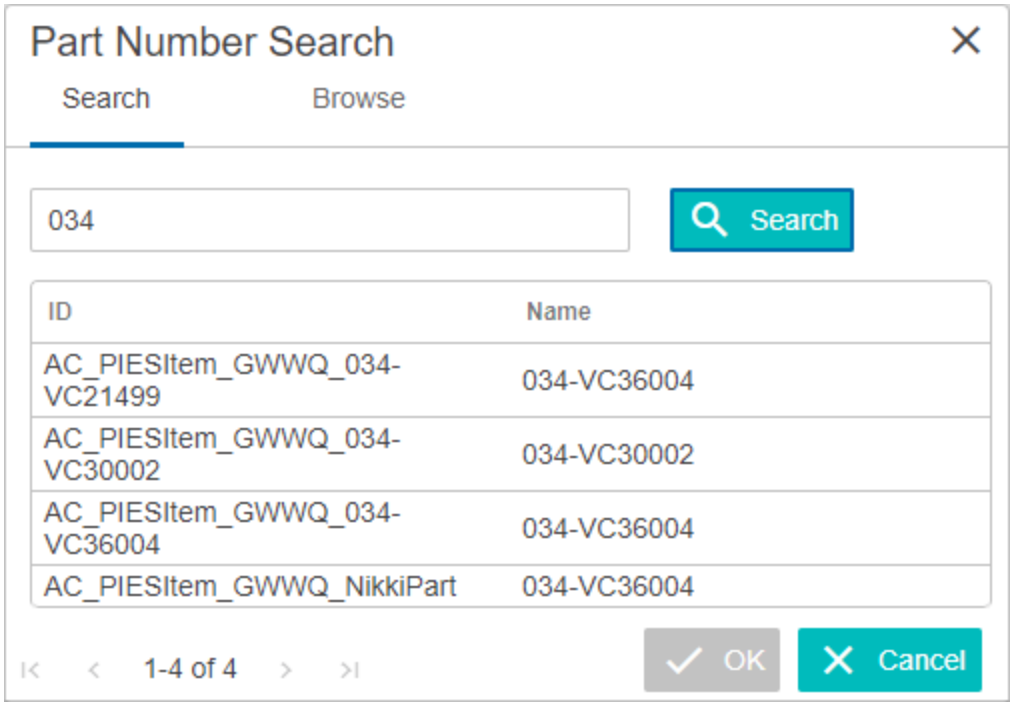

Once the Part Number cell is active, users can type in the necessary part number. As the Part Number is typed in, the typeahead dropdown will display a list of potential part number matches. Users can use the down arrow key to navigate from the field down into the list, and then use the Enter key to populate the field with the part number. Optionally, the mouse pointer can be used to click on the valid part number.




Optionally, the node browser icon can be clicked, and the Part Number Search dialog will display (as shown below).



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.



## 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, applicating 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:

- Identifying the application and following the ACES Application To Base Vehicle reference (AC_ACESApplicationToBaseVehicle) to the Base Vehicle object type (AC_BaseVehicle).
- Examining all vehicle children (AC Vehicle) of the Base Vehicle.
- Following the Vehicle To Engine Config reference (AC_VehicleToEngineConfig) on the vehicle to the target object.
- 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:

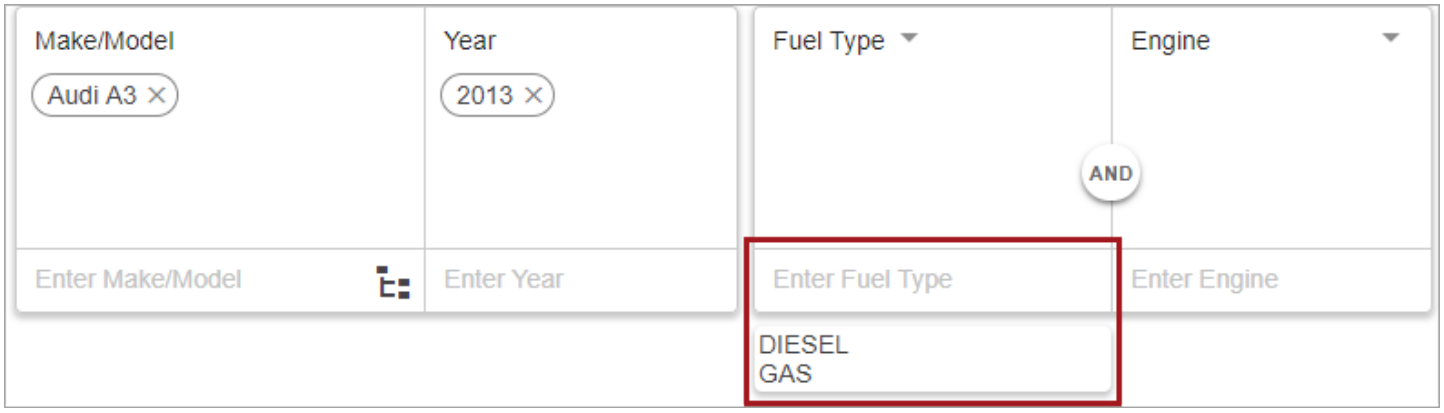
- **Validation Path Functionality and Search Box Criteria**
- **Validation Path Functionality and Options on Applications**
- **Validation Path Functionality and Missing Applications**
- **Validation Path Functionality and Advanced Filtering**
- **Workbench Configurations Related to Validation Path**

## Validation Path Functionality and Search Box Criteria

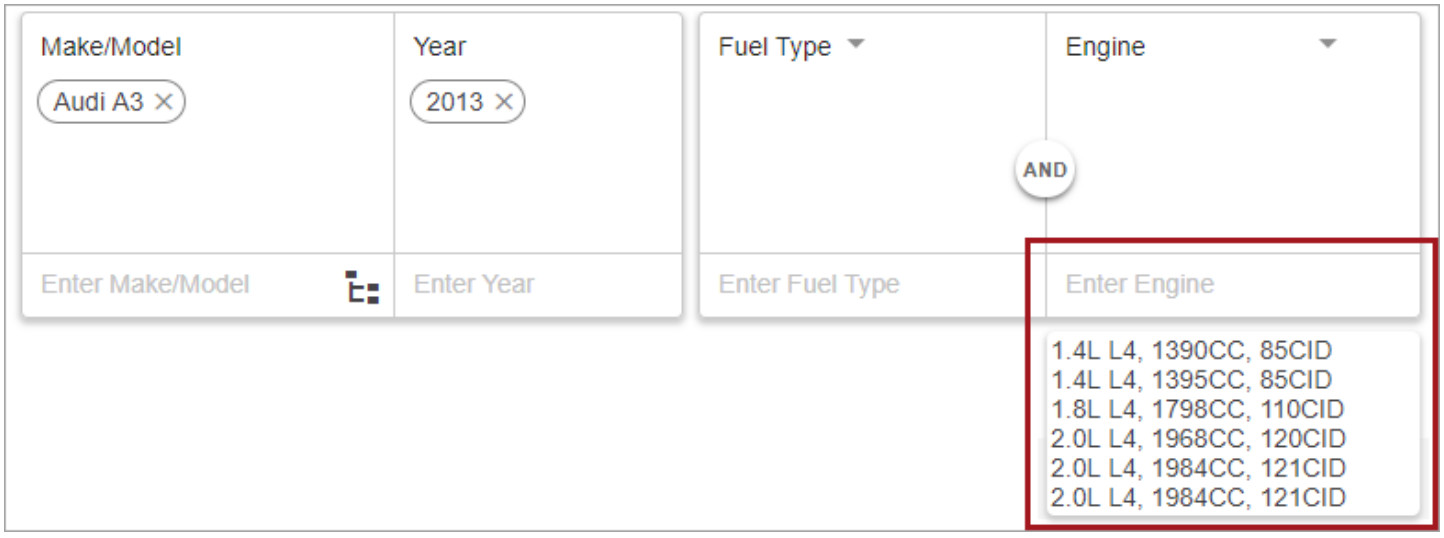
The validation path concept drives the intelligence that allows Application Manager search boxes to display only valid options filtered by previous selections and/or specific vehicle configurations.

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).

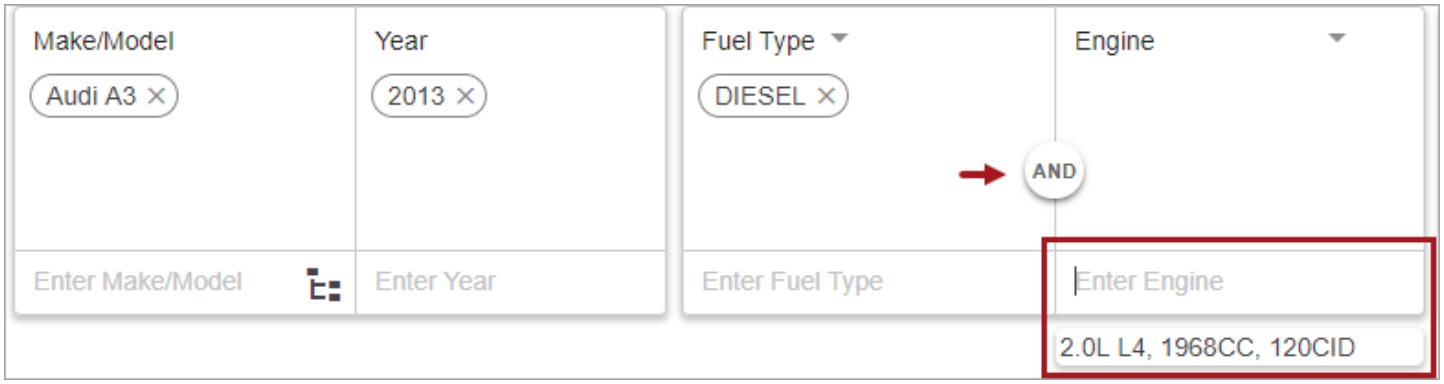
**Note:** Options search boxes can be configured with many different attributes and/or references. For the following examples, the Options search boxes are displayed using the Fuel Type and Engines attributes.



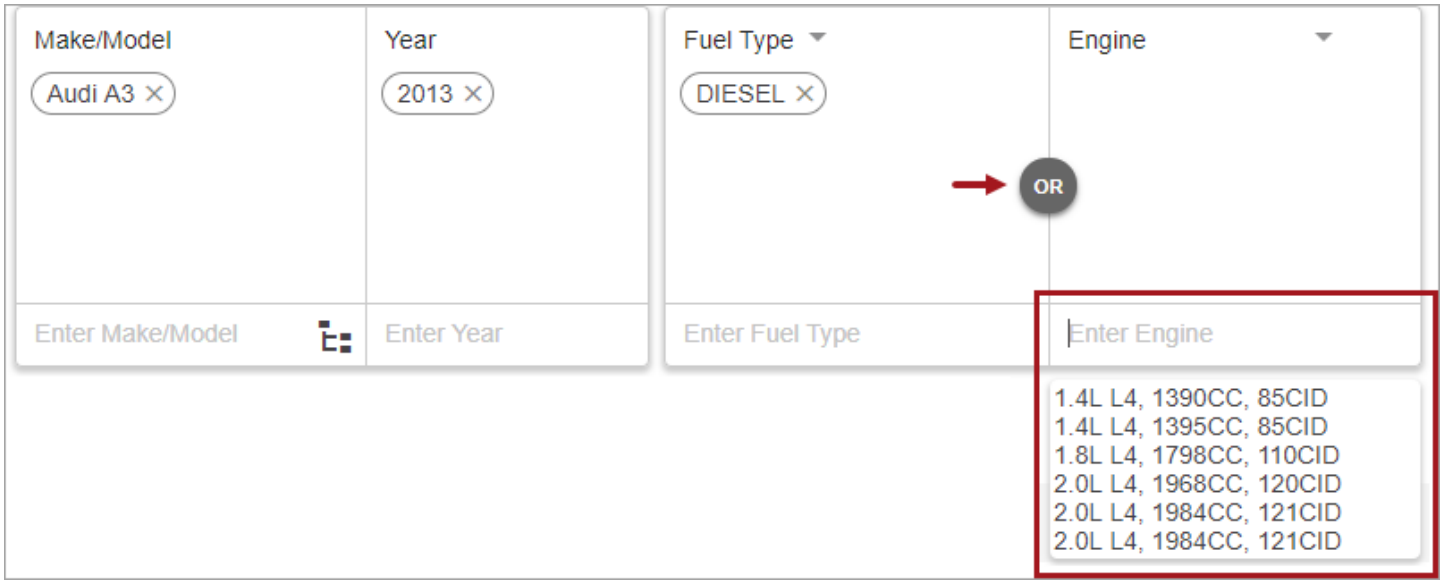

When the Fuel Type options search box is left blank, then the Engine options search box will display the six engines available for the 2013 Audi A3 (as shown below).



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).




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.




For more information on the Options search boxes, refer to the **Using the Options Search Boxes** topic within the Intelligent Search Interface section of this guide.

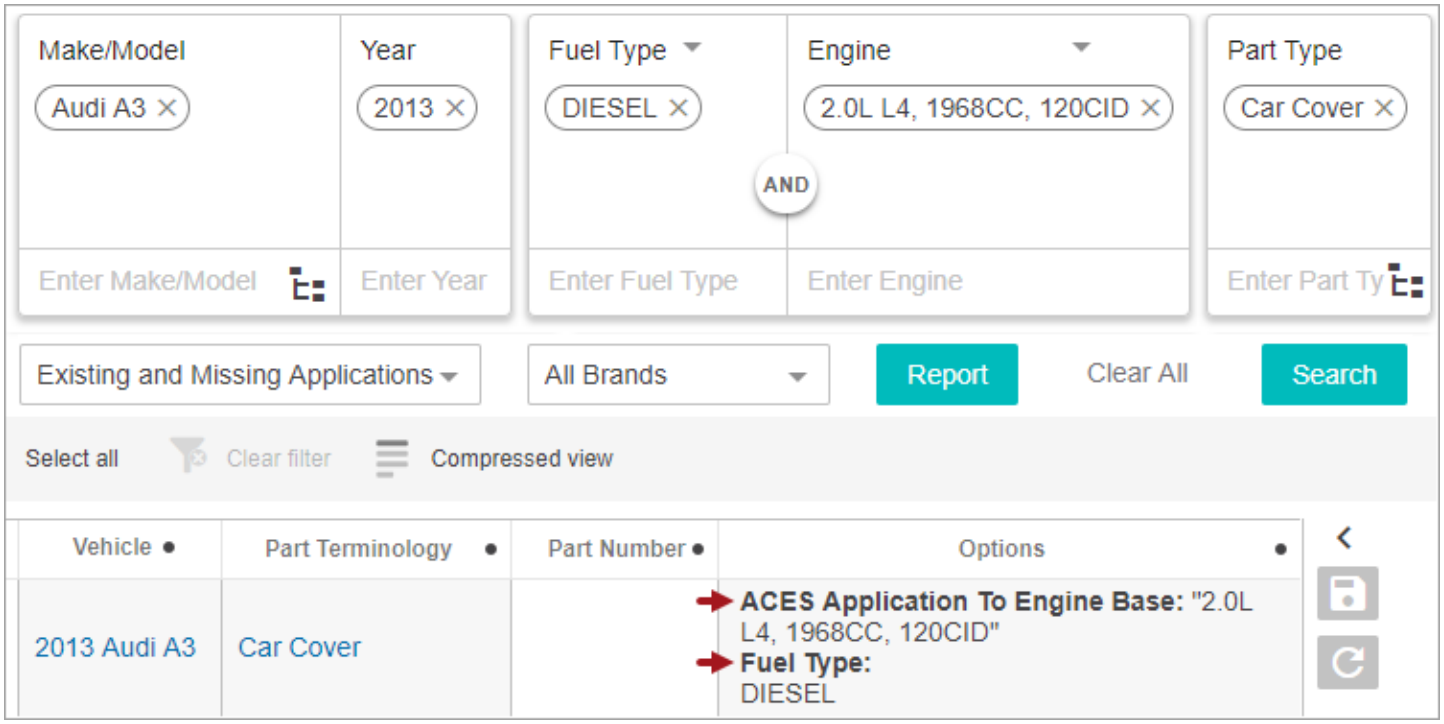
## Validation Path Functionality and Options on Applications

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 applicated 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).




### 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 applicated 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.

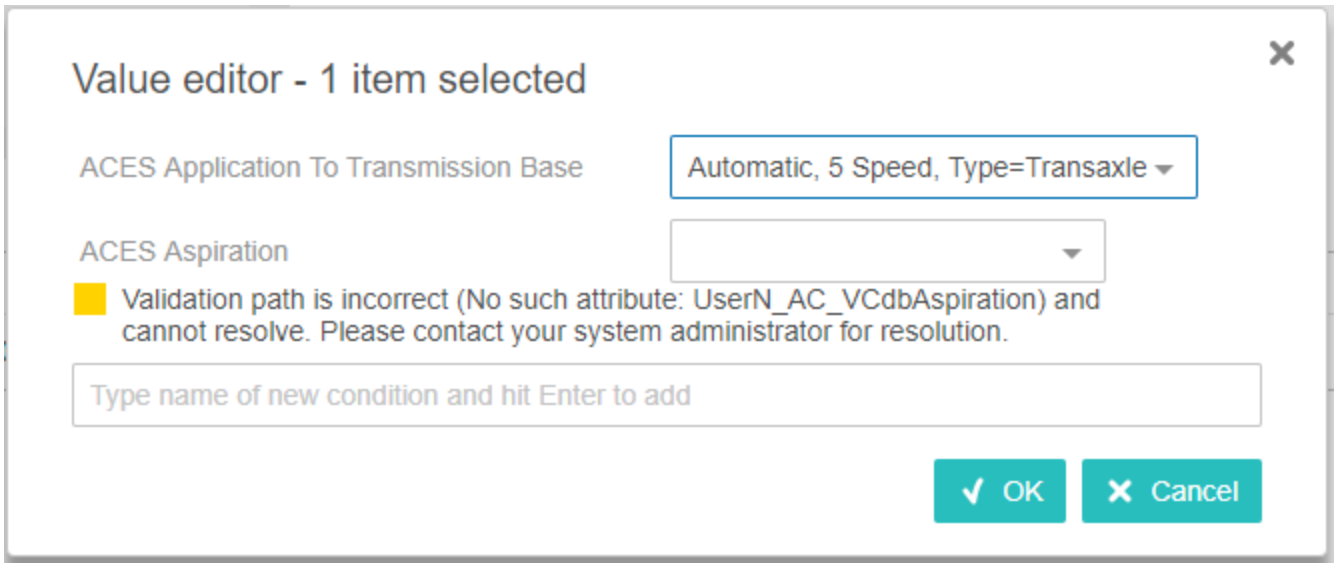
## Error Handling

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.

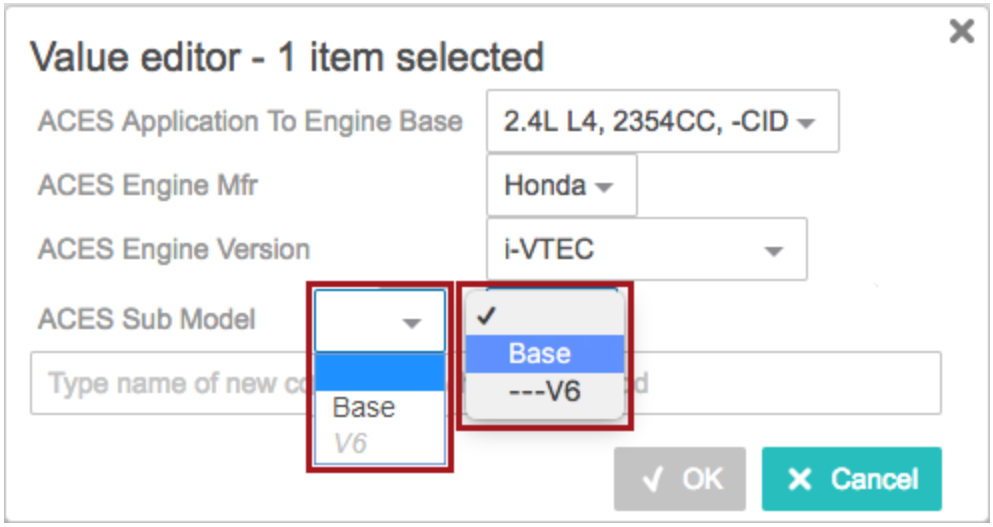
If the condition attribute with the invalid path is displayed in its own column, then a warning icon will be displayed and hovering over the icon will display a message indicating which reference or attribute contains the error.

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
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:

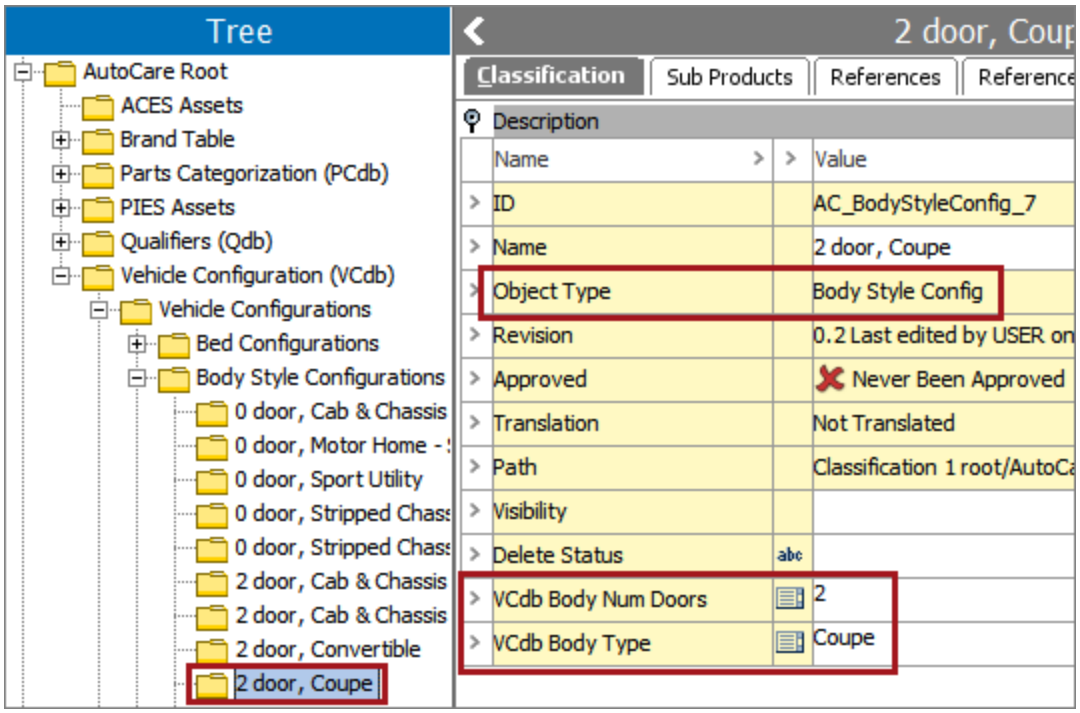
- **Vehicle Configurations**
- **Base Vehicle Applications**
- **Options on Applications**

**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.

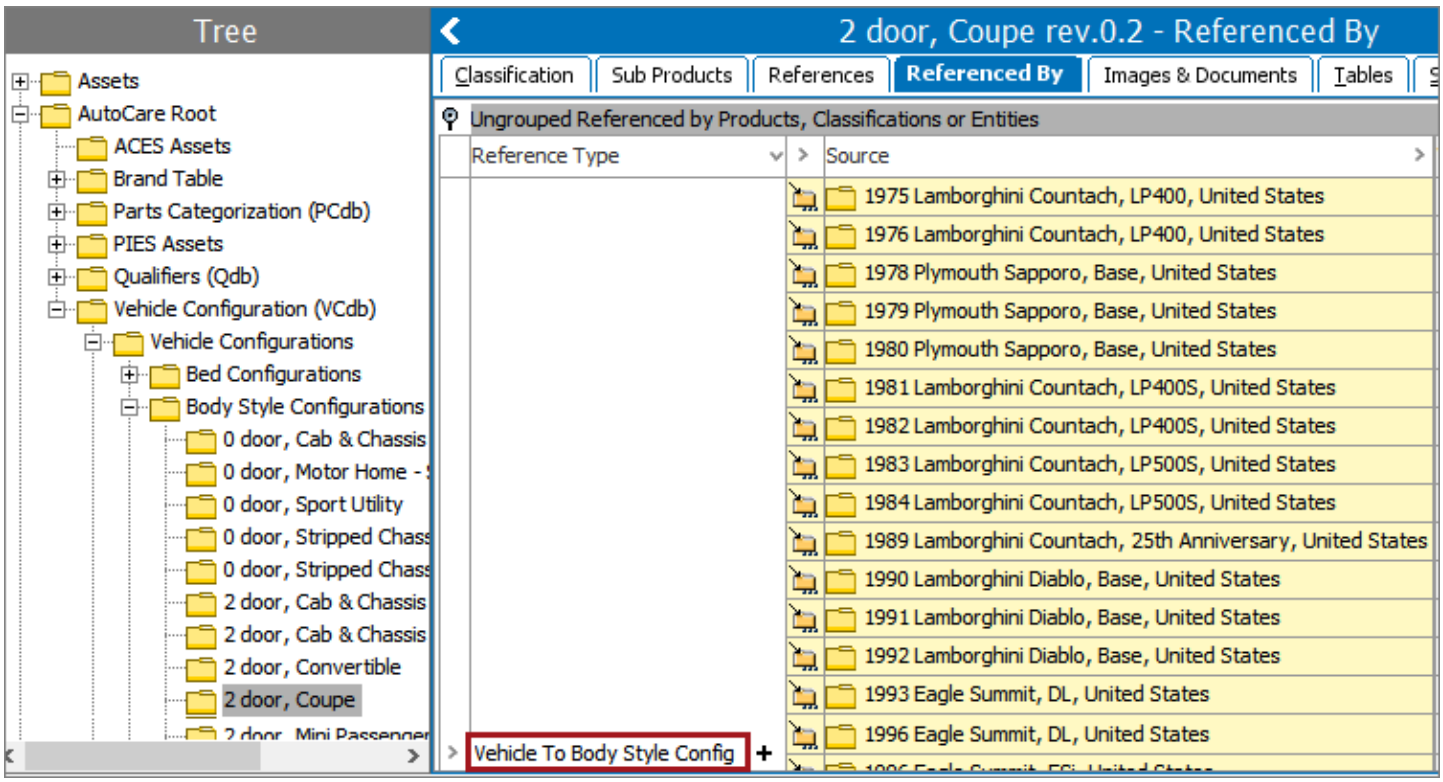
### Vehicle Configurations

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.

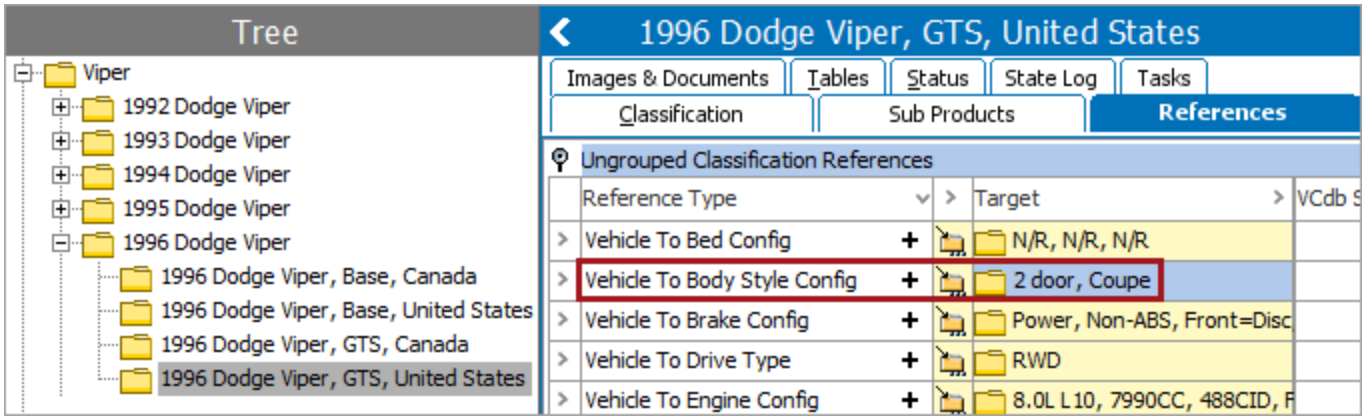
In the example below, the **VCdb Body Num Doors**, and **VCdb Body Type** AutoCare options are modeled as attributes within the **Body Style Config** classification object type. The valid configuration for the '2 door, Coupe' consists of '2' doors and a 'Coupe' body type.




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).




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**.

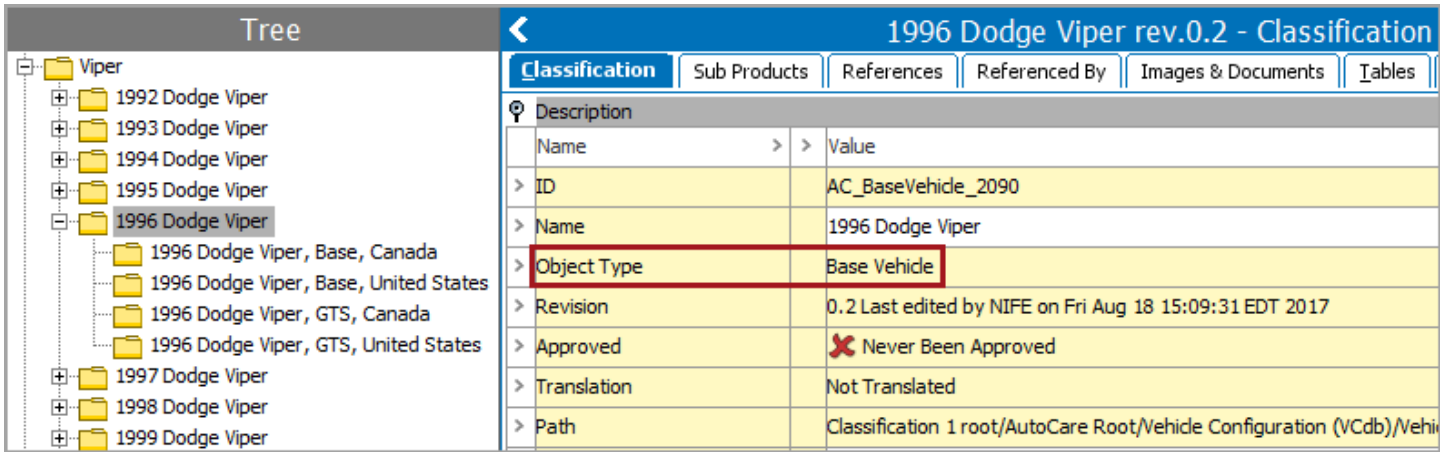



Since only one valid body style exists for this vehicle, it would be invalid for an application to be supplied for the vehicle where any other body style was implied. Additionally, within the Application Manager, it is not possible for a user to select a value other than 2 for the Body Num Doors value, or Coupe for the Body Type value.

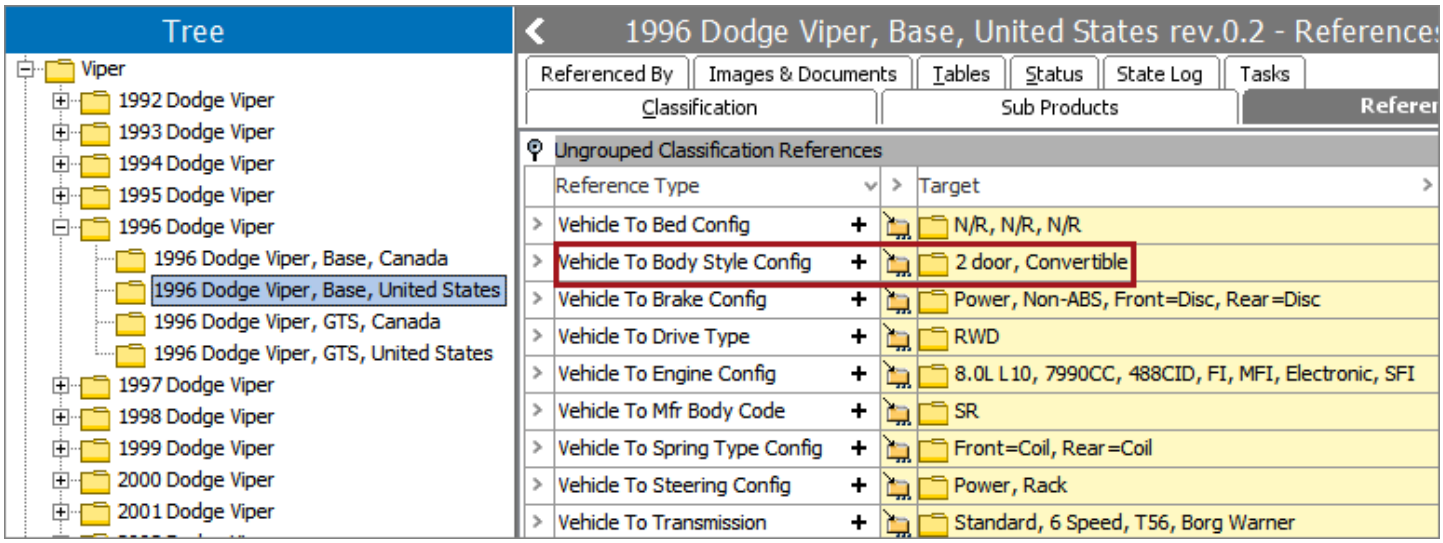
### Base Vehicle Applications

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




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 observed 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).

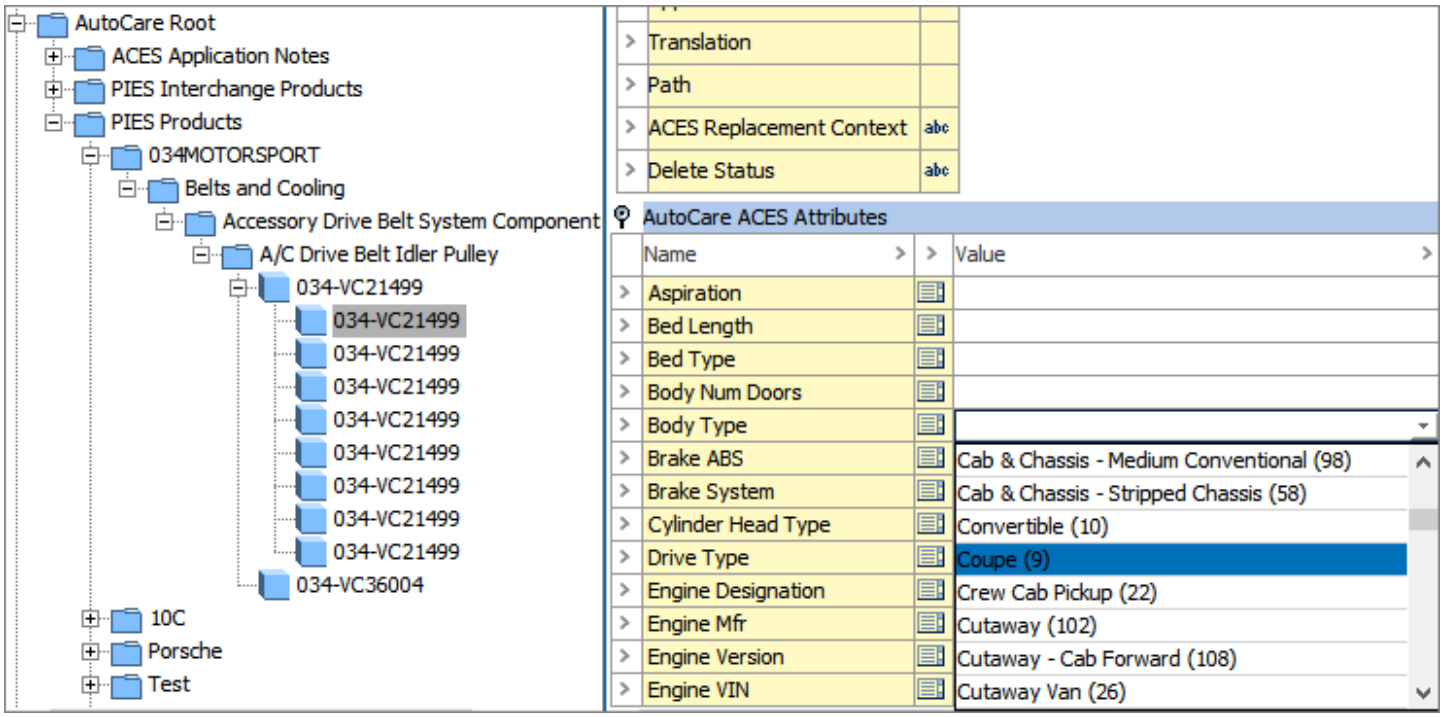


Therefore the **1996 Dodge Viper Base Vehicle** object type has two valid body style configurations; both have only 2 doors, but the body type can be either Coupe or Convertible. If a user tries to provide an application for the vehicle that specifies '4' doors or a 'Sport Utility' Vehicle body style, it would be allowed, but also invalid.

### Options on Applications

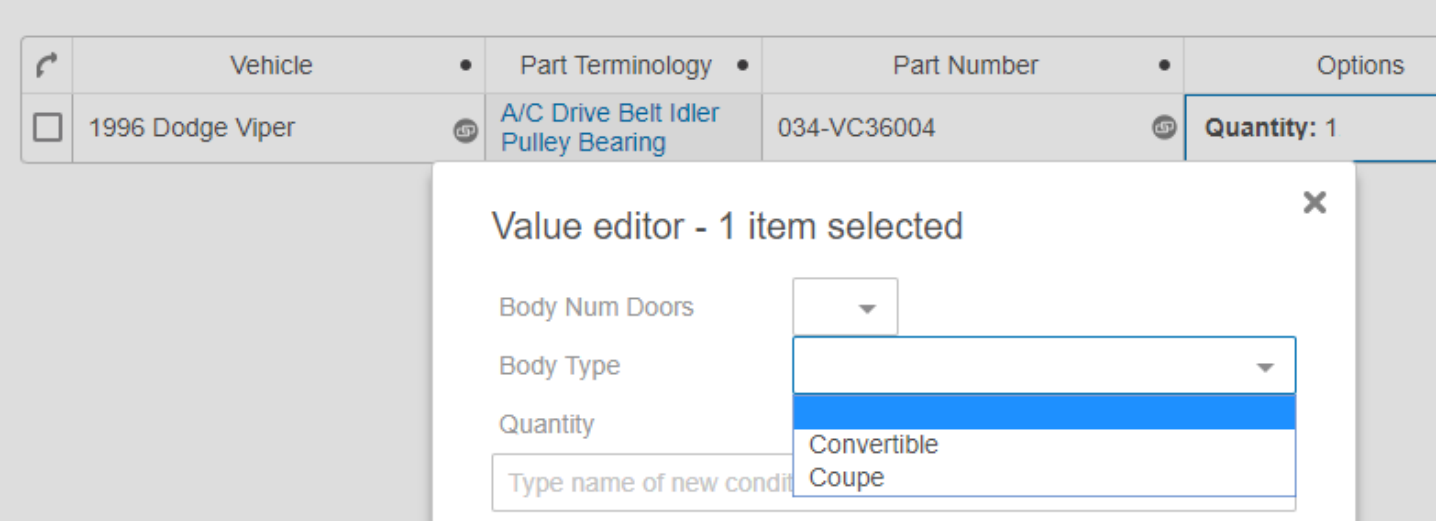
On an application, an attribute or reference exists to model each data point in a valid configuration. For the **Vehicle To Body Style Config** reference type an attribute for **Body Num Doors** and **Body Type** exists (similar to the configuration classification object shown above).

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.




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 | |
| Automotive Validation Path | 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

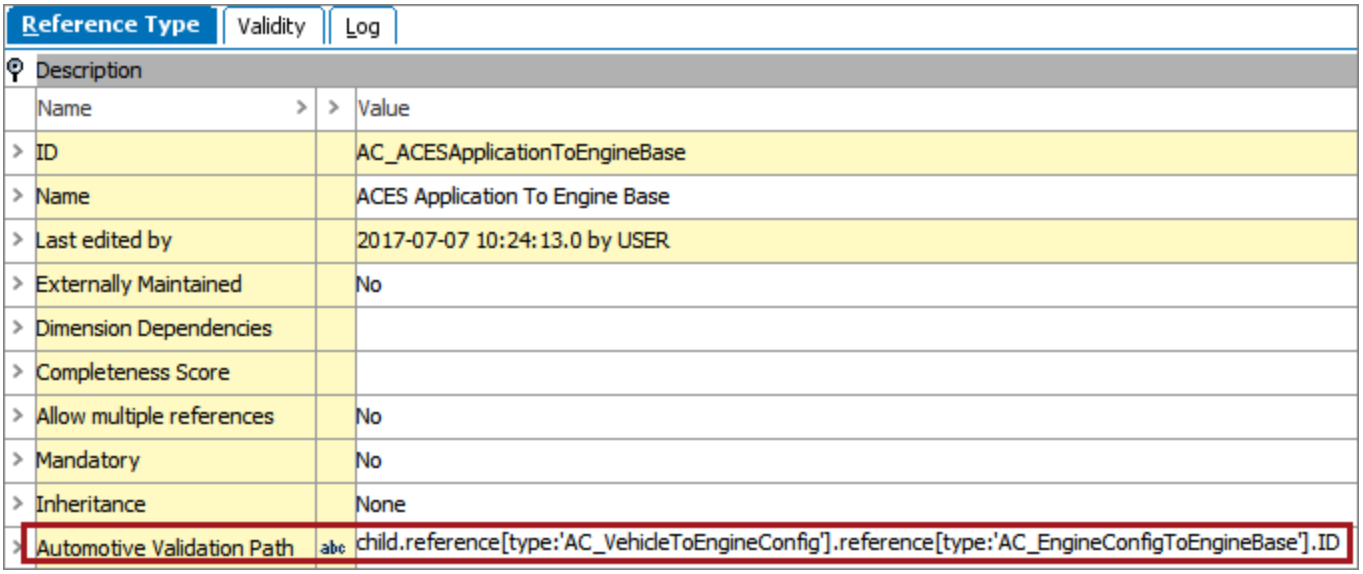
The syntax of the automotive validation path is a series of commands, each separated by a period (.). Using the child, reference, ID (shown below), and attribute elements (including IDs of the applicable references and attributes to be followed), an automotive validation path can be applied to any attribute 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.




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_VehicleToEngineConfig) 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, refer to 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, refer to 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.

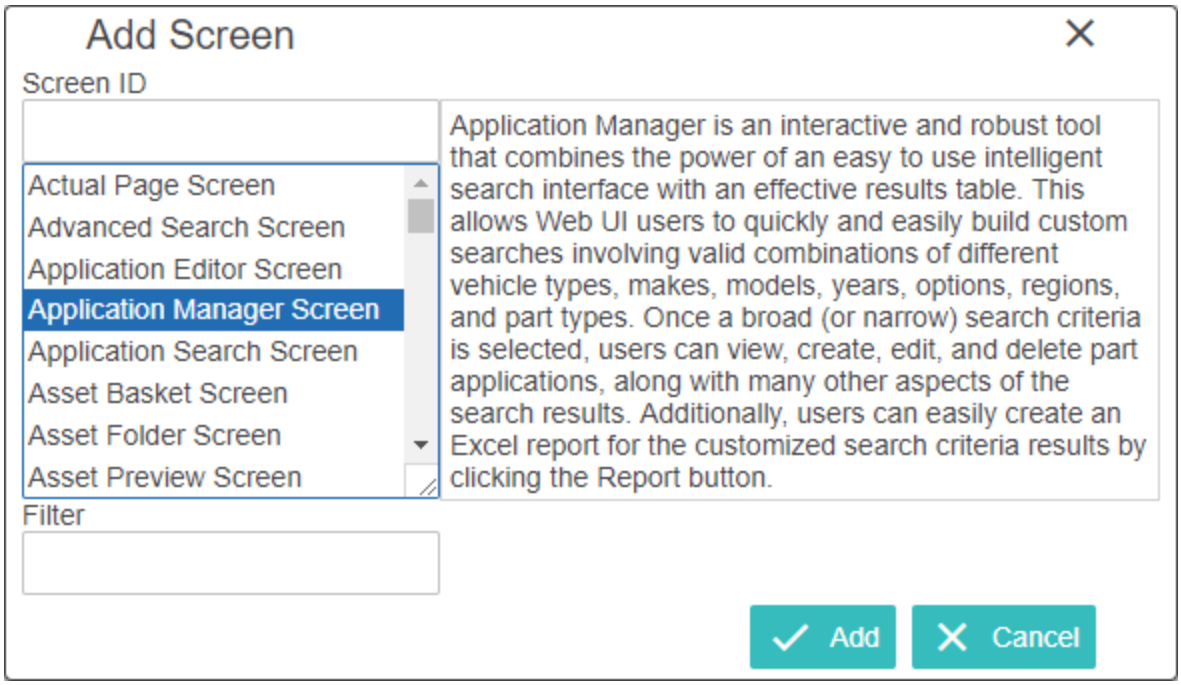
## Configuring an Application Manager Screen in Web UI

Steps to configure an Application Manager screen in Web UI are below.

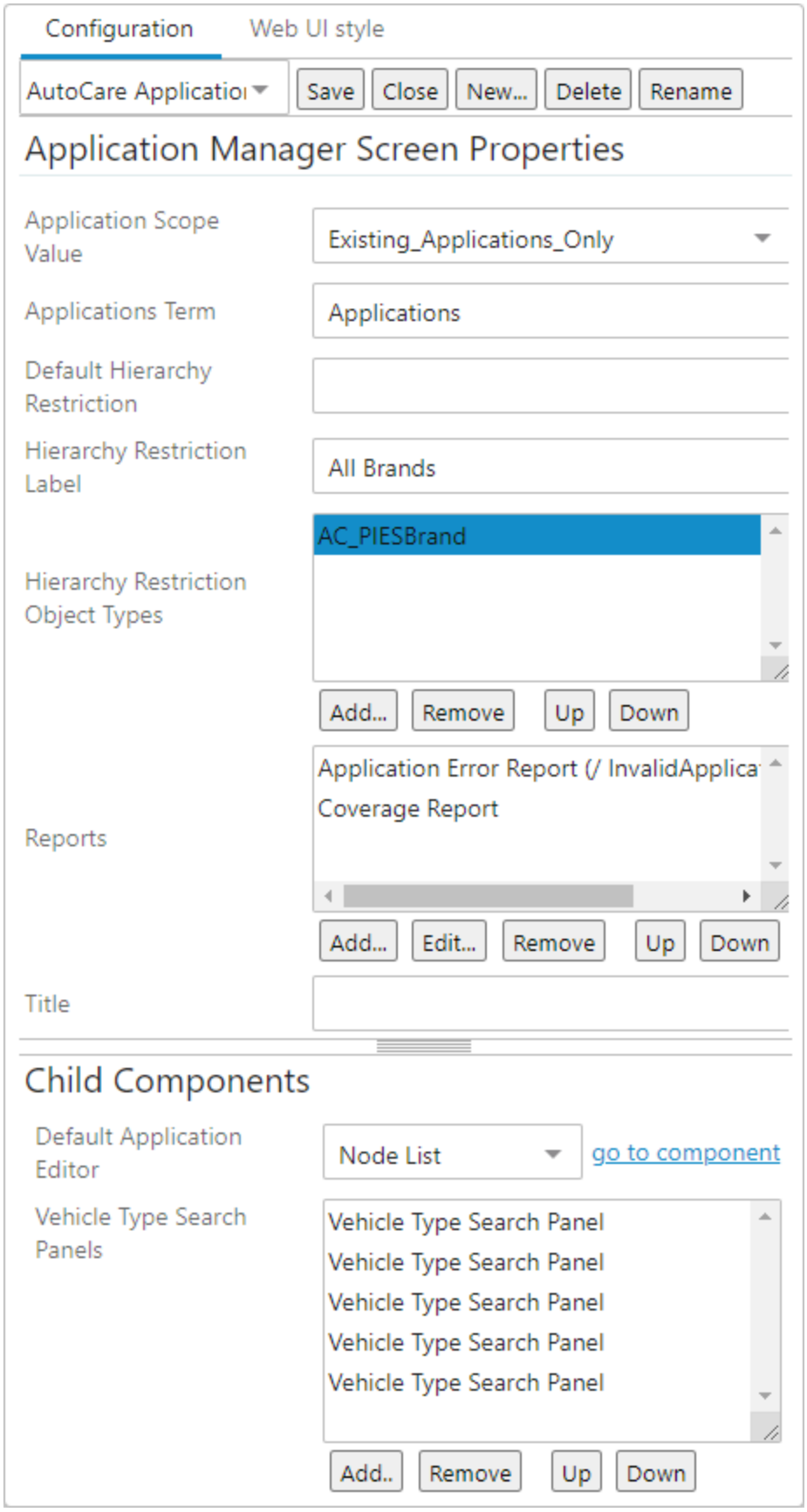
**Note:** After the initial setup, access to the configuration dialogs does not change, therefore these steps can be helpful when editing the configuration. However, when editing, the 'Add' button is replaced with the 'Save' button, and the 'Add component...' labels at the top of the dialogs display 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.


Configuration
Web UI style
AutoCare Applicatio
Save
Close
New...
Delete
Rename
Application Manager Screen Properties
Application Scope Value
Existing_Applications_Only
Applications Term
Applications
Default Hierarchy Restriction
Hierarchy Restriction Label
All Brands
Hierarchy Restriction Object Types
AC_PIESBrand
Add...
Remove
Up
Down
Reports
Application Error Report (/ InvalidApplica
Coverage Report
Add...
Edit...
Remove
Up
Down
Title
Child Components
Default Application Editor
Node List
go to component
Vehicle Type Search Panels
Vehicle Type Search Panel
Vehicle Type Search Panel
Vehicle Type Search Panel
Vehicle Type Search Panel
Vehicle Type Search Panel
Add..
Remove
Up
Down

Within the Application Manager Screen Properties dialog, the following parameter is required:

**Vehicle Type Search Panel:** Allows to configure Vehicle Type Search Panels 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 five vehicle type search panels are added, an ellipsis button will be shown below the fifth Vehicle Type Search Panel that allows users to view more Vehicle Type Search Panels.

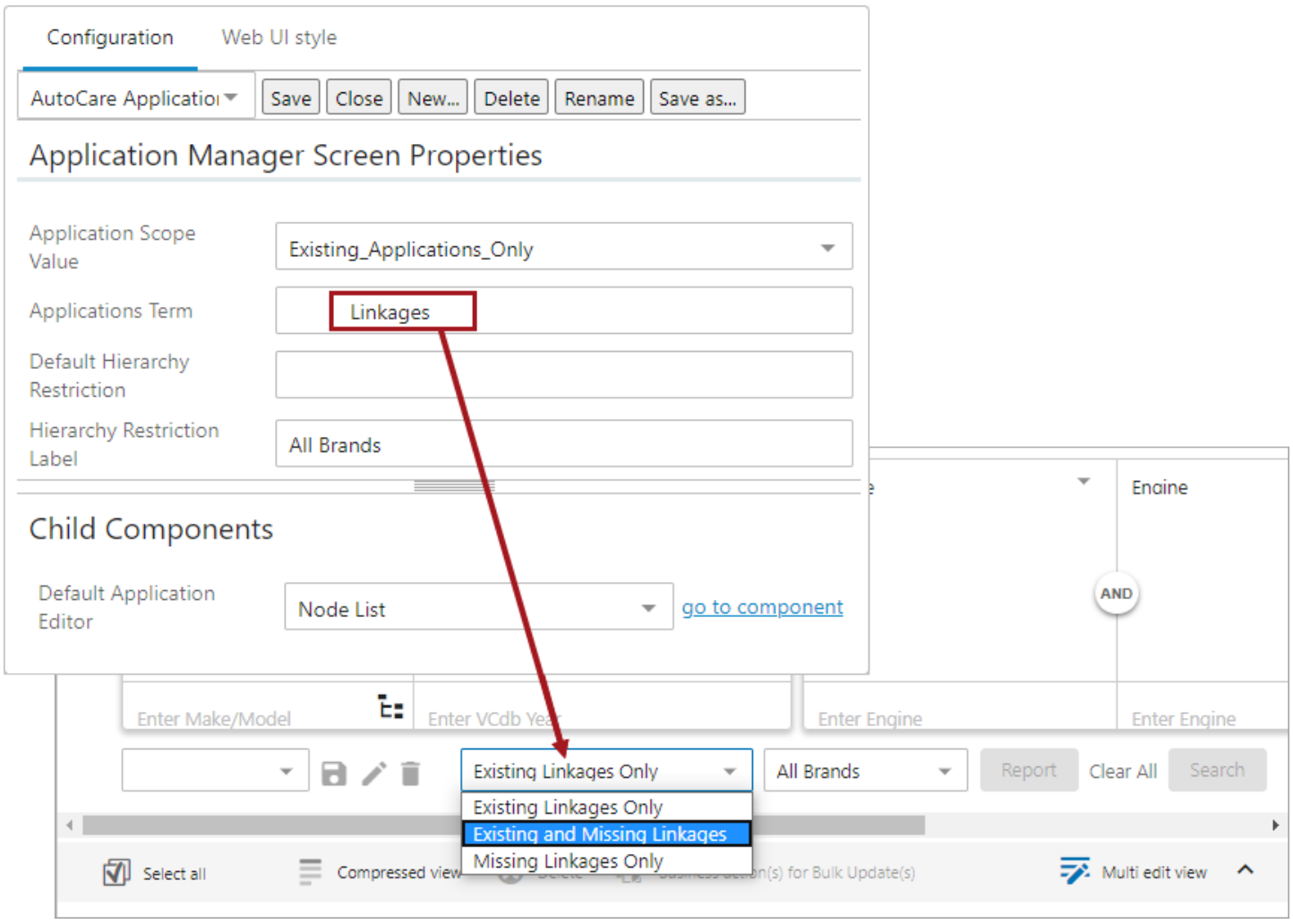
**Note:** If a vehicle search type is created, and the vehicle root node that has been configured for the Vehicle Type is later removed from STEP, then the vehicle search type created within the Application Manager will no longer display. If the vehicle root node is re-created, then the Application Manager 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, refer to the **Missing Application Coverage Functionality** topic.

**Applications Term:** This parameter provides users the ability to change the terminology 'Applications' within the Existing and Missing Applications drop-down list. This parameter is defaulted to be populated with value 'Applications' as shown below.

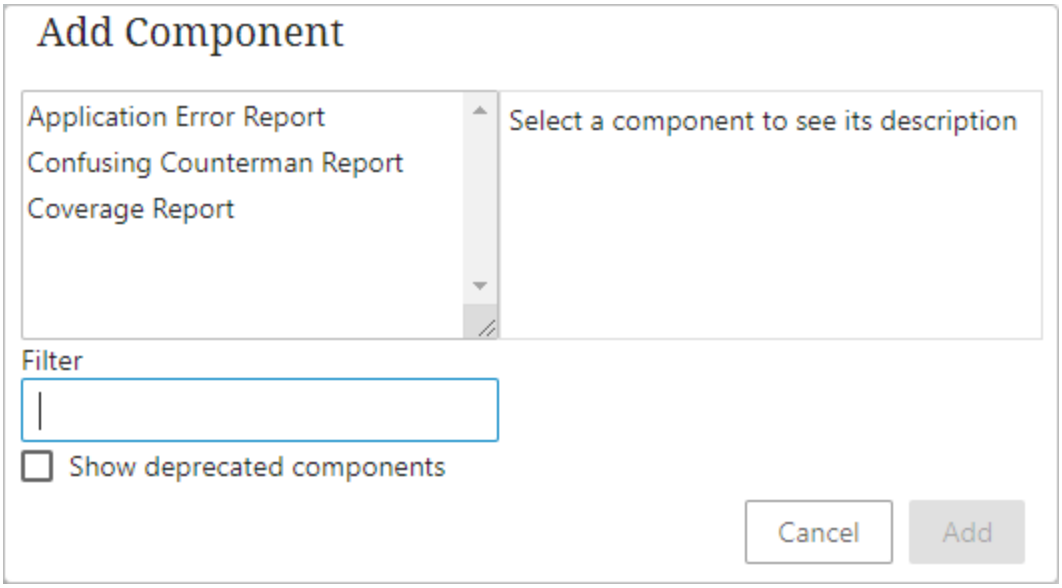
Below is an example of the Applications Term parameter populated with the value 'Linkages'.



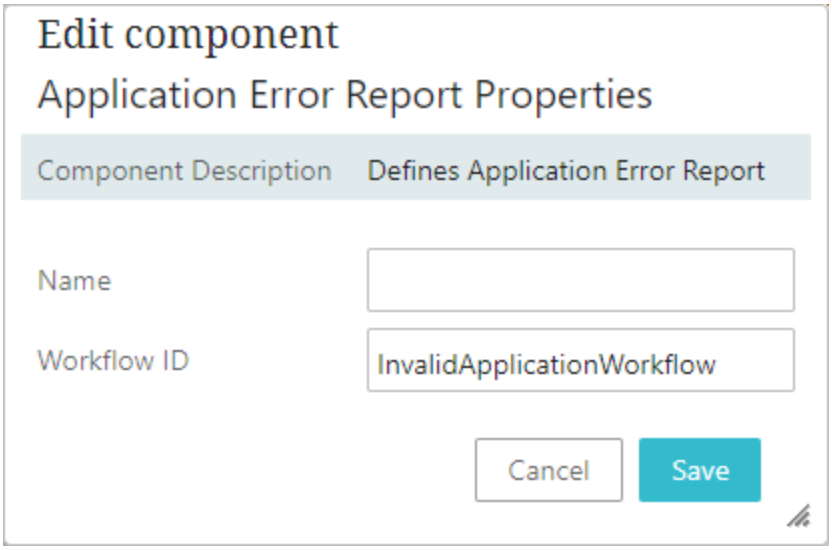

**Hierarchy Restriction Label:** Type a label that best describes the Hierarchy Restriction option to the Application Manager user. By default, the label is populated with 'All data.' If users are filtering by brand, then 'Brand' could be an appropriate label.

**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.

**Reports:** Allows to configure the Create Report button within each Application Manager. Within this parameter, user can add components that defines the type of reports that can be exported. Click the Add button below the parameter field to include the following options:




- **Application Error Report:** Adding this component allows users to export a report of invalid application records in an Excel spreadsheet (XLSX file type) and initiate the invalid application records into a workflow. Double click on the added Application Error Report in the Reports parameter to further configure the following parameters:



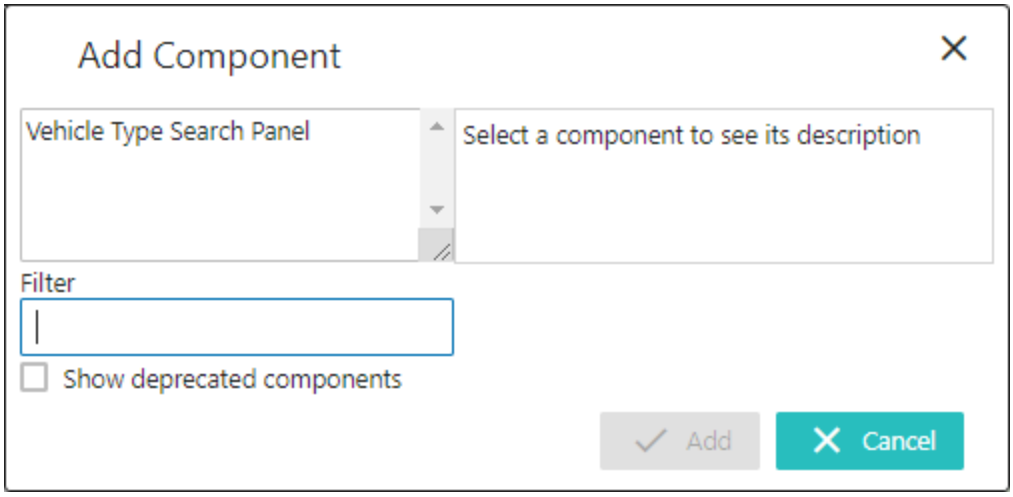

  - **Name:** Enter a text that needs to be displayed as a radio button when the user clicks the Create Report button in the Application Manager.
  - **Workflow ID:** Enter the ID of the workflow to which the invalid application records are to be initiated. The workflow should be able to handle invalid applications. The workflow defined in this parameter should be valid for the object type of the application record. For the example above, the Invalid Application Workflow (ID = InvalidApplicationWorkflow) is entered.

- **Confusing Counterman Report:** Allows users to generate and export a Confusing Counterman Report in an Excel spreadsheet (XLSX file type). This component generates a report of ambiguous application records that would confuse a counterman.
- **Coverage Report:** Allows users to export an Application Coverage Report in an Excel spreadsheet (XLSX file type). Double click on the Coverage Report component to enter a name that needs to be displayed as a radio button when the user clicks the Create Report button in the Application Manager.

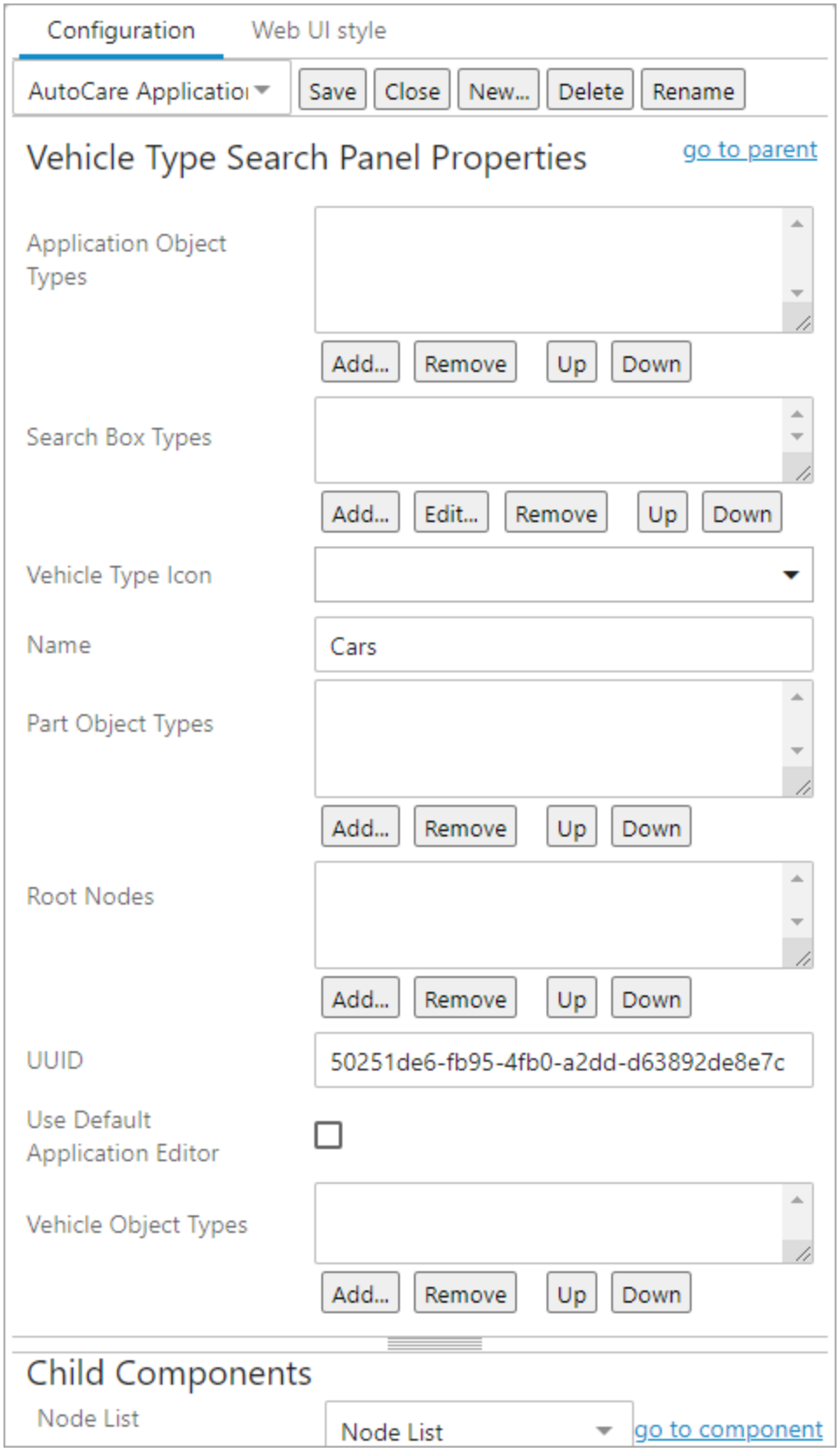
**Title:** If desired, type in a title which will display on the screen in the top left corner. If no text is entered, no label is displayed.

**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, refer to 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.

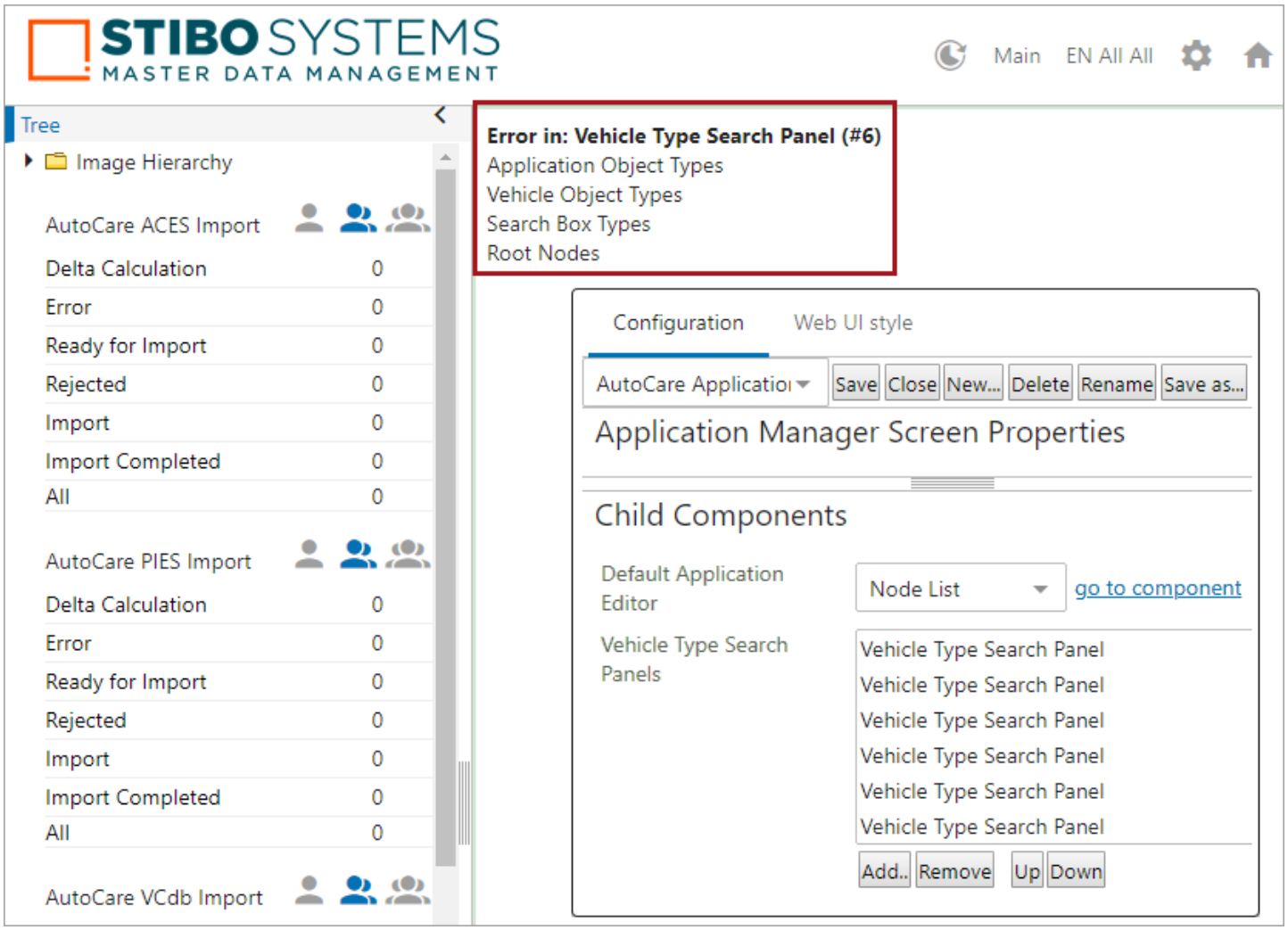



5. Select the Vehicle Type Search Panel component, click **Add**, and the newly added component will display within the Vehicle Type Search Panel parameter.
6. Double click the newly added component available in the Vehicle Type Search Panel parameter, and the Vehicle Type Search Panel Properties dialog will display (as shown below). Parameter options are explained below:


Configuration
Web UI style
AutoCare Applicatio
Save
Close
New...
Delete
Rename
Vehicle Type Search Panel Properties
go to parent
Application Object Types
Add...
Remove
Up
Down
Search Box Types
Add...
Edit...
Remove
Up
Down
Vehicle Type Icon
Name
Cars
Part Object Types
Add...
Remove
Up
Down
Root Nodes
Add...
Remove
Up
Down
UUID
50251de6-fb95-4fb0-a2dd-d63892de8e7c
Use Default Application Editor
Vehicle Object Types
Add...
Remove
Up
Down
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.




**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.

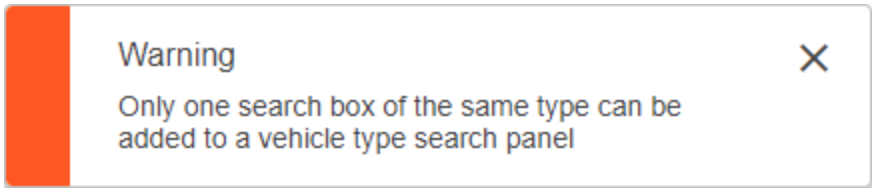
- Object type(s) selected for this parameter must be configured as an Application within the 'Automotive - Application Model.' Easy Setup configures the Automotive - Application Model in this way.
- Any object that is selected, and is configured as an Application in the Automotive - Application Model, 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 Search Box
- Configuring Part Type Search Box
- Configuring Year Search Box

Because each search box type can 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.

**Part Object Types:** This parameter allows adding one or more part object types to be used for filtering application records. Adding a part object type ensures that the Application Manager results table will only display the application records belonging to the parts of the listed object type. Object type(s) selected for this parameter must be configured as a part 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.

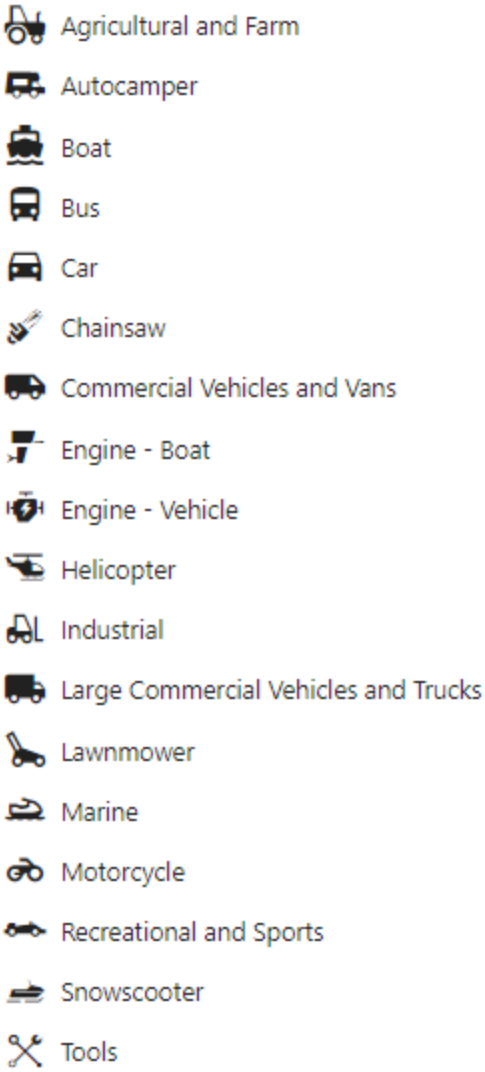
**Root Nodes:** Allows for the restriction of root nodes to be used within the Application Manager when searching for a vehicle. The root node(s) listed should correspond to the Icon and Name parameters because the search input field displays only suggestions for Make/Model objects that belong in the 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. The following Vehicle Type icons are available for selection:

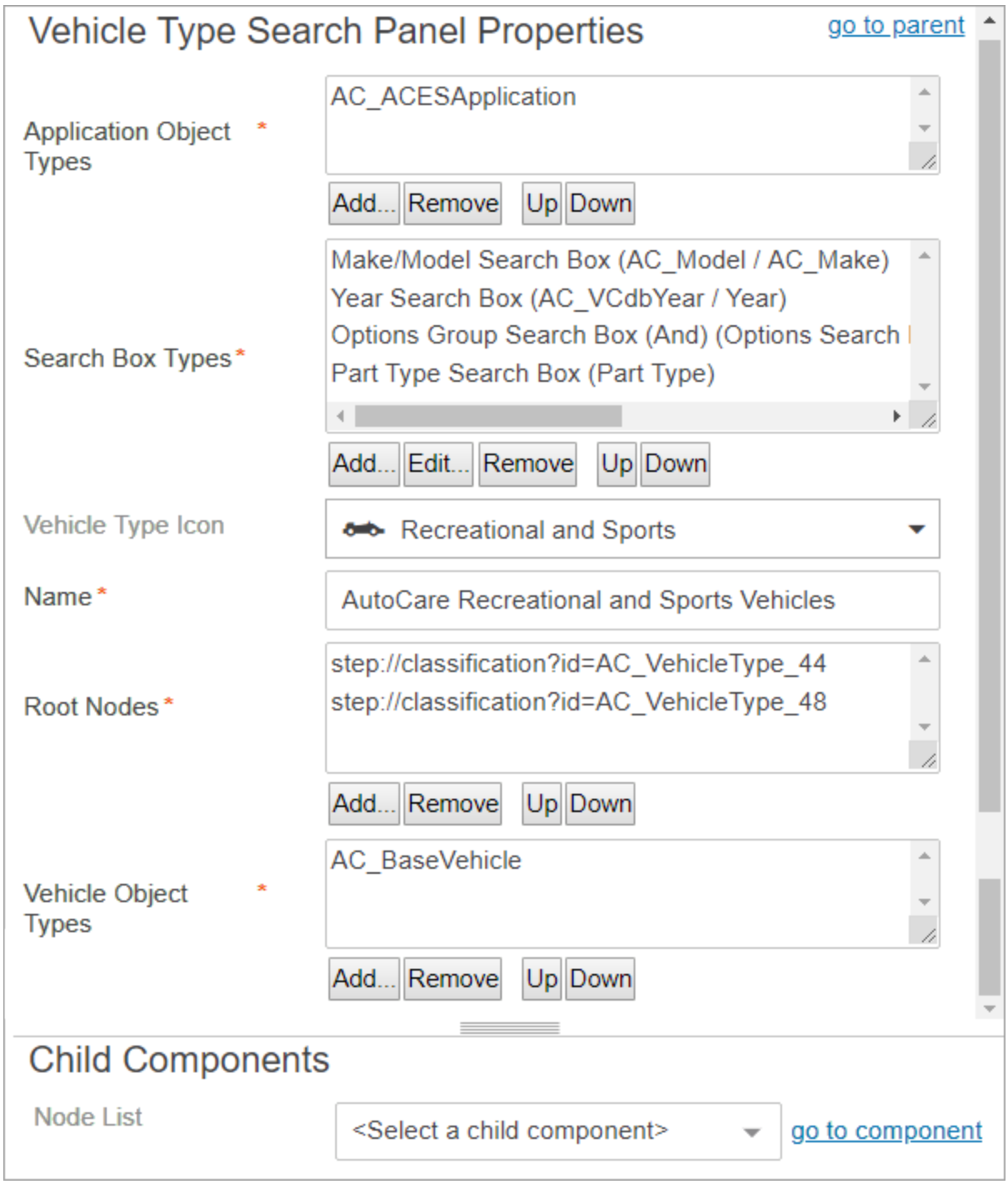


**Note:** The values populated for the required fields should all correspond to one or more vehicle types being configured for the Application Manager.

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, refer to 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.



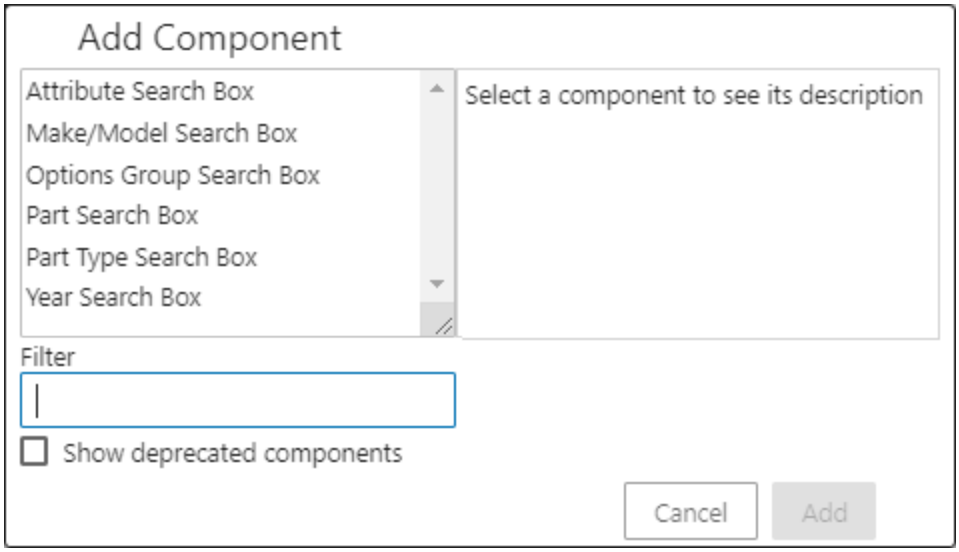

7. Once the required parameters are populated, click the **Save** and **Close** buttons to exit the Designer.

Many optional parameters are available within the Application Manager Screen Properties > Child Components > Node List.

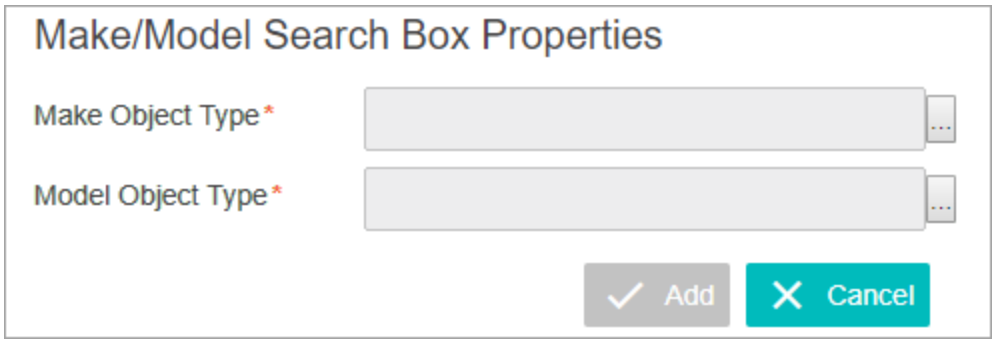
## Configuring Make/Model Search Box

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.

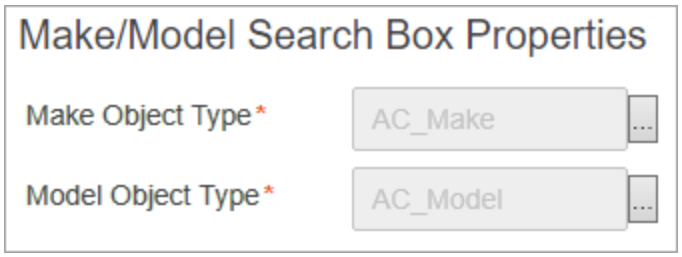



**Make Object Type:** Determines the Make objects that a user can select within the Make/Model search box. Click the ellipsis button (...) to find and select the Make Object Type for the Makes that exist in the vehicle classification (defined in the 'Root Nodes' parameter) and should be displayed for 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.

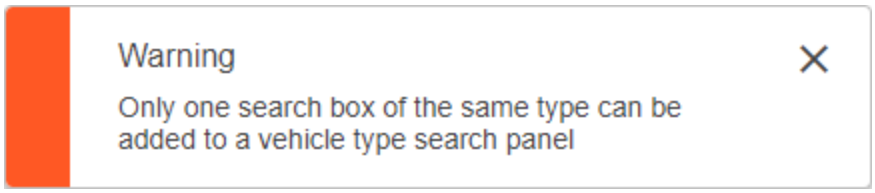
4. Click the ellipsis button (...) to select the Object Type for the makes that should display within the search box for a user to select.
5. Click the ellipsis button (...) to select the Object Type for the Models that should display within the search box for a user to select.

   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.

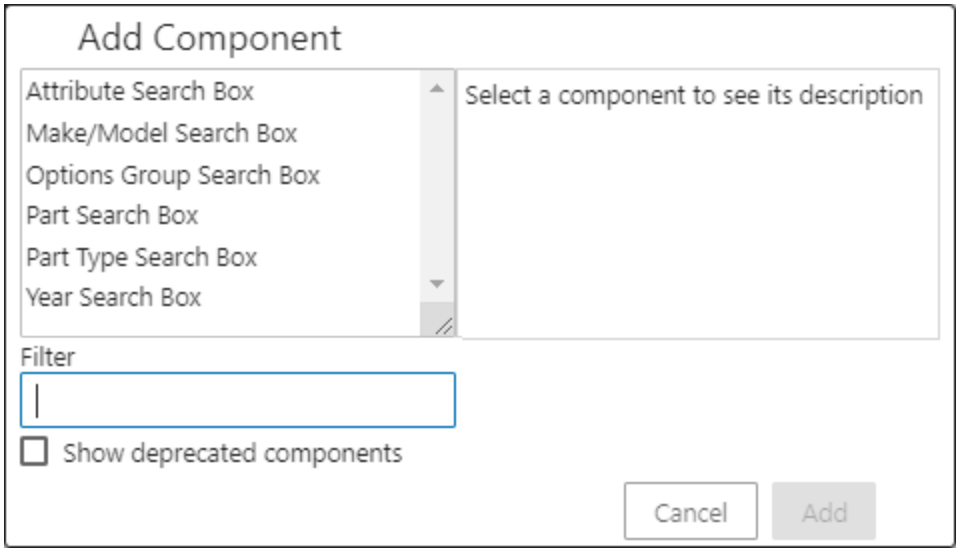
   Because each search box type can only be configured once for each Application Manager, if a second search box of the same type is added, a Warning dialog will display as shown below, and the Search Box Types parameter will remain unchanged.



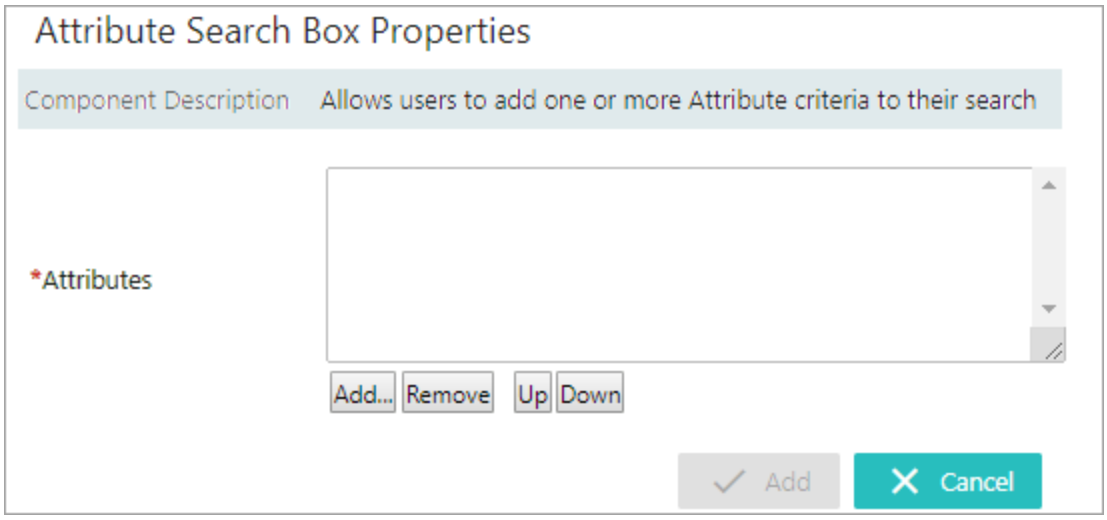
## Configuring Attribute Search Box

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.



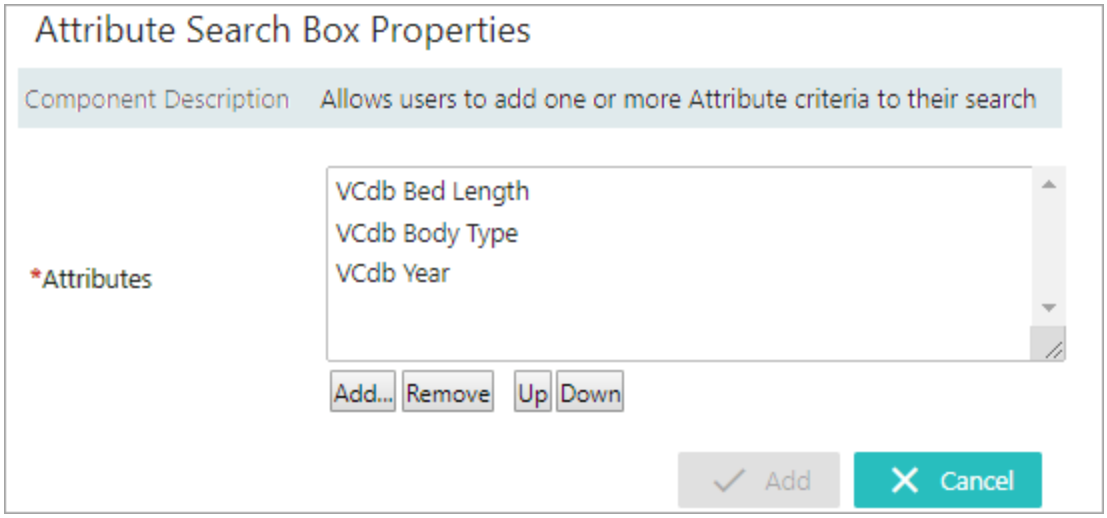

2. Click **Attribute Search Box** from the list of components.
3. Click the **Add** button, and the Attribute Search Box Properties will display, as shown below.



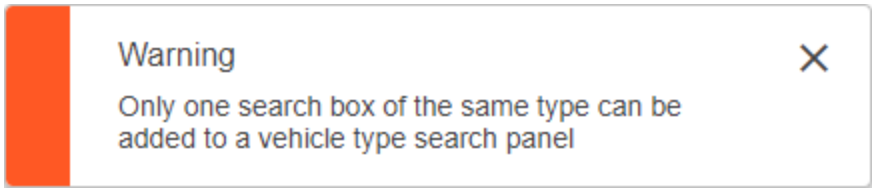

**Attributes:** This required field determines the attributes that allows users to add one or more attribute value criteria to their search. All attribute validation base types (e.g., Text, Number, List Of Values) are allowed. Click the Add button to find and select the required attributes. The selected 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.

4. 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 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.

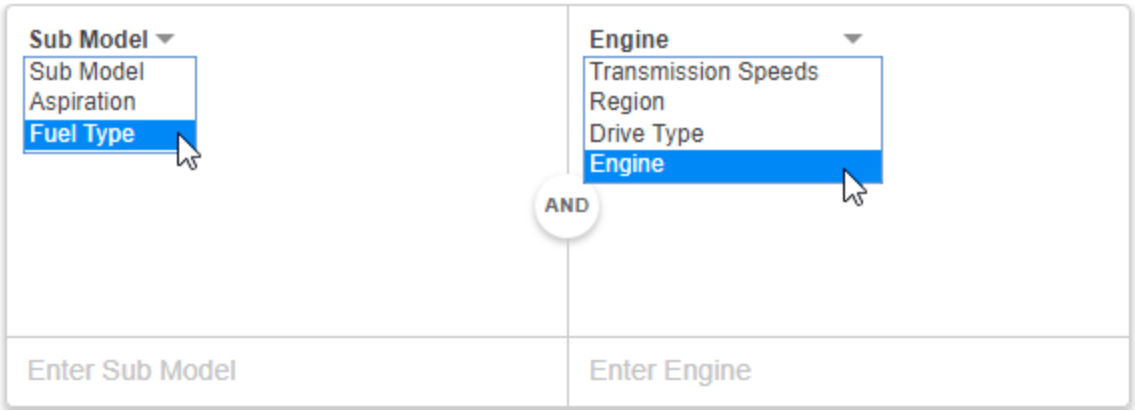


## Configuring Options Group Search Box

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 loose 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.

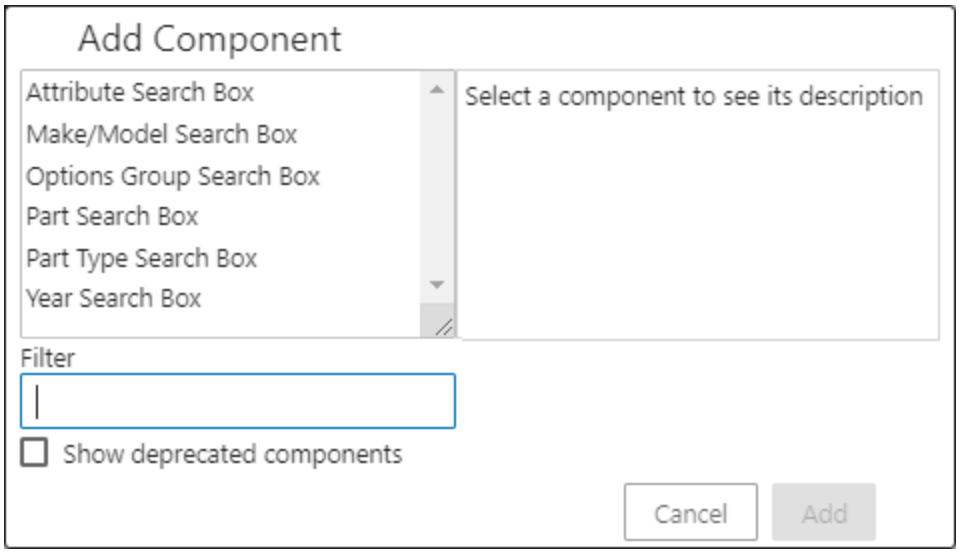
## Prerequisites

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.

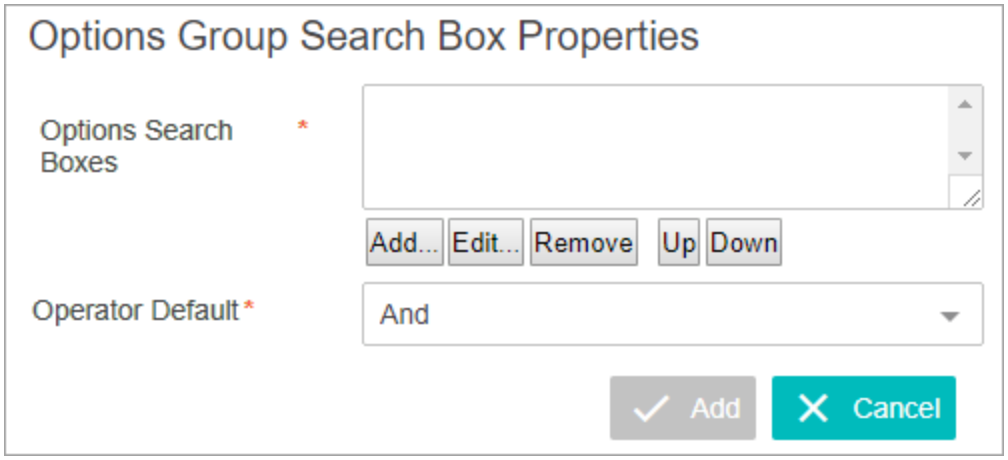
## Configuring Options Group Search Box

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.

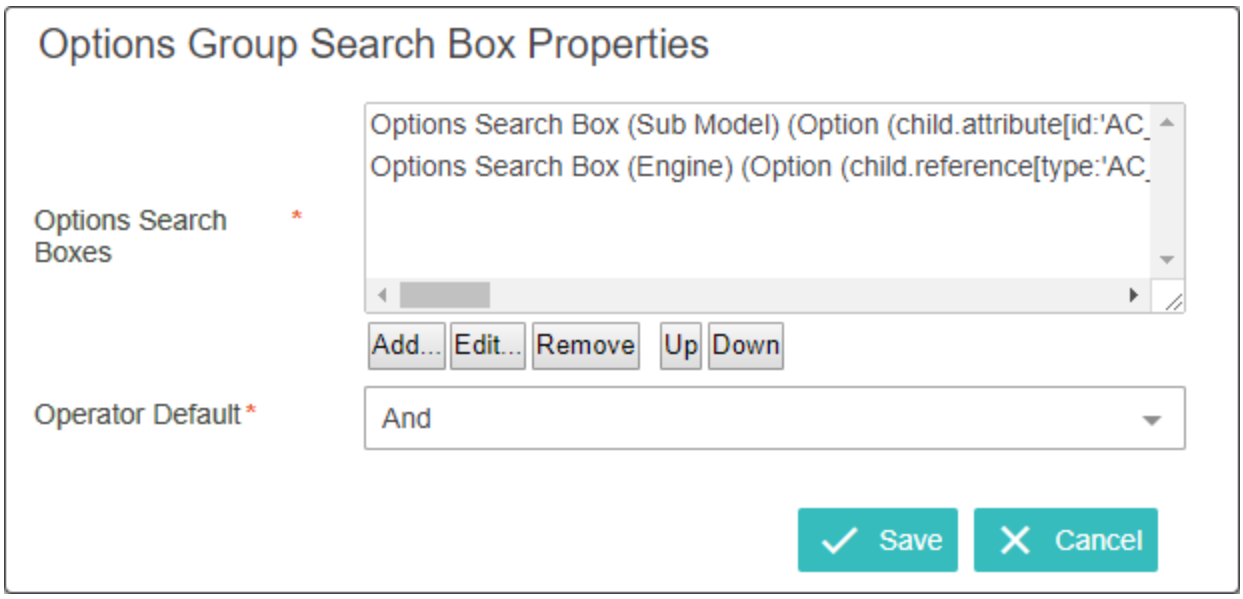



Within the Options Group Search Box Properties dialog, the following parameters are required:

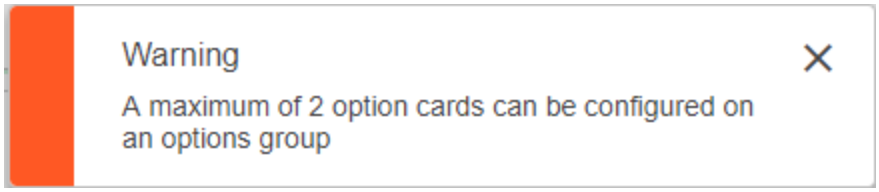
**Options Search Boxes:** Allows either one or two Options search boxes to display within the Vehicle Type Search Panel. Each Options search box can be configured to allow users to choose from a dropdown list of options. At a maximum, only two Options Search Boxes can be added to an Options Group Search Box.

**Operator Default:** Allows the default AND / OR operator to be configured for two Options search boxes. By default, AND is selected.

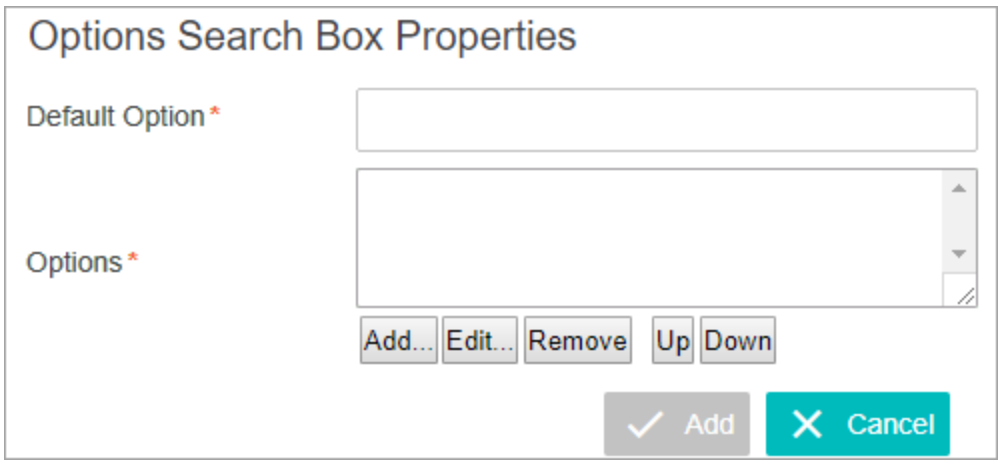
Below is an example of an Options Group Search Box Properties dialog configured for the AutoCare standard.




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.




4. Click the **Add** button beneath the Options Search Boxes parameter, and the Options Search Box Properties will display as shown below.

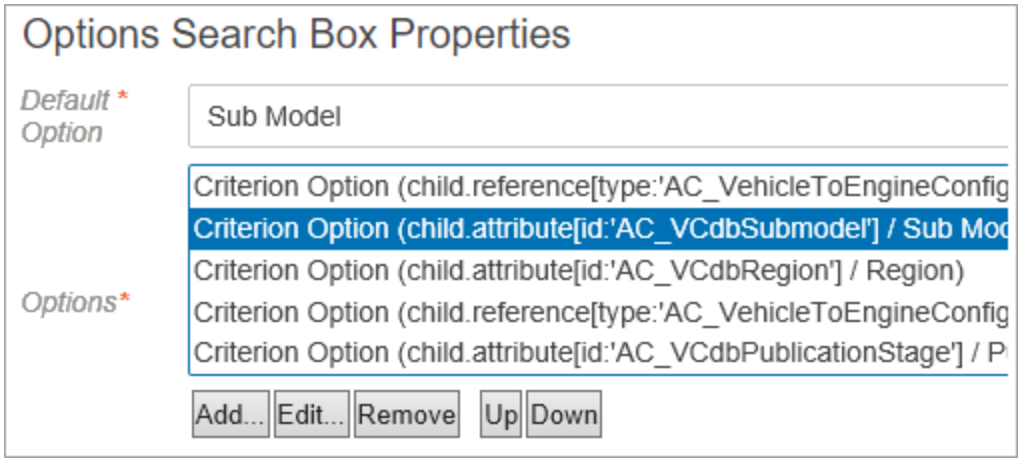



Within the Options Search Box Properties dialog, the following parameters are required:

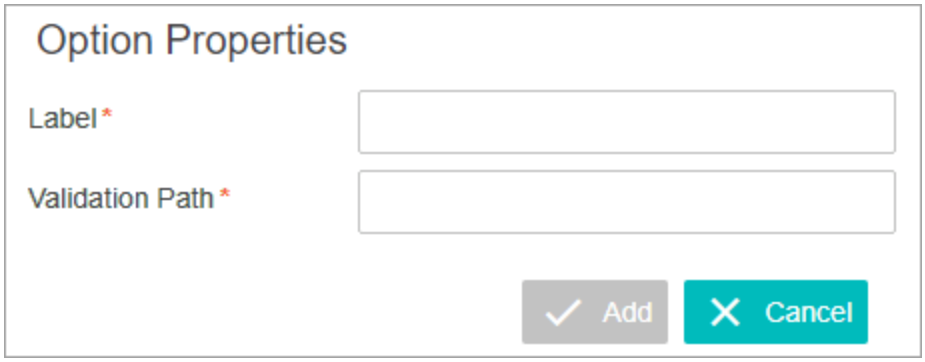
**Default Option:** Allows for a default option to display when users access the Application Manager. The value within this parameter must match the 'Label' parameter configured within the Options parameter listed below this parameter.

**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.




5. Click the **Add** button to add an option to an Options Search Box, and the Option Properties dialog will display two required parameters:



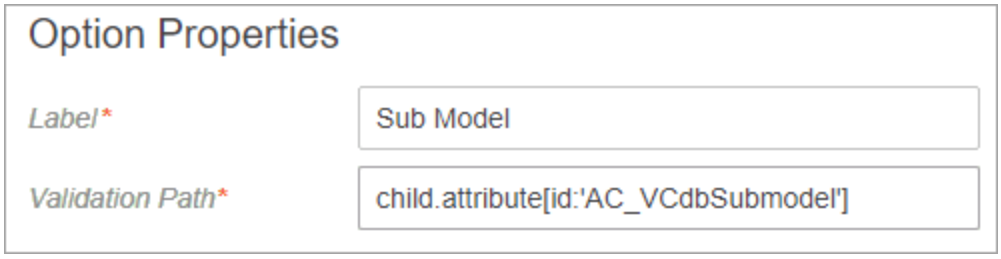

Within the Option Properties dialog, the following parameters are required:

**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.

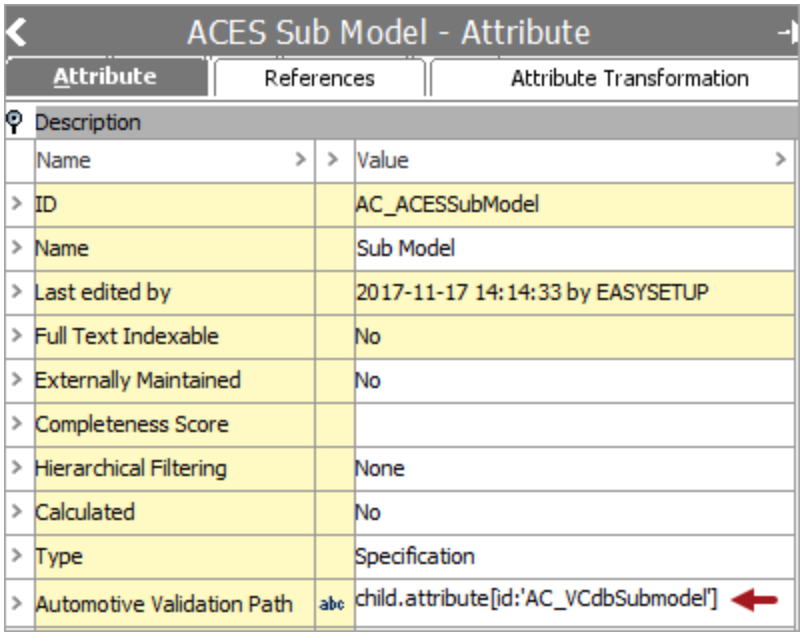
**Validation Path:** Provide a validation path for the option being configured. The option should correlate to an attribute or reference. It is recommended to copy the 'Automotive Validation Path' value for the attribute or reference from the workbench, and paste it into the Validation Path parameter field. Examples for each type of validation path can be found below.

- Attribute Validation Path Example:

  Below is an example of an Option Properties dialog configured for the ACES attribute, Sub Model.

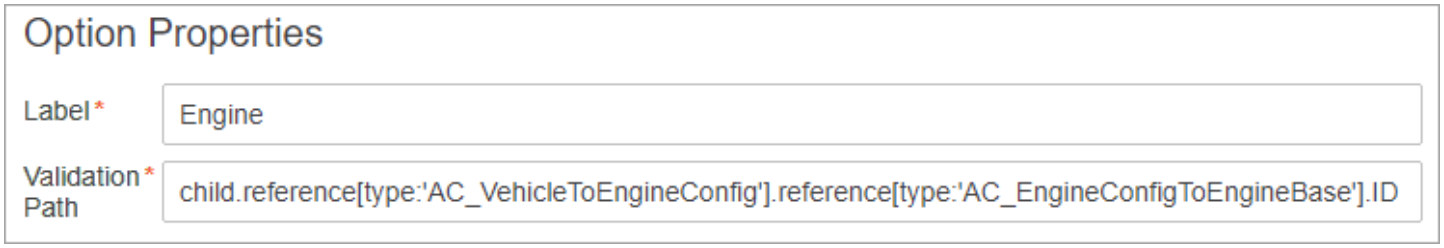



  The Validation Path is configured by the Easy Setup and displayed in the workbench. For this example, the Validation Path for the ACES Sub Model can be found by navigating to System Setup > Attribute Groups > AutoCare Attributes > AutoCare ACES Attributes > ACES Sub Model.

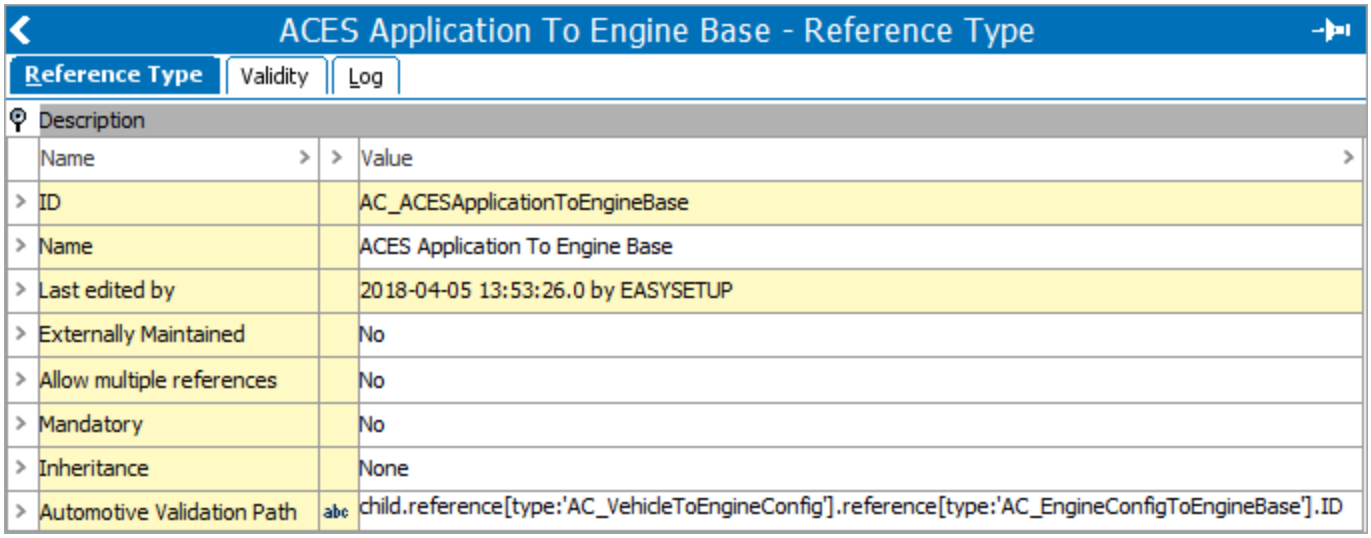



- Reference Validation Path Example:

  Below is an example of how to configure the Option Properties for the ACES reference, ACES Application To Engine Base.




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.

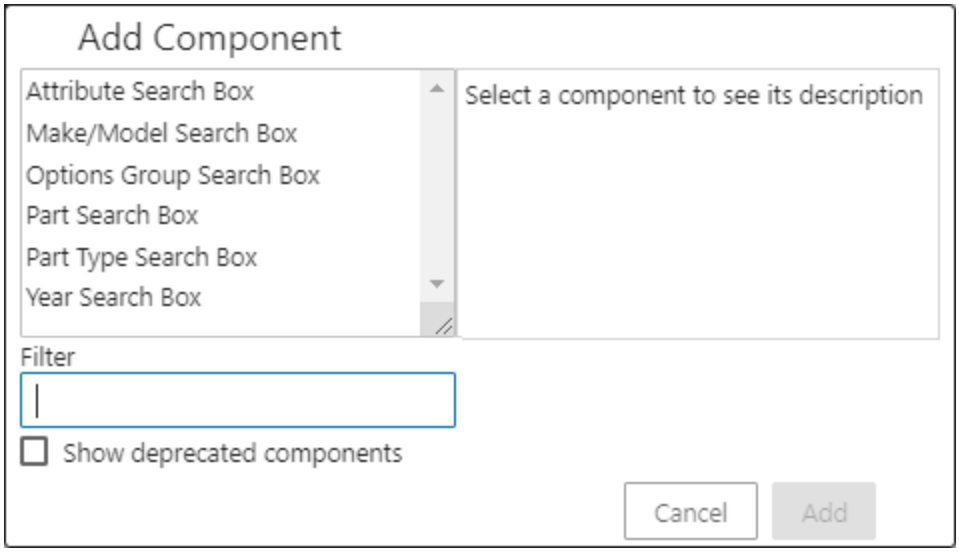



6. Once the required parameters are populated, click the **Add** button to return to the Options Search Box Properties, and optionally add additional options for the Options Search Box, or click the **Add** button and return to the Options Group Search Box Properties.
7. Optionally repeat the steps above 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

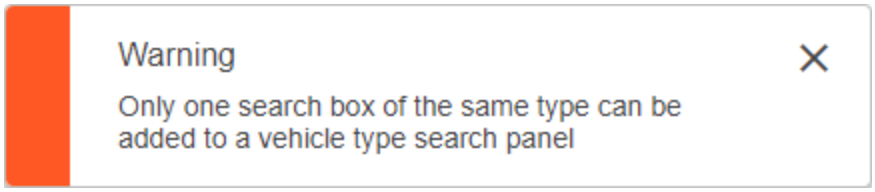
Because the Part Type search box type does not require additional configuration, it can quickly be added to the Search Box Types parameter within the Vehicle Type Search Panel Properties dialog by following the steps listed below:

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 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.

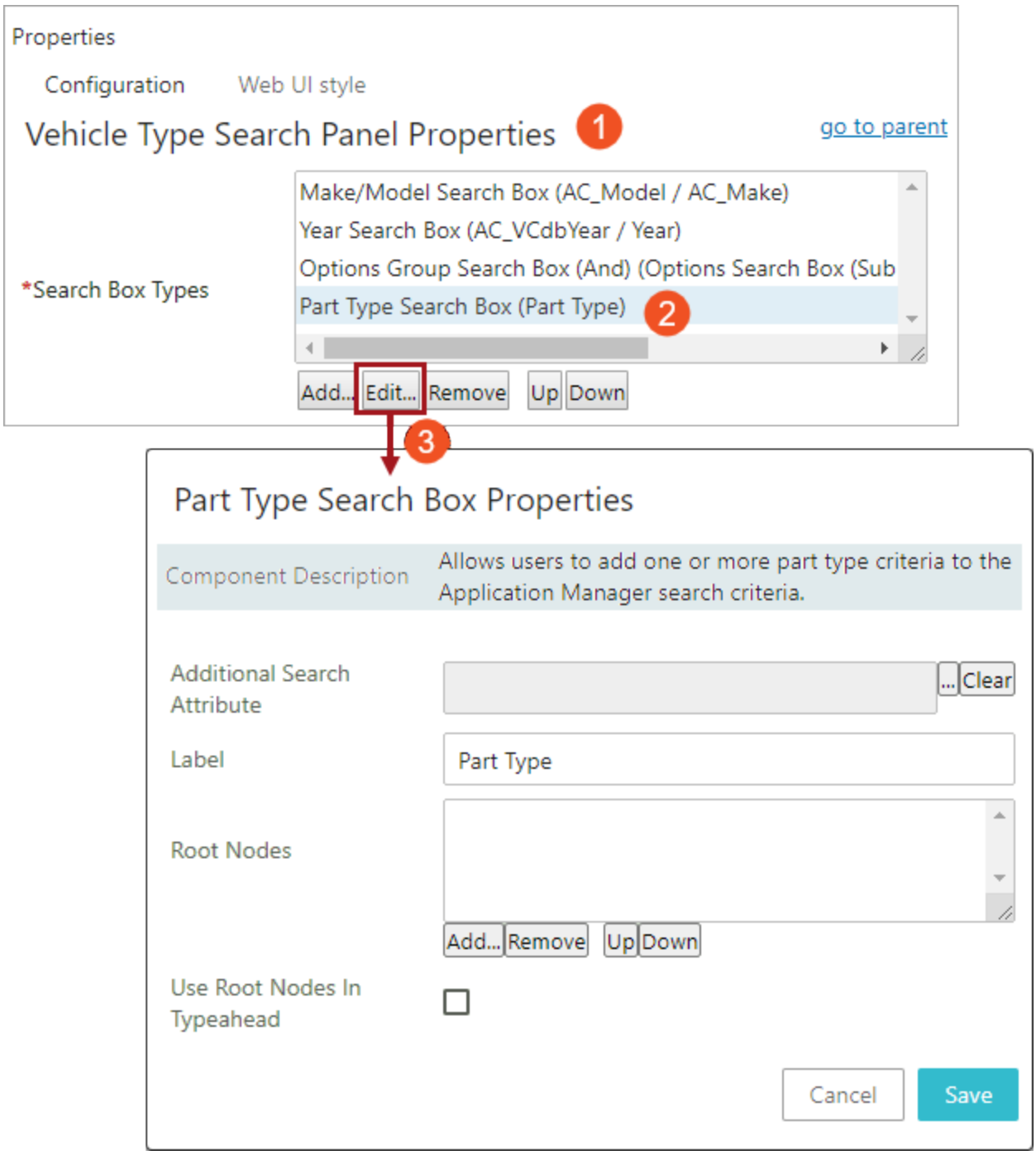



### Editing the Part Type Search Box Label

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.
   - **Additional Search Attribute:** This optional field allows users to add attribute 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 attributes. The selected attributes must be valid for the Part Type object type, and the attribute value must be populated on the part type in order for it to display correctly in the typeahead search field.
   - **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).
   - **Use Root Nodes In typeahead:** Selecting this parameter allows the display of suggestions in the typeahead field of only those objects that belong in the root node defined within the Root Nodes parameter. However, the user needs to be aware that the selection of this parameter would affect the performance of the display of suggestions in the typeahead field.

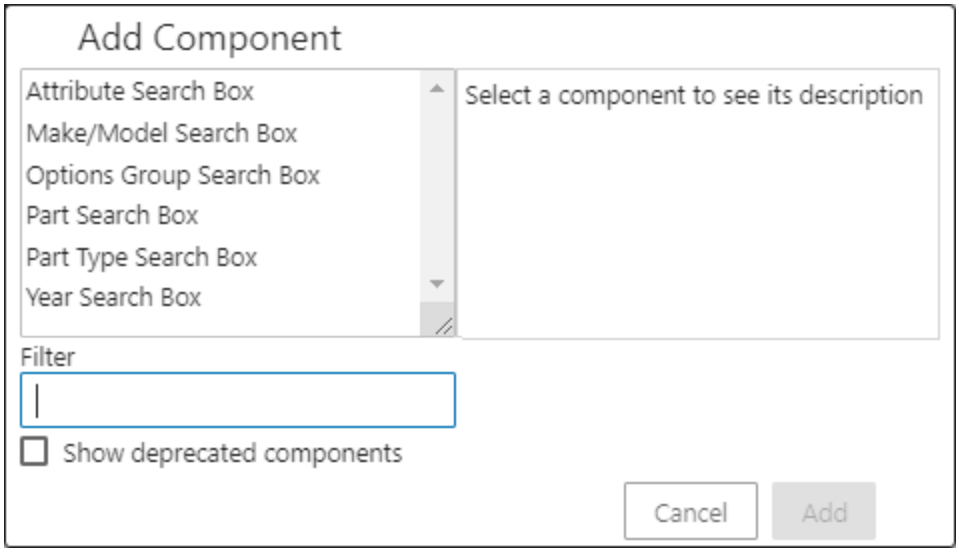

Properties
Configuration
Web UI style
Vehicle Type Search Panel Properties
1
go to parent
*Search Box Types
Make/Model Search Box (AC_Model / AC_Make)
Year Search Box (AC_VCdbYear / Year)
Options Group Search Box (And) (Options Search Box (Sub
Part Type Search Box (Part Type)
2
Add...
Edit...
Remove
Up
Down
3
Part Type Search Box Properties
Component Description
Allows users to add one or more part type criteria to the Application Manager search criteria.
Additional Search Attribute
...
Clear
Label
Part Type
Root Nodes
Add...
Remove
Up
Down
Use Root Nodes In Typeahead
Cancel
Save

## 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.



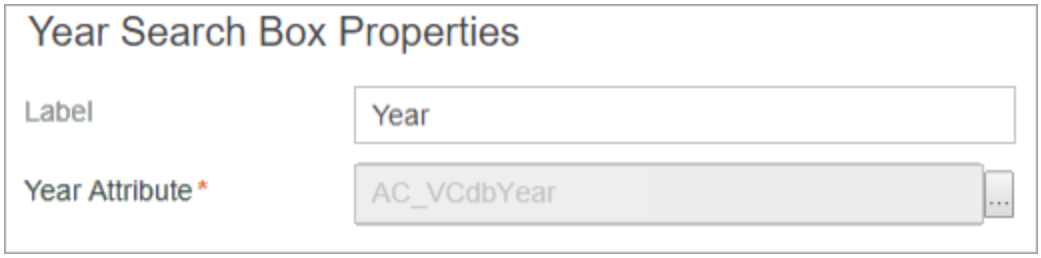

2. Click **Year Search Box** from the list of components.
3. Click the **Add** button, and the Year Search Box Properties will display, as shown below.

**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.'

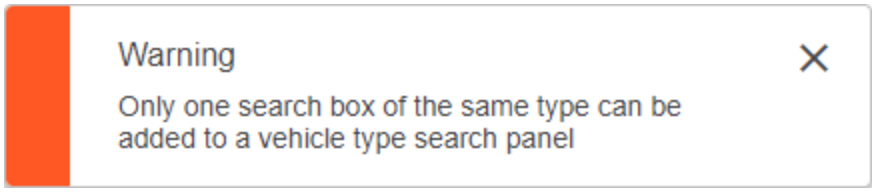
**Year Attribute:** This required field determines the Year Attribute that allows users to add one or more year criteria to their search. All attribute validation base types (e.g., Text, Number, List Of Values) are allowed. Click the ellipsis button (...) to find and select the necessary Year Attribute. 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.

4. 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.



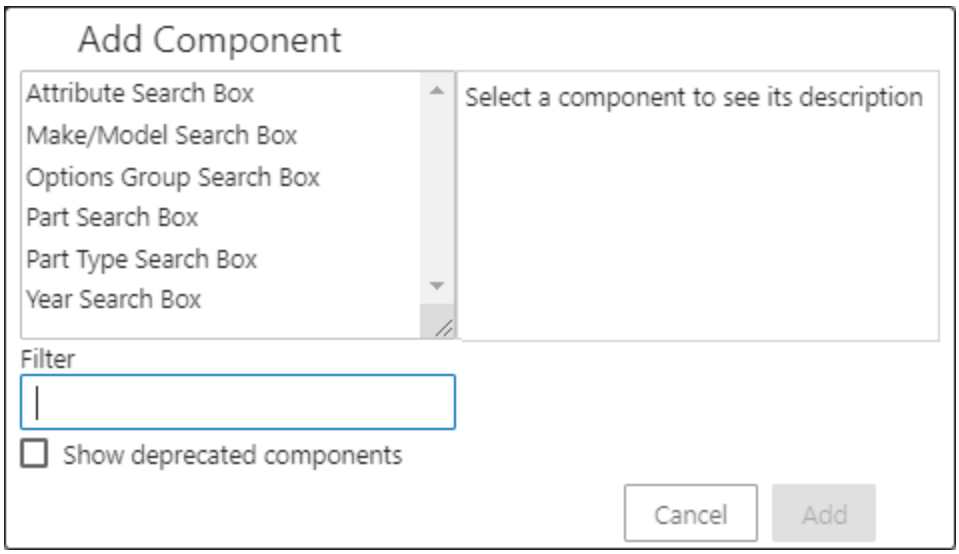

Because each search box type can only be configured once for each Application Manager, if a second search box of the same type is added, a Warning dialog will display as shown below, and the Search Box Types parameter will remain unchanged.




## Configuring Part Search Box

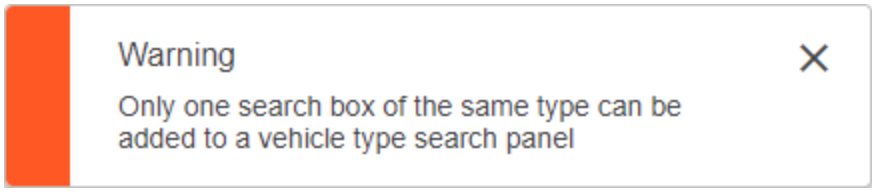
Steps for adding the Part Search Box component to a Vehicle Type Search Panel are below.

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 **Part Search Box** from the list of components.
3. Click the **Add** button, and the Vehicle Type Search Panel Properties will display with the Part Search Box added to the Search Box Types parameter.

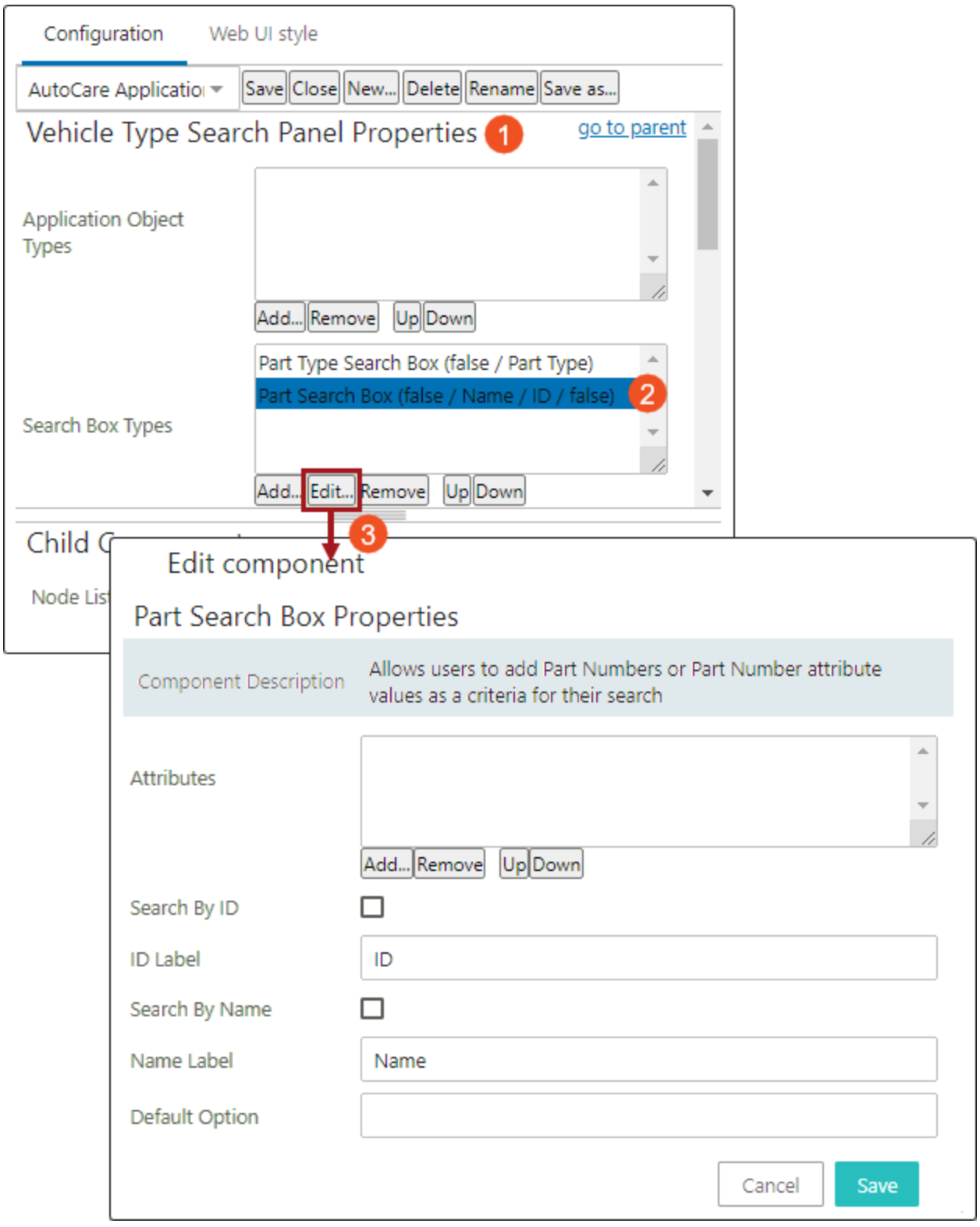
   Because each search box type can only be configured once for each Application Manager, if a second search box of the same type is added, a Warning dialog will display as shown below, and the Search Box Types parameter will remain unchanged.




### Editing the Part Search Box Parameters

The Part Search Box parameters 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 Search Box**.
3. Click the **Edit** button, and the Part Search Box Properties dialog will display as shown below.
4. Edit the following parameters as necessary.


Configuration
Web UI style
AutoCare Applicatio
Save
Close
New...
Delete
Rename
Save as...
Vehicle Type Search Panel Properties
1
go to parent
Application Object Types
Add...
Remove
Up
Down
Part Type Search Box (false / Part Type)
Part Search Box (false / Name / ID / false)
2
Search Box Types
Add...
Edit...
Remove
Up
Down
3
Edit component
Part Search Box Properties
Component Description
Allows users to add Part Numbers or Part Number attribute values as a criteria for their search
Attributes
Add...
Remove
Up
Down
Search By ID
ID Label
ID
Search By Name
Name Label
Name
Default Option
Cancel
Save

**Attributes:** This optional field allows users to add one or more attribute 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 attributes. The selected attributes must be valid for the Part object type, and the attribute value must be populated on the Part in order for it to display correctly in the typeahead search field.

**Search By ID:** This optional field allows users to include the ID criteria to their search. If selected, the ID criteria will be displayed in the dropdown and the user has to type the part ID within the typeahead field of the search box in order for it to display correctly.

**ID Label:** This optional field allows users to provide a label that best describes the part ID search option. This label will be used to be displayed in the Part Search Box drop-down list. If this field is left empty, the default value that will be displayed is 'ID.' This value can be used within the 'Default Option' parameter.

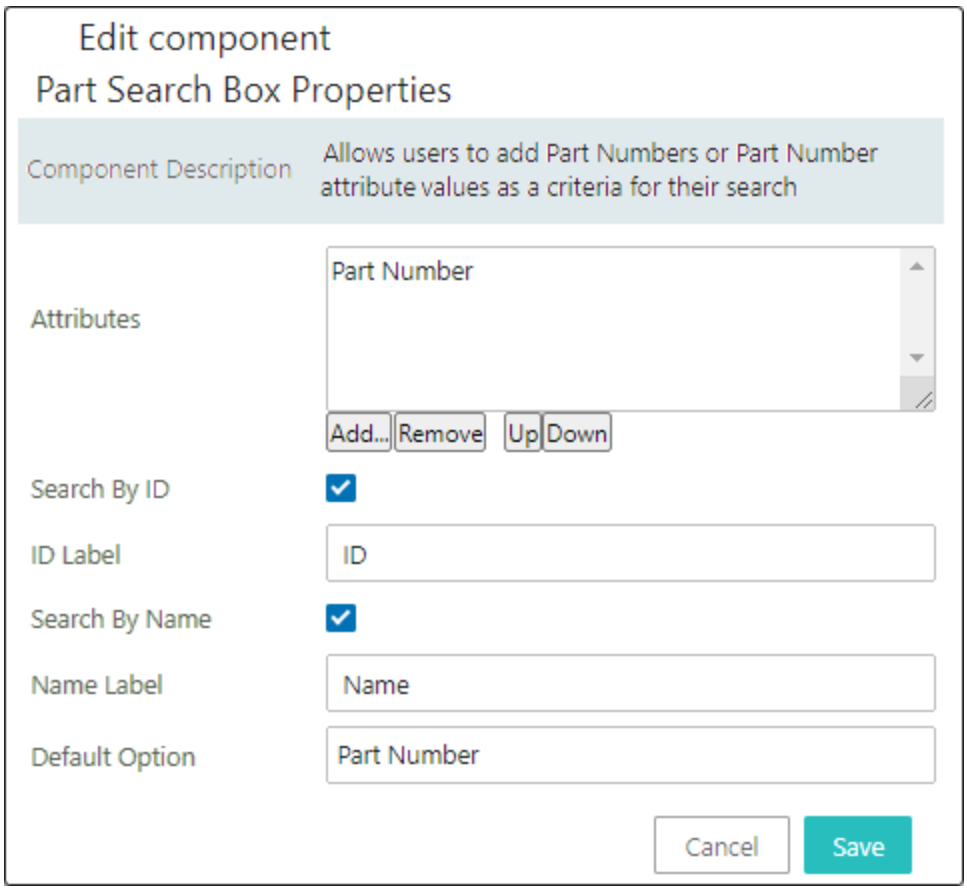
**Search By Name:** This optional field allows users to include the Name criteria to their search. If selected, the Name criteria will be displayed in the dropdown and the user has to type the part name within the typeahead field of the search box in order for it to display correctly.

**Name Label:** This optional field allows users to provide a label that best describes the part Name search option. This label will be used to be displayed in the Part Search Box drop-down list. If this field is left empty, the default value that will be displayed is "Name." This value can be used within the 'Default Option' parameter.

**Default Option:** This optional field allows for a default option to display within the Part Search Box drop-down list when users access the Application Manager. The value within this parameter must match a value for an attribute Name for the attribute that is configured in the 'Attributes' parameter, or a value that has been defined within the 'ID Label' or 'Name Label' parameter.

5. Once the parameters are populated, click the **Save** button to return to the Vehicle Type Search Panel Properties.

   Below is an example of a configured Part Search Box type for the AutoCare standard.


Edit component
Part Search Box Properties
Component Description
Allows users to add Part Numbers or Part Number attribute values as a criteria for their search
Attributes
Part Number
Add...
Remove
Up
Down
Search By ID
ID Label
ID
Search By Name
Name Label
Name
Default Option
Part Number
Cancel
Save

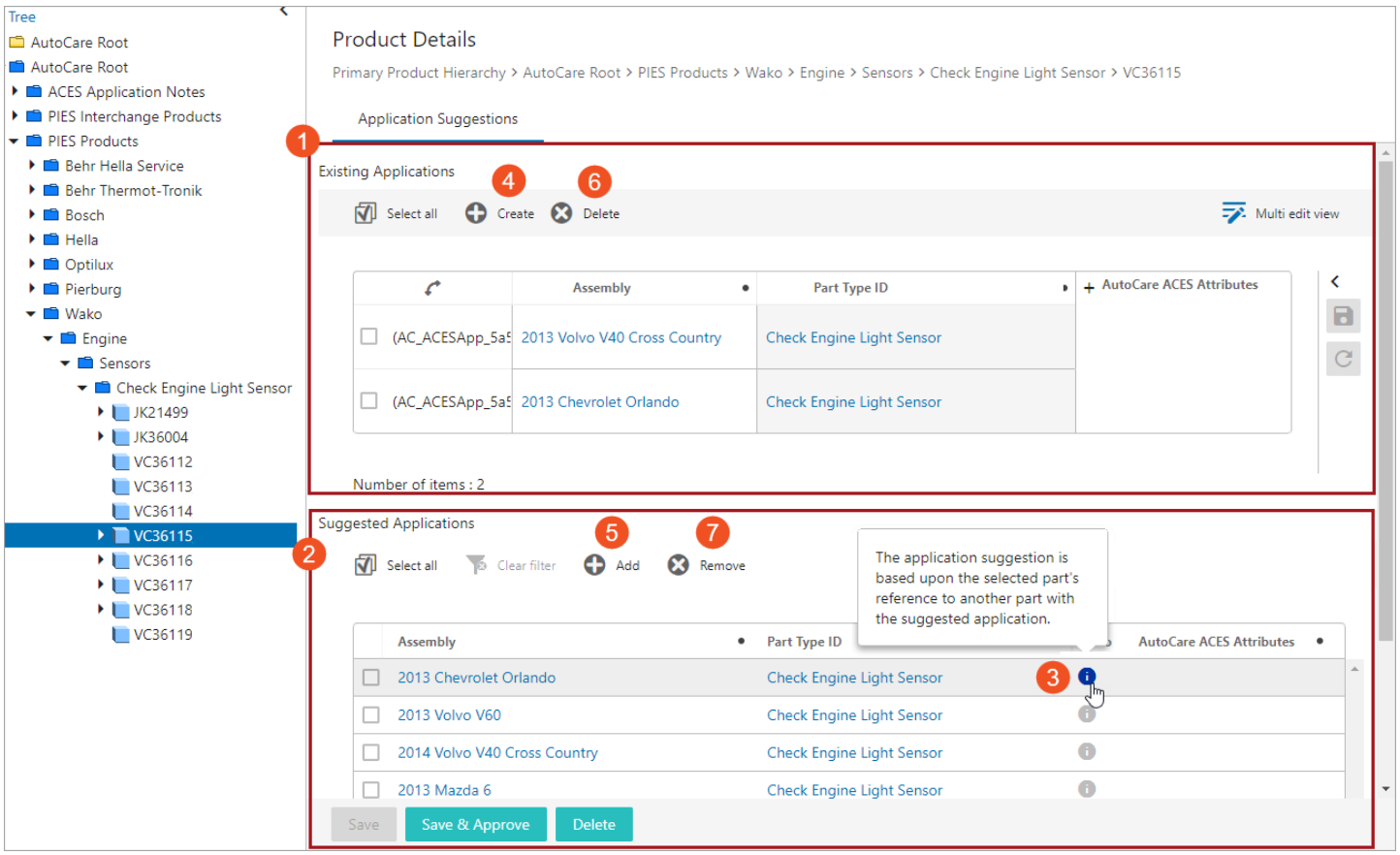
# Application Suggestion Solution

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:

- View existing applications and related data.
- View Target (competitor and/or OE part number) part's application suggestions and related data.
- View information about the suggested application.
- Add applications manually.
- Add applications from suggestions.
- Remove existing applications.
- 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

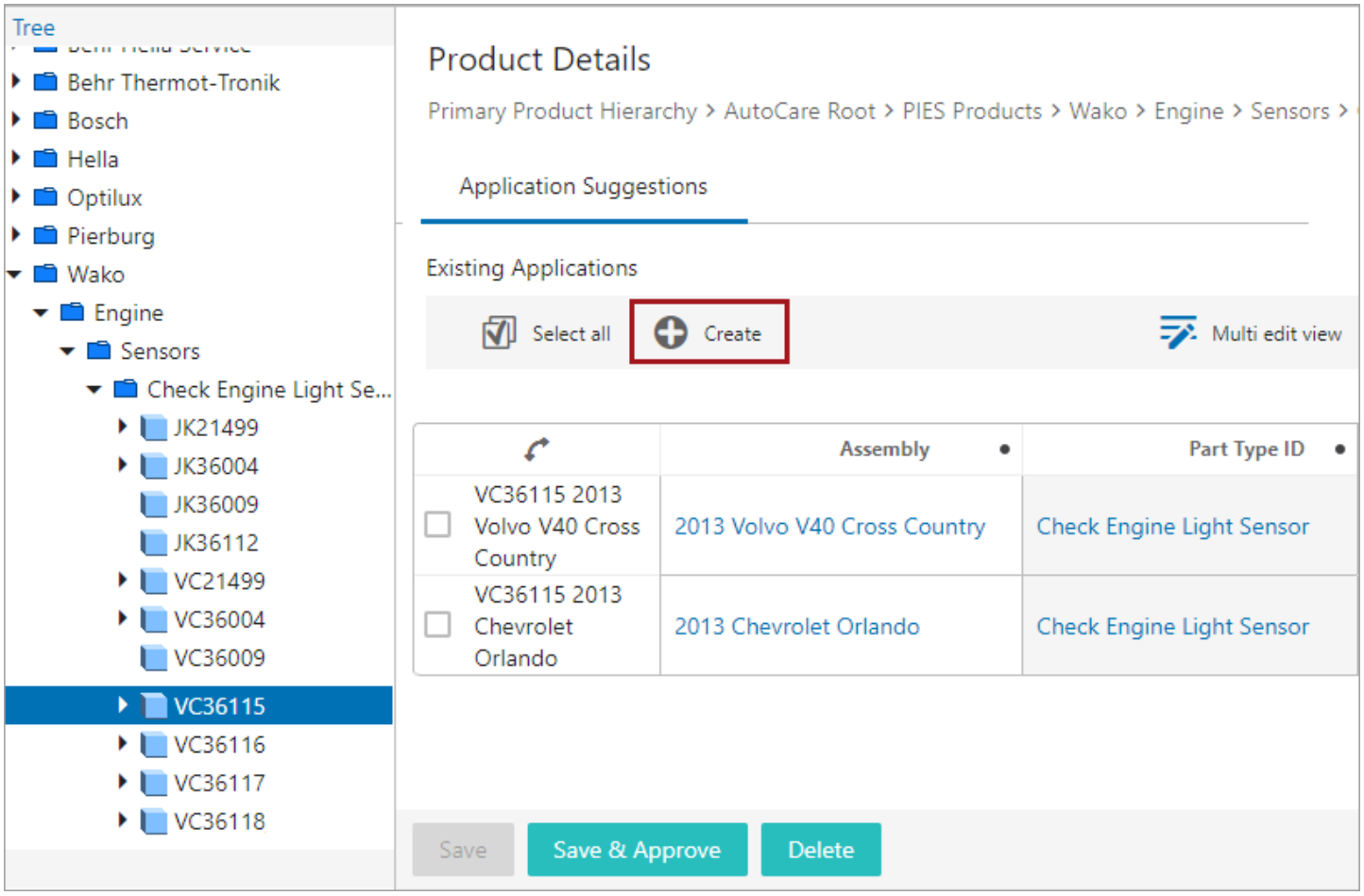
'Application Suggestions' components. The 'Part Application Editor' component can be configured to display existing applications. Whereas, the 'Application Suggestions' component can be configured to display suggested applications.

For more details on configuring screen mappings so that a 'Product Details' screen will display when a product is selected within the Tree navigator, refer to the **Mappings** topic within the **Main Properties Overview** section of Web User Interfaces 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.

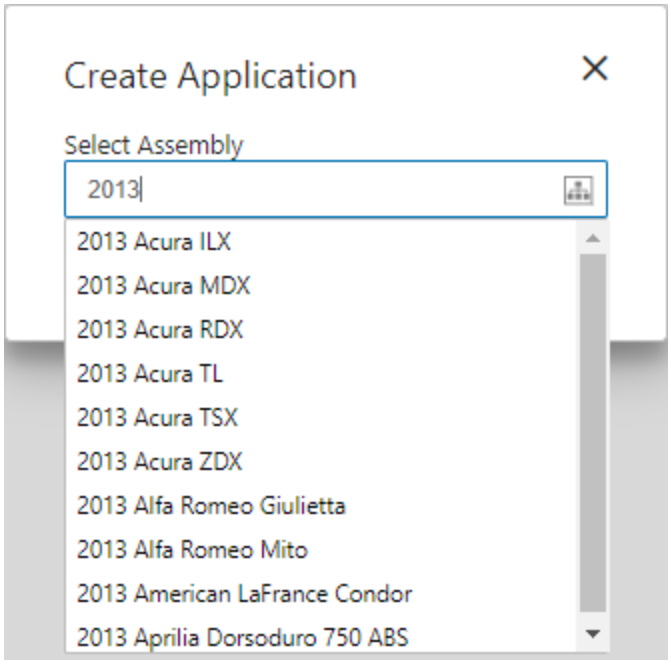



When a user clicks on the 'Create' button, a Create Application dialog displays prompting users to select the vehicles / assemblies. Once the vehicle / assembly name (or ID) is typed in the typeahead field, available list of vehicles / assemblies will display in the dropdown list. Alternatively, the 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.

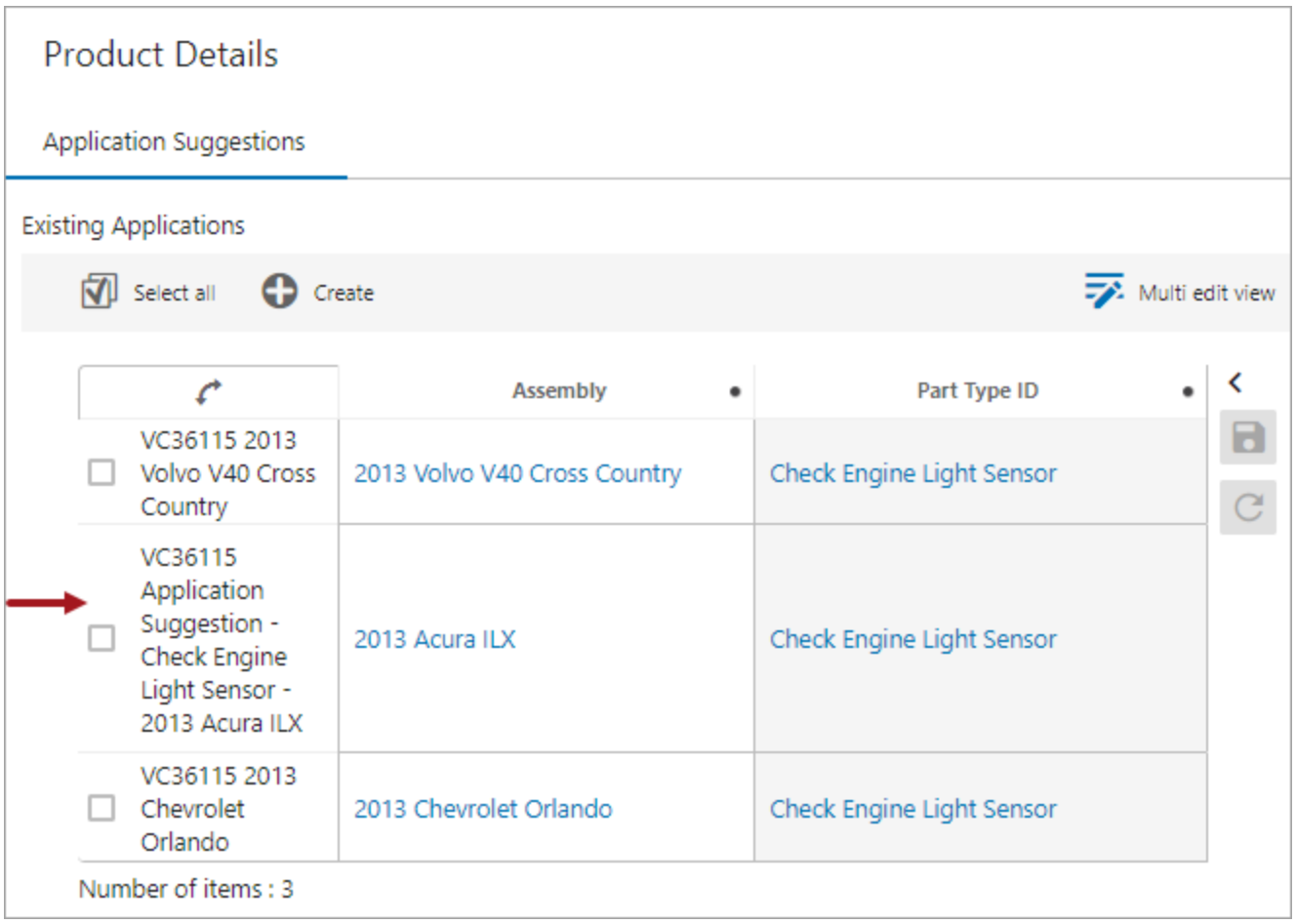
Following are the IDs of vehicles / assemblies that are valid to be selected across different standards.

- 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.




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.

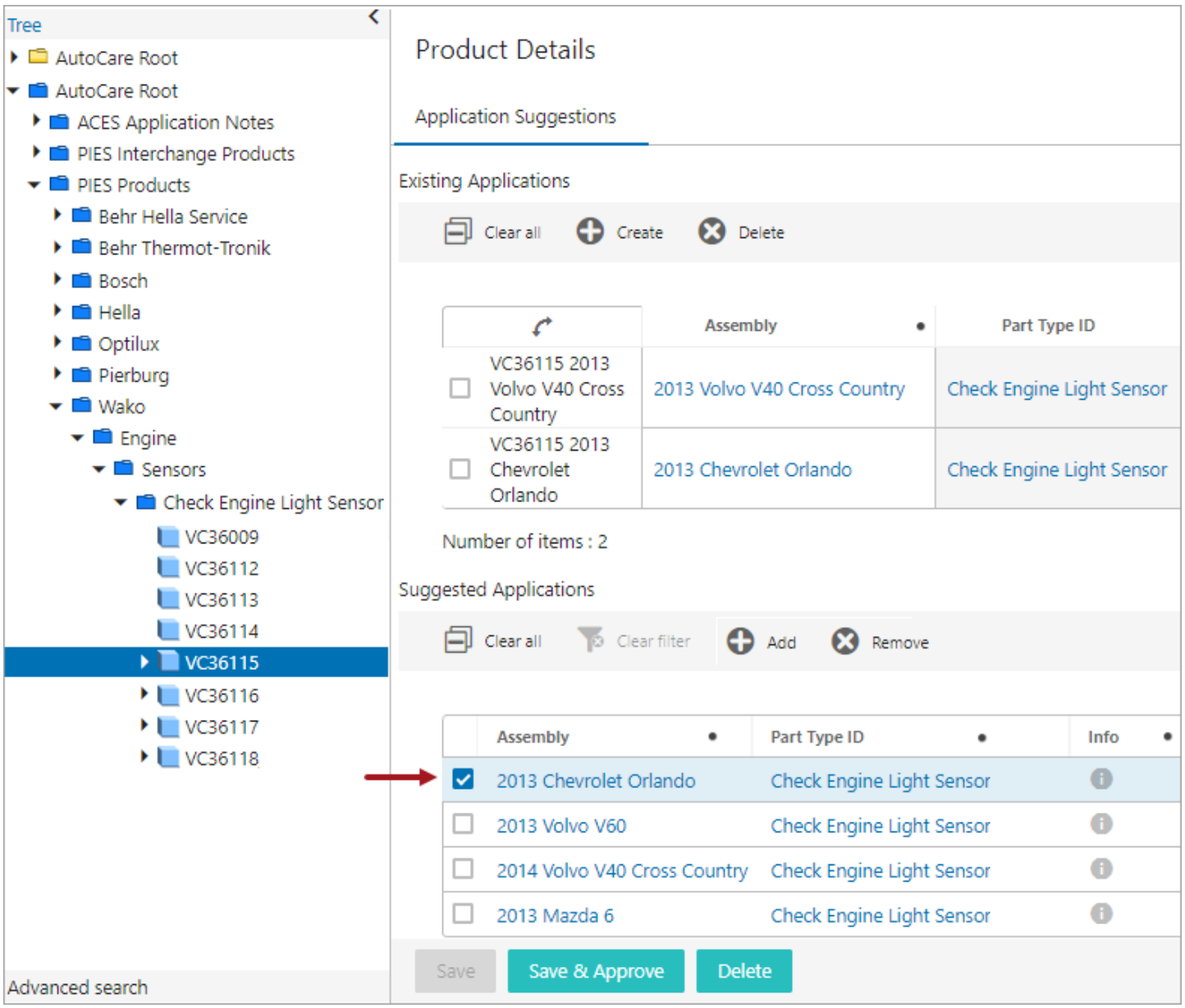
# Adding Applications from Suggestions

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.

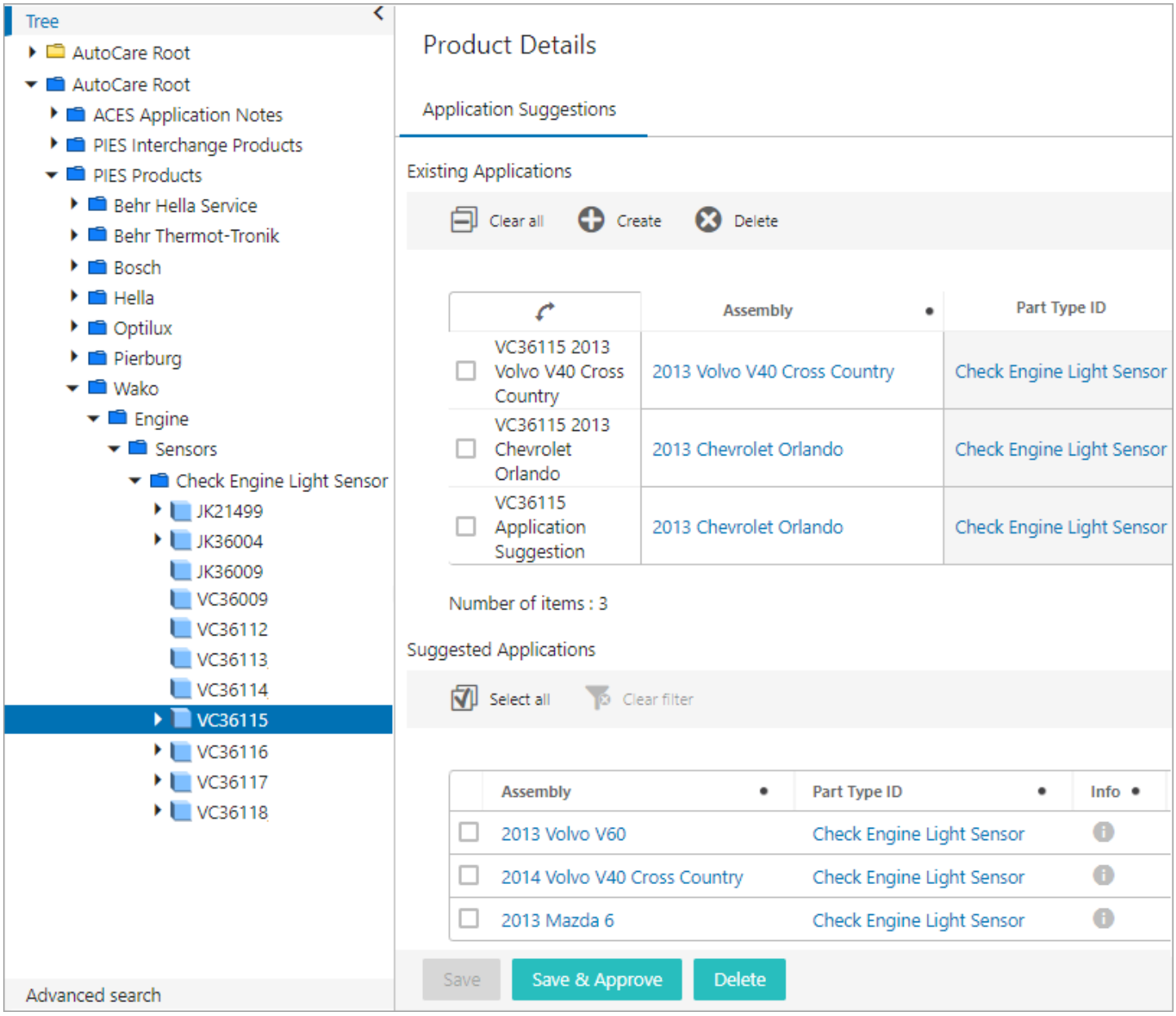
When any potential application records are displayed in the Application Suggestions list, selecting an application record from the suggestion list will activate and display the **Add** button (+) at the component tool bar. Clicking the **Add** button will add the selected application record(s) to the Part Application Editor list.

Clicking the **Remove** button (×) will remove the suggested application record from the Application Suggestions list. For more information, refer to 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.




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).

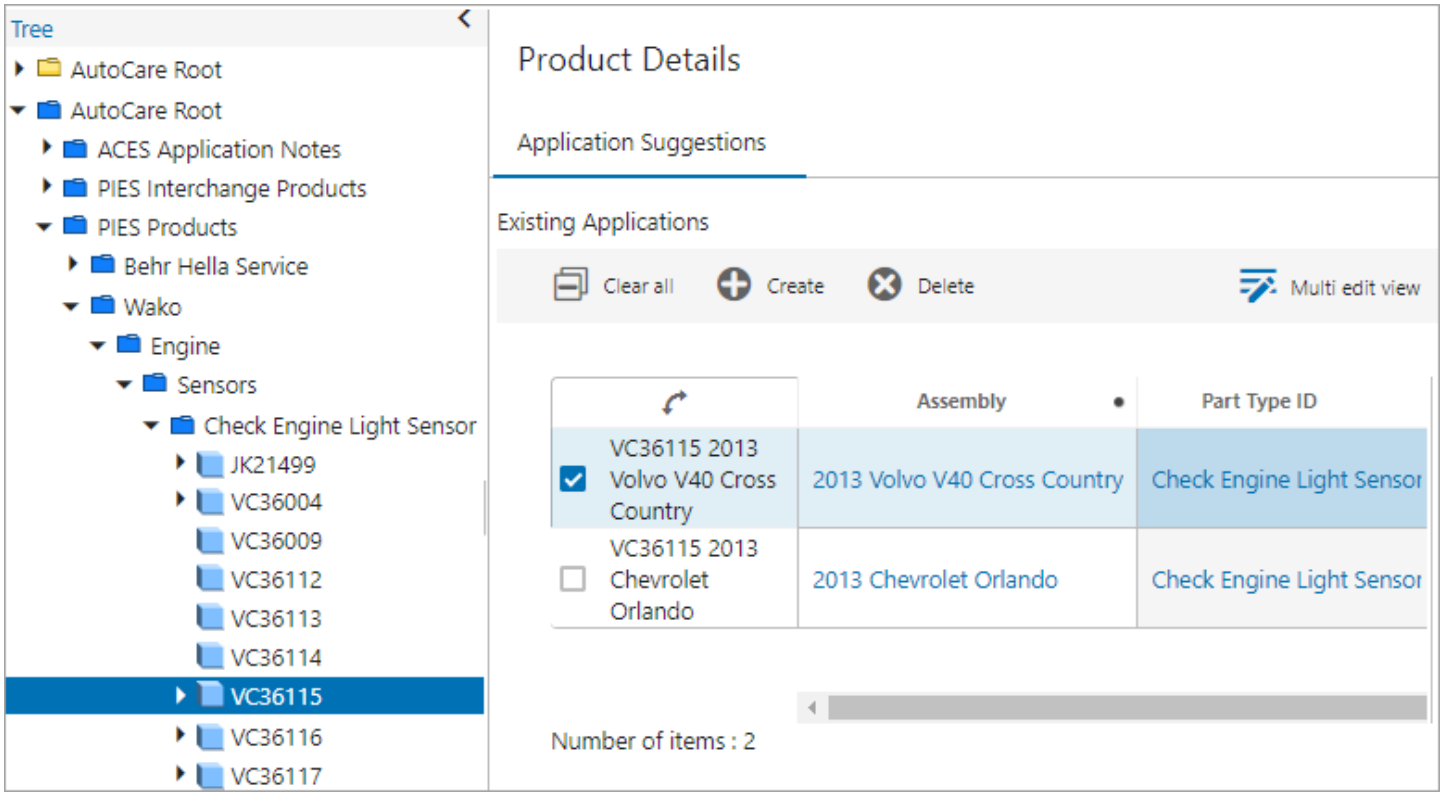



Once the application record is added to the selected part and now displays within the Part Application Editor table, the same application record will no longer display within the Application Suggestions list.

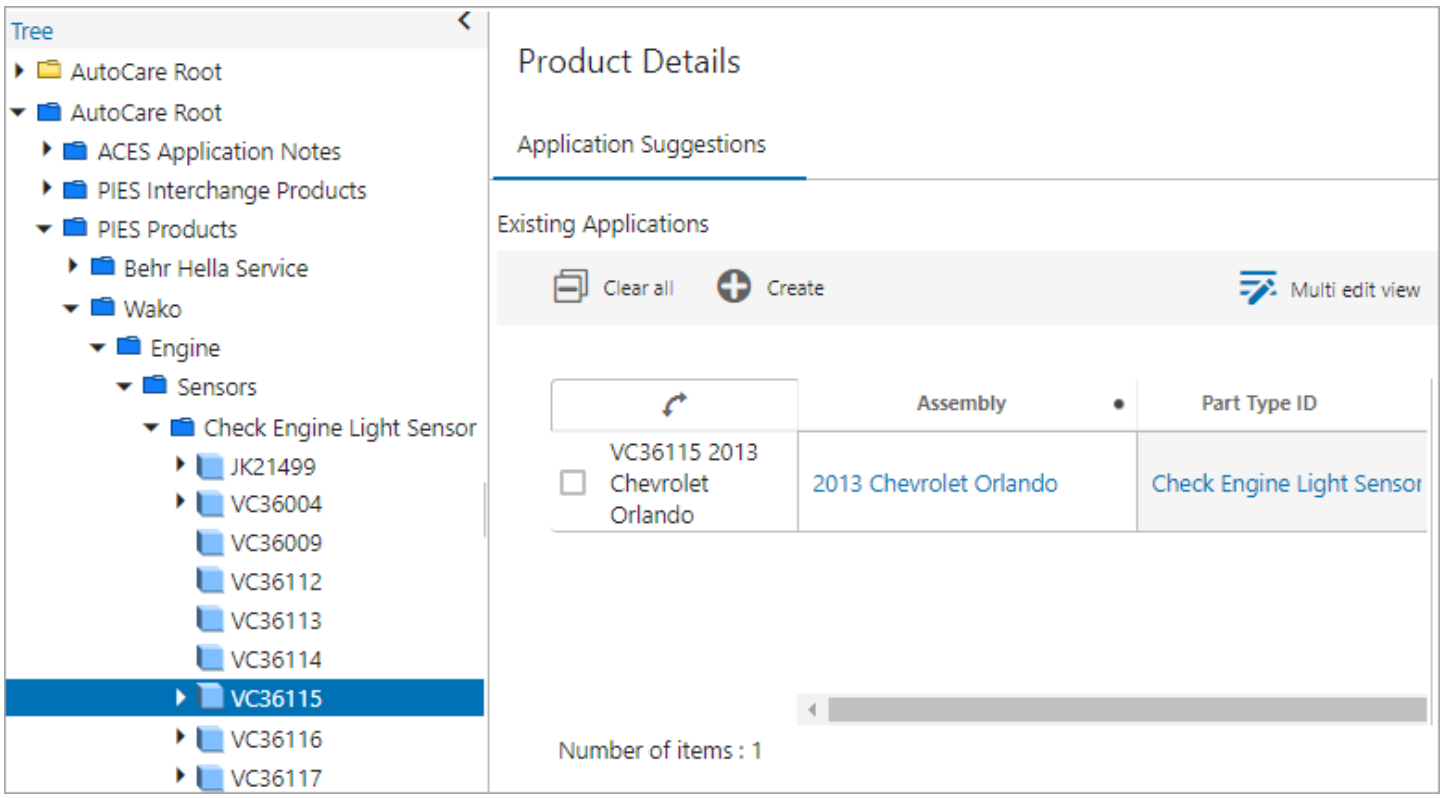
# Removing Existing Applications

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.




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).

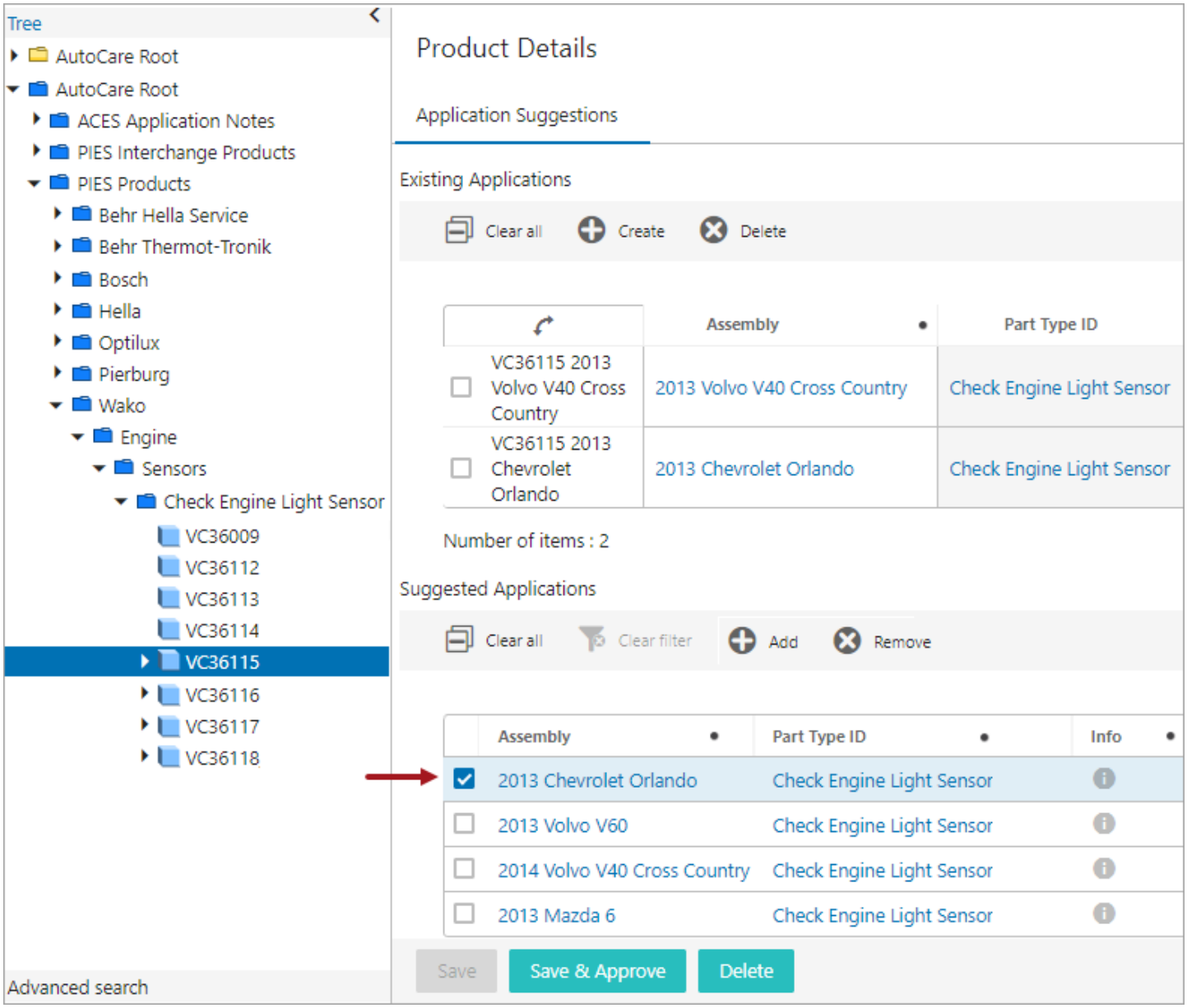



If an application record is removed from the Existing Applications list, and it previously displayed within the Suggested Applications list, then the deleted application record will display within the suggestions list. However, if the removed application record was not previously displayed within the suggestions list (i.e., manually added), then it will not display within the suggestions list.

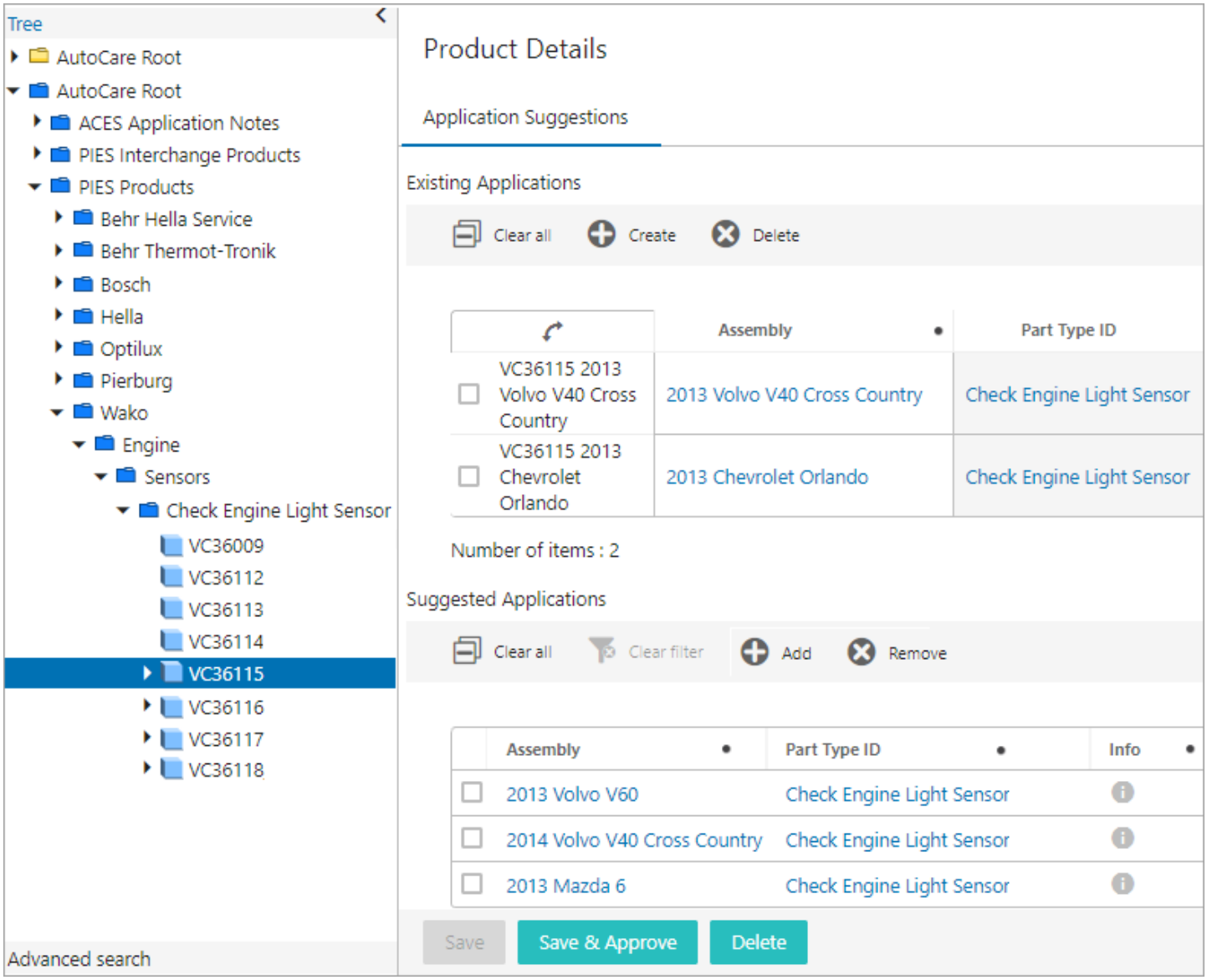
# Removing Application from Suggestions List

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 ( ✕ ) 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** ( ✕ ) button thereby allowing users to remove an application record for the selected part.




In the example, after removing the application record '2013 Chevrolet Orlando' from part 'VC36115' the Suggested Applications list displays as shown below.




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, refer to the **Application Suggestion Solution** topic.

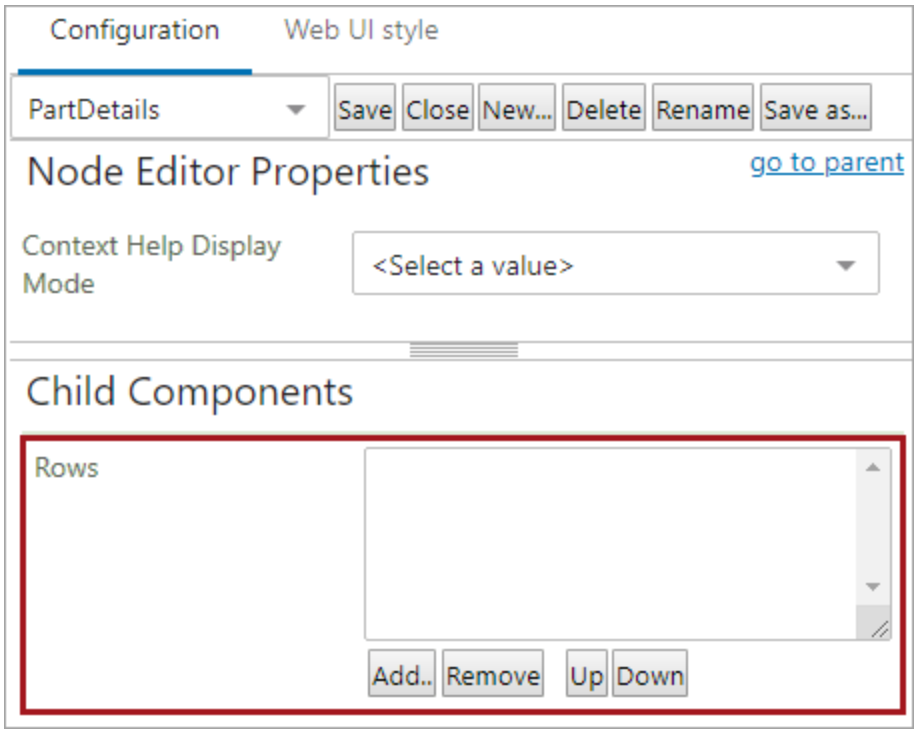
## Prerequisites

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, refer to the **Node Editor Component** topic within the **Using a Web UI** section of **STEP Online Help**.

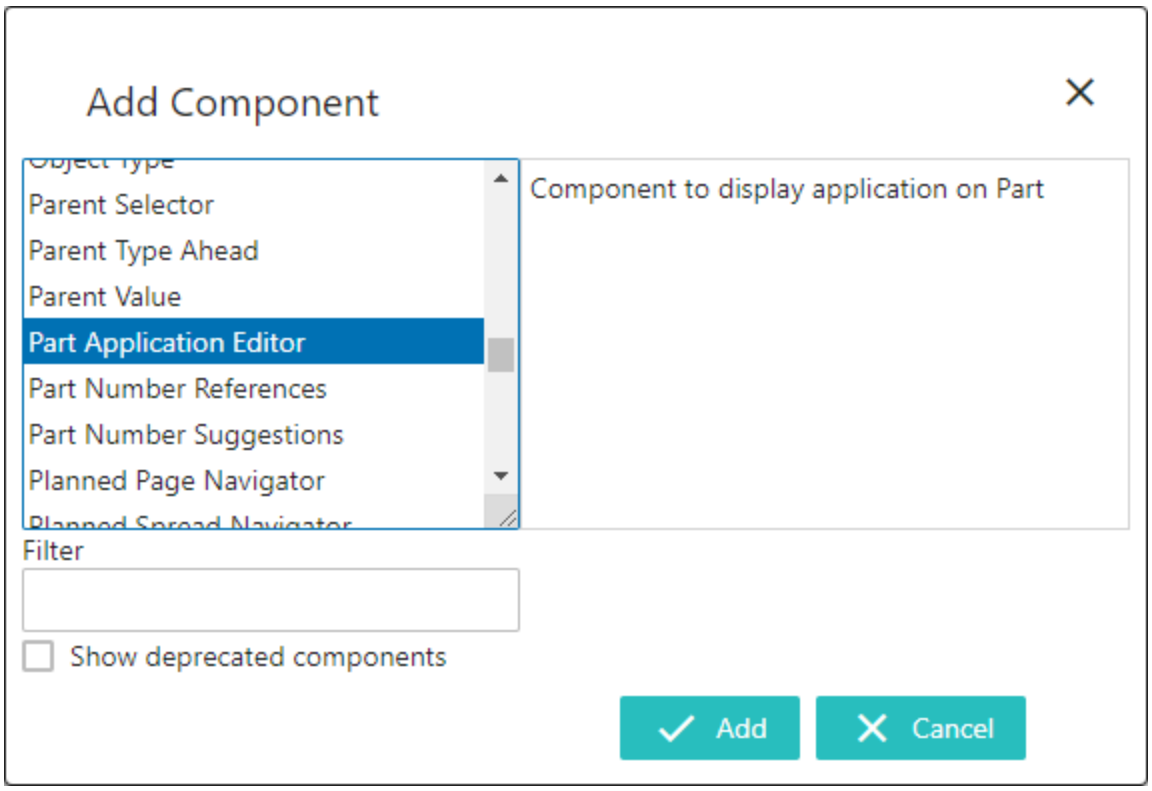
## Configuration Steps

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.

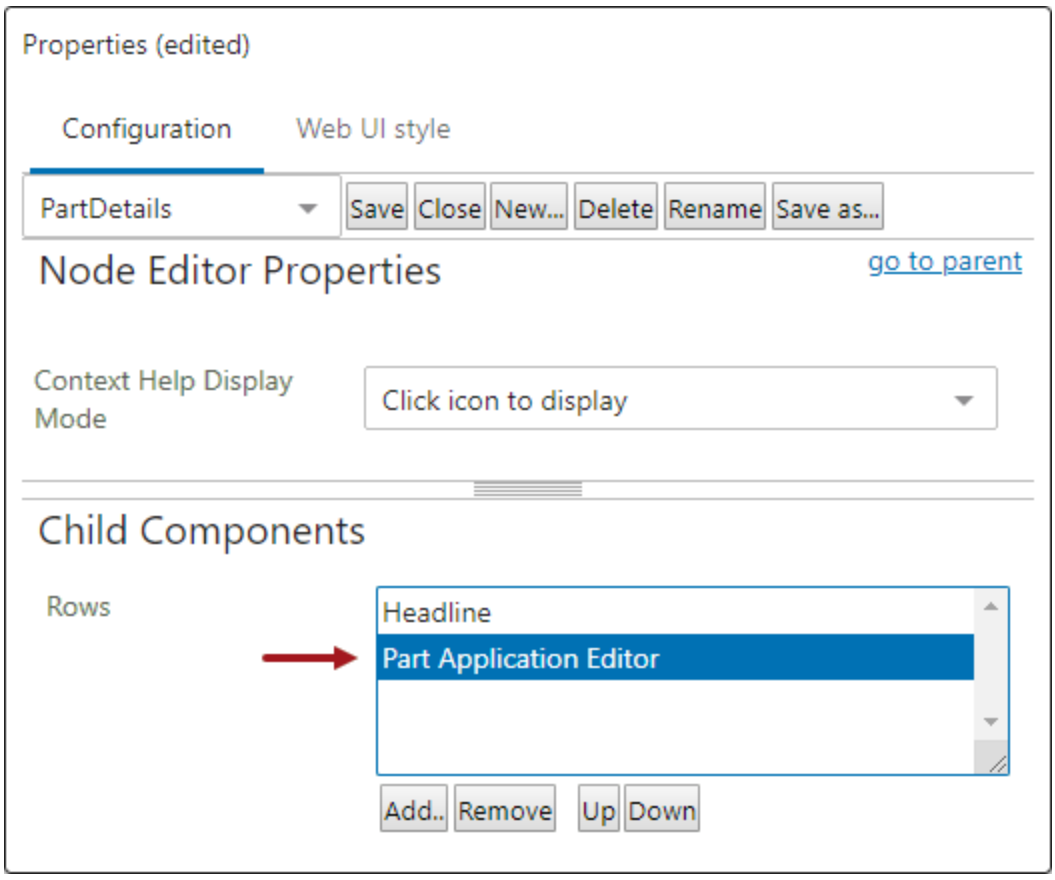
1. In the Web UI Designer, go to Node Editor Properties > Child Components > Rows parameter for the screen that needs to display application record information.



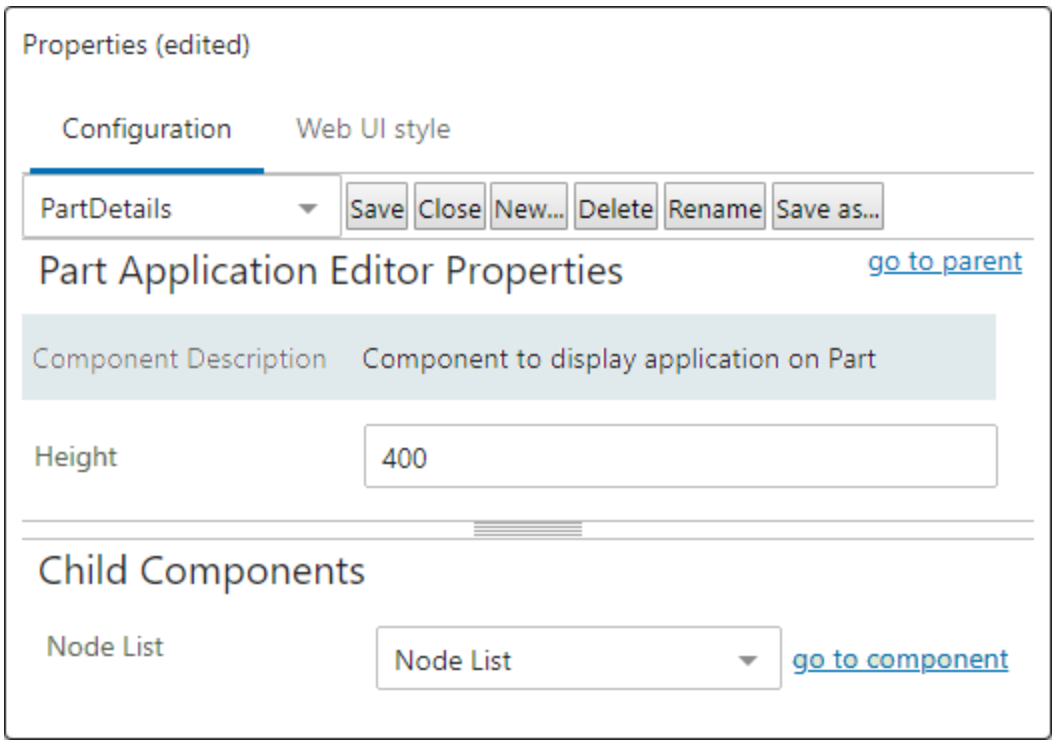

2. Click the **Add** button for the Child Components Rows parameter, and the Add Component dialog will display.




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:




**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, refer to 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.

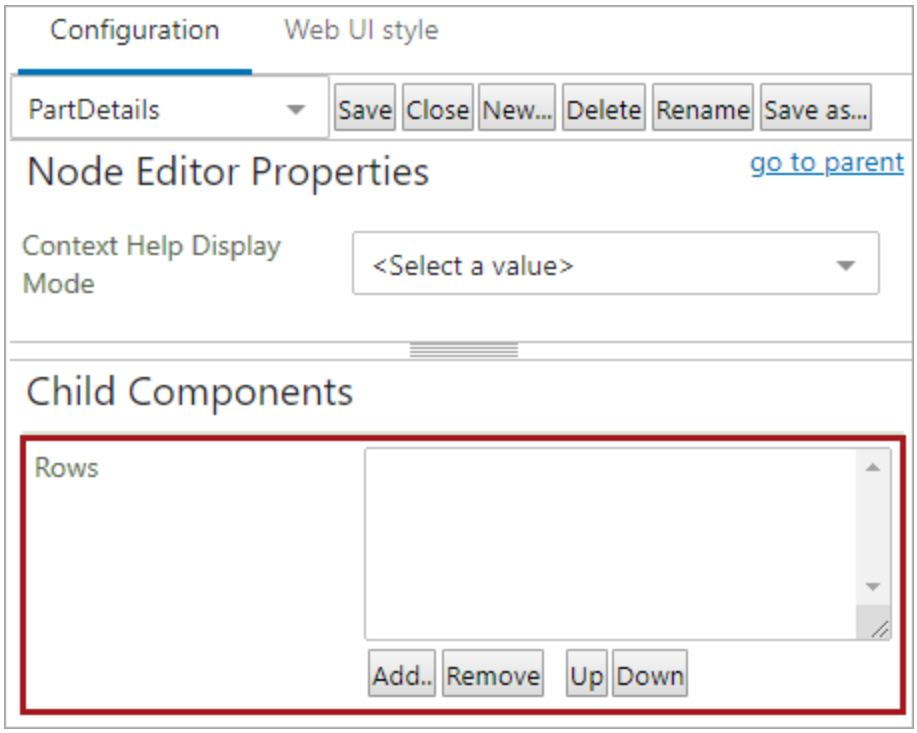
# Configuring the Application Suggestions Component

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, refer to the **Application Suggestion Solution** topic.

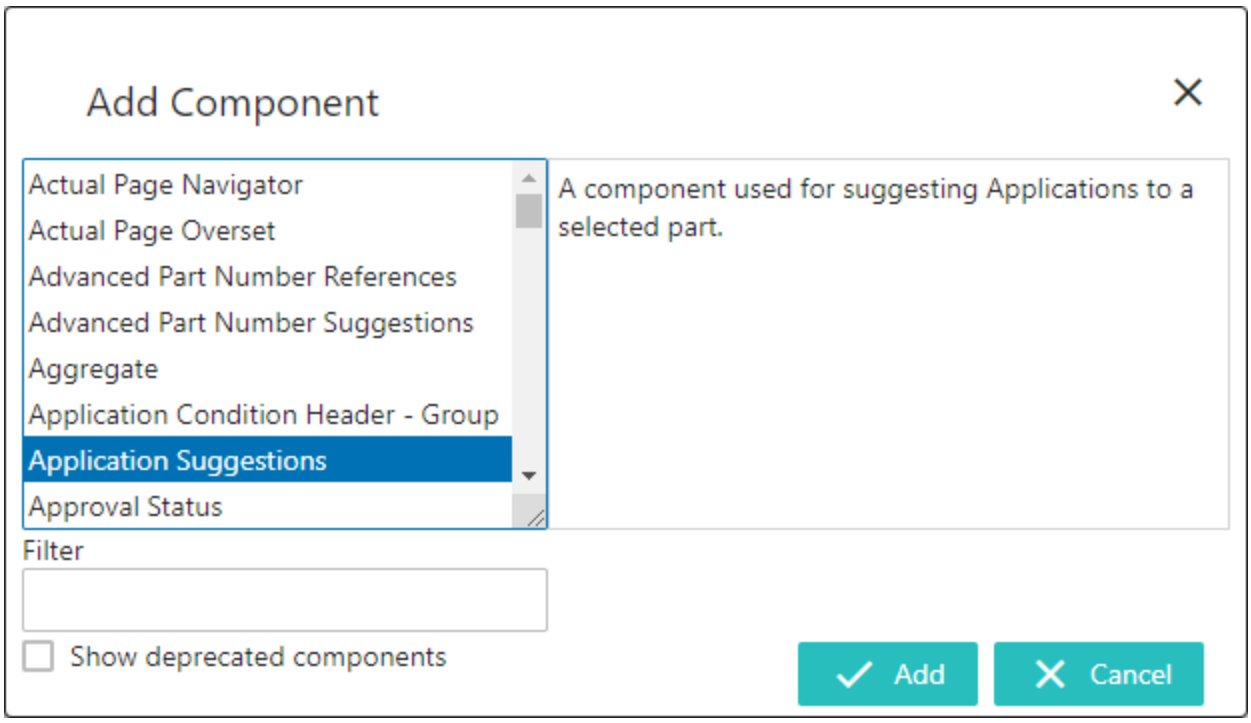
## Configuration

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.

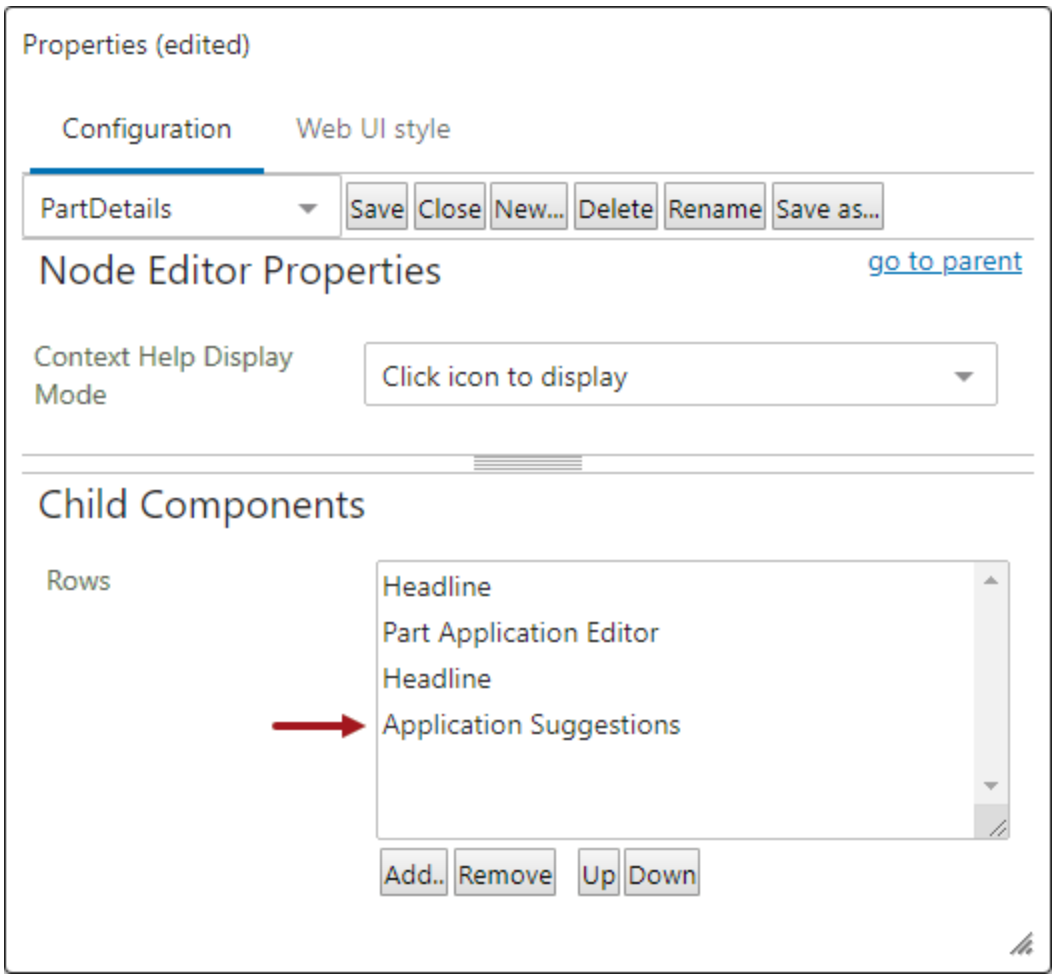
1. In the Web UI Designer, go to Node Editor Properties > Child Components > Rows parameter for the screen that needs to display application record suggestions.



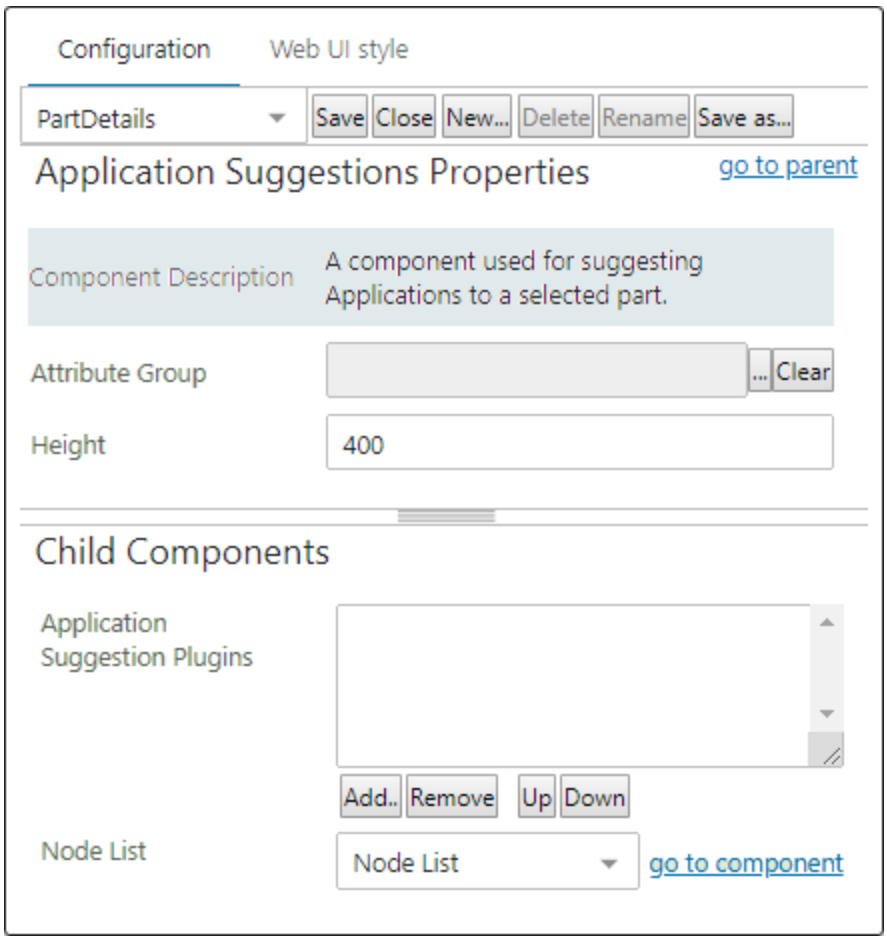

2. Click the **Add** button, and the Add Component dialog will display.




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:



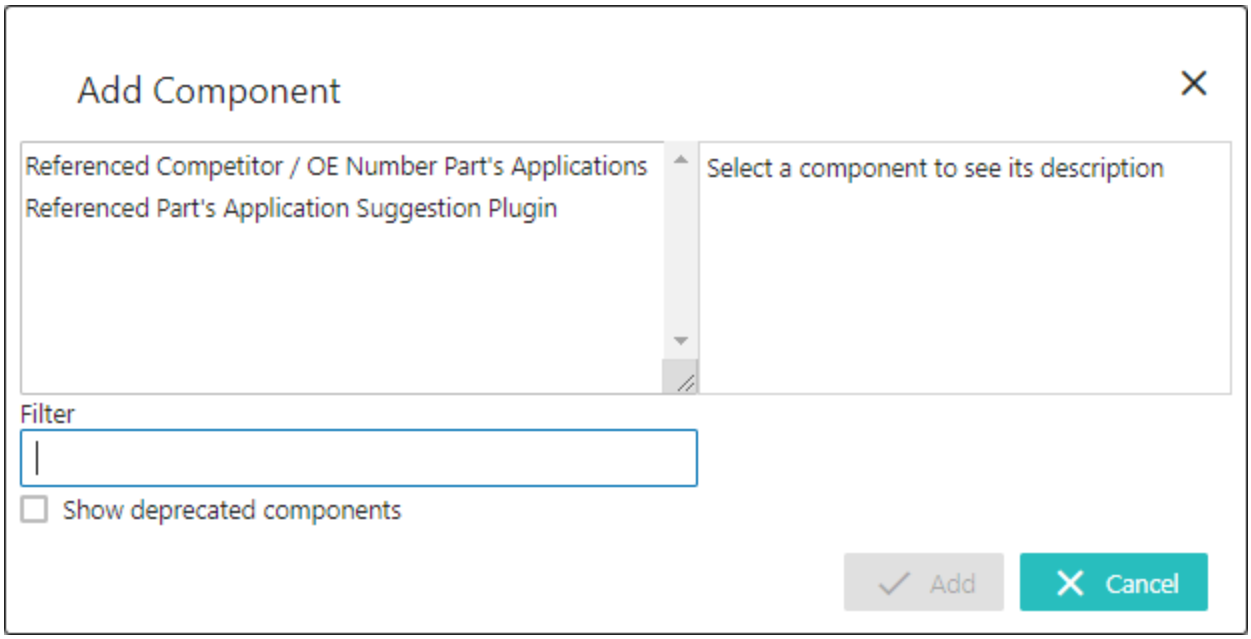

**Height:**This optional parameter specifies the visual height of the Advanced Part Number Suggestions component in pixels. By default, this parameter is pre-populated with a value '400.' Users can enter a numeric value to alter height of the component.

**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

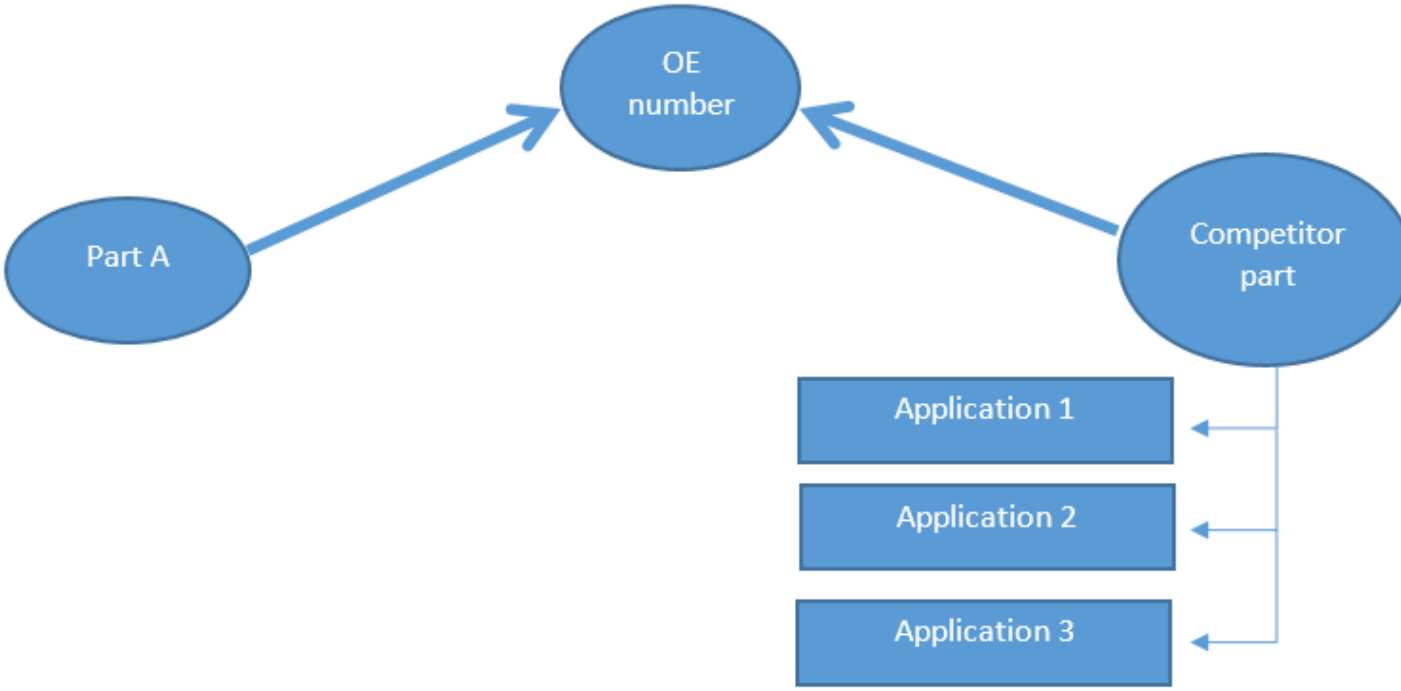
or different information about the listed objects to be displayed. The behavior of many of the various actions can also be further configured. For more information, refer to the **Node List Component** topic within the **Web User Interfaces** section of **STEP Online Help**.

**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.

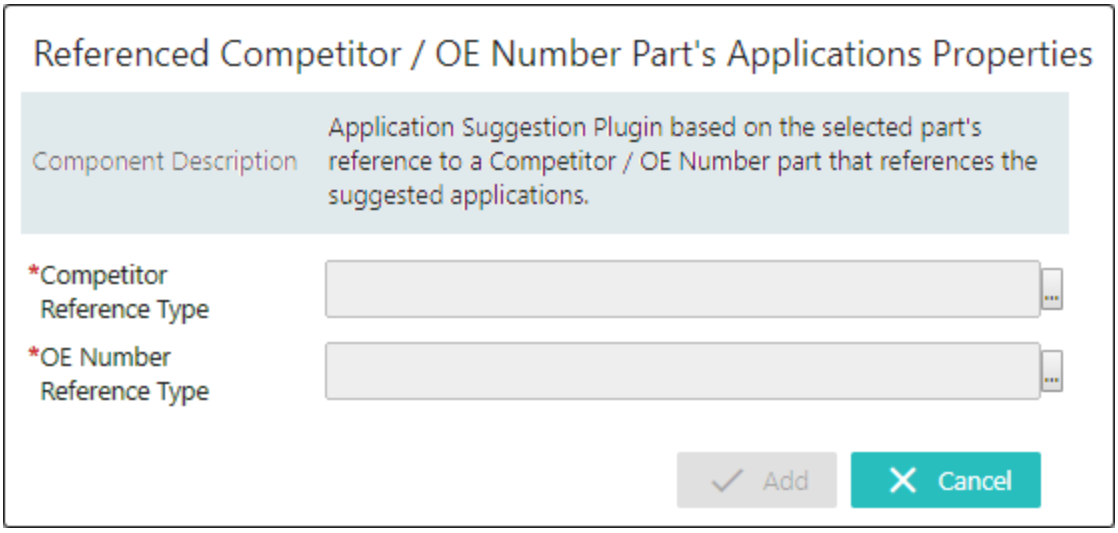



The Application Suggestion Plugins options are the following:

**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.




- **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

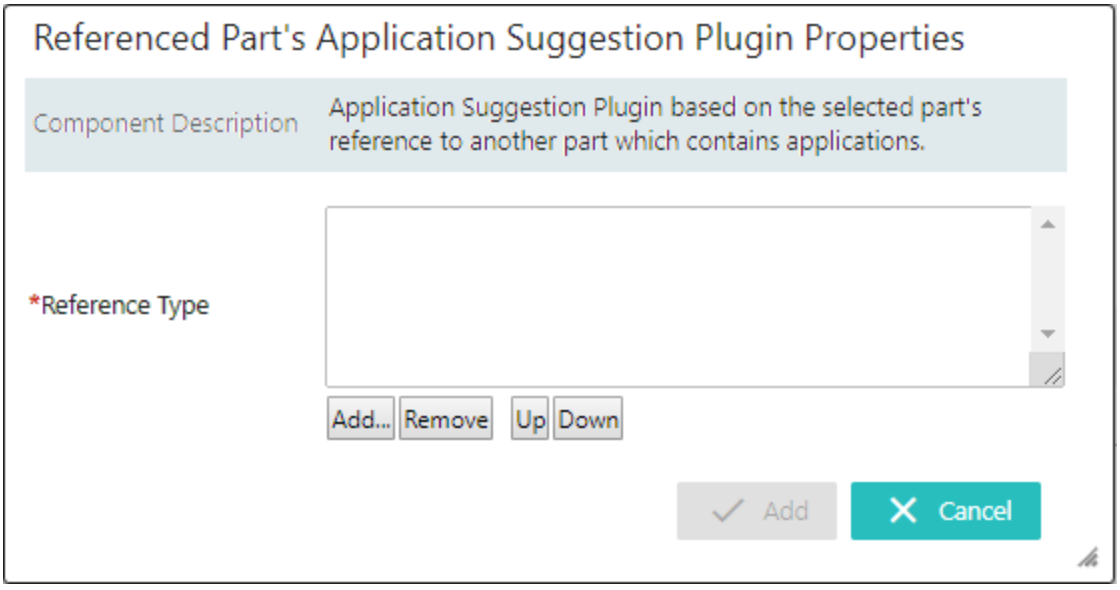
the targets. In order to have this parameter configured, user must establish a reference link between the Competitor to OE Number.

- 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.



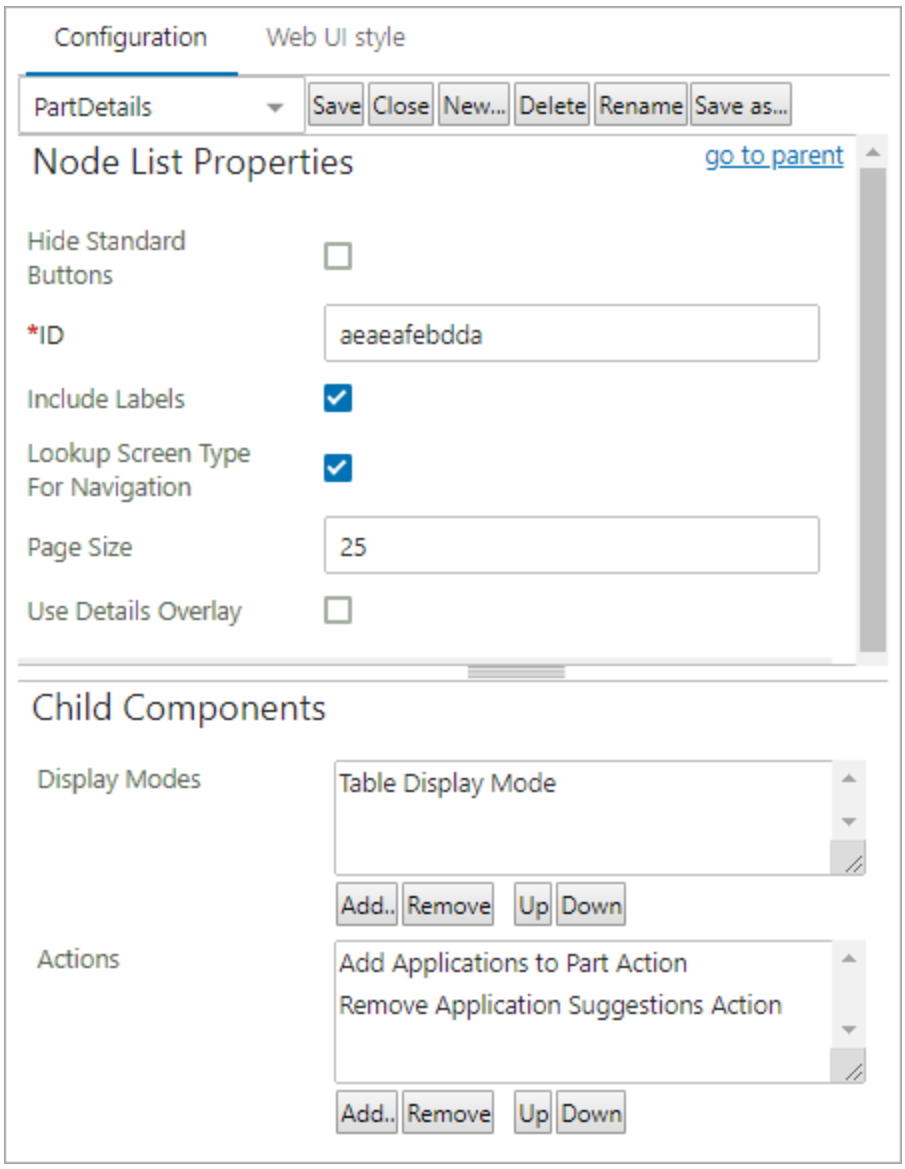

**Reference Type:** By default, this required parameter is blank. However, it is mandatory to add at least one Product Reference Type so that when a part is selected, the reference types listed within this parameter are followed and display the application records of the target part.

- 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'

**Referenced By Part's Application Suggestion Plugin:** This plugin is used to suggest an other part's application records. The selected part must be referenced by a part so that the source part's application records are displayed as suggestions. 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.

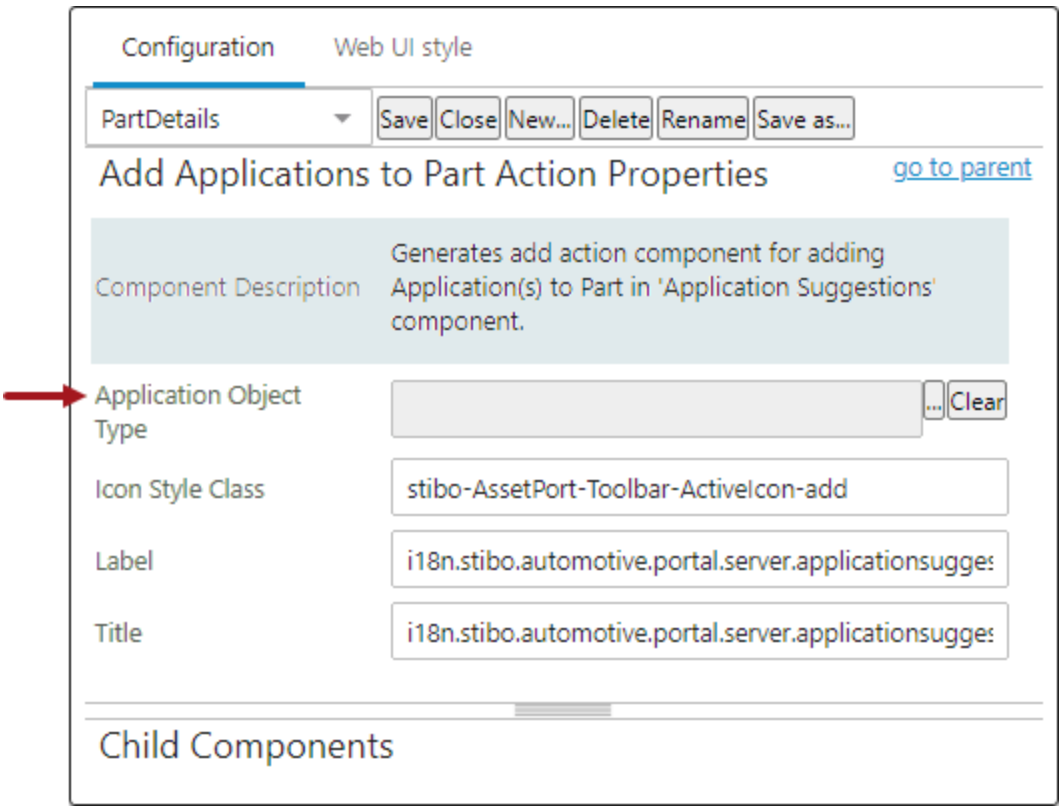
**Reference Type:** By default, this required parameter is blank. However, it is mandatory to add at least one Product Reference Type so that when a part is selected, the component follows the reference type and suggests the application records of the source part that references the selected part.

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. Users can double click on the 'Add Applications to Part Action' to further configure the Application Object Type, Icon style, Title, and Label of the action button if necessary.

**Note:** It is essential to configure the Application Object Type parameter with the right application object type. If there are multiple object types for an application record in the system and the 'Application Object Type' parameter is blank, then the system will throw an error when an application is added from the application suggestion list.




**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.

# Competitor and OE Numbers Solution

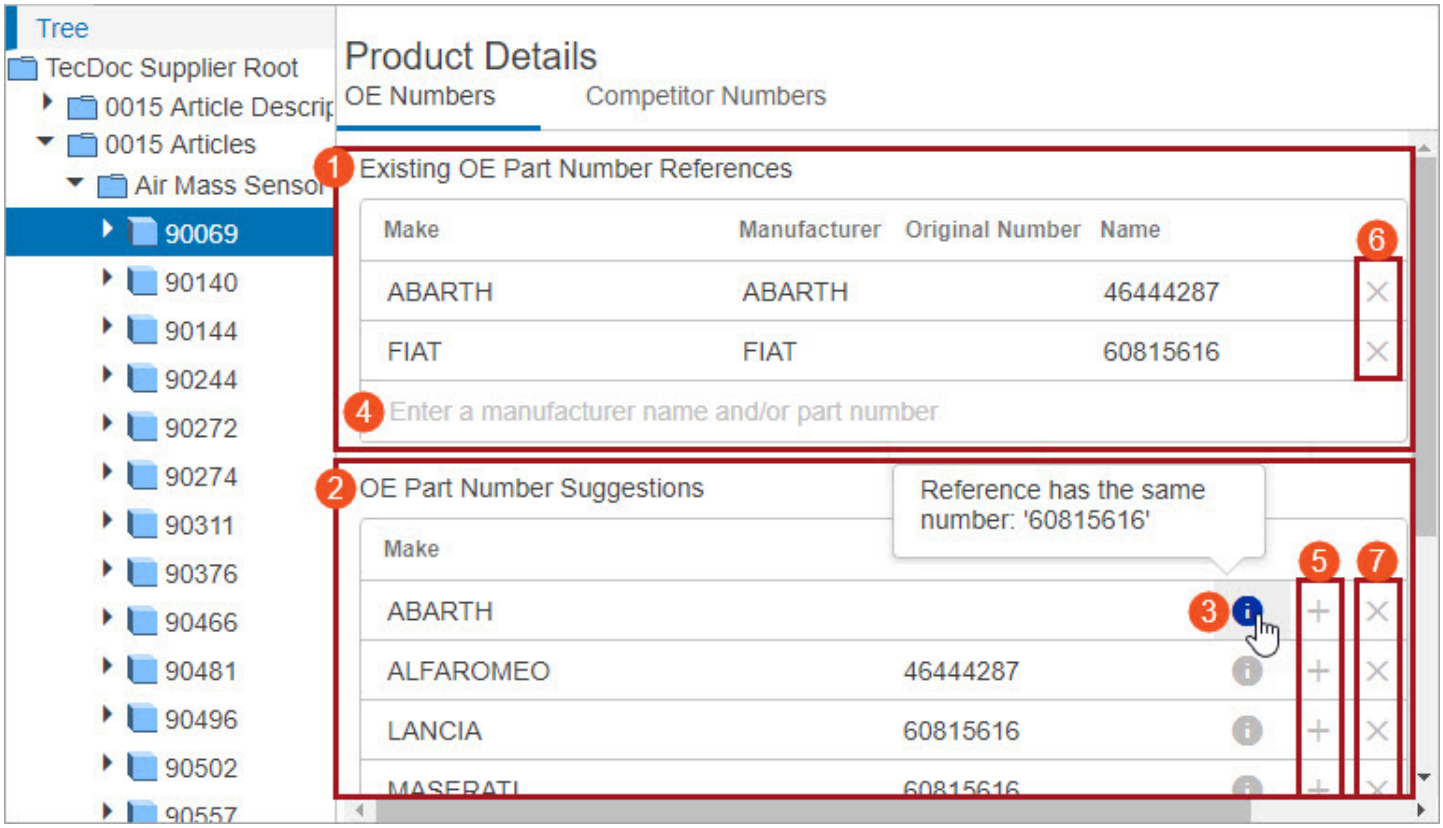
A competitor number refers to the part number used by a competitor. Whereas an OE number refers to the original parts used by the original equipment manufacturer to assemble a vehicle at the factory. This number is only unique within parts from that manufacturer. For example, an OE number should always 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

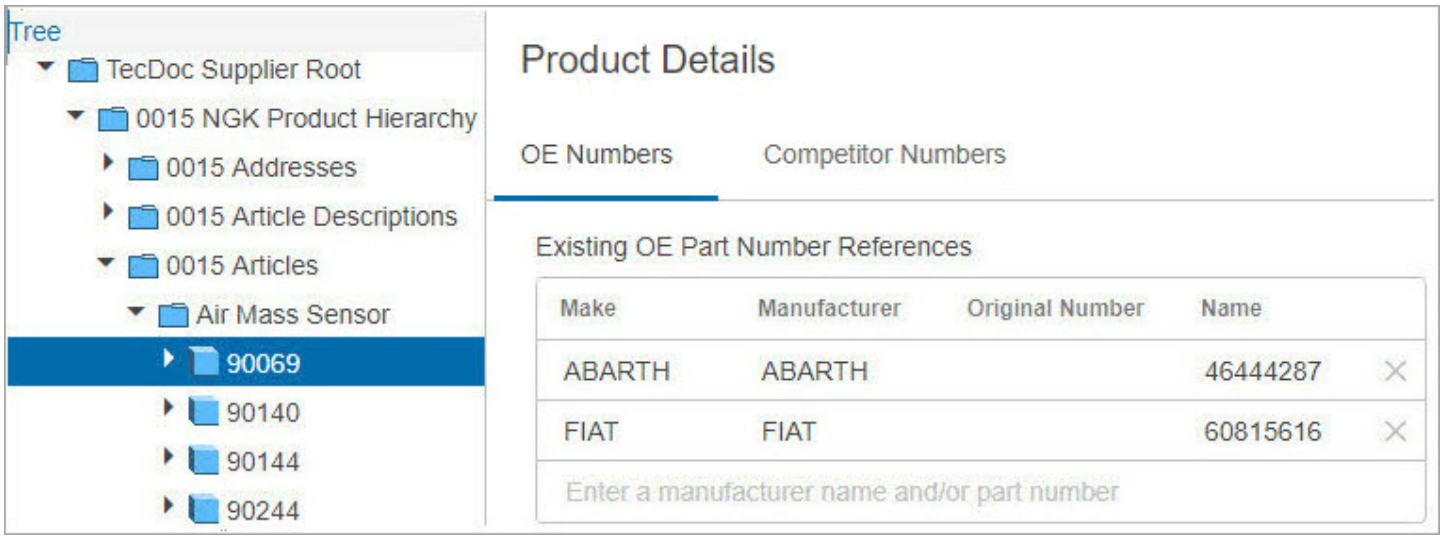
When implementing this solution, it is suggested to create a 'Product Details' screen within a Web UI and configure it to display when a product is selected within the Tree navigator. The Product Details screen should be created via a Node Details screen with a Tab Control that is configured to display 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, refer to the **Mappings** topic within the **Main Properties Overview** section of **Web User Interfaces** 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.

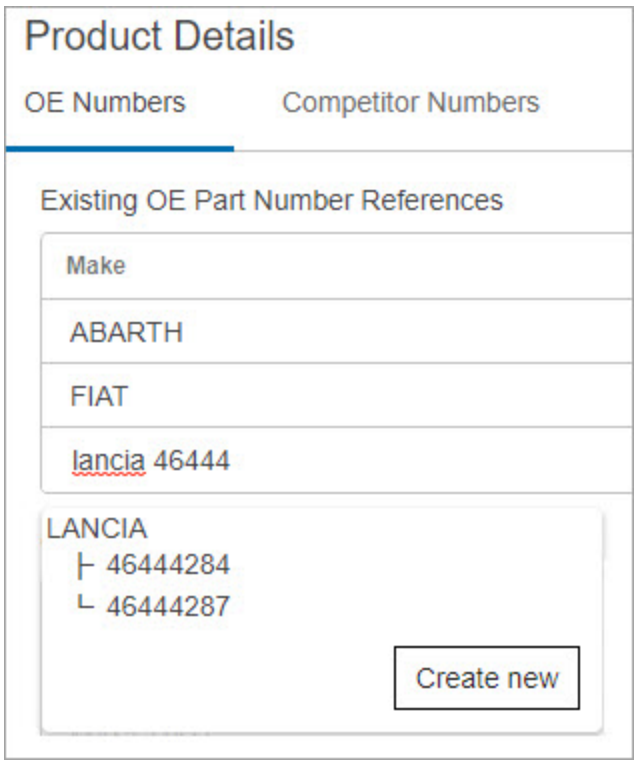
In the example below, part 90069 is selected within the Web UI Tree, and an 'OE Numbers' tab is configured to display results for 'Existing OE Part Number References.' Within the results, two Make / Manufacturer part numbers are displayed, along with the blank typeahead field that prompts users to 'Enter a manufacturer name and/or part number.'



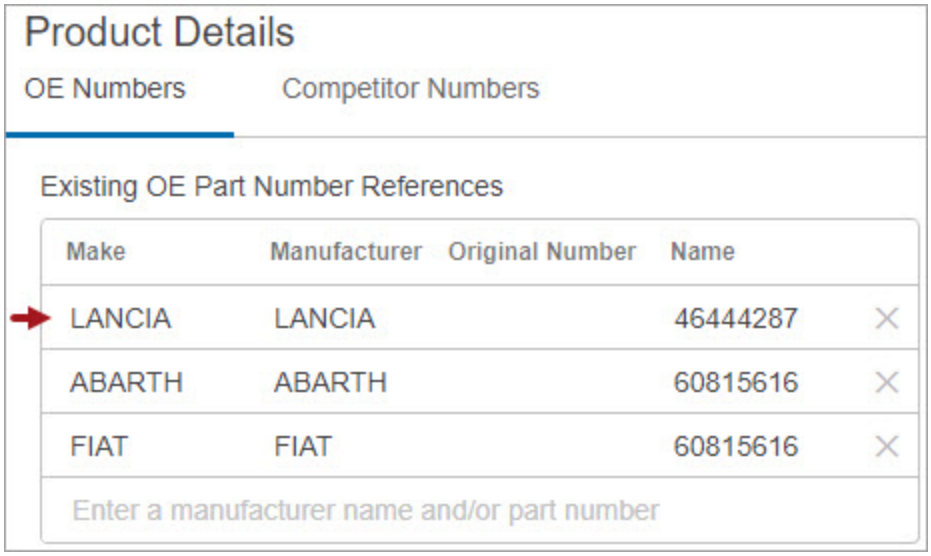

**Note:** Though a new part number can be created using the Create New Part dialog, a new manufacturer name cannot be created.

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.




Clicking on 46444287 adds the part number to the top of the references list, as shown below.

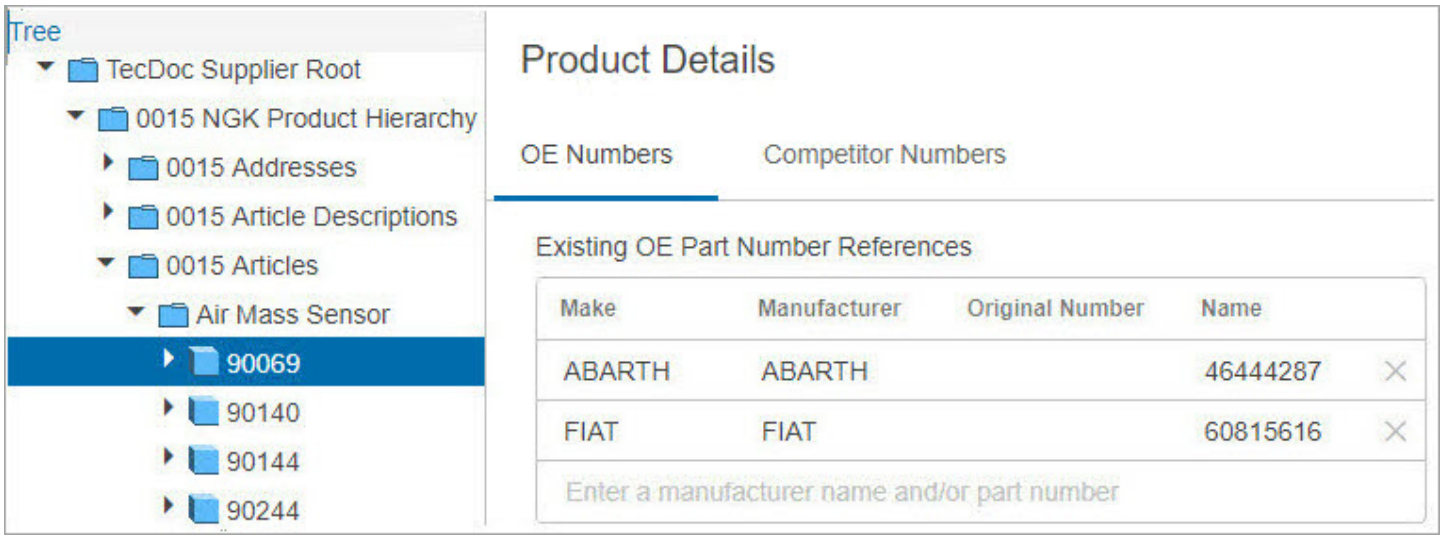



If the Part Number Suggestions component is configured below the references list, then the suggestions list will update as new part numbers are added to the references list.

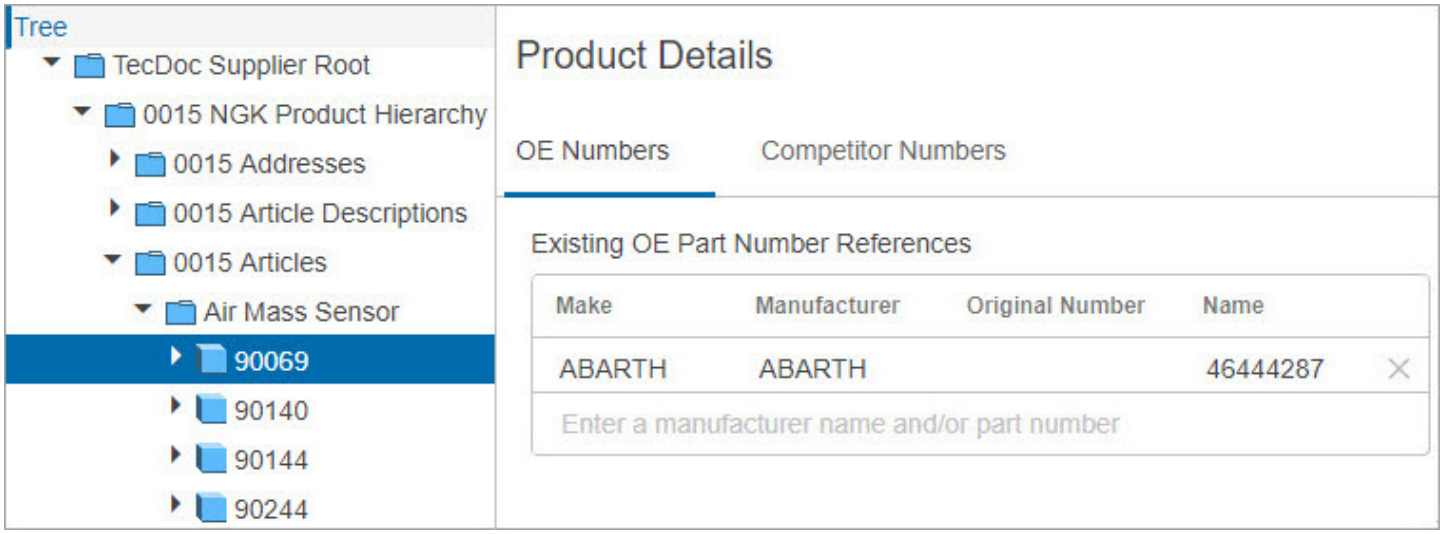
# Removing Existing Part Number References

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 (×) 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 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 (×) buttons that allow users to remove the part number reference from the selected part.




Clicking the remove (×) 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).

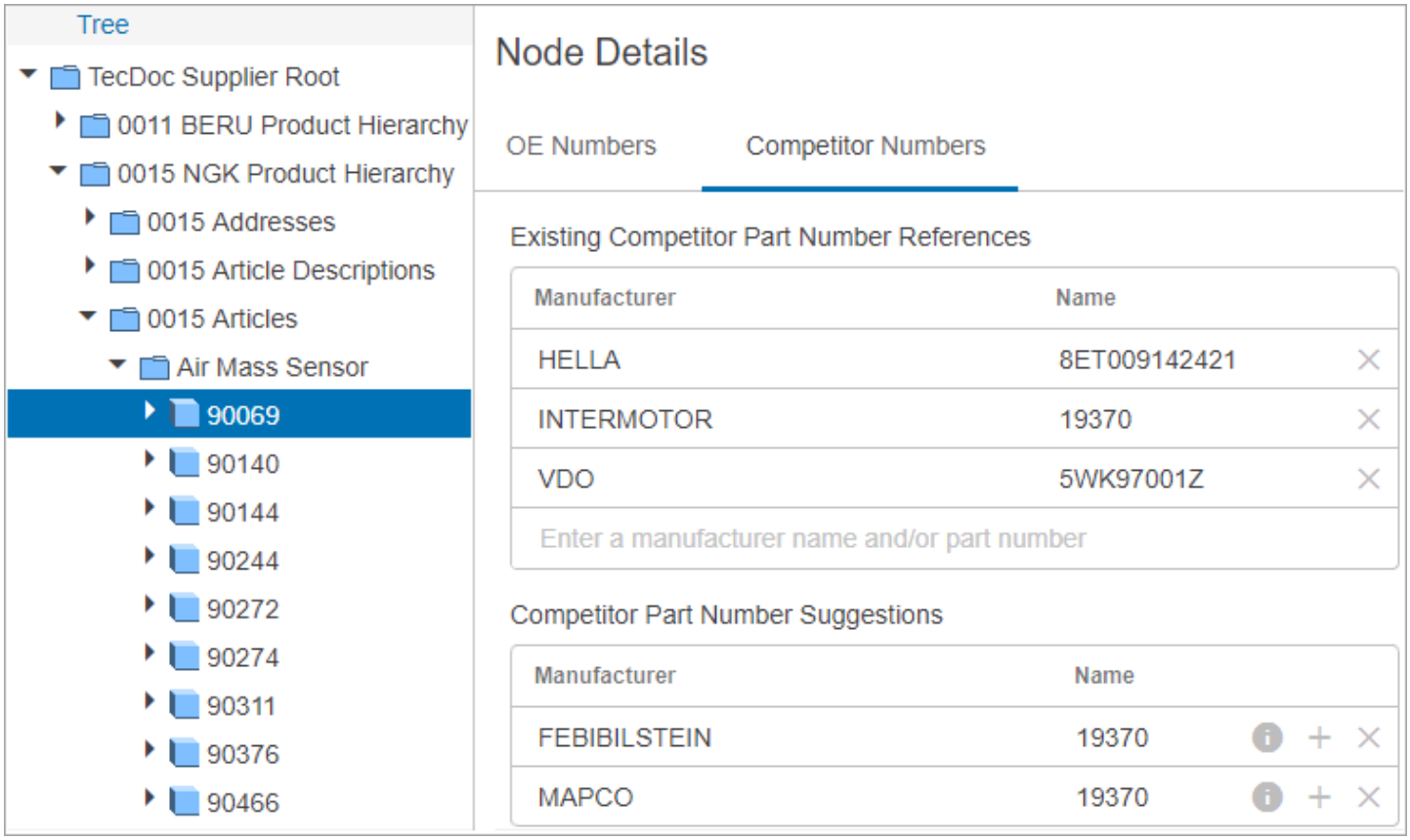



If an OE Number / Competitor part number is removed from the Part Number References list, and it previously displayed within the Part Number Suggestions list, then the unlinked OE Number / Competitor part number will display within the suggestions list. However, if the removed part number was not previously displayed within the suggestions list (i.e., manually added), then it will not display within the suggestions list.

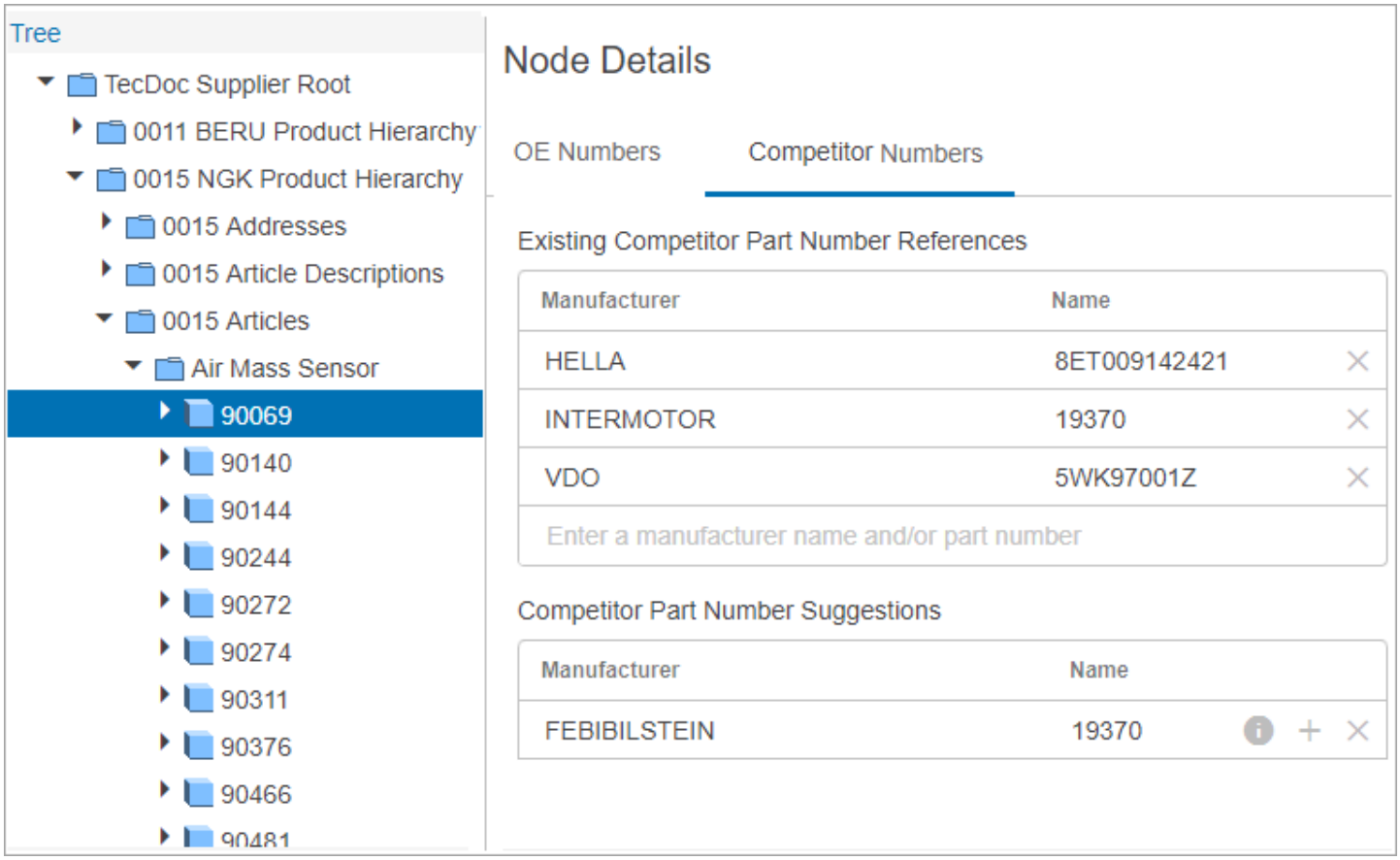
# Removing Part Number Reference Suggestions

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 (×) 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 (×) button that allows users to remove a part number reference of a suggested part number from the selected part.




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.



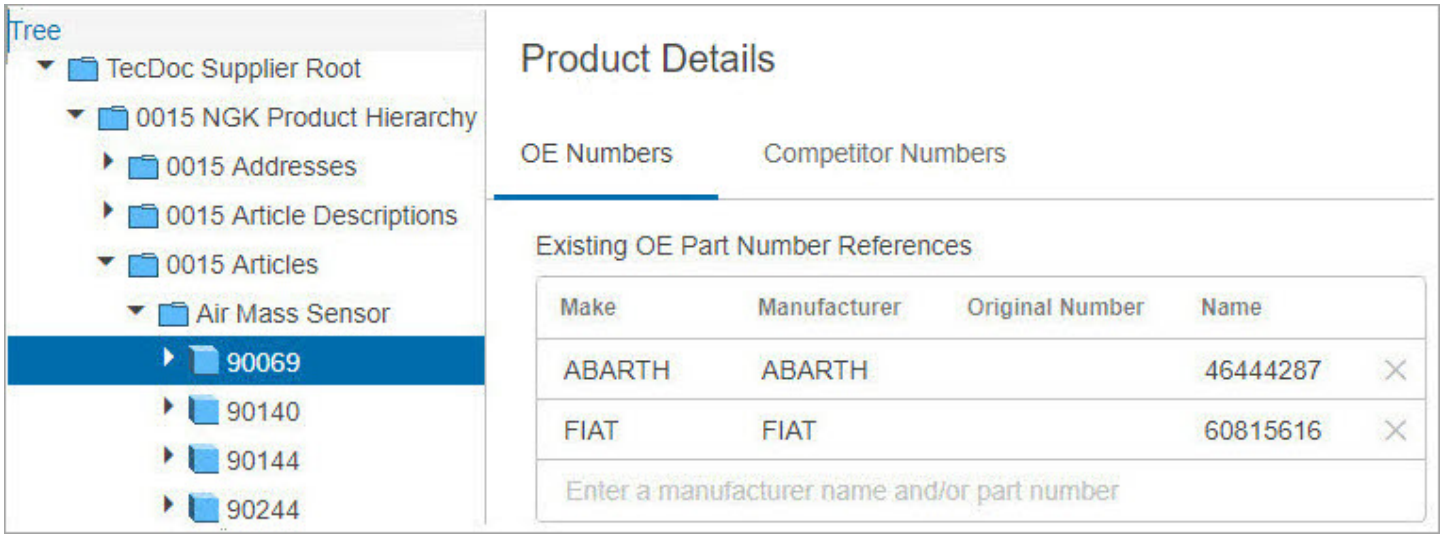

If the Part Number is removed from the Part Number Suggestions list, then the unlinked OE Number / Competitor part number will no longer display within the Part Number Suggestions list.

# 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, refer to the **Adding Part Number References Manually** topic.

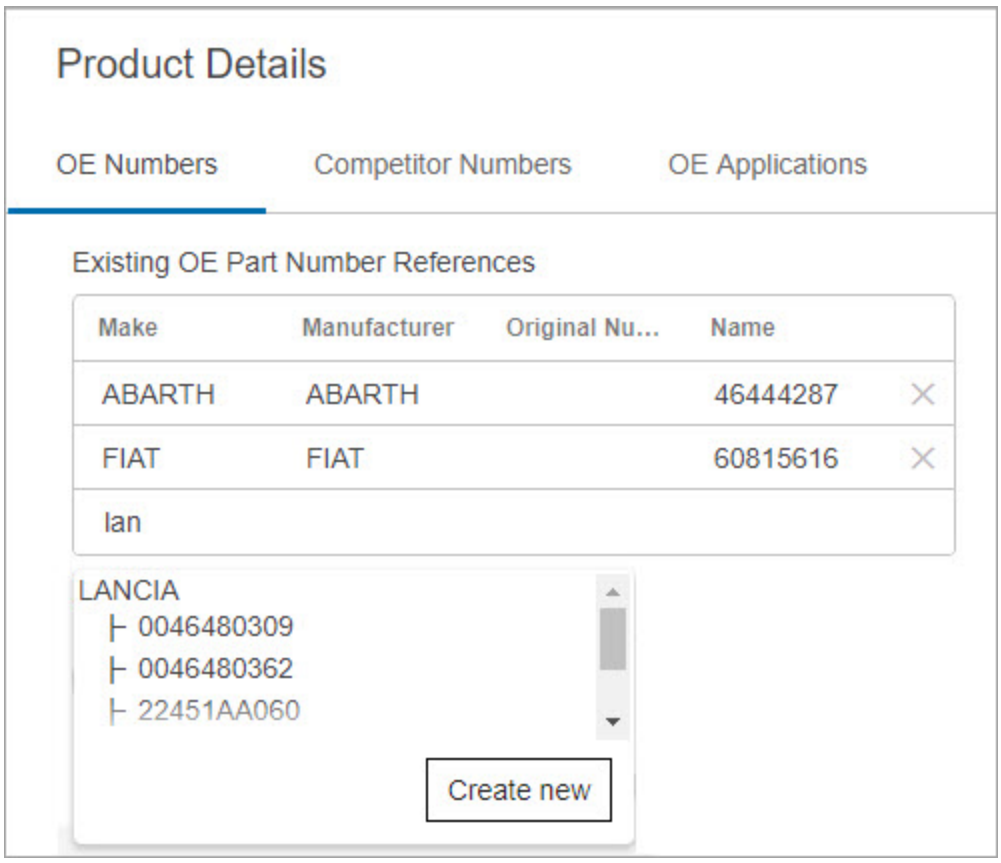
In the example below, part '90069' is selected within the Web UI Tree, and an 'OE Numbers' tab is configured to display results for 'Existing OE Part Number References.' Within the results, two part numbers display with different Makes and Manufacturers, along with the blank typeahead field that prompts users to 'Enter a manufacturer name and/or part number.'




**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.



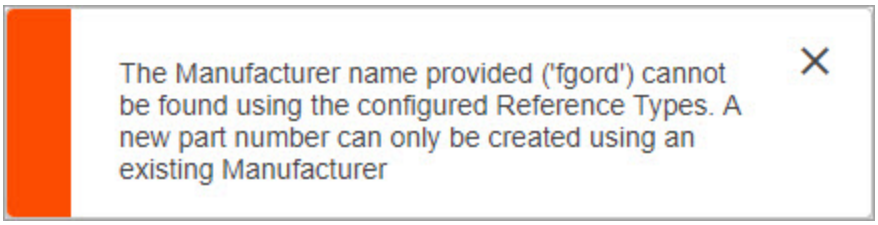

Clicking on a manufacturer name in the dropdown list, or using the arrow keys to navigate to the desired manufacturer name and pressing the Enter key, or clicking the 'Create new' button will display the Create New Part dialog with the Manufacturer Name field pre-populated (as shown below).

Once the Create New Part dialog displays, the Part Number field must be populated before the OK button will enable.

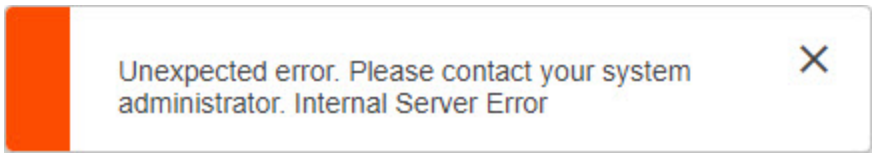
Optionally, the text within the pre-populated Manufacturer Name field can be edited. Typing text into this typeahead field will display a list of existing manufacturer names that begin with the provided text.

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.



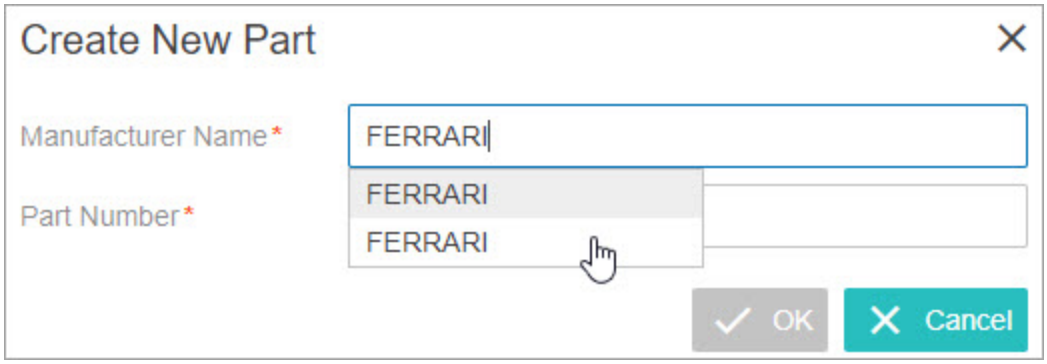

Additionally, if the attribute configured within the 'Original Number Attribute' parameter is not valid for the object type used for the new part number created, a warning dialog will display (as shown below). For more information, refer to the 'Original Number Attribute' parameter within the **Configuring 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.'

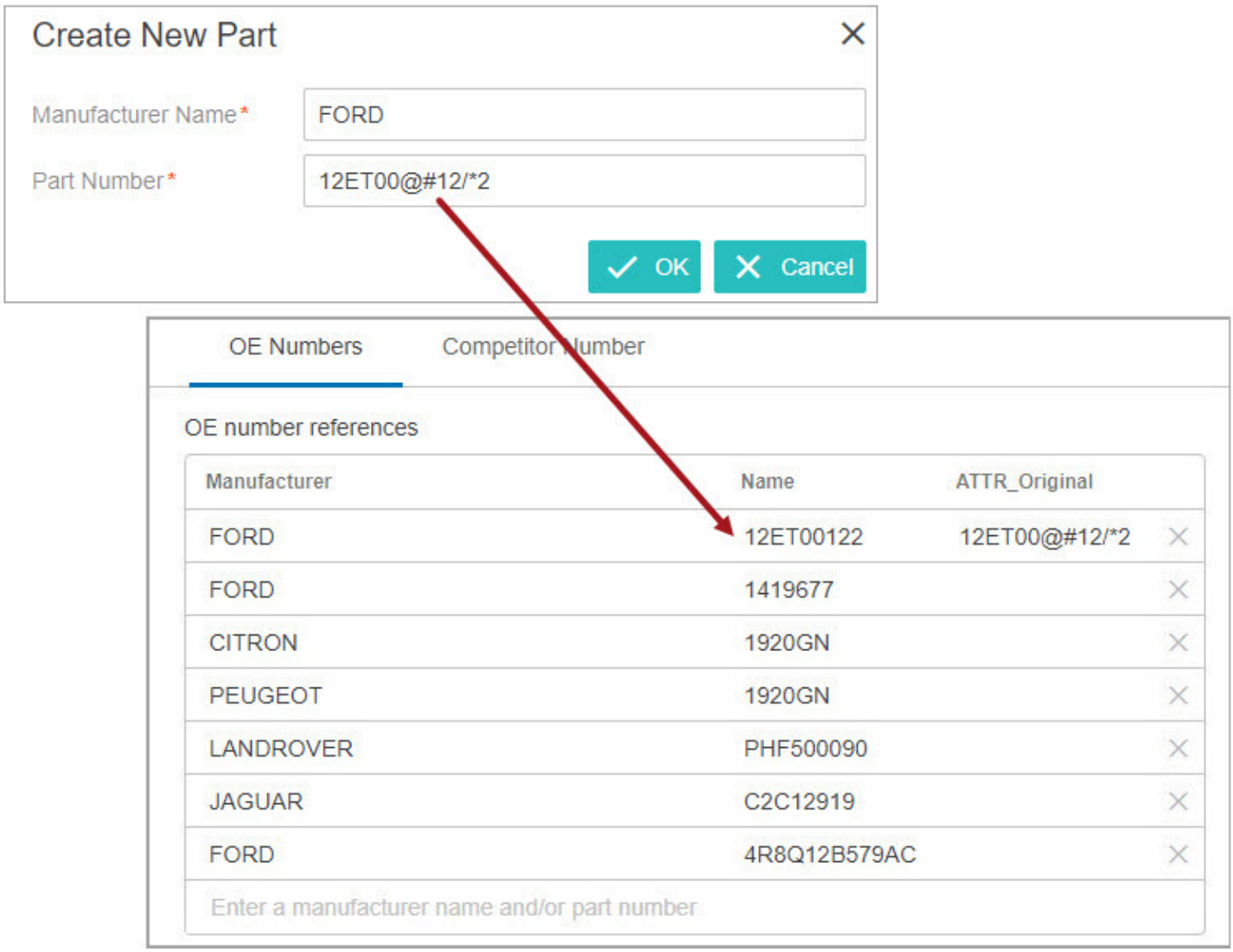


**Note:** To ensure users are able to easily distinguish the correct manufacturer name, be sure the manufacturer names within STEP are unique.

## 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, refer to 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 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, refer to the **Competitor and OE Numbers Solution** topic.

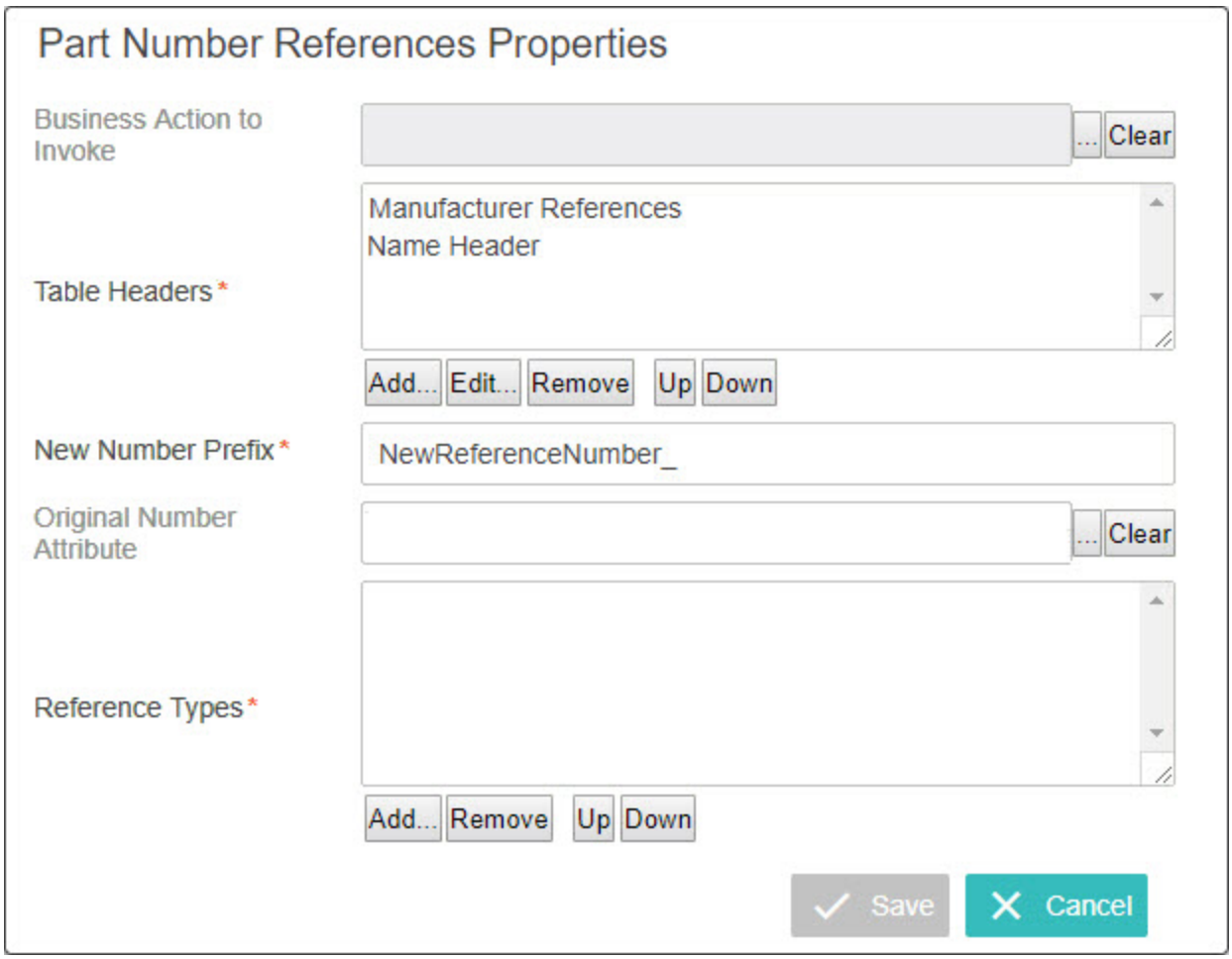
## Prerequisites

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, refer to 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.

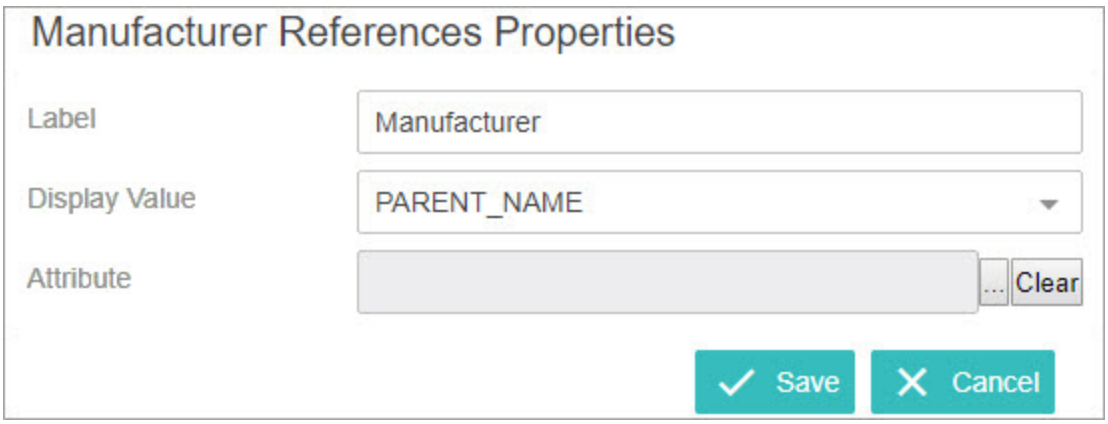
1. Using the Designer, go to Node Editor Properties Child Components Rows parameter for the screen that needs to display competitor and/or OE number information.
2. Click the **Add** button for the Child Components Rows parameter, and the Add Component dialog will display.
3. Find and select the **Part Number References** component, click the **Add** button, and the Part Number References Properties dialog will display (as shown below).




**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, refer to 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:

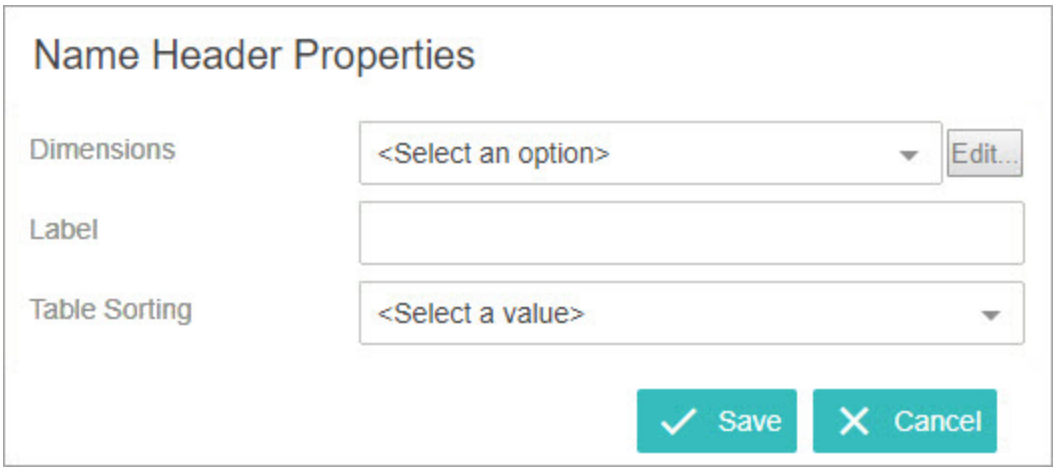
**Manufacturer References:** Table header component used to display the manufacturer and/or attribute value references of a selected part. Includes the following parameters:




- **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:

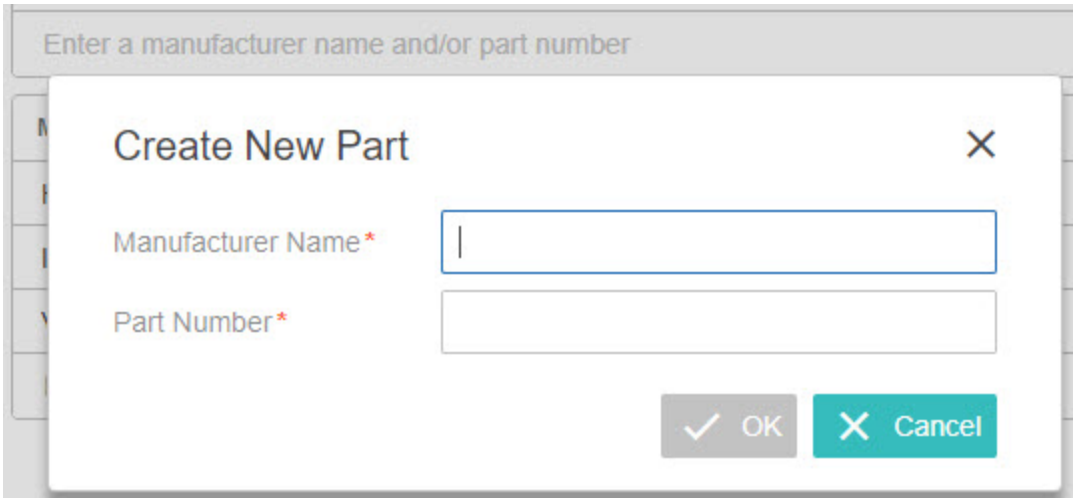



- **Dimensions:** By default, the parameter is blank. Optionally, select to use the 'Table Header Dimensions.'

- **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.




**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 in 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.

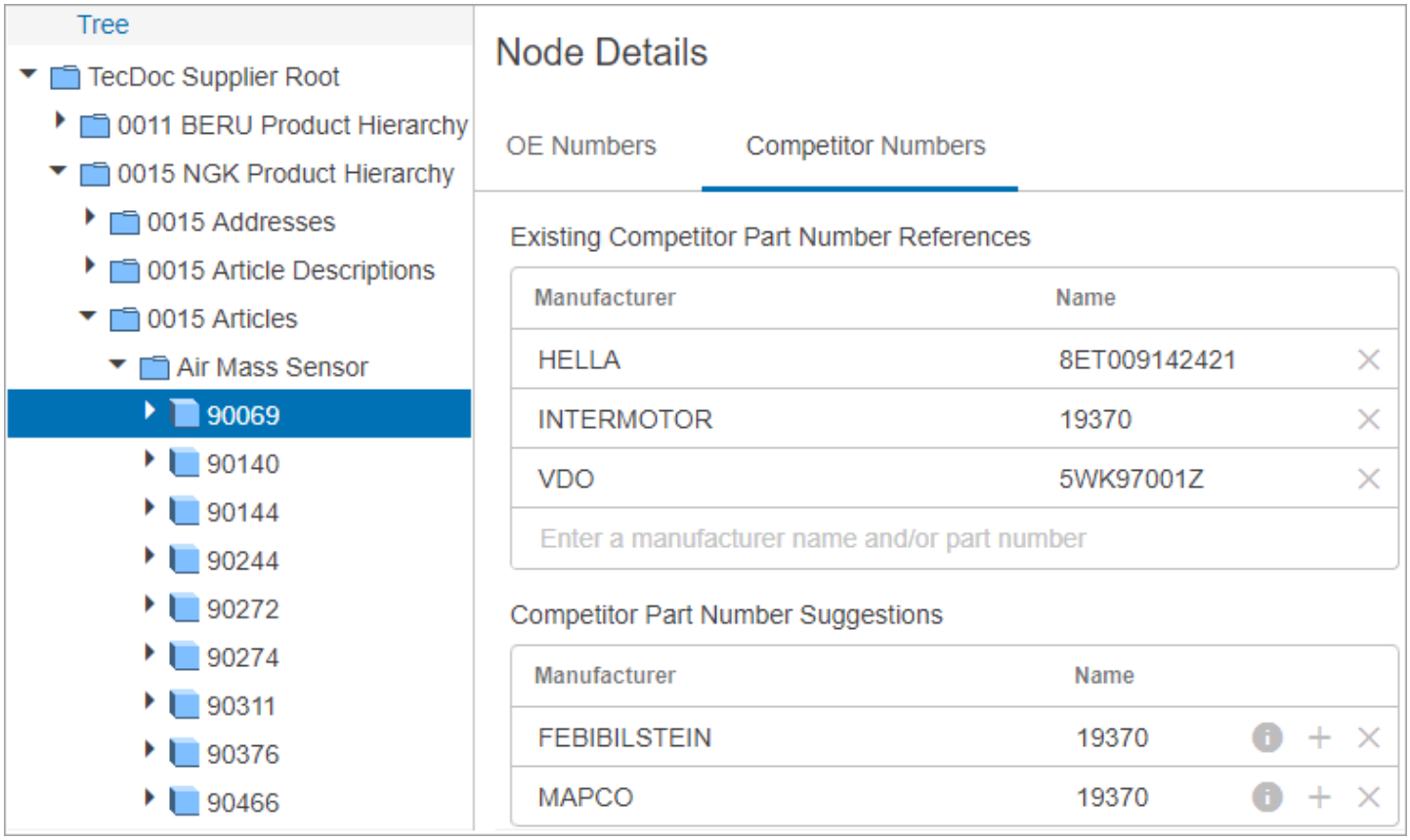
# Adding Part Number References from Suggestions

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.

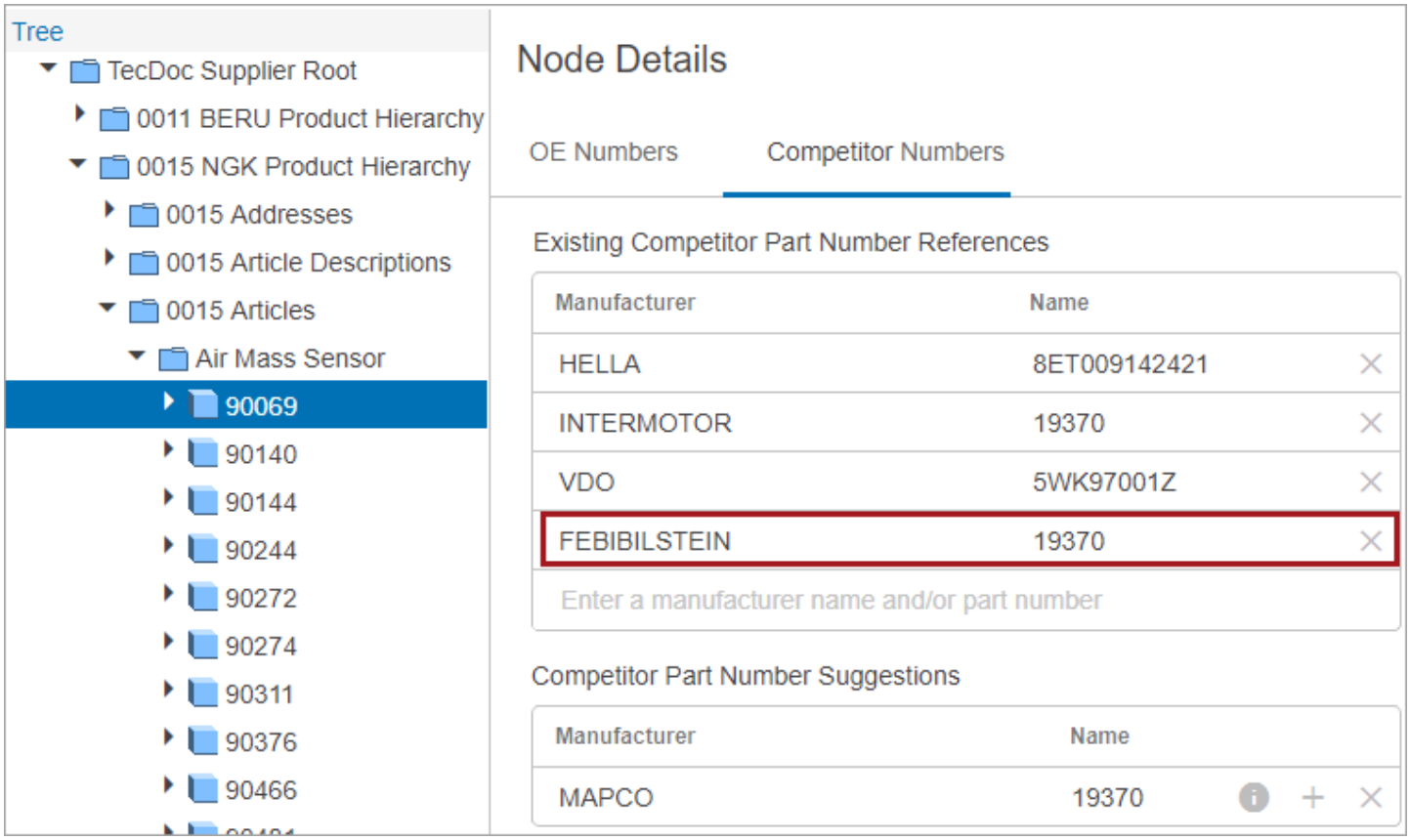
When any OE Number / Competitor Number is displayed in the Part Number Suggestions list, an add button (+) displays alongside each OE Number / Competitor Number. Clicking the add button will add an OE Number / Competitor Number to the Part Number References list.

Clicking the remove button (×) will remove the OE Number / Competitor Number reference from the Part Number References list / Part Number Suggestions list. For more information, refer to 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.



Clicking the (+) button to the right of the 'FEBIBILSTEIN 19730' suggestions list will add the 'FEBIBILSTEIN 19730' part to the Existing Competitor Part Number References list (as shown below).




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.

# 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, refer to 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.

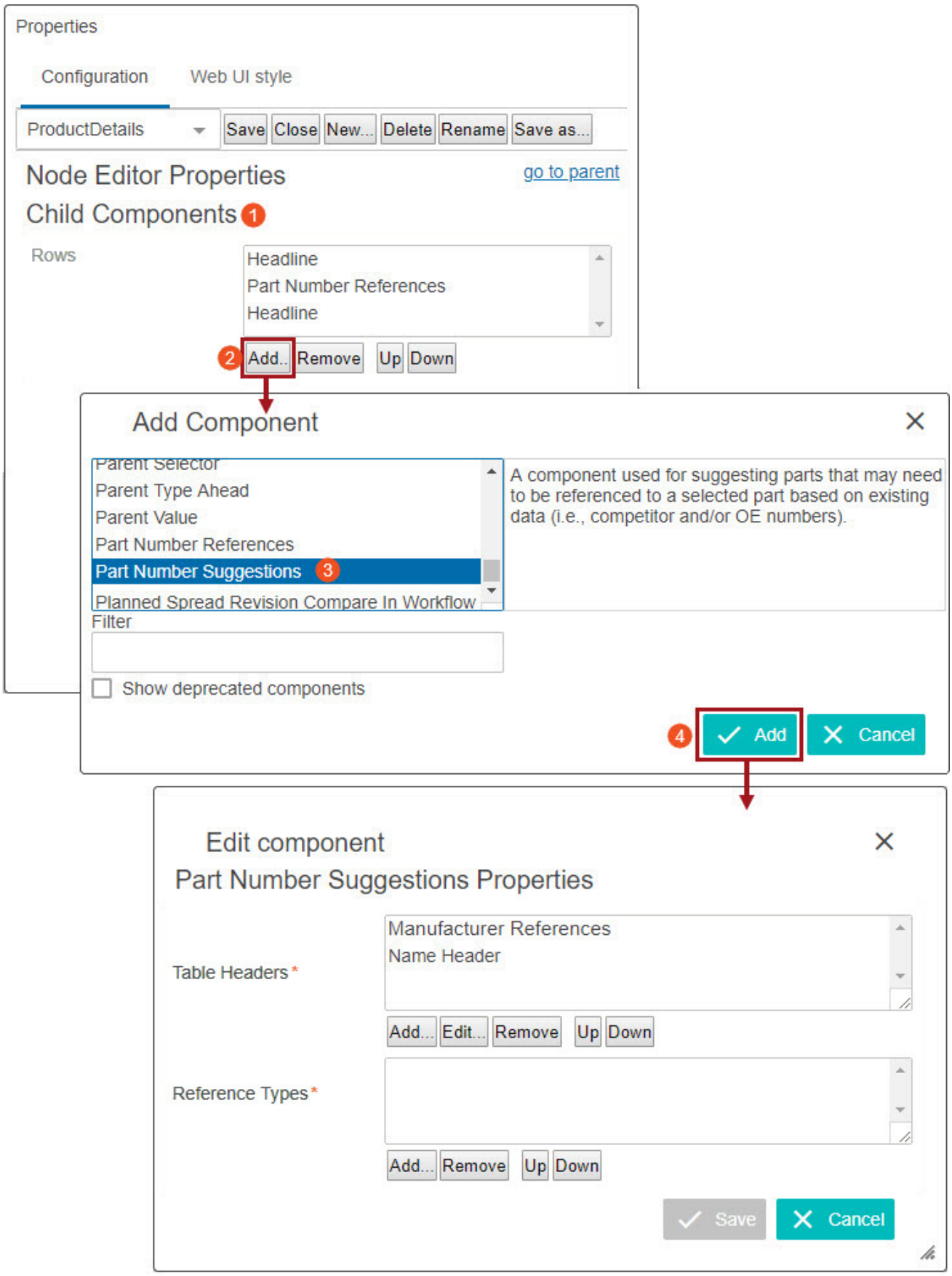

Properties
Configuration
Web UI style
ProductDetails
Save
Close
New...
Delete
Rename
Save as...
Node Editor Properties
go to parent
Child Components
1
Rows
Headline
Part Number References
Headline
2
Add..
Remove
Up
Down
Add Component
Parent Selector
Parent Type Ahead
Parent Value
Part Number References
Part Number Suggestions
3
Planned Spread Revision Compare In Workflow
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).
Filter
Show deprecated components
4
Add
Cancel
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

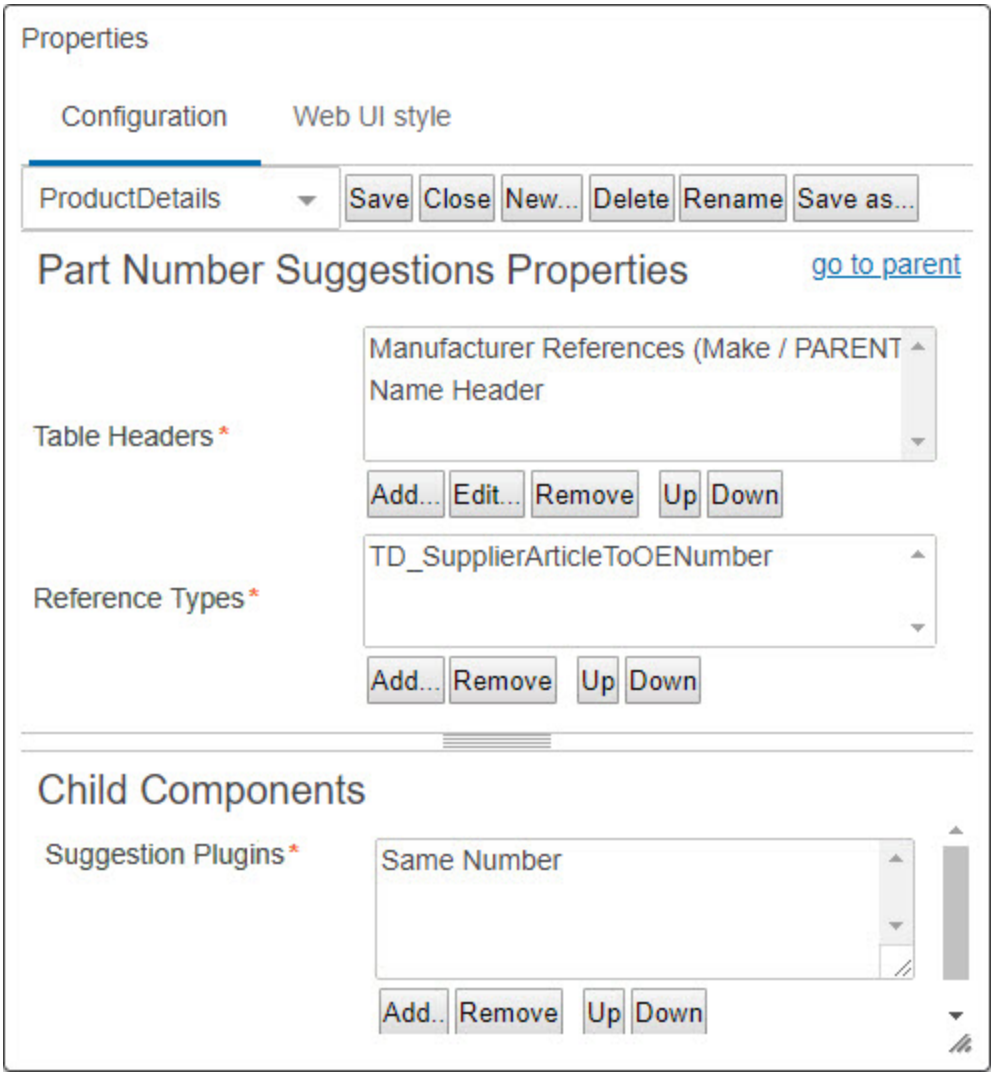
**Table Headers:** By default, this required parameter is pre-populated with the Manufacturer References and Name Header components. Optionally edit these components, and/or add additional Headers components to display information about the suggested competitor and/or OE number.

**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.

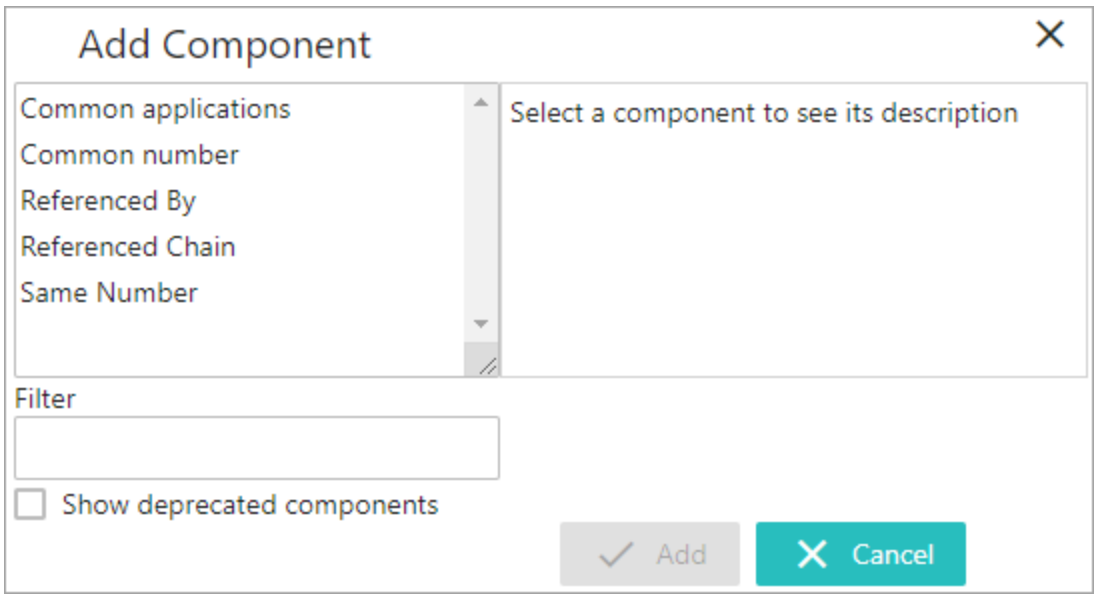
- For example, within the AutoCare solution, when a part is selected STEP would need to follow the **PIES Interchange** reference and display the targets and/or target related information.
- For example, within the NAPA solution, when a part is selected STEP would need to follow the **Product to Interchange 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.



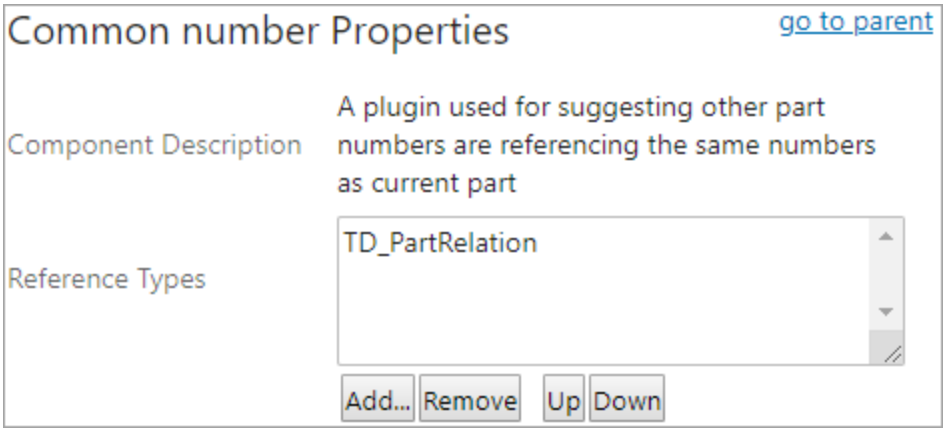

**Suggestion Plugins:** By default, this required parameter is pre-populated with the Same Number plugin. Optionally, a user can add one or more plugins by clicking the **Add** button. It is mandatory to add at least one plugin so that when a part is selected, information about the suggested competitor 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.

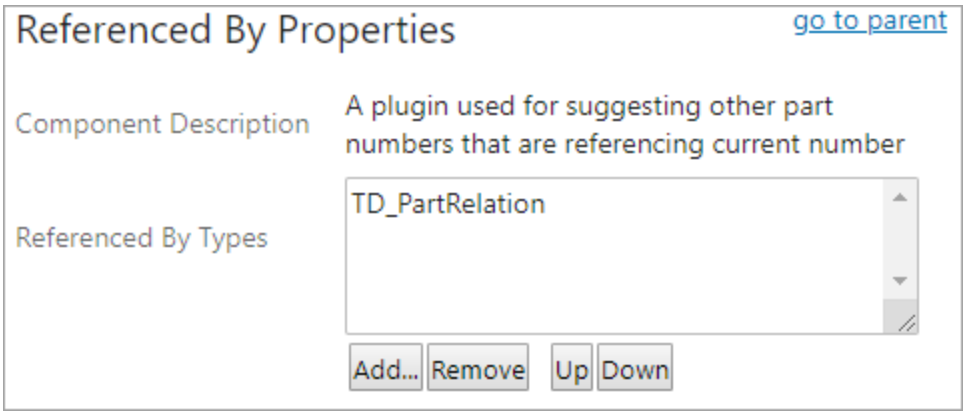
**Common number:** This plugin suggests the other part number(s) that are referencing the same part number (or Competitor Number or OE Number) as current part number does. In other words, If the current part refers to a part number (OE Number / Competitor Number) that is referenced by an other part number (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.



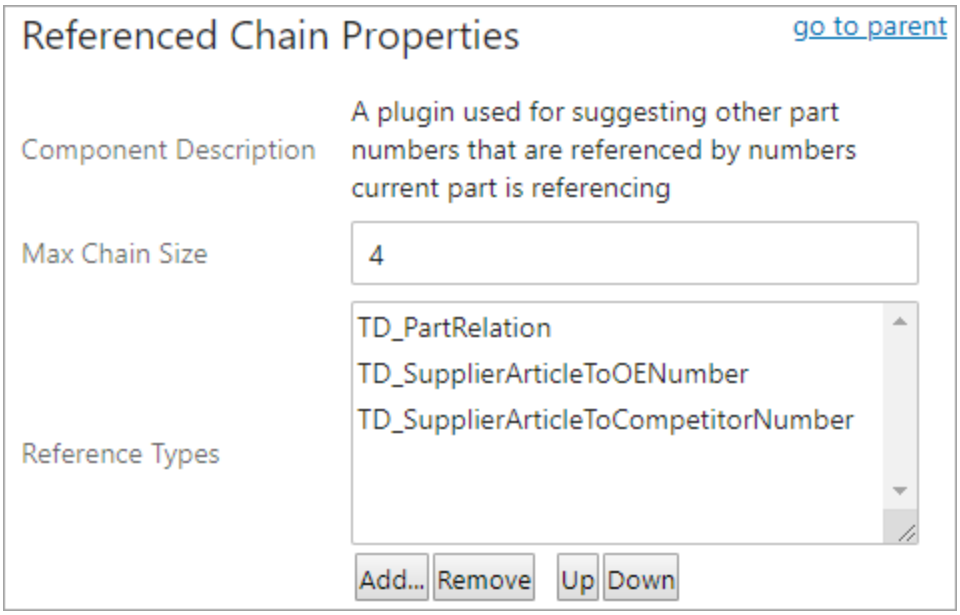

- **Reference Types:** By default, this required parameter is blank. However, it is mandatory to add at least one Product Reference Type so that when a part is selected, the reference types listed within this parameter are followed and display the target and/or related competitor and/or OE number information 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.




- 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.



- **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 Stibo Systems.

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, refer to the **Competitor and OE Numbers Solution** topic.

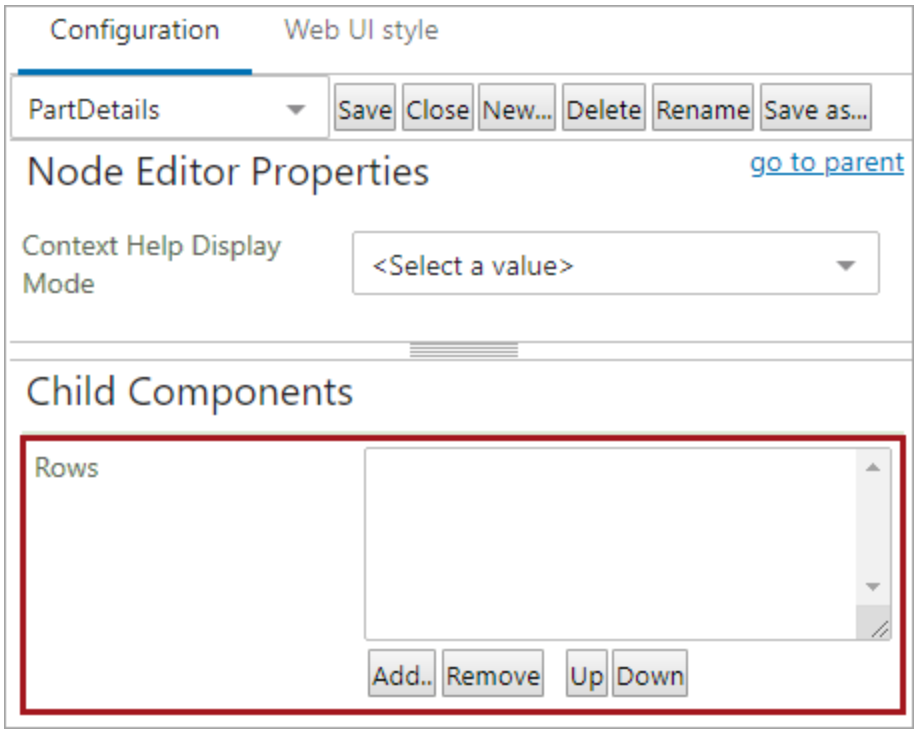
## Prerequisites

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, refer to the **Node Editor Component** and **Node List Component** topics within the **Using a Web UI** section of **STEP Online Help**.

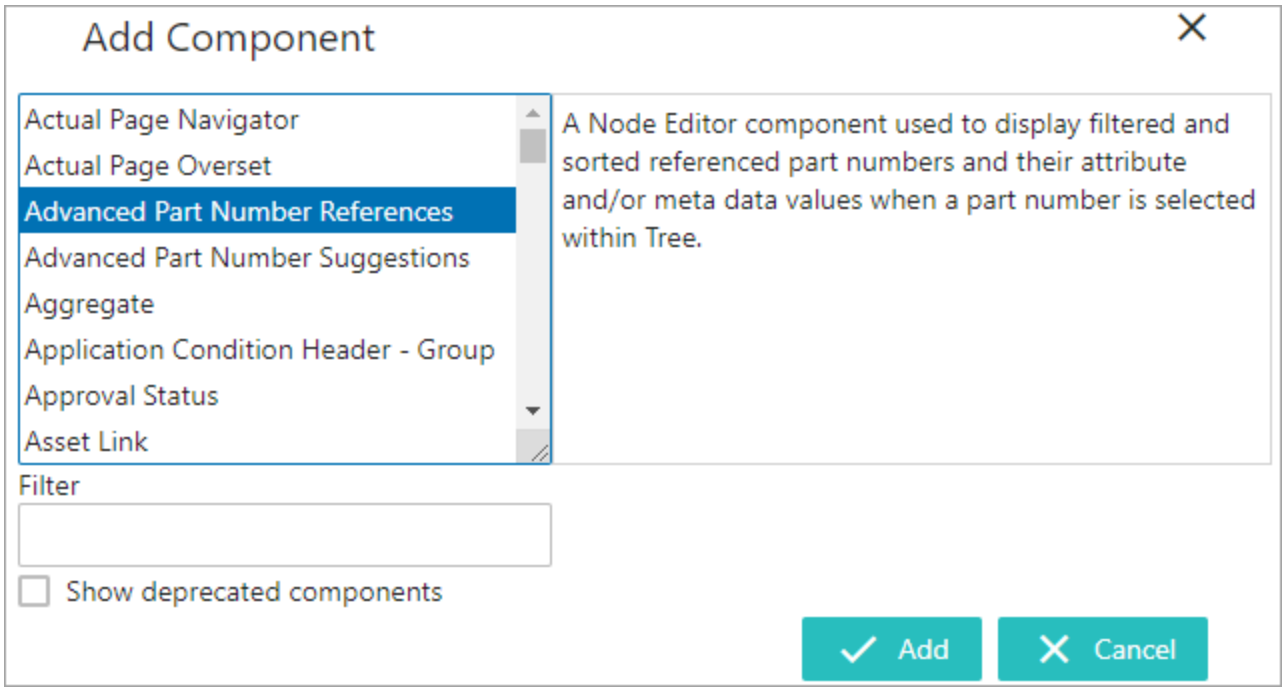
## Configuration Steps

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.

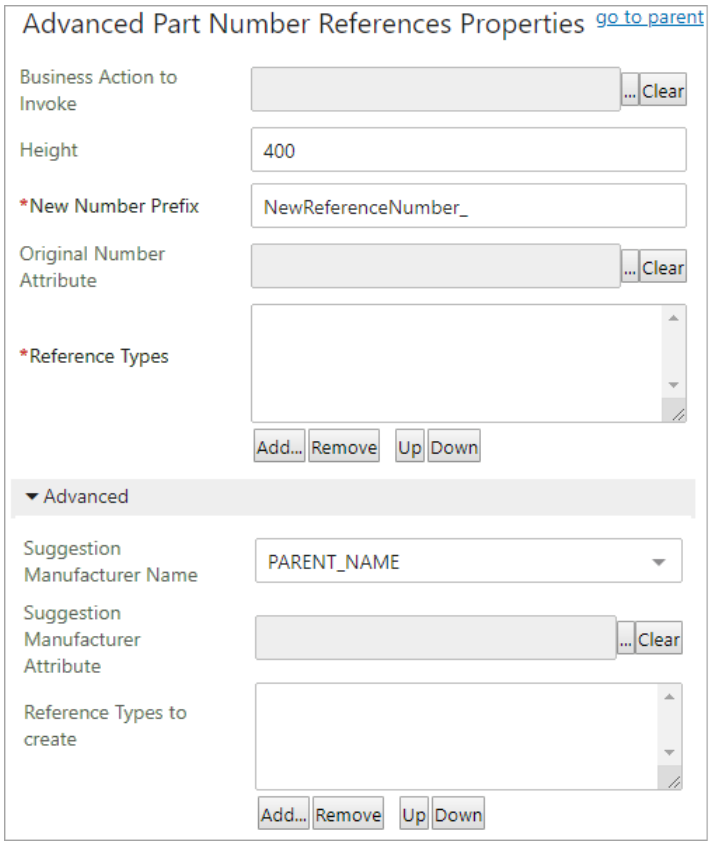
1. In the Web UI Designer, go to Node Editor Properties > Child Components > Rows parameter for the screen that needs to display competitor and/or OE number information.



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:

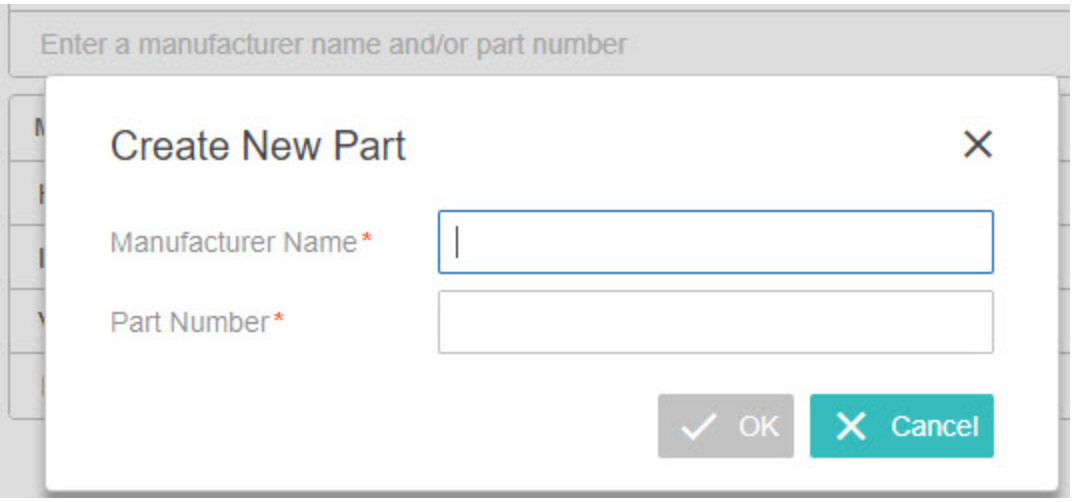


**Business Action to Invoke:** By default, this field is blank. Optionally, select a Business Action to be invoked after a new part is created using the Create New Part dialog. This can be helpful when a Business Action is configured to normalize / harmonize a part number. For more information about using this feature, refer to 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 in 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.

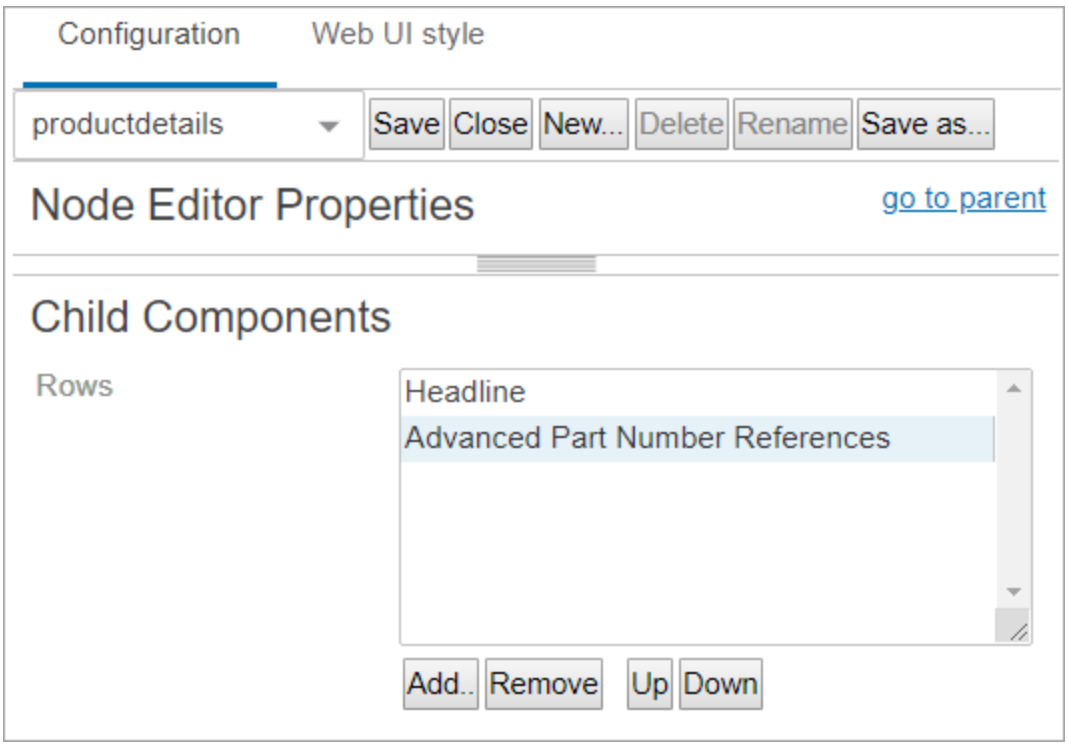
**Suggestion Manufacturer Name:** By default, the parameter is populated with a value 'PARENT_NAME'. This parameter defines the search criteria in the typeahead field whether if the competitor / OE number search is based on the manufacturer name or the configured attribute value (configured in the Suggestion 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, refer to 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.

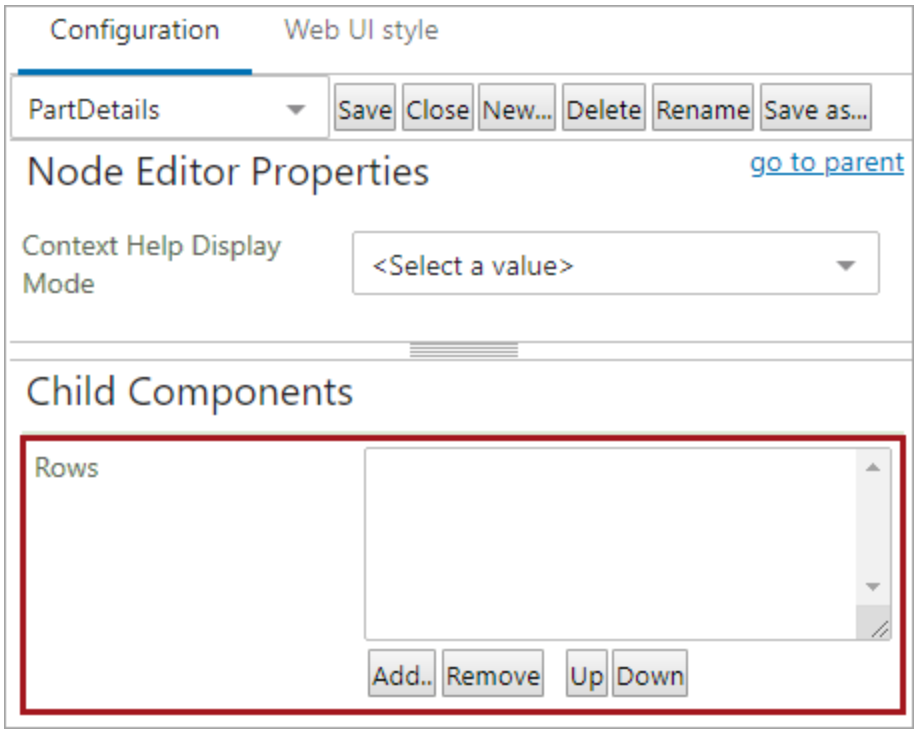
# Configuring the Advanced Part Number Suggestions Component

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, refer to the **Competitor and OE Numbers Solution** topic.

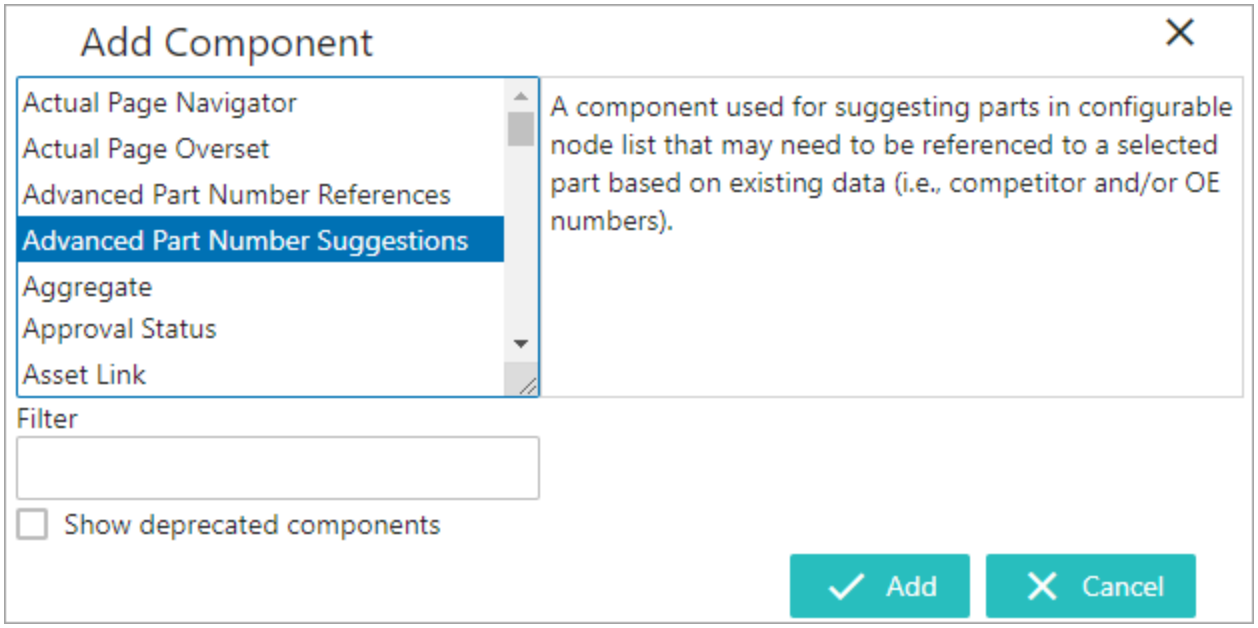
## Configuration Steps

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.

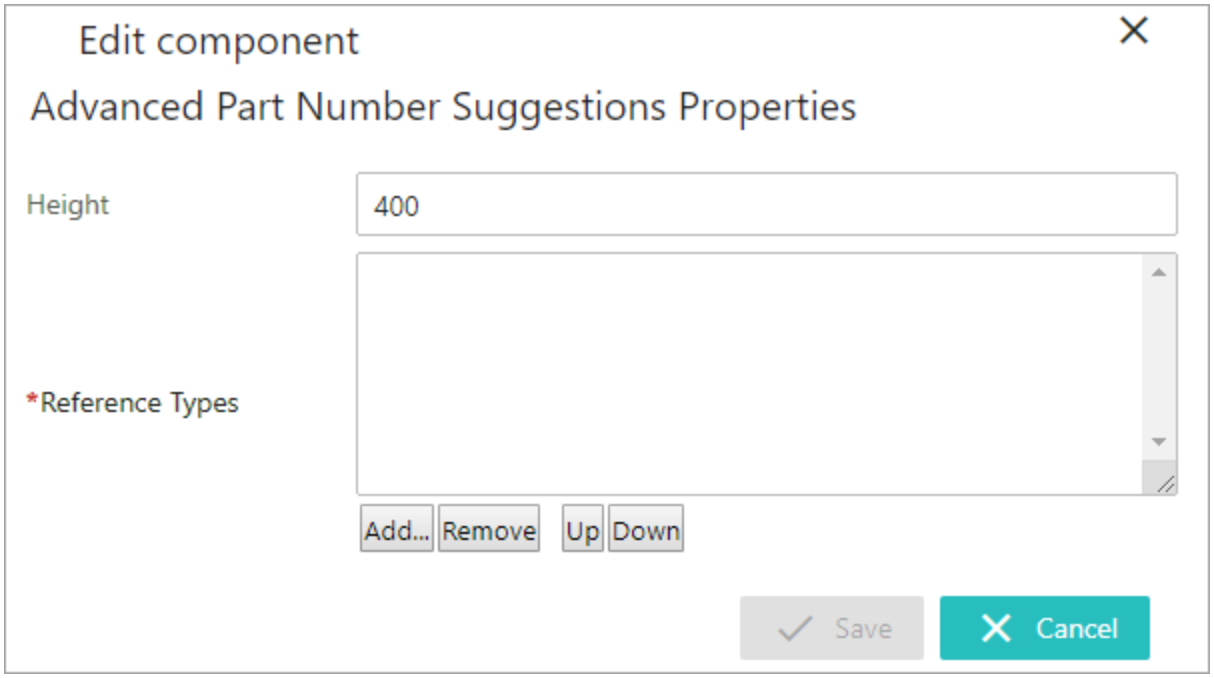
1. In the Web UI the Designer, go to Node Editor Properties > Child Components > Rows parameter for the screen that needs to display part number suggestions.




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:

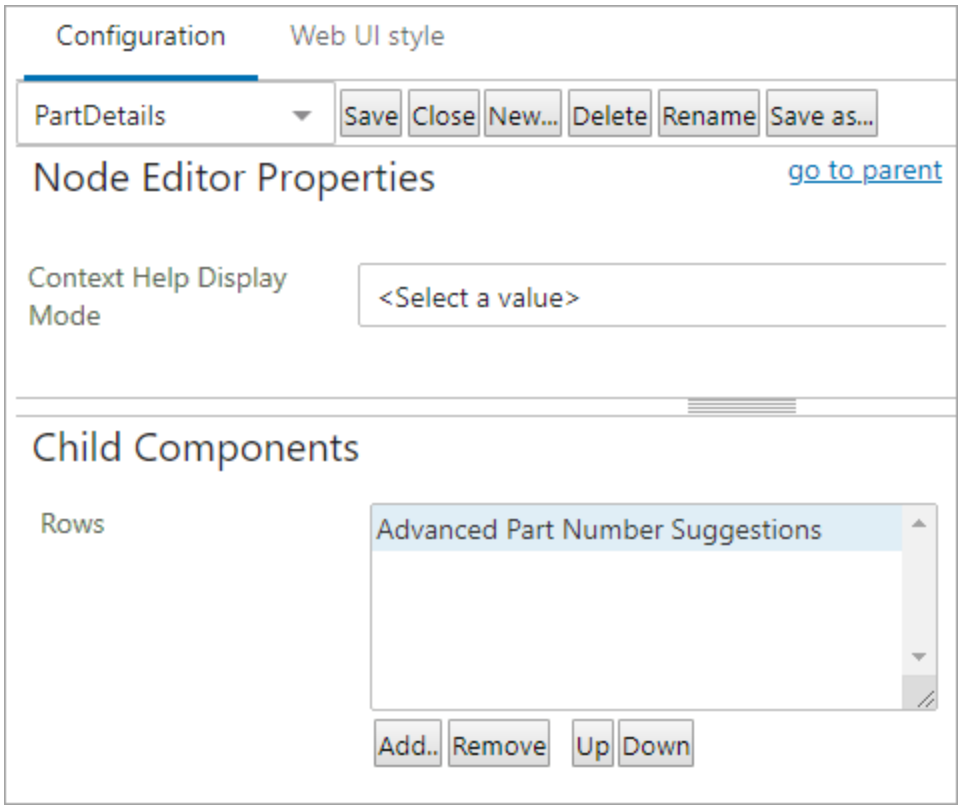



**Height:** This optional parameter specifies the visual height of the Advanced Part Number Suggestions component in pixels. By default, this parameter is pre-populated with a value '400.' Users can enter a numeric value to alter height of the component.

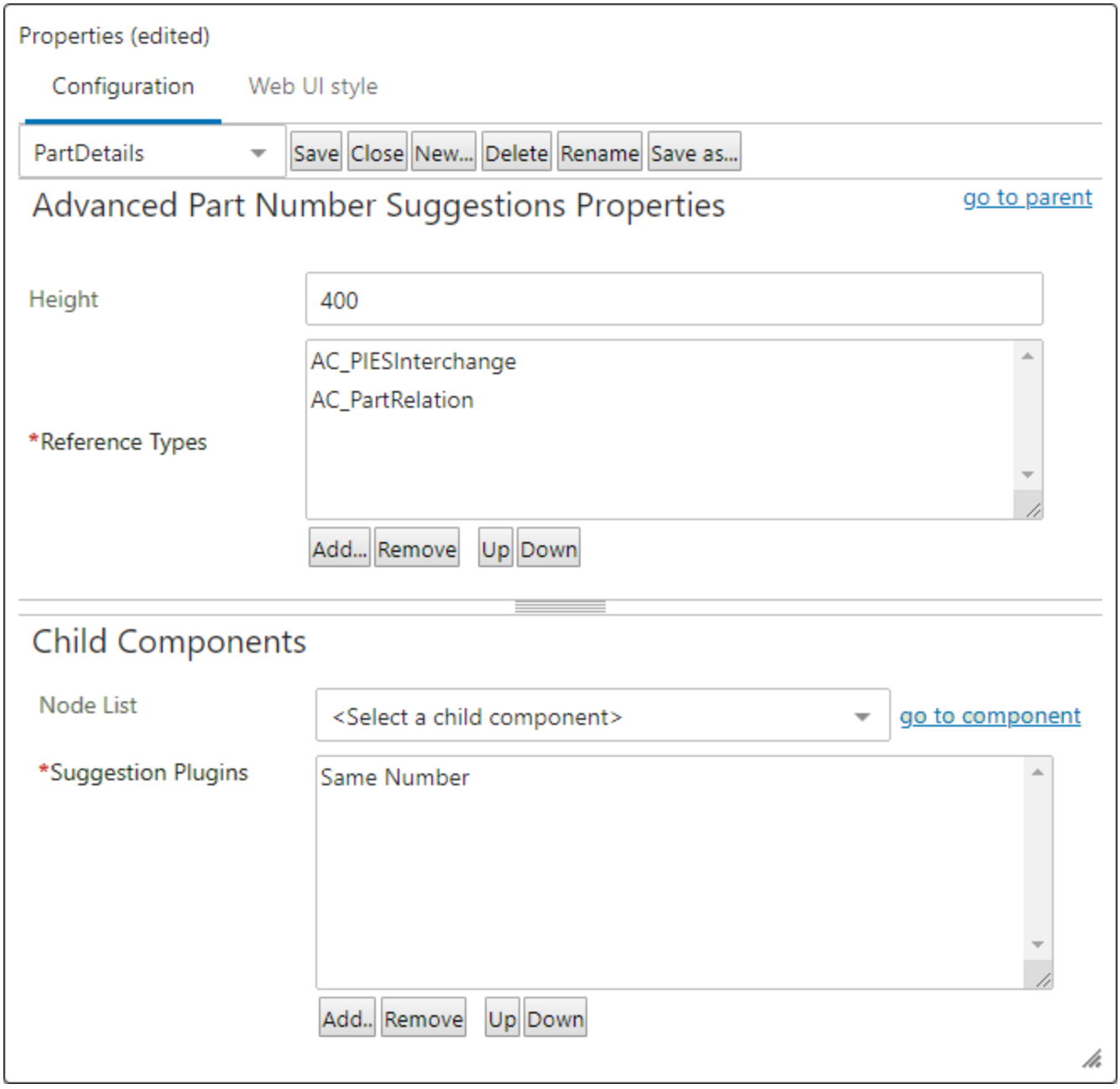
**Reference Types:** By default, this required parameter is blank. However, it is mandatory to add at least one Product Reference Type so that when a part is selected, STEP can follow the reference types listed within this parameter and directly display the target and/or targets related information based 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.

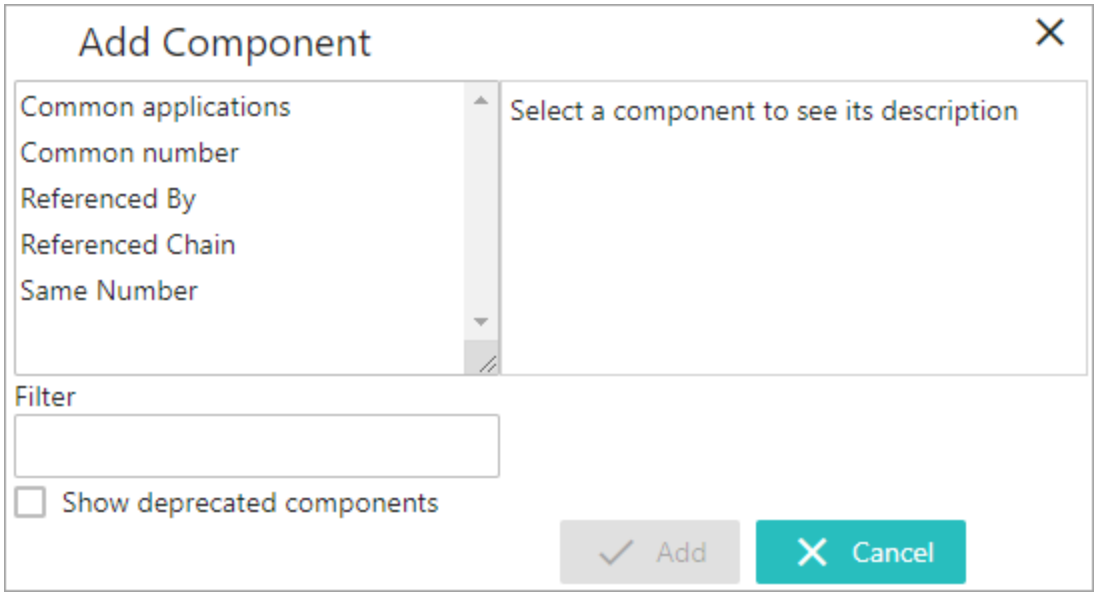



5. Double click the newly added Advanced Part Number Suggestions component, and the Advanced Part Number Suggestions Properties will display as shown below.




**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, refer to 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.

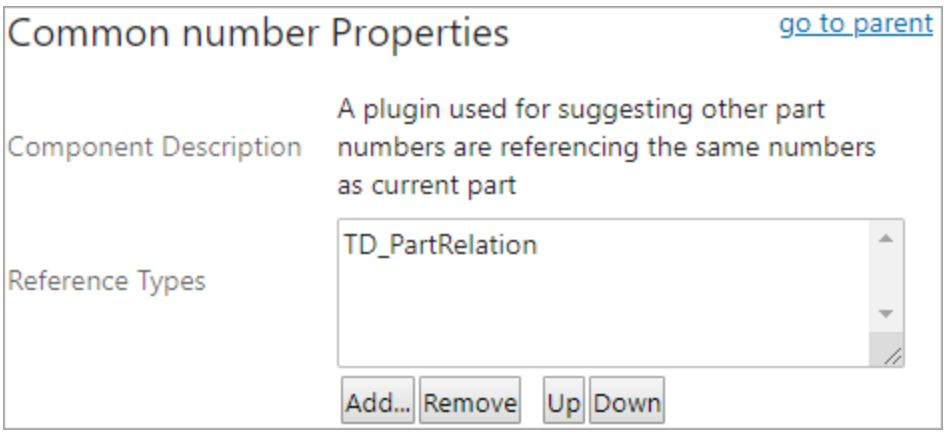



The Suggestion Plugins options are the following:

**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.

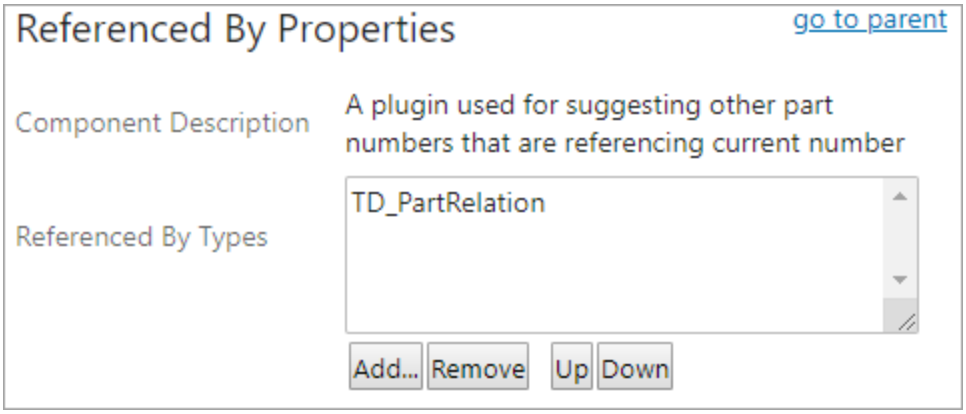
**Common number:** This plugin suggests other part number(s) that are referencing the same part number (or Competitor Number or OE Number) as the currently selected part number does. In other words, if the current part refers to a part number (OE Number / Competitor Number) that is referenced by another 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.



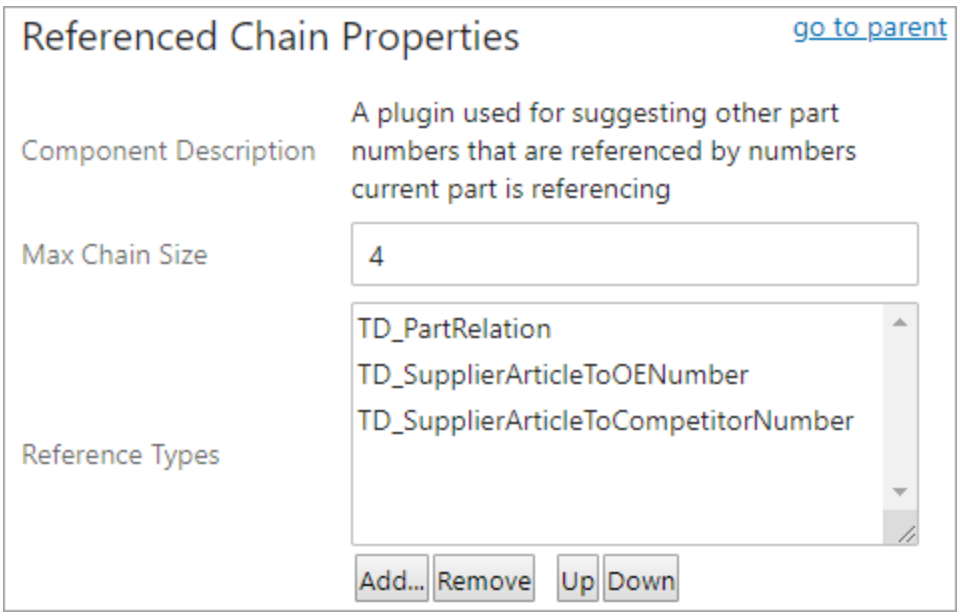

- **Reference Types:** By default, this required parameter is blank. However, it is mandatory to add at least one Product Reference Type so that when a part is selected, the reference types listed within this parameter are followed and display the target and/or related competitor and/or OE number information 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.




- 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.

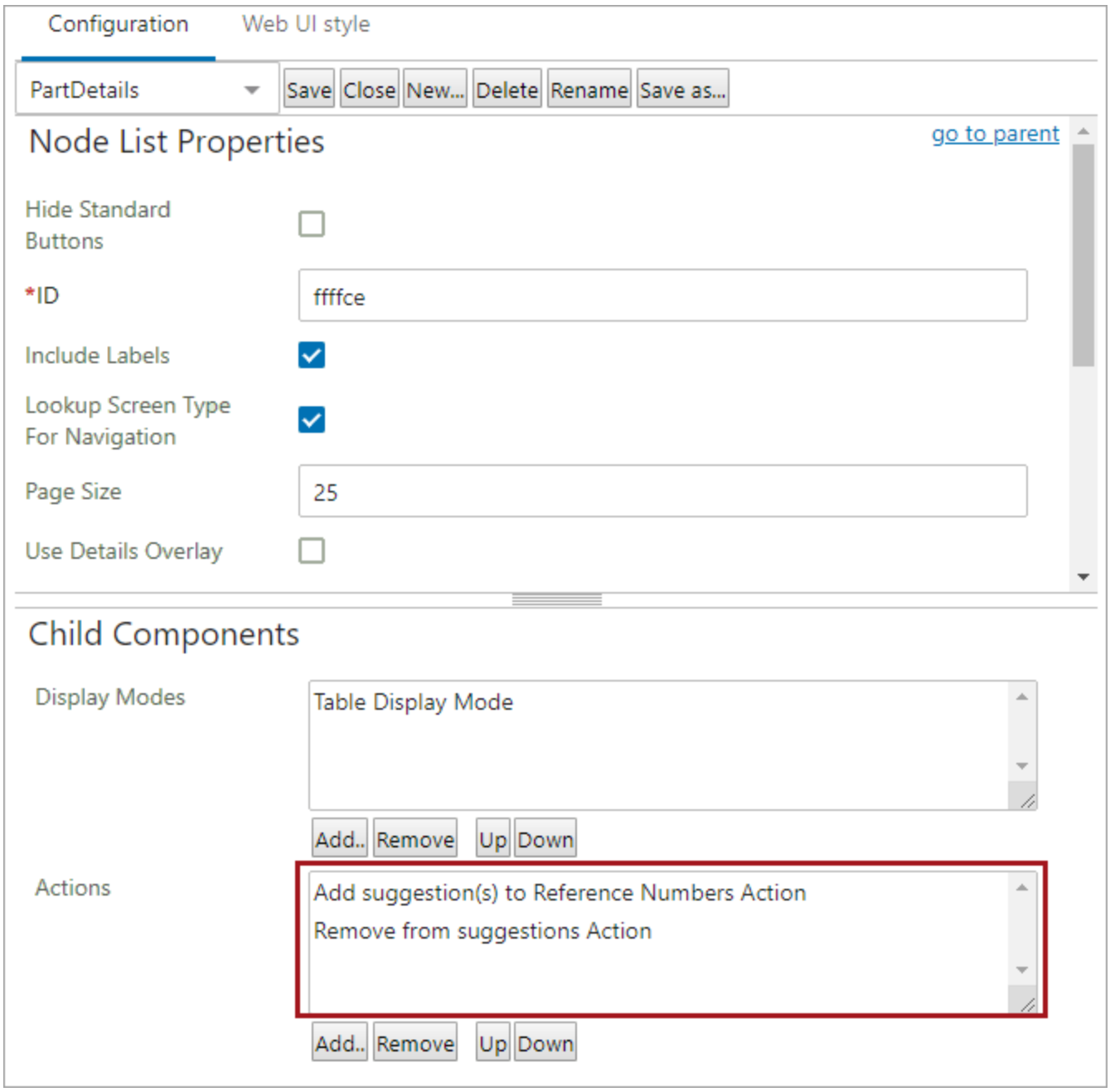



- **Max Chain Size:** By default, this required parameter is blank and allows a numeric value. This parameter defines the maximum allowed chain size to follow the reference types listed within the next parameter Reference Types. For example, If the value '4' is entered against the Max Chain Size parameter, 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 Stibo Systems.

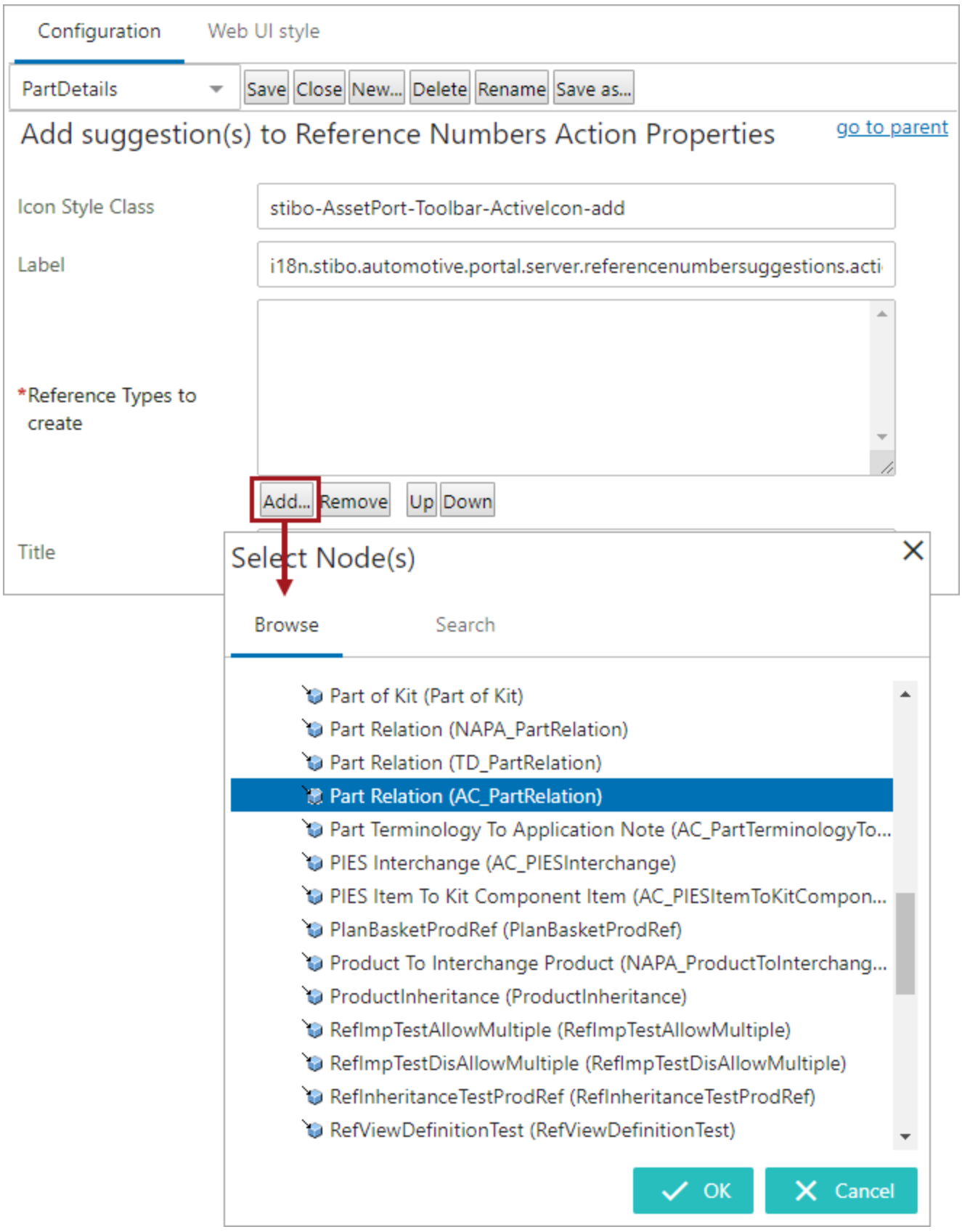
6. Go to Child Components > Node List > in the dropdown select 'Node List' > click on 'go to component.' The Node List Properties dialog with two pre-configured action buttons will display as shown below.




**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.

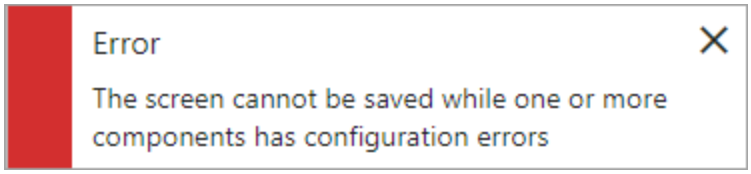
**Remove from suggestions Action:** This action button removes the selected Competitor and/or OE part number reference of a selected part from the Part Number Suggestions list. This is a ready to use button and user can double click on the 'Remove from suggestions Action' to further configure the Icon 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.referencenumbersuggestions.acti
*Reference Types to create
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

8. 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.




9. 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, refer to 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)).

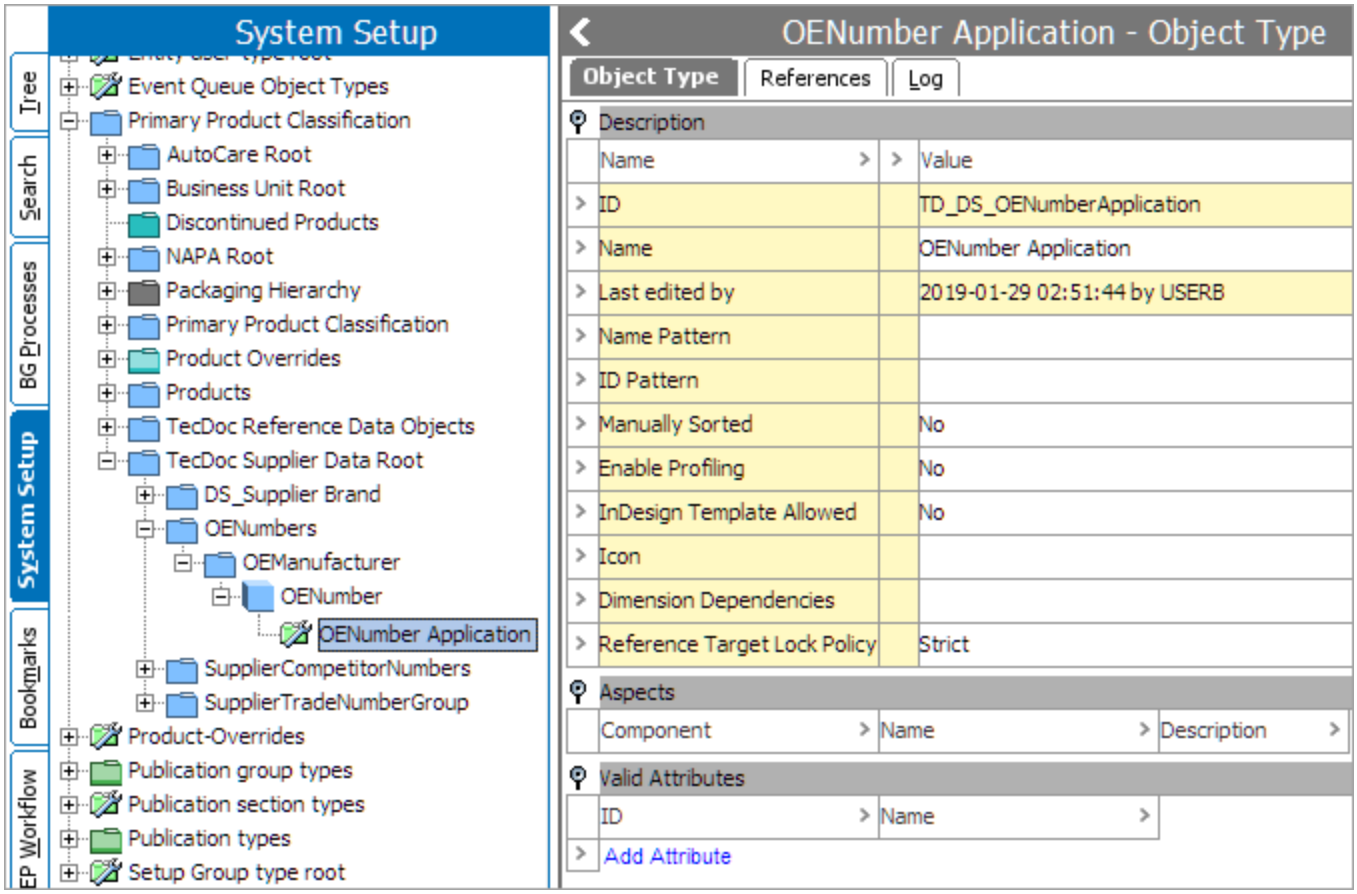
Below are the steps to configure in the workbench.

**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'

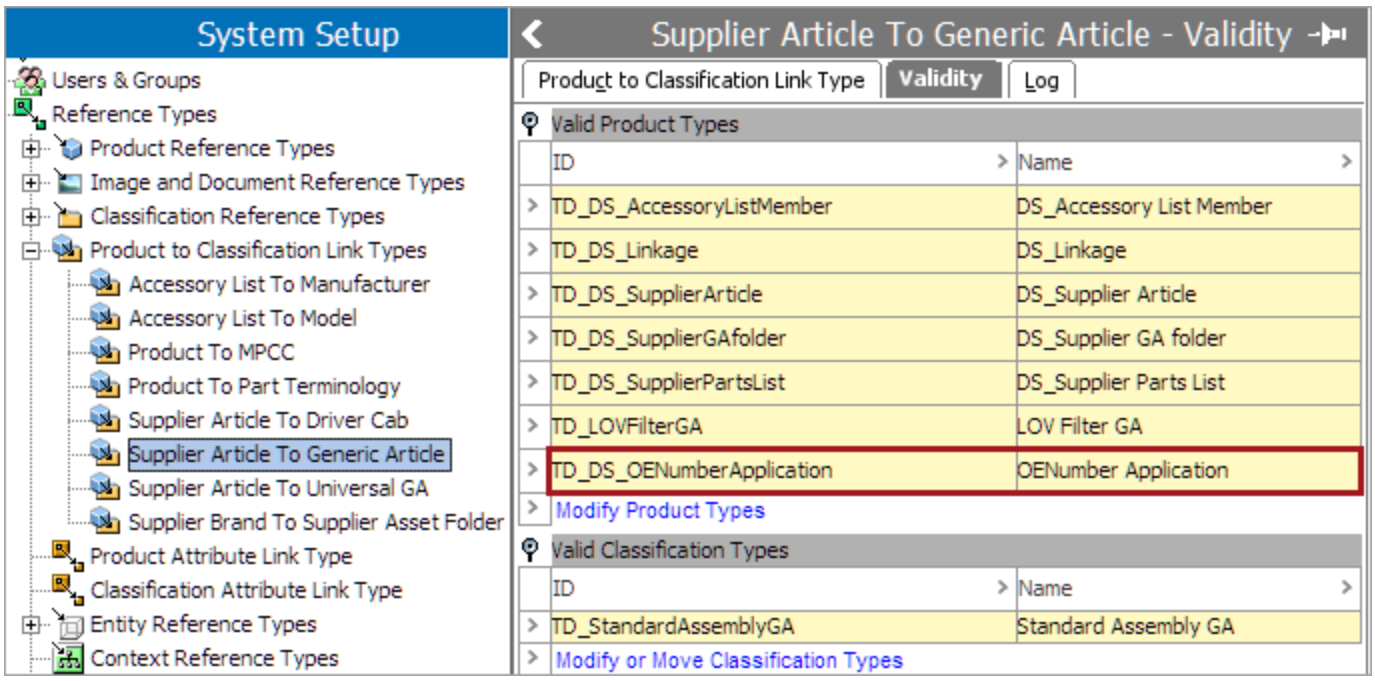
  Below is an example of a created OE Number application for the TecDoc solution.




3. 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_CompetitorNumberApplication' as valid Product Type to 'TD_SupplierArticleToGenericArticle' Classification Link Type

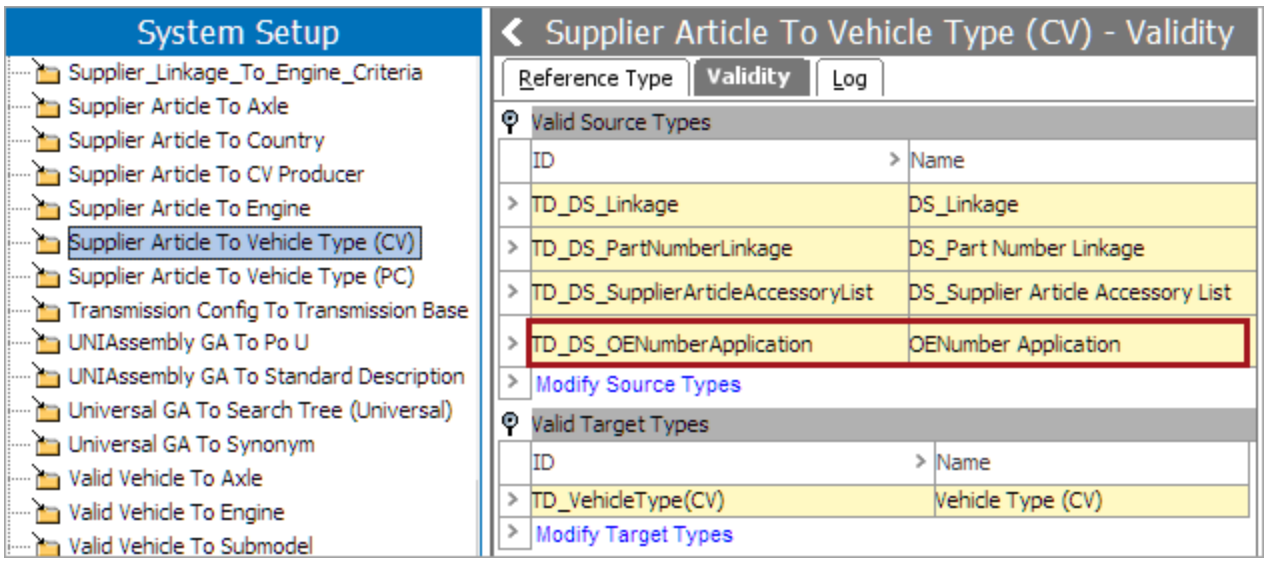
Below is an example of OE Number application being defined as the valid Product Type to 'TD_SupplierArticleToGenericArticle' Classification Link Type.




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_CompetitorNumberApplication' as valid Source Type to 'TD_SupplierArticleToVehicleType(CV)' and/or 'TD_SupplierArticleToVehicleType(PC)' Classification Reference Type

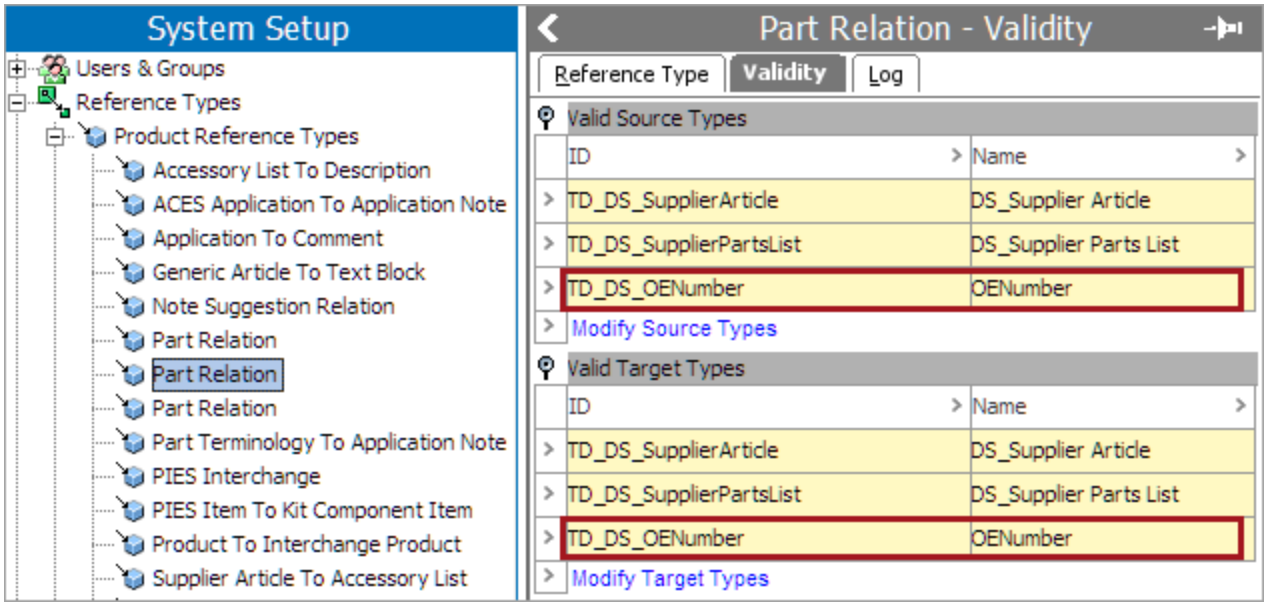
Below is an example of OE Number application object type being defined as the valid Source Type to 'TD_SupplierArticleToVehicleType(CV)' Classification Reference Type.




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.

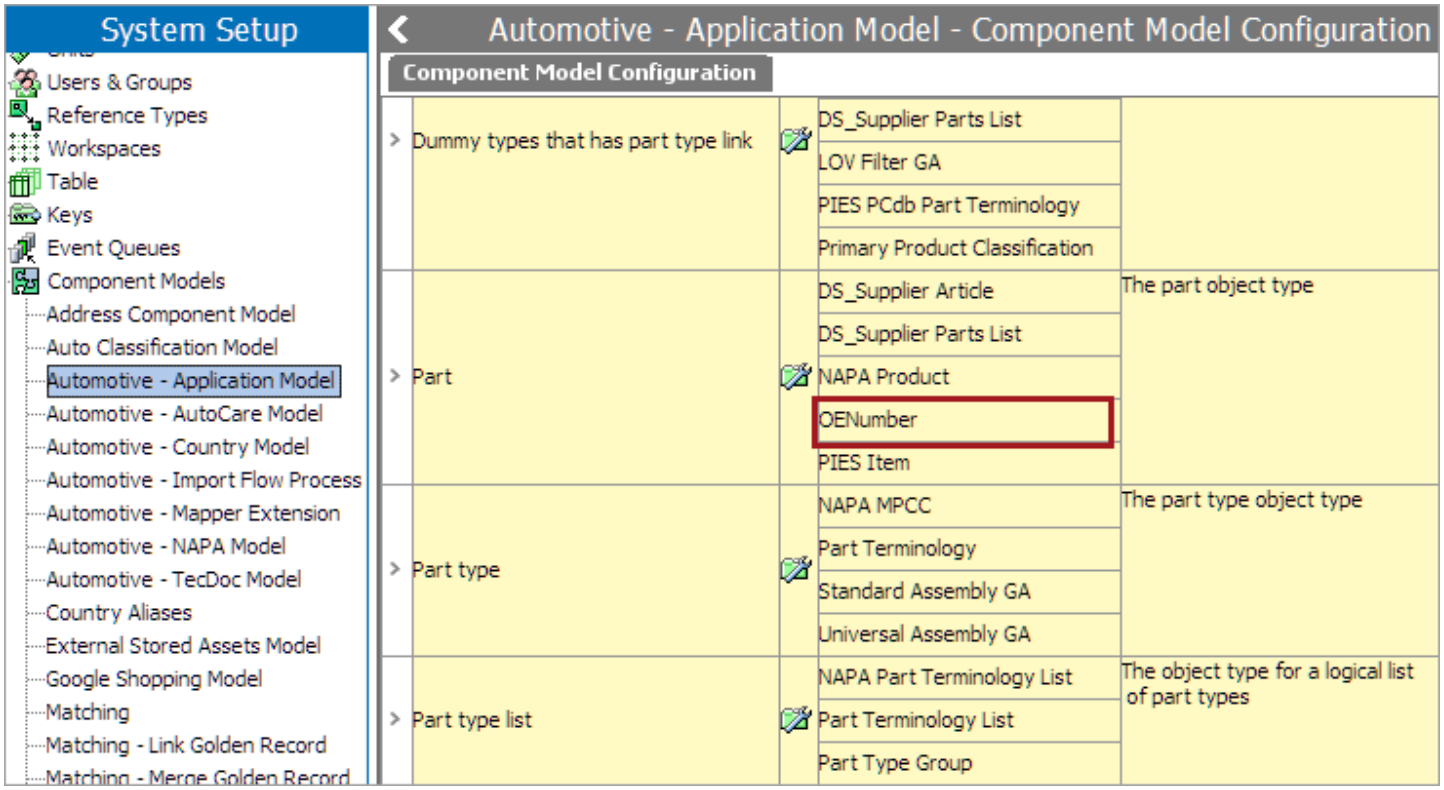
Below is an example of OE Number object type being defined as the valid Source Type and valid Target Type to 'TD_PartRelation' Product Reference Type.



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'

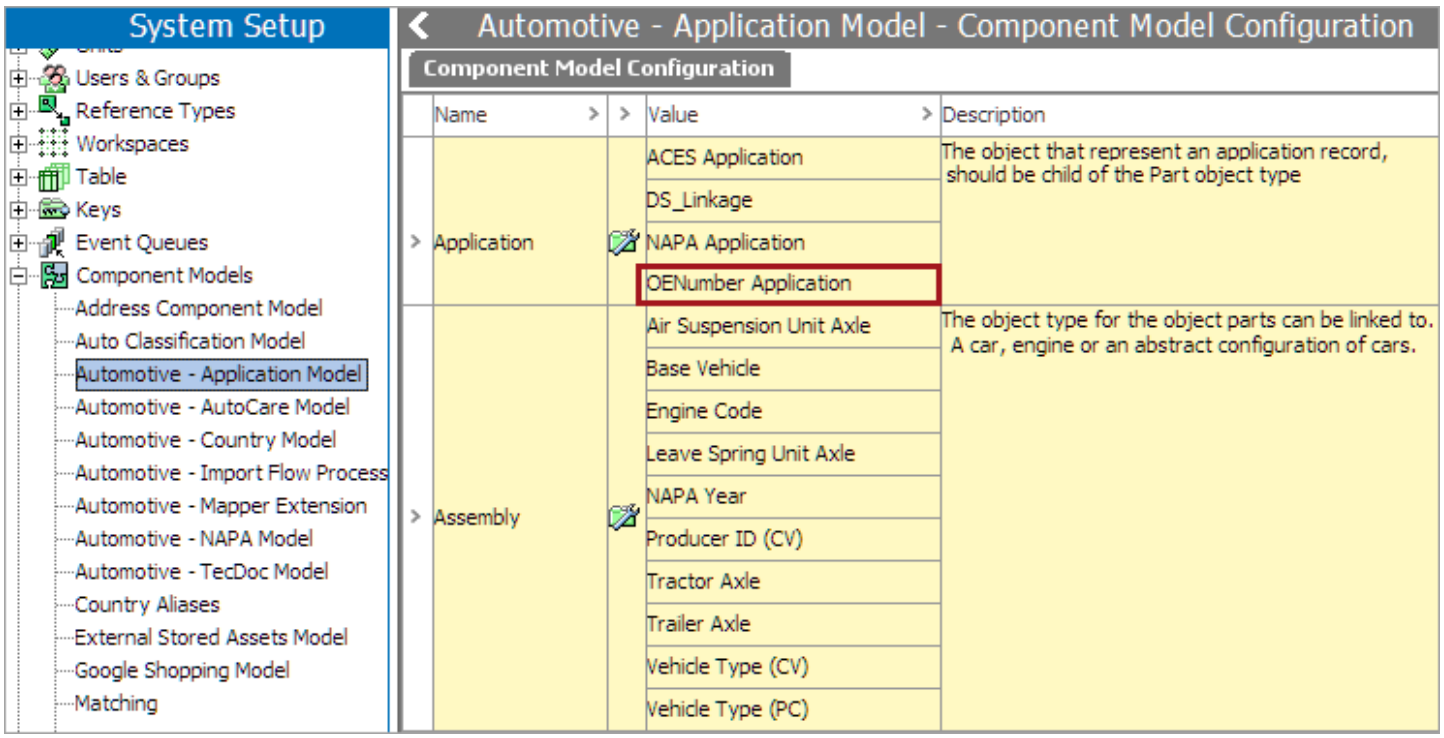
Below is an example of OE Number application object type being added to 'Part' aspect in Automotive - Application Model.




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_CompetitorNumberApplication'

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
DS_Linkage
NAPA Application
OENumber Application
The object that represent an application record, should be child of the Part object type
Assembly
Air Suspension Unit Axle
Base Vehicle
Engine Code
Leave Spring Unit Axle
NAPA Year
Producer ID (CV)
Tractor Axle
Trailer Axle
Vehicle Type (CV)
Vehicle Type (PC)
The object type for the object parts can be linked to. A car, engine or an abstract configuration of cars.

# Configuring the Application Competitor or OE Part Numbers Table Header Component

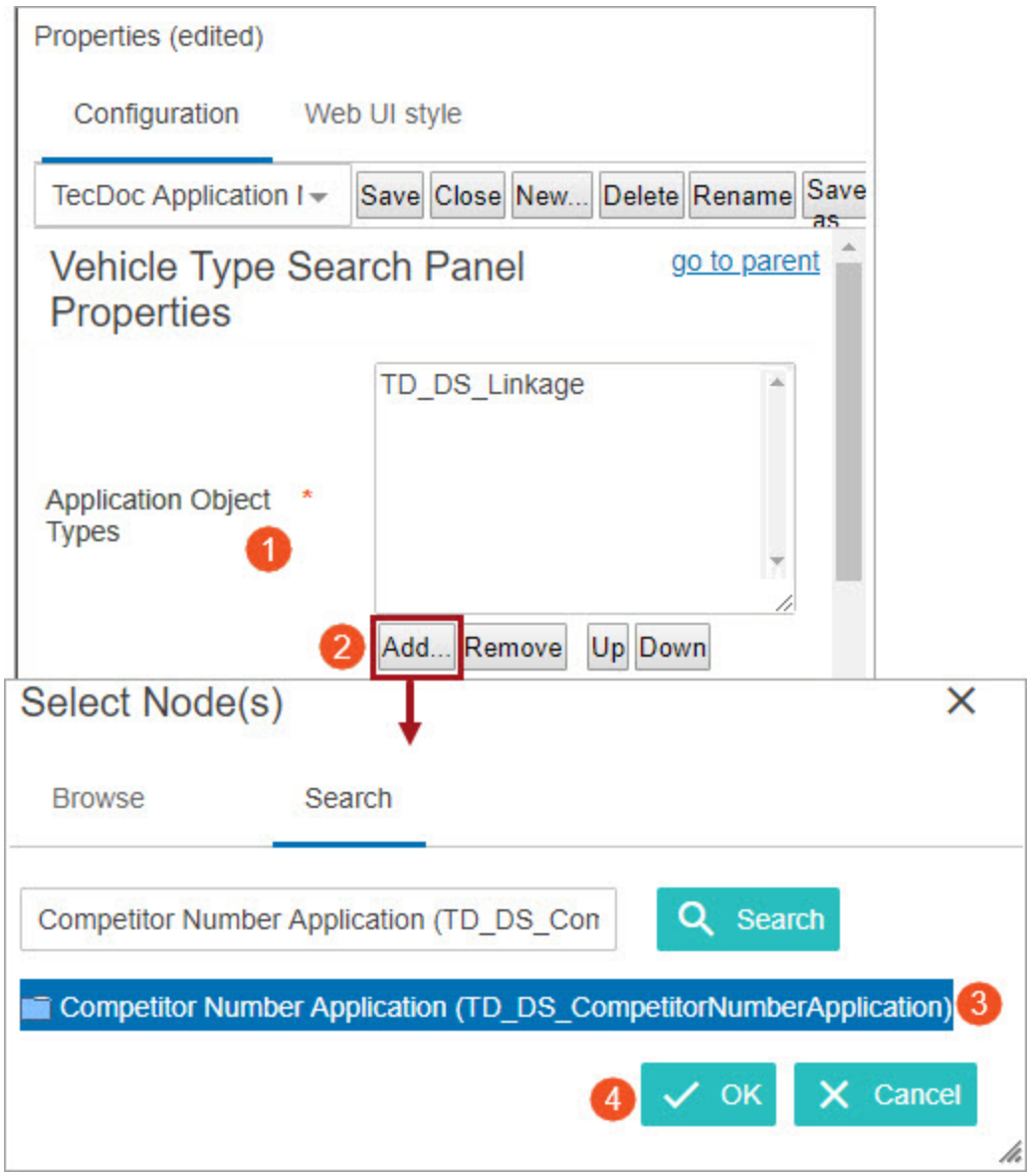
The Application Competitor or OE Part Numbers Table Header component is a Table Header component used to display the competitor or OE manufacturer part number and name in the Application Manager results table. Additionally, when a part number value within the column is clicked, a read-only dialog will display the competitor or OEM applications. For information about how to use this component and its prerequisites, refer to 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.

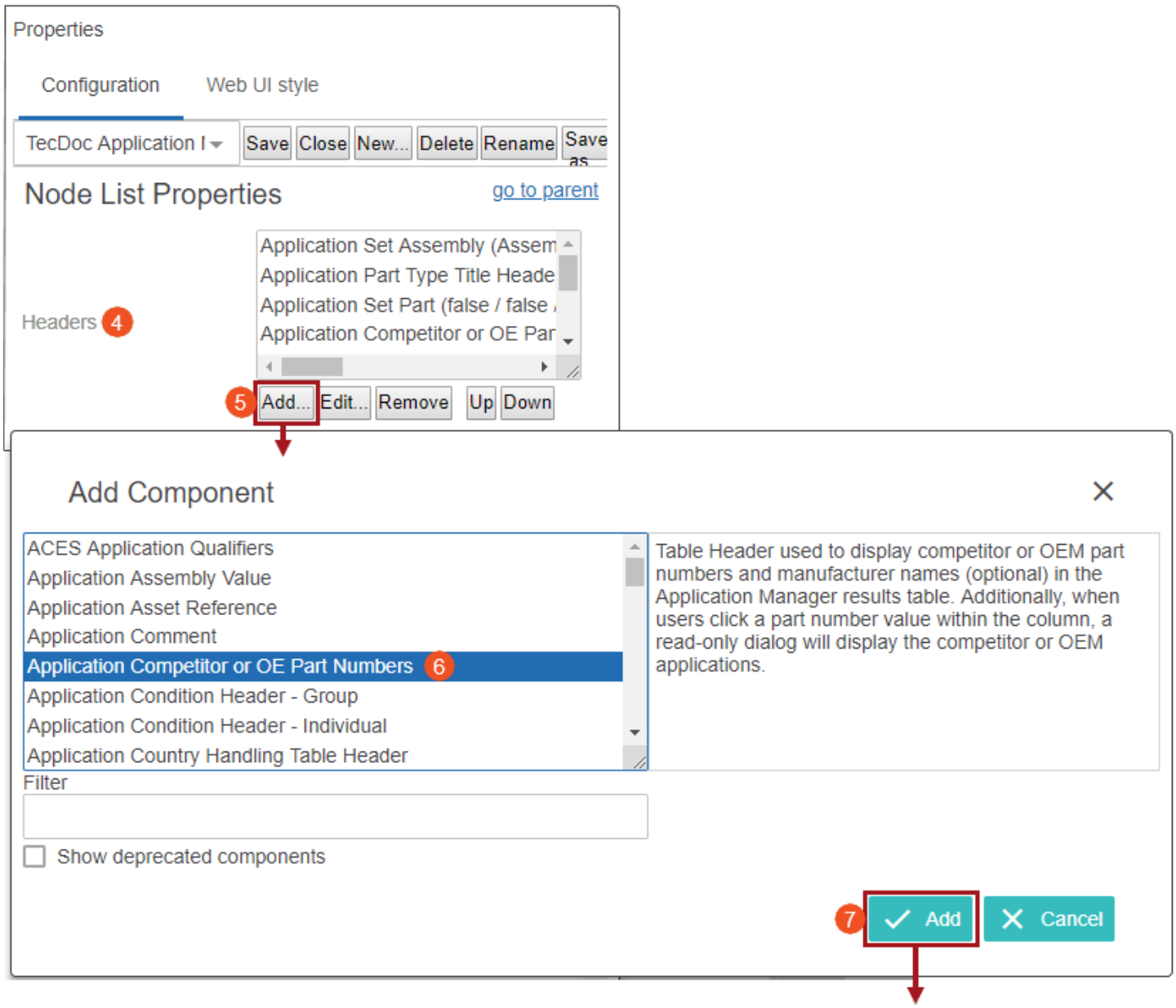
As a prerequisites, after Easy Setup actions, user needs to configure the standard data model in workbench along with the Web UI configuration steps mentioned below. For more information about how to setup the data model to use this component, refer to the **Configuring the Data Model for Application Competitor or OE Part Numbers Table Header** topic.

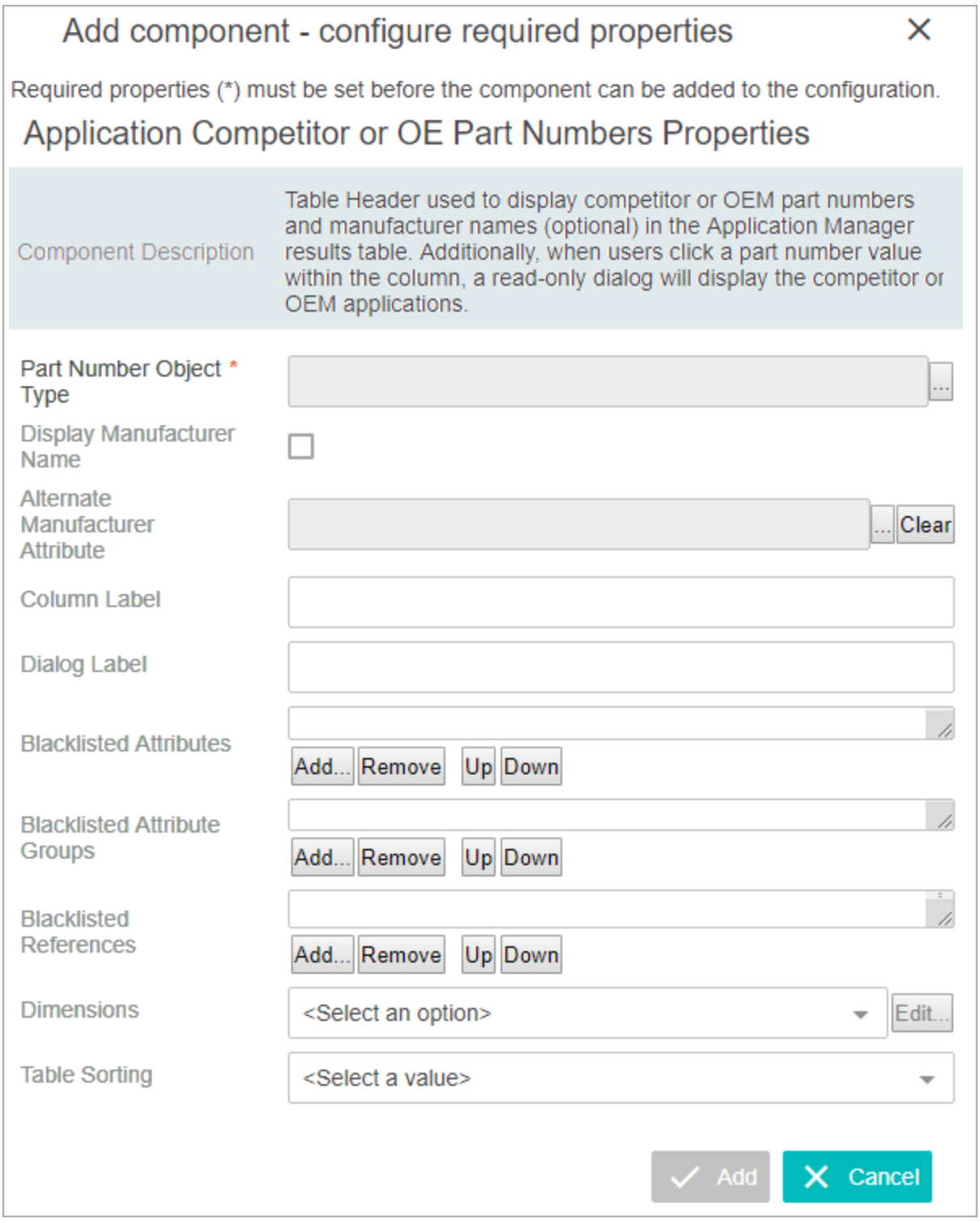
Below are steps to configure the Application Competitor or OE Part Numbers Table Header Component within an Application Manager Node List. Each of the parameters for the Application Competitor or OE Part Numbers Properties are described below.

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
Application Set Assembly (Assem
Application Part Type Title Heade
Application Set Part (false / false
Application Competitor or OE Par
Headers 4
5
Add...
Edit...
Remove
Up
Down
Add Component
ACES Application Qualifiers
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
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.
Filter
Show deprecated components
7
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

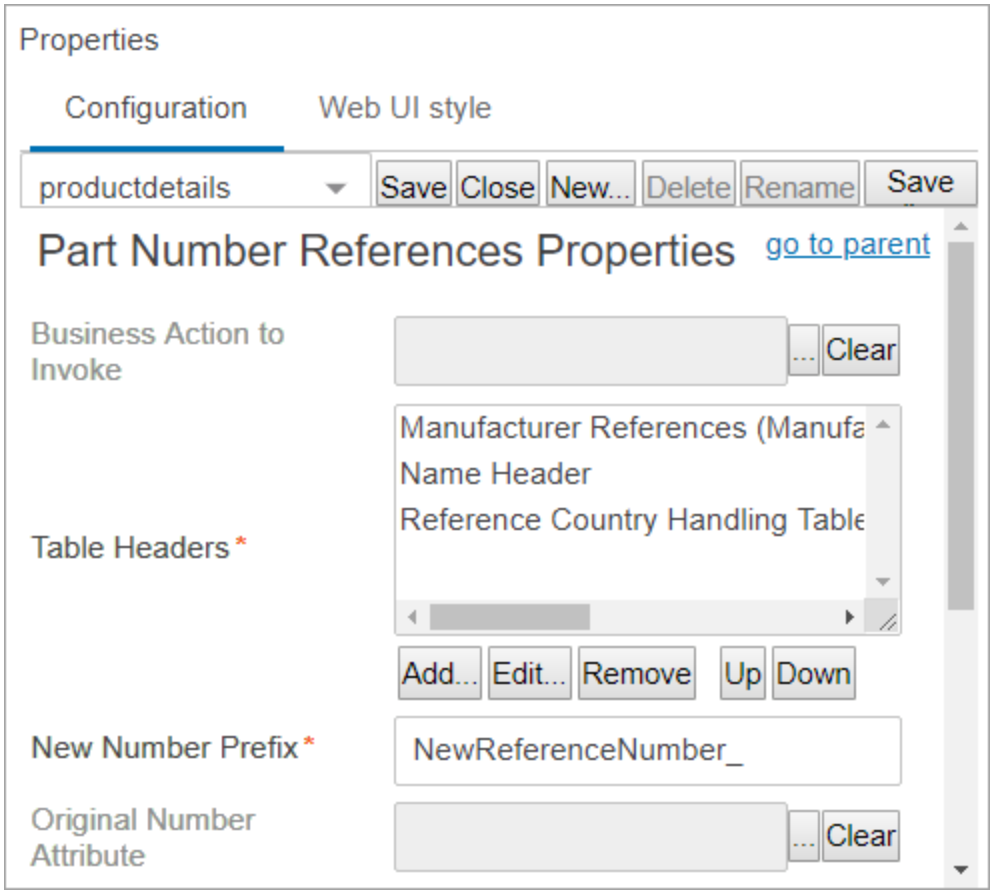
The Metadata Attribute References component is configured as a table header in many Web UI components. It is used to display the metadata attributes value of Reference and Link types within the Part Number References, Part Number Suggestions, Advanced Part Number References, and Advanced Part Number 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.

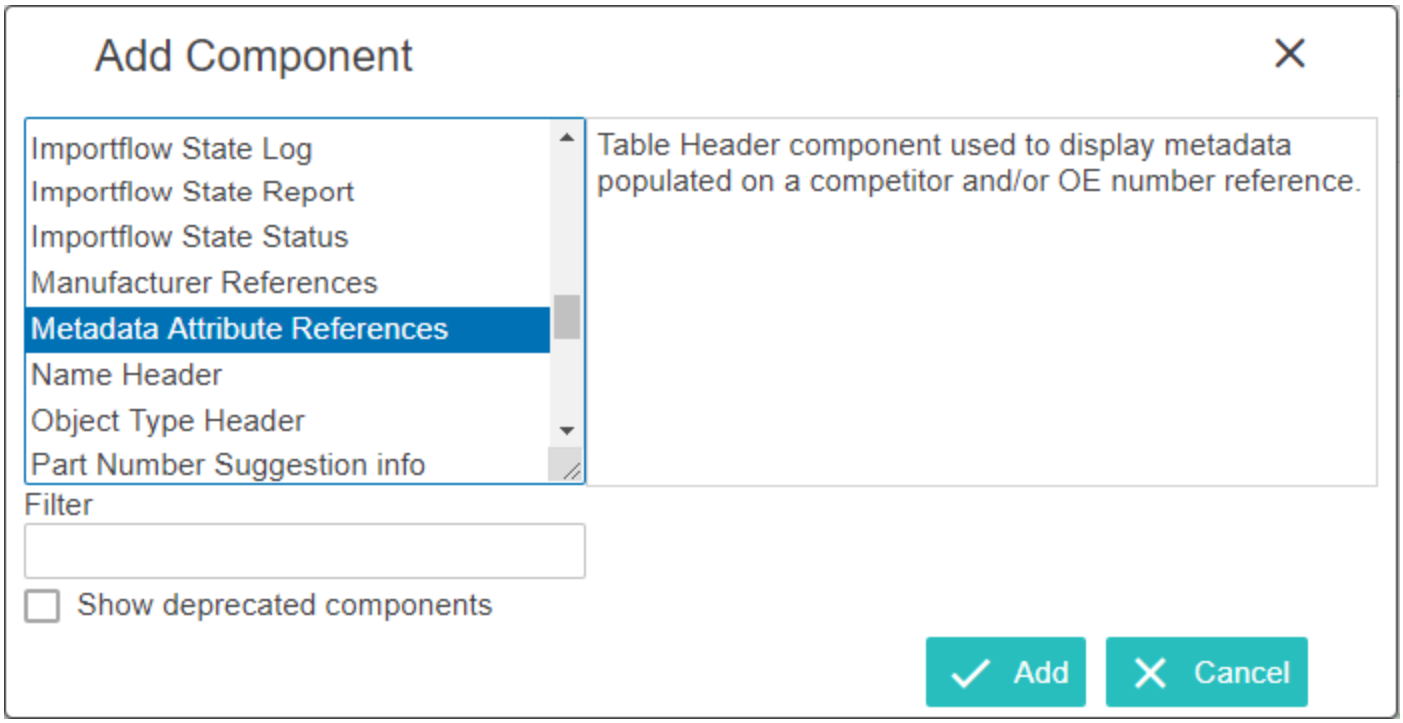
## Configuration Steps

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.

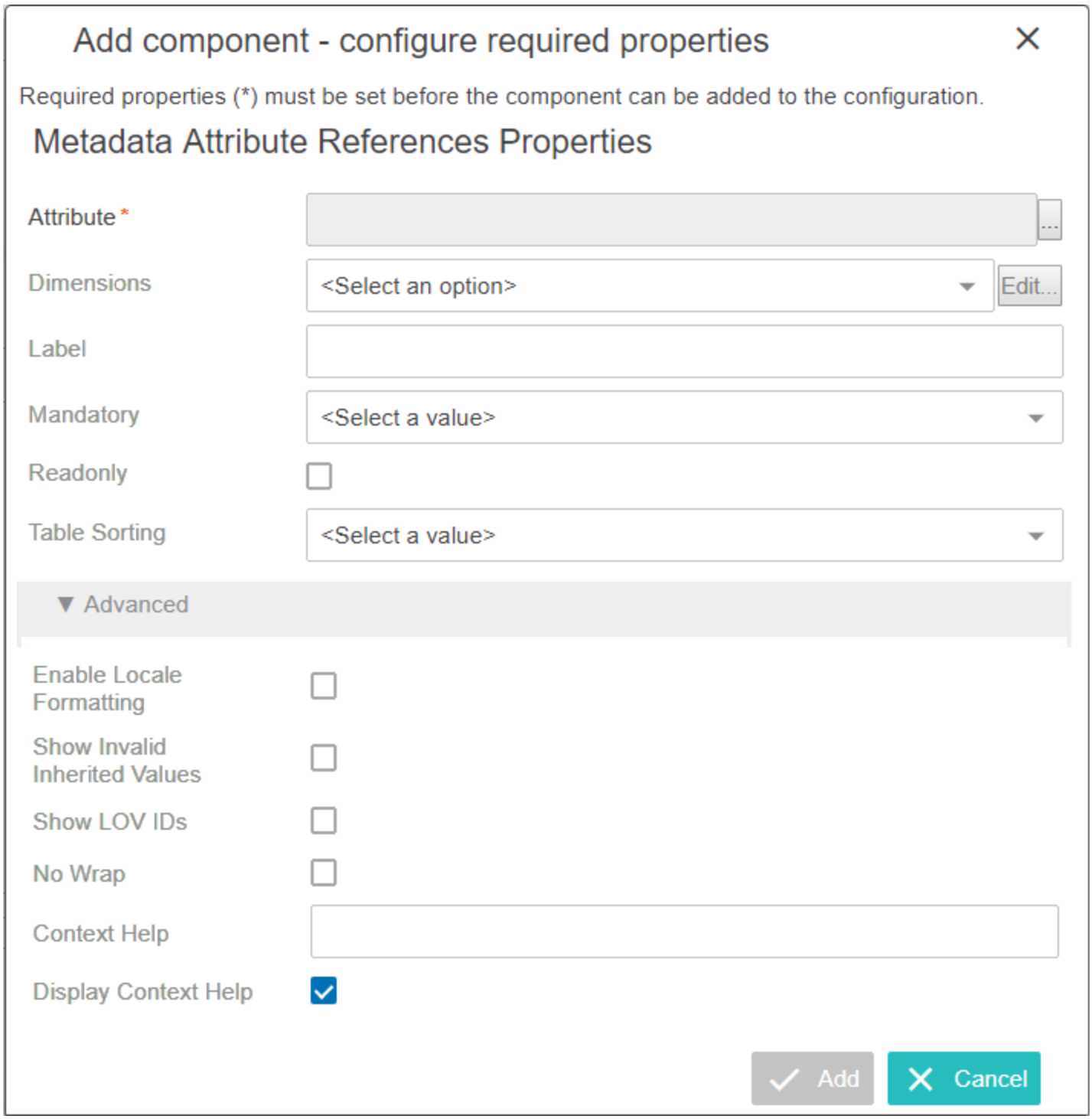
1. In the Web UI Designer, navigate to the Table Header / Header parameter for the component that will display metadata attributes. The example below displays the Table Header parameter within the Part Number References Properties dialog.



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:

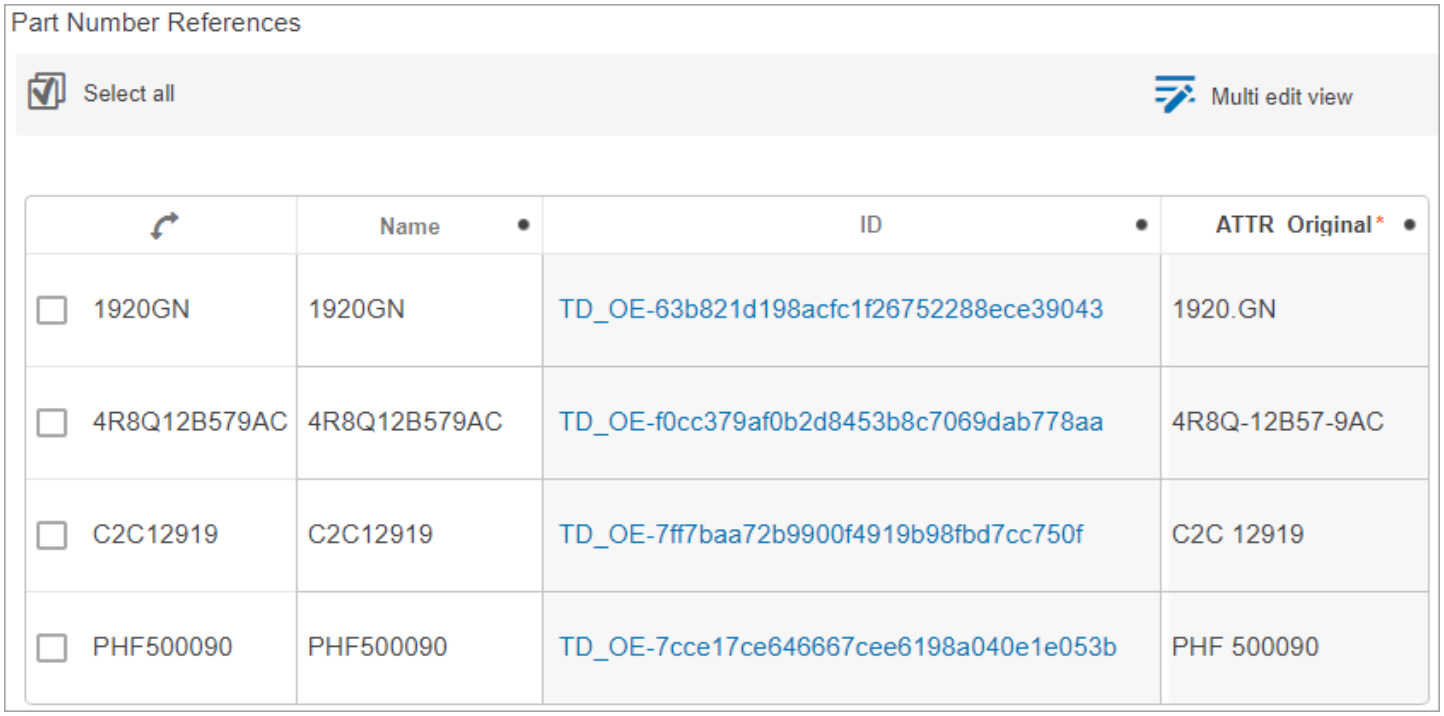



**Attribute:**The only mandatory parameter for this component. To select an attribute, click the ellipsis to the right of the Attribute field. A Select Node(s) dialog will appear and an attribute can be selected by browsing and navigating down to the desired attribute or by using the search feature.

**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'.




**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, refer to 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.

# 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, refer to 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, refer to **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, refer to **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, refer to **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, refer to **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, refer to **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, refer to **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.<br>**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, refer to **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, refer to **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, refer to **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, refer to **Business Action: Set Import Status Attributes**. |

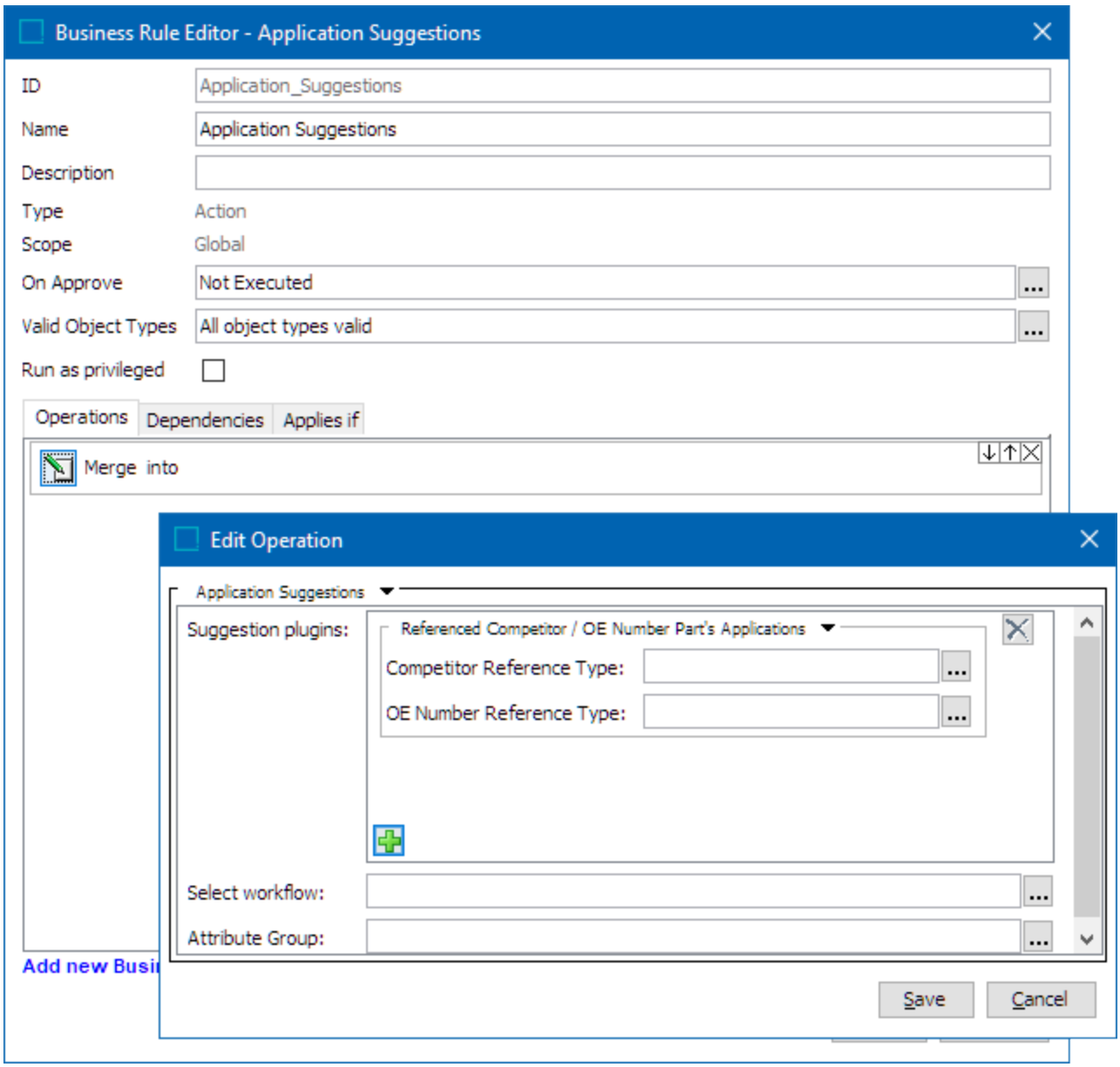
## Business Condition Operations

| Menu | Operation Name | Components required for use | 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 |

| Menu | Operation Name | Components required for use | Description |
|---|---|---|---|
| | | | specific part types do not apply to certain vehicle configurations. For more information, refer to **Business Condition: Check Path for Missing Application**. |
| TecDoc | TecDoc reference start condition | TecDoc Reference Import workflow, TecDoc Supplier Import workflow, and TecDoc Supplier controller group | Prevents the start of reference data import if there is a supplier data import running.<br>For more information, refer to **Business Condition: TecDoc Reference Start Condition**. |
| TecDoc | TecDoc supplier start condition | TecDoc Supplier Import workflow, TecDoc Reference Import workflow, and TecDoc Reference controller group | Prevents the start of supplier data import if there is a reference data import running.<br>For more information, refer to **Business Condition: TecDoc Supplier Start Condition**. |

## Business Action: Application Suggestions

The 'Application Suggestions' operation is found within the STEP Workbench Business Rule Editor under the Automotive menu, and requires multiple parameters to be populated with values. Setup is required in the STEP Workbench for the business action to be available to users.




When used within a business action, this operation checks the selected part for any application suggestions. When an application is suggested that meet the condition defined in the suggestion plugin, it will initiate the selected part into a workflow configured within the 'Select workflow' parameter of the business action.

If the selected part finds no suggested application based on the condition defined within the 'Suggestion plugins' parameter, then the part will not be initiated into the workflow.

As a prerequisite, one unique workflow that can handle the part is necessary. This allows each customer to determine their own strategy for managing parts that have application suggestions.

**Note:** How the customer handles such part in a different workflow is beyond the scope of this topic. It is a customized solution that the Solution Consultant can implement on to their system.

### Configuring Business Action for Application Suggestions

To properly configure this business action to handle parts that has application suggestions, the following is needed:

- Create / Identify a valid workflow that handles parts that has application suggestions.
- Configure a business action to use the 'Application Suggestions' operation

## Create / Identify a Valid Workflow

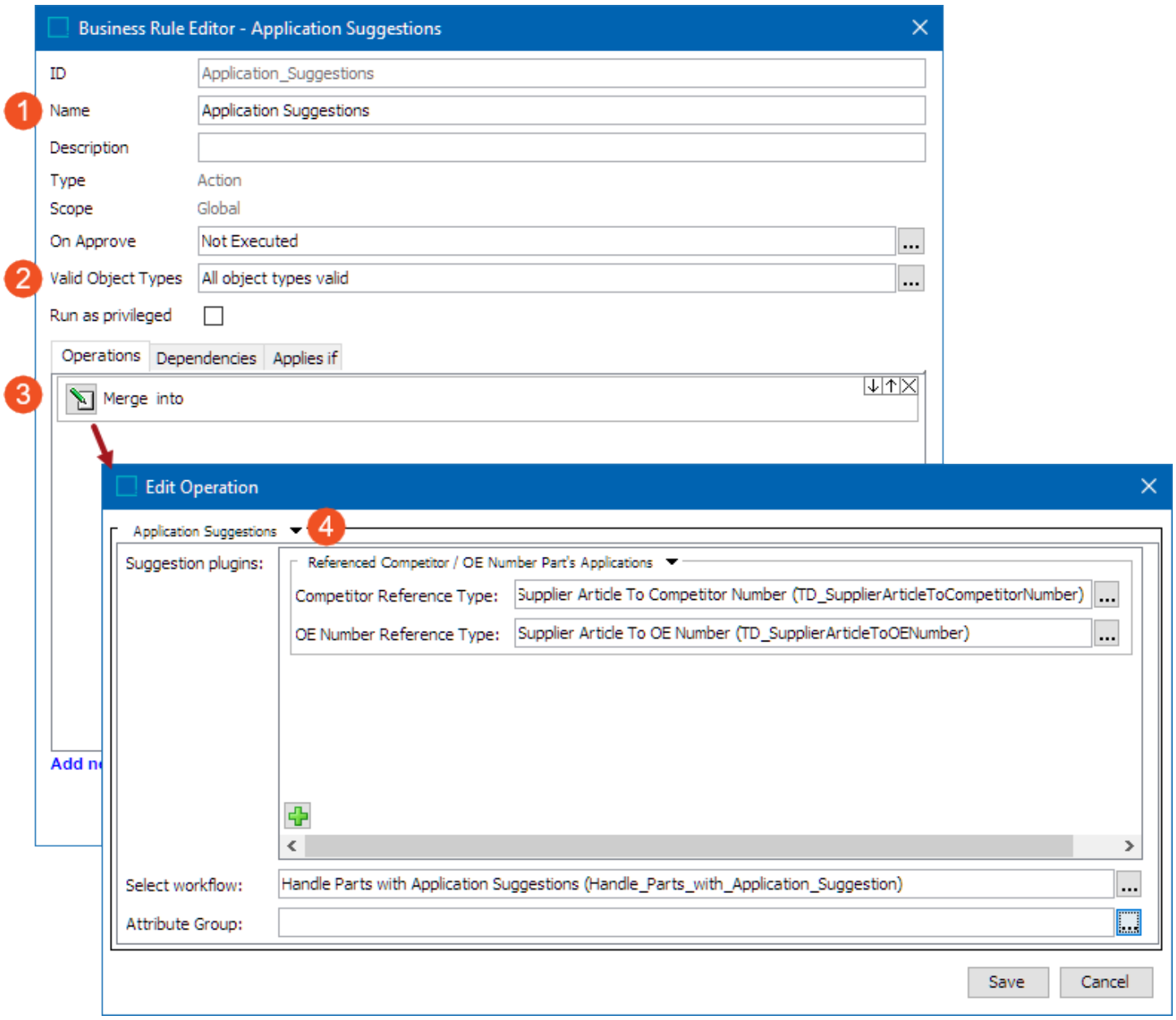
1. Create the necessary workflow (or verify that it exists) to be used to handle the parts that have application suggestions.
2. Make the workflow valid for the object type to be handled.

   For example:

   - The workflow used to handle PIES Item that has application suggestions must be valid on the AC_PIESItem object type.
   - The workflow used to handle TecDoc Supplier article that has linkage suggestions must be valid on the DS_Supplier Article object type.
   - The workflow used to handle NAPA Product that has application suggestions must be valid on the NAPA_Product object type.

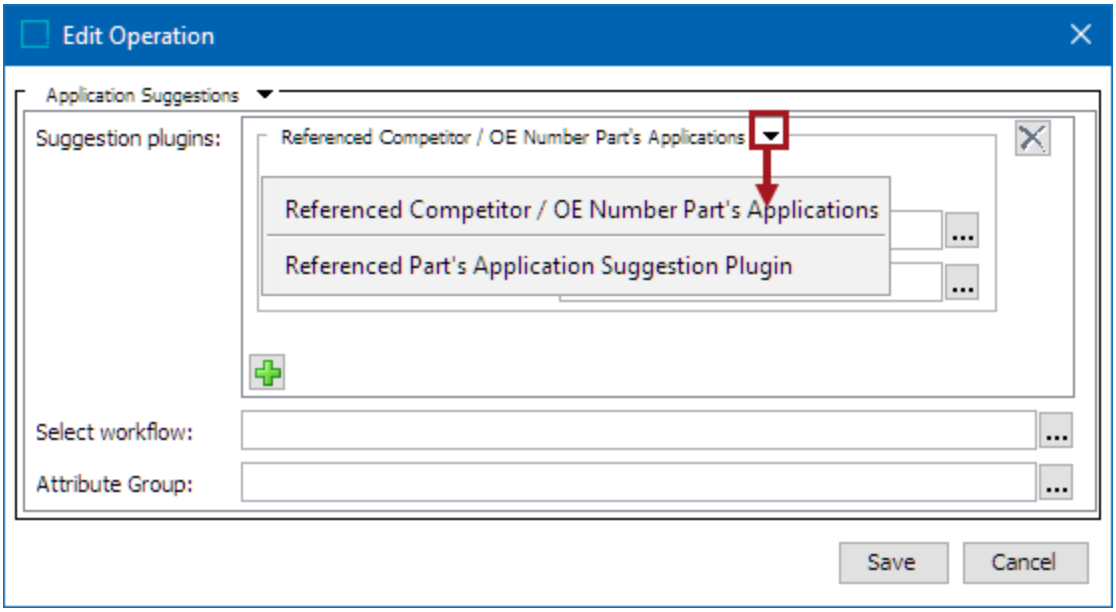
## Configure a Business Action to Use the 'Application Suggestions' Operation

1. Create a business action and give it a name that users can easily identify. In the example below, the 'Application Suggestions' 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 example below, 'All object types valid' is selected.
3. On the Operations tab, click the 'Merge into' icon, and the Edit Operation dialog will display.
4. Click on the Edit Operation dropdown, click 'Automotive,' and then click '**Application Suggestions.**'



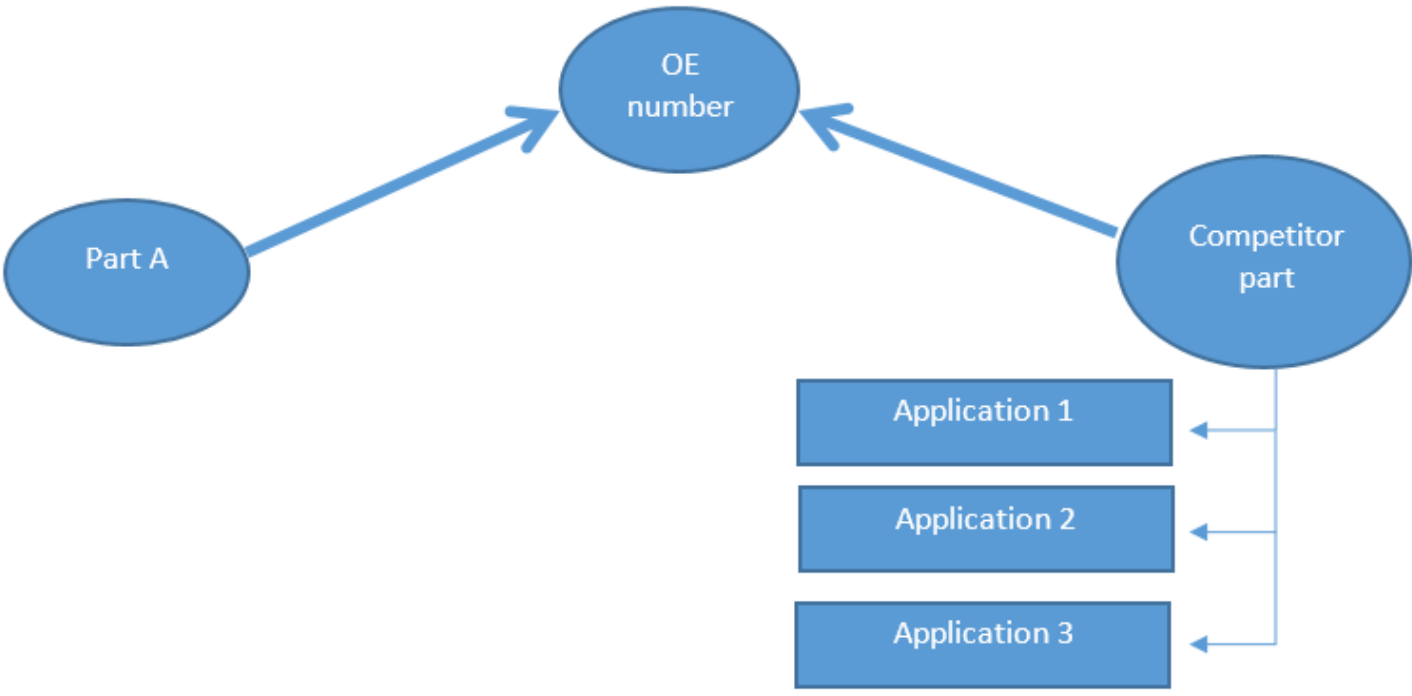

5. The following parameters will be displayed. Configure the parameters.
   - **Suggestion plugins:** Within this parameter, the user can add one or more plugins by clicking the add icon ( ). The criteria defined in this plugin will set the condition to look for the potential application records for the part. It is mandatory to have at least one plugin so that when a part is selected, the system would follow the configurations provided in the plugin and look for the potential application records in accordance.

     In the dropdown that is available next to the 'Suggestion plugins' parameter, the following options are available :

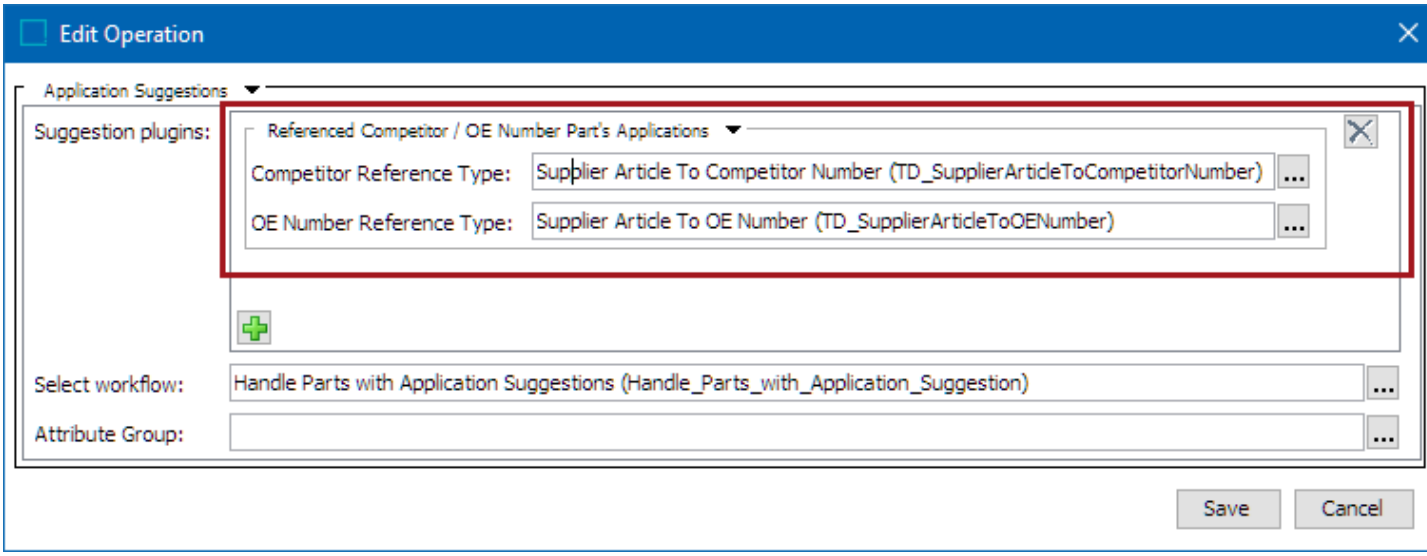



- **Referenced Competitor / OE Number Part's Applications:** This plugin is used to suggest the Competitor's application records where the Competitor Number is referencing the OE Number of the current part. Selecting this plugin requires the configuration of two additional parameters with the associating reference types linked between the part and the OE Number, and the reference types linked between the Competitor and OE Number.

  Below is a chart explaining the path followed by the plugin suggesting application records to a part. Application 1, Application 2, and Application 3 are suggested as the potential application records to Part A.

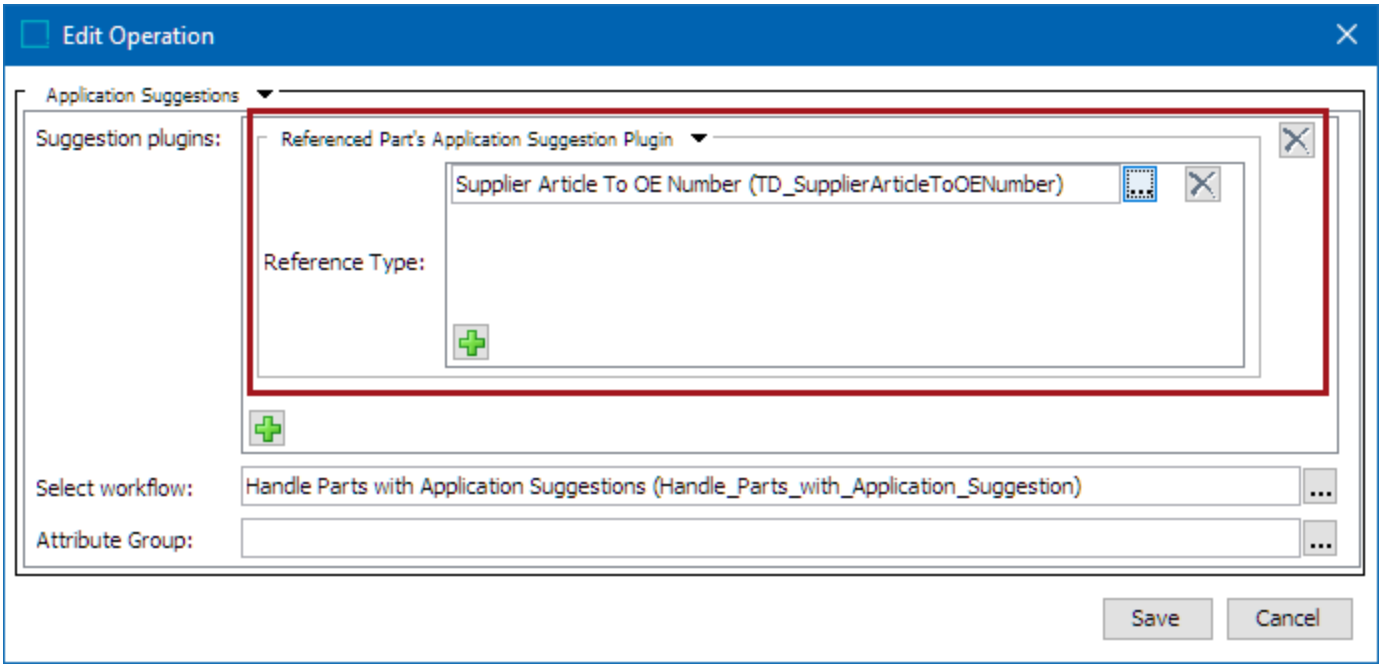


Additional configurations are required for this plugin.




- **Competitor Reference Type:** 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 look for 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:** 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 associating reference type linked between the two parts must be configured.




- **Reference Type:** 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 look for 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'

- **Select workflow:** Click the ellipsis button (...) to find and select the workflow. The configured workflow should be able to handle the parts that have application suggestions.
- **Attribute Group:** When configured with an attribute group, this optional parameter , 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. The attribute values present within this attribute group then 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 will suggest two different potential application records.

**Note:** Upon configuration of the 'Applications Suggestion' business action, the Automotive solution only provides an option to initiate the parts with application suggestions in another workflow. Users need to determine their own strategy for how to handle these parts in another workflow.

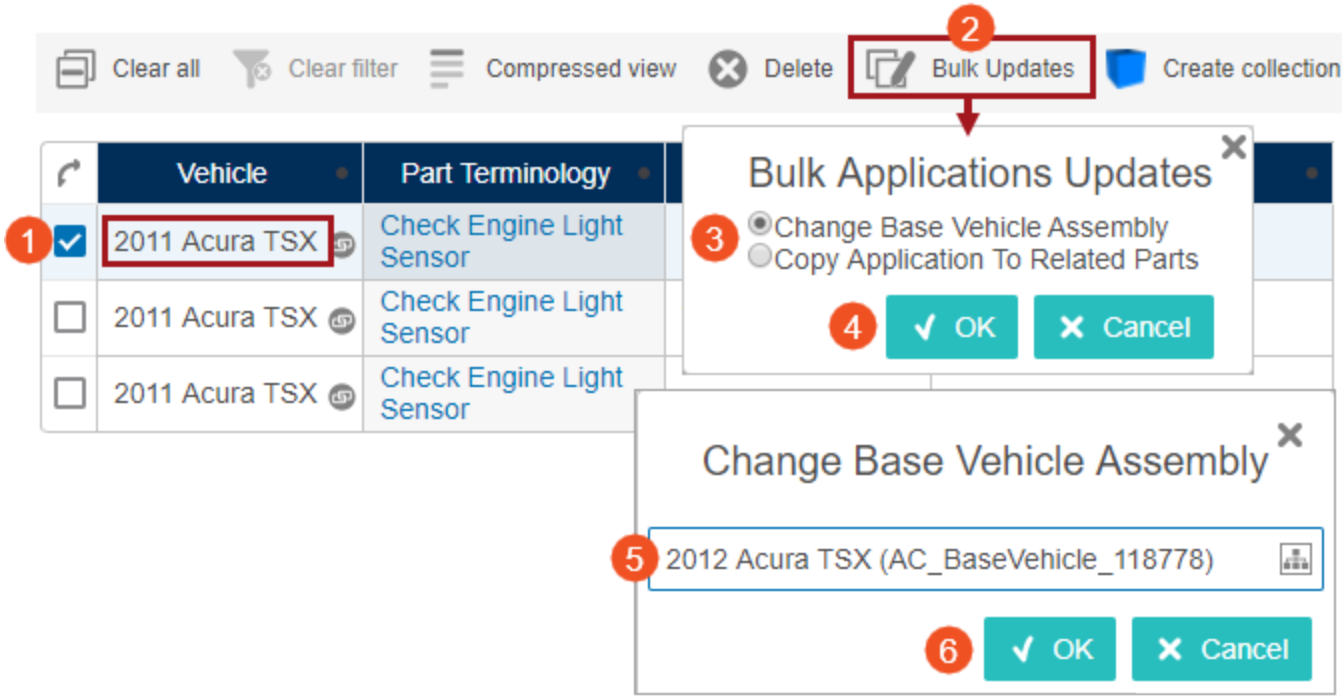
6. Click the **Save** buttons to save and close the business rule.

For more information on creating and/or editing business rules, refer to the **Creating a Business Rule, Function, or Library** topic, and the **Editing a Business Rule or Function** topic within the **Business Rules** guide.

## 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, refer to the **Missing Application Coverage Functionality** topic within this guide.

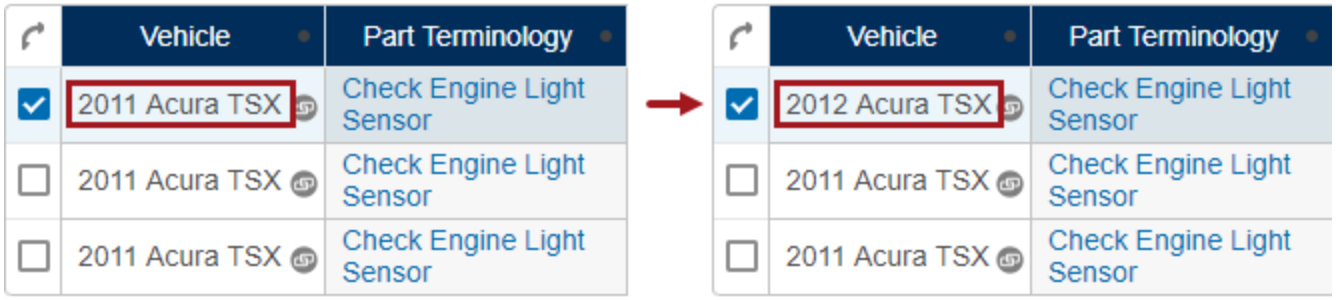
Once configured, a Bulk Updates button within an Application Manager can be used to change the assembly for one or more existing application. Below are the steps for using the business action in Web UI.




**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.

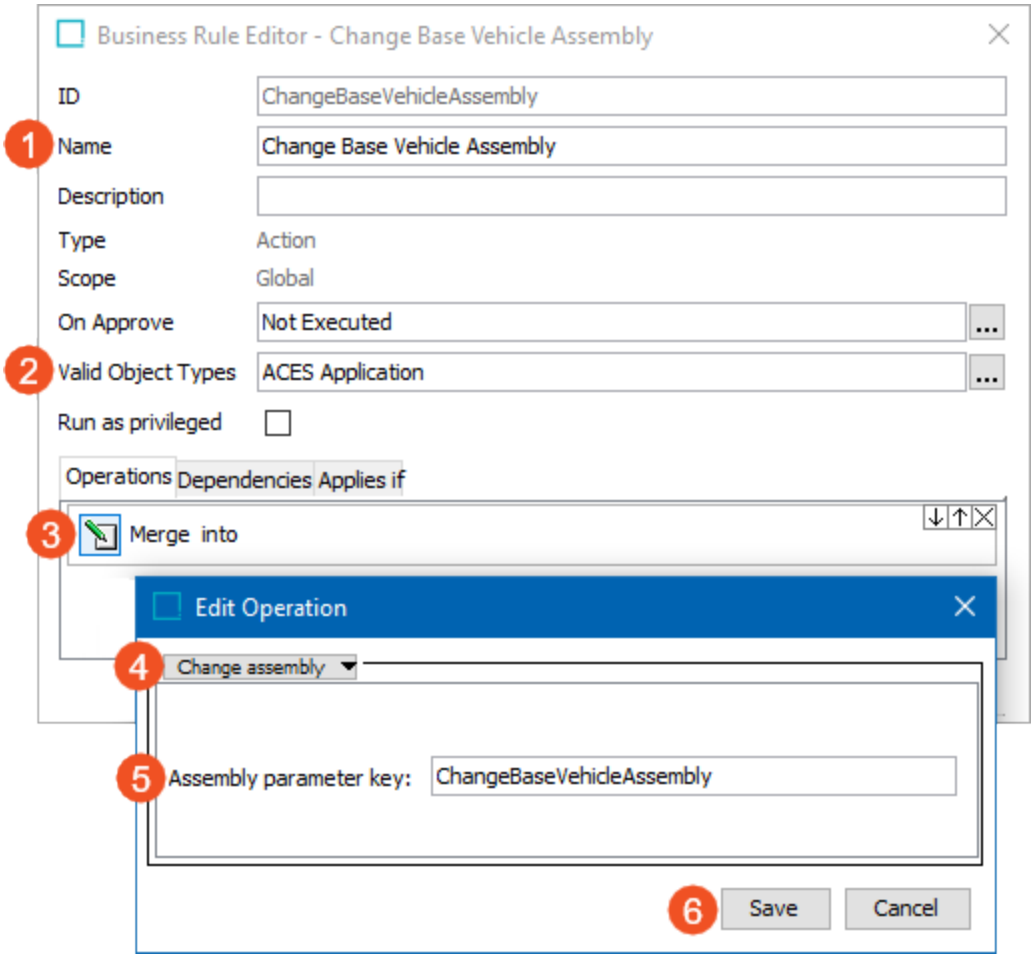



Setup is required within both STEP Workbench and Web UI for the action to be available to users. For more information, refer to the configuration topics below:

- 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.

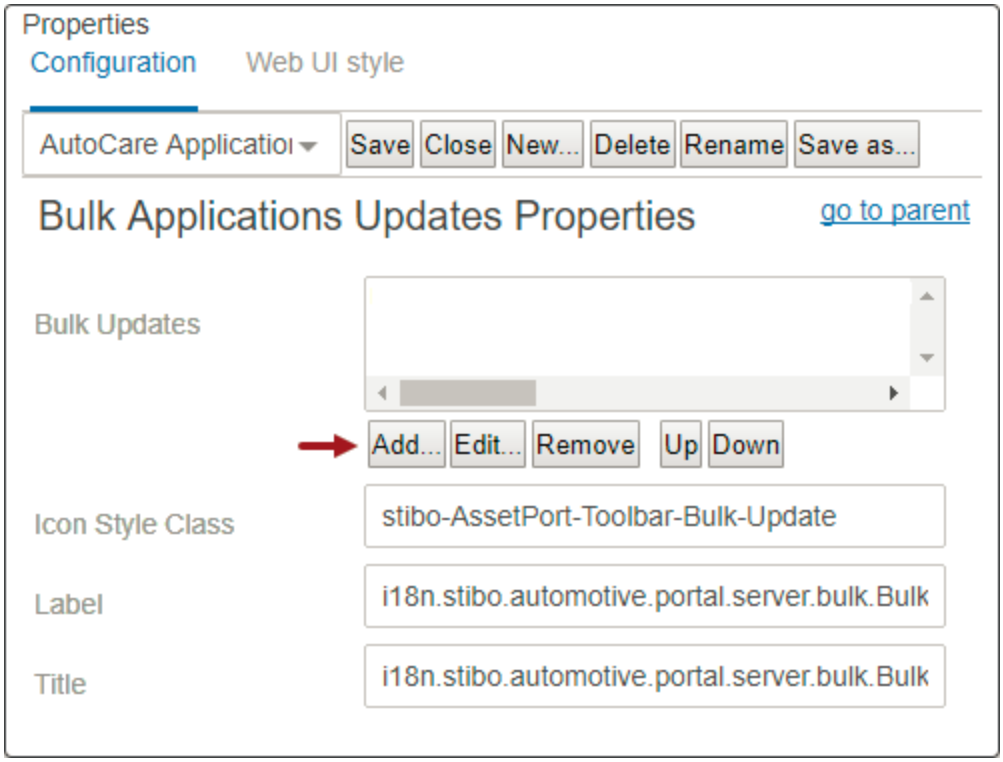
6. Click the **Save** button and continue to the next topic, **Configuring the Change Assembly Business Action in Web UI**.

### Configuring the Change Assembly Business Action in Web UI

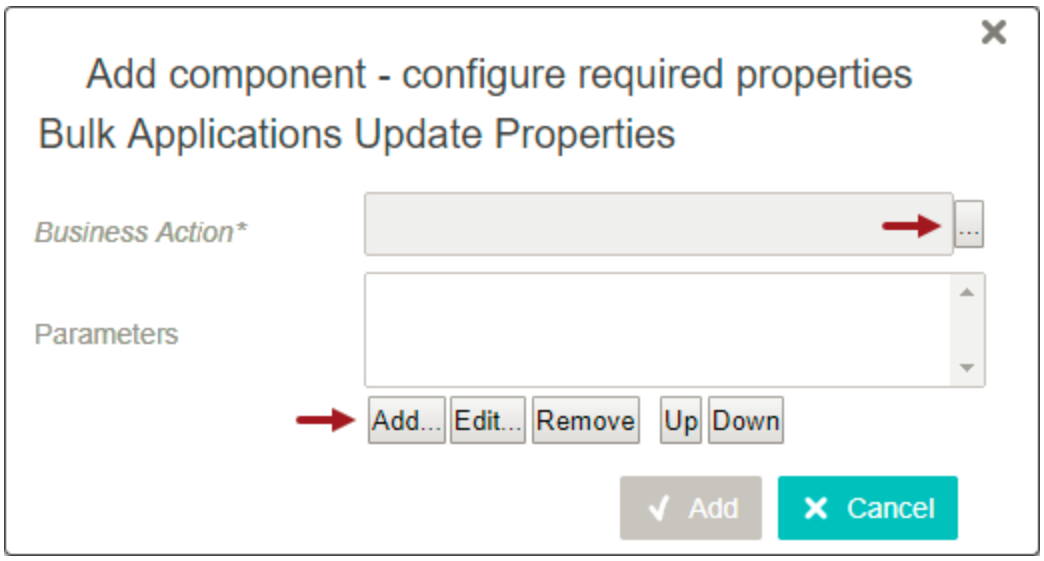
Once the business action has been configured in the workbench, configuration within Web UI is necessary. This section addresses the steps necessary within Web UI.

**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... .'

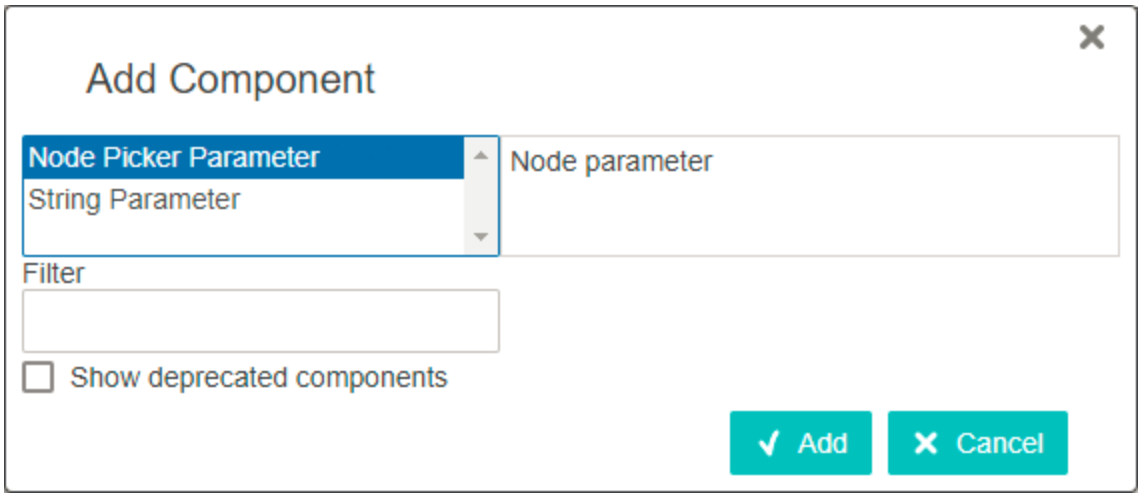
1. Access the Application Manager screen where the business action needs to be available to users.
2. Access the Designer > navigate to Node List Properties > Child Components > Actions > Double click **Bulk Applications Updates** and the 'Bulk Applications Updates Properties' dialog will display (as shown below).



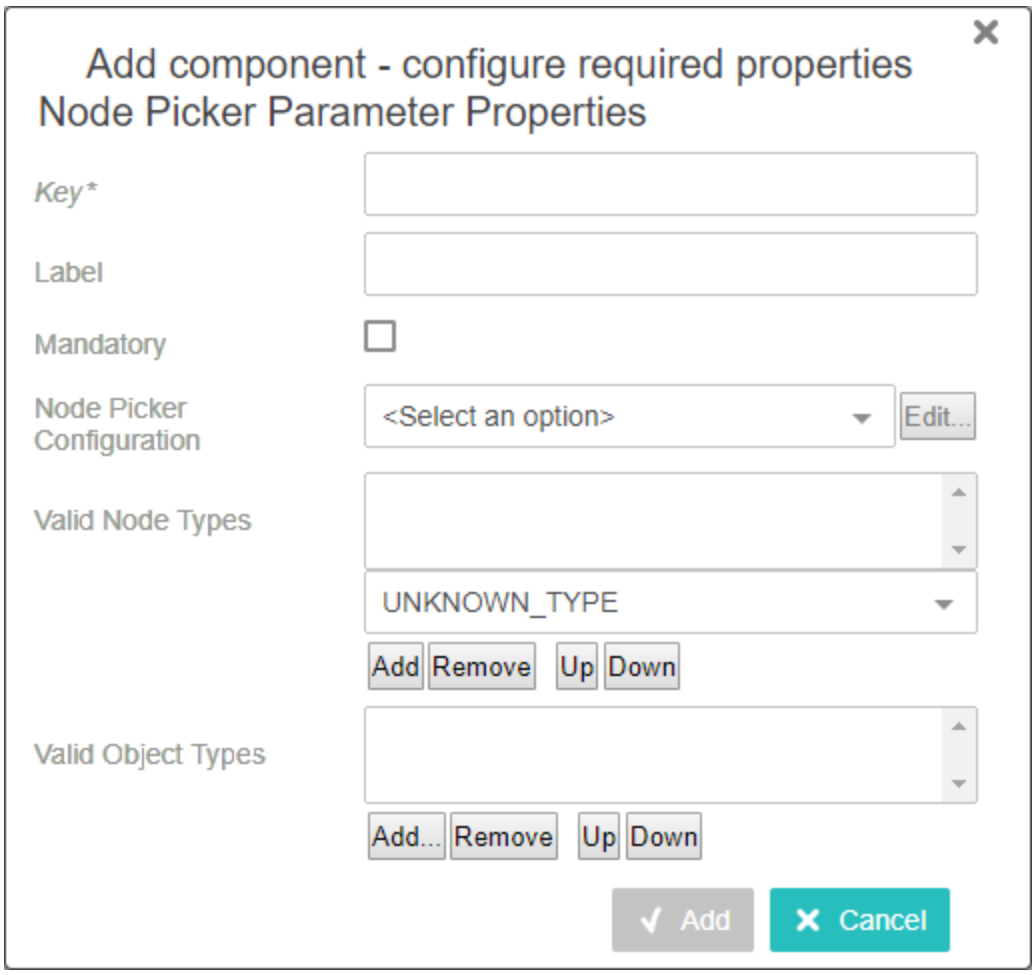

3. Click the **Add** button beneath the Bulk Updates parameter, and the 'Add component - configure required properties' for the 'Bulk Applications Update Properties' dialog will display (as shown below).




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).



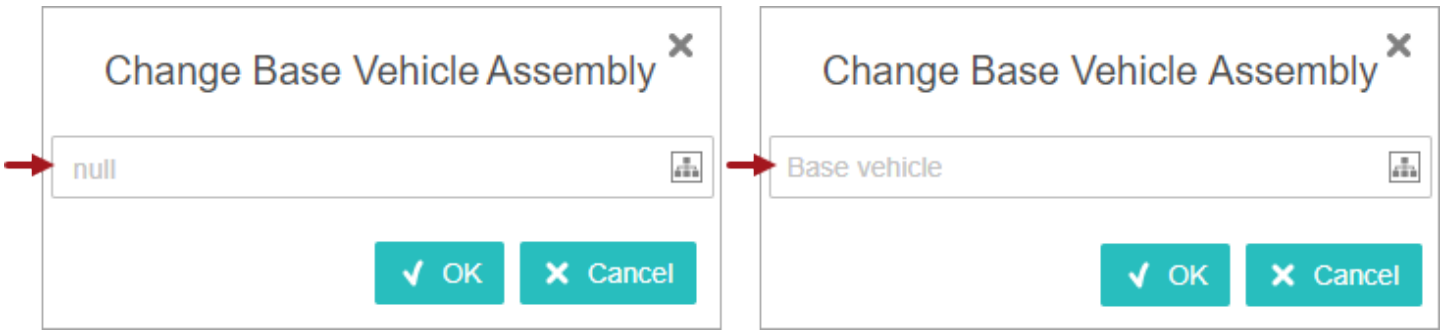

6. Select **Node Picker Parameter** > click the **Add** button to close the dialog, and the 'Add component - configure required properties' for the 'Node Picker Parameter Properties' dialog will display (as shown below).



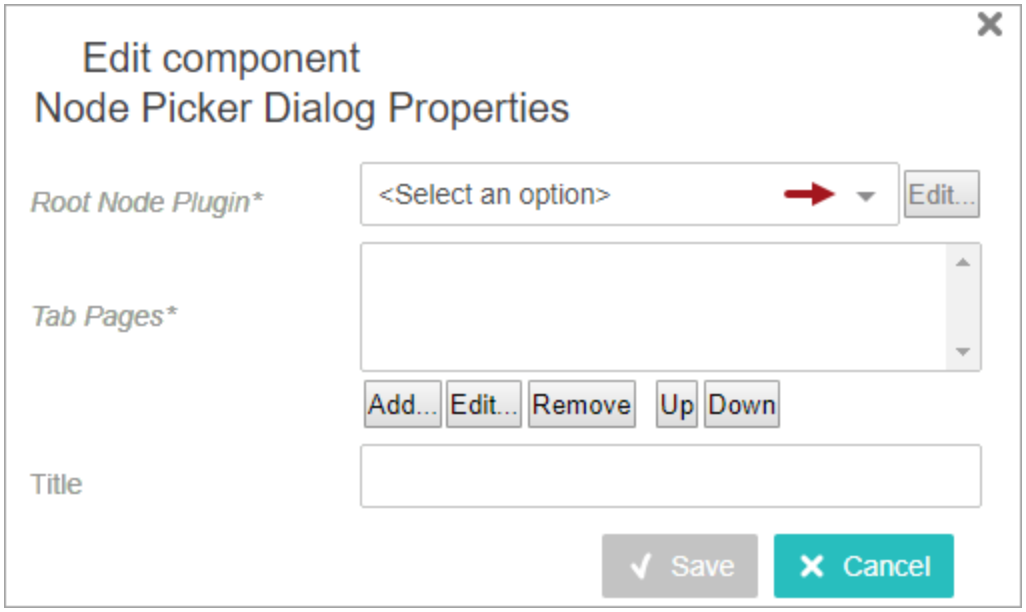

7. Within the Key parameter, enter the exact key created within the 'Assembly parameter key' parameter during step 5 of the **Configuring the Change Assembly Business Action in Workbench** topic. This is the only required parameter.

   **Important:** If this key is not entered exactly in both places, then the business action will not properly function within the Web UI.

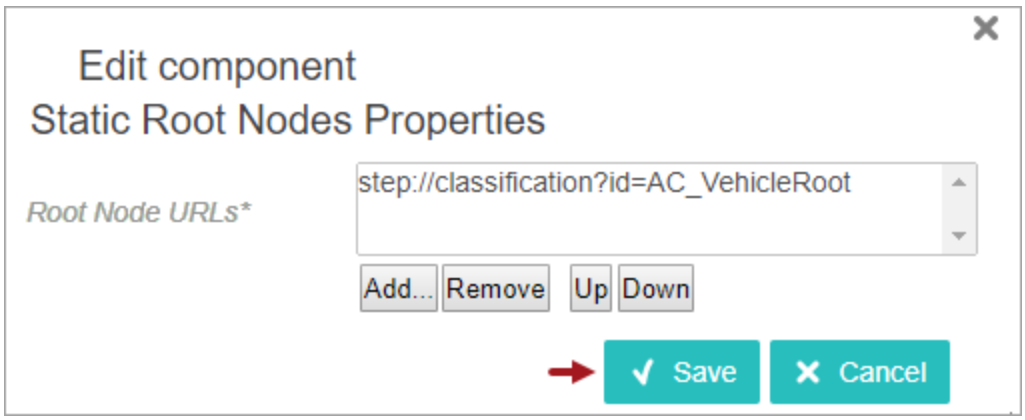
8. Within the Label parameter, enter text that will prompt the user as to what they should select when using this business action. For example, when the Label parameter is blank the parameter within the dialog will display as 'null' as shown in the image on the left. If the Label parameter is populated with 'Base vehicle' the parameter within the dialog will display with 'Base vehicle' as shown within the image on the right.



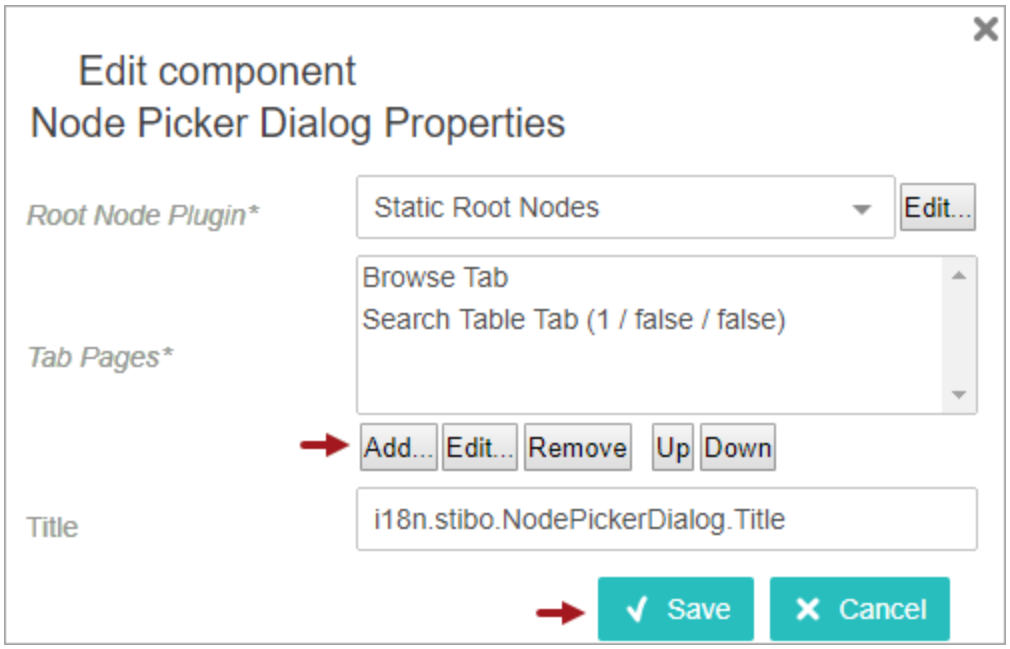

9. Use the Node Picker Configuration parameter dropdown to select the **Node Picker Dialog** option, and the 'Edit component' for the 'Node Picker Dialog Properties' dialog will display.




10. 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.
11. 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.



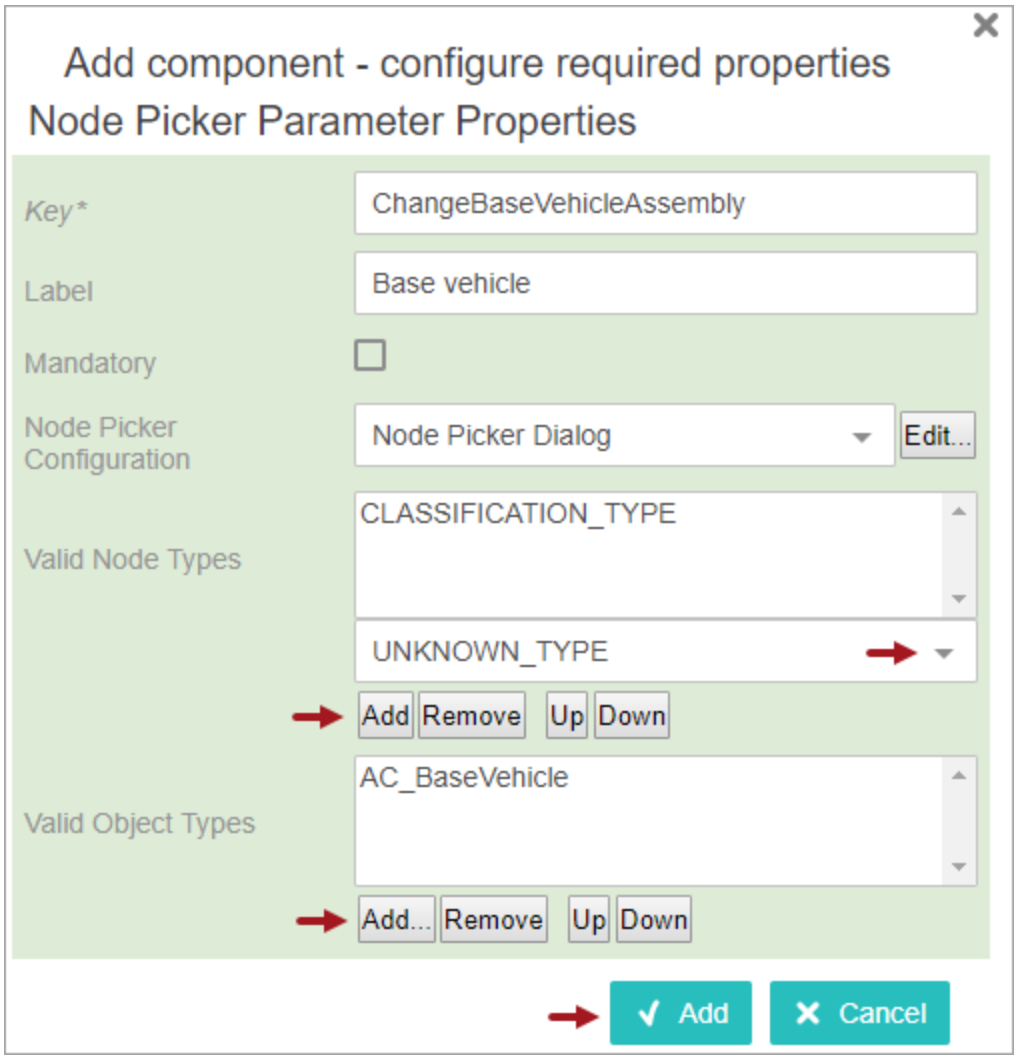

12. Click the **Save** button, and the 'Edit component' for the 'Node Picker Dialog Properties' dialog will display with the Root Node Plugin parameter populated with Static Root Nodes.
13. Click the **Add** button beneath the Tab Pages parameter > select the **Browse Tab** component > click the **Add** 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.



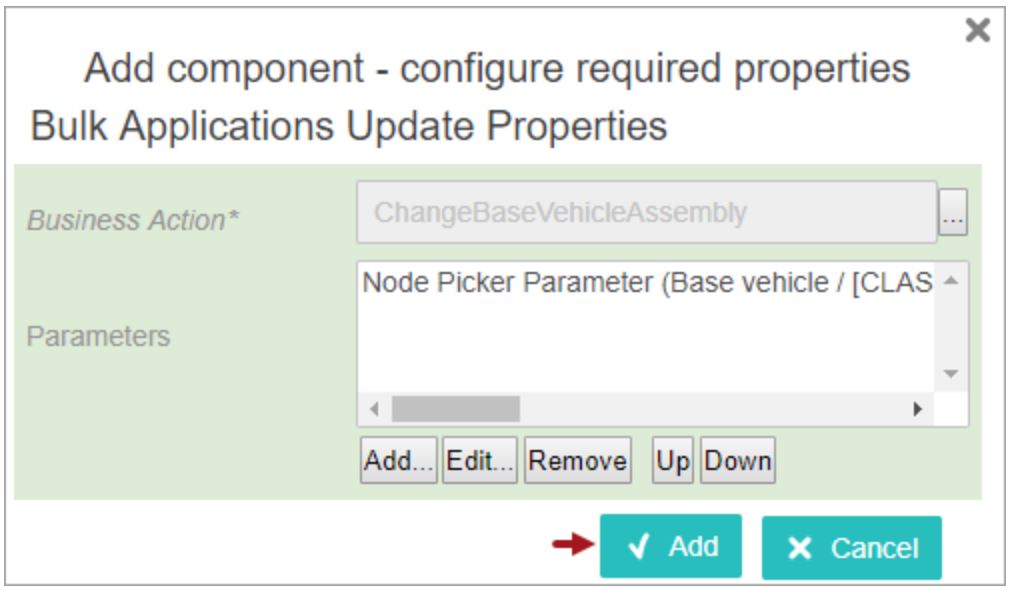

**Note:** Leave the Title parameter blank and *after the configuration is saved* an i18n key will be populated (as shown above). For more information, refer to the **Localization** topic within the **Administration Portal** section of the **STEP Online Help**. Otherwise, text entered within the Title 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.

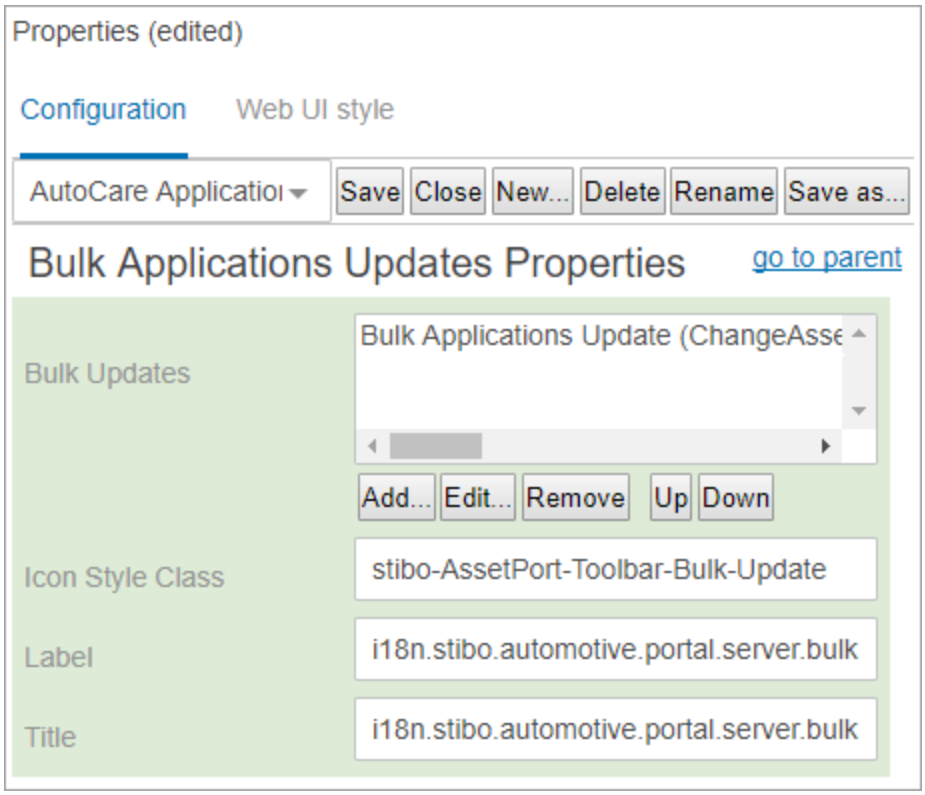
15. Use the dropdown located beneath the Valid Node Types parameter to select **CLASSIFICATION_TYPE** > click the **Add** button beneath the Valid Node Types parameter so that CLASSIFICATION_TYPE is displayed within the Valid Node Types parameter (as shown below).
16. Click the **Add** button beneath 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.




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.



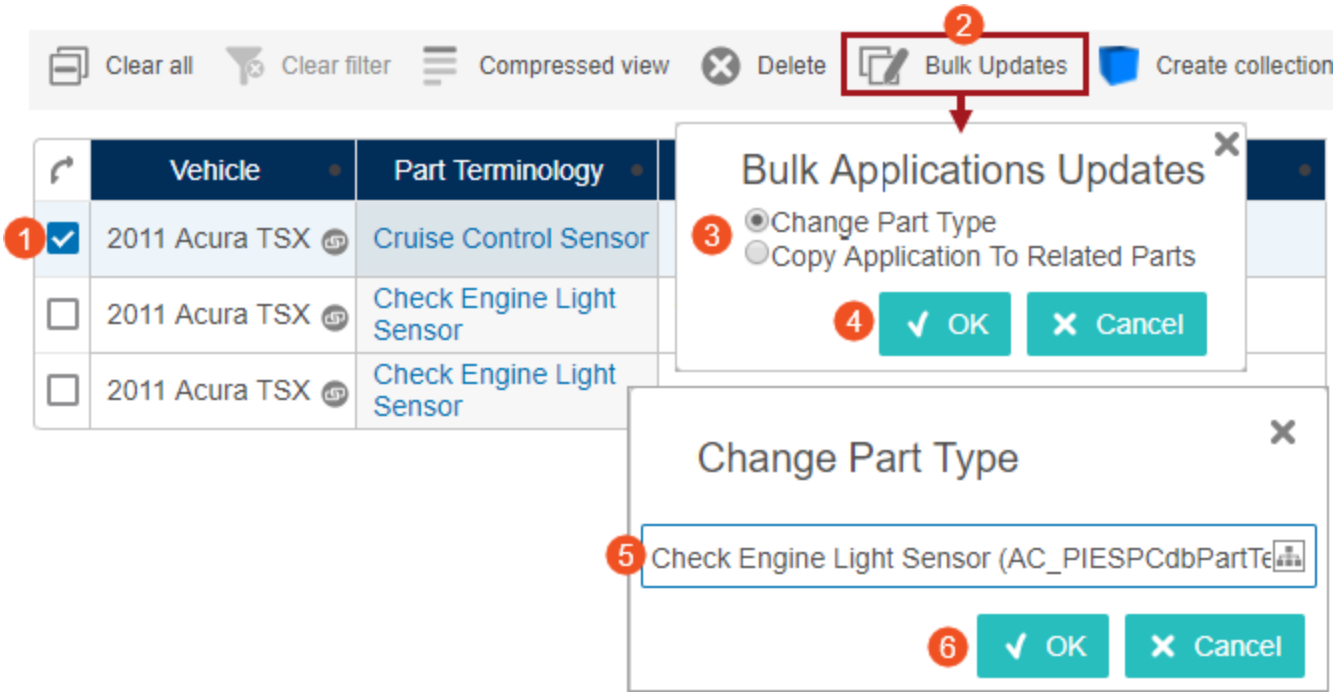

19. Click the **Save** button and then click the **Close** button to close the designer.

To use the newly configured business action, refer to 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, refer to 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.

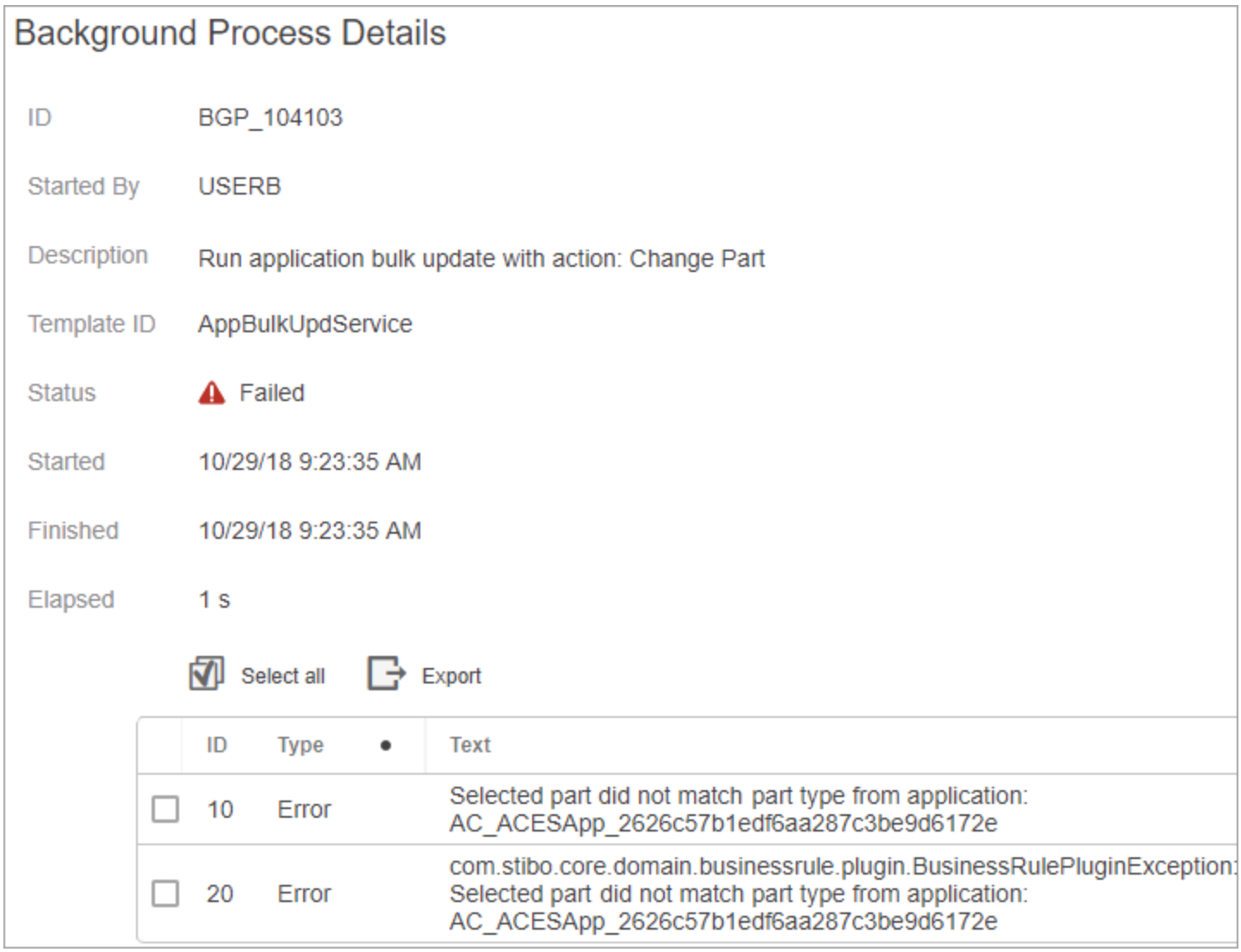
1. Within the configured Application Manager, search for applications and select one or more applications to be changed.
2. Click the **Bulk Updates** action button. If more than one bulk update is configured, the Bulk Applications Updates dialog will display (as shown above), otherwise this dialog is 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 open the 'Recent background processes' side panel by clicking on the gray BGP icon, , in the upper right corner of Web UI. For more information, refer to the **Recent Background Processes Side Panel** 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, refer to 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].



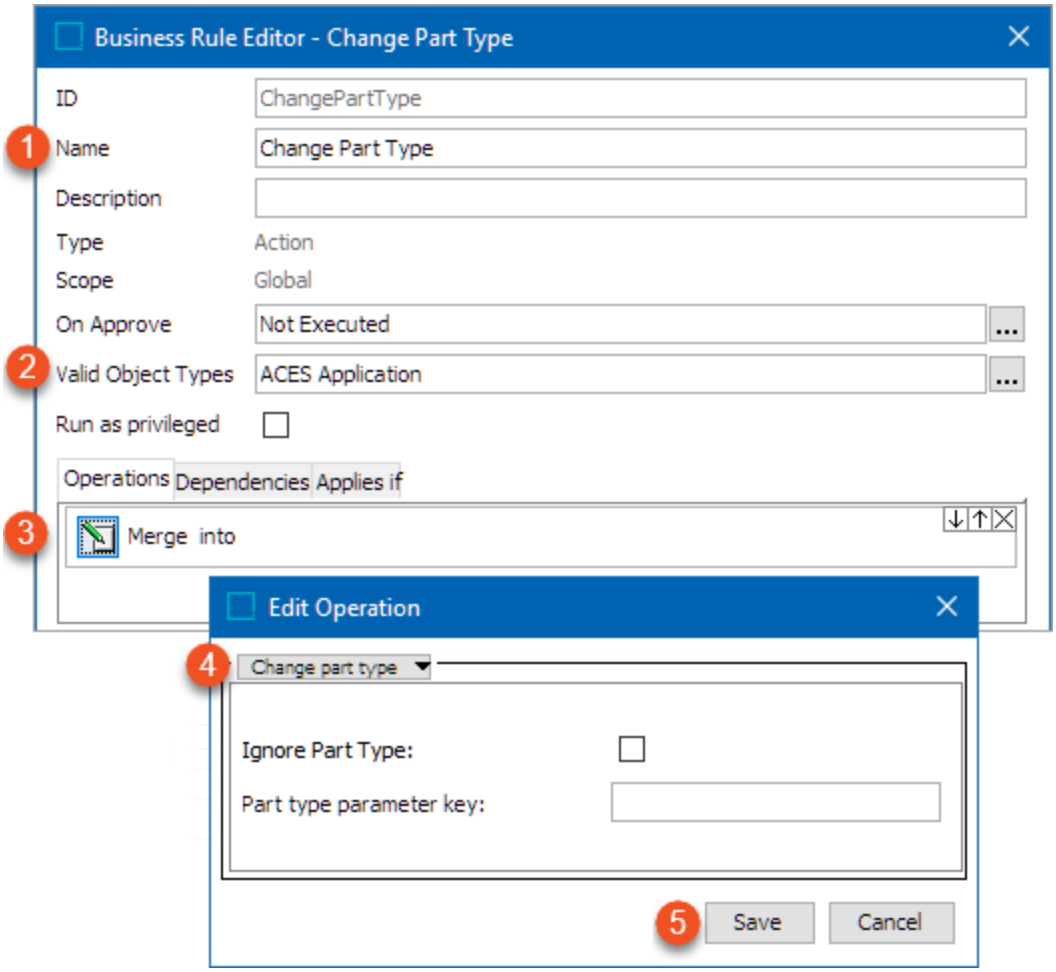

8. Once the background process has completed, click the **Find applications** button to display the newly created application(s) within the Application Manager results table.

Setup is required within both STEP Workbench and Web UI for the action to be available to users. For more information, refer to 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.'

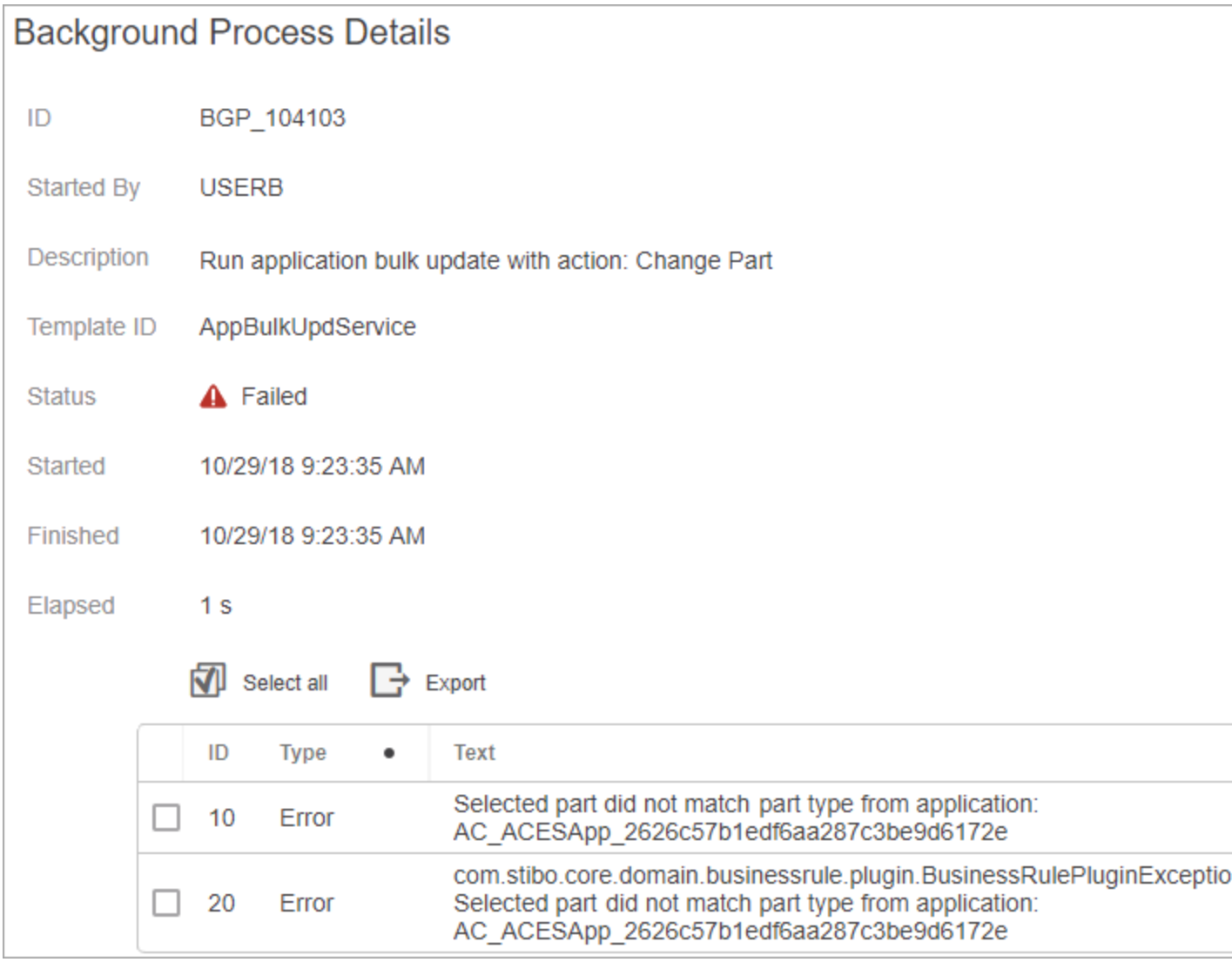
   **Note:** The business action Name will display to the Application Manager user once the Bulk Updates action button is selected. If more than one bulk update is configured, then the name displays both within the Bulk Applications Update dialog, and within the change part type dialog where the user enters 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.'



- **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.

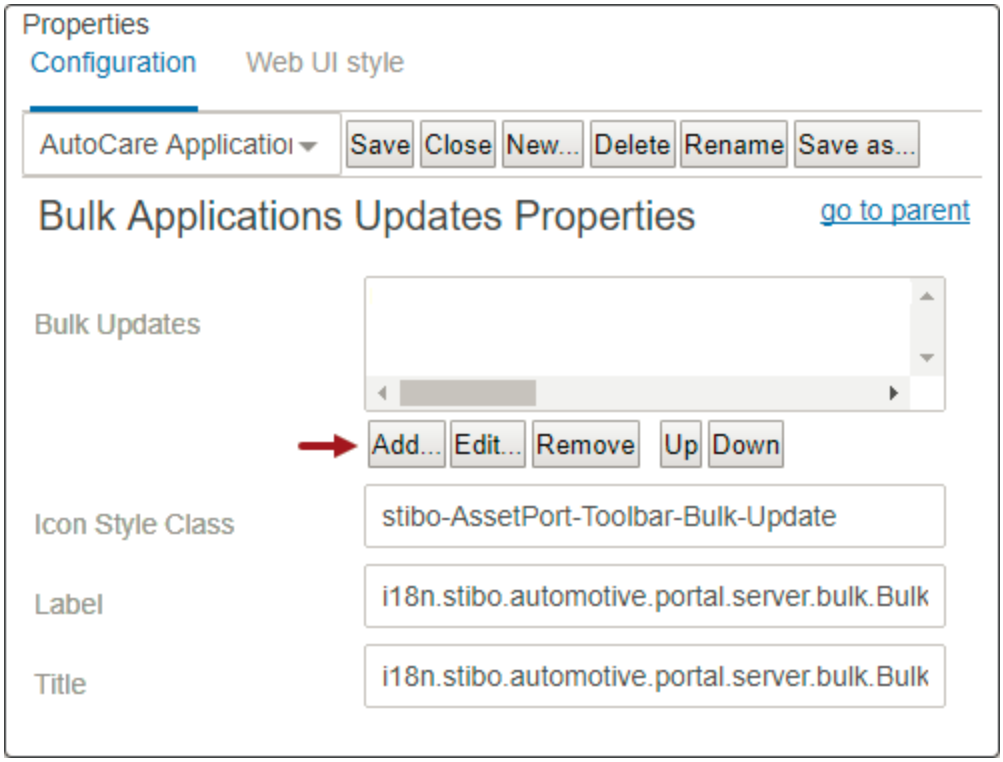
5. Click the **Save** button and continue to the next topic, **Configuring the Change Part Type Business Action in Web UI**.

### Configuring the Change Part Type Business Action in Web UI

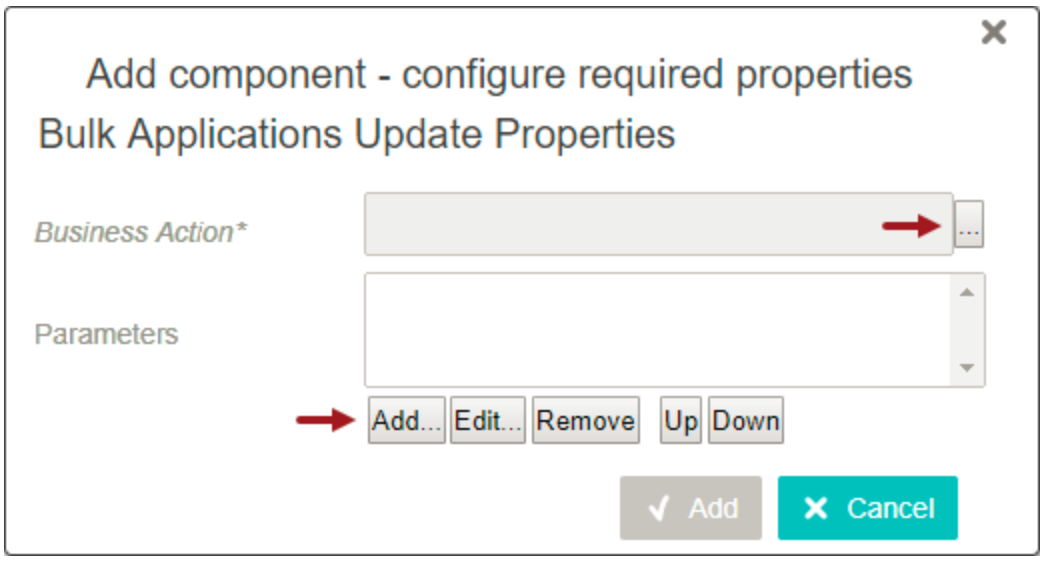
Once the business action has been configured in the workbench, configuration within Web UI is necessary. This section addresses the steps necessary within Web UI.

**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... .'

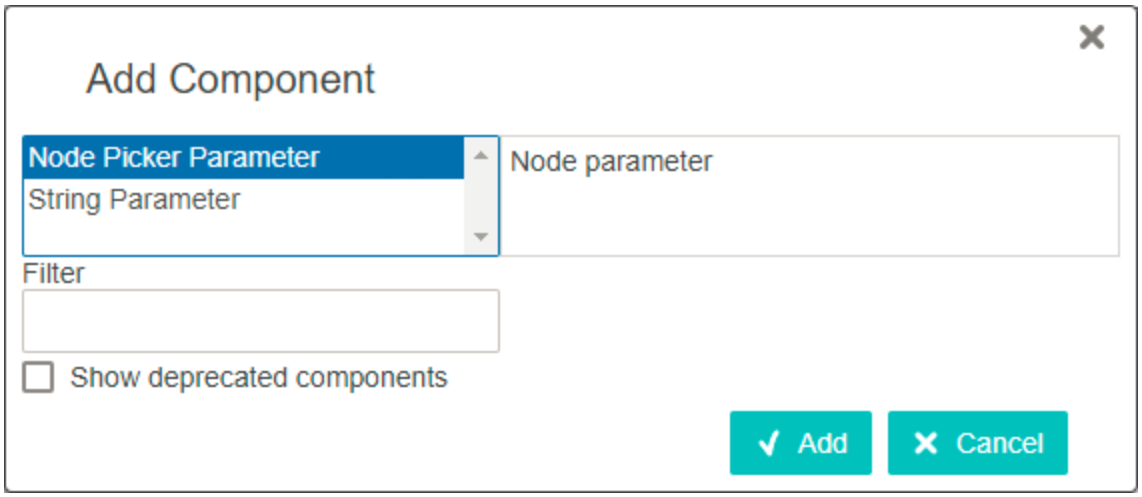
1. Access the Application Manager screen where the business action needs to be available to users.
2. Access the Designer > navigate to Node List Properties > Child Components > Actions > Double click **Bulk Applications Updates** and the 'Bulk Applications Updates Properties' dialog will display (as shown below).



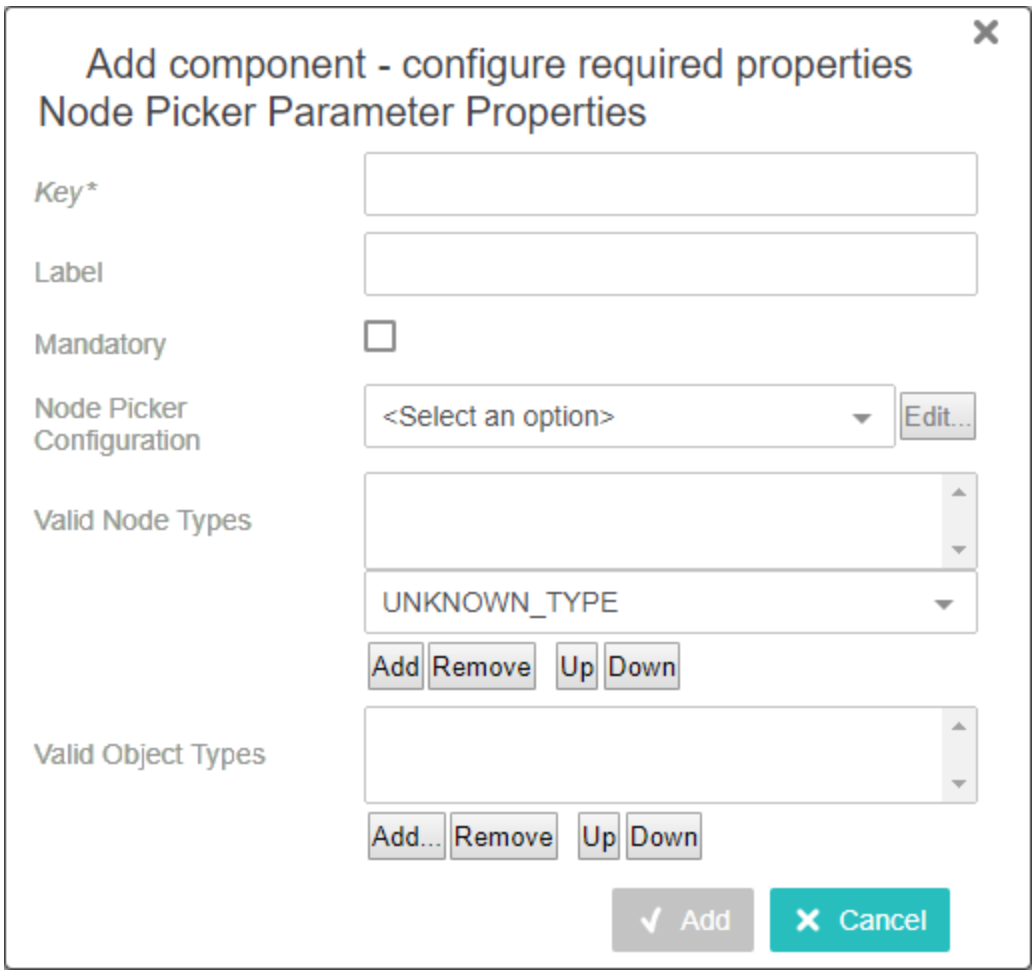

3. Click the **Add** button beneath the Bulk Updates parameter, and the 'Add component - configure required properties' for the 'Bulk Applications Update Properties' dialog will display (as shown below).




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).



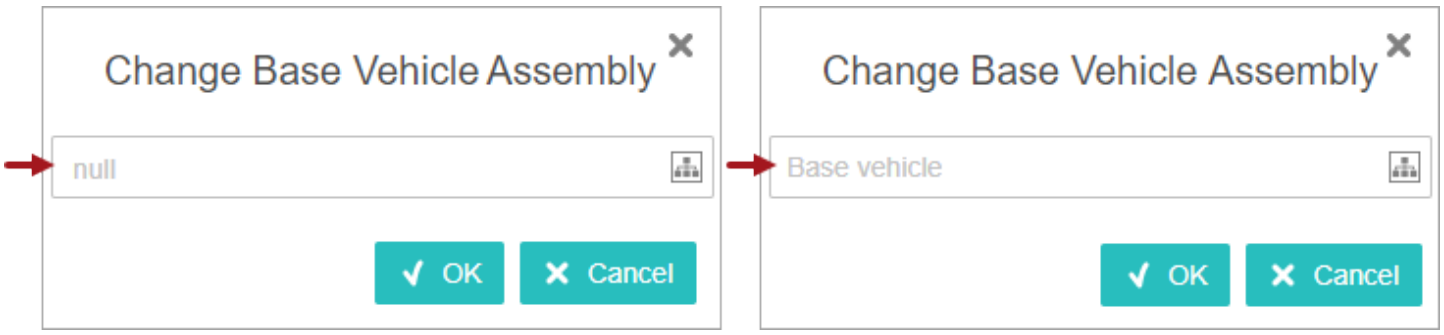

6. Select **Node Picker Parameter** > click the **Add** button to close the dialog, and the 'Add component - configure required properties' for the 'Node Picker Parameter Properties' dialog will display (as shown below).



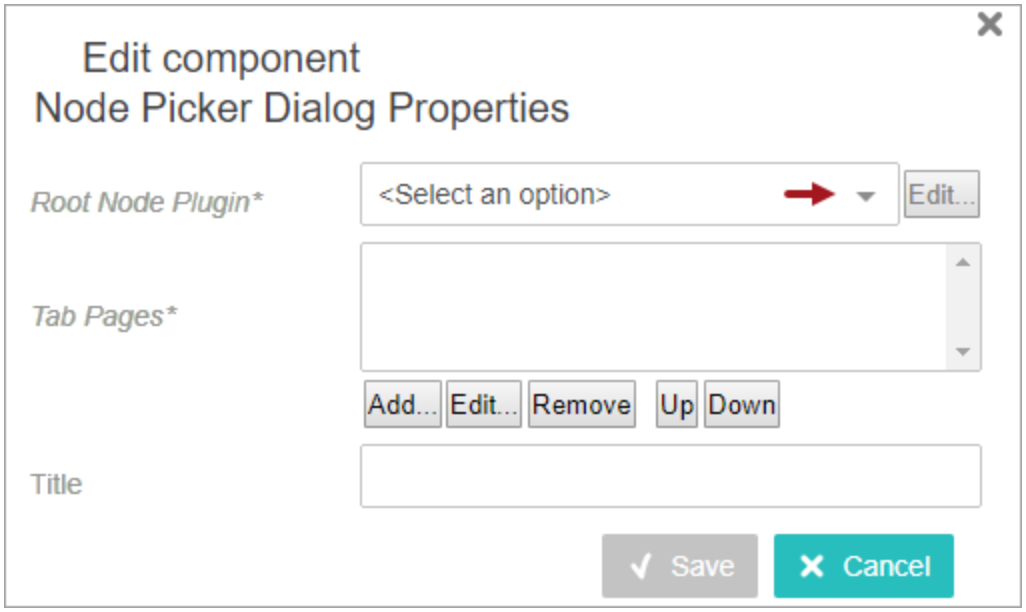

7. Within the Key parameter, enter the exact key created within the 'Part type parameter key' parameter during step 5 of the **Configuring the Change Part Type Business Action in Workbench** topic. This is the only required parameter.

   **Important:** If this key is not entered exactly in both places, then the business action will not properly function within the Web UI.

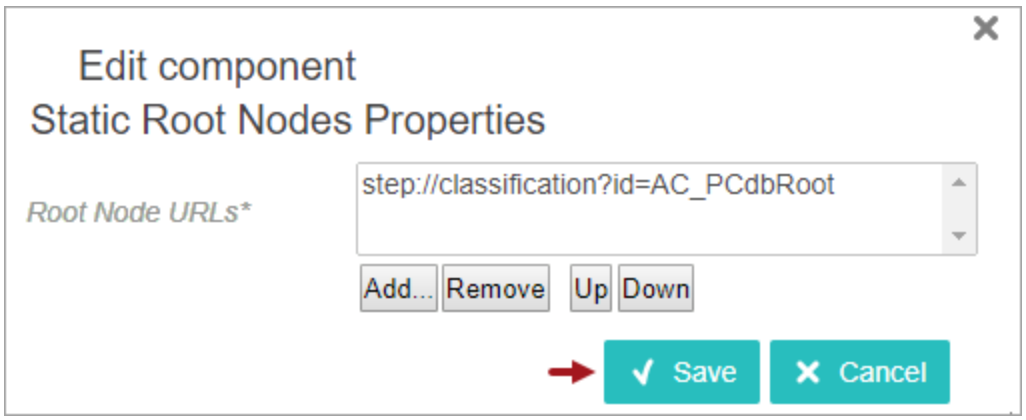
8. Within the Label parameter, enter text that will prompt the user as to what they should select when using this business action. For example, when the Label parameter is blank the parameter within the dialog will display as 'null' as shown in the image on the left. If the Label parameter is populated with 'Base vehicle' the parameter within the dialog will display with 'Base vehicle' as shown within the image on the right.



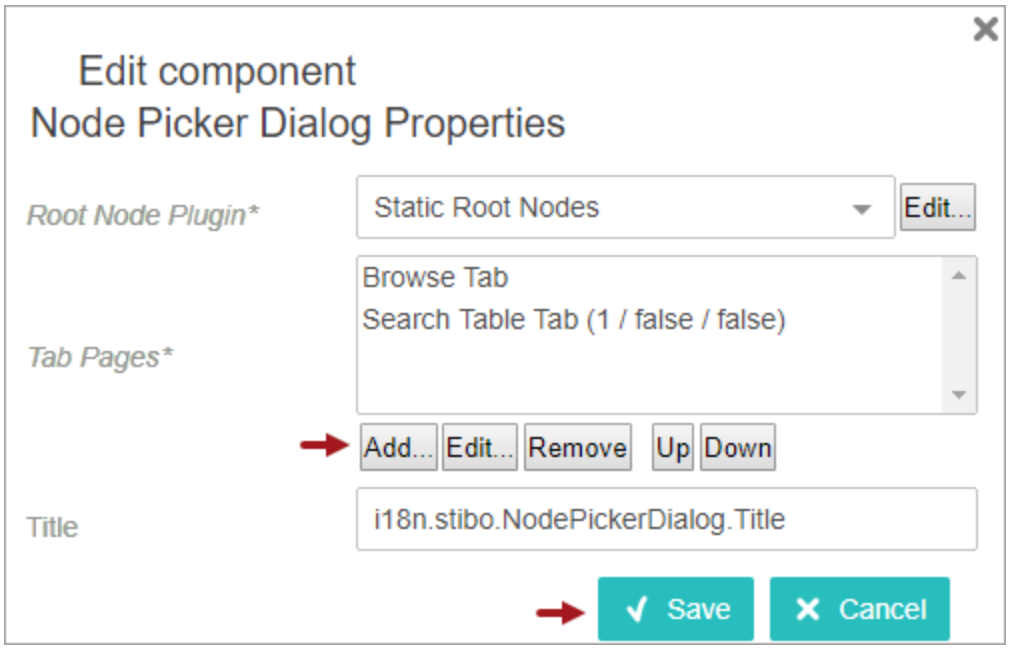

9. Use the Node Picker Configuration parameter dropdown to select the **Node Picker Dialog** option, and the 'Edit component' for the 'Node Picker Dialog Properties' dialog will display.




10. 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.
11. 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.



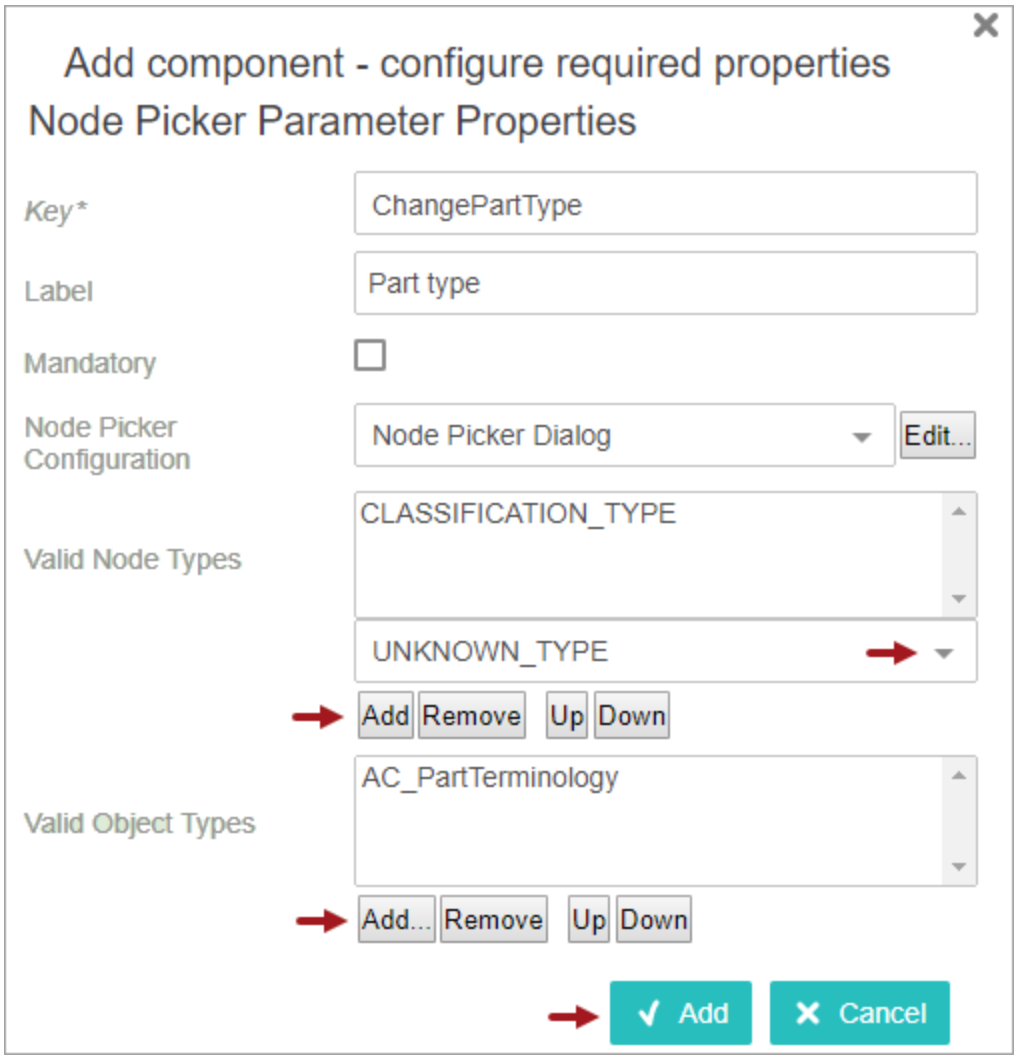

12. Click the **Save** button, and the 'Edit component' for the 'Node Picker Dialog Properties' dialog will display with the Root Node Plugin parameter populated with Static Root Nodes.
13. Click the **Add** button beneath the Tab Pages parameter > select the **Browse Tab** component > click the **Add** 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.



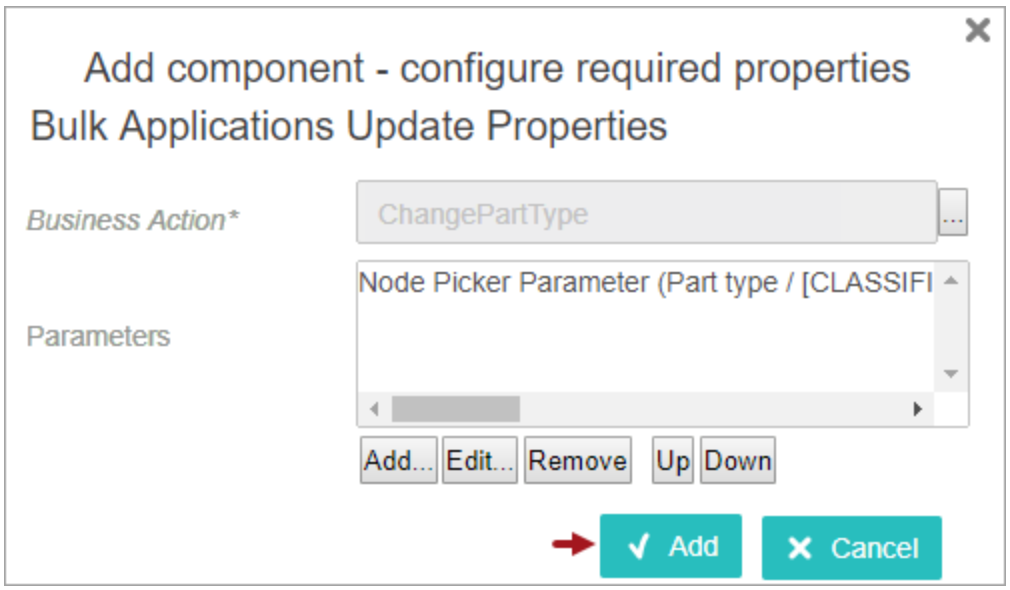

**Note:** Leave the Title parameter blank and *after the configuration is saved* an i18n key will be populated (as shown above). For more information, refer to the **Localization** topic within the **Administration Portal** section of the **STEP Online Help**. Otherwise, text entered within the Title 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.

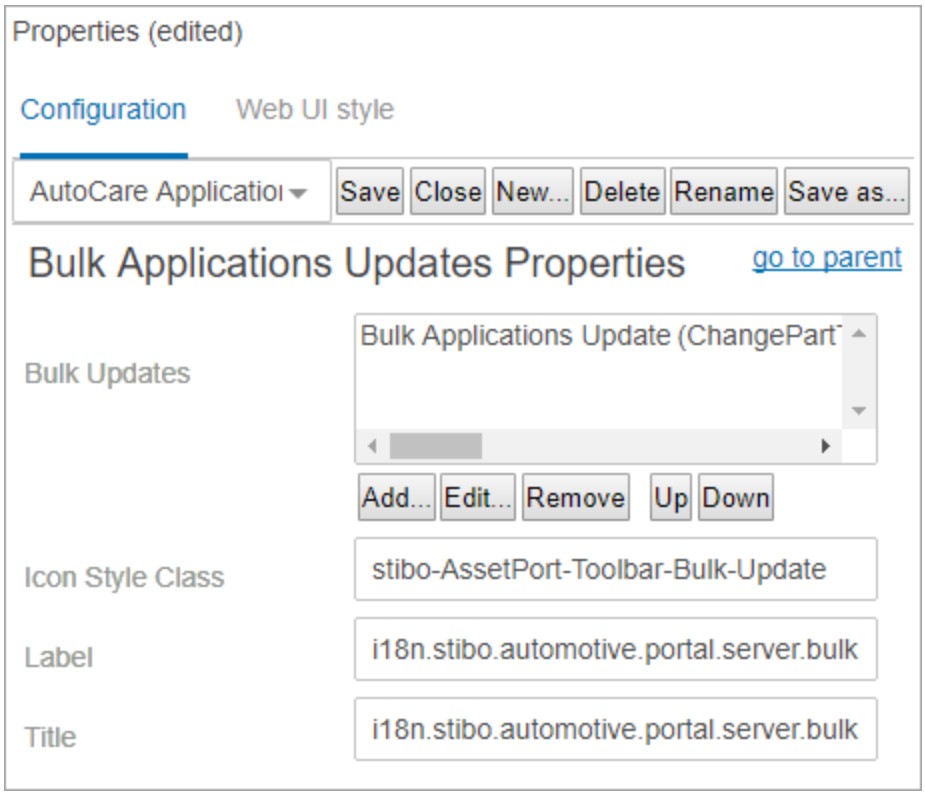
15. Use the dropdown located beneath the Valid Node Types parameter to select **CLASSIFICATION_TYPE** > click the **Add** button beneath the Valid Node Types parameter so that CLASSIFICATION_TYPE is displayed within the Valid Node Types parameter (as shown below).
16. Click the **Add** button beneath 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.



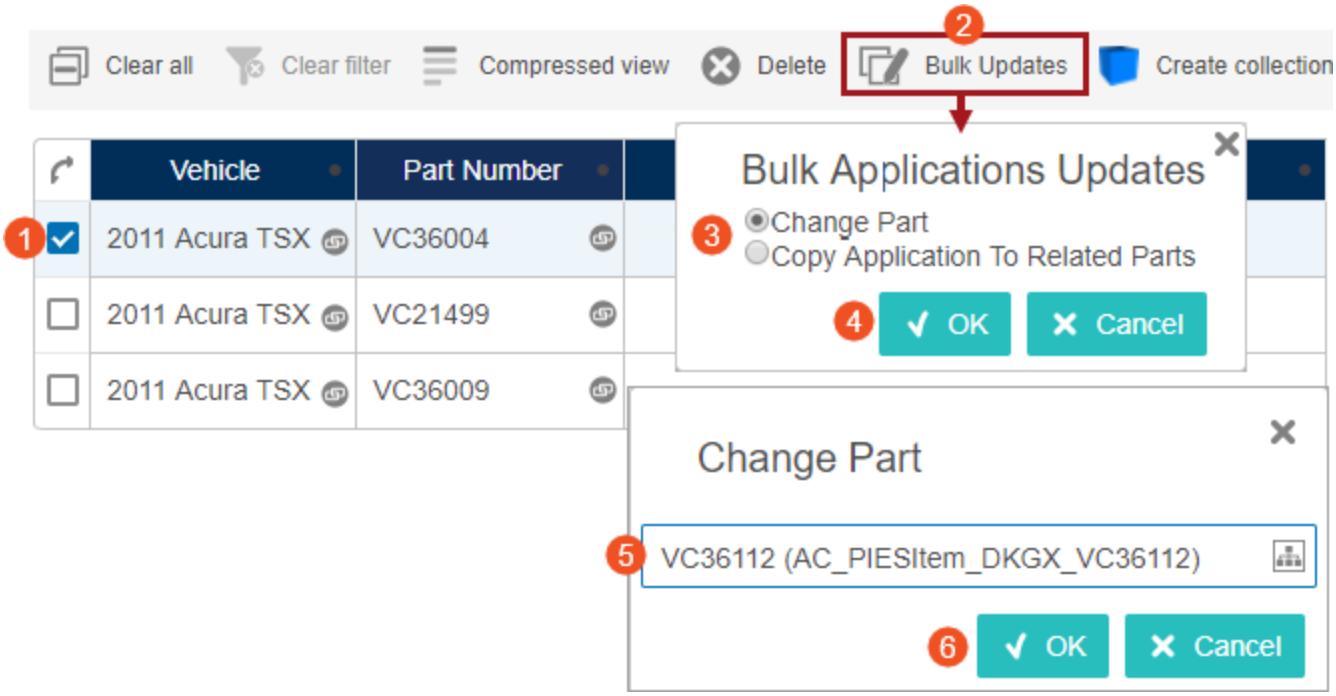

19. Click the **Save** button and then click the **Close** button to close the designer.

To use the newly configured business action, refer to 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, refer to 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.

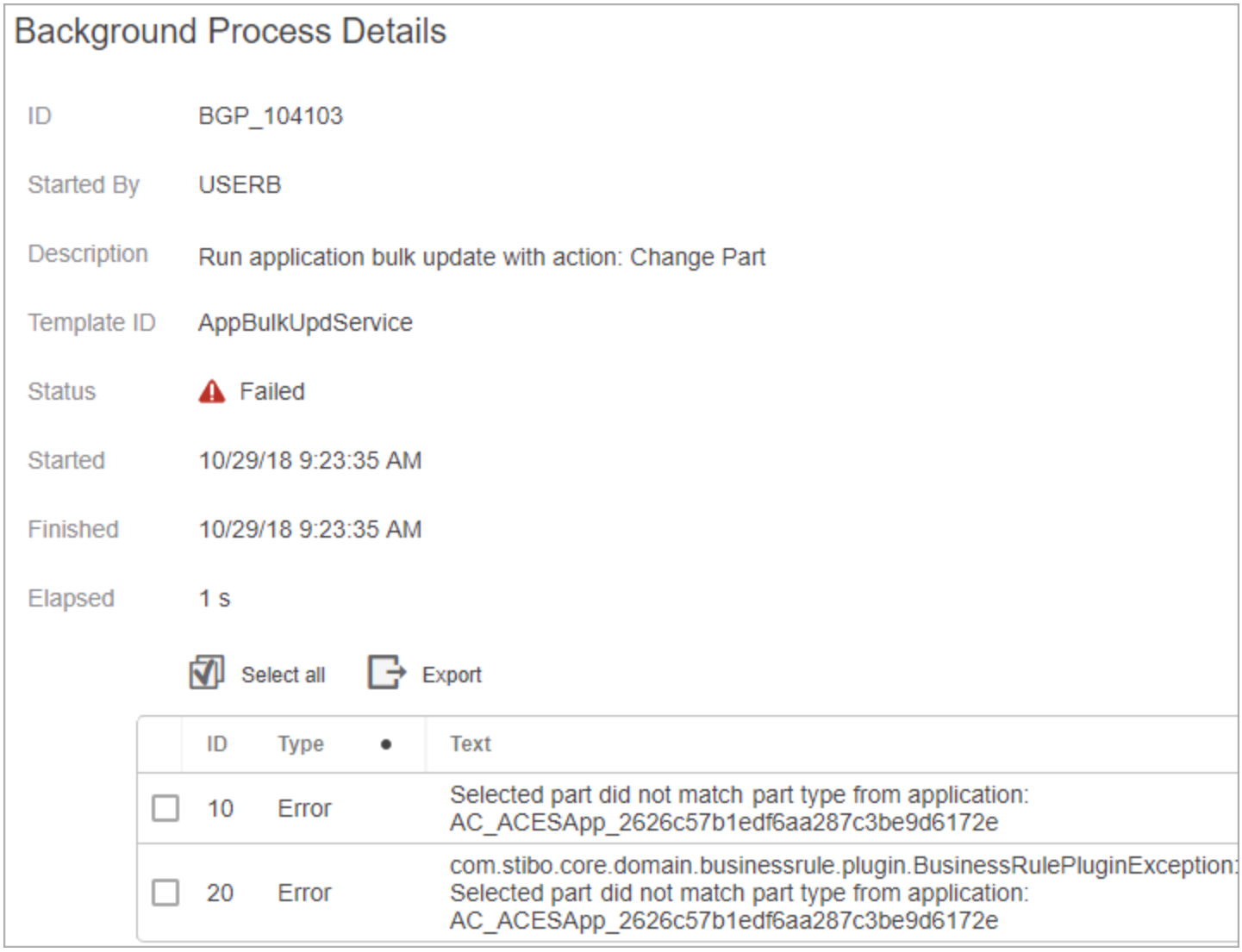
1. Within the configured Application Manager, search for applications and select one or more applications to be changed.
2. Click the **Bulk Updates** action button. If more than one bulk update is configured then the Bulk Updates dialog will display (as shown above), otherwise this dialog is skipped 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 open the 'Recent background processes' side panel by clicking on the gray BGP icon, , in the upper right corner of Web UI. For more information, refer to the **Recent Background Processes Side Panel** topic.

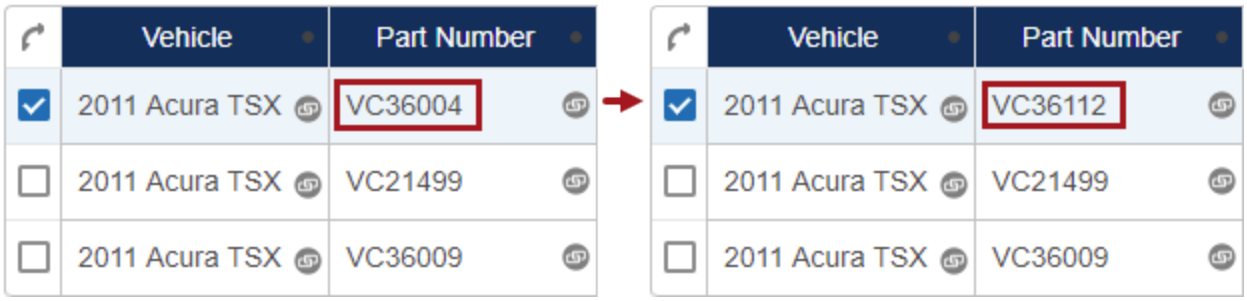
   If the business action BGP fails, access the Background Process Details page for more information.

**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, refer to 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].



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.

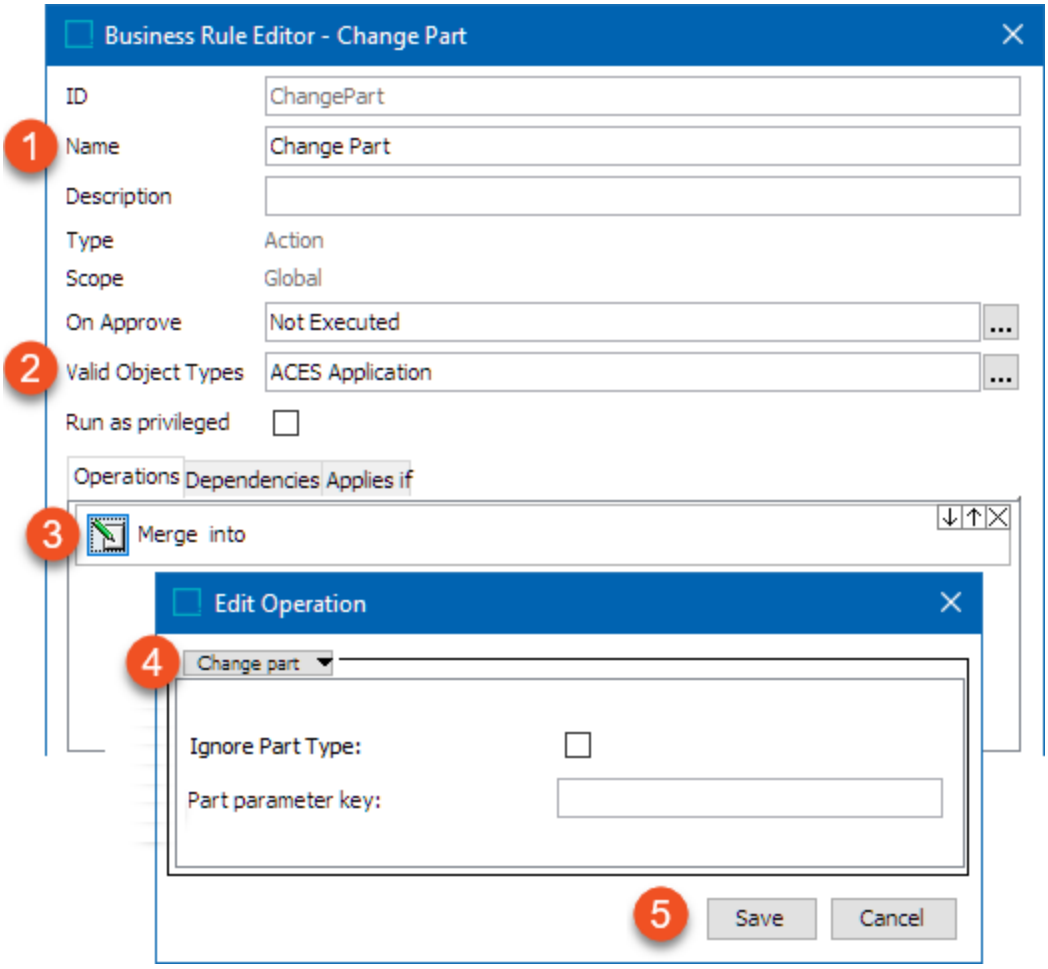



Setup is required within both STEP Workbench and Web UI for the action to be available to users. For more information, refer to the configuration topics below:

- 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.



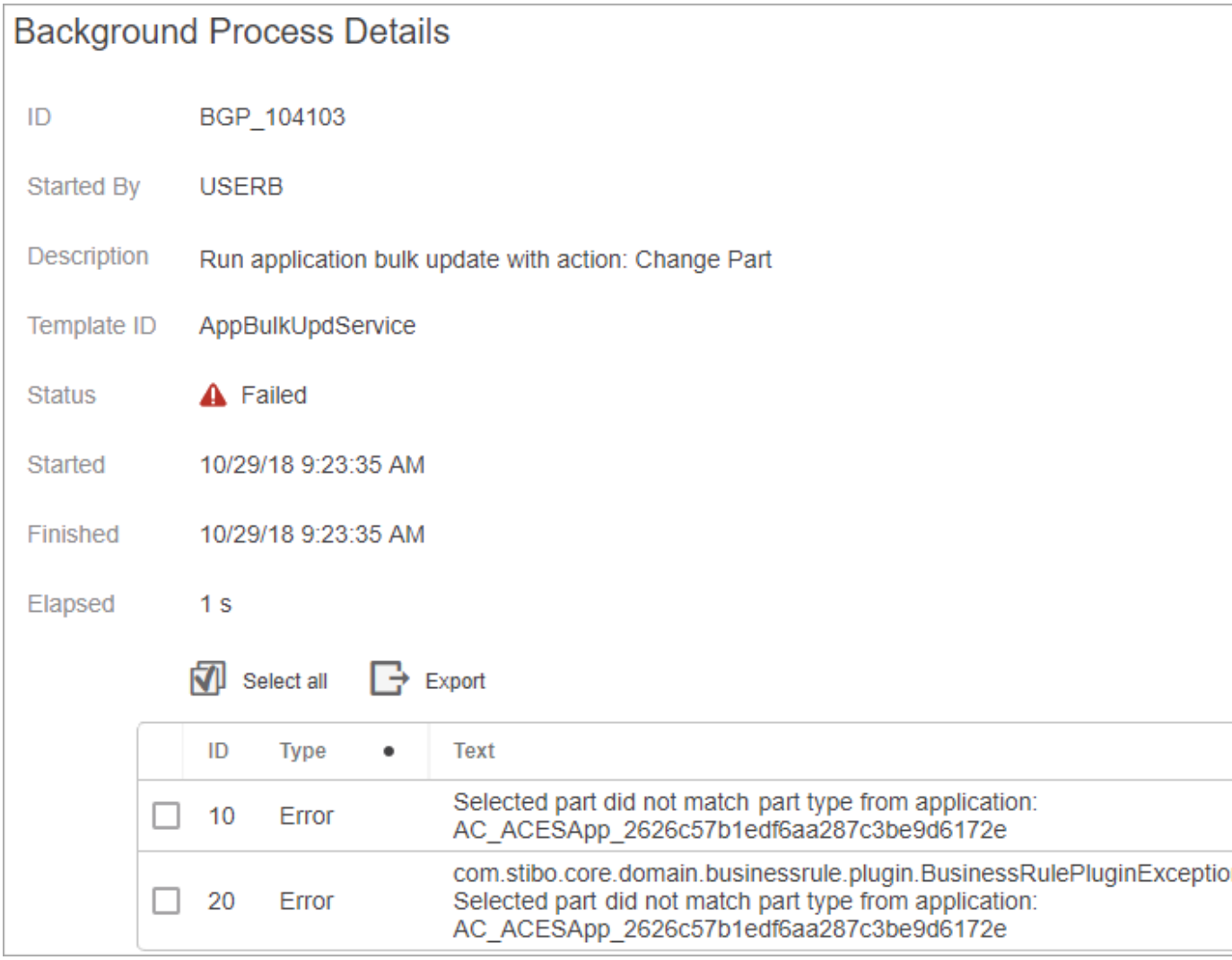

1. Create the business action with a name that accurately describes to the user what this action will do. The name of the business action displays within the Web UI and should be easy for the user to identify. For this example, the business action name is 'Change Part.'

   **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.




   - **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.

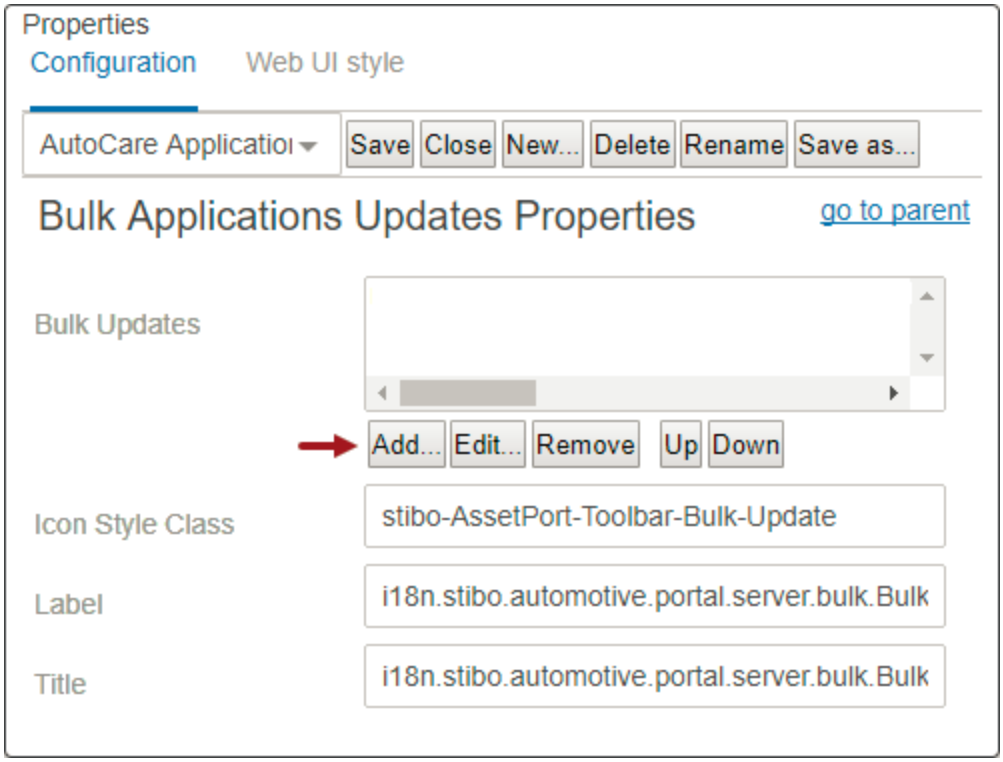
5. Click the **Save** button and continue to the next topic, **Configuring the Change Part Business Action in Web UI**.

### Configuring the Change Part Business Action in Web UI

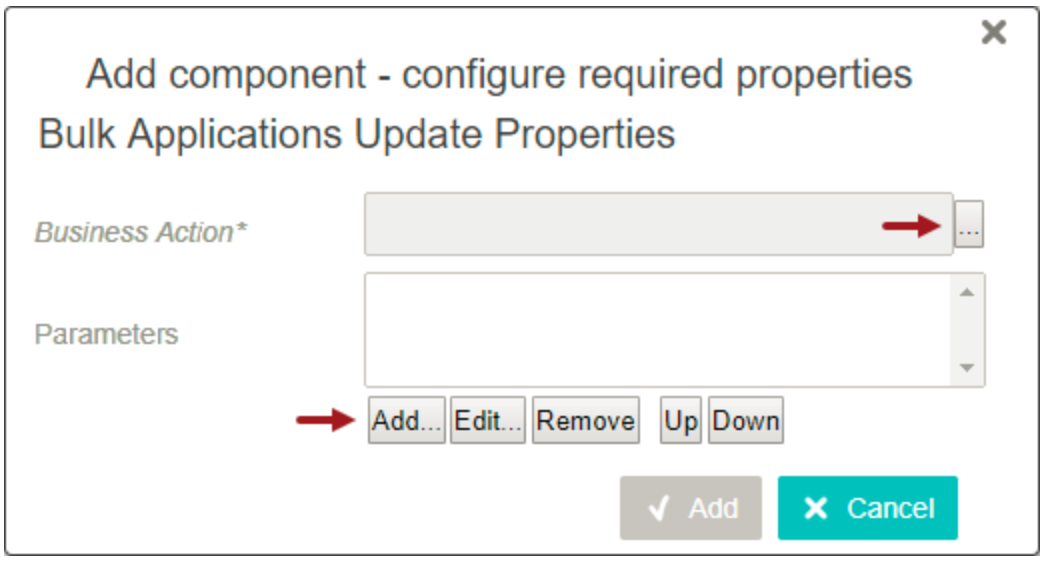
Once the business action has been configured in the workbench, configuration within Web UI is necessary. This section addresses the steps necessary within Web UI.

**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... .'

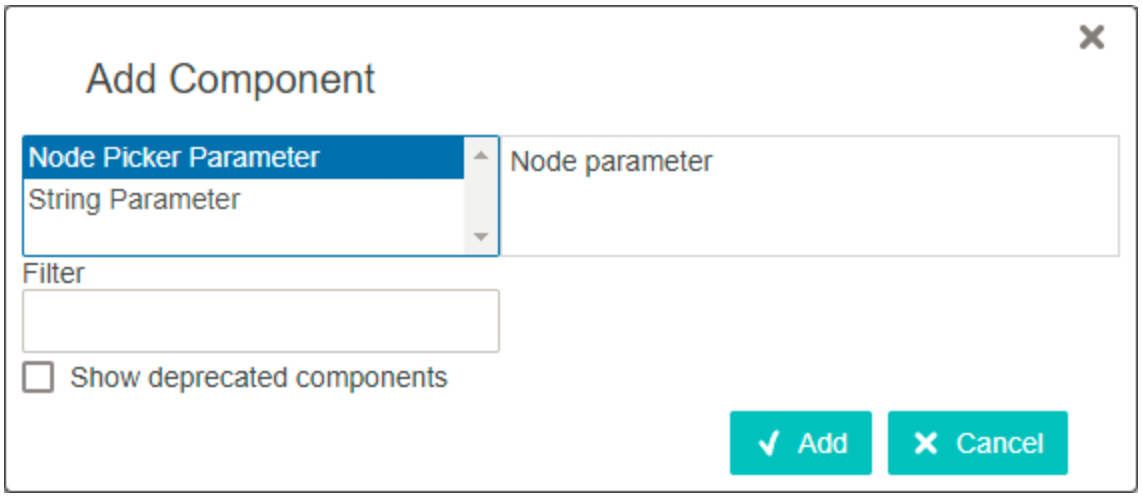
1. Access the Application Manager screen where the business action needs to be available to users.
2. Access the Designer > navigate to Node List Properties > Child Components > Actions > Double click **Bulk Applications Updates** and the 'Bulk Applications Updates Properties' dialog will display (as shown below).



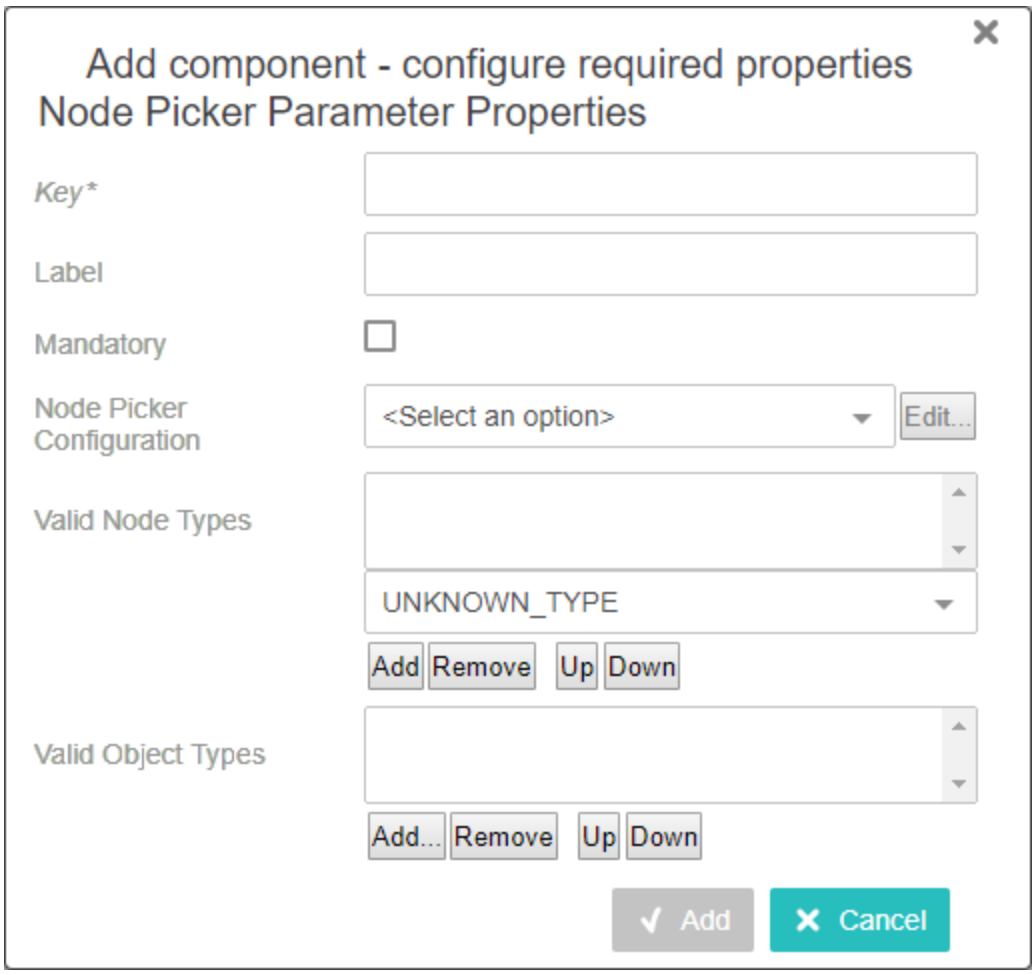

3. Click the **Add** button beneath the Bulk Updates parameter, and the 'Add component - configure required properties' for the 'Bulk Applications Update Properties' dialog will display (as shown below).




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).



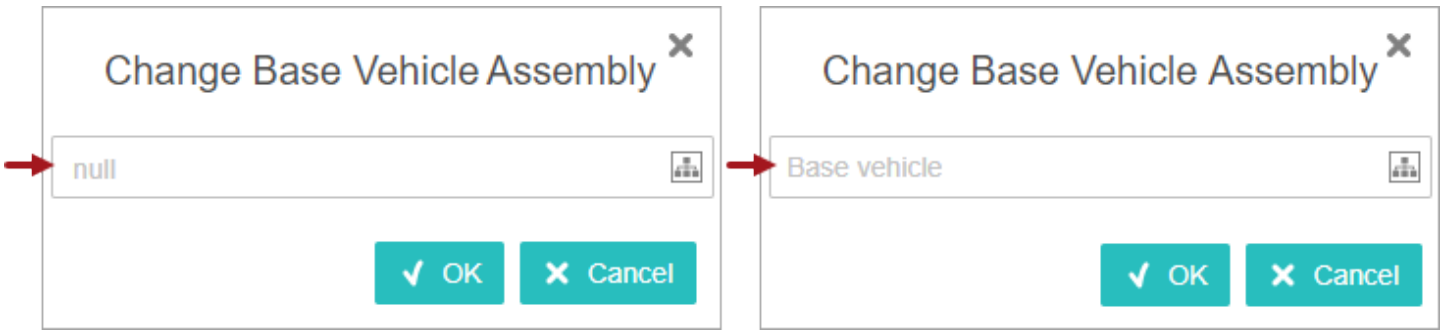

6. Select **Node Picker Parameter** > click the **Add** button to close the dialog, and the 'Add component - configure required properties' for the 'Node Picker Parameter Properties' dialog will display (as shown below).



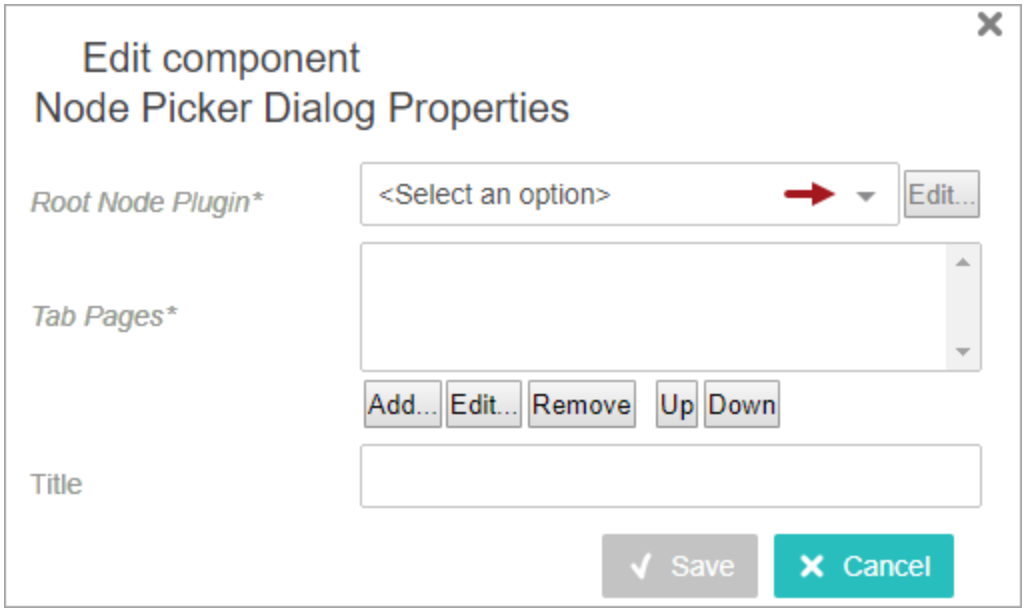

7. Within the Key parameter, enter the exact key created within the 'Part parameter key' parameter during step 5 of the **Configuring the Change Part Business Action in Workbench** topic. This is the only required parameter.

   **Important:** If this key is not entered exactly in both places, then the business action will not properly function within the Web UI.

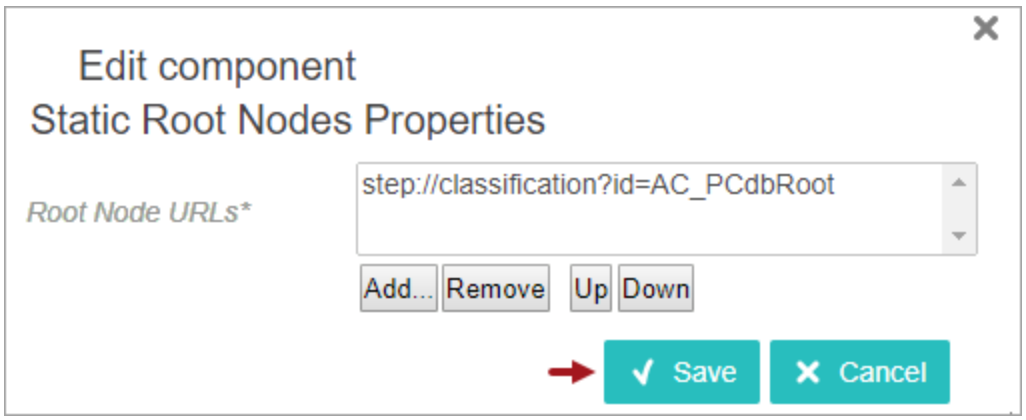
8. Within the Label parameter, enter text that will prompt the user as to what they should select when using this business action. For example, when the Label parameter is blank the parameter within the dialog will display as 'null' as shown in the image on the left. If the Label parameter is populated with 'Base vehicle' the parameter within the dialog will display with 'Base vehicle' as shown within the image on the right.



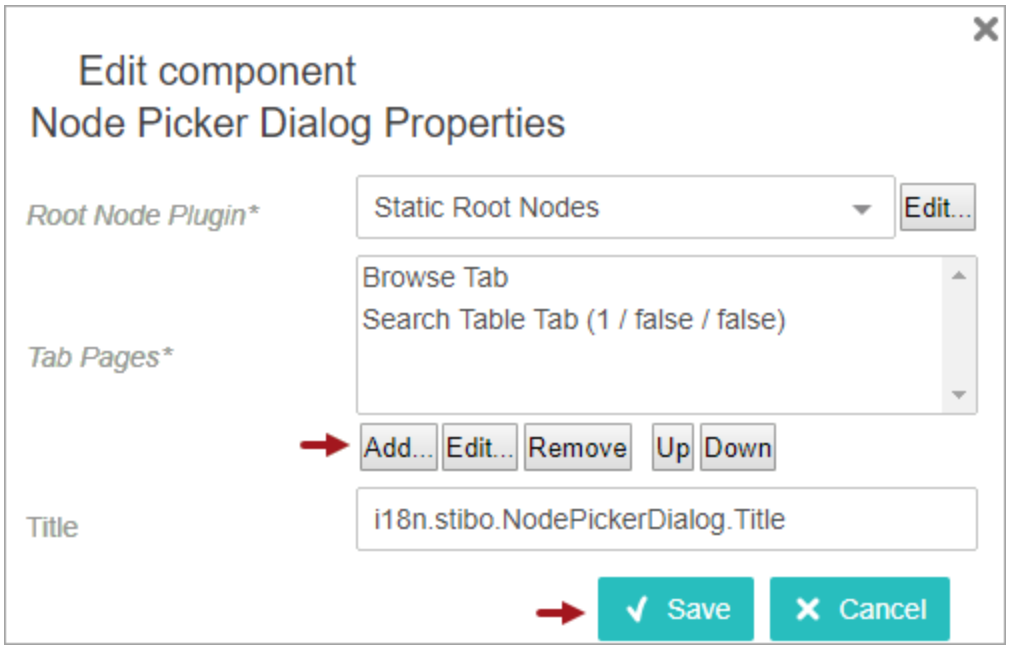

9. Use the Node Picker Configuration parameter dropdown to select the **Node Picker Dialog** option, and the 'Edit component' for the 'Node Picker Dialog Properties' dialog will display.




10. 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.
11. 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.



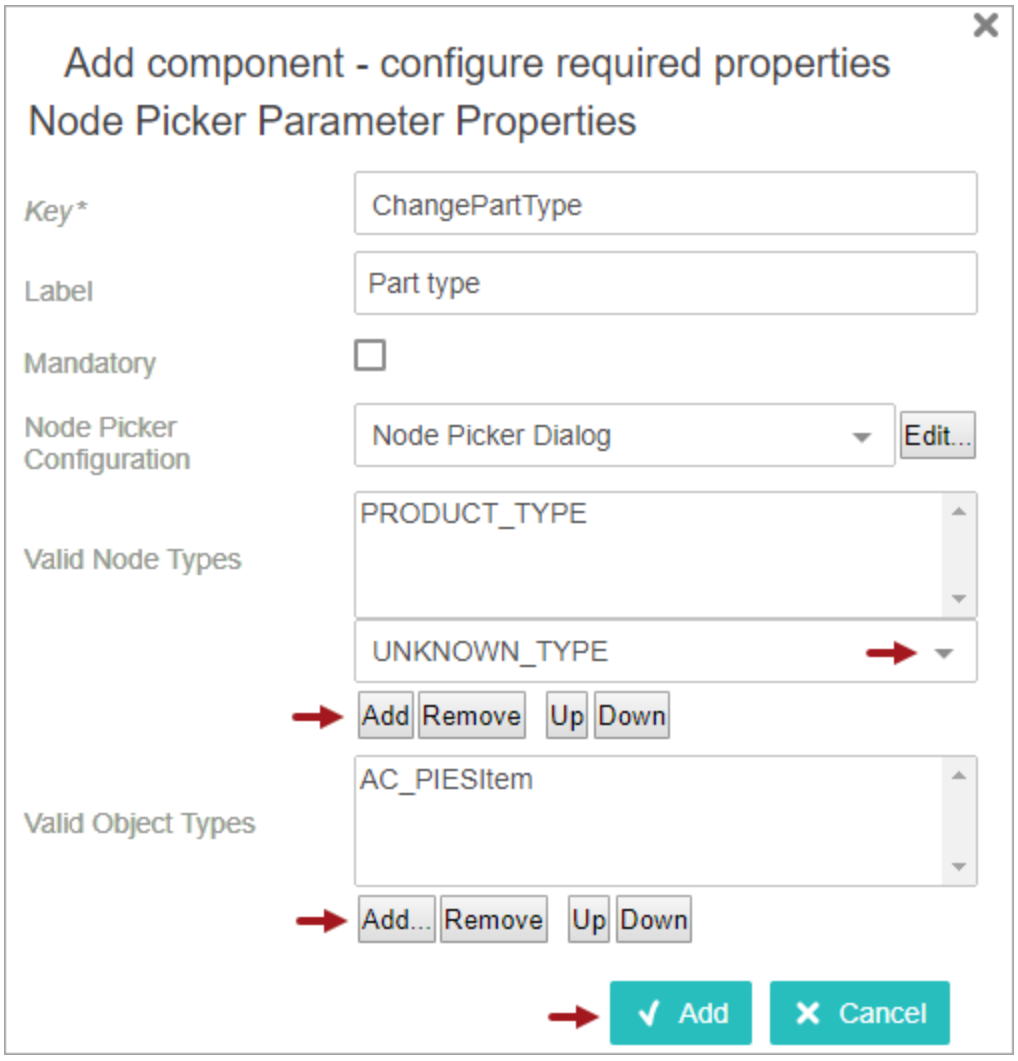

12. Click the **Save** button, and the 'Edit component' for the 'Node Picker Dialog Properties' dialog will display with the Root Node Plugin parameter populated with Static Root Nodes.
13. Click the **Add** button beneath the Tab Pages parameter > select the **Browse Tab** component > click the **Add** 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.



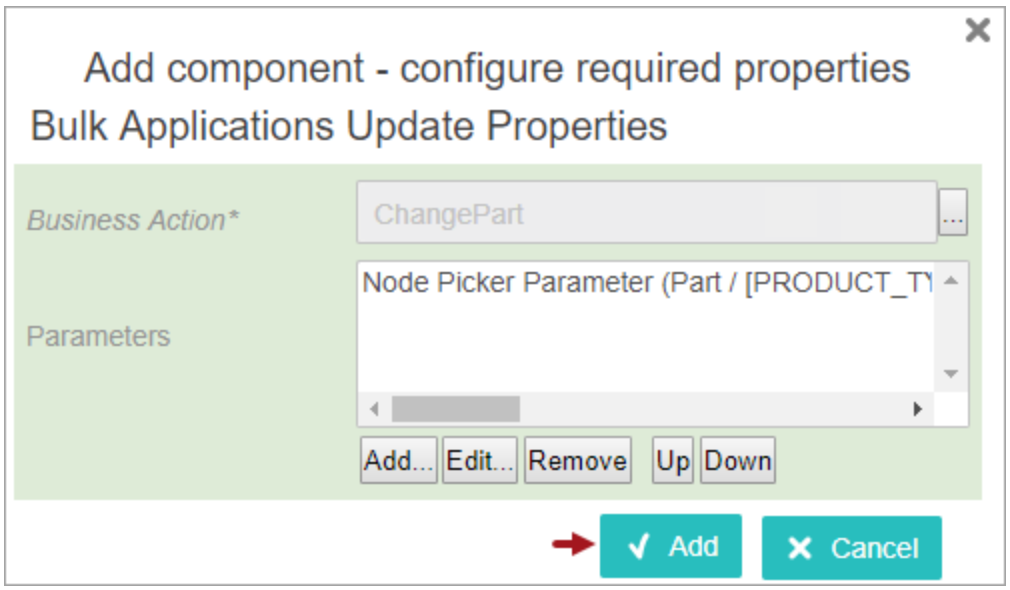

**Note:** Leave the Title parameter blank and *after the configuration is saved* an i18n key will be populated (as shown above). For more information, refer to the **Localization** topic within the **Administration Portal** section of the **STEP Online Help**. Otherwise, text entered within the Title 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.

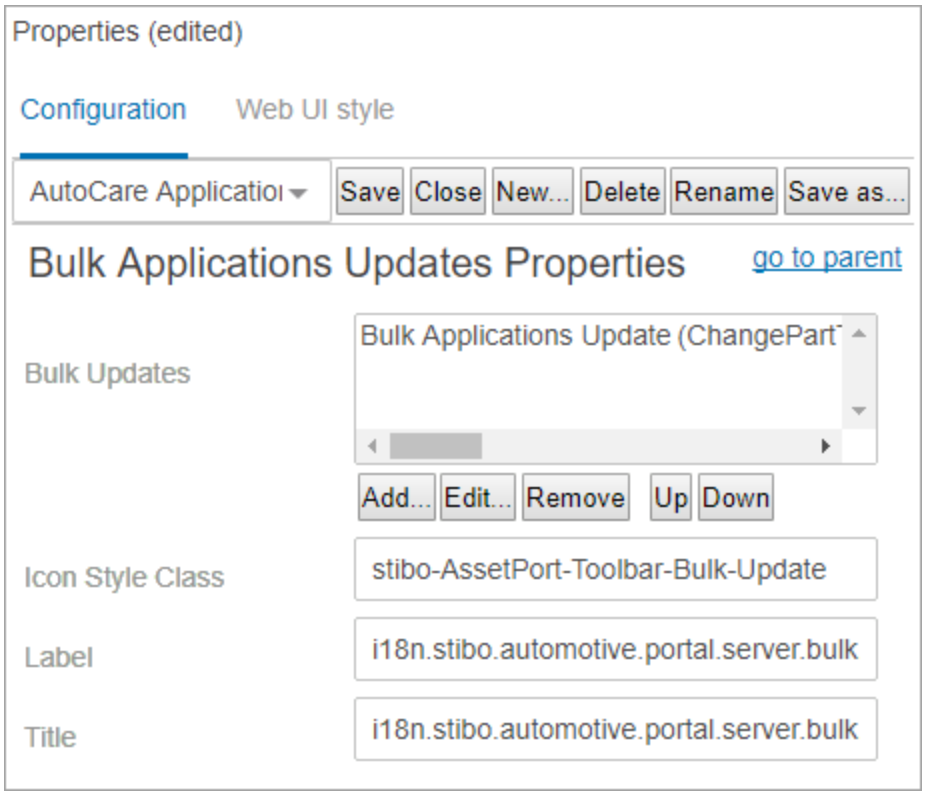
15. Use the dropdown located beneath the Valid Node Types parameter to select **PRODUCT_TYPE** > click the **Add** button beneath the Valid Node Types parameter so that PRODUCT_TYPE is displayed within the Valid Node Types parameter (as shown below).
16. Click the **Add** button beneath the Valid Object 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.




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.



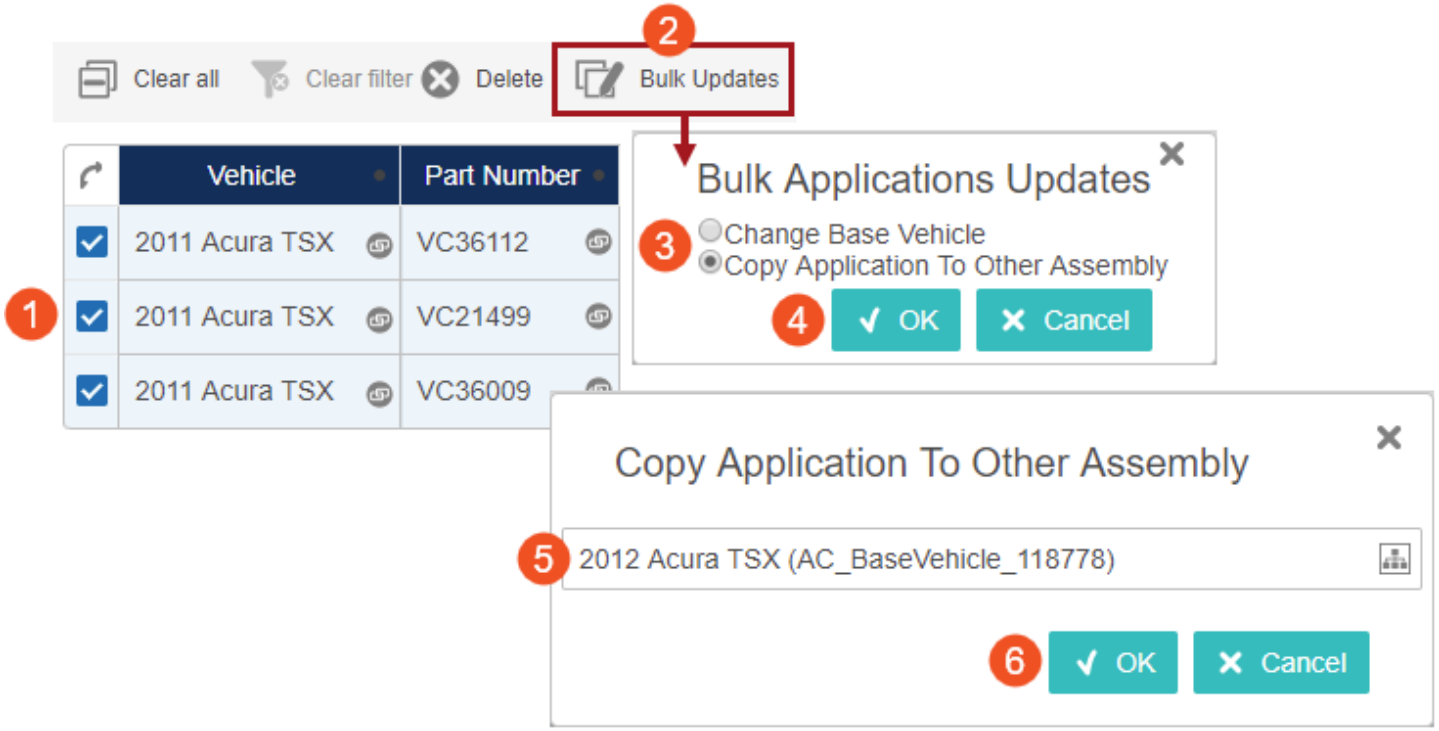

19. Click the **Save** button and then click the **Close** button to close the designer.

To use the newly configured business action, refer to 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, refer to 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.

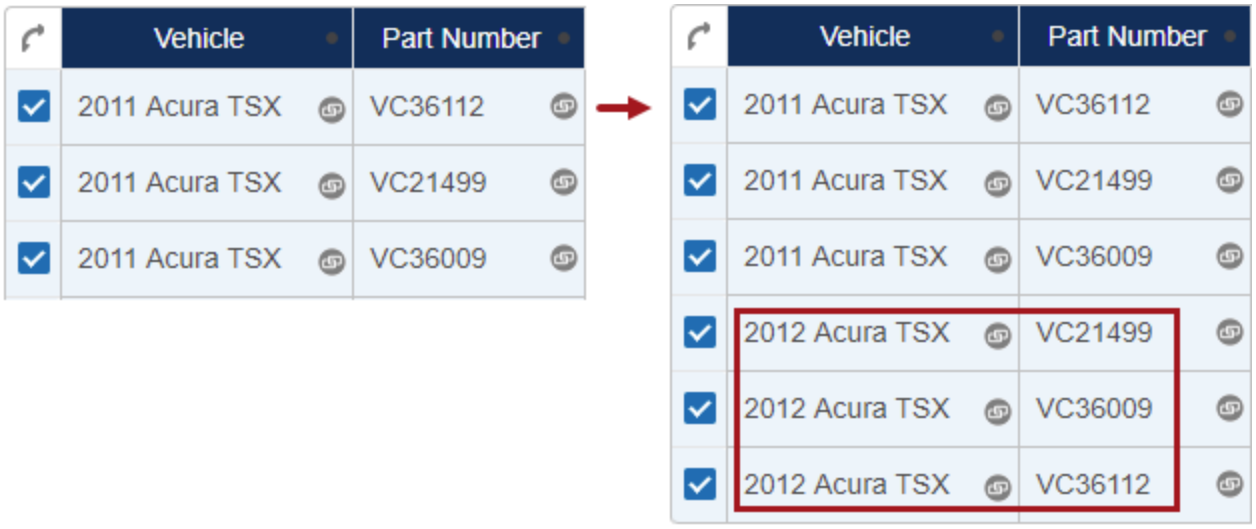
Once configured, a Bulk Updates button within an Application Manager can be used to copy one or more existing applications to another assembly / vehicle. Below are the steps for using the business action in Web UI.




**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.

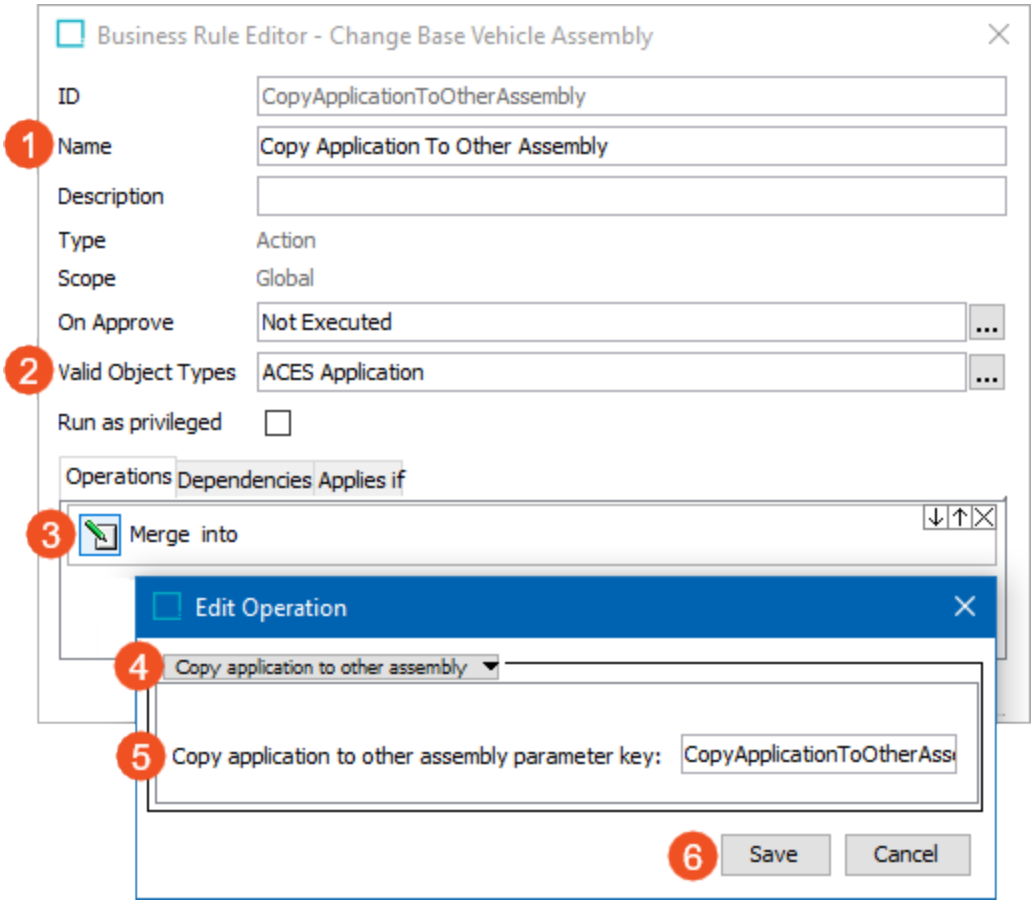



Setup is required within both STEP Workbench and Web UI for the action to be available to users. For more information, refer to the configuration topics below:

- 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.

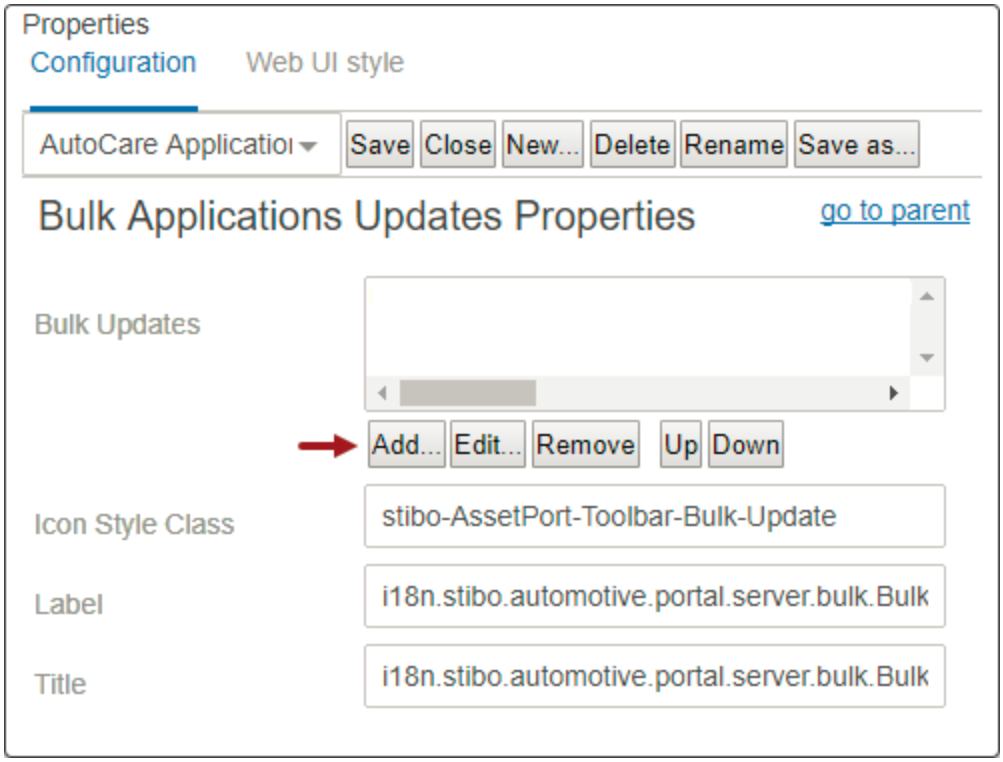
6. Click the **Save** button and continue to the next topic, **Copy Application to Other Assembly Business Action in Web UI**.

### Configuring the Copy Application to Other Assembly Business Action in Web UI

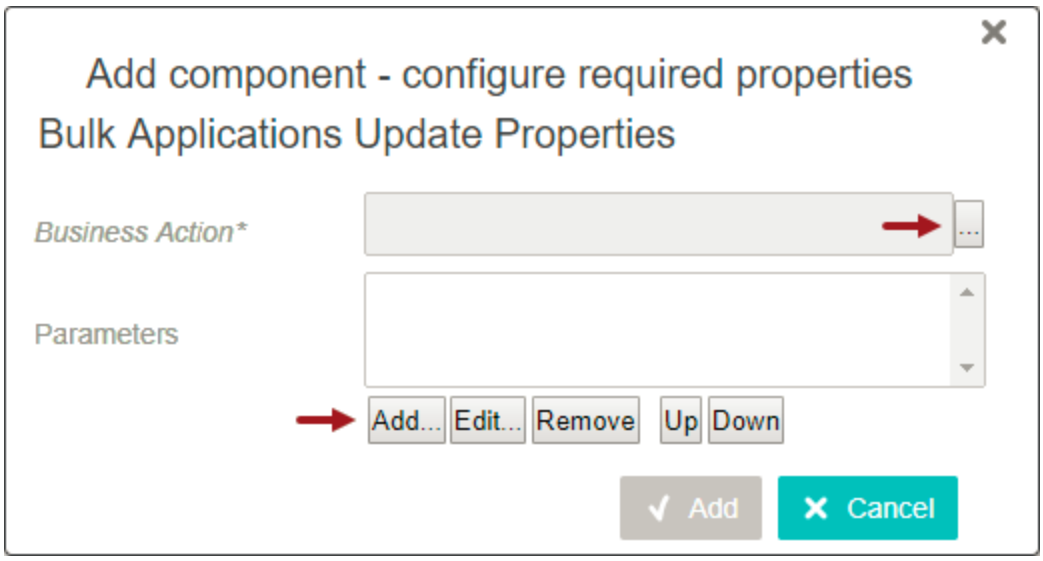
Once the business action has been configured in the workbench, configuration within Web UI is necessary. This section addresses the steps necessary within Web UI.

**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... .'

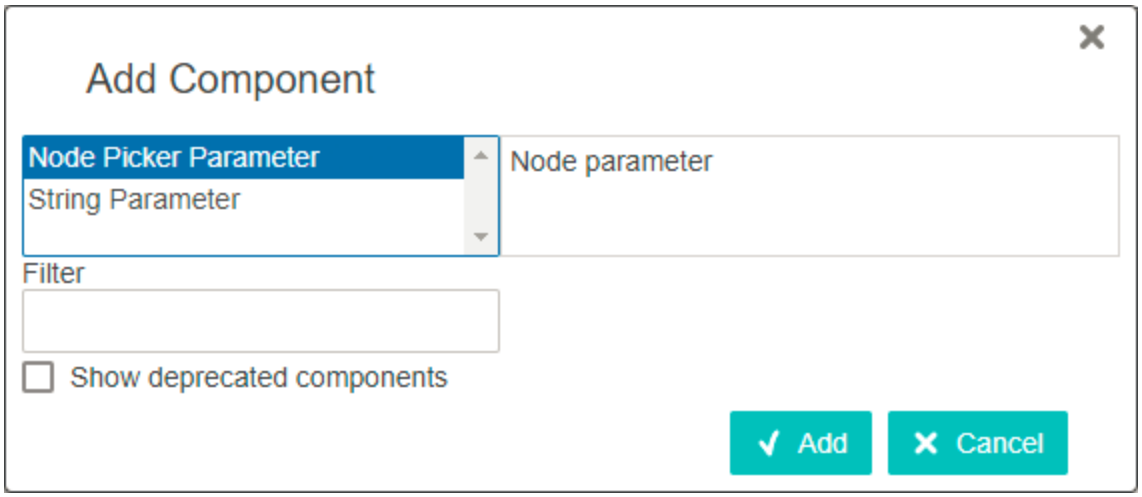
1. Access the Application Manager screen where the business action needs to be available to users.
2. Access the Designer > navigate to Node List Properties > Child Components > Actions > Double click **Bulk Applications Updates** and the 'Bulk Applications Updates Properties' dialog will display (as shown below).



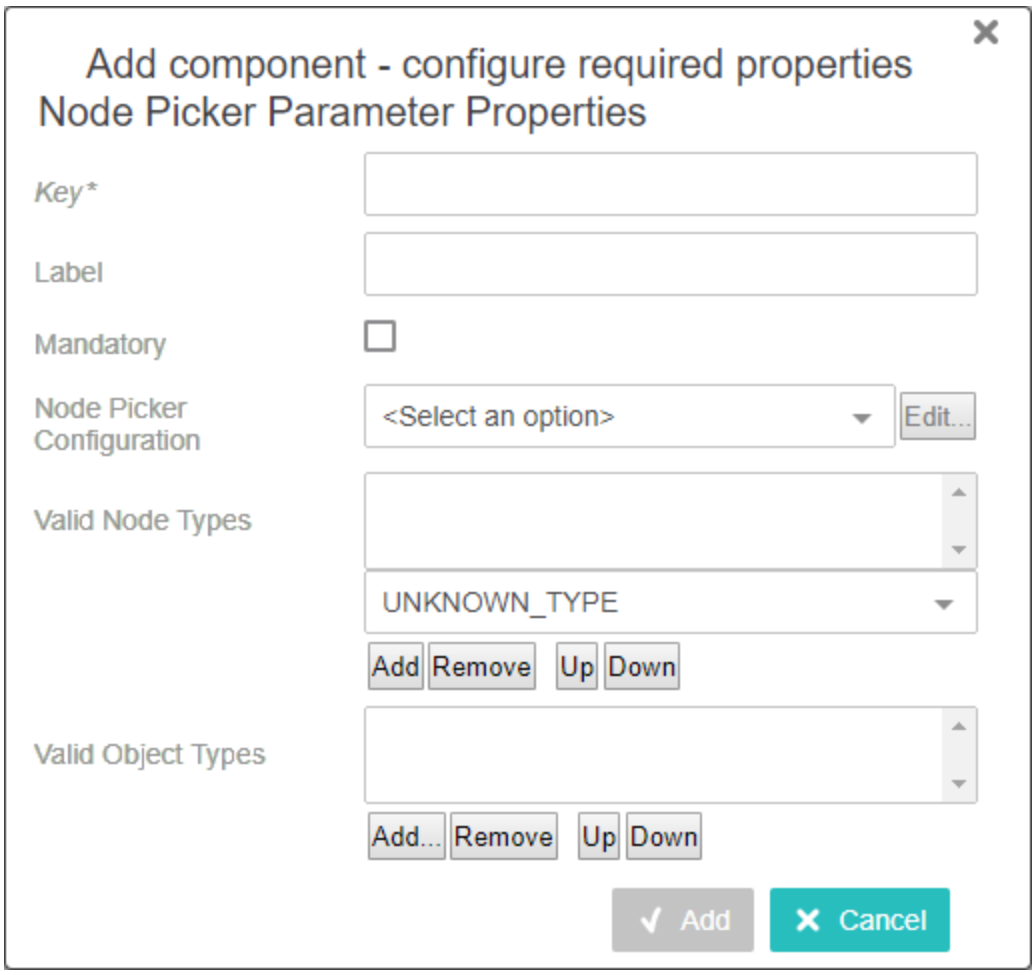

3. Click the **Add** button beneath the Bulk Updates parameter, and the 'Add component - configure required properties' for the 'Bulk Applications Update Properties' dialog will display (as shown below).




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).



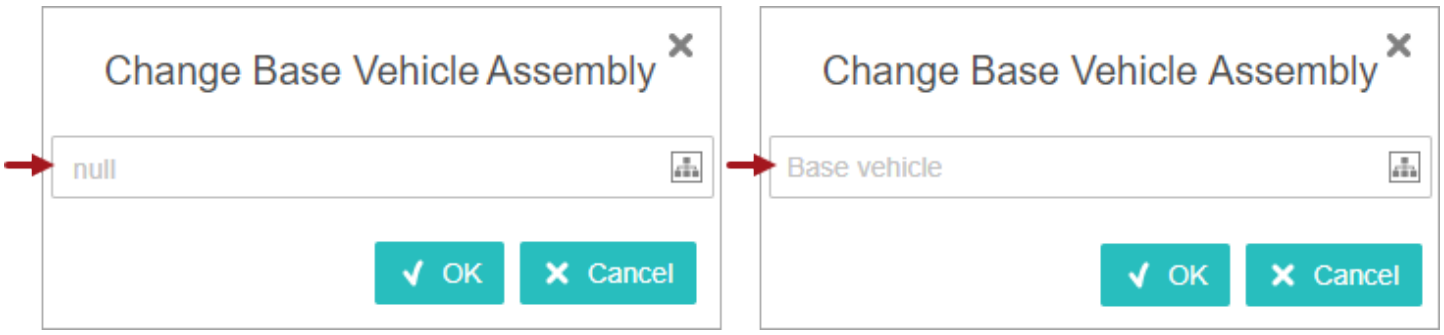

6. Select **Node Picker Parameter** > click the **Add** button to close the dialog, and the 'Add component - configure required properties' for the 'Node Picker Parameter Properties' dialog will display (as shown below).



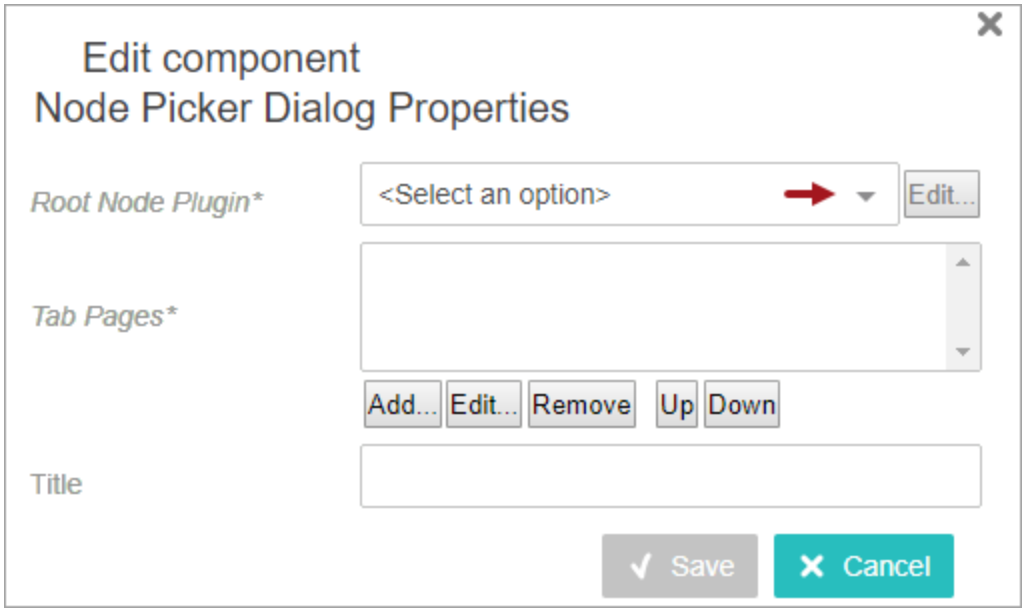

7. Within the Key parameter, enter the exact key created within the 'Assembly parameter key' parameter during step 5 of the **Configuring the Copy Application to Other Assembly Business Action in Workbench** topic. This is the only required parameter.

   **Important:** If this key is not entered exactly in both places, then the business action will not properly function within the Web UI.

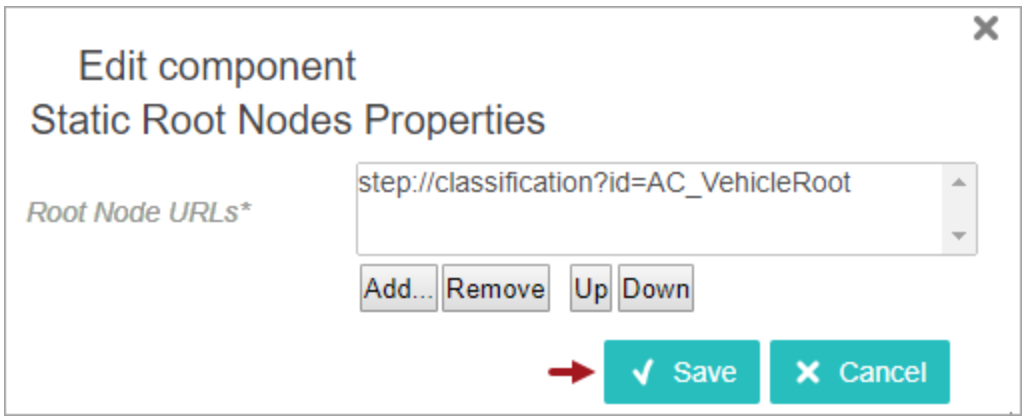
8. Within the Label parameter, enter text that will prompt the user as to what they should select when using this business action. For example, when the Label parameter is blank the parameter within the dialog will display as 'null' as shown in the image on the left. If the Label parameter is populated with 'Base vehicle' the parameter within the dialog will display with 'Base vehicle' as shown within the image on the right.



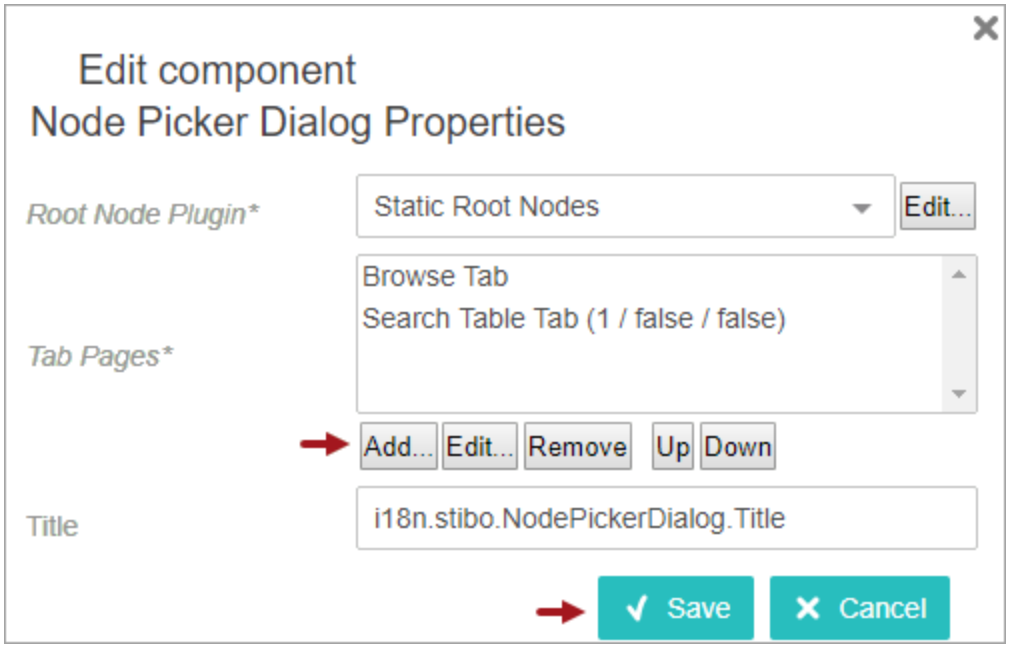

9. Use the Node Picker Configuration parameter dropdown to select the **Node Picker Dialog** option, and the 'Edit component' for the 'Node Picker Dialog Properties' dialog will display.




10. 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.
11. 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.



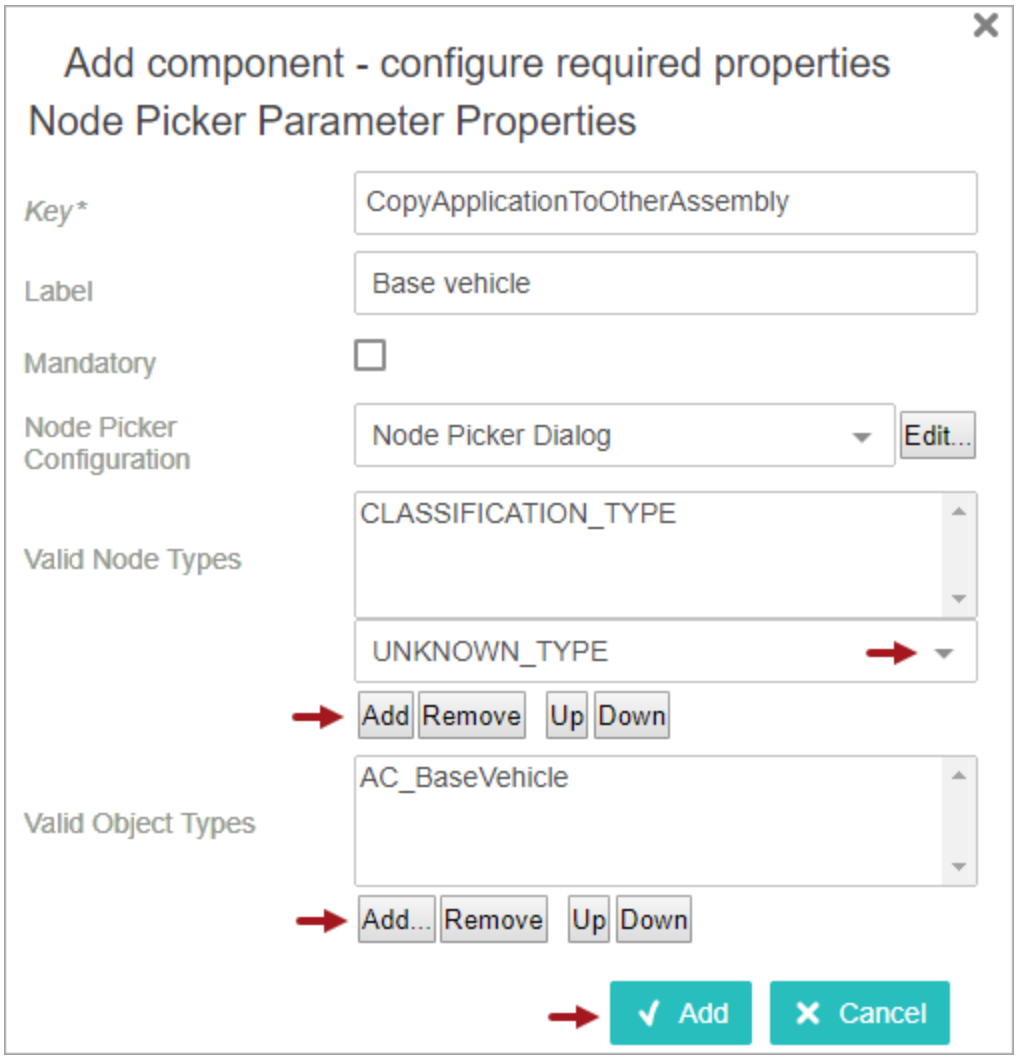

12. Click the **Save** button, and the 'Edit component' for the 'Node Picker Dialog Properties' dialog will display with the Root Node Plugin parameter populated with Static Root Nodes.
13. Click the **Add** button beneath the Tab Pages parameter > select the **Browse Tab** component > click the **Add** 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.



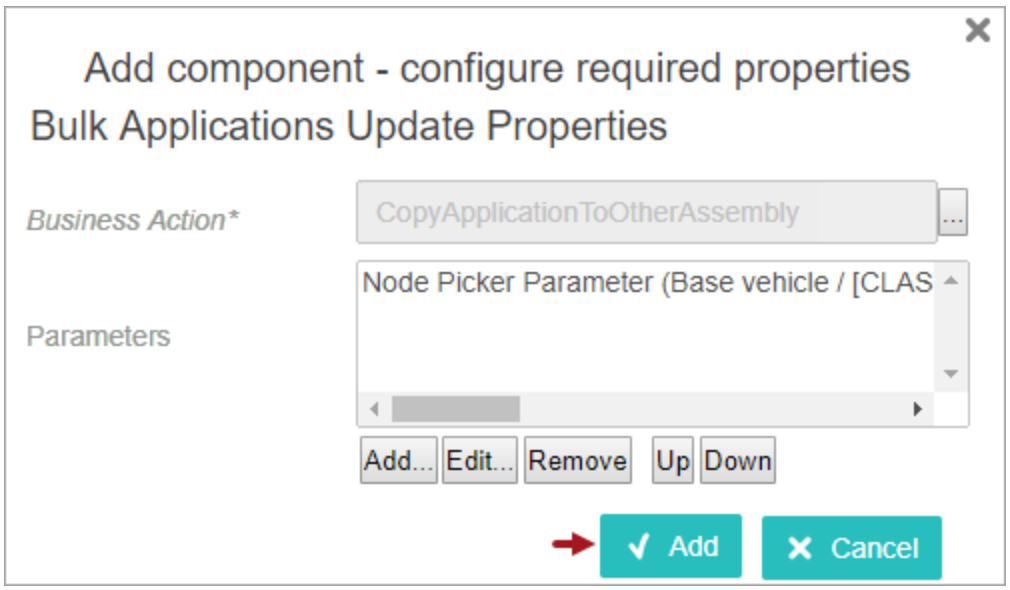

**Note:** Leave the Title parameter blank and *after the configuration is saved* an i18n key will be populated (as shown above). For more information, refer to the **Localization** topic within the **Administration Portal** section of the **STEP Online Help**. Otherwise, text entered within the Title 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.

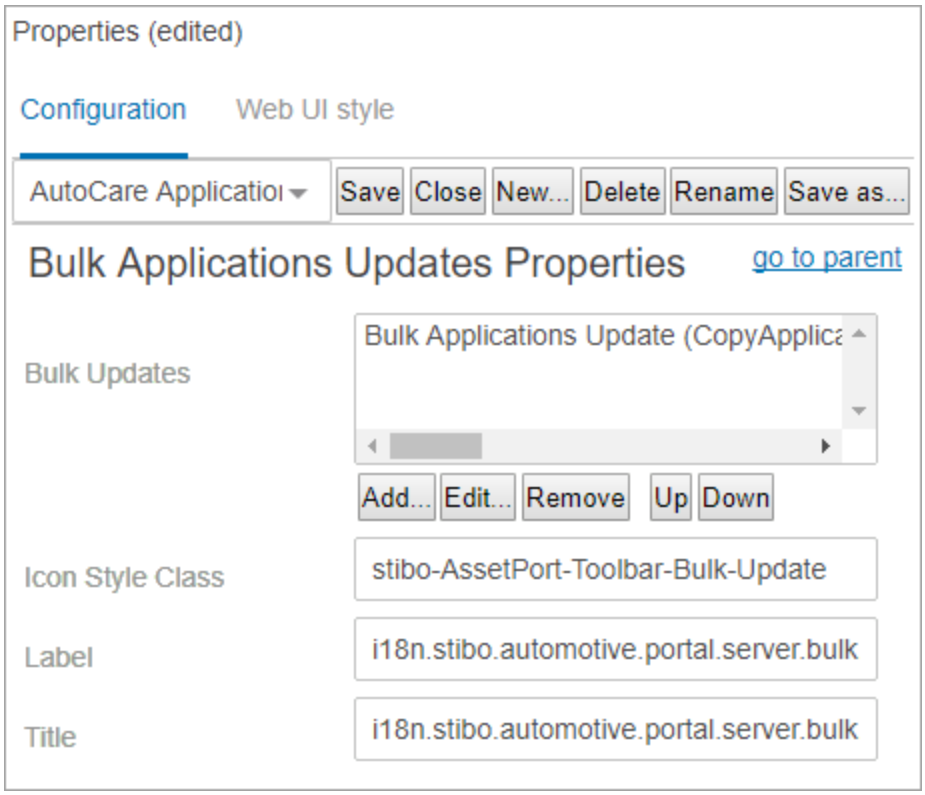
15. Use the dropdown located beneath the Valid Node Types parameter to select **CLASSIFICATION_TYPE** > click the **Add** button so that CLASSIFICATION_TYPE is displayed within the Valid Node Types parameter (as shown below).
16. Click the **Add** button beneath the Valid Object Types parameter > select 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.




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.



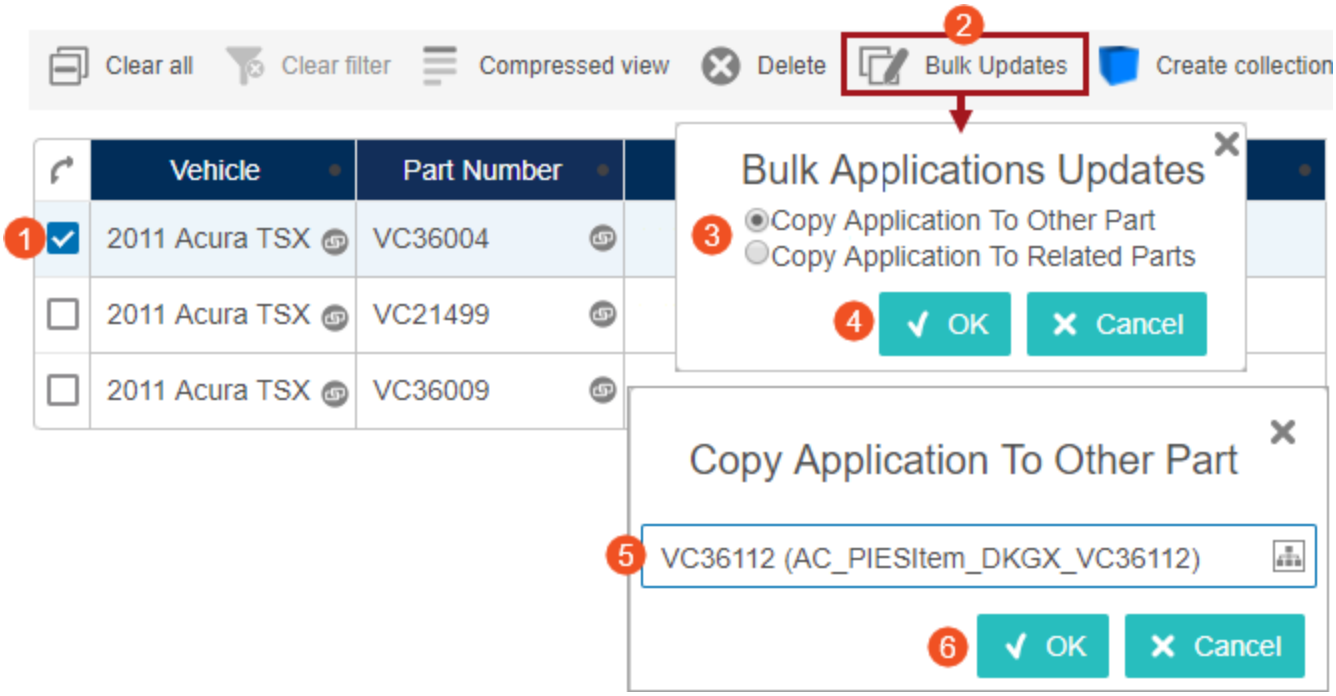

19. Click the **Save** button and then click the **Close** button to close the designer.

To use the newly configured business action, refer to 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, refer to 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.

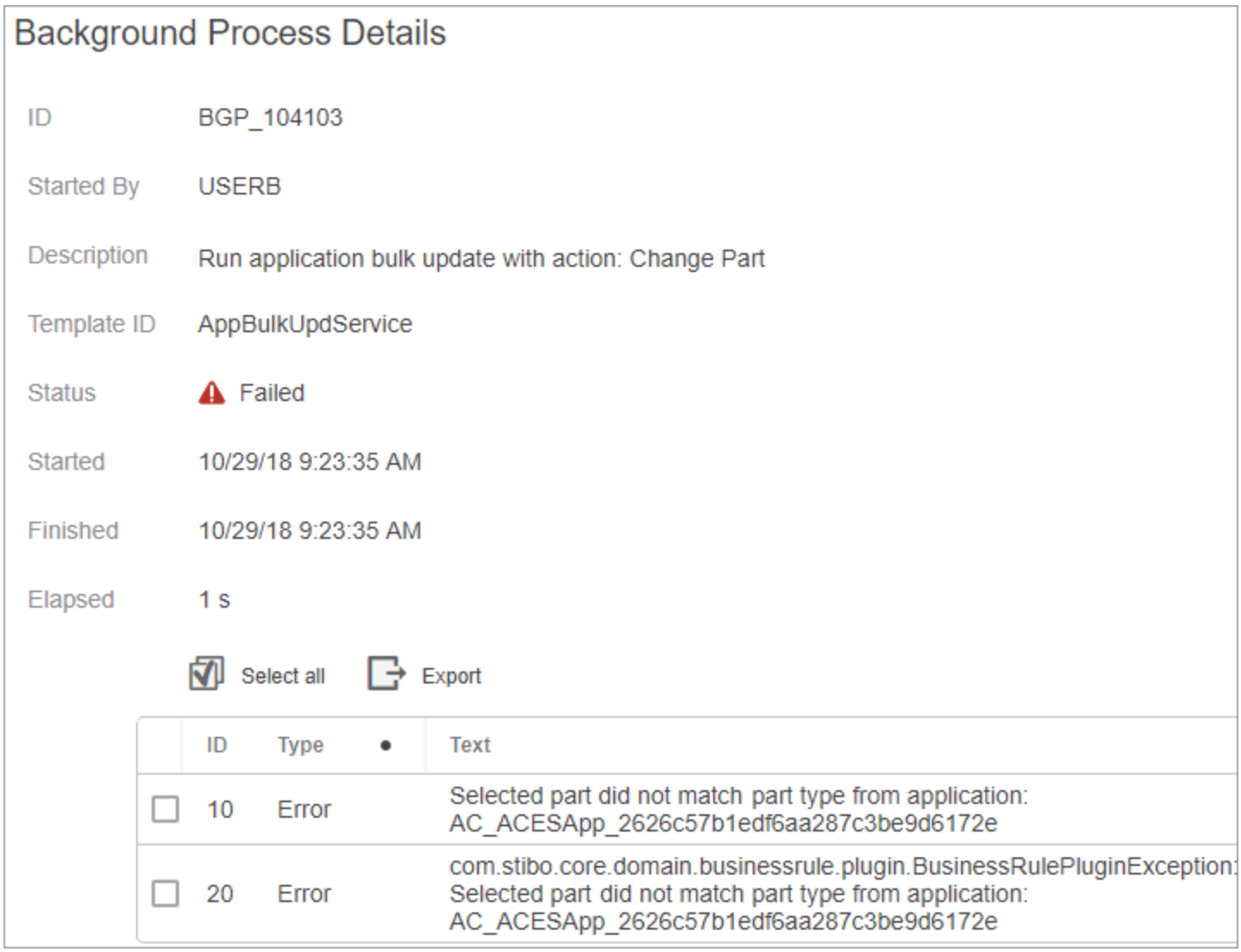
1. Within the configured Application Manager, search for applications and select one or more applications to be copied.
2. Click the **Bulk Updates** action button. If more than one bulk update is configured then the Bulk Updates dialog will display (as shown above), otherwise this dialog is skipped 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 open the 'Recent background processes' side panel by clicking on the gray BGP icon, , in the upper right corner of Web UI. For more information, refer to the **Recent Background Processes Side Panel** topic.

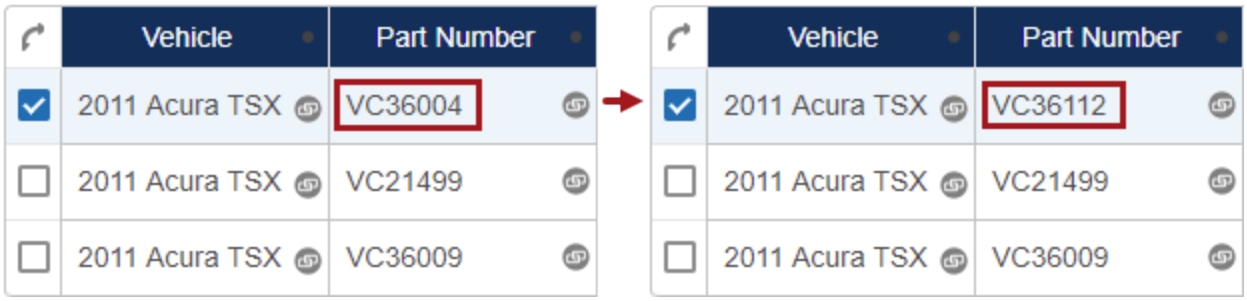
   If the business action BGP fails, access the Background Process Details page for more information.

   **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, refer to 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].




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.

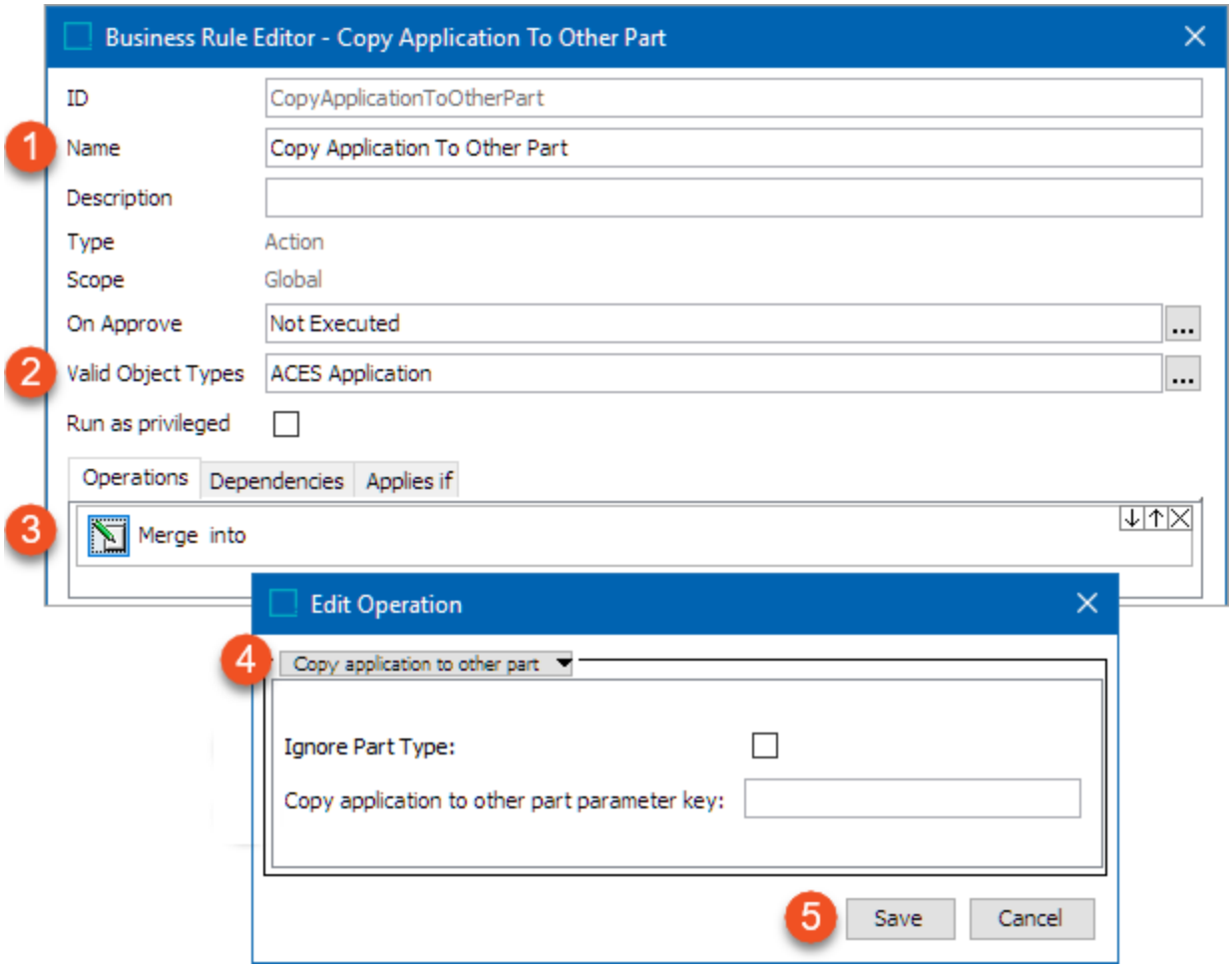



Setup is required within both STEP Workbench and Web UI for the action to be available to users. For more information, refer to the configuration topics below:

- 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.

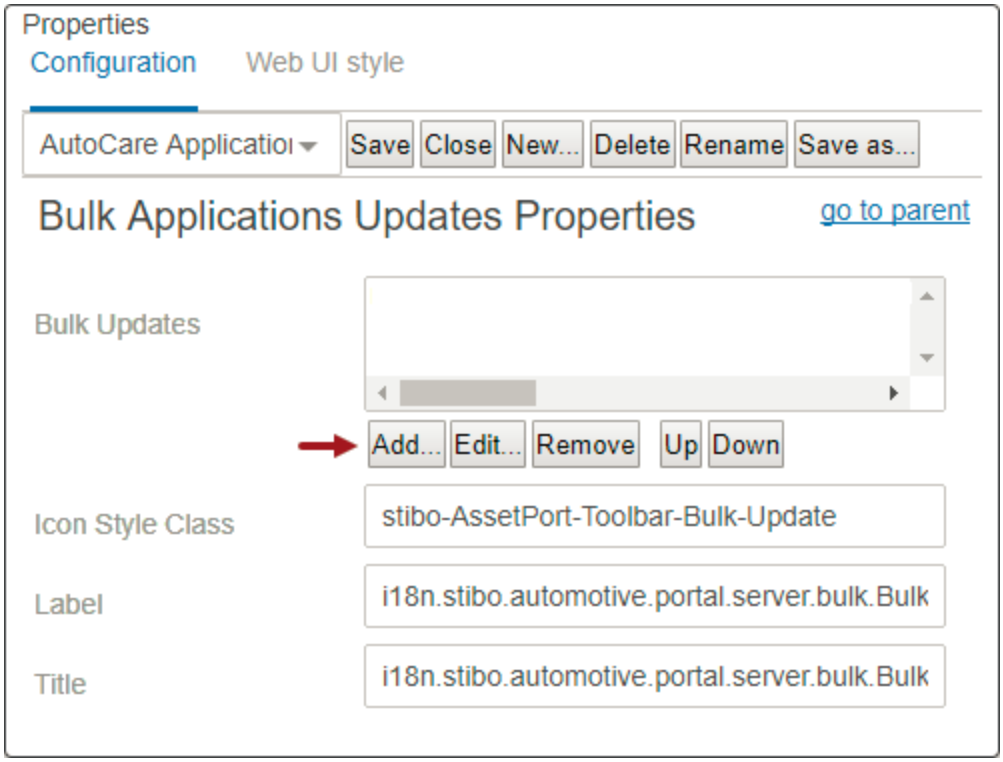
5. Click the **Save** button and continue to the next topic, **Configuring the Copy Application to Other Part Business Action in Web UI**.

### Configuring the Copy Application to Other Part Business Action in Web UI

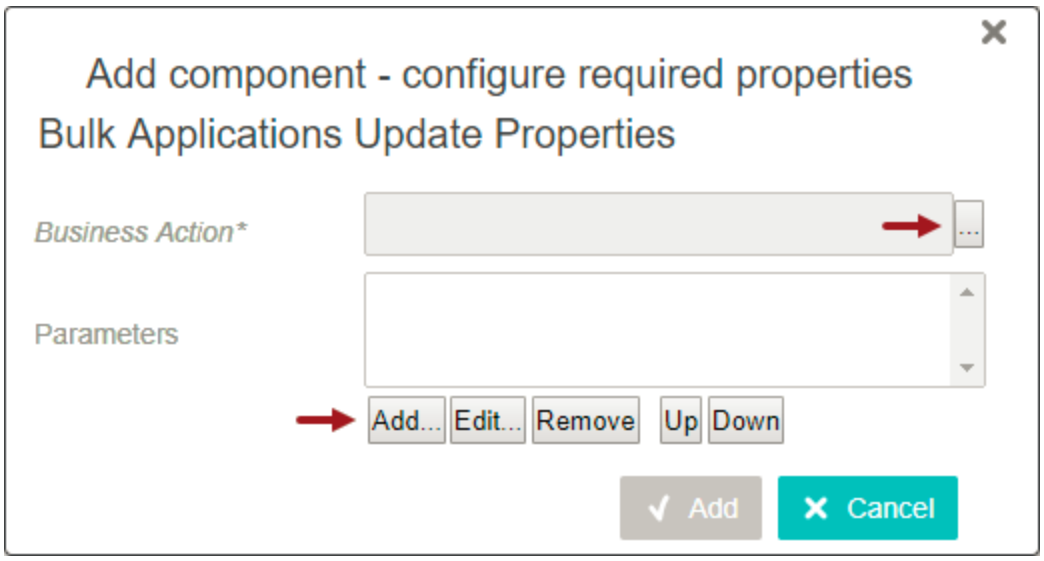
Once the business action has been configured in the workbench, configuration within Web UI is necessary. This section addresses the steps necessary within Web UI.

**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... .'

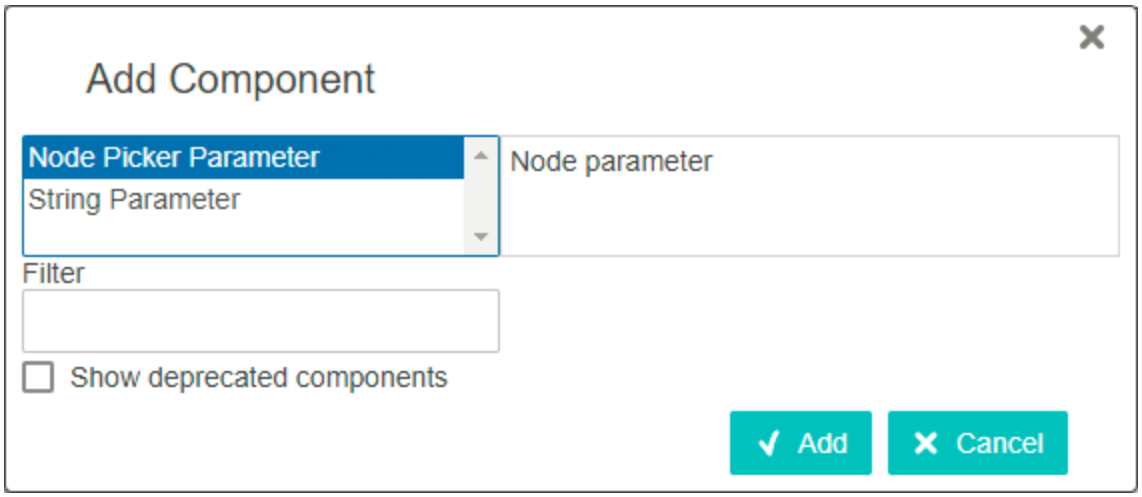
1. Access the Application Manager screen where the business action needs to be available to users.
2. Access the Designer > navigate to Node List Properties > Child Components > Actions > Double click **Bulk Applications Updates** and the 'Bulk Applications Updates Properties' dialog will display (as shown below).



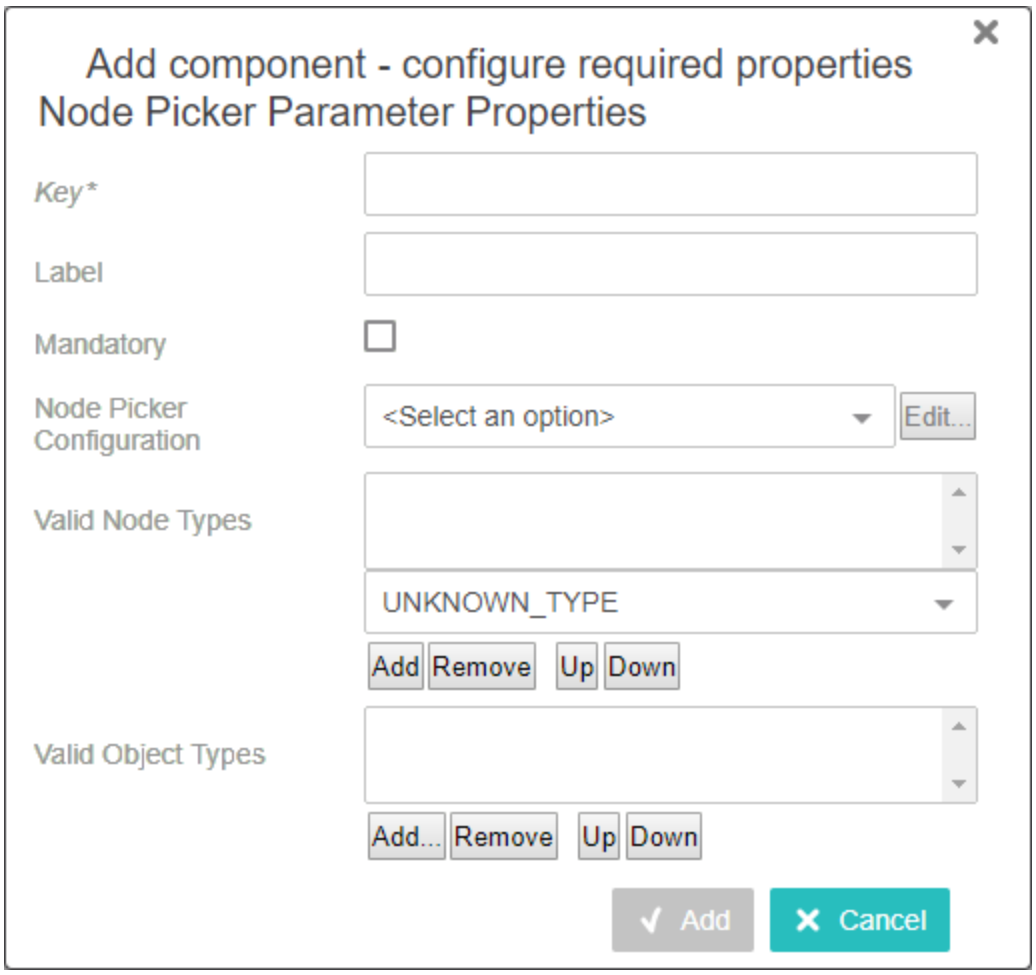

3. Click the **Add** button beneath the Bulk Updates parameter, and the 'Add component - configure required properties' for the 'Bulk Applications Update Properties' dialog will display (as shown below).




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).



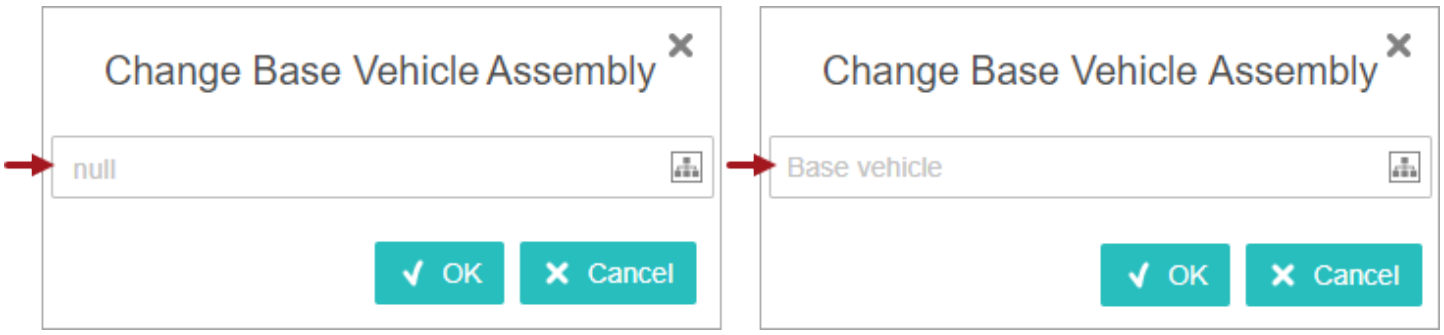

6. Select **Node Picker Parameter** > click the **Add** button to close the dialog, and the 'Add component - configure required properties' for the 'Node Picker Parameter Properties' dialog will display (as shown below).



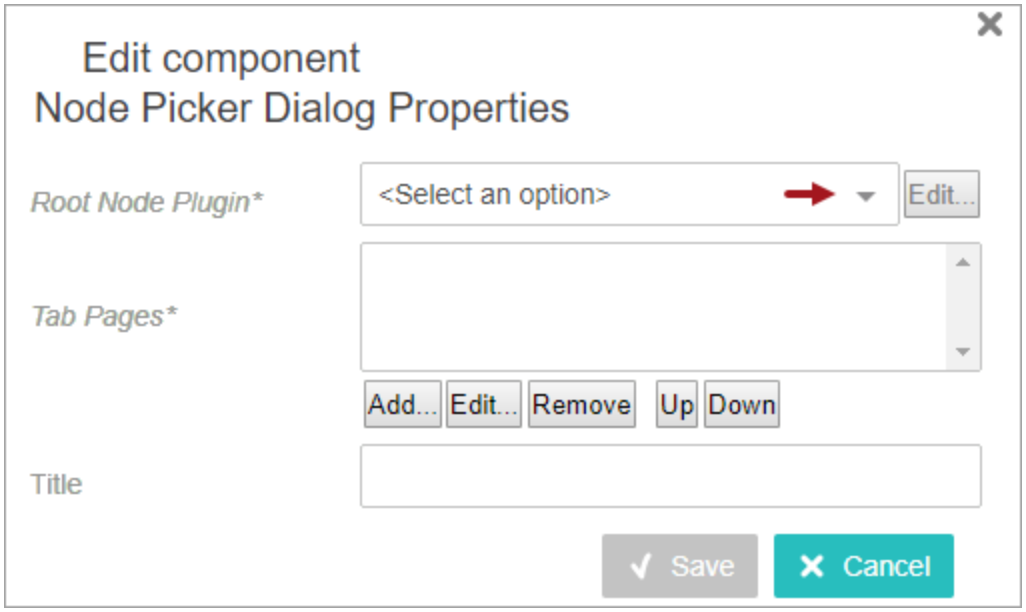

7. Within the Key parameter, enter the exact key created within the 'Copy application to other part parameter key' parameter during step 5 of the **Configuring the Copy Application to Other Part Business Action in Workbench** topic. This is the only required parameter.

   **Important:** If this key is not entered exactly in both places, then the business action will not properly function within the Web UI.

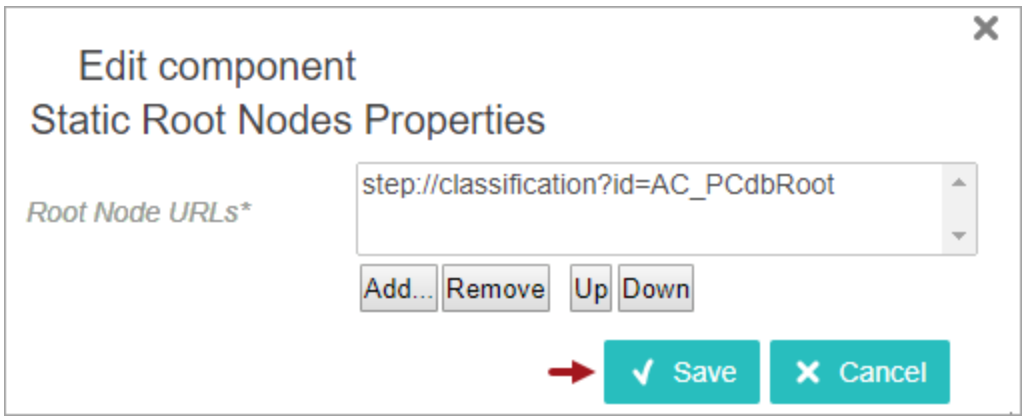
8. Within the Label parameter, enter text that will prompt the user as to what they should select when using this business action. For example, when the Label parameter is blank the parameter within the dialog will display as 'null' as shown in the image on the left. If the Label parameter is populated with 'Base vehicle' the parameter within the dialog will display with 'Base vehicle' as shown within the image on the right.



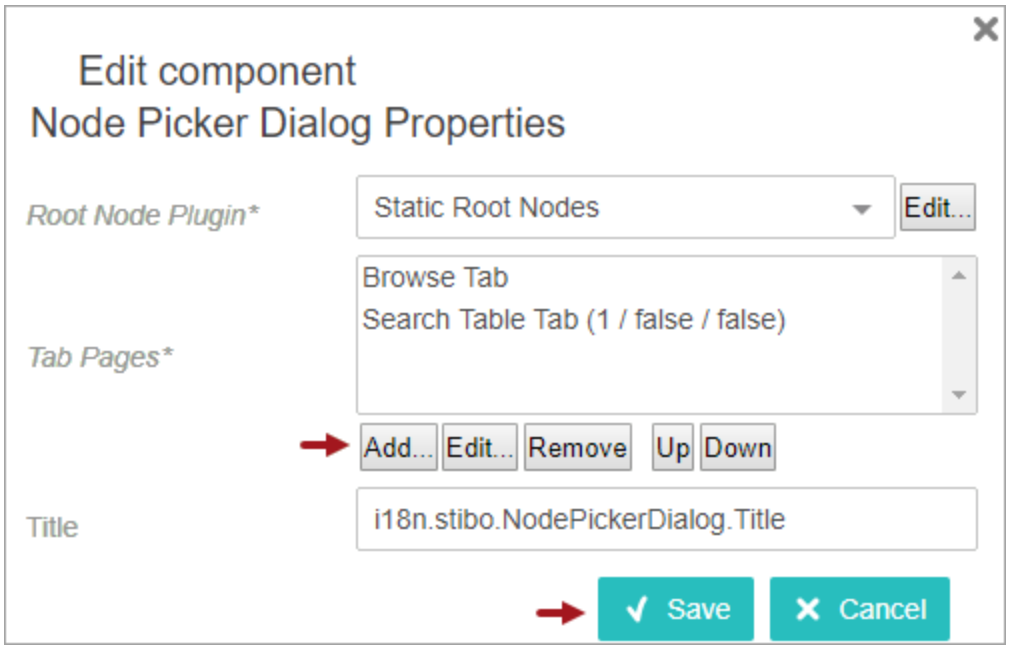

9. Use the Node Picker Configuration parameter dropdown to select the **Node Picker Dialog** option, and the 'Edit component' for the 'Node Picker Dialog Properties' dialog will display.




10. 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.
11. 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.



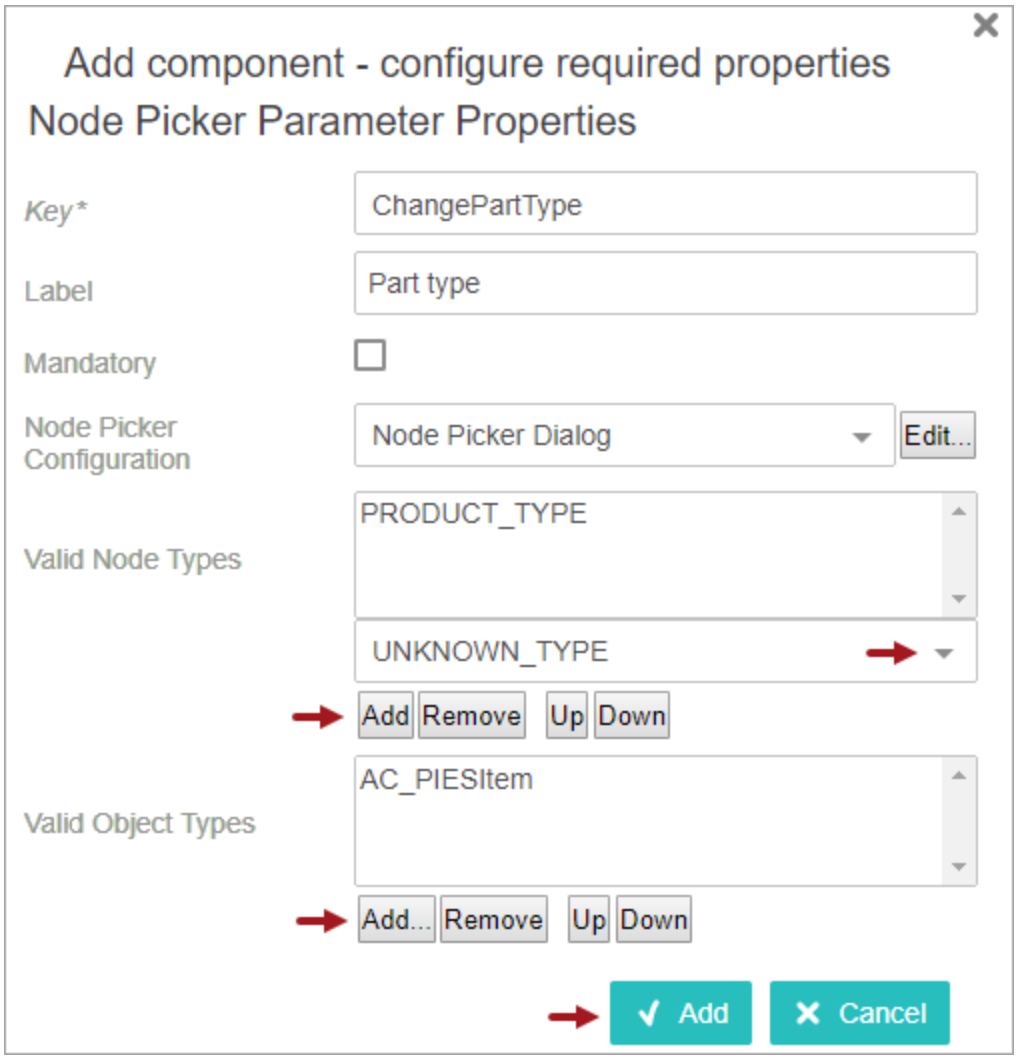

12. Click the **Save** button, and the 'Edit component' for the 'Node Picker Dialog Properties' dialog will display with the Root Node Plugin parameter populated with Static Root Nodes.
13. Click the **Add** button beneath the Tab Pages parameter > select the **Browse Tab** component > click the **Add** 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.



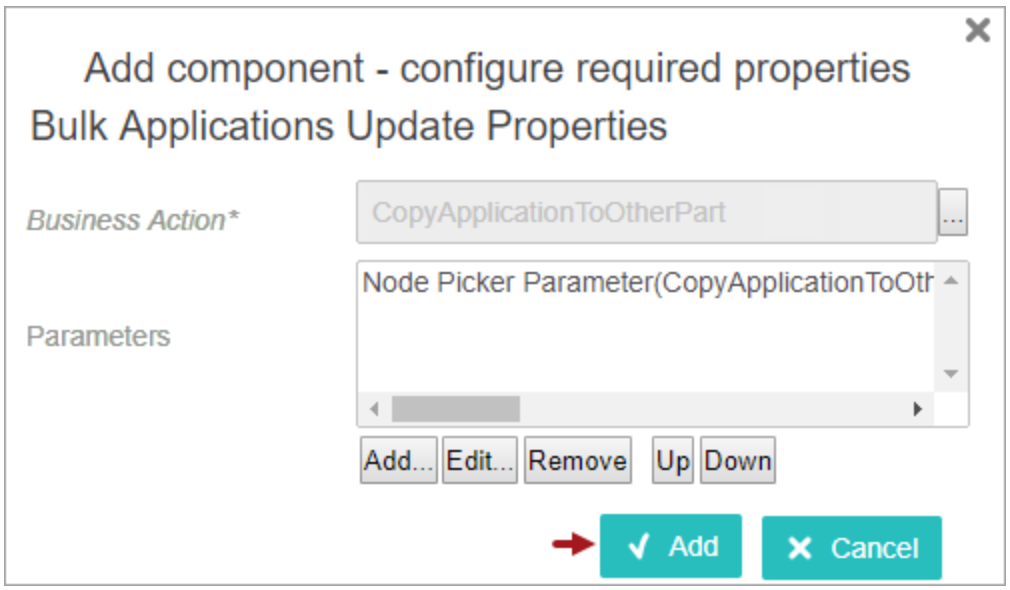

**Note:** Leave the Title parameter blank and *after the configuration is saved* an i18n key will be populated (as shown above). For more information, refer to the **Localization** topic within the **Administration Portal** section of the **STEP Online Help**. Otherwise, text entered within the Title 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.

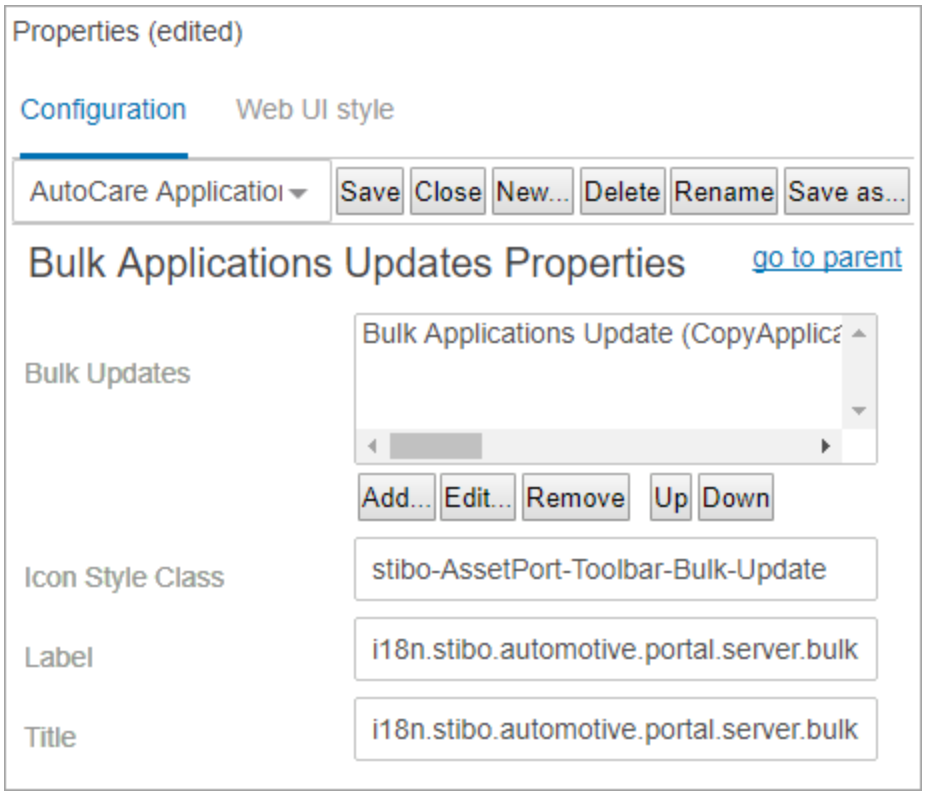
15. Use the dropdown located beneath the Valid Node Types parameter to select **PRODUCT_TYPE** > click the **Add** button so that PRODUCT_TYPE is displayed within the Valid Node Types parameter (as shown below).
16. Click the **Add** button beneath the Valid Object Types parameter > select the desired 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.




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, refer to 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

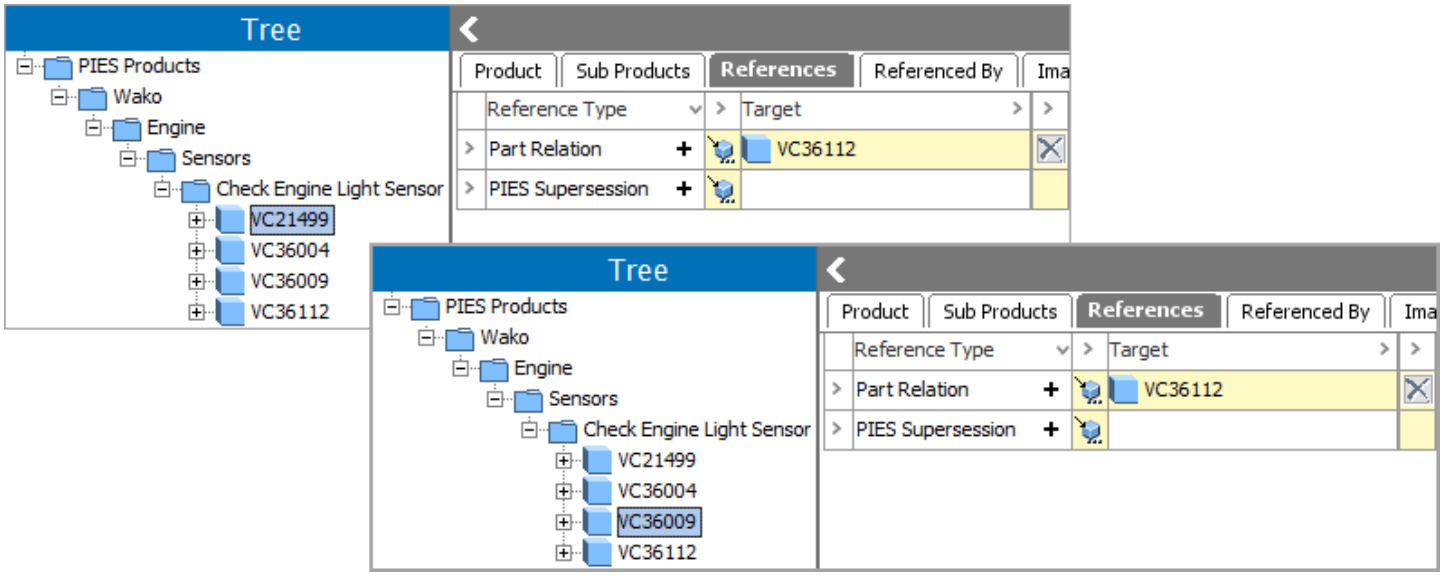
This automotive business action allows users to copy one or more existing applications to one or more related parts by clicking a Bulk Updates action button within an Application Manager. However, it cannot copy the part of a missing application, and thus any missing applications selected when the bulk update is run will be ignored. For more information, refer to 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.

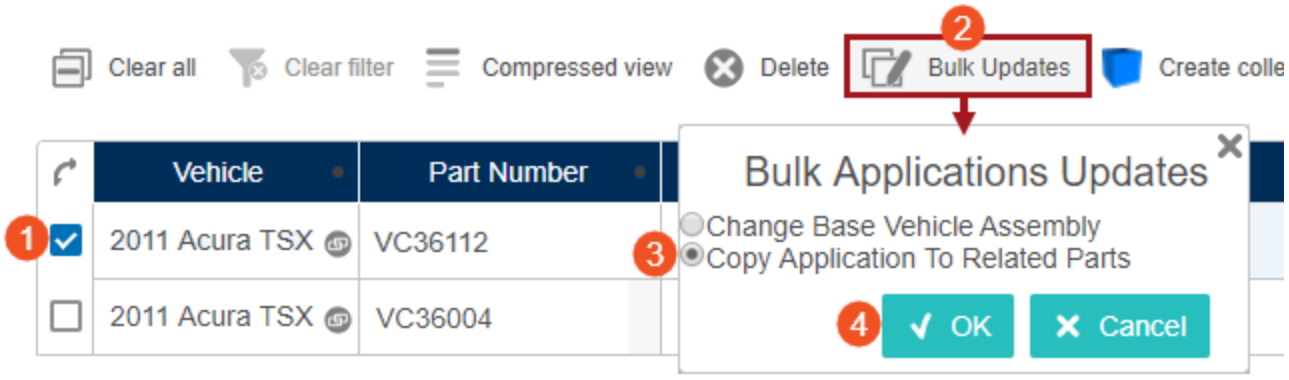
For example, when using the References tab in workbench to view part VC21499 and VC36009, the Part Relation Reference Type can be viewed as 'VC36112.'




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).

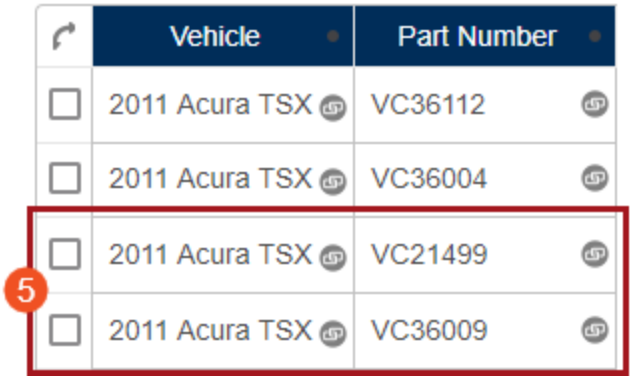
**Steps for Using the Business Action in Web UI**

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).



Setup is required within both STEP Workbench and Web UI for the action to be available to users. For more information, refer to the configuration topics below:

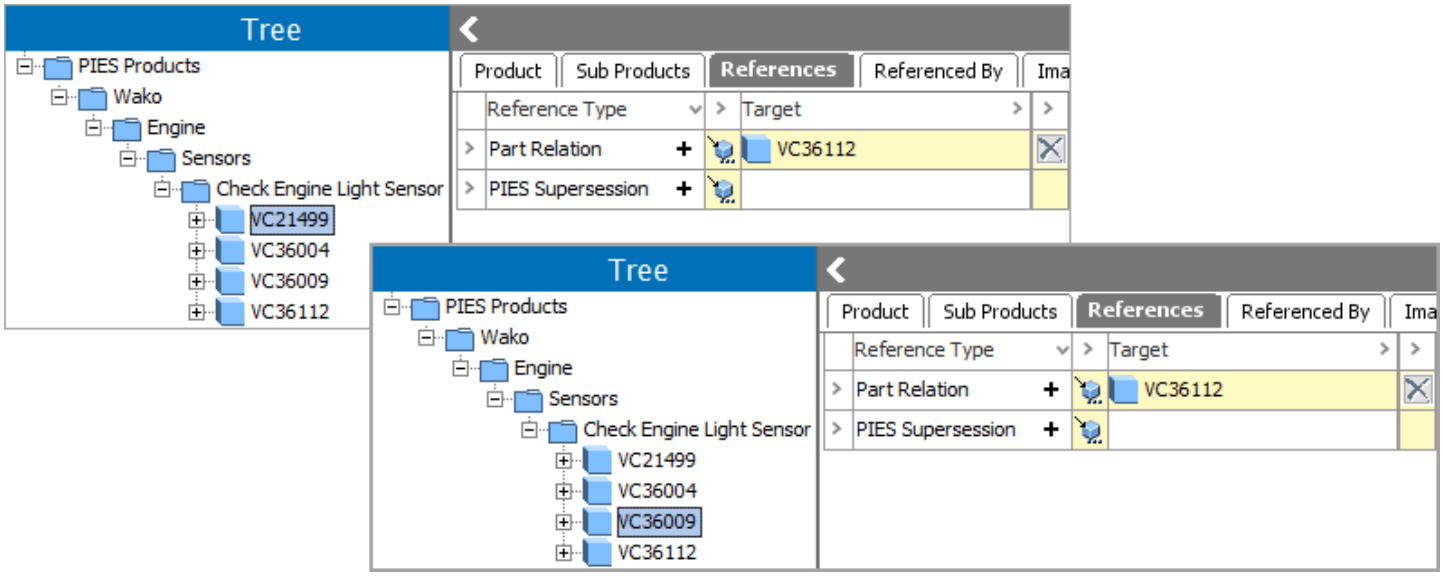
- Configuring the Copy Applications to Related Parts Business Action in Workbench
- Configuring the Copy Applications to Related Parts Business Action in Web UI

## 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.

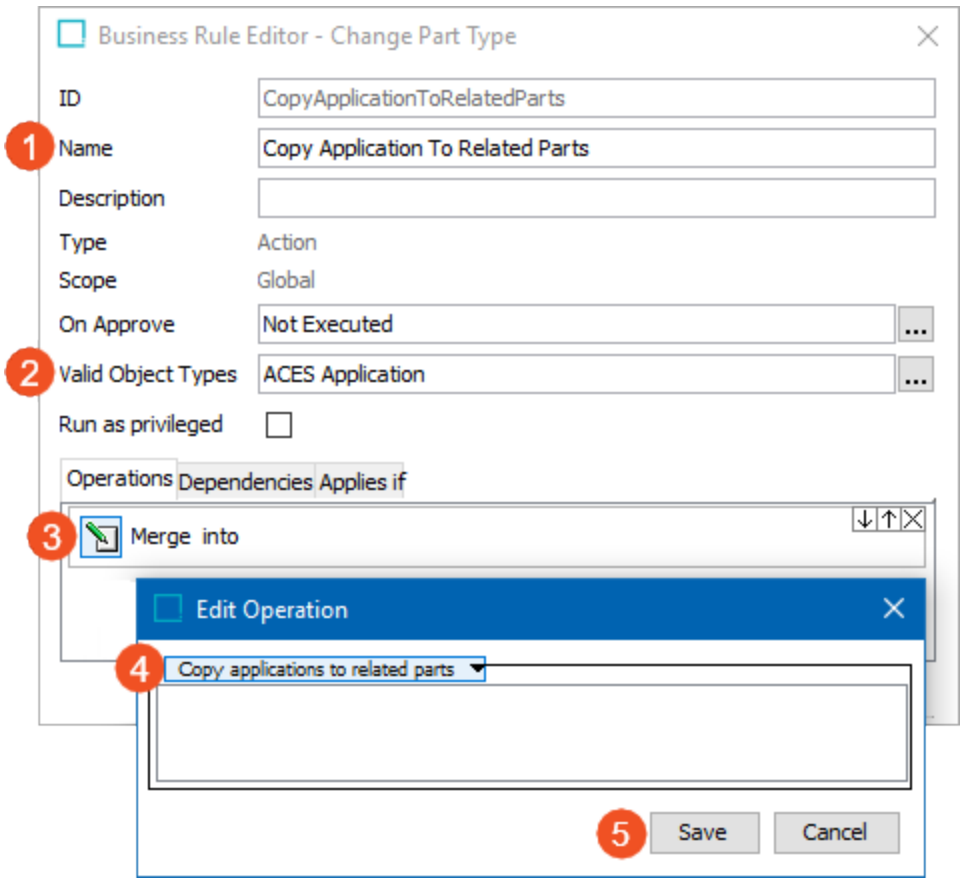
### Configuring the Reference Type

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.




### 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).

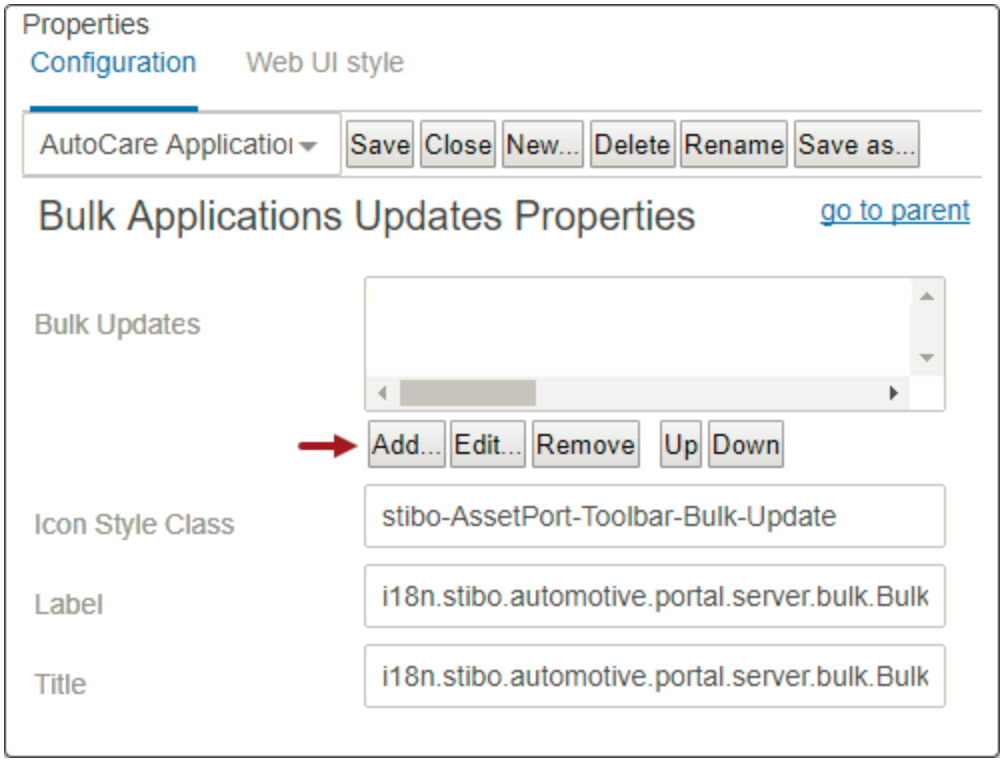
2. Edit the new business rule, click the ellipsis button (...) next to the Valid Object Types parameter and select the valid object types for this business action. For this example, the 'ACES Application' object type is selected, however the TecDoc and NAPA applications can also be selected within the 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

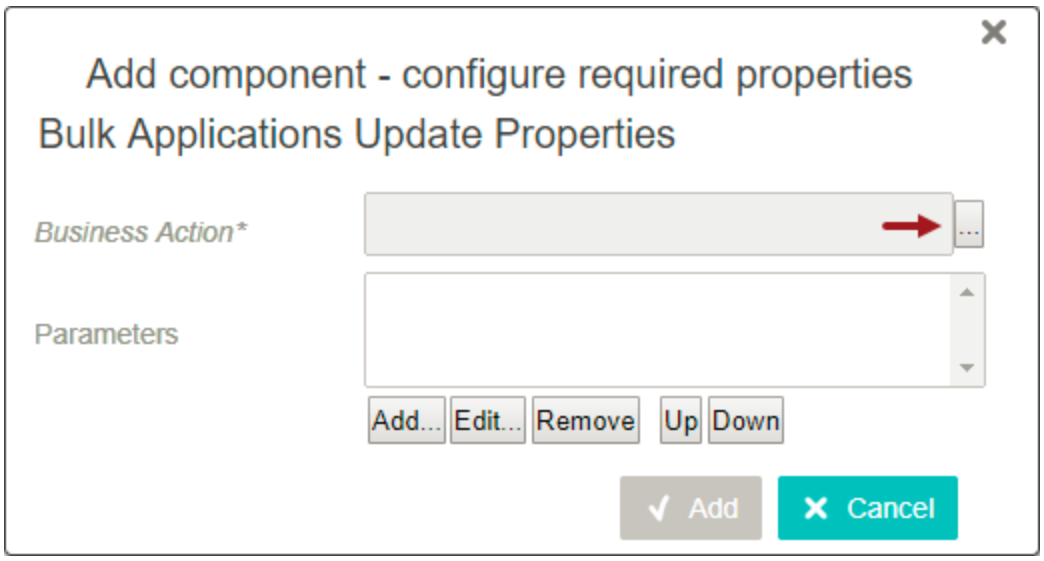
Once the business action has been configured in the workbench, configuration within Web UI is necessary. This section addresses the steps necessary within Web UI.

**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... .'

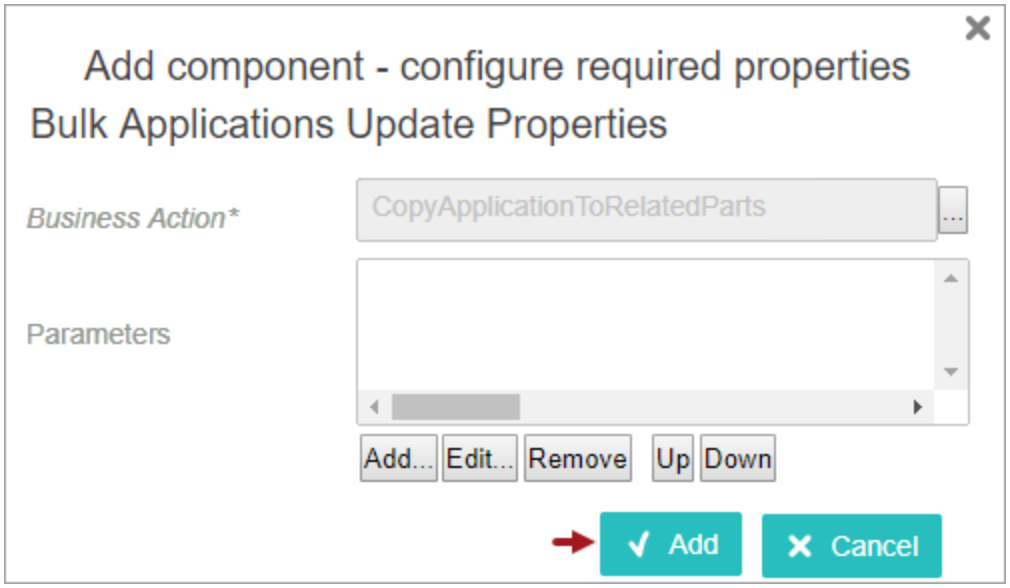
1. Access the Application Manager screen where the business action needs to be available to users.
2. Access the Designer > navigate to Node List Properties > Child Components > Actions > Double click **Bulk Applications Updates** and the 'Bulk Applications Updates Properties' dialog will display (as shown below).



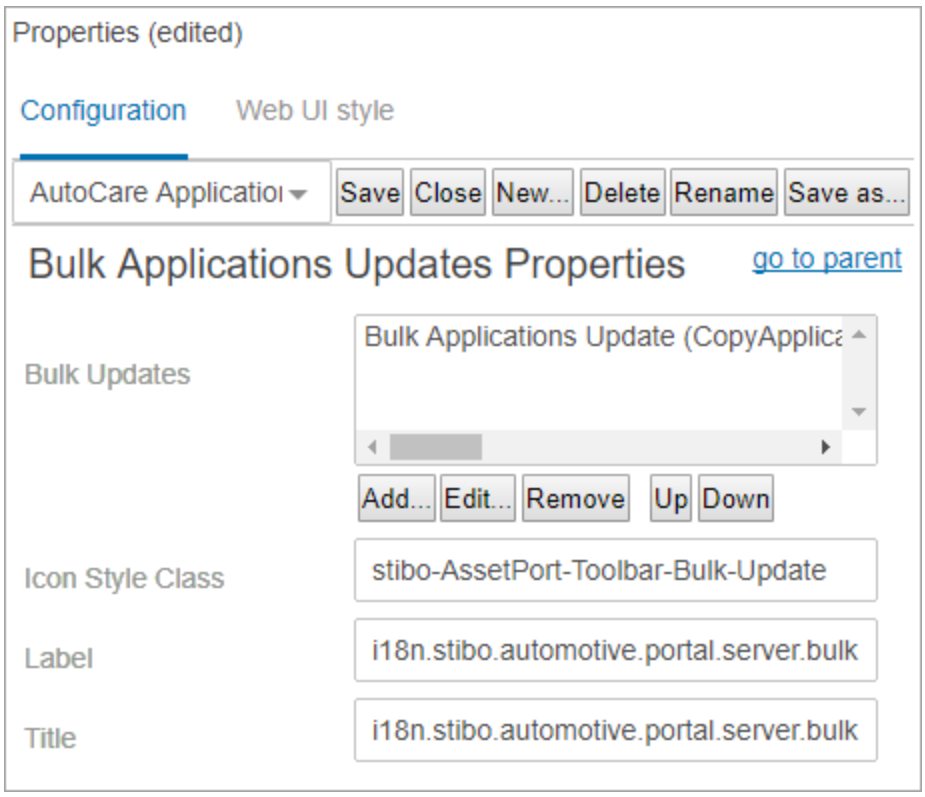

3. Click the **Add** button beneath the Bulk Updates parameter, and the 'Add component - configure required properties' for the 'Bulk Applications Update Properties' dialog will display (as shown below).




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.

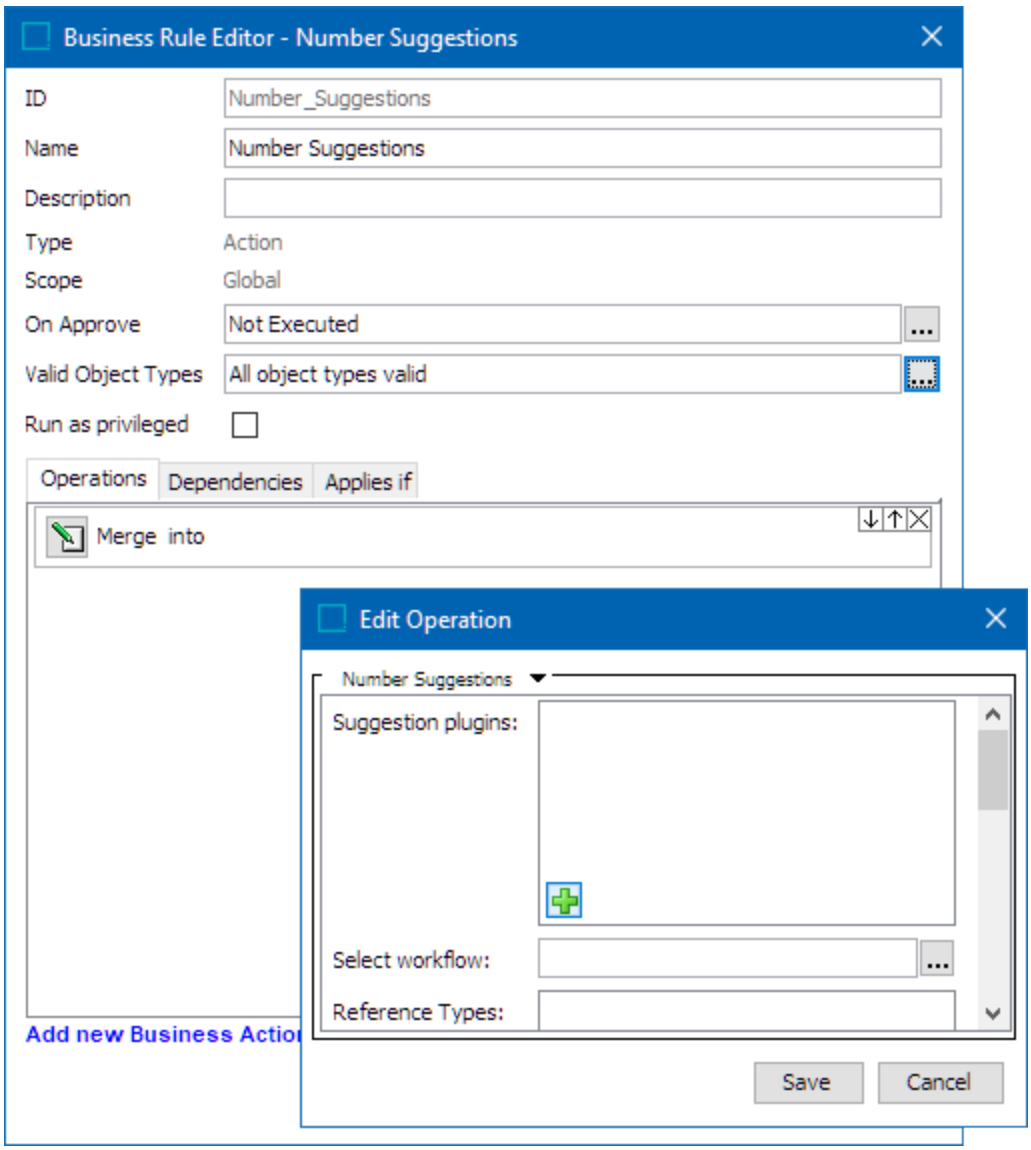



6. Click the **Save** button and then click the **Close** button to close the designer.

To use the newly configured business action, refer to 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: Number Suggestions

The 'Number Suggestions' operation is found within the STEP Workbench Business Rule Editor under the Automotive menu and requires population of multiple parameters. Setup is required in the STEP Workbench for the action to be available to users.




When used within a business action, this operation checks the selected part has any potential part numbers as suggestions, and when a part number is suggested that meet the condition defined in the suggestion plugin, it will initiate the selected part into a workflow configured within the 'Select workflow' parameter of the business action.

If the part finds no suggested part numbers based on the condition defined within the 'Suggested plugins' parameter then the part will not be initiated into the workflow.

As a prerequisite, one unique workflow that can handle the part is necessary. This allows each customer to determine their own strategy for managing parts that has potential part numbers as suggestions.

**Note:** How the customer handles such part in a different workflow is beyond the scope of this topic. It is a customized solution that the Solution Consultant can implement on to their system.

### Configuring Business Action for Number Suggestions

To properly configure this business action to handle parts that has part number as suggestions, the following is needed:

- Create / Identify a valid workflow that handles parts that has other part number as suggestions.
- Configure a business action to use the 'Number Suggestions' operation

## Create / Identify a Valid Workflow

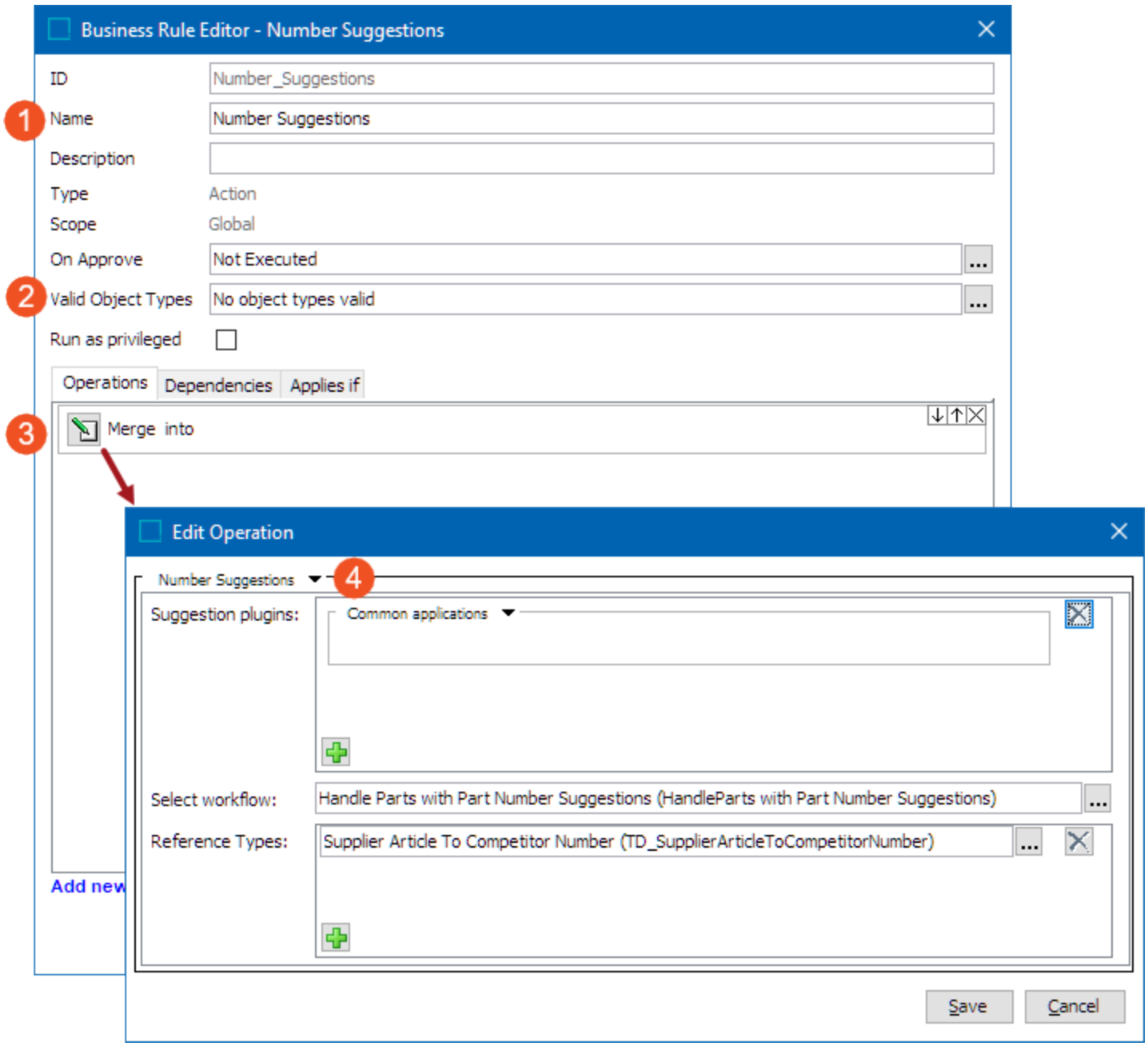
1. Create the necessary workflow (or verify that it exists) to be used to handle the parts that has part number as suggestions.
2. Make the workflow valid for the object type to be handled.

   For example:

   - The workflow used to handle PIES Item that has part number (Pies Interchange Item) as suggestions must be valid on the AC_PIESItem object type.
   - The workflow used to handle TecDoc Supplier article that has part number (competitor and/or OE number) as suggestions must be valid on the DS_Supplier Article object type.
   - The workflow used to handle NAPA Product that has part number (NAPA Interchange Product) as suggestions must be valid on the NAPA_Product object type.

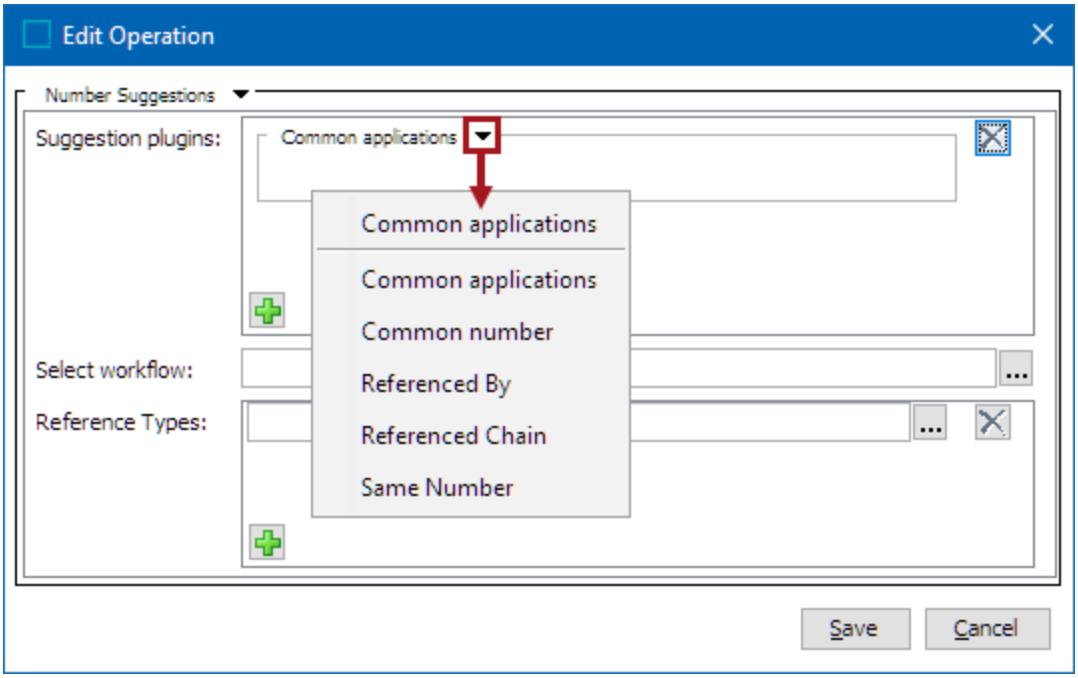
## Configure a Business Action to Use the 'Number Suggestions' Operation

1. Create a business action and give it a name that users can easily identify. In the example below, the 'Number Suggestions' 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 Number Suggestions 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 **Number Suggestions**.



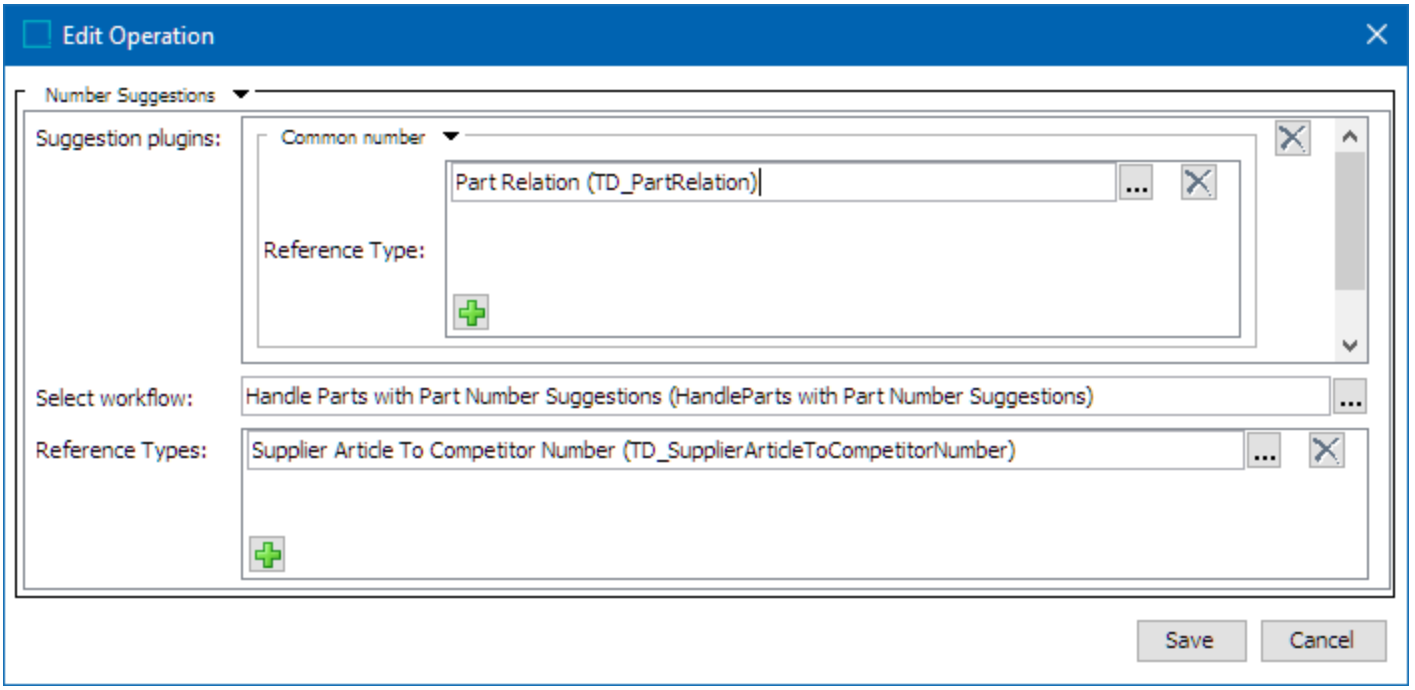

5. The following parameters will be displayed. Configure the parameters.
   - **Suggestion plugins:** Within this parameter, the user can add one or more plugins by clicking the add icon (). Technically, the criteria defined in this plugin will set the condition to look for the suggested part numbers for the part. 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 look for the suggested part numbers in accordance.

     In the dropdown that is available next to the 'Suggestion plugins' parameter, the following are the available options:



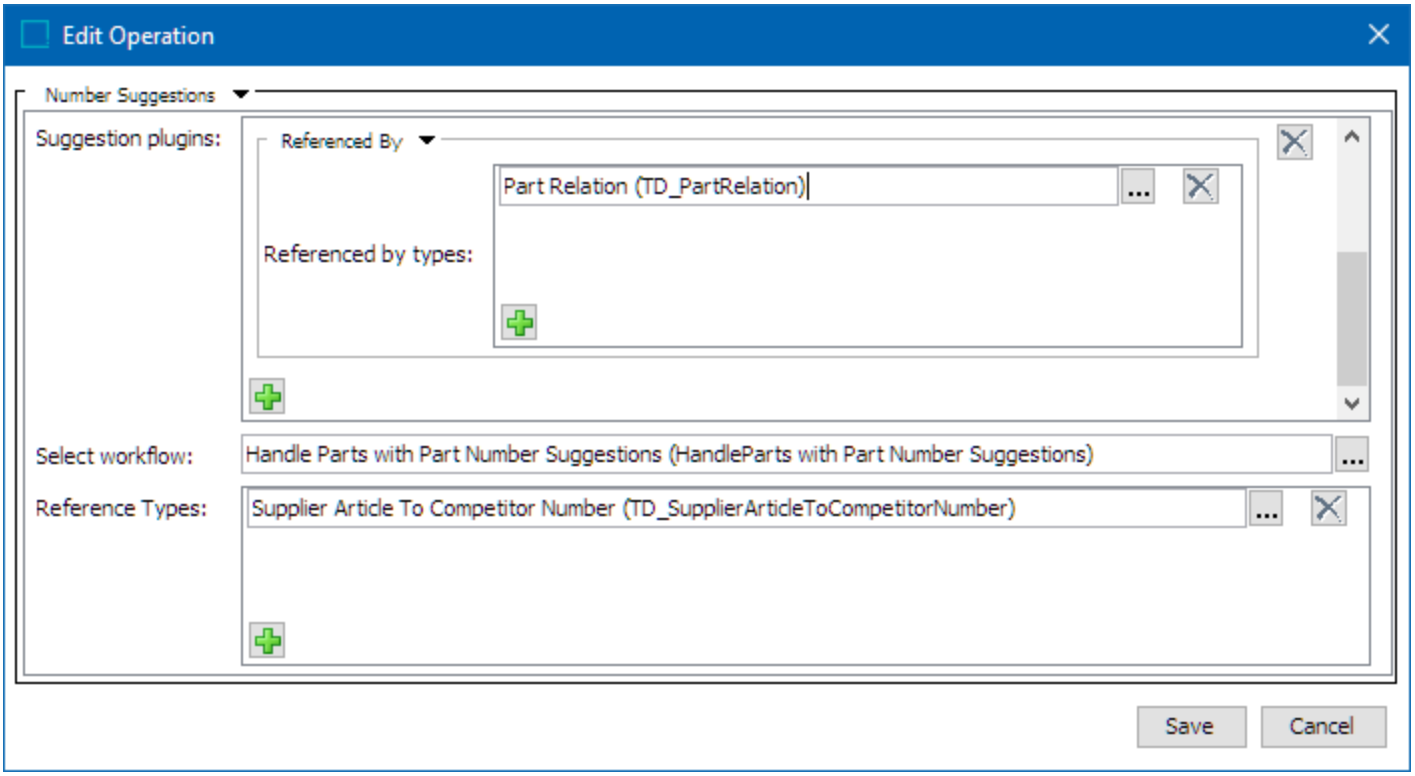

- **Common applications:** This plugin is used to suggest other part numbers that have the same applications as the current part number. Additional configuration parameters are not available for this plugin.
- **Common number:** This plugin suggests the other part number(s) that are referencing the same 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.

  Additional configuration is required for this plugin.



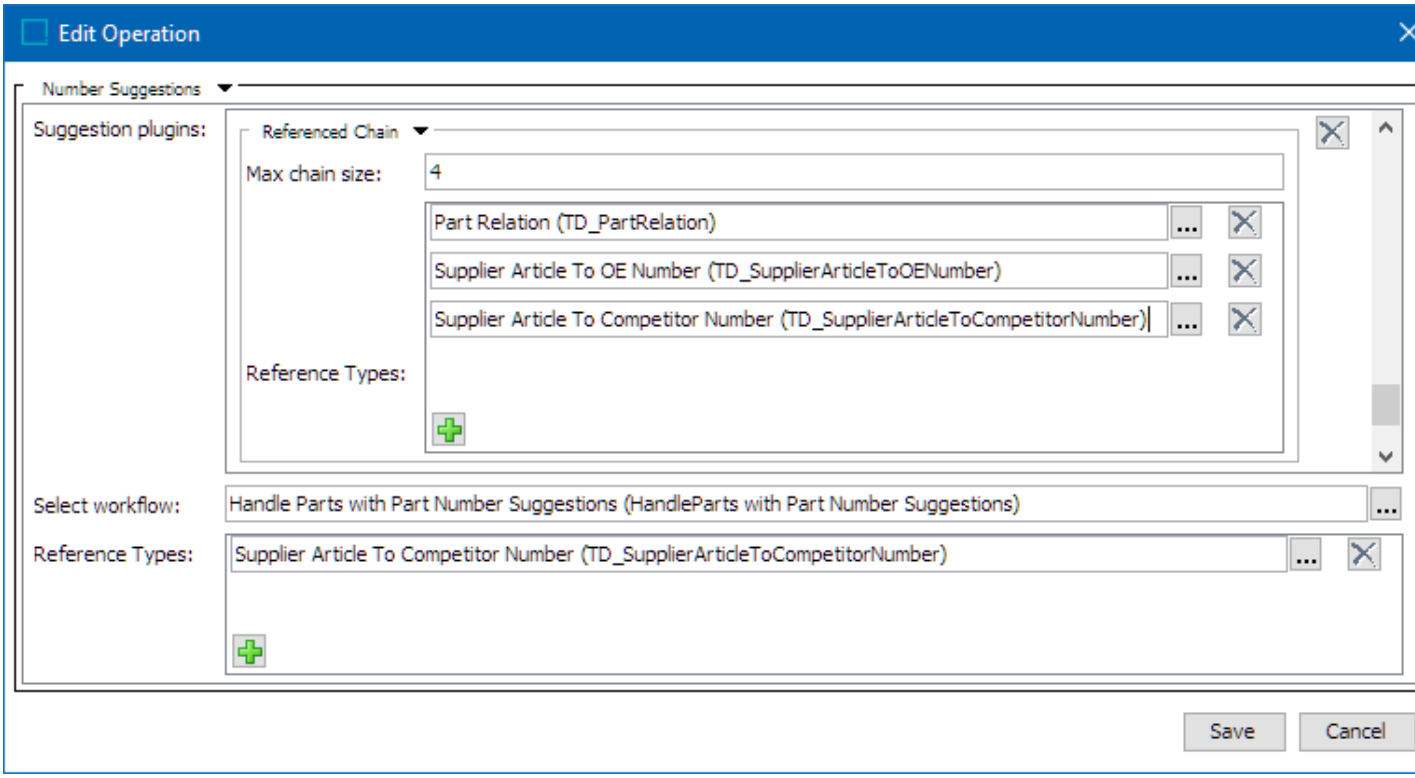

- **Reference Types:** It is required to add at least one Product Reference Type within this parameter so that when a part is selected, the reference types listed within this parameter are followed and display the target and/or related competitor and/or OE number information on the targets.

- **Referenced By:** This plugin is used for suggesting part numbers that are referencing the current part number. Additional configuration is required for this plugin. When a part is selected, the reference types listed within this parameter are followed, and the source part number and its related information is looked for.




- 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. The below mentioned parameters are to be configured for this plugin.




- **Max Chain Size:** 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:** It is required to add at least one Product Reference Type within this parameter. 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 Stibo Systems.

- **Workflow:** Click the ellipsis button (...) to find and select the workflow. The configured workflow should be able to handle the parts that has part numbers as suggestion.

- **Reference Types:** It is required 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.

**Note:** Upon configuration of the 'Numbers Suggestion' business action, the Automotive solution only provides an option to initiate the parts in another workflow. Users need to determine their own strategy for how to handle these parts 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, refer to the **Creating a Business Rule, Function, or Library** topic, and the **Editing a Business Rule or Function** topic within the **Business Rules** guide.

## 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.

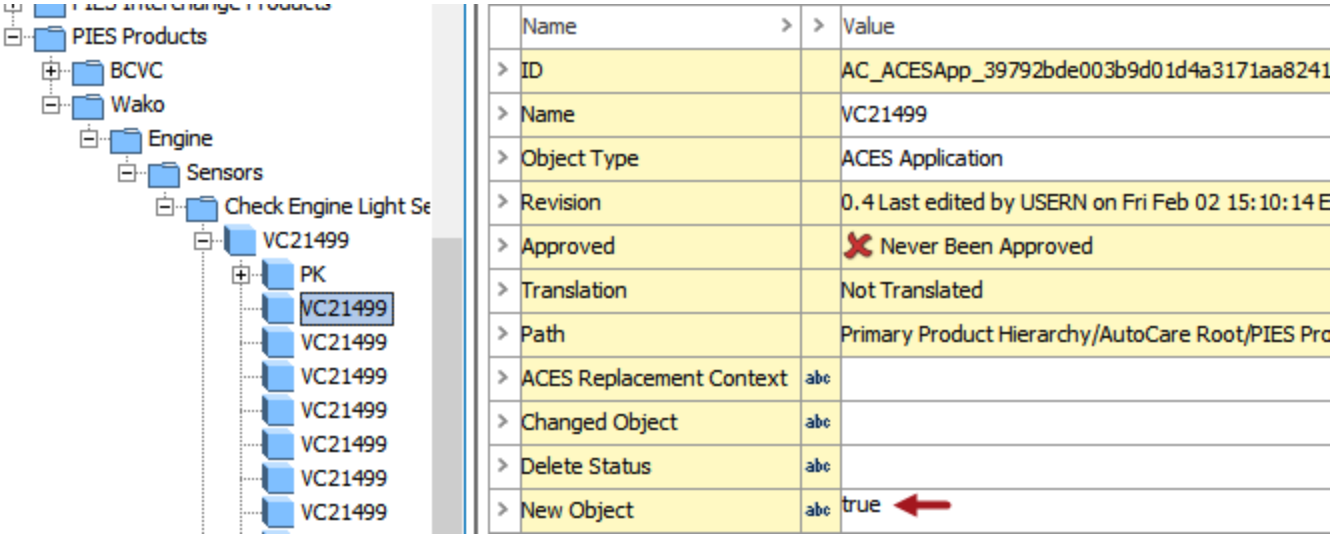
If attribute or reference links already exist on any part type for the populated conditions, the existing links are not altered by running the rule - only new links are added.

## Business Action: Set Import Status Attributes

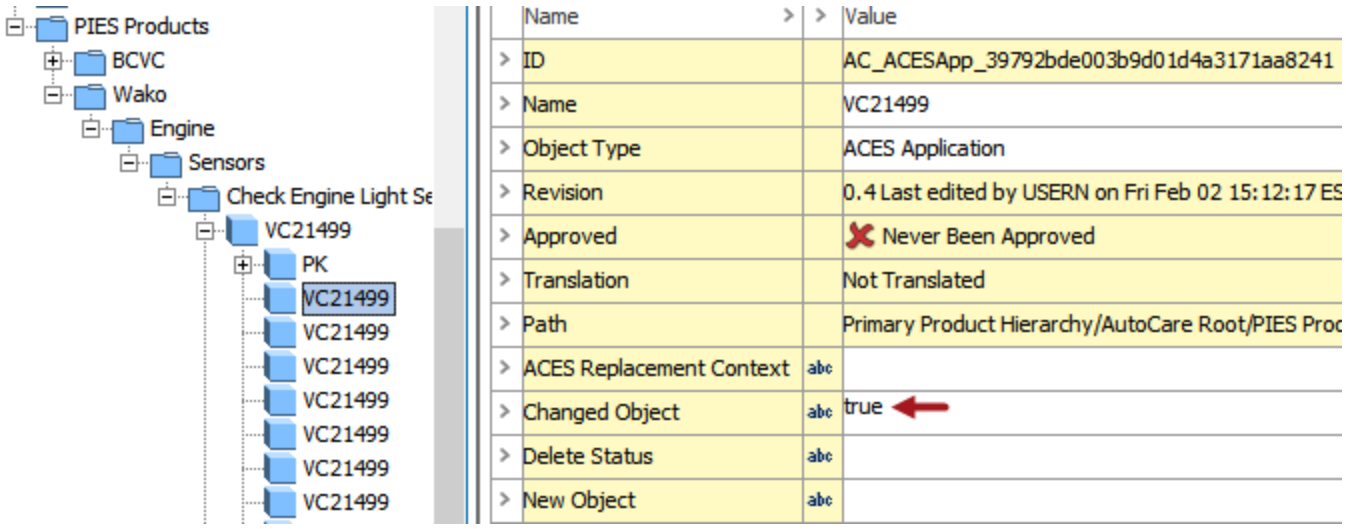
While not strictly required, it may be helpful to implement change flags; a way for users to view what data has been created or changed due to an import.

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.'




If the application had previously existed and had information changed during the import, then only the Changed Object attribute value would display 'true.'




**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

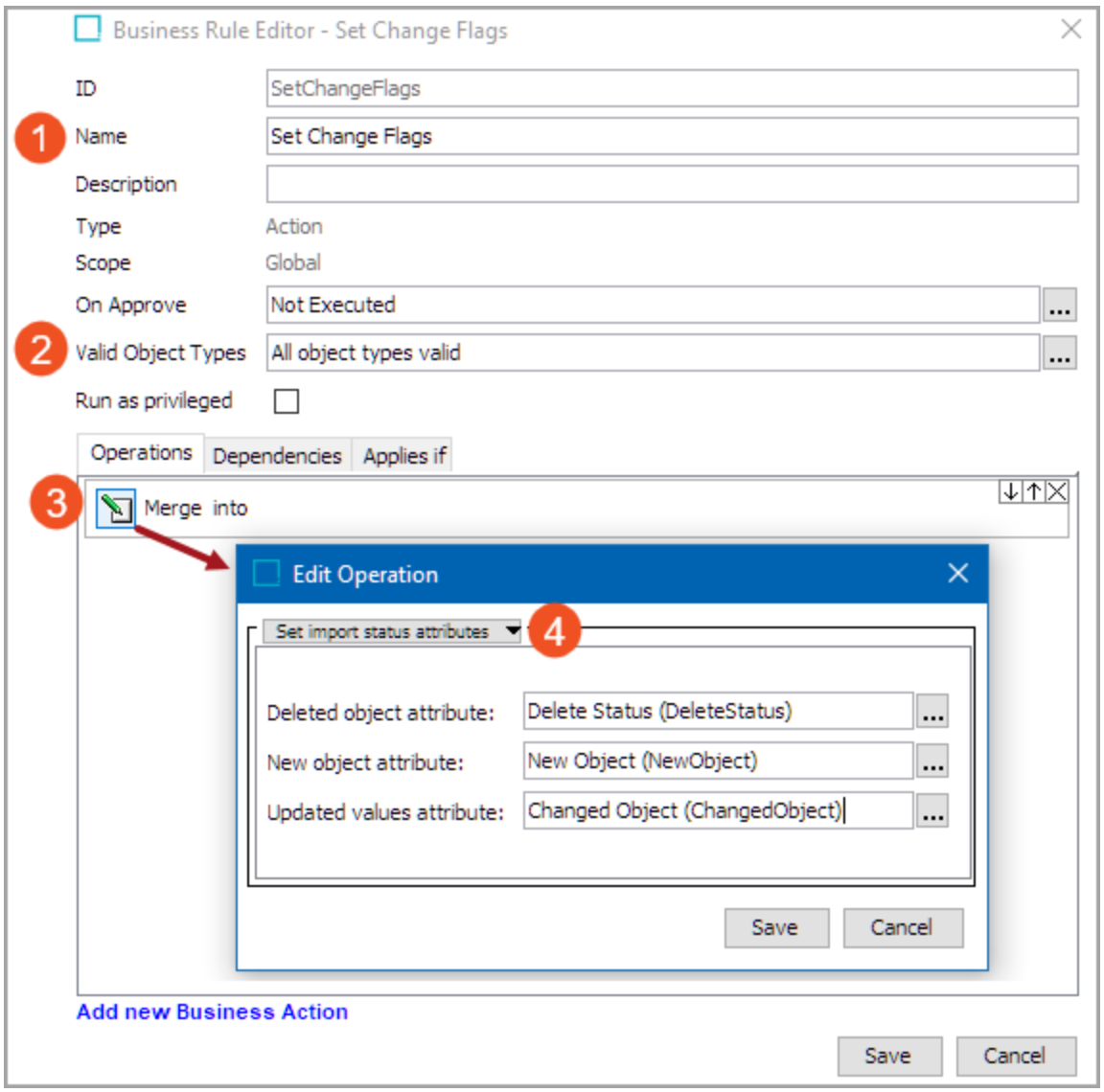
## Create / Identify Valid Attributes

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, refer to topic **Use Cases - Clearing Delete Status Flag**.

6. Click the **Save** buttons to save and close the business rule.

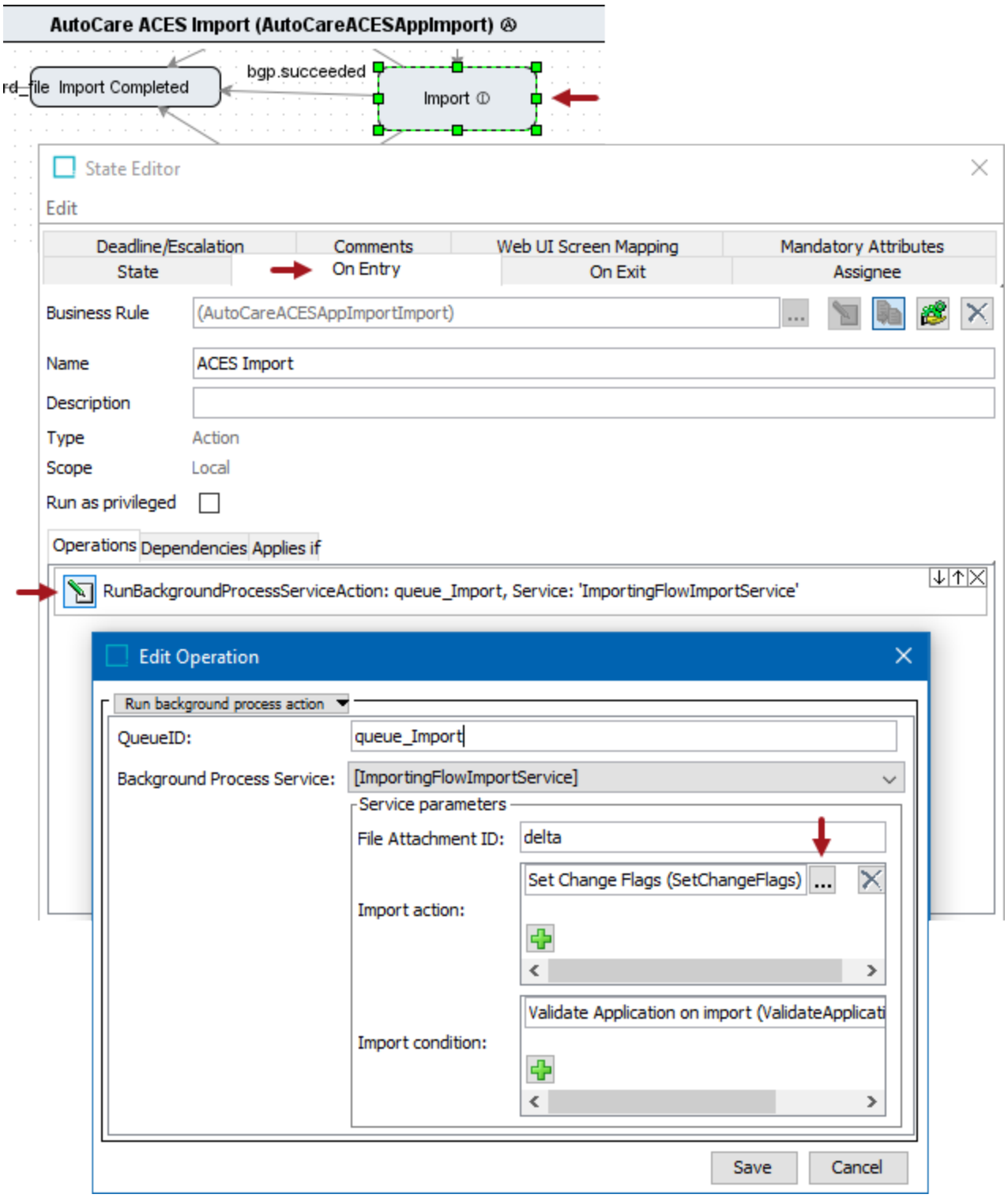
   For more information on creating and/or editing business rules, refer to the **Creating a Business Rule, Function, or Library** topic, and the **Editing a Business Rule or Function** topic within the **Business Rules** guide.

## 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, refer to topics **5. Update Delete Status Attribute and Delta Calculation Method in Import Workflows** and **Delta Calculation State** within the **Automotive Reference Guide**.

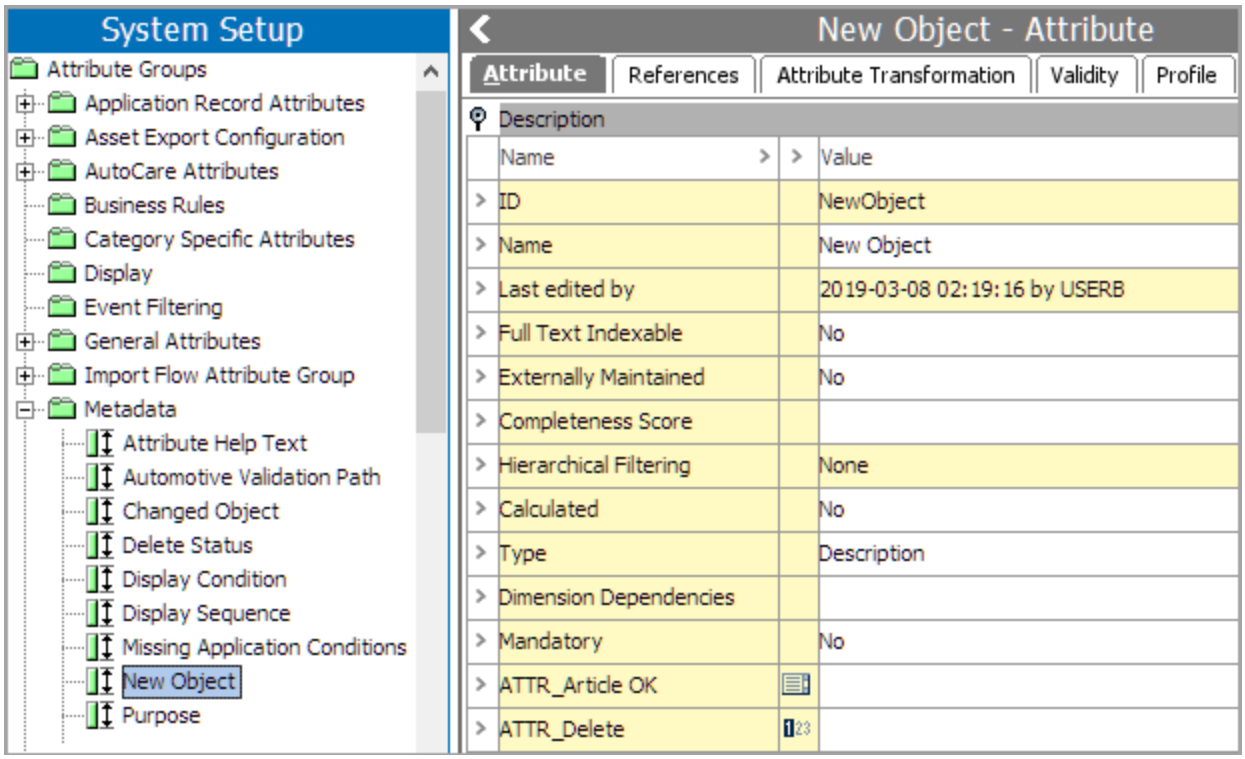
**Use Cases - Tracking Newly Created Objects**

The user is responsible for configuring the PMDM for Automotive system to keep track of all newly created objects on subsequent imports.

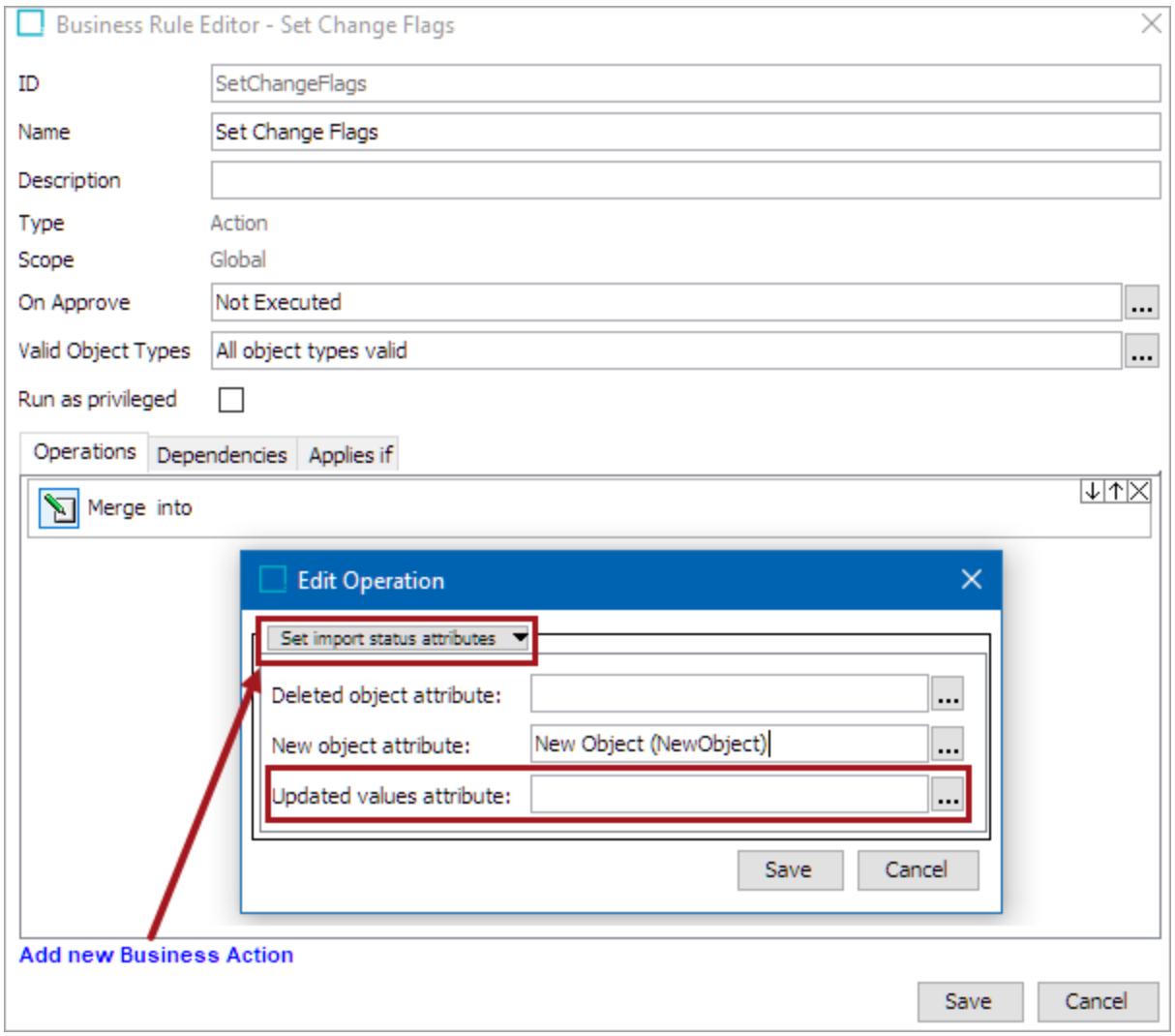
Consider an example that a new ACES file is imported into the system and all applications are stored in the system under their respective hierarchy. The next time the same file with an addition of few more applications is imported into the system, the newly created applications are tracked as defined below.

## Track new objects created during import

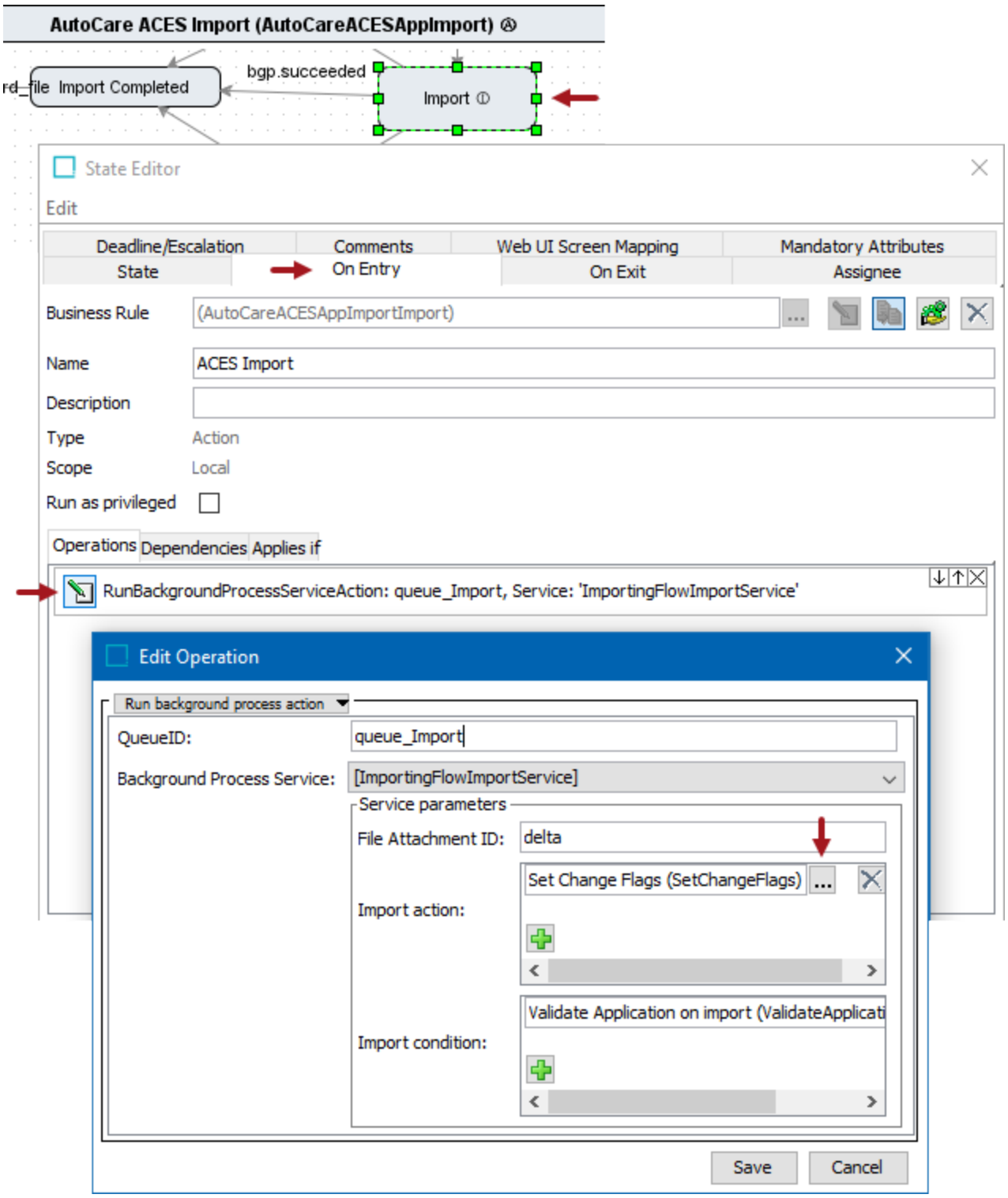
1. Create an attribute (or verify that one exists) with the validity set to the object type(s) that need to be tracked. In this example, an attribute 'New Object' is created with the validity set to object type ACES Application.




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, refer to 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.'

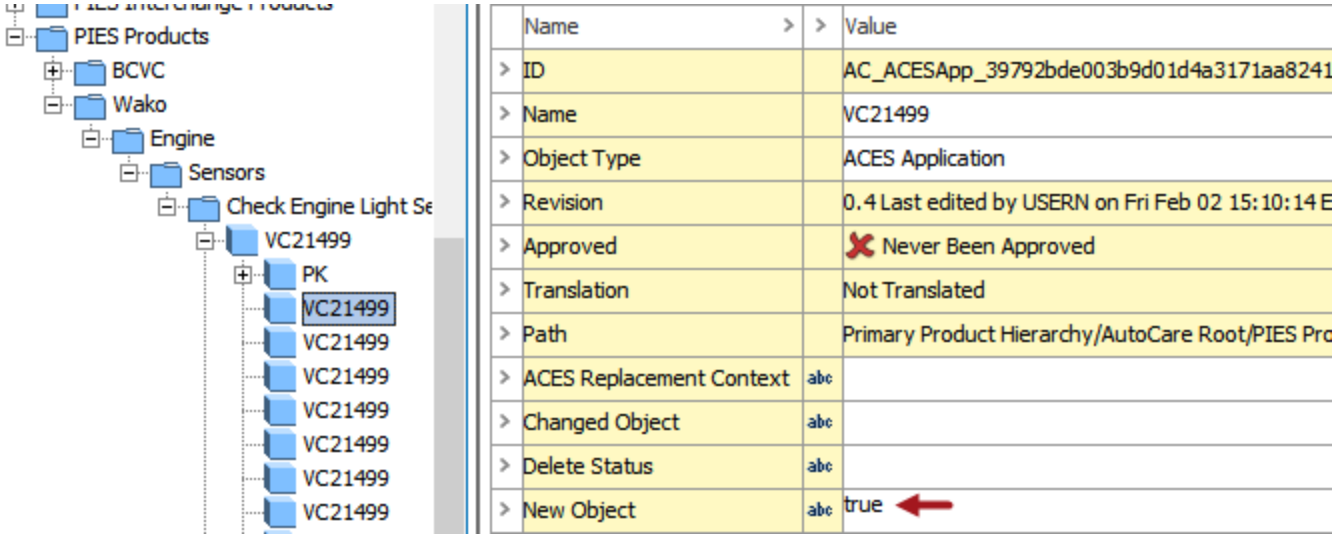



3. Configure the **Import** state of the workflow with the updated business action discussed in the previous step. For detailed procedures on updating the workflow settings, refer to the topic **Business Action: Set Import Status Attributes**. In the example below, the Import state in the AutoCare ACES Import workflow is configured with Set Change Flags business action.




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.'



**Use Cases - Tracking Deleted Objects**

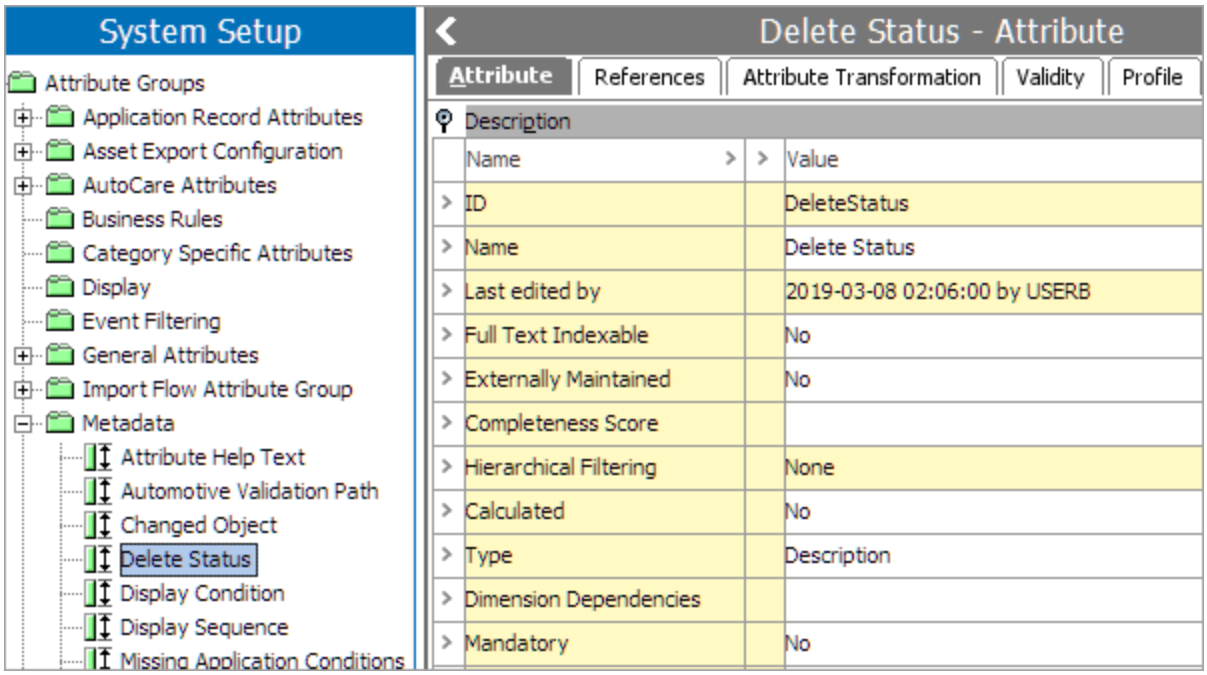
The user is responsible for configuring STEP to keep track of all the deleted objects on subsequent imports and handle the deleted objects manually in the system.

**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.

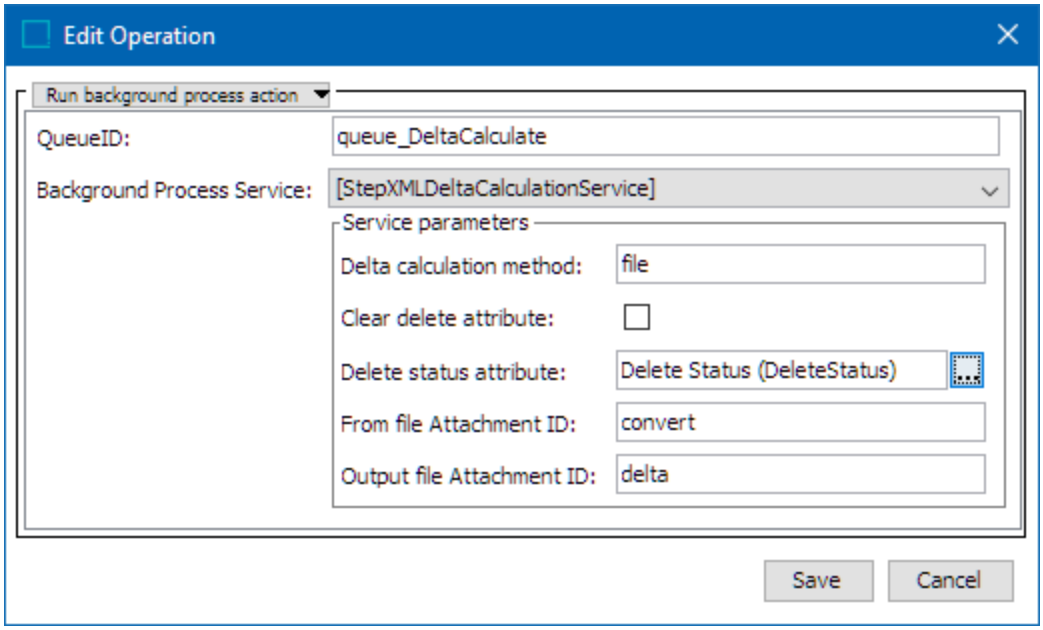
Consider an example that a file containing the Brand details is imported into the system and all Brands including their additional information are stored in the system under their respective hierarchy. The next time the same type of file is imported with some brand name deleted, the deleted Brand names 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.

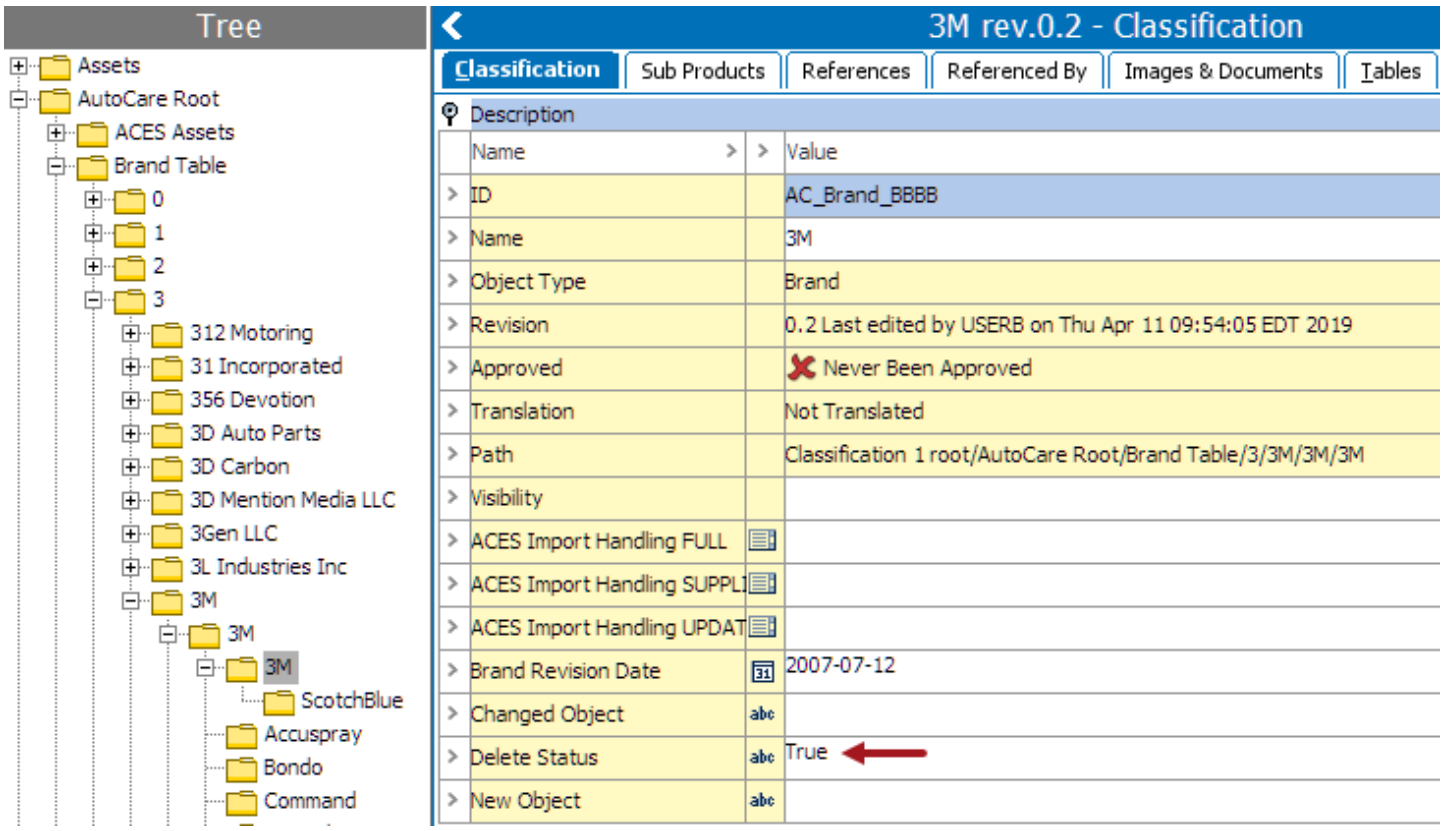



2. Within the Delta Calculation state of the workflow, configure the newly created attribute within the 'Delete status attribute' parameter. For more information on configuring the attribute in the workflow, refer to the topic **5. Update Delete Status Attribute and Delta Calculation Method in Import 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.




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 verify 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
Delete Status
True
New Object

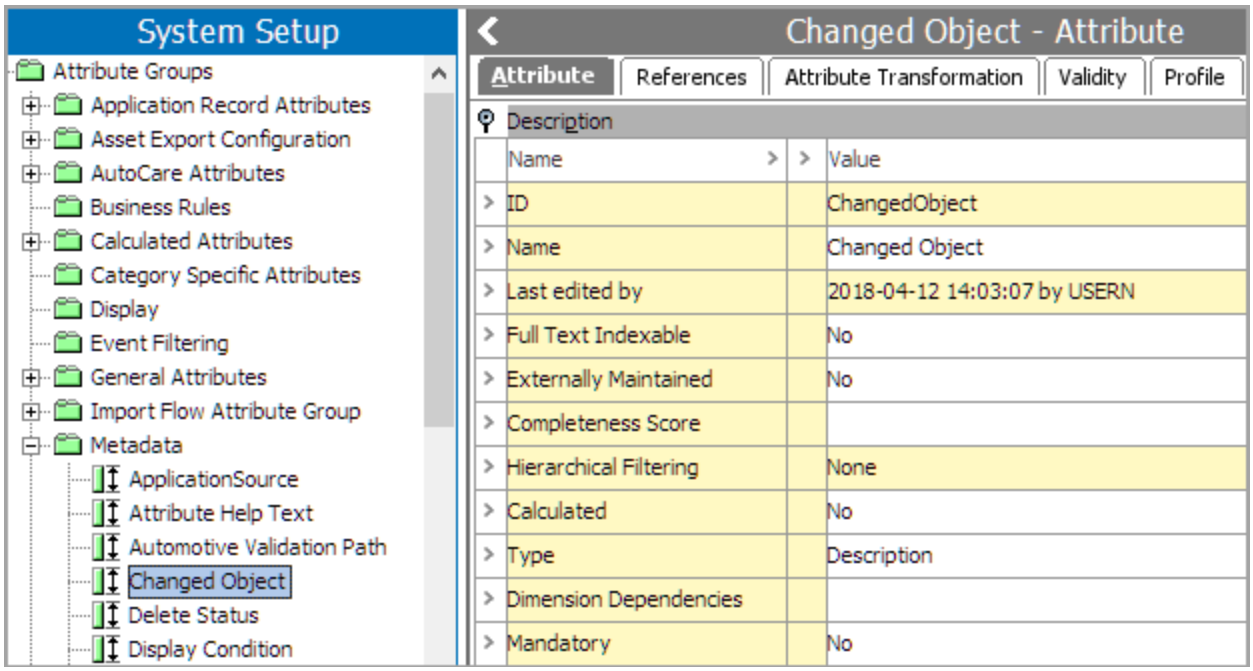
**Use Cases - Tracking Changed Objects**

The user is responsible for configuring the STEP to keep track of all changed objects on subsequent imports.

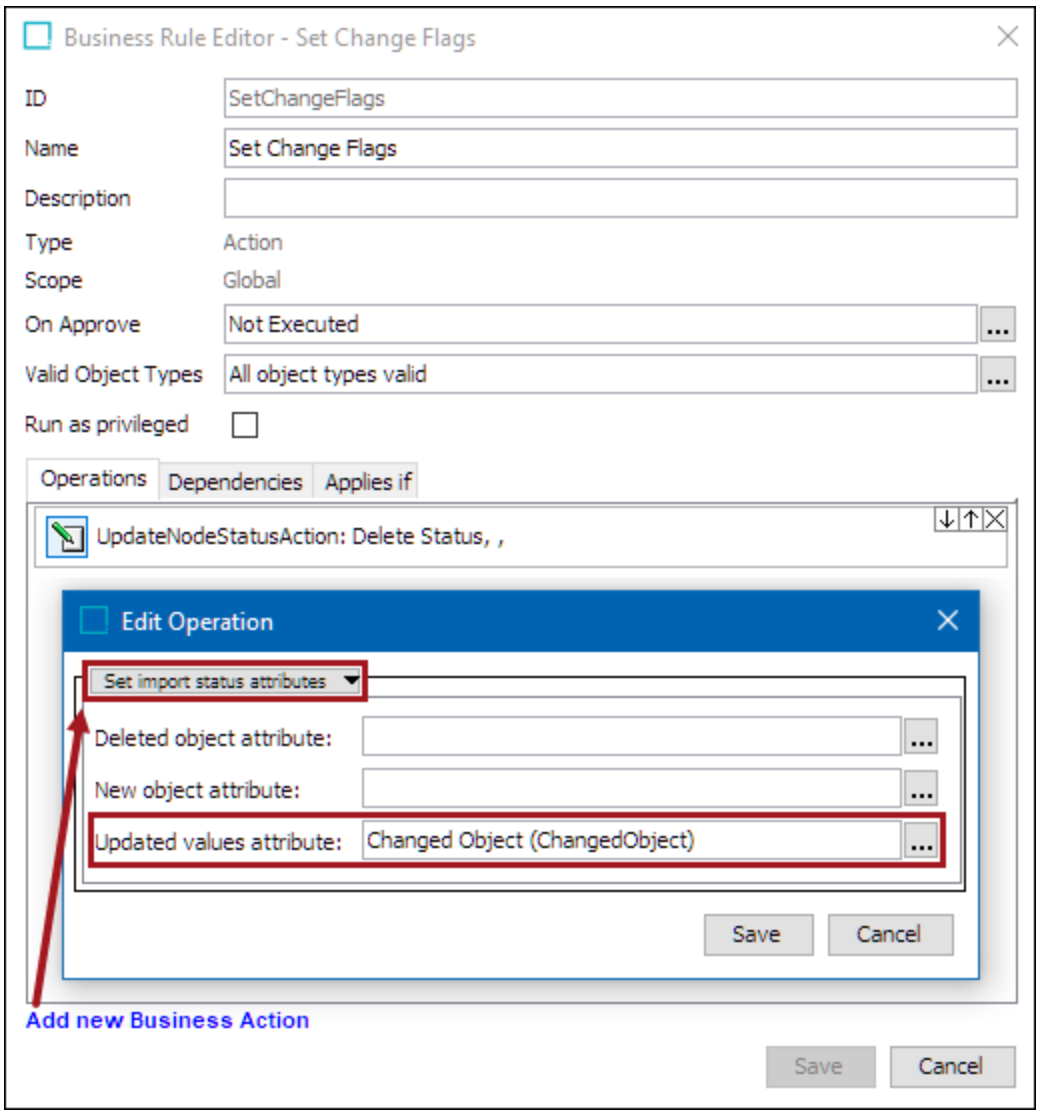
Consider an example that a file containing the Brand details is imported into the system and all Brands including their additional information are stored in the system under their respective hierarchy. The next time the same file with any changes in the existing Brands are imported into the system, the Brands with changes are tracked as defined below.

## Track changed objects created during import

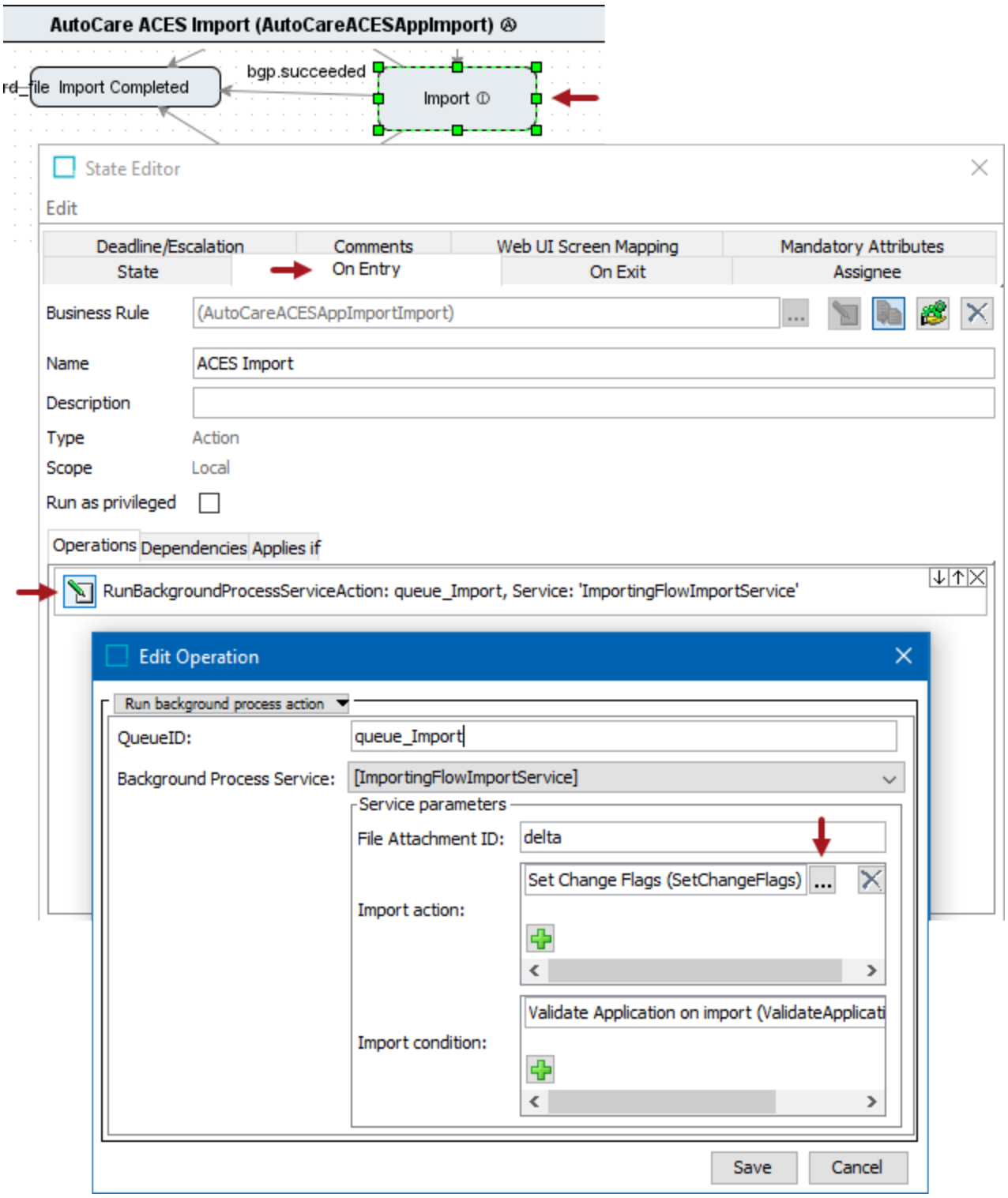
1. Create an attribute (or verify that one exists) with the validity set to the object type(s) that need to be tracked. In this example, an attribute 'Changed Object' is created with the validity set to object type Brand.




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, refer to 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.'

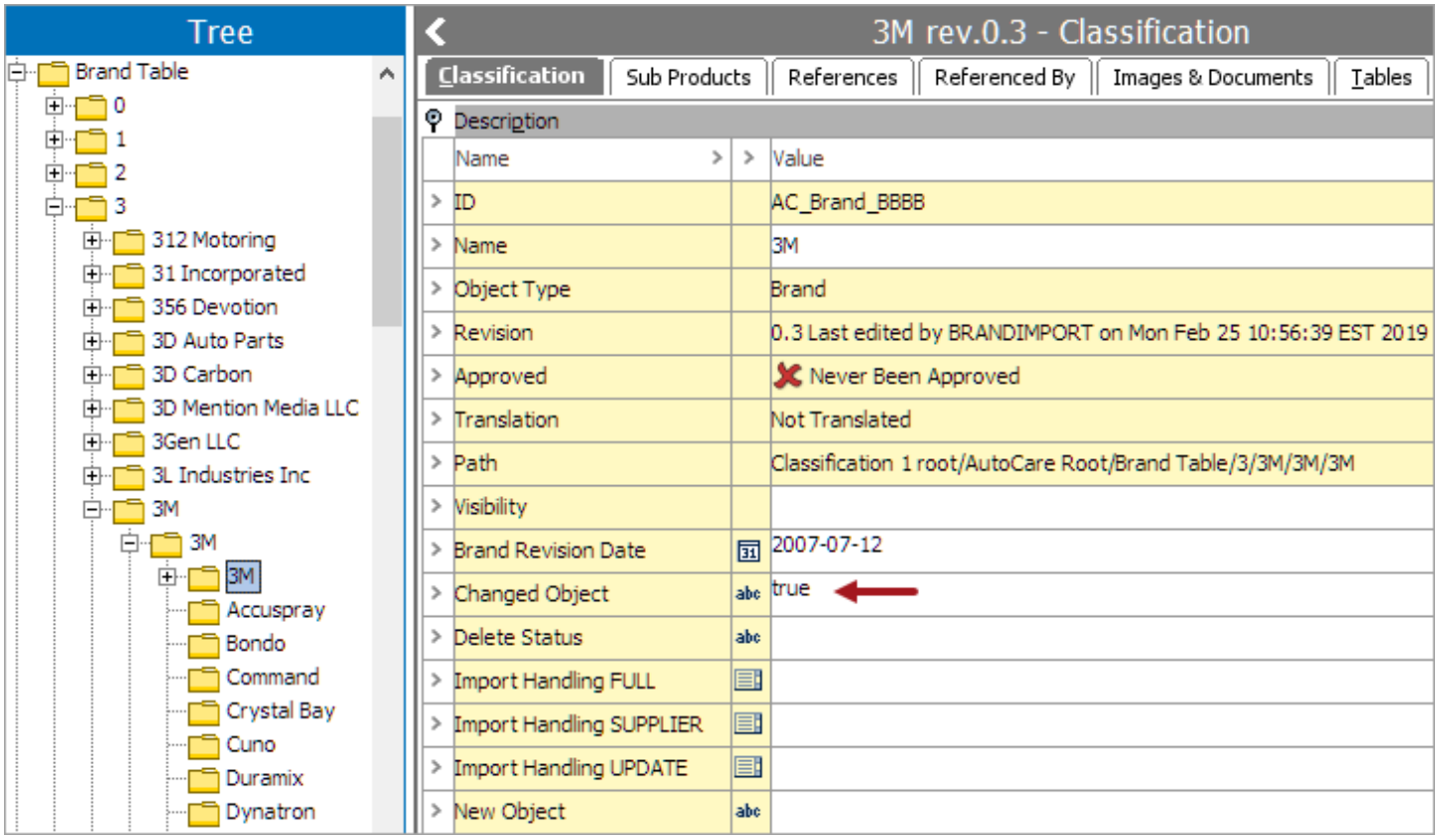



3. Configure the **Import** state of the workflow with the updated business action discussed in the previous step. For detailed procedures on updating the workflow settings, refer to the topic **Business Action: Set Import Status Attributes**. In the example below, the Import state in the AutoCare Brand Import workflow is configured with Set Change Flags business action.




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.'



**Use Cases - Clearing Delete Status Flag**

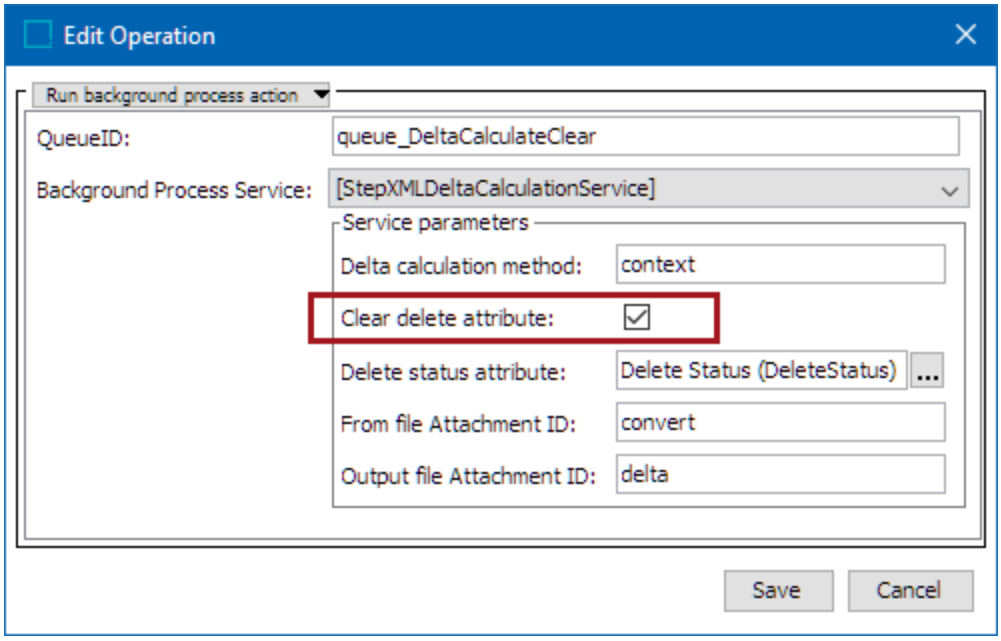
When an object marked for deletion is resurrected in a later import, it is necessary that the previously set delete status flags are erased and the latest status information of an object is displayed in the system.

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.

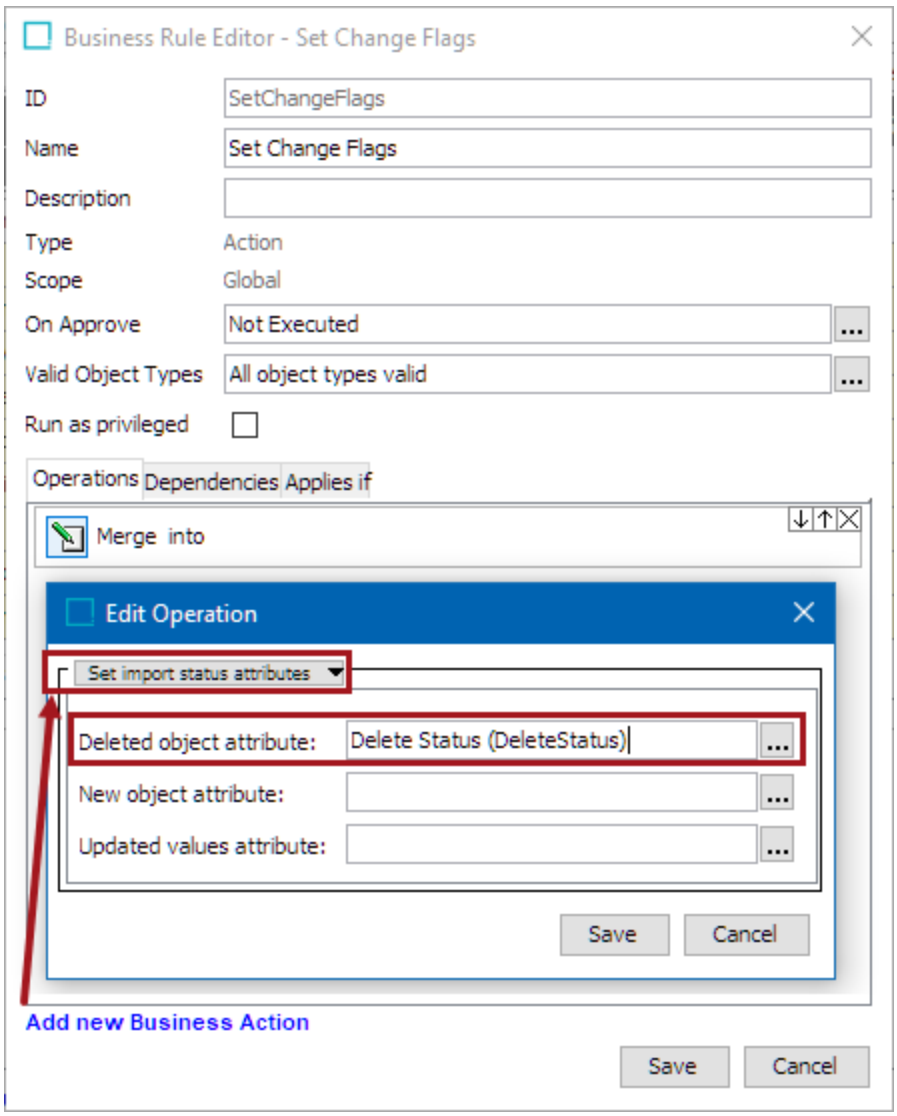
## Method 1: Using 'Clear delete attribute' parameter in Delta Calculation state

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, refer to the topic **Delta Calculation State**.

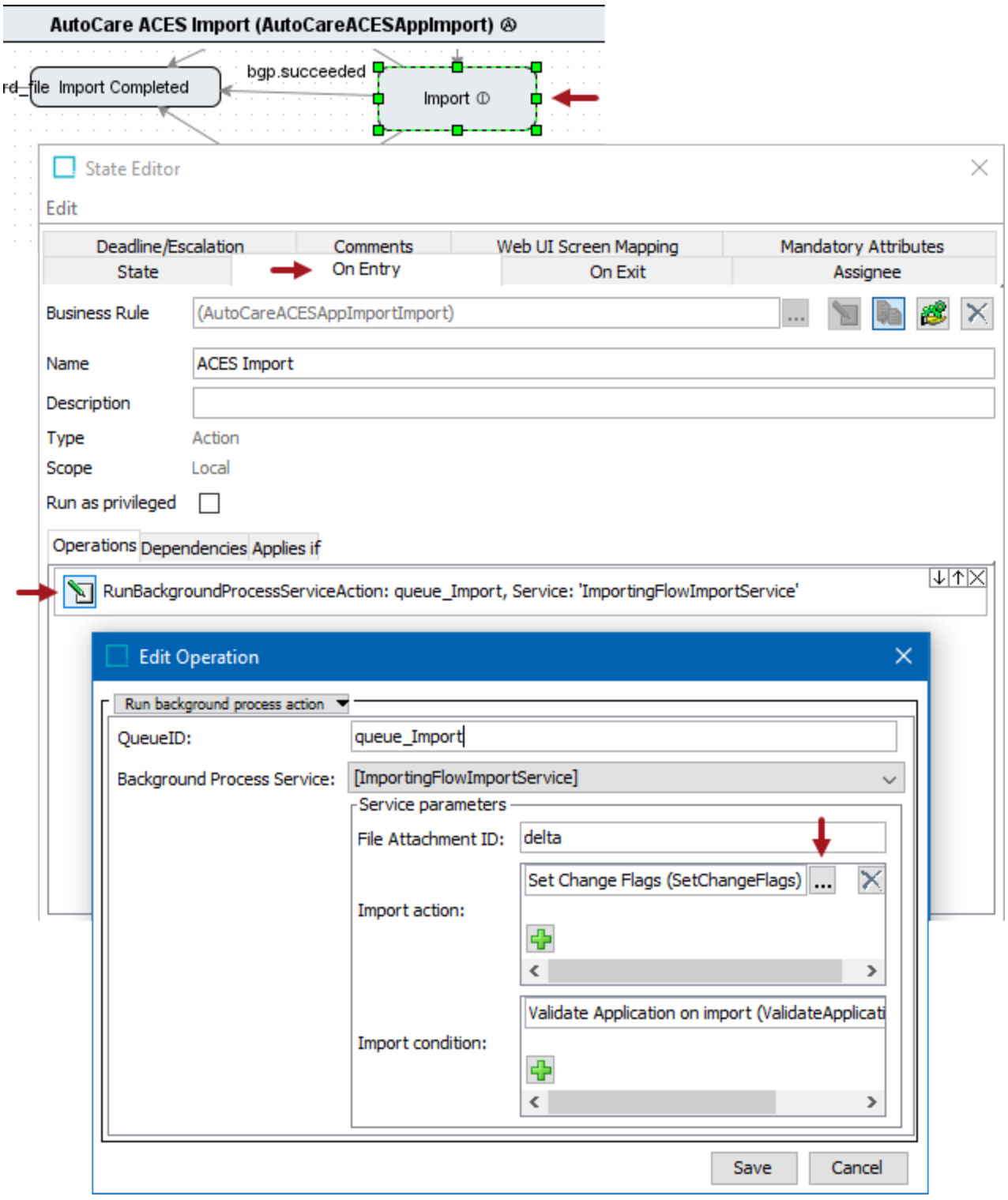



## Method 2: Using 'Set import status attributes' Operation

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, refer to 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, refer to 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)
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

## 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.

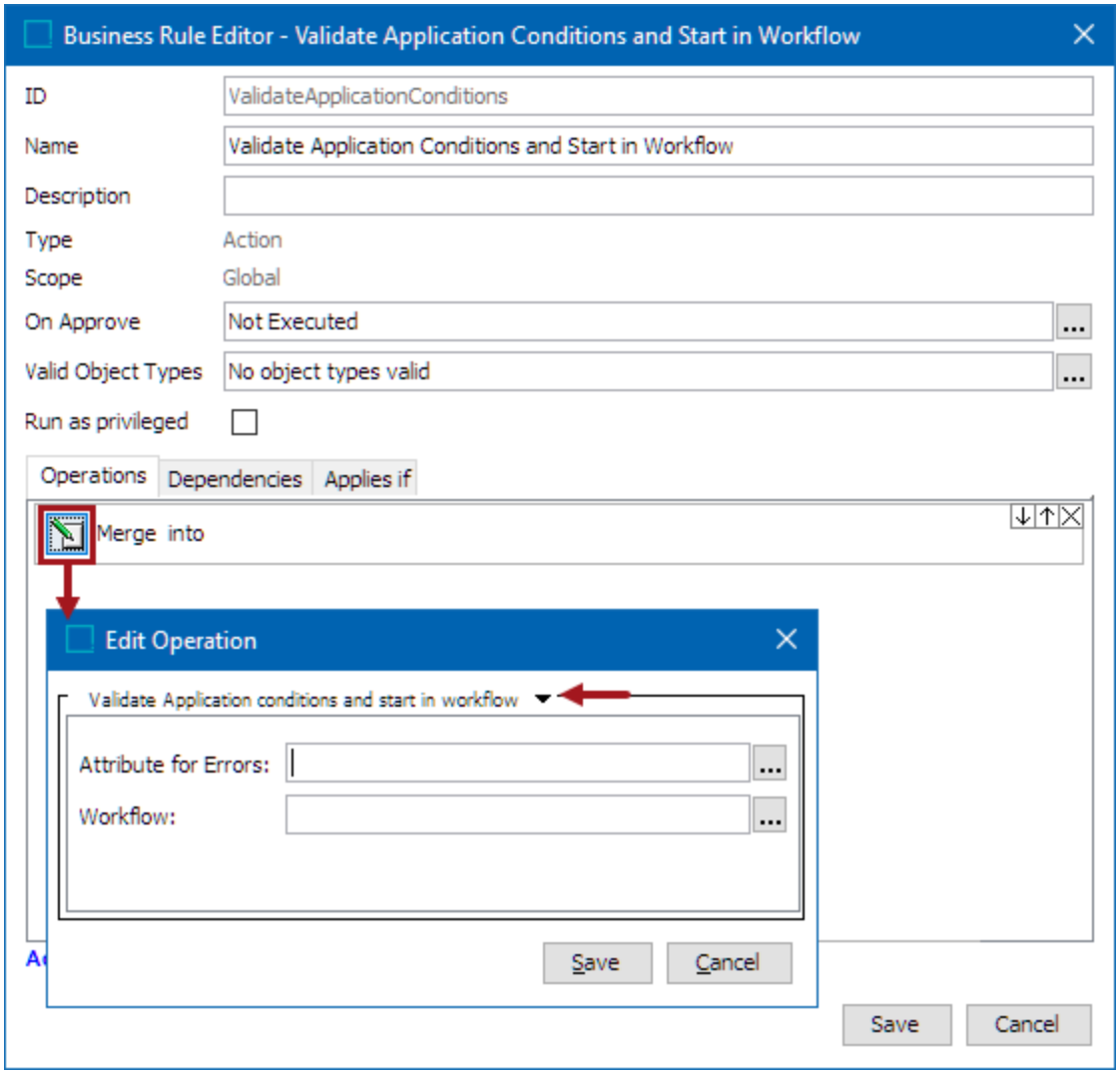
The action evaluates all ACES applications that are child to the selected part on which the rule is being executed, and the applications on the target(s) of the reference indicated in the business rule configuration. The records are then synchronized between the parts, excluding duplicates. For example, 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. Setup is required in the STEP Workbench for the action to be available to users.




Business action with the 'Validate Application conditions and start in workflow' operation can check for applications with invalid vehicle configurations. When an application record is detected that does not meet the condition, an error is written to the execution report of the import process, and the invalid application record is imported and then initiated into the workflow configured within the '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 the business action 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.

**Note:** How the customer handles such invalid applications in a different workflow is beyond the scope of this topic. It is a customized solution that the SC can implement on to their system.

### 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.

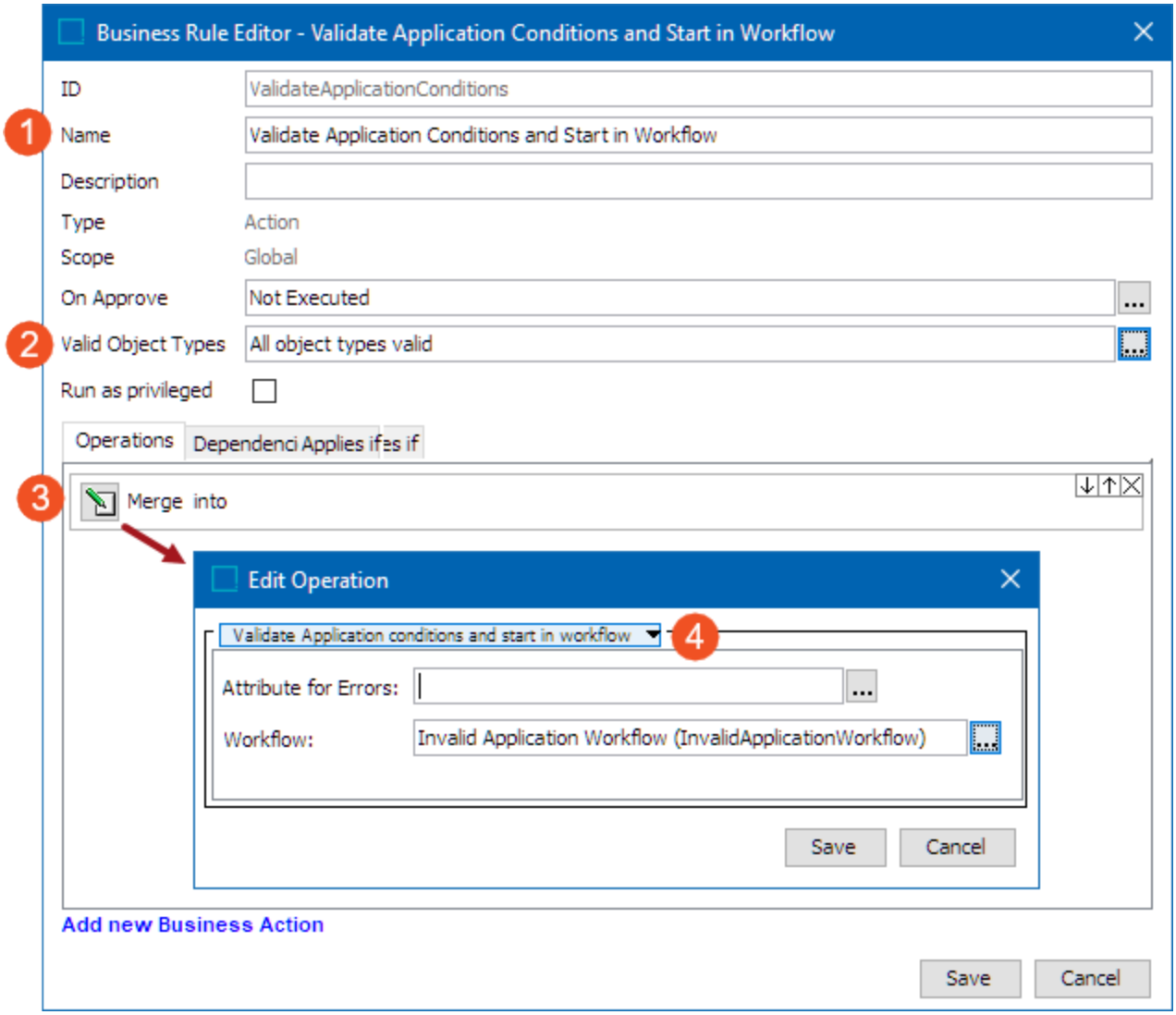
   For example:

   - 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 be displayed. Configure the parameter to use the necessary workflow.
   - **Attribute for Errors:** The 'Attribute for Errors' is an optional parameter that writes all the vehicle validation error messages in the attribute when configured with an attribute valid for the application object type. Step 1 of the Automotive - AutoCare Model Easy Setup creates the 'AC_ACESErrors' attribute below the Metadata attribute group and makes it valid for the ACES Applications object type. Click the ellipsis button (...) to search and select the 'AC_ACESErrors' attribute. Configuring the 'AC_ACESErrors' attribute in the parameter writes all the vehicle validation error messages of the faulty application record in the attribute.

- **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 for 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 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 how to handle these invalid applications in another workflow.

6. Click the **Save** buttons to save and close the business rule.

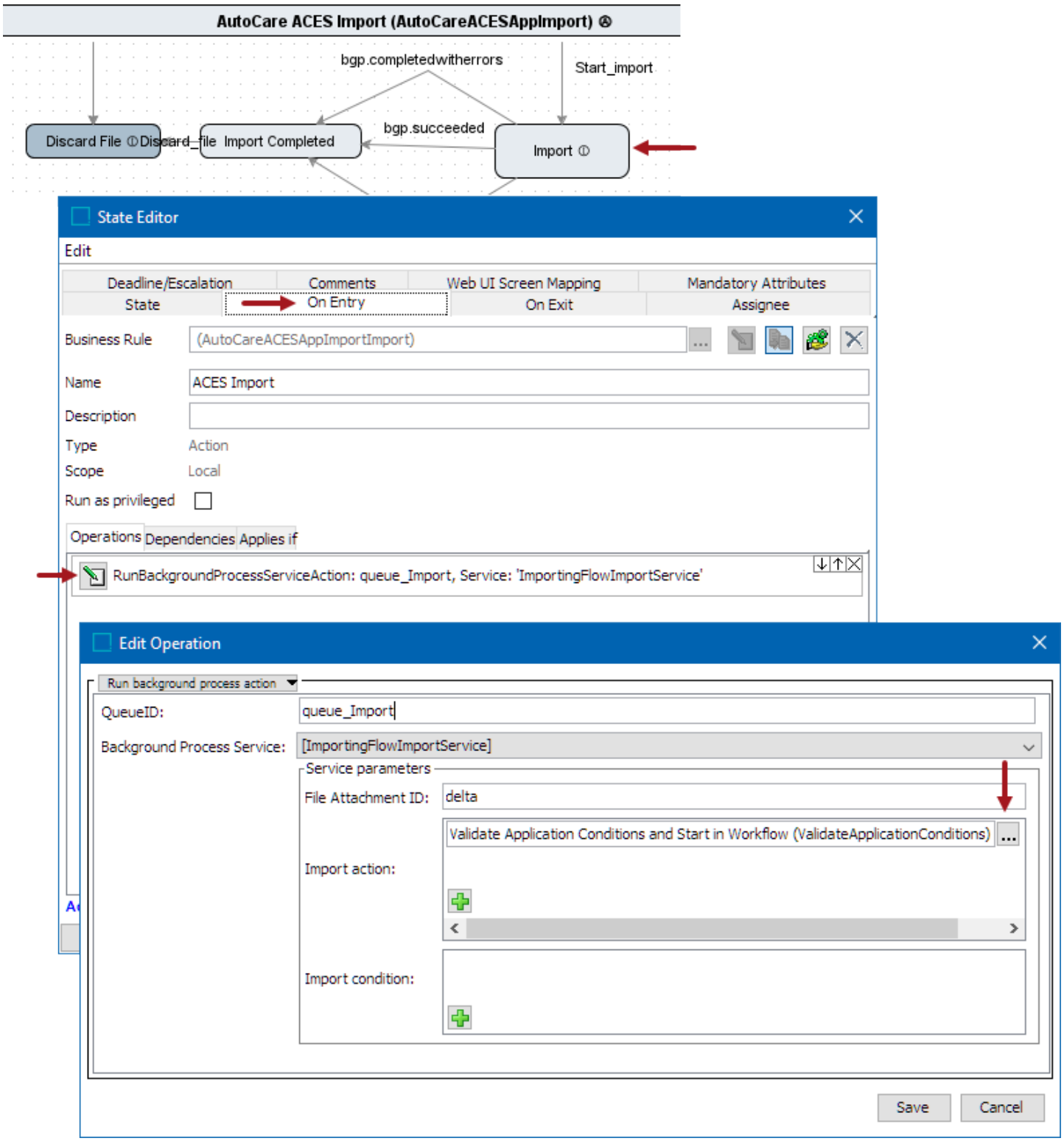
   For more information on creating and/or editing business rules, refer to the **Creating a Business Rule, Function, or Library** topic, and the **Editing a Business Rule or Function** topic within the **Business Rules** guide.

## 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).



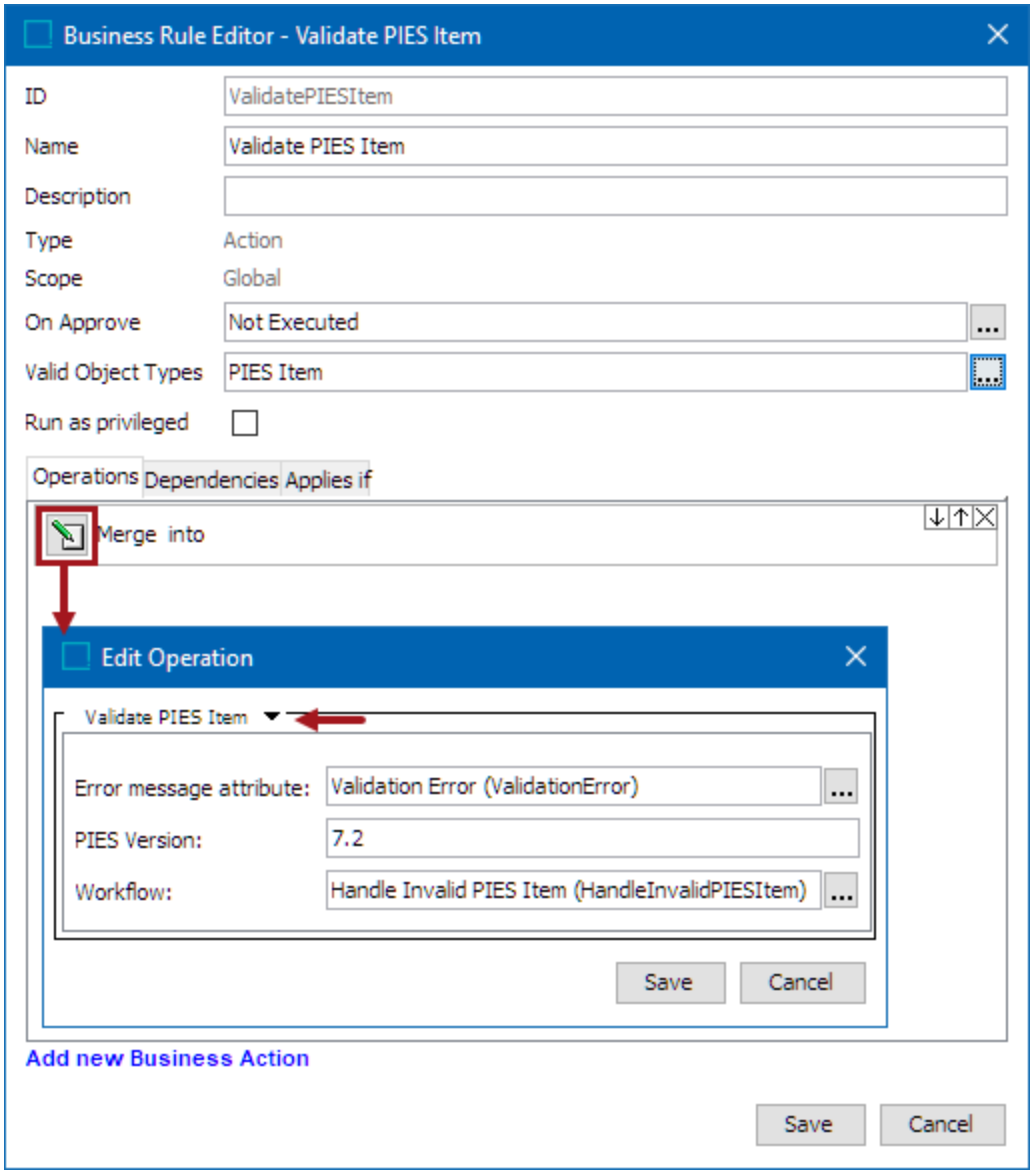

5. In the Edit Operation dialog, click the ellipsis button (...) to the right of the 'Import action' parameter to find and select the business action previously created.
6. Click the **Save** button and exit the workflow.

## Business Action: Validate PIES Item

The 'Validate PIES Item' operation is found within the STEP Workbench Business Rule Editor under the Automotive menu and applies only to Auto Care PIES Item objects. This business action is to be run on the PIES Item objects that exist in the system. Users can determine their own strategy to run this business action.

Setup is required in the STEP Workbench for the action to be available to users.

**Important:** This business action can ONLY be used to validate PIES Items that exist in the system and cannot be used on any other objects.



Running this business action on the selected PIES Items will run a validation check against the defined PIES version and will perform the following actions:

- Initiates the PIES Items with validation errors into the workflow defined in the Workflow parameter.
- Writes an error message into the attribute configured in the 'Error message attribute' parameter.

As a prerequisite, one unique workflow that can handle the invalid PIES Items is necessary. In all cases, when the business action decides that the invalid PIES Items should be handled separately, it routes those PIES Items to the workflow indicated in the business action configuration. This allows each customer to determine their own strategy for managing invalid PIES Items.

How the customer handles such invalid PIES Items in a different workflow is beyond the scope of this topic. It is a configurable solution that the solution consultant can implement on their system.

#### Configuring Business Action to Handle Invalid PIES Items

To properly configure this business action to handle invalid PIES Items, the following is needed:

- Create / Identify an attribute to store PIES Item errors
- Create / Identify a valid workflow that handles invalid PIES Items
- Configure a business action to use the 'Validate PIES Item' operation
- Determine a strategy to run the business action in the system

## Create / Identify a Valid Error Attribute

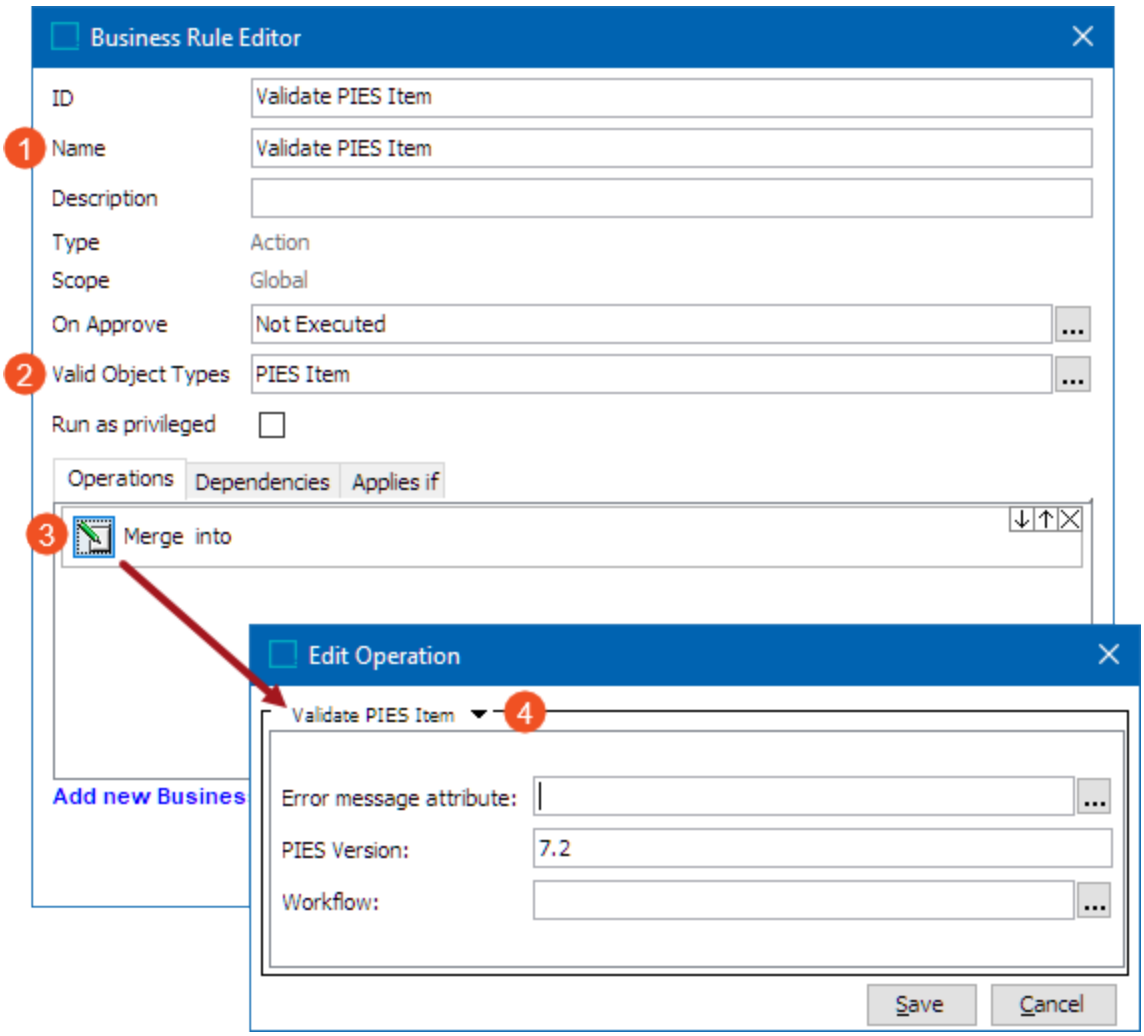
1. Create the necessary attribute (or verify that it exists) to store the PIES Item validation error(s).
2. The attribute must have Text Validation Base Type and have Multi Valued = Yes.
3. The attribute must be made valid for the object type PIES Item (ID = AC_PIESItem).

## Create / Identify a Valid Workflow

1. Create the necessary workflow (or verify that it exists) to handle the invalid PIES Items.
2. Make the workflow valid for the object type PIES Item (ID = AC_PIESItem).

## Configure a Business Action to Use the 'Validate PIES Item' Operation

1. Create a business action and give it a name that users can easily identify. In the example below, the 'Validate PIES Item' name is used.




2. Click the ellipsis button (...) to the right of the Valid Object Types parameter to find and select the object type PIES Item (ID = AC_PIESItem).
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 PIES Item**.
5. The following parameters will be displayed. Configure the parameters as described below:
   - **Error message attribute:** Click the ellipsis button (...) to define an attribute that holds the reason for the validation failure. The defined attribute must be valid and linked to the PIES Items that are being run through the validation check.
   - **PIES Version:** Enter the PIES version against which the selected PIES Items are to be validated. By default, version 7.2 is populated. Users can change it to run the validation against the following versions:
     - 6.5
     - 6.7
     - 7.0
     - 7.1

- 7.2

- **Workflow:** Click the ellipsis button (...) to find and select the workflow into which the PIES Items with validation errors are to be initiated. The configured workflow should be able to handle the invalid PIES Items (if any). The workflow defined in this parameter should be valid for the PIES Item object type.

**Note:** Upon configuration of the 'Validate Pies Item' business action, the Automotive solution only provides an option to identify invalid PIES Items and start them into a workflow. Users need to determine their own strategy for how to handle these invalid PIES Items within the workflow.

6. Click the **Save** button to save and close the business rule.

   For more information on creating and/or editing business rules, refer to the **Creating a Business Rule, Function, or Library** topic, and the **Editing a Business Rule or Function** topic within the **Business Rules** guide.

## Using 'Validate PIES Item' Business Action

Once the business rule is configured, it is up to the user to determine when and how to use it. Users can implement an effective strategy defined in the **Using Business Rules in STEP** topic that is available within the **Business Rules** section of **STEP Online Help**.

## 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.

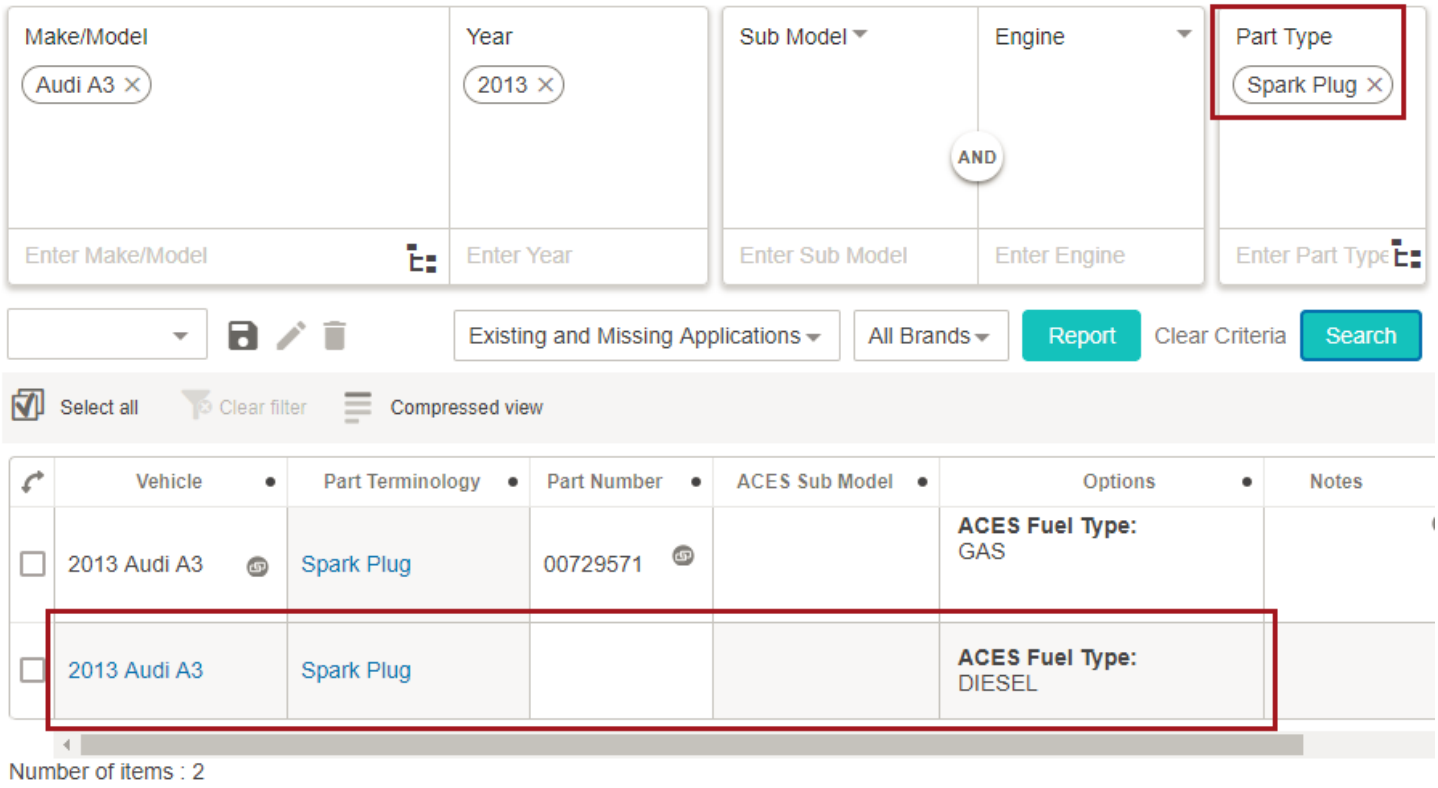
When Easy Setup actions are used to create the Missing Application Conditions attribute (MissingApplicationConditions), the attribute is also made valid for NAPA MPCC, Part Terminology, and Standard Assembly GA object types. This validity must be applied manually if the attribute was created manually.

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.




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.

Additional examples for part types that can have the Missing Application Conditions applied are for Coils or Ignition Module part types that are only valid for **Gas** Fuel Type but not valid for Diesel. Or for Part Types that are valid for **Gas** Fuel Type but not valid for Electric vehicles such as Oxygen Sensors, Catalytic Converters, and Mufflers. While Glow Plug part type is valid only for **Diesel** Fuel Type but not for Gas or Electric.

For more information, refer to the **Missing Application Coverage Functionality** topic within this guide.

For configuration instructions, refer to the **Configuring the Check Path for Missing Application in Workbench** topic of this guide.

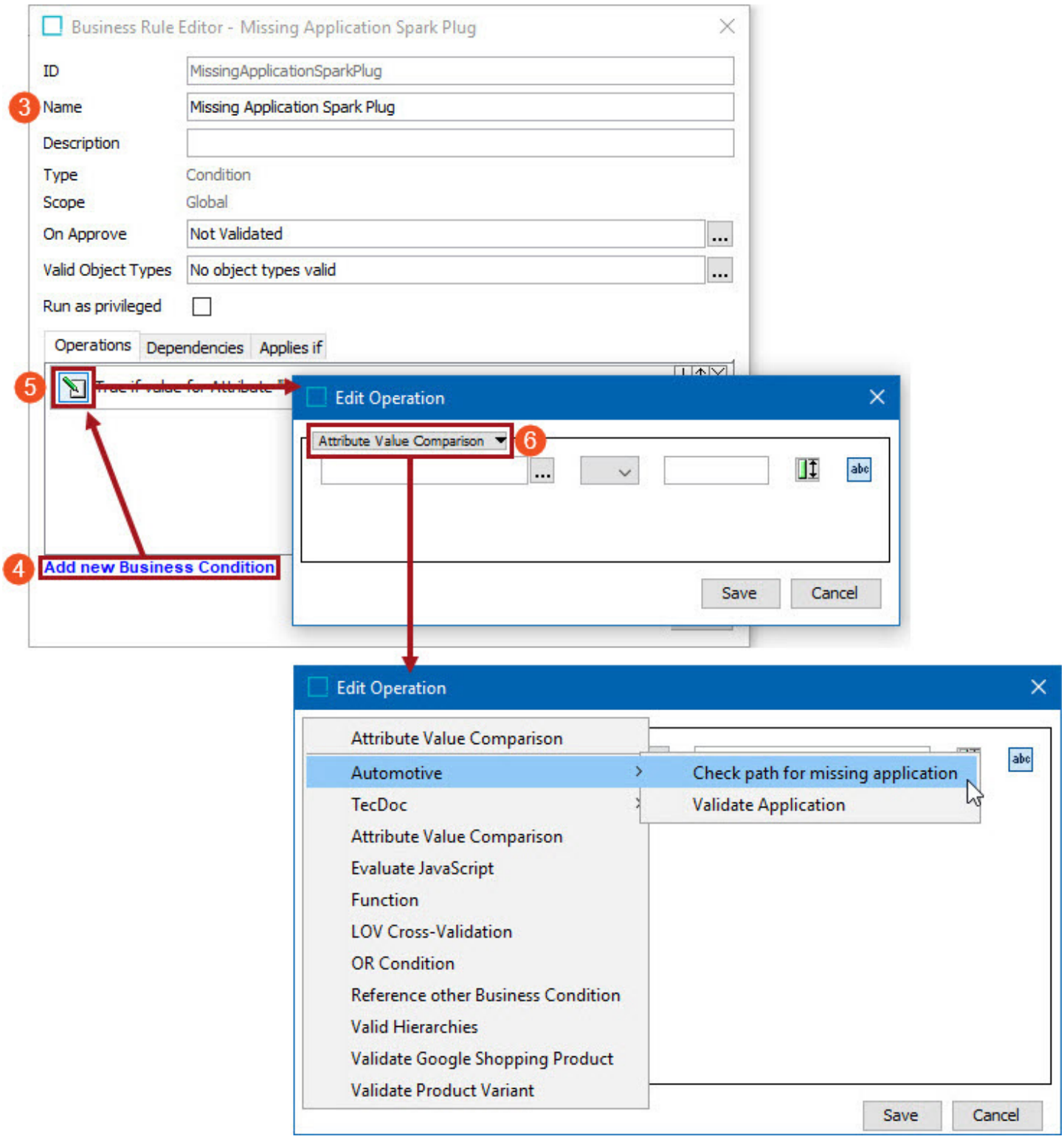
### Configuring the Check Path for Missing Application Business Condition in Workbench

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.

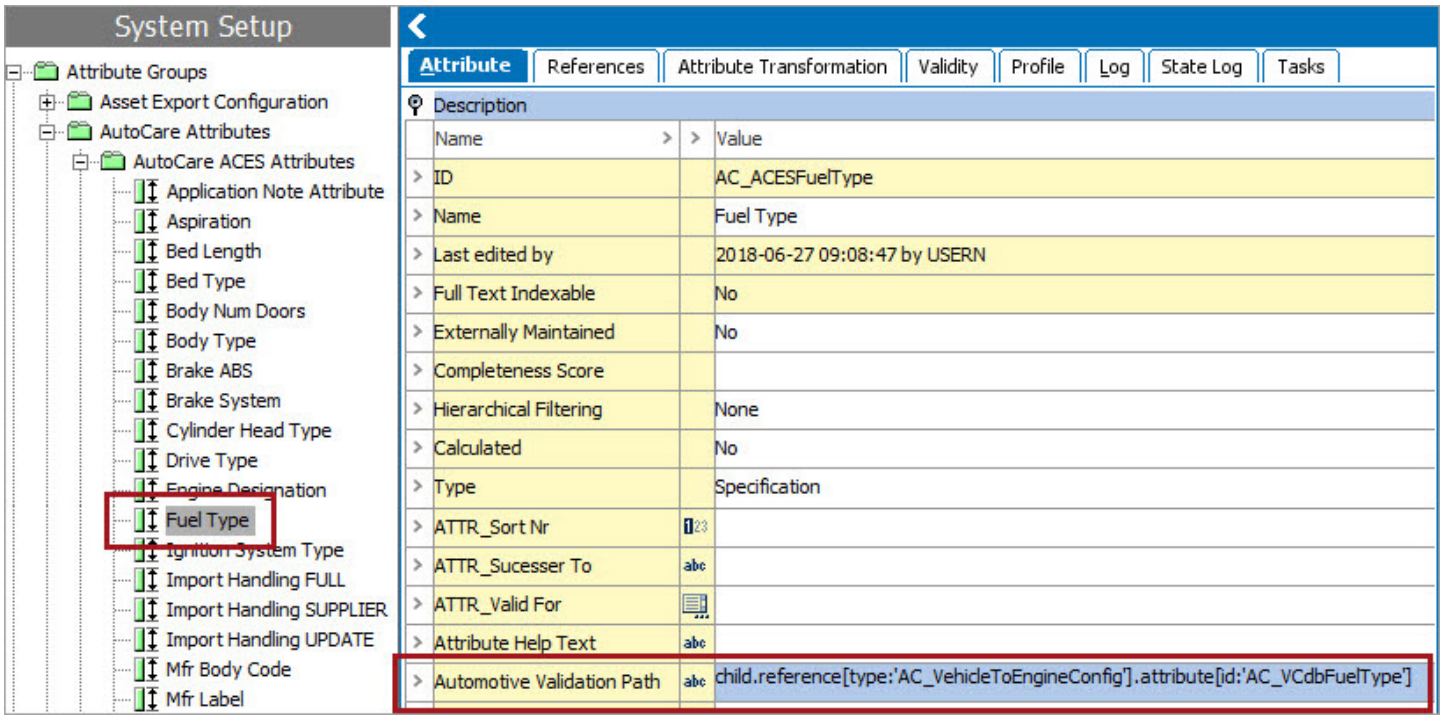
1. Confirm the 'Missing Application Conditions' attribute (MissingApplicationConditions) exists, otherwise create it.
2. Confirm the Missing Application Conditions attribute (MissingApplicationConditions) is valid for the necessary object types for each standard being used. For example, when Easy Setup 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).



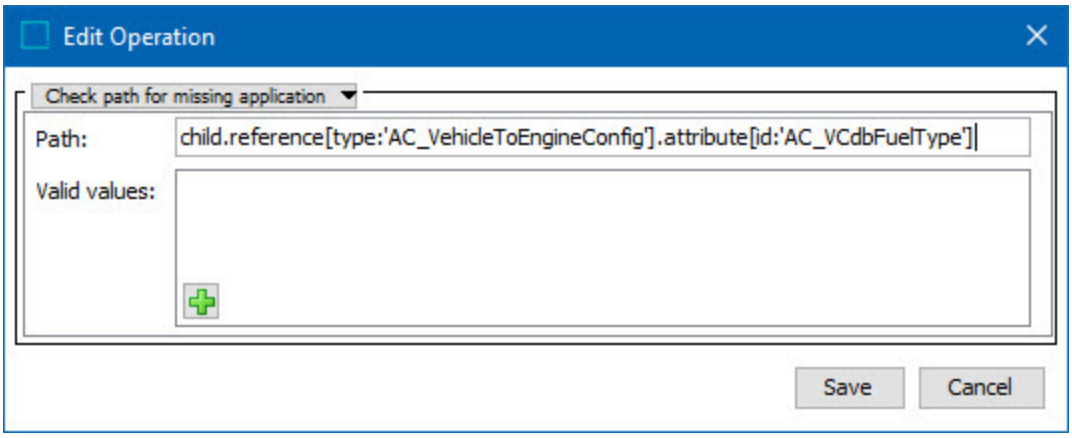

7. Once the plugin is selected, the Edit Operation dialog will display the following parameters:

- **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).

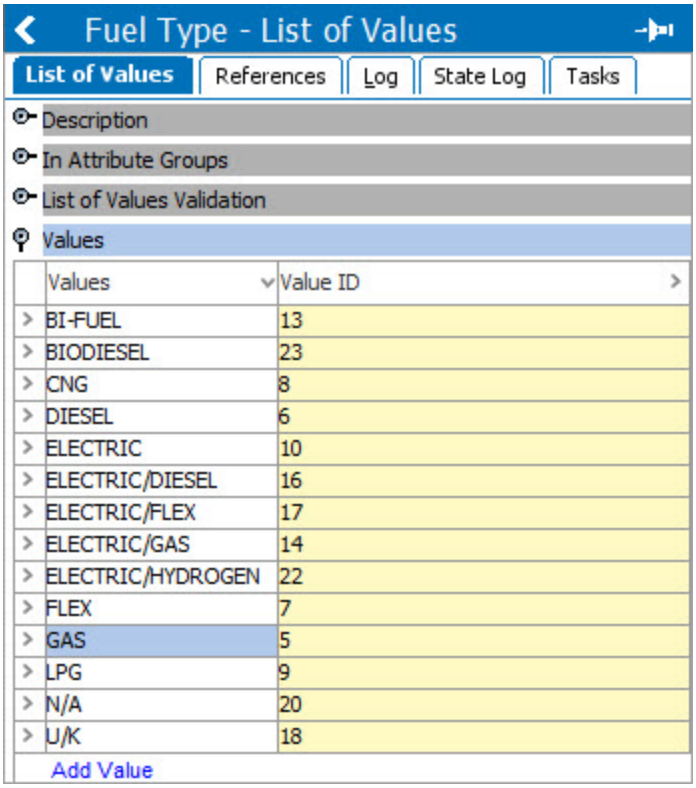
**Note:** For more information on the Automotive Validation Path, refer to the Automotive Validation Path Functionality section within the **Automotive Reference Guide** found within the **Solution Enablement** section of **STEP Online Help**.



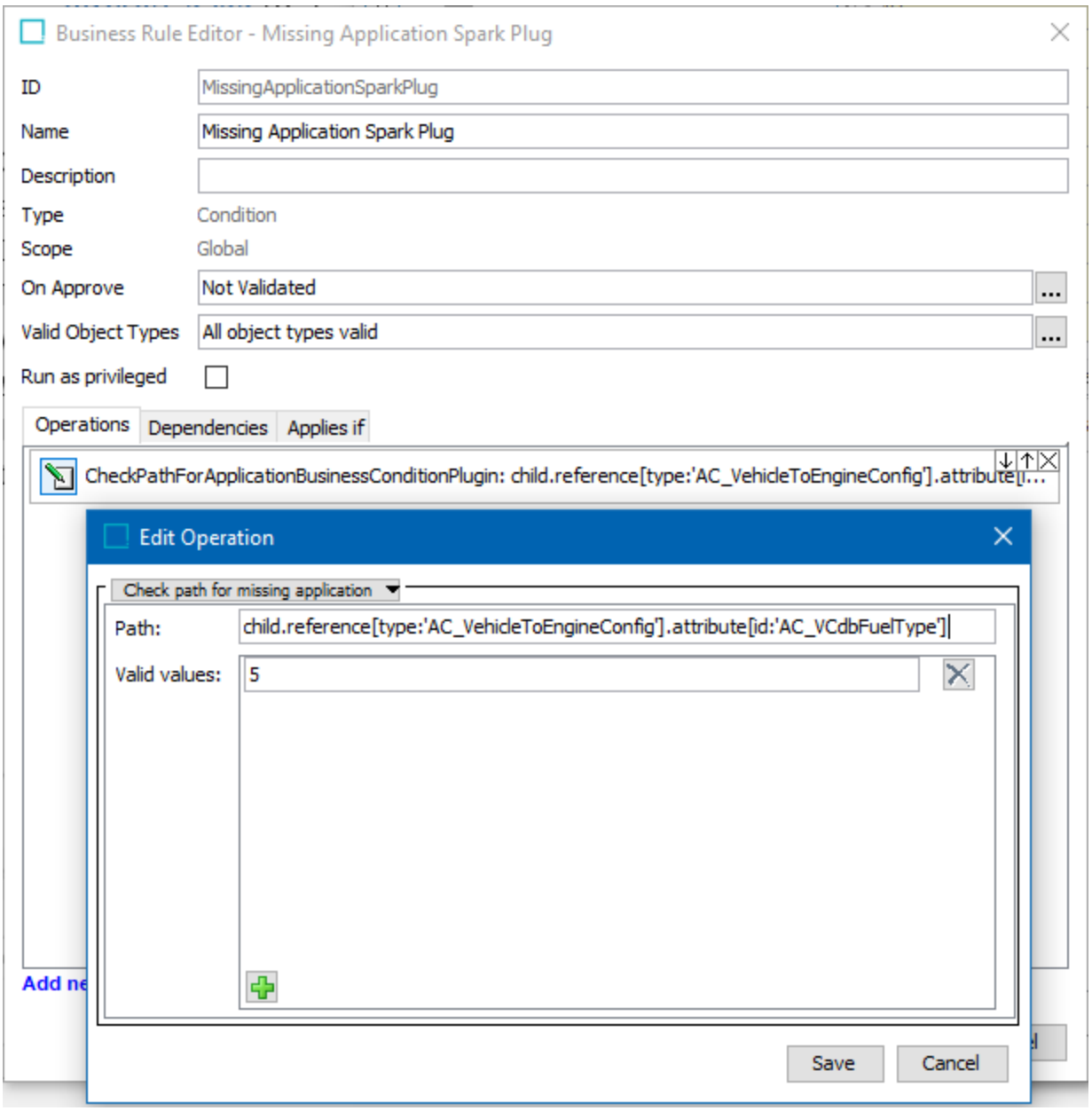

The Path information (attribute value) must be copied from the Fuel Type specification Automotive Validation Path attribute, and pasted into the Path parameter (as shown below).



- **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 Values GAS is 5.




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.



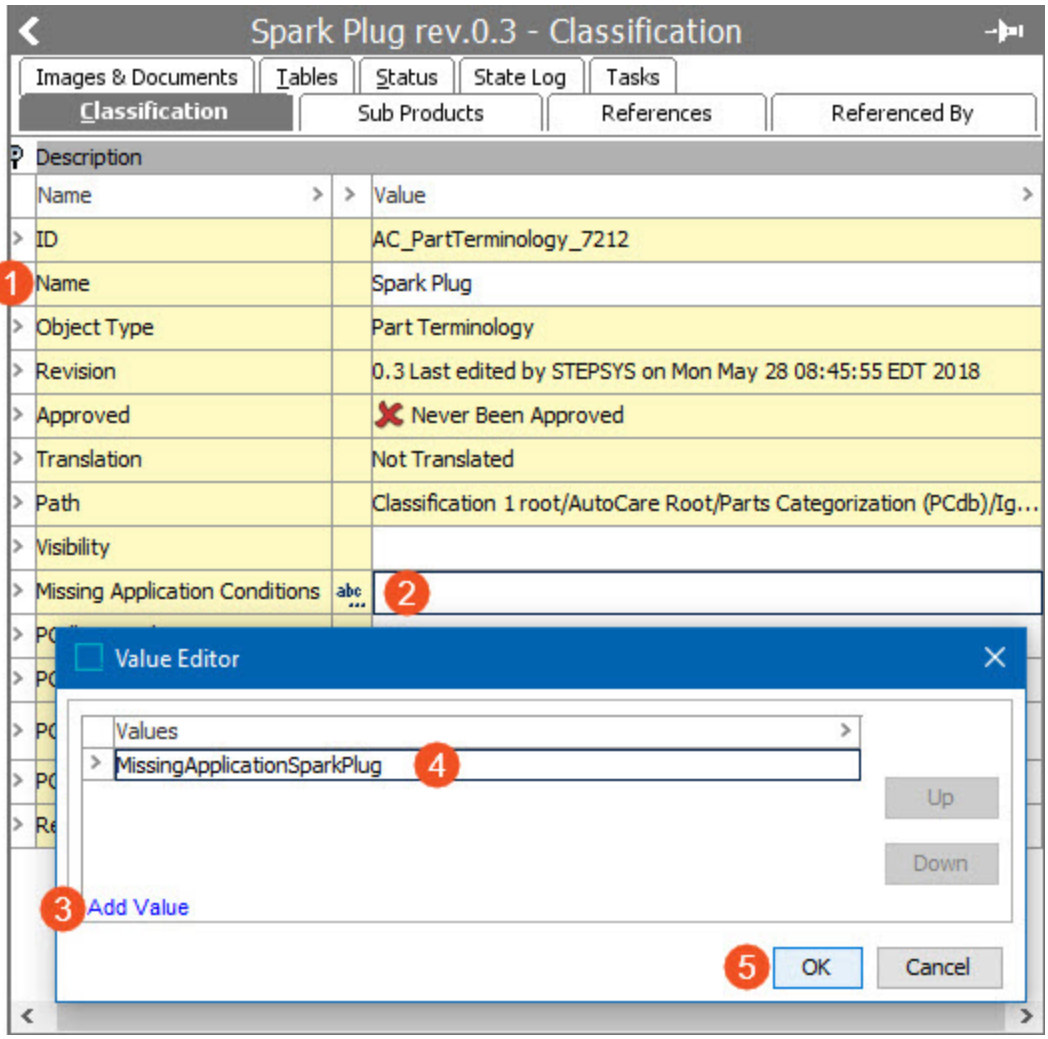

10. Optionally, click the **green plus sign** icon ( ) to add more value IDs, if more Fuel Types should display when the Spark Plug part type is searched.
11. Click the **Save** and **Close** buttons to save the Business Condition.

**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.




## Business Condition: TecDoc Reference Start Condition

The 'TecDoc reference start condition' operation is found within the STEP Workbench Business Rule Editor under the TecDoc menu and is used to prevent the start of a reference data import if there is a supplier data import running in the background process

It requires multiple parameters to be populated with values, and setup is required in the STEP Workbench for the business condition to be available to users. Once this business condition is created, users need to add this as the Start Condition in the TecDoc Reference import workflow. This business condition will not be created by an Easy Setup and must be manually created and configured by a user. If this business condition rejects an import, no error will be written / displayed on the import list screen.

### Prerequisites

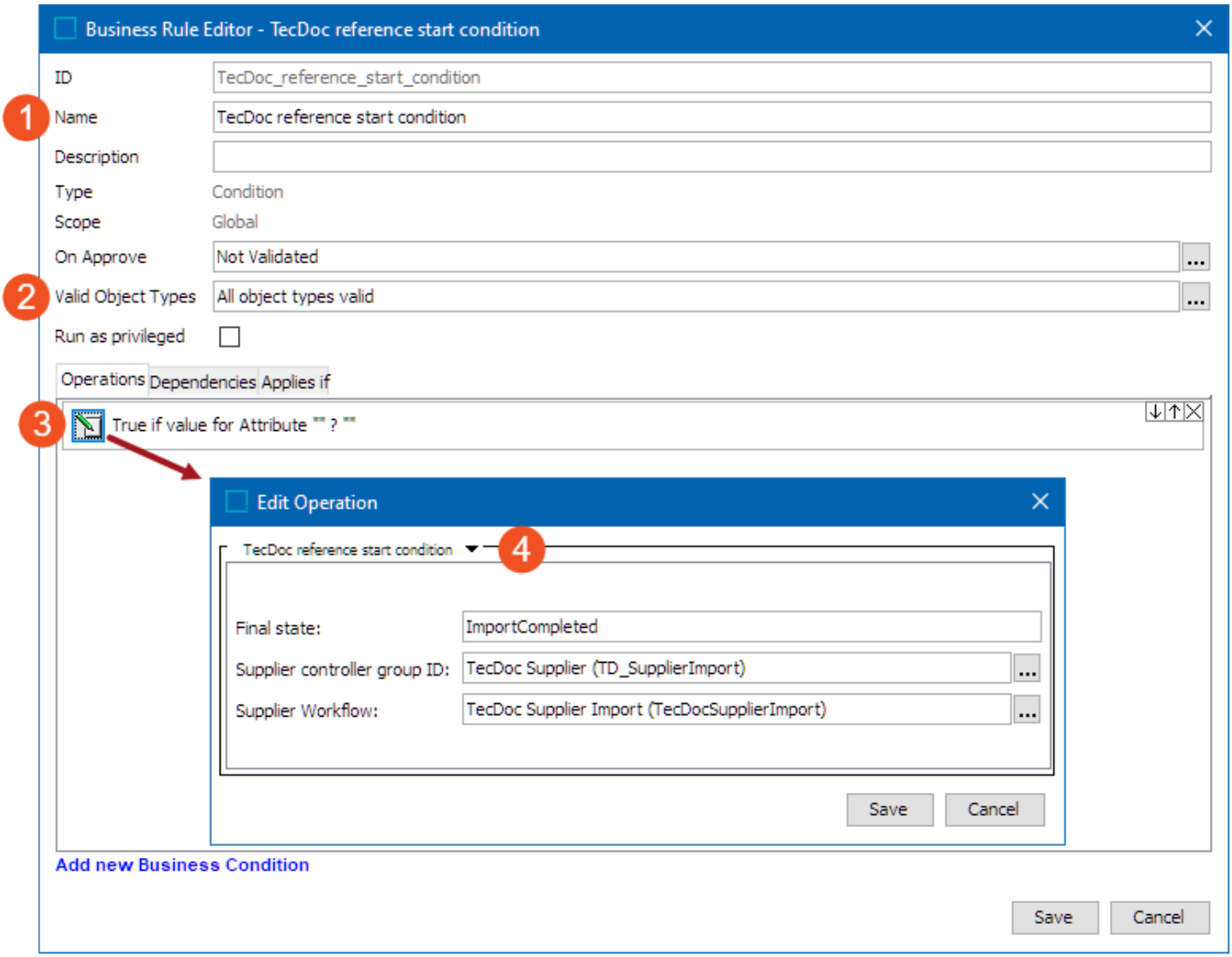
Easy Setup actions automatically create the TecDoc Reference Import workflow, TecDoc Supplier Import workflow, and TecDoc Supplier controller group. The user needs to be aware of the ID of the final state in the TecDoc Supplier Import workflow.

### Configuring 'TecDoc Reference Start Condition' Business Condition

To configure this business condition that controls when to import reference and supplier data, the following is needed:

- Create a business condition using the 'TecDoc reference start condition' operation
- Update workflow settings

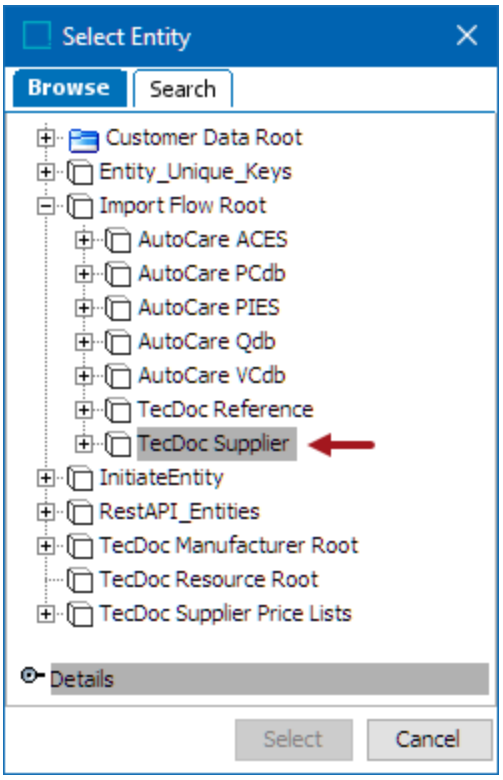
## Create a Business Condition Using the 'TecDoc reference start condition' Operation



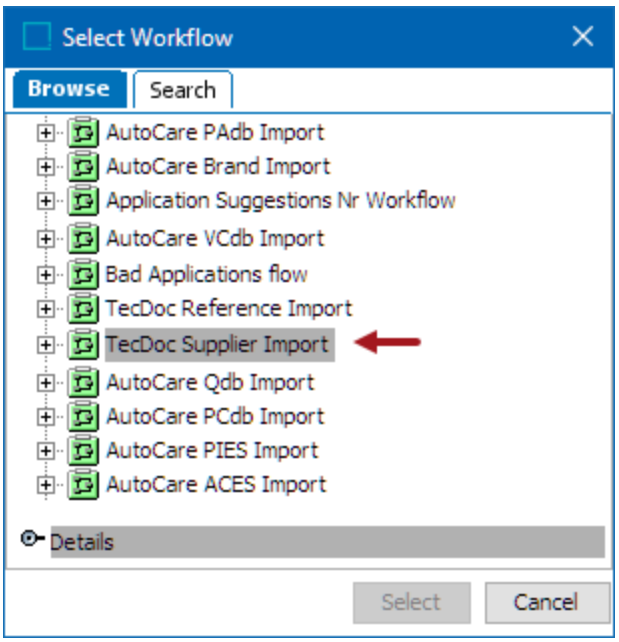

1. Create a business condition, and give it a name that users can easily identify. In the example above, the 'TecDoc reference start condition' name is used.
2. Click the ellipsis button (...) to the right of the Valid Object Types parameter to find and select the necessary object type(s). In the example above, 'All object types valid' is selected.
3. On the Operations tab of the Business Rule Editor, click the 'Add new Business Condition' link, and click the edit button to open the Edit Operation dialog.
4. Use the dropdown menu within the Edit Operation dialog to select **TecDoc** > **TecDoc reference start condition** operation, and the following parameters will display (as shown above).

**Note:** All fields are case sensitive and it is required to be correctly populate them. Business conditions with an erroneous value added to the workflow will fail all the imports without displaying errors in the import list screen.

- **Final state:** Key in the ID of the last workflow state of the TecDoc Reference Import workflow. This should be the typical workflow state that exempts the business condition triggered when any Supplier Controller entity resides in this state. This value will usually be the ID of the Import Completed state (ImportCompleted) as this is the last resorting state for the Reference controller entity. This value is case sensitive.
- **Supplier controller group ID:** Click the ellipsis button (...) to display the Select Entity dialog. Select the supplier controller group where all the controller entities of the supplier files reside.




- **Supplier Workflow:** Click the ellipsis button (...) to display the Select Workflow dialog. Next, select the TecDoc Supplier Import workflow.



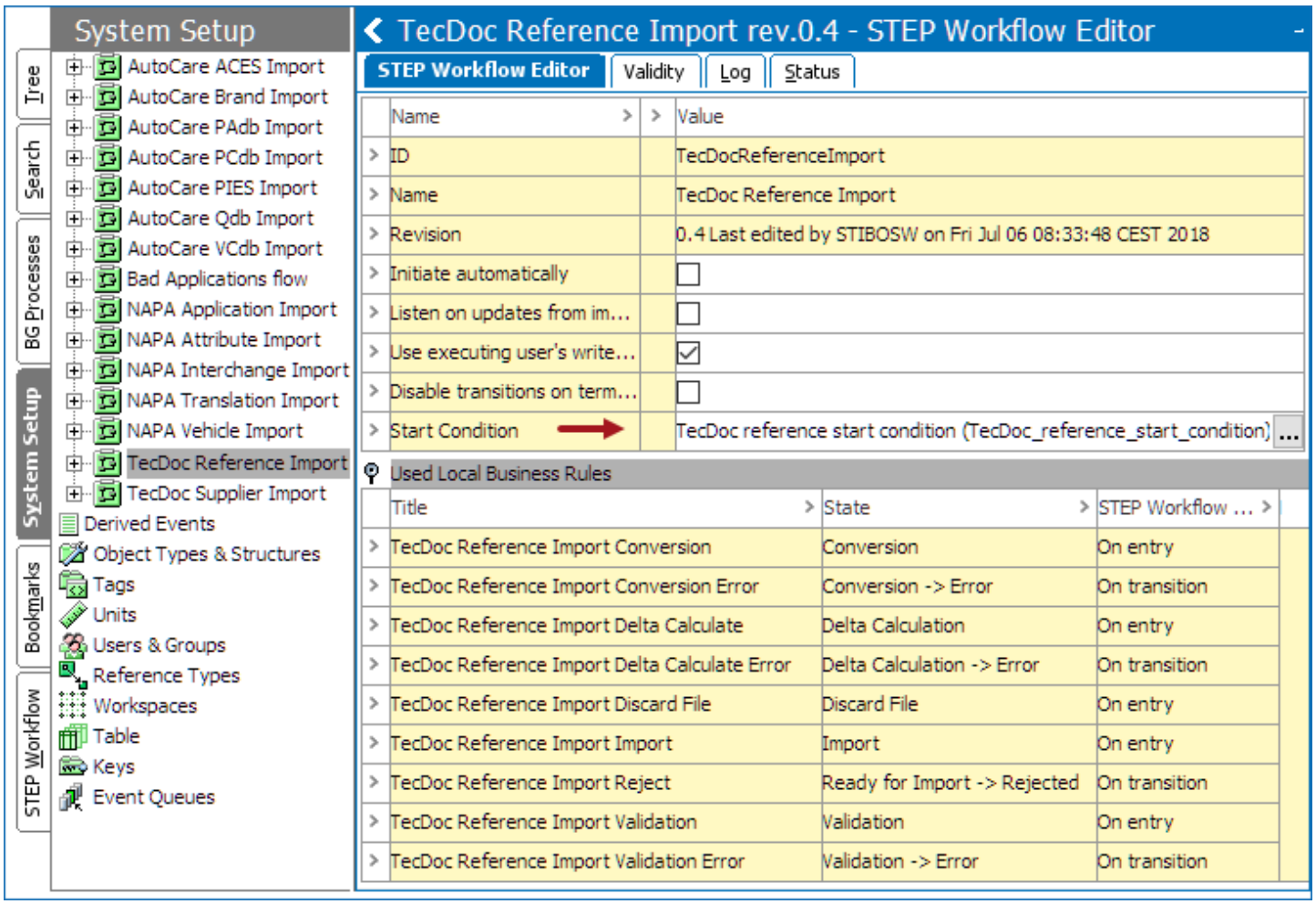

5. Click the **Save** button to save and close the business rule.

   For more information on creating and/or editing business rules, refer to the **Creating a Business Rule, Function, or Library** topic, and the **Editing a Business Rule or Function** topic within the **Business Rules** guide.

## Update Workflow Settings

Once the business condition is created, the necessary workflow needs to be updated to use the business condition. The 'TecDoc reference start condition' business condition is added as the start condition in the TecDoc Reference Import workflow. Below are the steps to update the TecDoc Reference Import workflow to use the business condition created in the previous section.

1. Go to workbench > System Setup > Workflows > select TecDoc Reference Import workflow (ID = TecDocReferenceImport).
2. In the STEP Workflow Editor tab, click the ellipsis button (...) to the right of the Start Condition parameter to find and select the business condition previously created.




## Business Condition: TecDoc Supplier Start Condition

The 'TecDoc supplier start condition' operation is found within the STEP Workbench Business Rule Editor under the TecDoc menu and is used to prevent the start of supplier data import if there is a reference data import running in the background process. It requires multiple parameters to be populated with values, and setup is required in the STEP Workbench for the business condition to be available to users.

Once this business condition is created, users need to add this as the Start Condition in the TecDoc Supplier Import workflow. This business condition will not be created by an Easy Setup and must be manually created and configured by a user. If this business condition rejects an import, no error will be written / displayed on the import list screen.

### Prerequisites

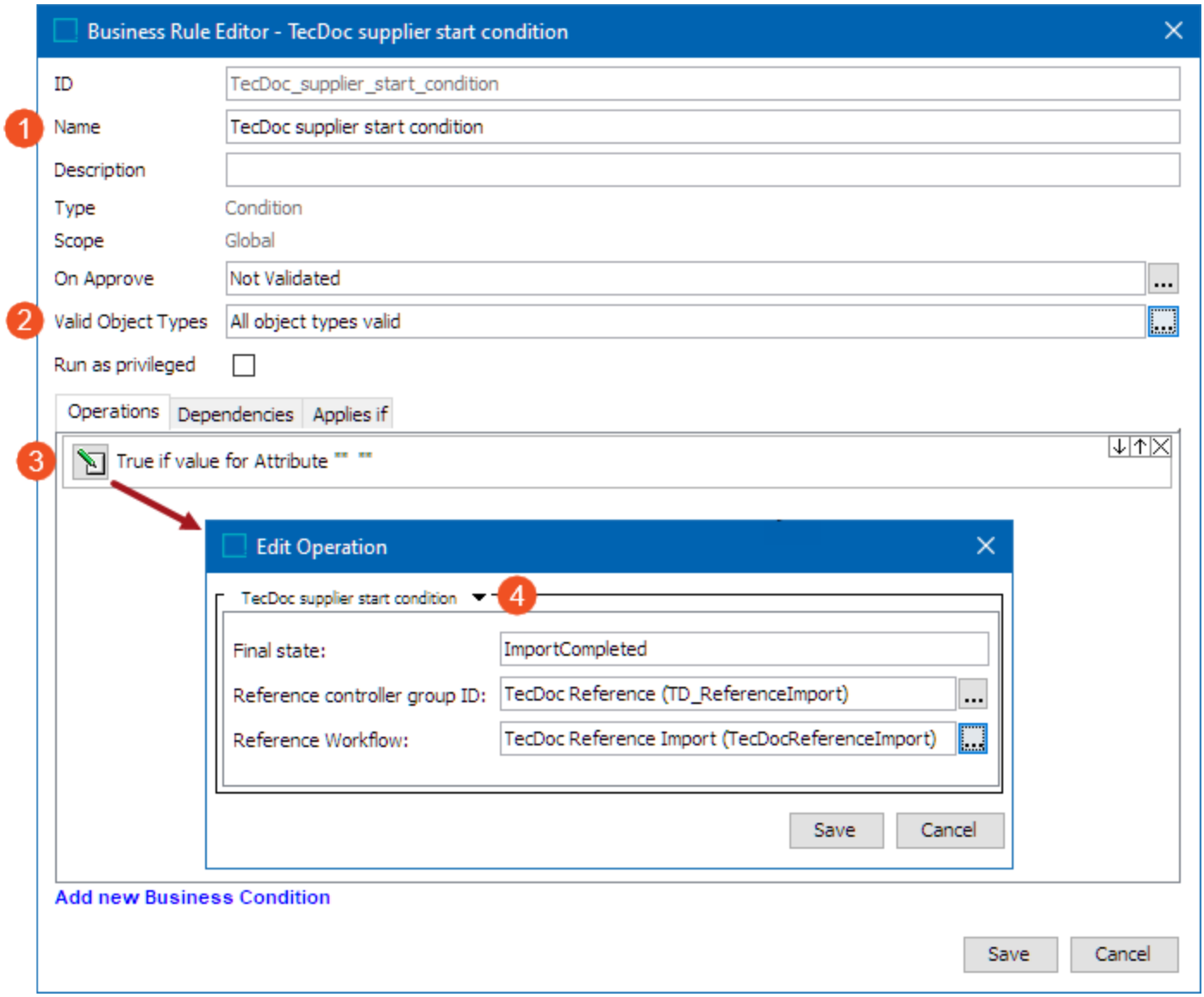
Easy Setup actions automatically create the TecDoc Supplier Import workflow, TecDoc Reference Import workflow, and TecDoc Reference controller group. The user needs to be aware of the ID of the final state in the TecDoc Reference Import workflow.

### Configuring Business Condition

To properly configure this business condition that controls when to import reference and supplier data, the following is needed:

- Create a business condition using the 'TecDoc supplier start condition' operation
- Update workflow settings

## Create a Business Condition Using the 'TecDoc supplier start condition' Operation

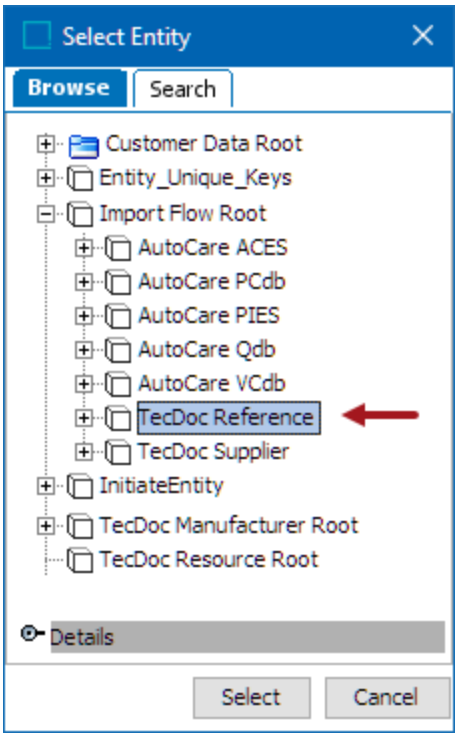



1. Create a business condition and give it a name that users can easily identify. In the example above, the 'TecDoc supplier start condition' name is used.

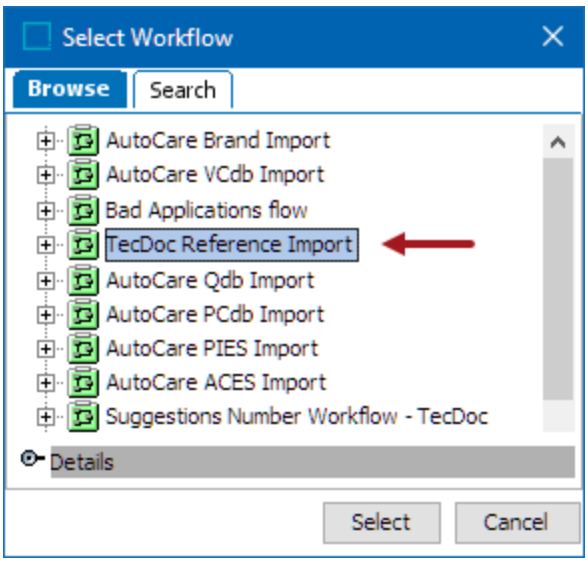
2. Click the ellipsis button (...) to the right of the Valid Object Types parameter to find and select the necessary object type(s). In the example above, 'All object types valid' is selected.
3. On the Operations tab of the Business Rule Editor, click the 'Add new Business Condition link,' and click the edit button to open the Edit Operation dialog.
4. Use the dropdown menu within the Edit Operation dialog to select **TecDoc** > **TecDoc supplier start condition** operation, and the following parameters will display (as shown above).

**Note:** All fields are case sensitive and it is required to correctly populate them. Business conditions with an erroneous value added to the workflow will fail all the imports without displaying errors in the import list screen.

- **Final state:** Key in the ID of the last workflow state of the TecDoc Supplier Import workflow. This value will usually be the ID of the Import Completed state (ImportCompleted), as this is the last resorting state for the supplier controller entity. This value is case sensitive. This should be the typical workflow state that exempts the business condition triggered when any supplier controller entity resides in this state.
- **Reference controller group ID:** Click the ellipsis button (...) to display the Select Entity dialog. Select the reference controller group where all the controller entities of the reference files reside.




- **Reference Workflow:** Click the ellipsis button (...) to display the Select Workflow dialog. Next, select the TecDoc Reference Import workflow.



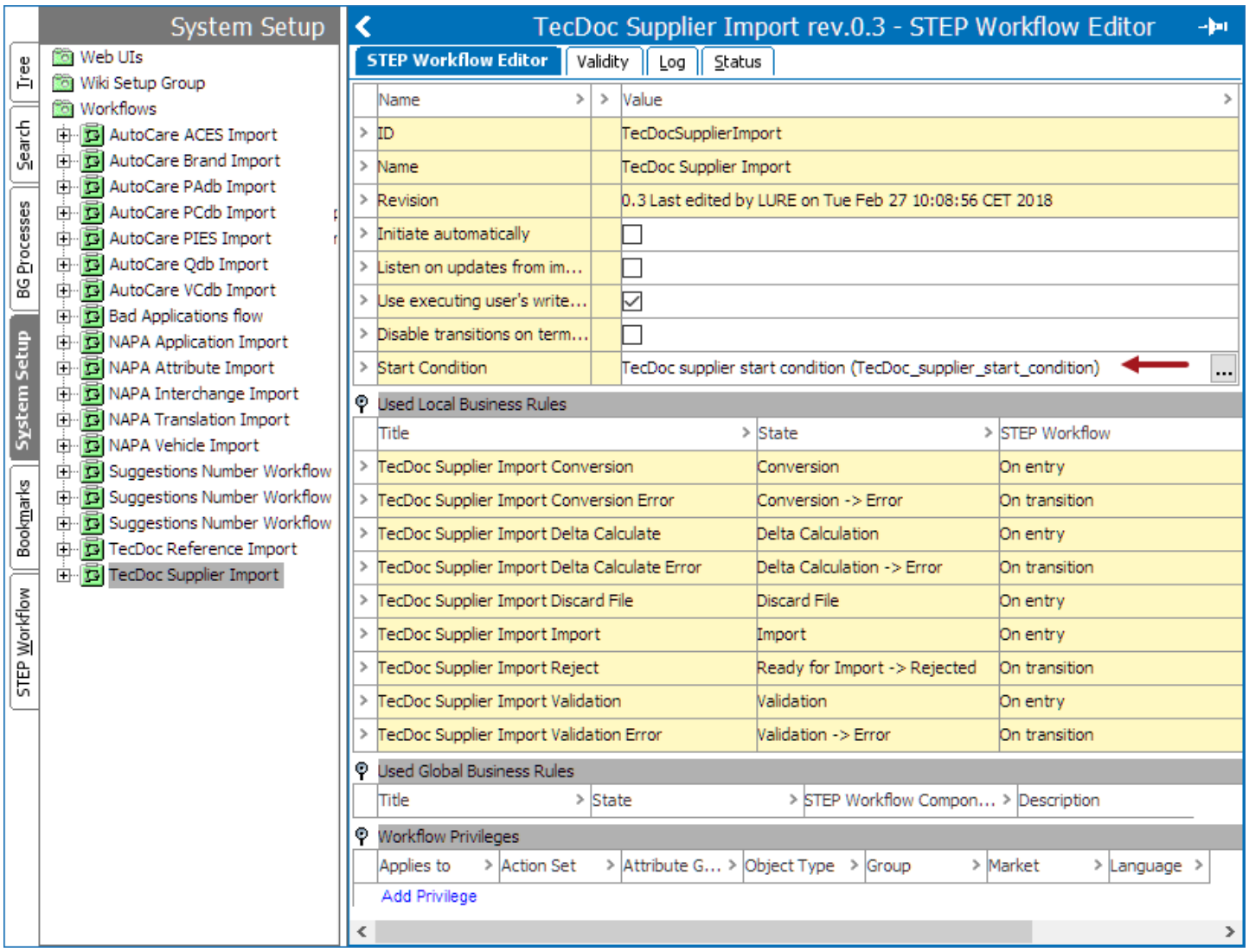

6. Click the **Save** button to save and close the business rule.

   For more information on creating and/or editing business rules, refer to the **Creating a Business Rule, Function, or Library** topic, and the **Editing a Business Rule or Function** topic within the **Business Rules** guide.

## Update Workflow Settings

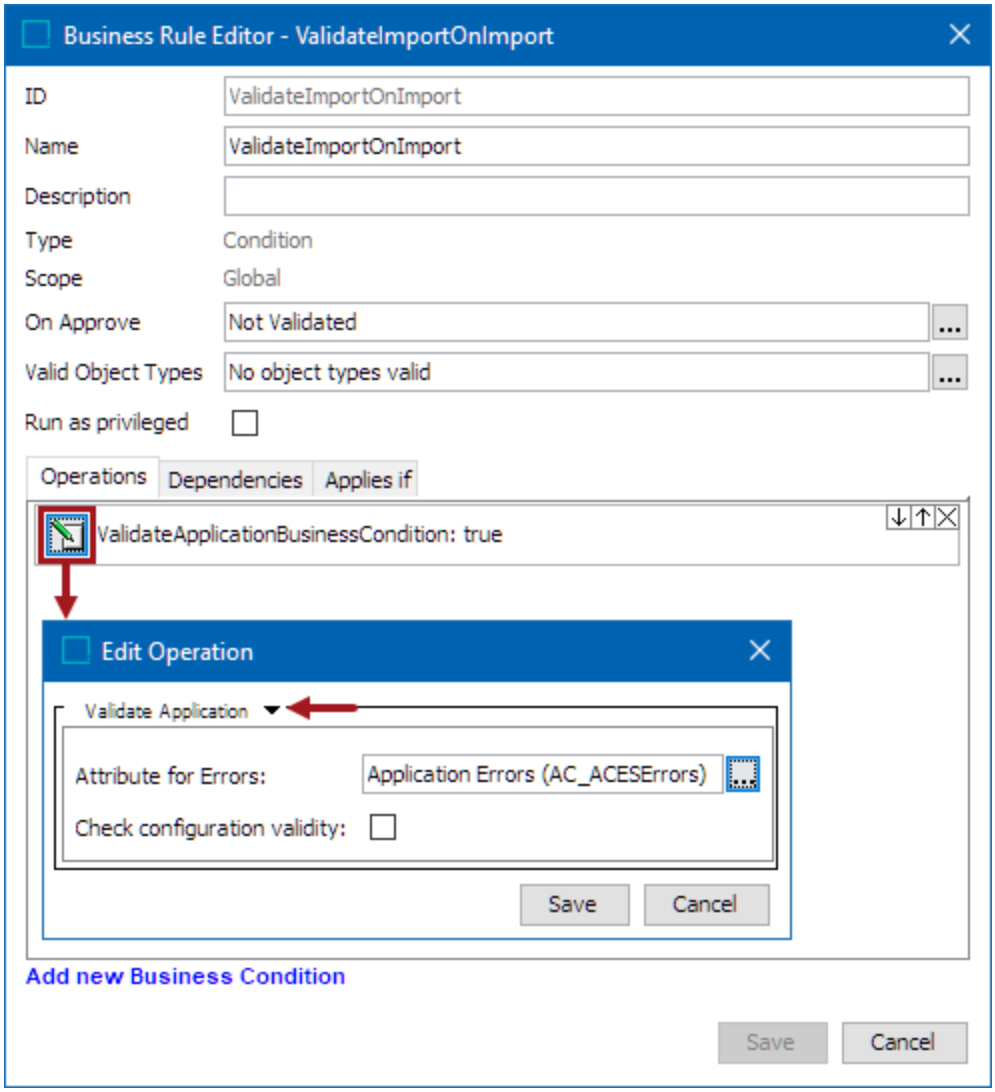
Once the business condition is created, the necessary workflow needs to be updated to use the business condition. The 'TecDoc supplier start condition' business condition is added as the start condition in the TecDoc Supplier Import workflow. Below are the steps to update the TecDoc Supplier Import workflow to use the business condition created in the previous section:

1. Go to workbench > System Setup > Workflows > select TecDoc Supplier Import workflow (ID = TecDocSupplierImport).
2. In the STEP Workflow Editor tab, click the ellipsis button (...) to the right of the Start Condition parameter to find and select the business condition previously created.

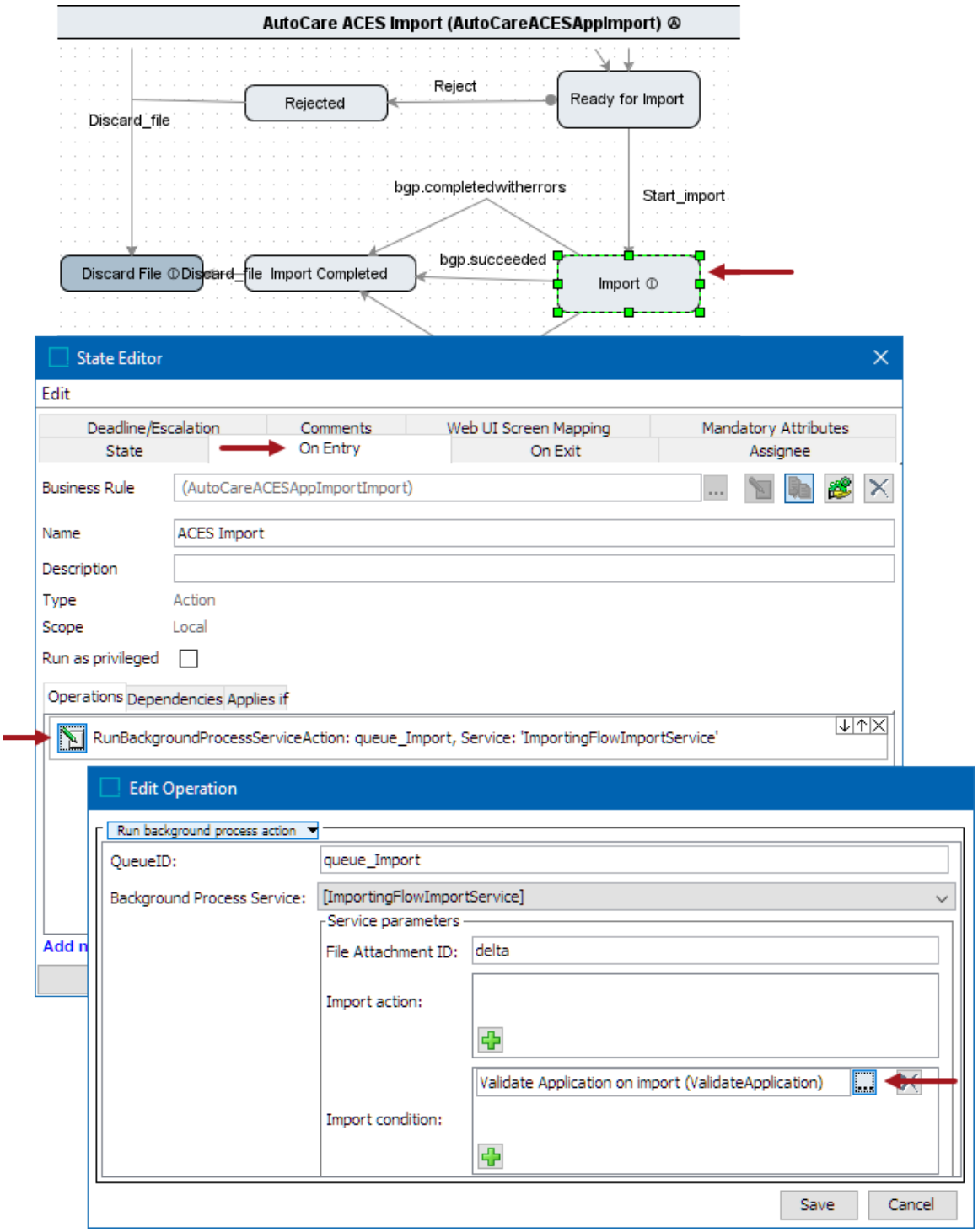

System Setup
Web UIs
Wiki Setup Group
Workflows
AutoCare ACES Import
AutoCare Brand Import
AutoCare PAdb Import
AutoCare PCdb Import
AutoCare PIES Import
AutoCare Qdb Import
AutoCare VCdb Import
Bad Applications flow
NAPA Application Import
NAPA Attribute Import
NAPA Interchange Import
NAPA Translation Import
NAPA Vehicle Import
Suggestions Number Workflow
Suggestions Number Workflow
Suggestions Number Workflow
TecDoc Reference Import
TecDoc Supplier Import
Tree
Search
BG Processes
System Setup
Bookmarks
STEP Workflow
TecDoc Supplier Import rev.0.3 - STEP Workflow Editor
STEP Workflow Editor
Validity
Log
Status
Name
Value
ID
TecDocSupplierImport
Name
TecDoc Supplier Import
Revision
0.3 Last edited by LURE on Tue Feb 27 10:08:56 CET 2018
Initiate automatically
Listen on updates from im...
Use executing user's write...
Disable transitions on term...
Start Condition
TecDoc supplier start condition (TecDoc_supplier_start_condition)
Used Local Business Rules
Title
State
STEP Workflow
TecDoc Supplier Import Conversion
Conversion
On entry
TecDoc Supplier Import Conversion Error
Conversion -> Error
On transition
TecDoc Supplier Import Delta Calculate
Delta Calculation
On entry
TecDoc Supplier Import Delta Calculate Error
Delta Calculation -> Error
On transition
TecDoc Supplier Import Discard File
Discard File
On entry
TecDoc Supplier Import Import
Import
On entry
TecDoc Supplier Import Reject
Ready for Import -> Rejected
On transition
TecDoc Supplier Import Validation
Validation
On entry
TecDoc Supplier Import Validation Error
Validation -> Error
On transition
Used Global Business Rules
Title
State
STEP Workflow Compon...
Description
Workflow Privileges
Applies to
Action Set
Attribute G...
Object Type
Group
Market
Language
Add Privilege

## 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.

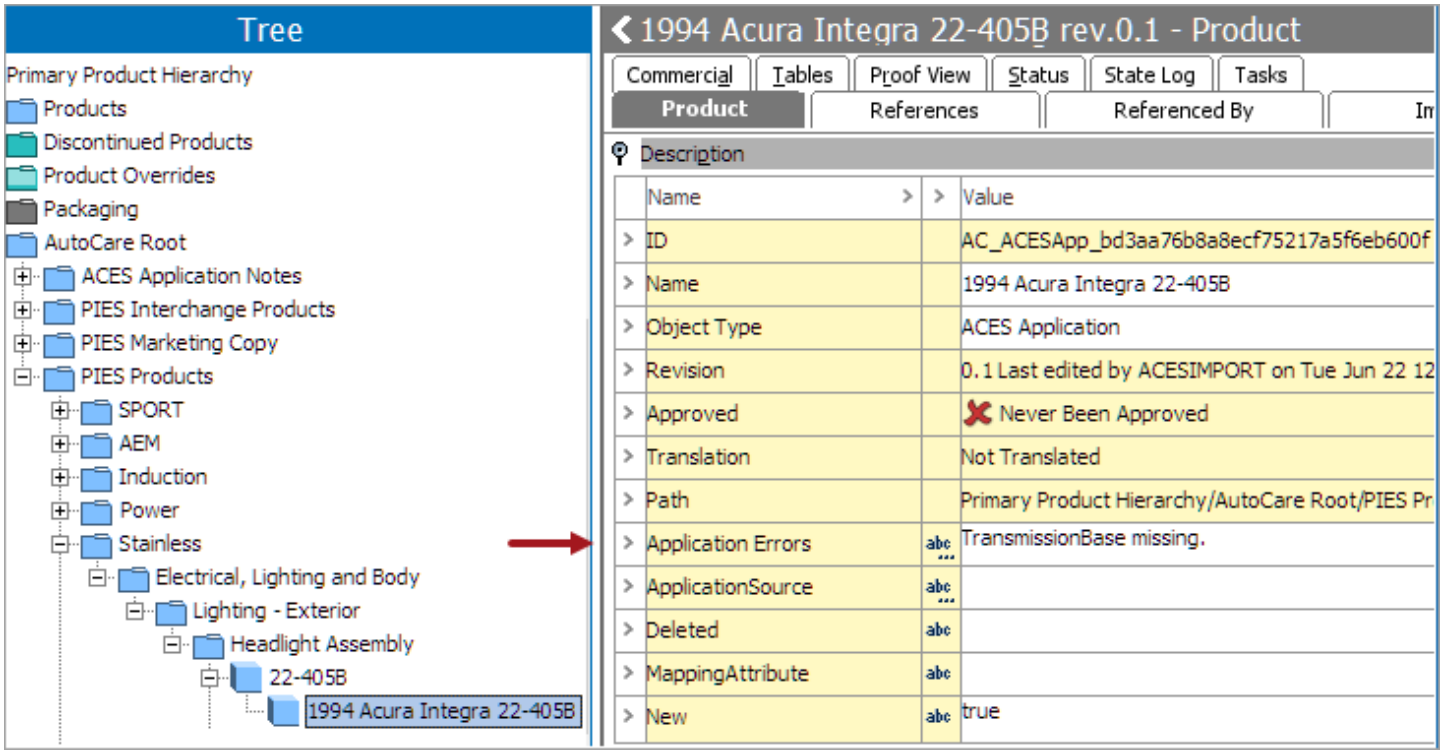



The 'Validate Application' operation is executed via the business condition 'Validate Application on import.' Running the Easy Setup of the import process automatically creates the 'Validate Application on import' business condition and adds it to the Import state of the relevant workflow.




The 'Attribute for Errors' is an optional parameter that writes all the vehicle validation error messages in the attribute when configured with an attribute valid for the application object type. Step 1 of the Automotive - AutoCare Model Easy Setup creates the 'AC_ACESErrors' attribute below the Metadata attribute group and makes it valid for the ACES Applications object type. Click the ellipsis button (...) to search and select the 'AC_ACESErrors' attribute. Configuring the 'AC_ACESErrors' attribute in the parameter writes all the vehicle

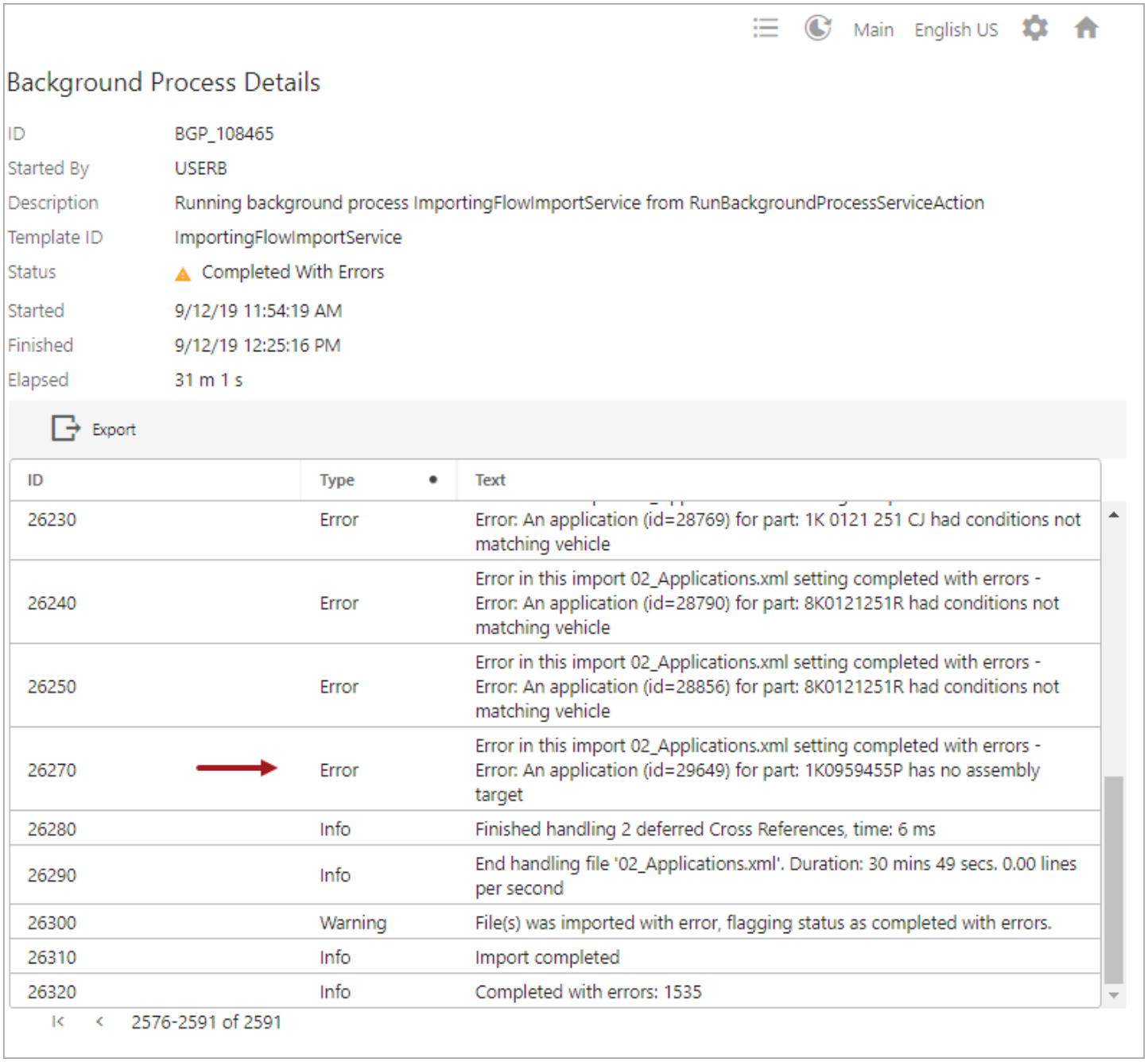
validation error messages of the faulty application record in the attribute. Below is an example of an error that is written to the attribute Application Errors (ID = AC_ACESErrors) for an application record (1994 Acura Integra 22-405B) when a Transmission Base mentioned in the ACES file does not exist in the STEP database.




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, or invalid combinations of vehicle configurations that are provided for the base vehicle, such as an invalid Engine Base or Drive Type) are imported or not.

Checking the 'Check configuration validity' parameter ensures that the application records with invalid configurations are not imported into the system. 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 application record is not imported. The error contains the application ID from the ACES file and 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.




If the 'Check configuration validity' parameter is not checked, importing ACES applications that have invalid configurations will pass the import and create the applications.

# 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, refer to the **Configuring Automotive - Country Model Component Model** topic.
3. If needed, configure the 'Application Country Handling Table Header' component.

   For more information, refer to the **Configuring Application Country Handling Table Header** topic.
4. If needed, configure the 'Country Handling Value' component.

   For more information, refer to 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, refer to 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.

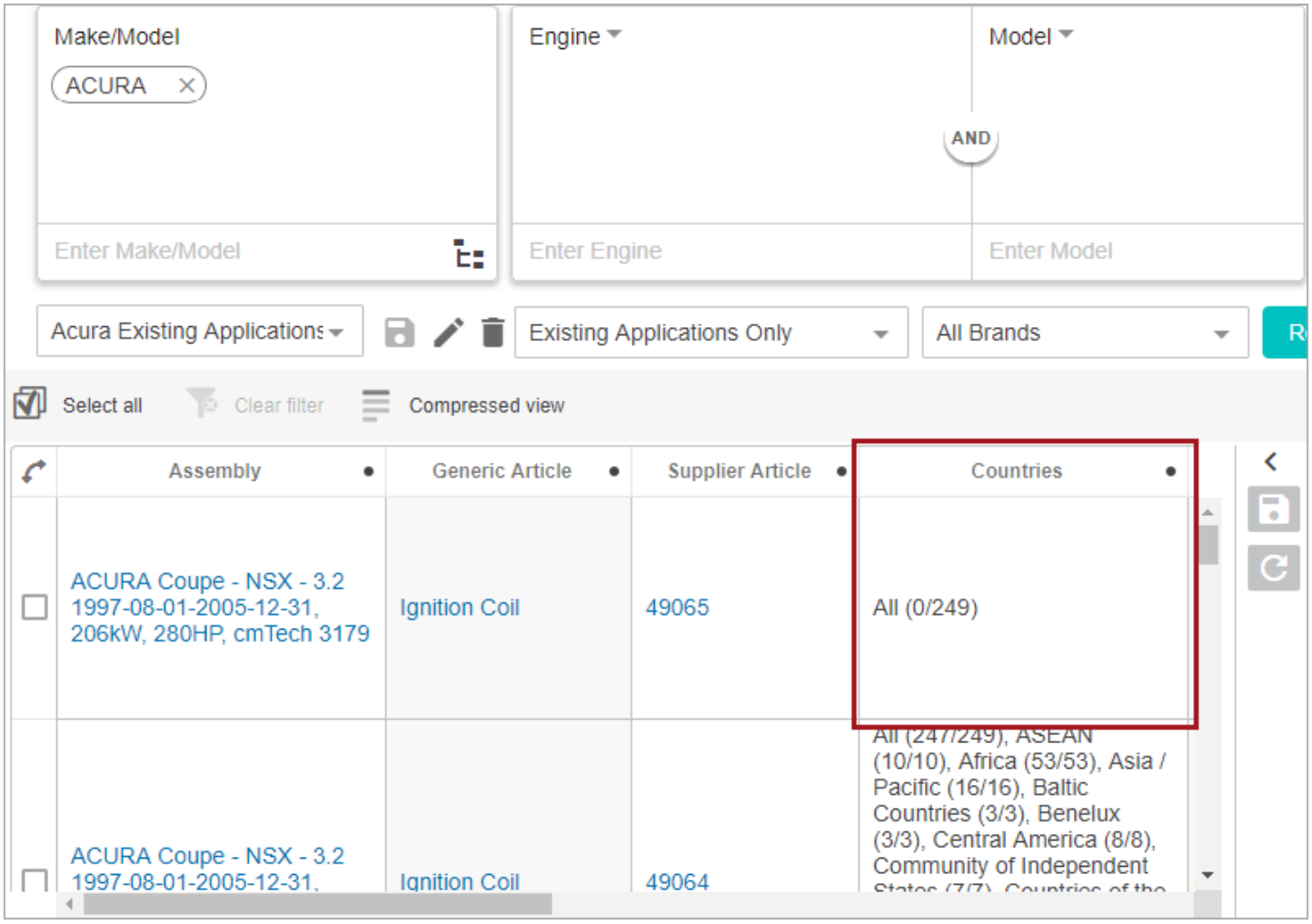
## Prerequisites

Configuration is required before the Country Handling solution can be used. For more information, refer to 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 Screen, the 'Application Country Handling Table Header' component can be configured to display the country inclusion data within the 'Countries' 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 between the configured exclusion and inclusion attributes, zero values are populated. In other words, for the possible 249 countries, no inclusions have been added to STEP. Therefore, the inclusion of all countries are assumed because no data exists in the exclusion and/or inclusion attributes.




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, refer to 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 |
|---|---|---|---|---|
| ☐ | 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.

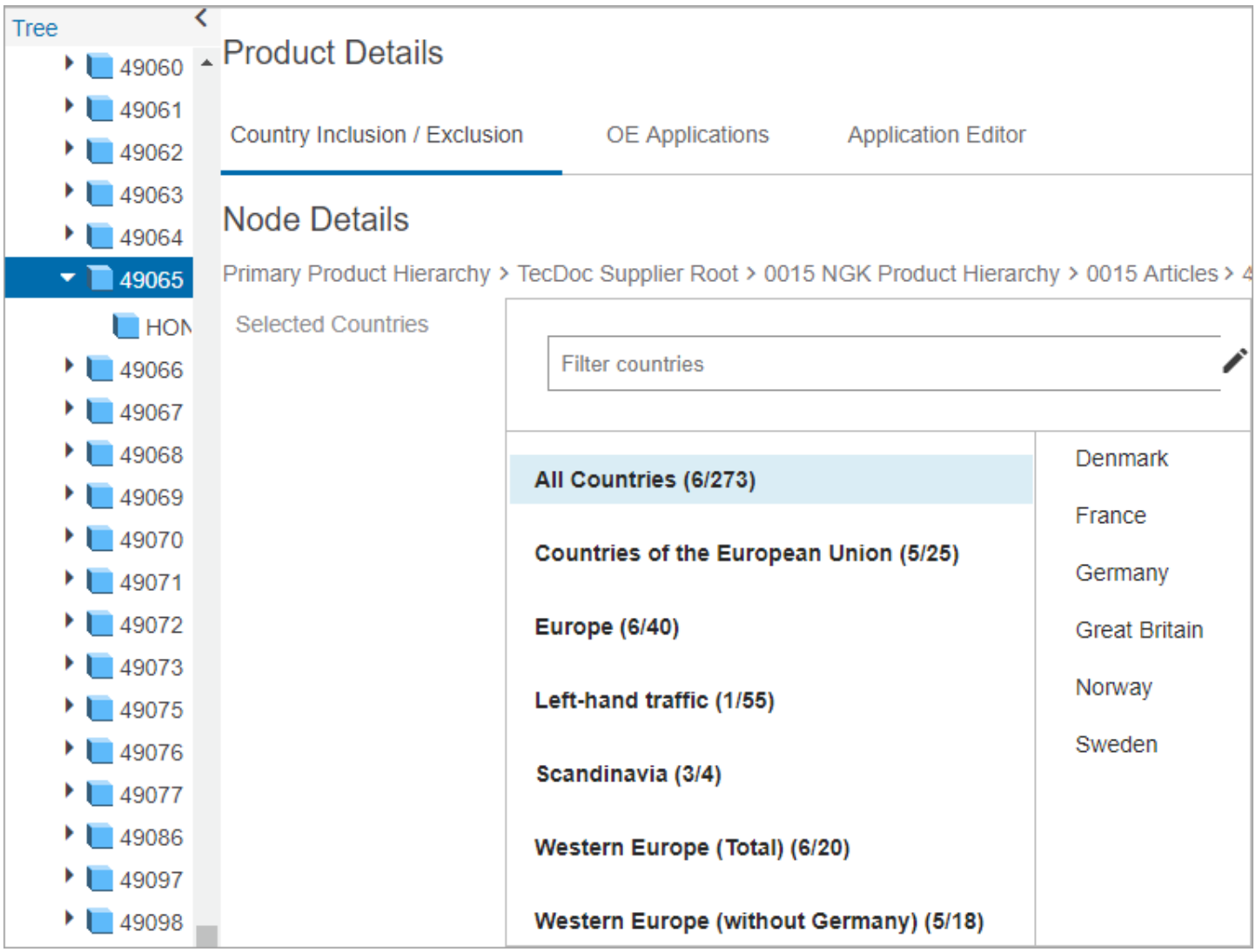
**Important:** Once a country value is added for inclusion, if it is then added to the exclusion attribute, it is automatically removed from inclusion attribute. The same country cannot exist in both the inclusion attribute and the exclusion attribute.

## 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).

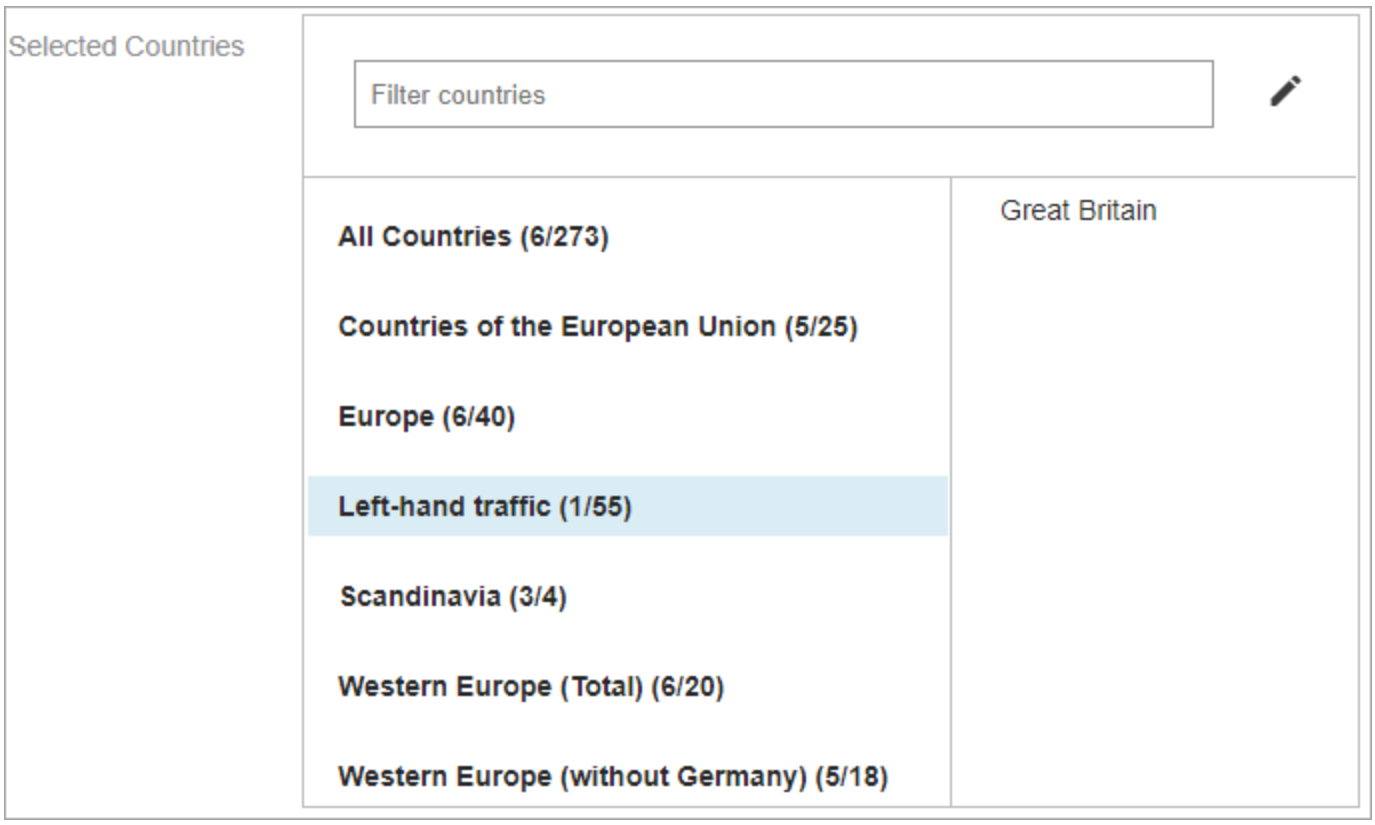
When configured, this component can be used to view and edit the country inclusion data for applications and/or parts. Once an application and/or part object type is selected, then the Country Handling Value component displays only those countries currently referenced to the attribute or part for inclusion. 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).

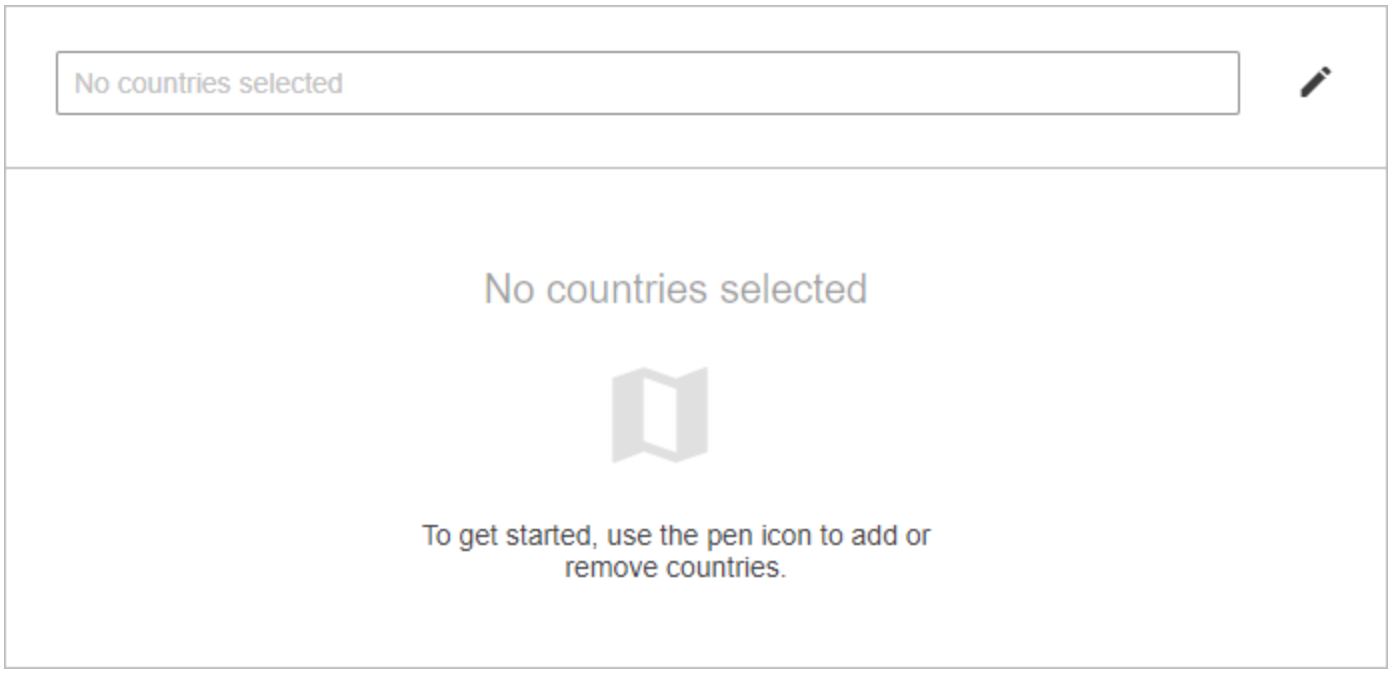



When the component first displays, the 'All Countries (number of countries referenced for inclusion / total number of countries)' group heading displays using bold text and highlighted in blue. Each country group containing a country referenced for inclusion will display in bold below the 'All Countries' 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, 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.

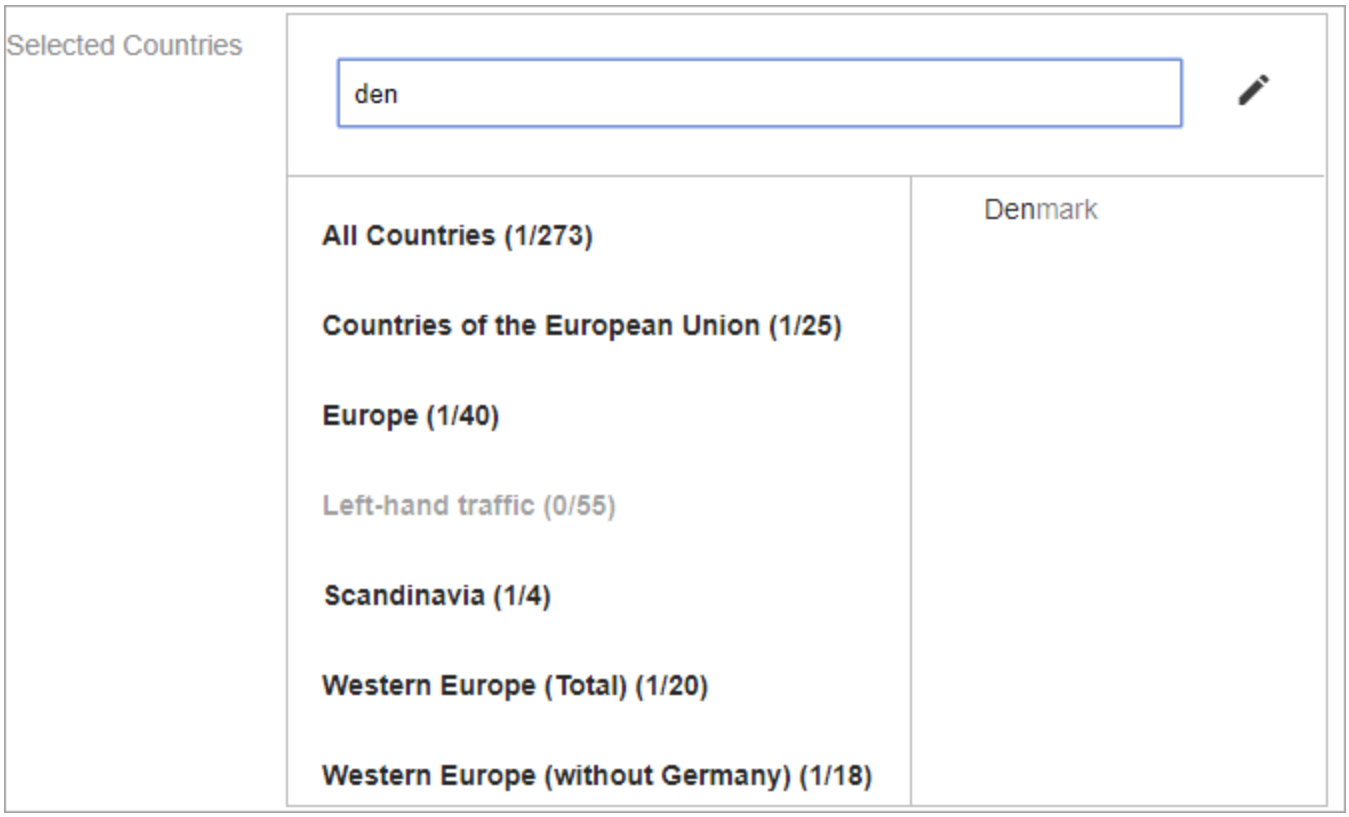



When country inclusion and/or exclusion data is not provided, then the component will display 'No countries selected' (as shown below). The edit button ( ) can be used to add country handling data.

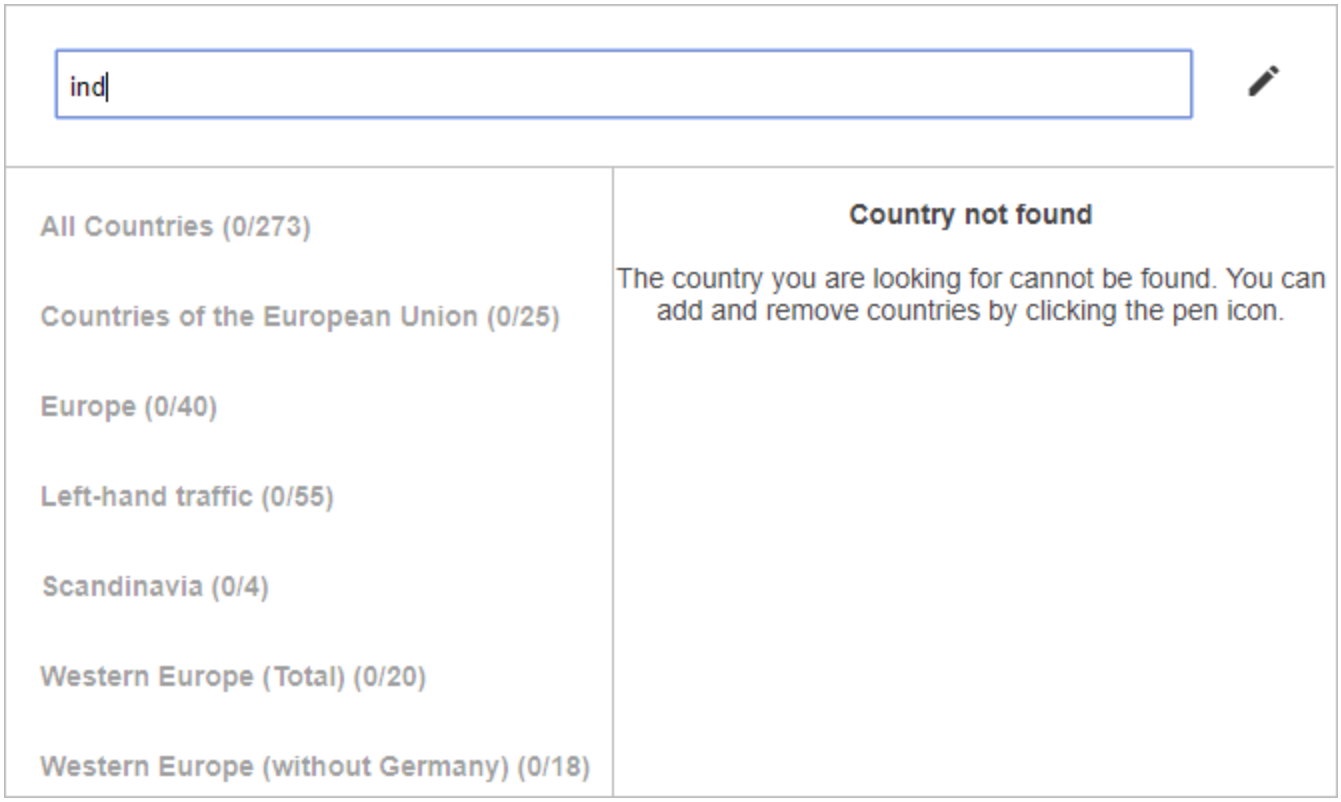


To search for a specific country within any of the listed country groups with inclusion data, type text into the 'Filter countries' text box and the results will display country names that match the typed text to the right of the country group headers.

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.




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).



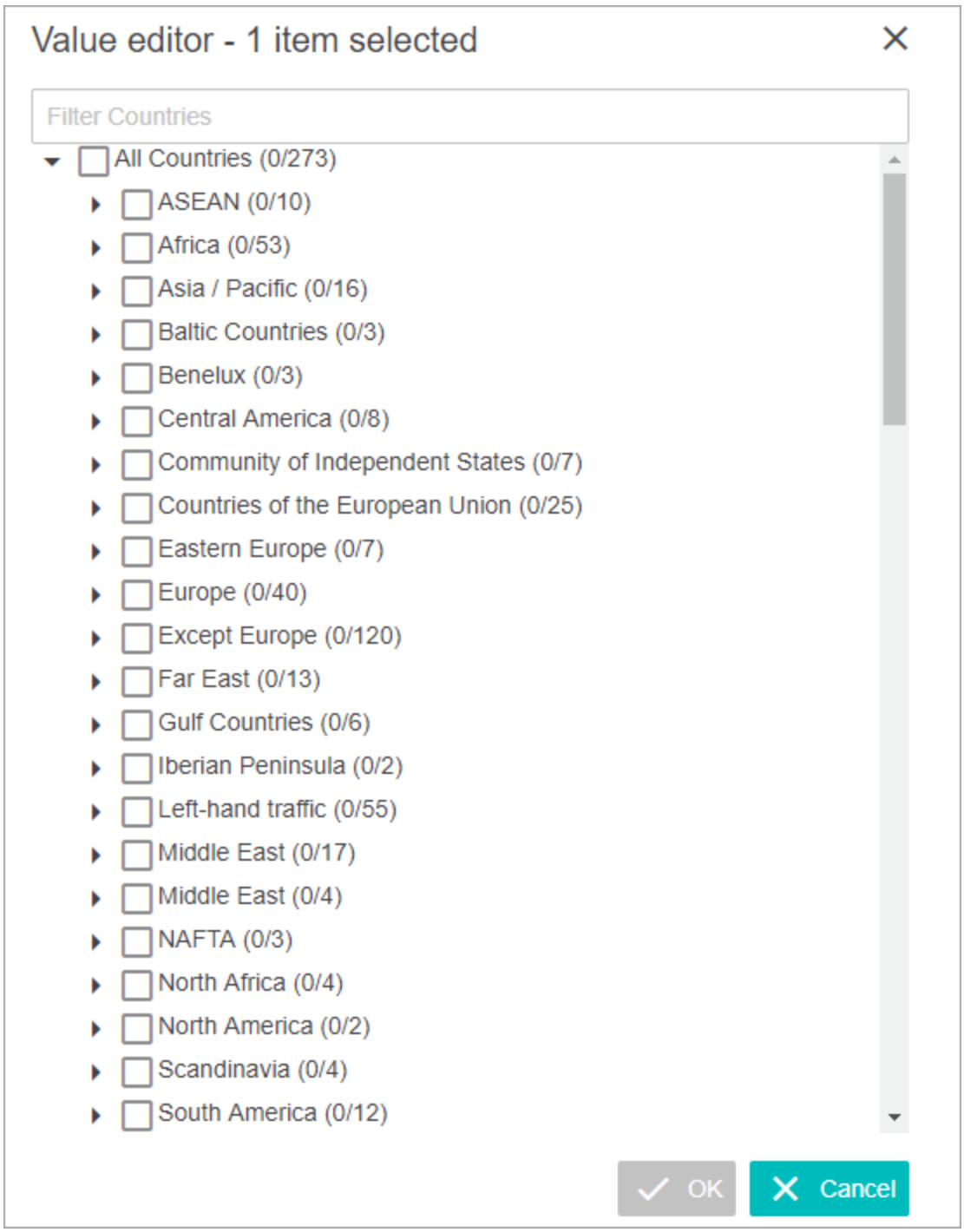

To add or edit country inclusion / exclusion data to an application or part using the 'Country Handling Value' component, click the edit button ( ) to the right of the Filter countries field, and a Value editor dialog will display. For more information on using the Value editor, refer to the **Using 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, refer to 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, refer to the **Configuring Country Handling Value Component** topic.

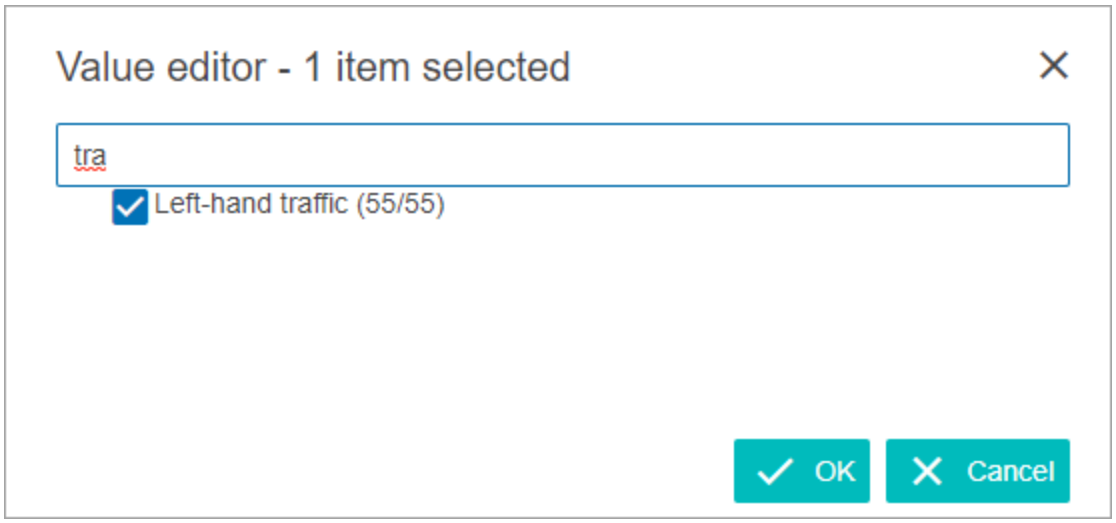



**Note:** The display of the countries and their groupings vary based upon the configuration of the Automotive - Country Model component model. For more information, refer to the **Configuring Automotive - Country Model Component Model** topic.

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 (☑).

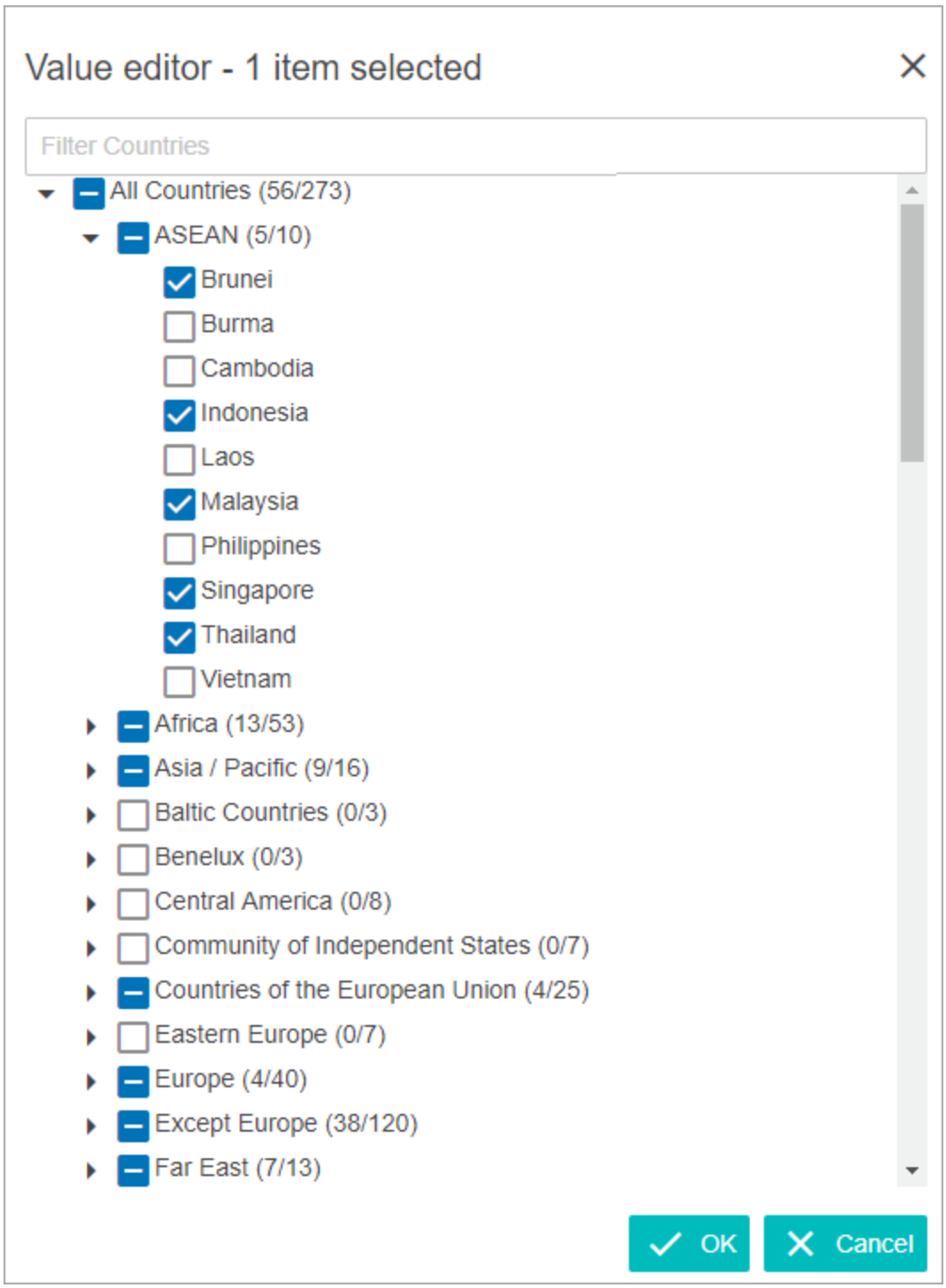



Clicking the OK button closes the dialog and displays the 55 countries selected from the 'Left-hand traffic' country group.

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 (⊟) next to the country group is used when only some of the countries are selected within the country group. Clicking on this icon will toggle to select / deselect all countries within the country group. The blue and white checkmark icon (☑) is used when all the countries within a country group are selected, or when an individual country is 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.

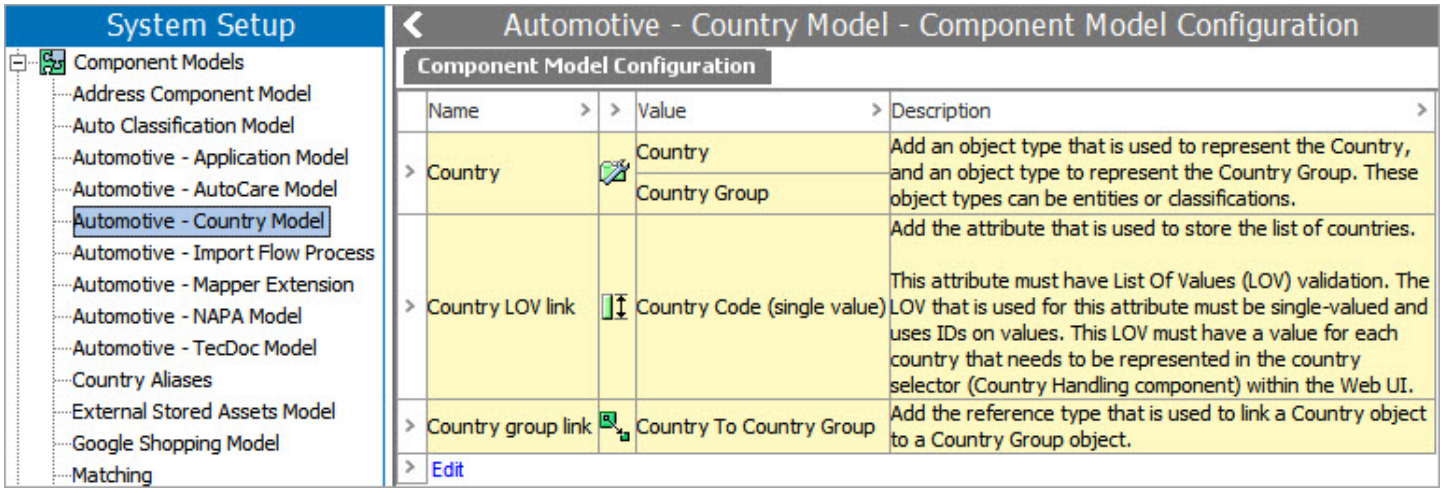
Once the desired countries for inclusion are selected, click the OK button to close the dialog and return to the previous screen.

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, attribute, and Reference Type have been used to configure the component model values.




**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 Country Group object types to be used within the Country Handling solution. Add an object type for Country and an object type for Country Group. An example of a Country is 'Denmark' (DK), and the Country Group that Denmark can belong to would be 'Northern Europe' and/or 'Scandinavia.'

  Importing the TecDoc reference data file will create Country and Country Group classification objects, as well as link the Country object to the Country Group object by using the 'Country To Country Group' (TD_CountryToCountryGroup) Classification Reference Type that is defined in the Country group link parameter.

  However, users can create Countries and Country Groups as entity objects instead of using classification objects. If this is the case, then the entity object types will need to be added to this parameter that represents the Country entity and Country Group entity.

**Important:** The Country object type must be configured as the Valid Source type, and the Country Group object type must be configured as the Valid Target Type in the Reference Type that is defined within the 'Country group link' parameter. And the Country object MUST be linked to a Country Group object in order

for the country selector to display properly within the Web UI. Also, this solution will only consider leaf objects as the countries and currently only handles one object type for Country and one object type for Country Group. It will not work if multiple object types are added to represent the Country and multiple object types are added to represent the Country Group.

- **Country LOV link:** Required parameter used to define the attribute that contains a List of Values for countries and country groups. 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 IDs. In the example above, the following LOV value is added: 'Country Code (single value)' (TD_ATTR_CountryCode).

  **Important:** In addition, the attribute should be made valid for any part, linkage, or target that a country can be related to. The countries will only be selectable within the Web UI Country Handling component when a Country object has been created and a value has been populated for this attribute.

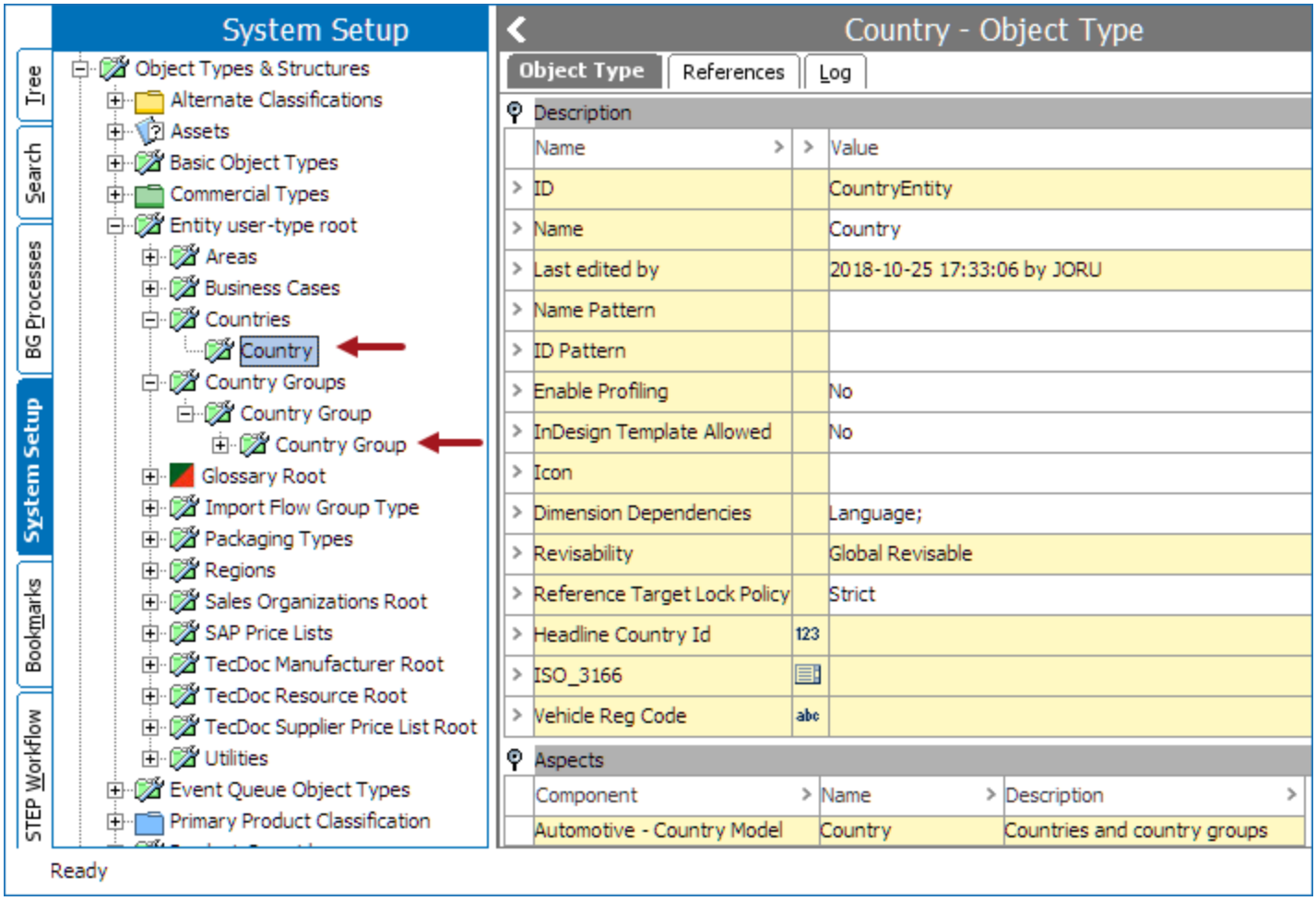
- **Country group link:** Required parameter used to define the reference type that is used to link a Country object to a Country Group object. In the example above, the following value is added: 'Country to Country Group' (TD_CountryToCountryGroup).

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, refer to the **Configuring Application Country Handling Table Header**, and the **Configuring Country Handling Value Component** topics.

The steps listed below is an example of what a user would need to do from beginning to end in order to configure the Country Handling solution.

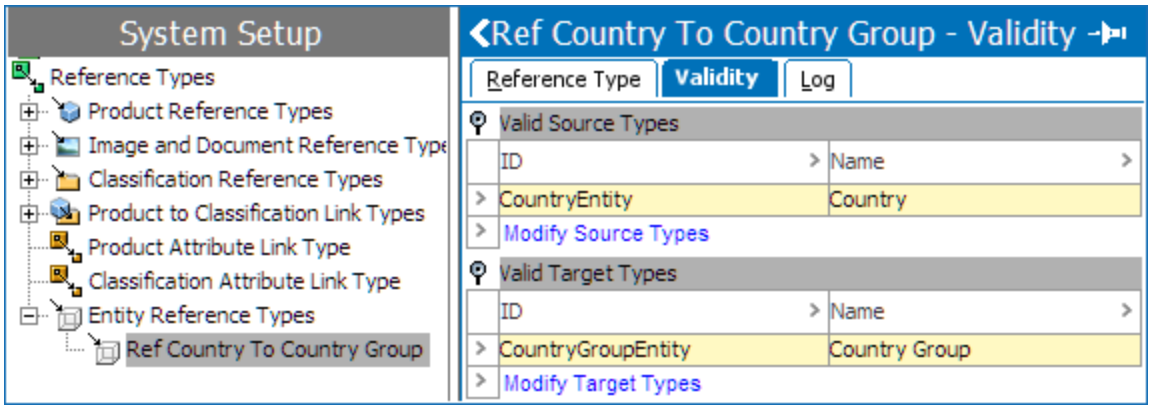
1. Create object types to represent the Country and Country Group.

   In the screenshot below, entity object types have been created for the Country and Country Group hierarchy. The leaf level object type for Country and Country Group is what will be used to be defined in the Automotive - Country Model component model.

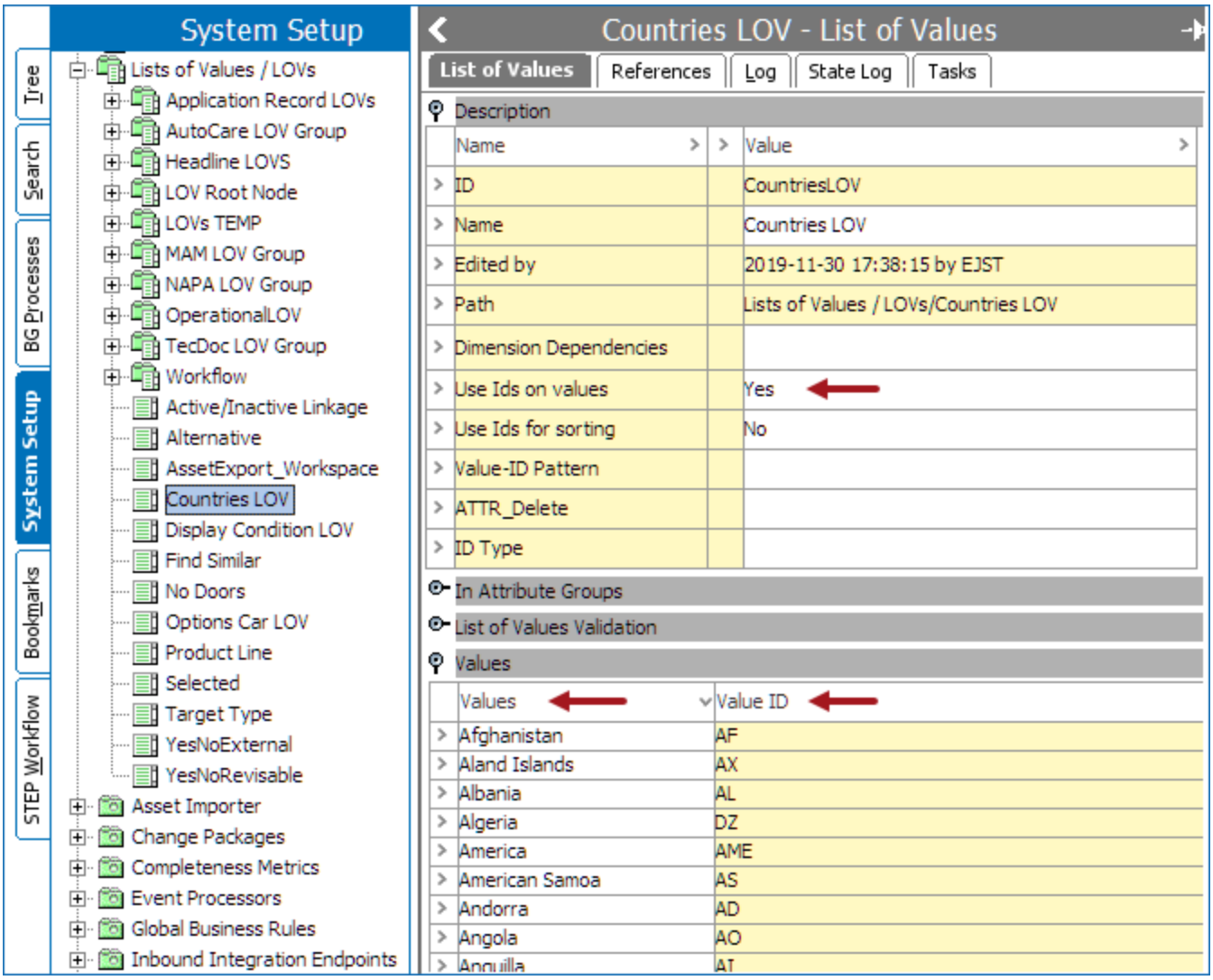



2. Create a reference type that will be used to link the Country object to the Country Group object.

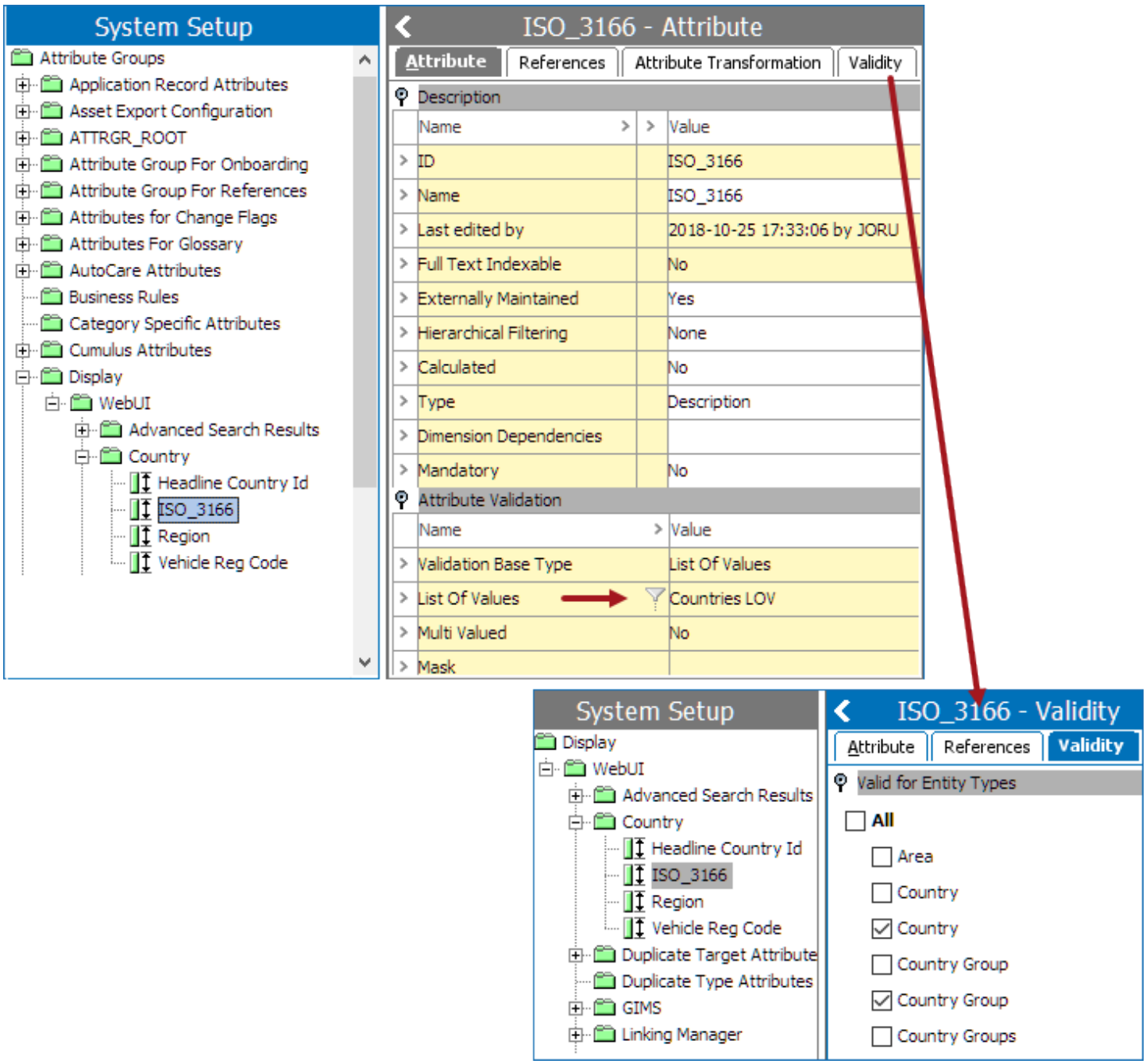
   In the screenshot below, reference type 'Ref Country To Country Group' has been created with the CountryEntity (that was created in step 1) as the Valid Source Type and CountryGroupEntity as the Valid Target Type.



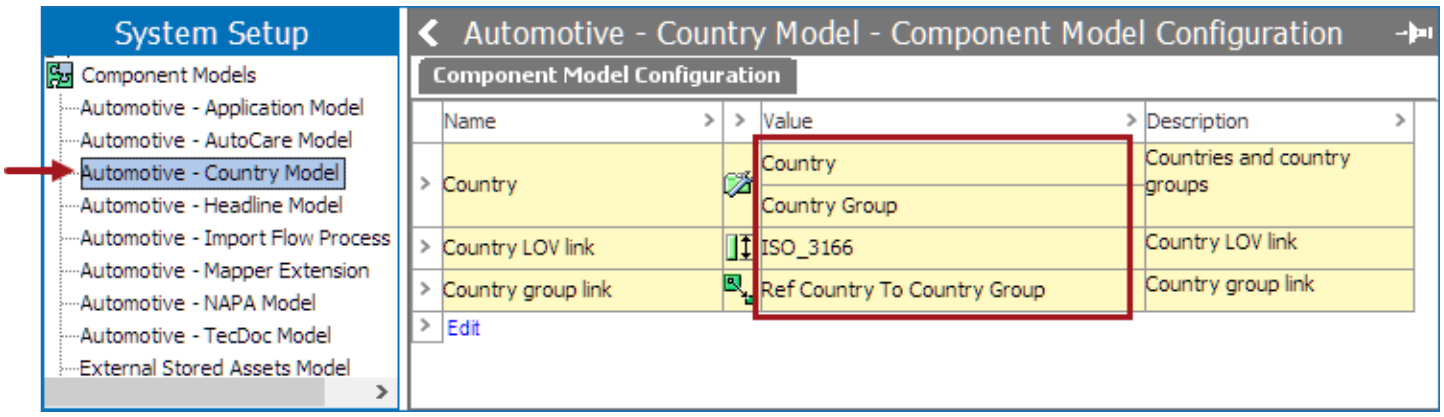
3. Create a List of Values (LOV) that contains the values for all of the Countries and Country Groups that can be selected.
   - The LOV must have Use Ids on values = Yes
   - The LOV Value ID is the Country Code (e.g. 'FO'), or the Country Group Code (e.g. 'NOC')
   - The LOV Values is the Country name (e.g. 'Faroe Islands'), or the Country Group name (e.g. 'Nordic Countries')



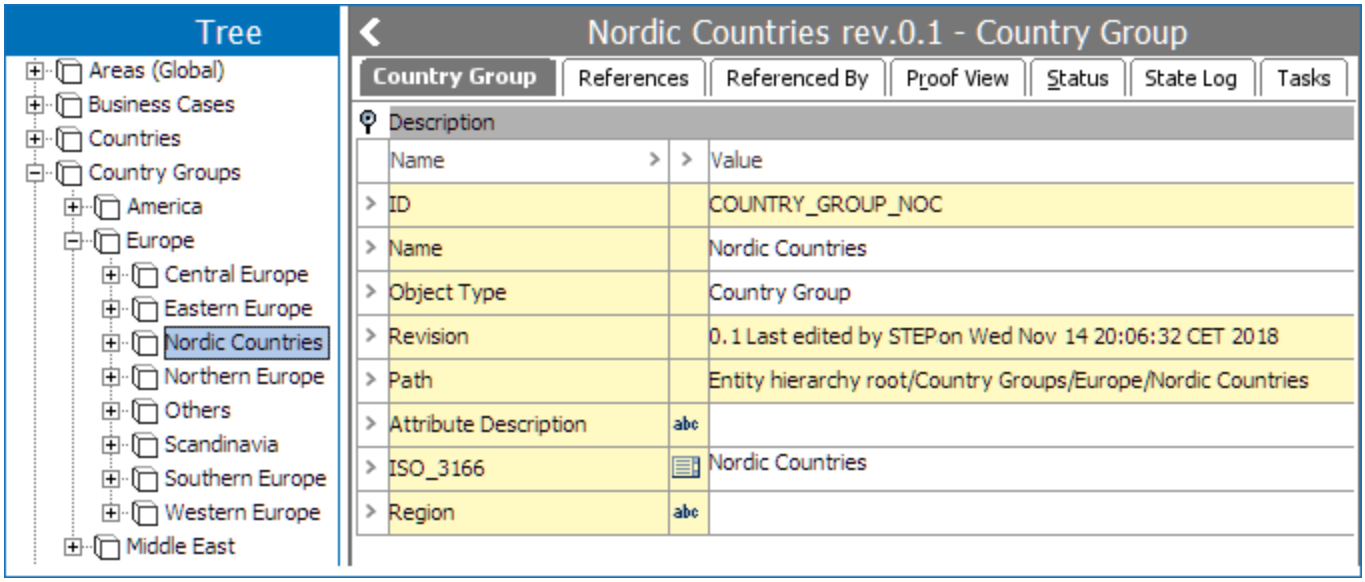

4. Create an attribute using the LOV from step 3 and make it valid for the Country and Country Group object types in order to be able to populate the value for those objects



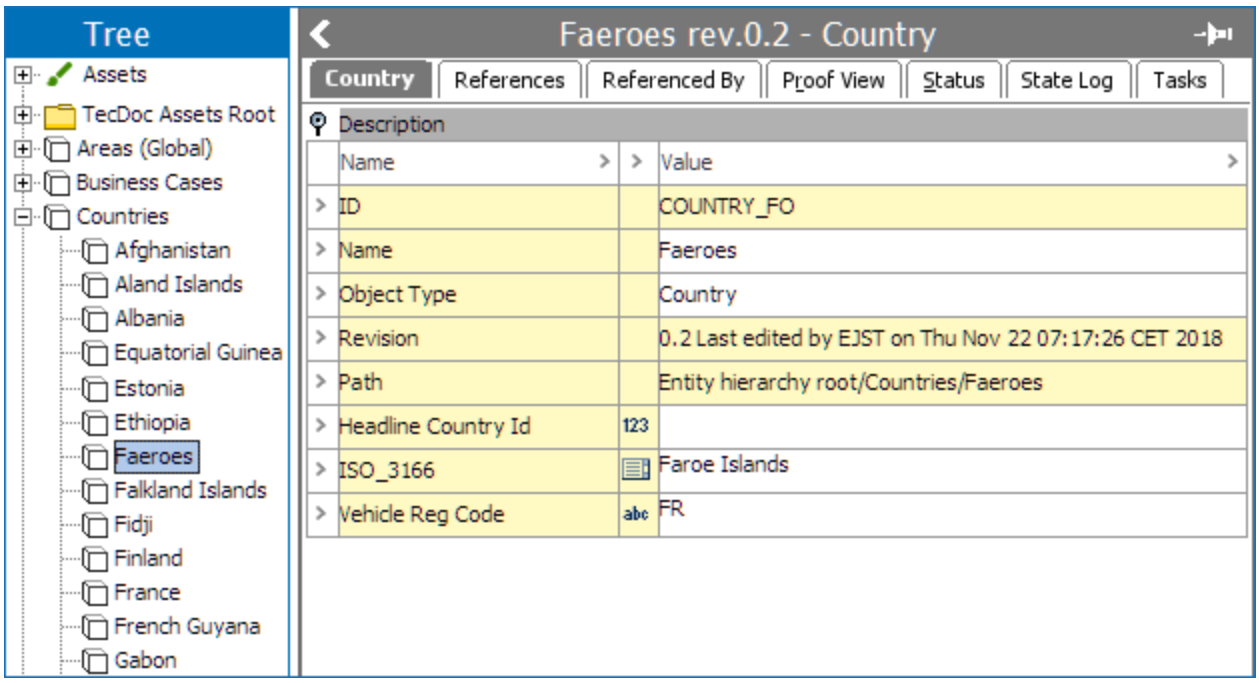

5. Add the object types, attribute, and reference type that were created in steps 1, 2, and 4 to the Automotive - Country Model configuration



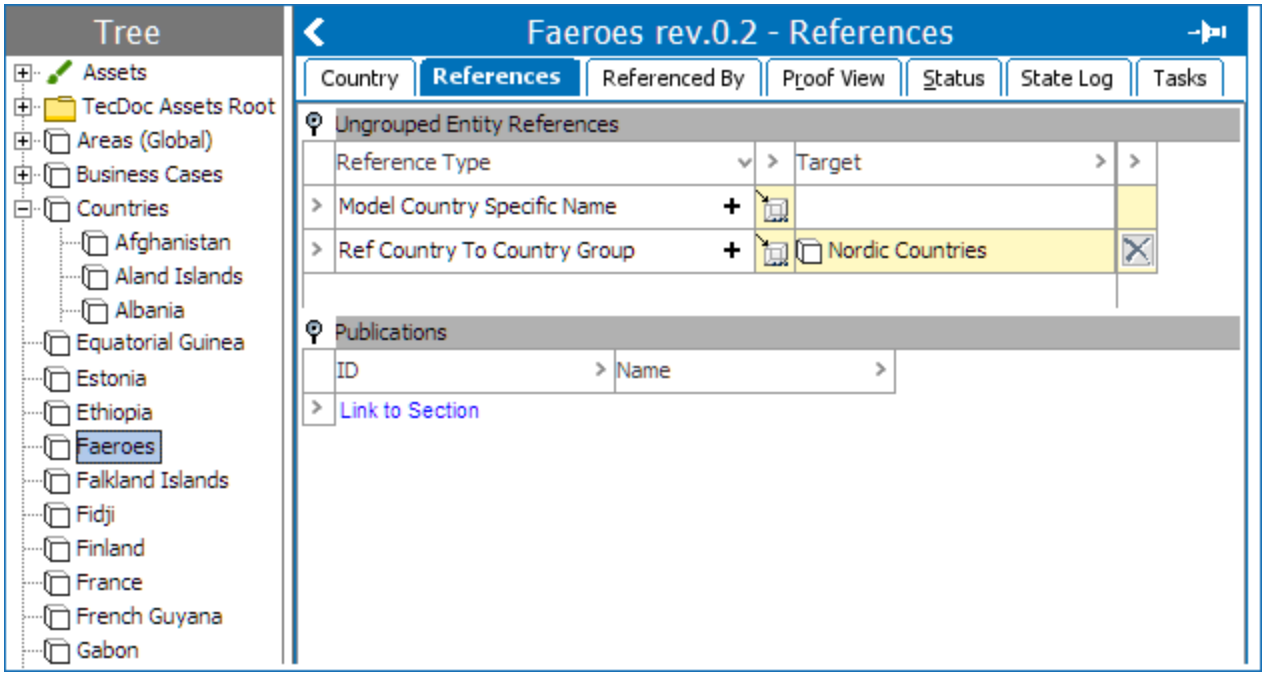

6. Create the Country Group objects and populate the country attribute value for each Country Group




7. Create the Country objects and populate the country attribute value for each Country

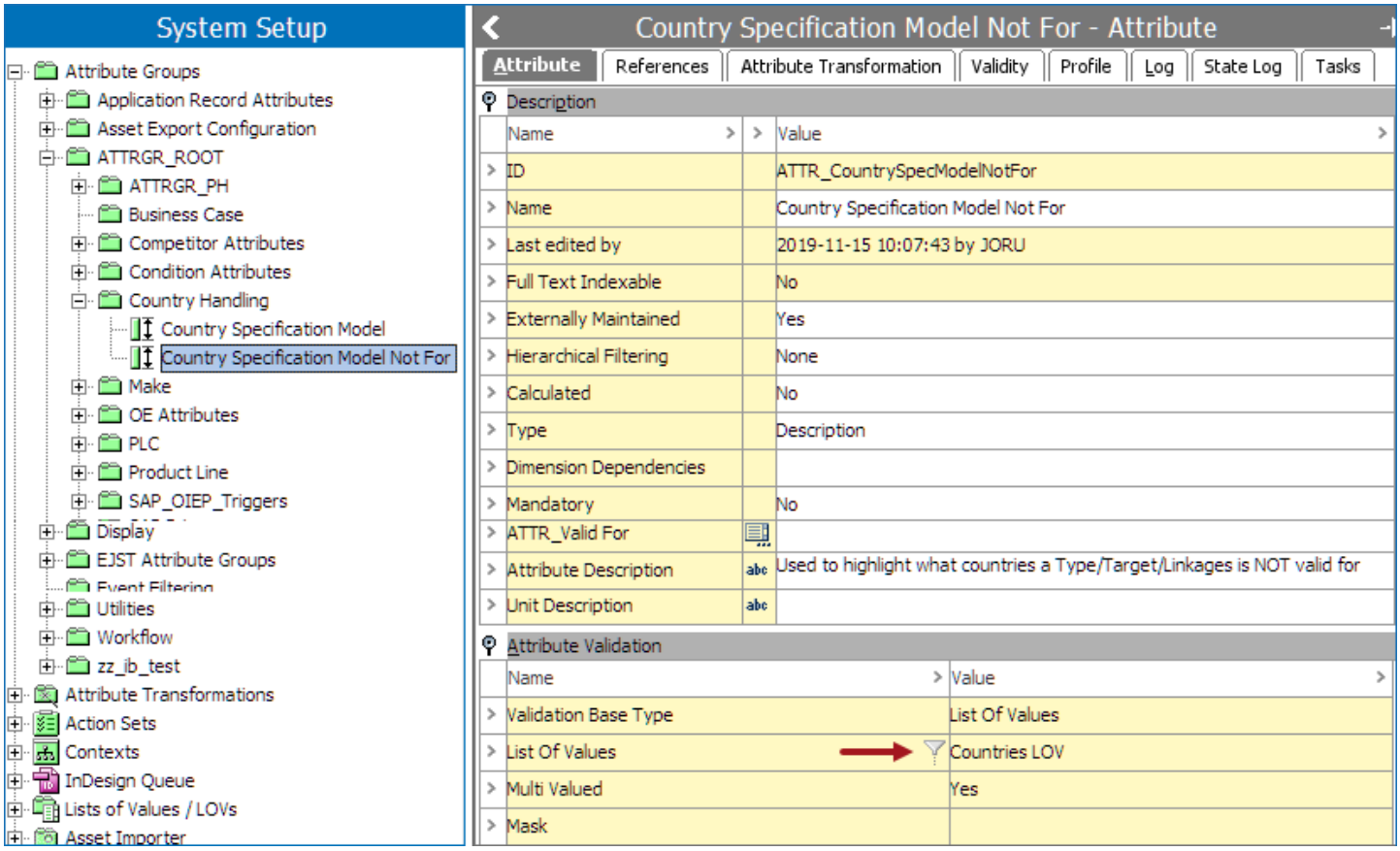


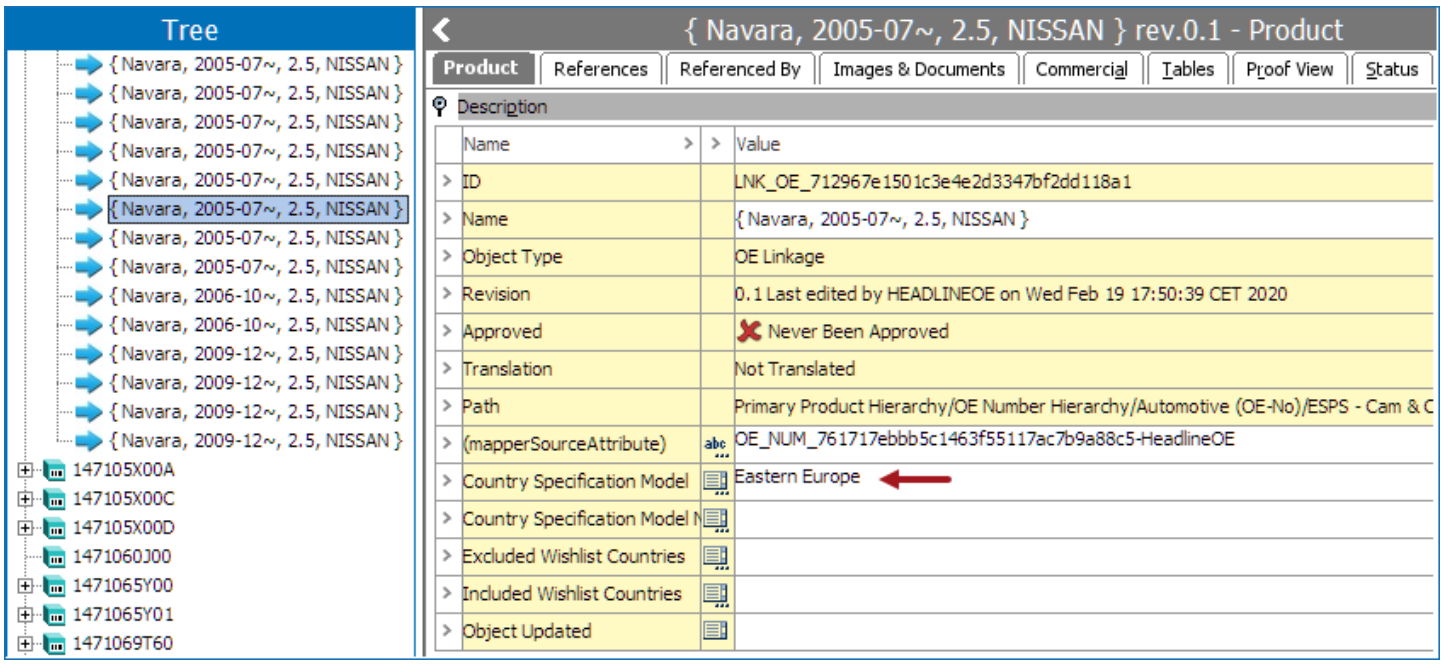

8. Link the Country object to a Country Group (if it should be grouped) by using the reference type that was created in step 2. (If a Country object is not linked to a Country Group, then it will be displayed below 'All Countries' in the Web UI.)



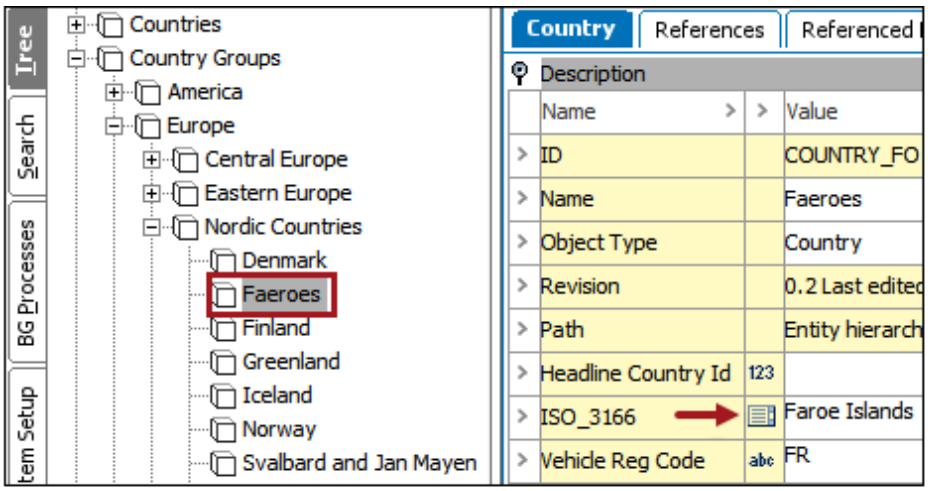

9. Create an attribute for included and excluded Countries by using the same LOV from step 3 and make the attributes valid for any object type that should have the values stored on, for example, valid for a part number or for an OE number. These two attributes will be used by the configuration within the Country Handling Web UI components to store the countries that have been selected for inclusion or exclusion.

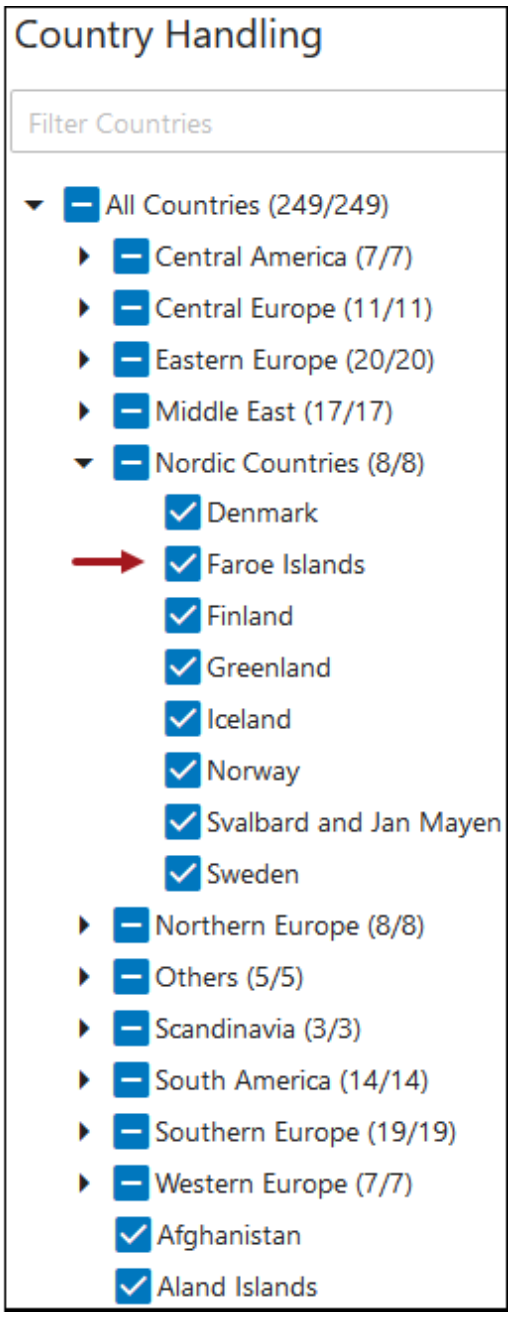
- These attributes must have Multi Valued = Yes.
- Both of these attributes need to use the same List Of Values that was created in step 3 (Countries LOV). The first screenshot below is an example of an attribute that has been created to store excluded Countries.
- The second screenshot below shows an example of the 'Eastern Europe' Country Group that has been selected to be included for the OE Linkage object.






10. Configure the Country Handling Web UI components in order to display the Countries and Country Groups to be selected for inclusion or exclusion. For more information on how to configure these components, refer to the **Configuring Application Country Handling Table Header** and the **Configuring Country Handling Value Component** topics. Once the Country Handling Value component has been configured, then the Country Groups and Countries will be available to be selected / deselected like in the screenshot below on the right.
    - The Country LOV Value will be displayed in the Web UI, not the STEP Name that is given to the Country object. In the example below, 'Faeroes' is the STEP Name for the Country object, but the LOV Value for the ISO_3166 country attribute is 'Faroe Islands' therefore, 'Faroe Islands' is the name that will be displayed in the Web UI.
    - If a Country object is not linked to a Country Group, then it will be displayed below 'All Countries,' such as 'Afghanistan' and 'Aland Islands' in the screenshot below.


Tree
Search
BG Processes
tem Setup
Countries
Country Groups
America
Europe
Central Europe
Eastern Europe
Nordic Countries
Denmark
Faeroes
Finland
Greenland
Iceland
Norway
Svalbard and Jan Mayen
Country
References
Referenced
Description
Name
Value
ID
COUNTRY_FO
Name
Faeroes
Object Type
Country
Revision
0.2 Last edited
Path
Entity hierarch
Headline Country Id
123
ISO_3166
Faroe Islands
Vehicle Reg Code
abc
FR


Country Handling
Filter Countries
All Countries (249/249)
Central America (7/7)
Central Europe (11/11)
Eastern Europe (20/20)
Middle East (17/17)
Nordic Countries (8/8)
Denmark
Faroe Islands
Finland
Greenland
Iceland
Norway
Svalbard and Jan Mayen
Sweden
Northern Europe (8/8)
Others (5/5)
Scandinavia (3/3)
South America (14/14)
Southern Europe (19/19)
Western Europe (7/7)
Afghanistan
Aland Islands

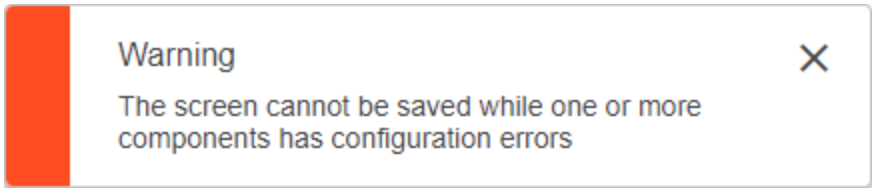
# Configuring Application Country Handling Table Header

When configured, the 'Application Country Handling able 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, refer to the **Prerequisites** section of the **Country Handling Solution** topic.

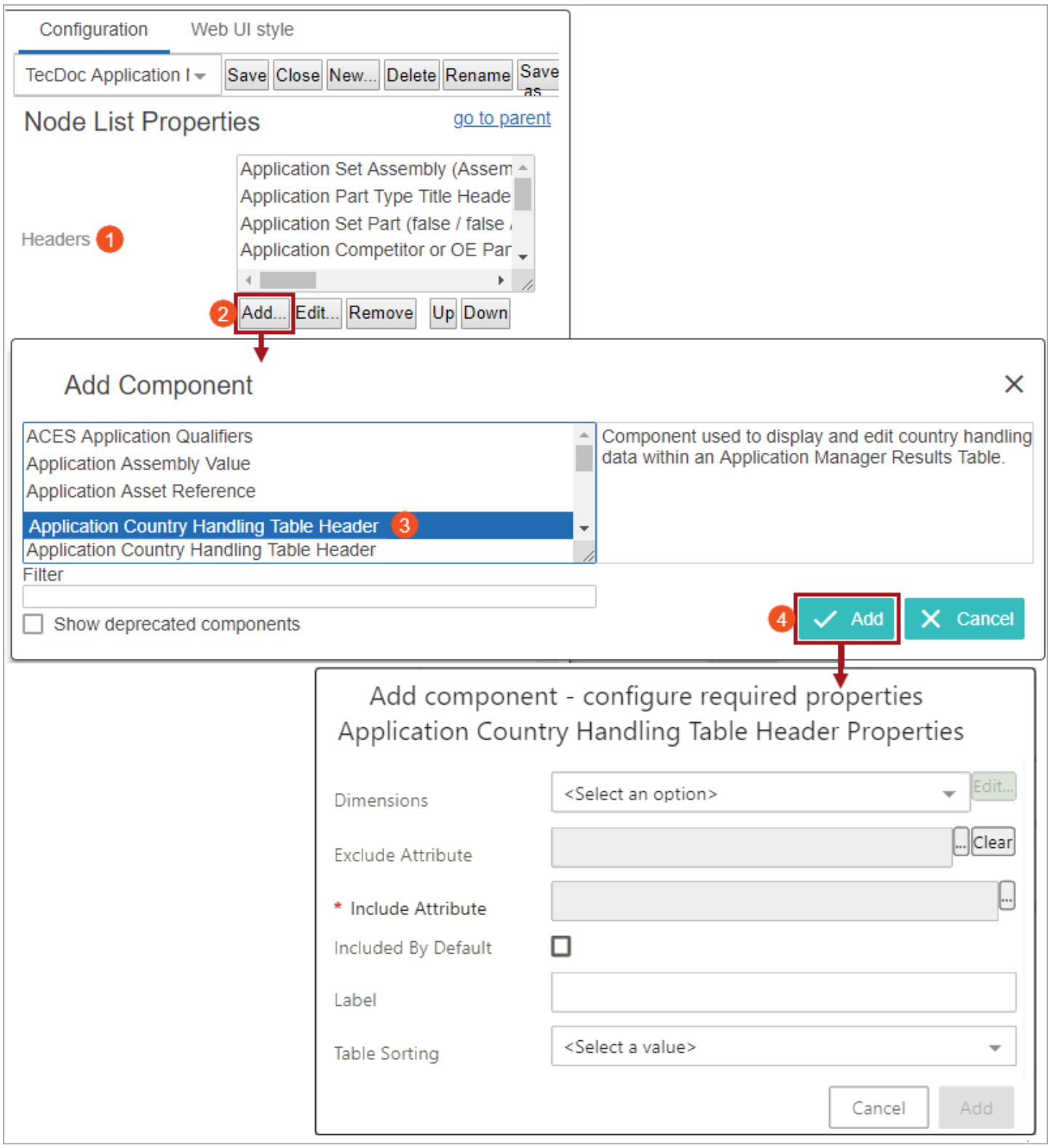
Attempting to add the Application Country Handling Table Header component to a Web UI prior to configuring the Automotive - Country Model component model will result in the following warning: The screen cannot be saved while one or more components has configuration errors.




## 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).




5. Populate the required parameters listed below.

- **Include Attribute:** This 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.'
   - **Included By Default:** By default, the parameter is disabled. This filter option provides user an option to include all countries by default in the Country Inclusion list. When disabled, all countries are listed in the Country Exclusion list and user needs to add the desired countries to the Country Inclusion list. When enabled, all countries are listed in the Country Inclusion list and user deselects the excluded countries from the Country Inclusion list.
   - **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

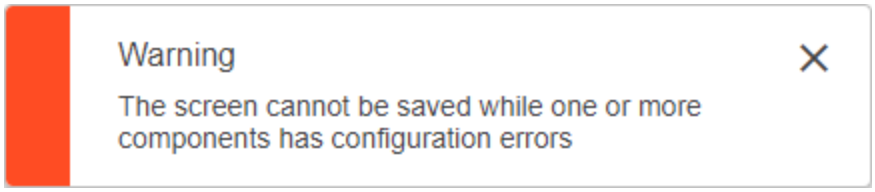
When using the Country Handling solution within a Node Editor, the 'Country Handling Value' component can be configured to display the country inclusion / exclusion data within a Node Editor screen used to view and/or edit country handling data on parts and/or applications (i.e., Product Details screen).

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, refer to the **Prerequisites** section of the **Country Handling Solution** topic.

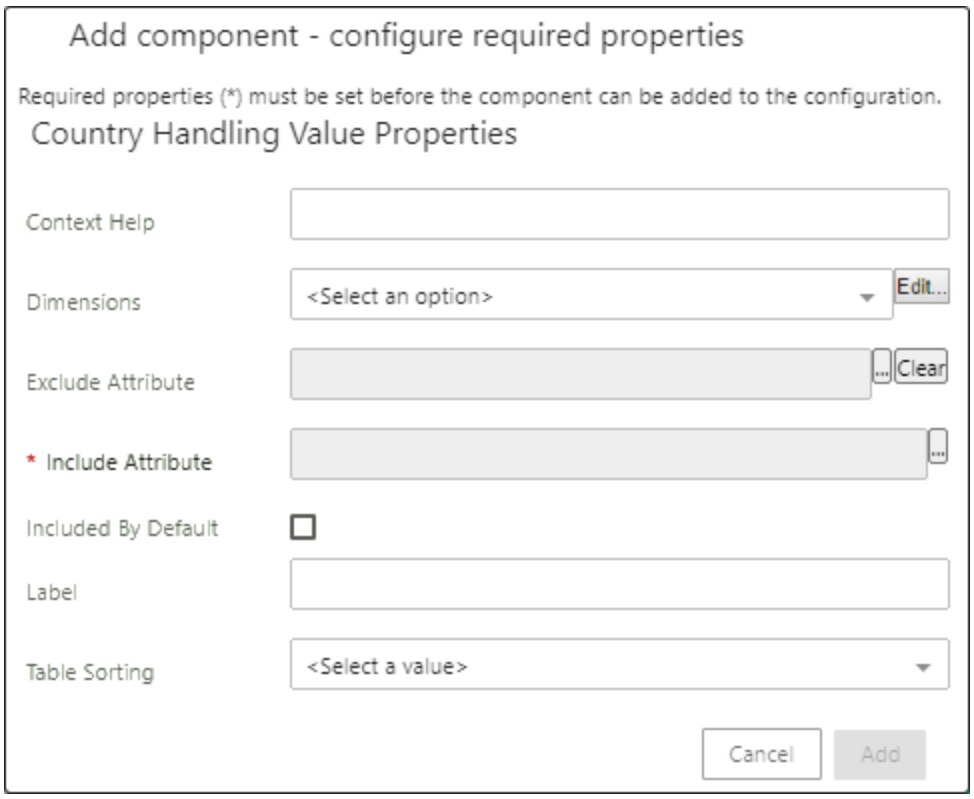
Attempting to add the Country Handling Value component to a Web UI prior to configuring the Automotive - Country Model component model will result in the following warning: The screen cannot be saved while one or more components has configuration errors.




## 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).




4. 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.

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.
- **Dimensions:** By default, the parameter is blank. Optionally, to configure column width and height, select 'Table Header Dimensions.'
- **Exclude Attribute:** This 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.

- **Included By Default:** By default, the parameter is disabled. This filter option provides user an option to include all countries by default in the Country Inclusion list. When disabled, all countries are listed in the Country Exclusion list and user needs to add the desired countries to the Country Inclusion list. When enabled, all countries are listed in the Country Inclusion list and user deselects the excluded countries from the Country Inclusion list.
- **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.
- **Table Sorting:** By default, the parameter is blank. Optionally, select from the dropdown values to specify the default sorting order of the column data.

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.

# Data Onboarding

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 be setting up mapping configurations to onboard or offboard 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

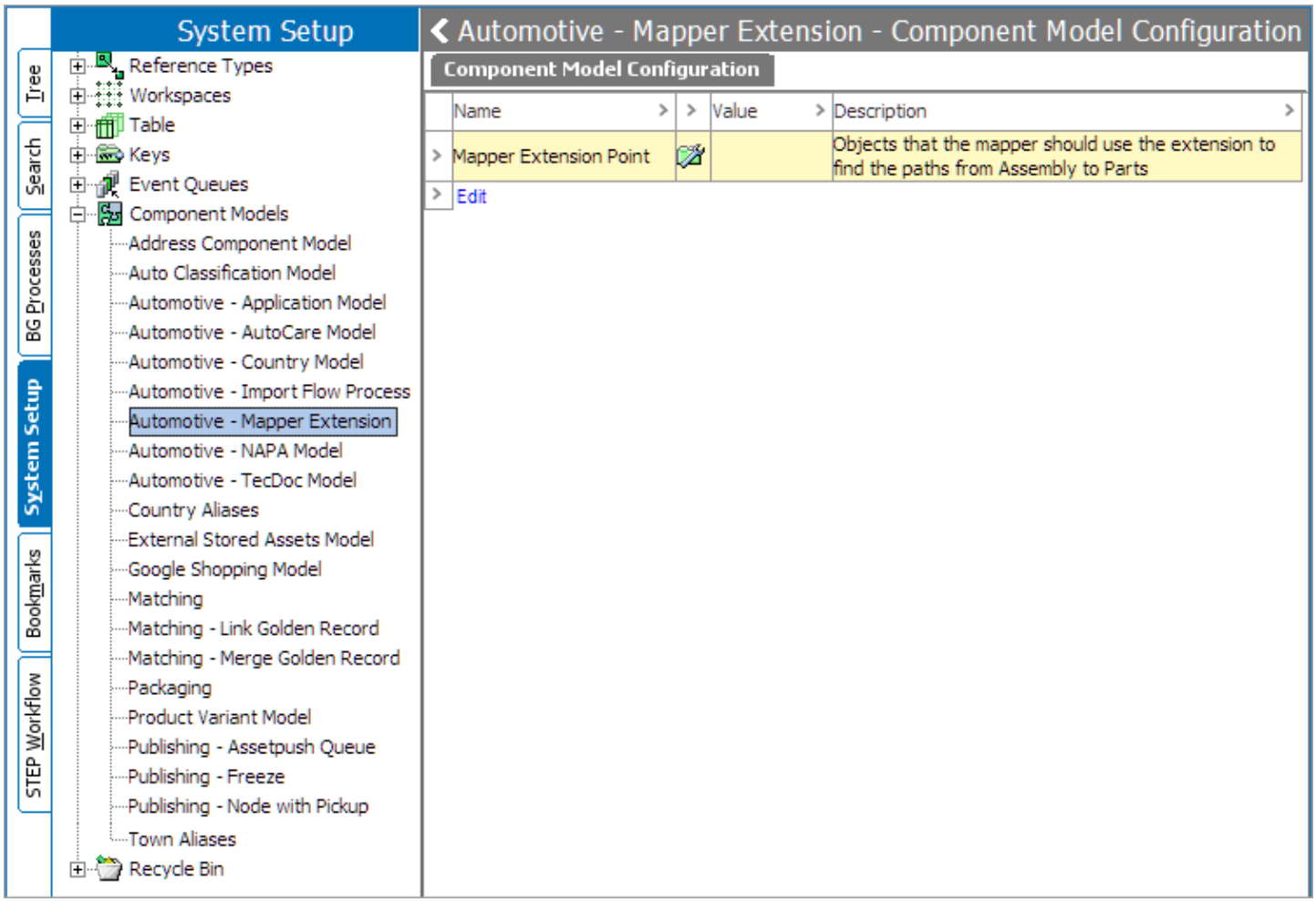
The following topics provide the configuration processes necessary to allow users to be able to set up the Data Onboarding solution. The initial part of the configuration is done in the workbench and the later part is configured in the Web UI.

- 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).



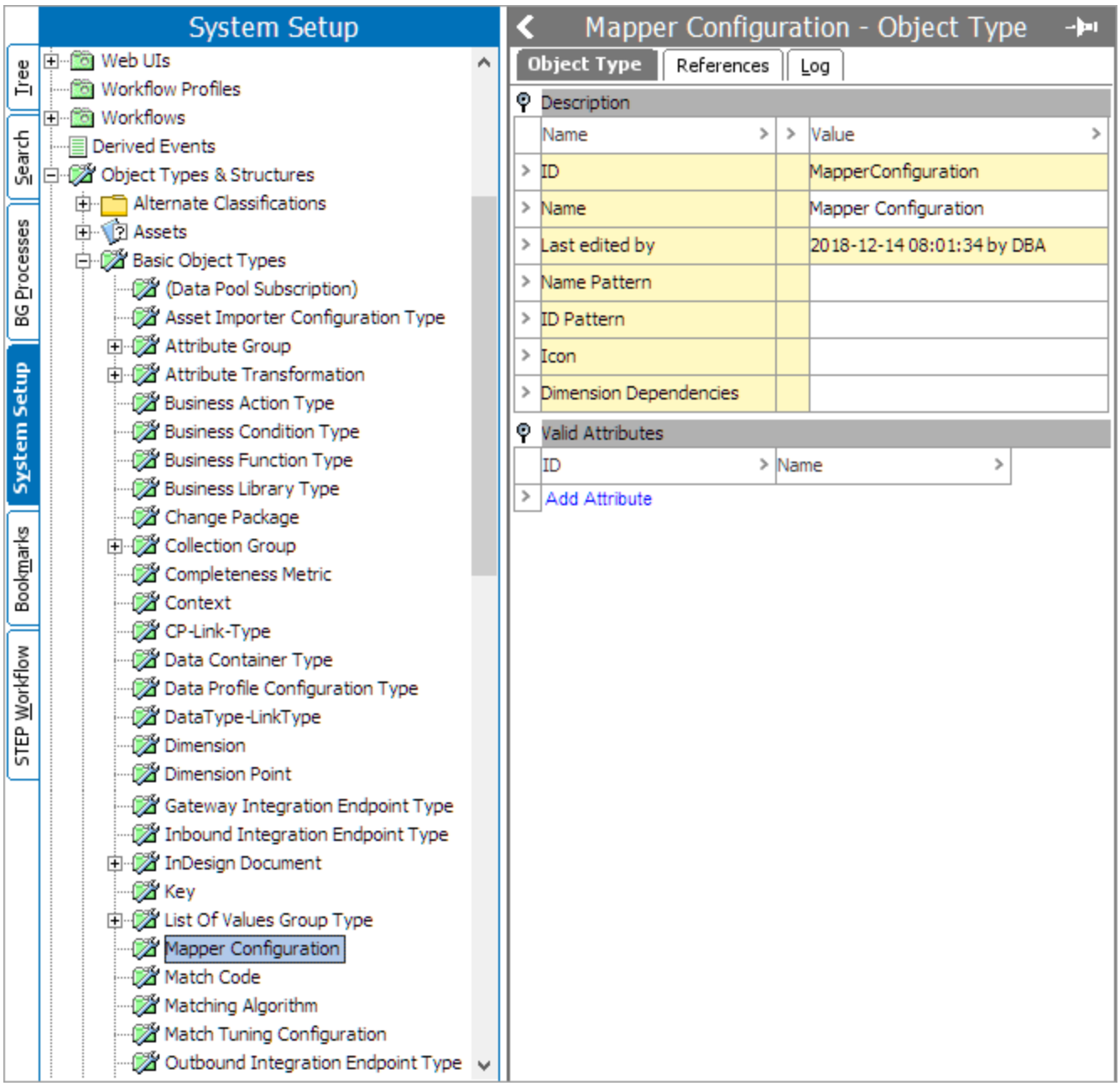
## Data Onboarding Solution Initial Setup

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.

Setup entity definitions can be exported as comments and submitted to an external source control system for comparison purposes. For details, refer to the Configuration Management documentation.

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, refer to 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.

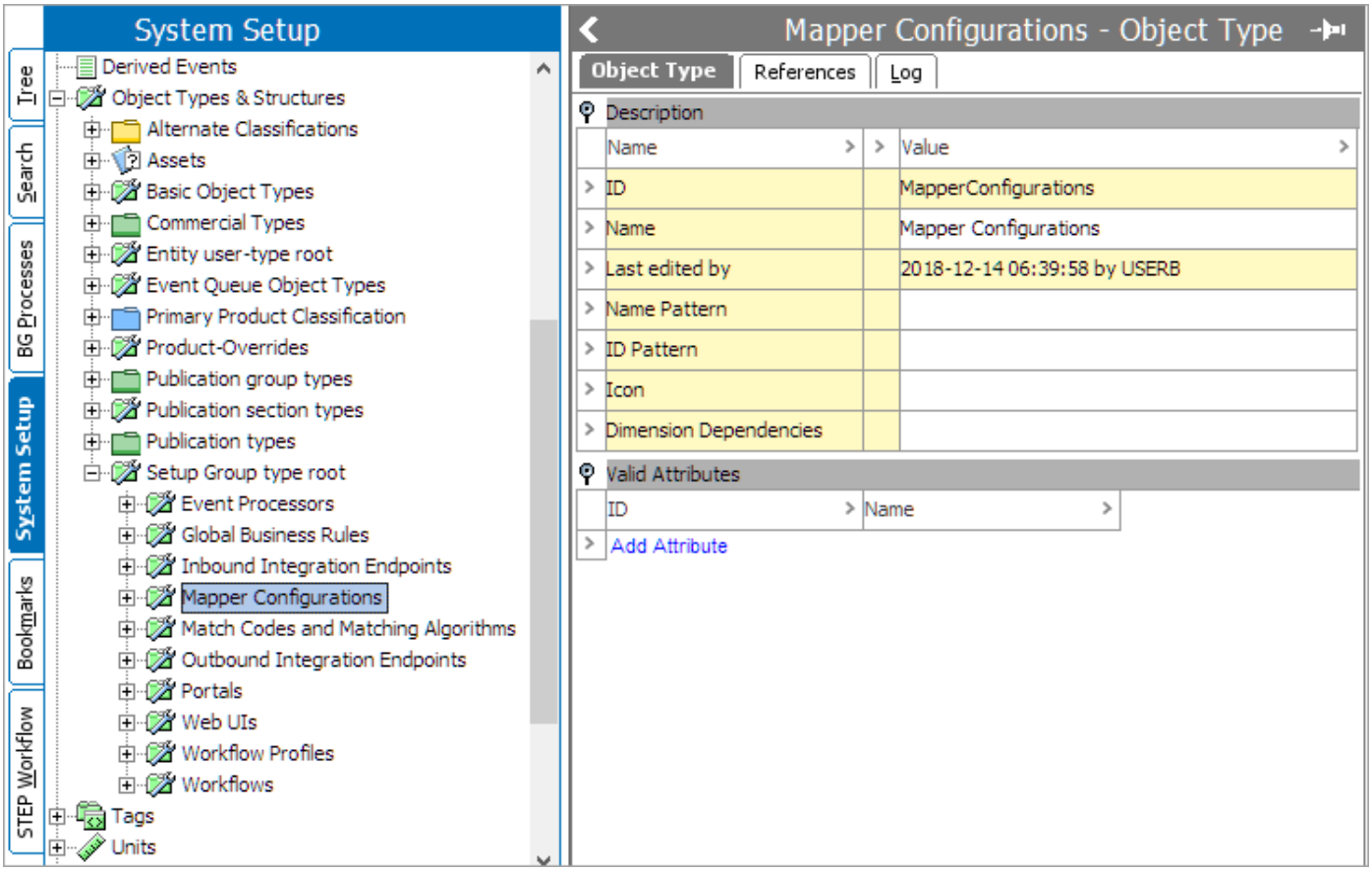



Below are the steps for an initial setup in the workbench.

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.

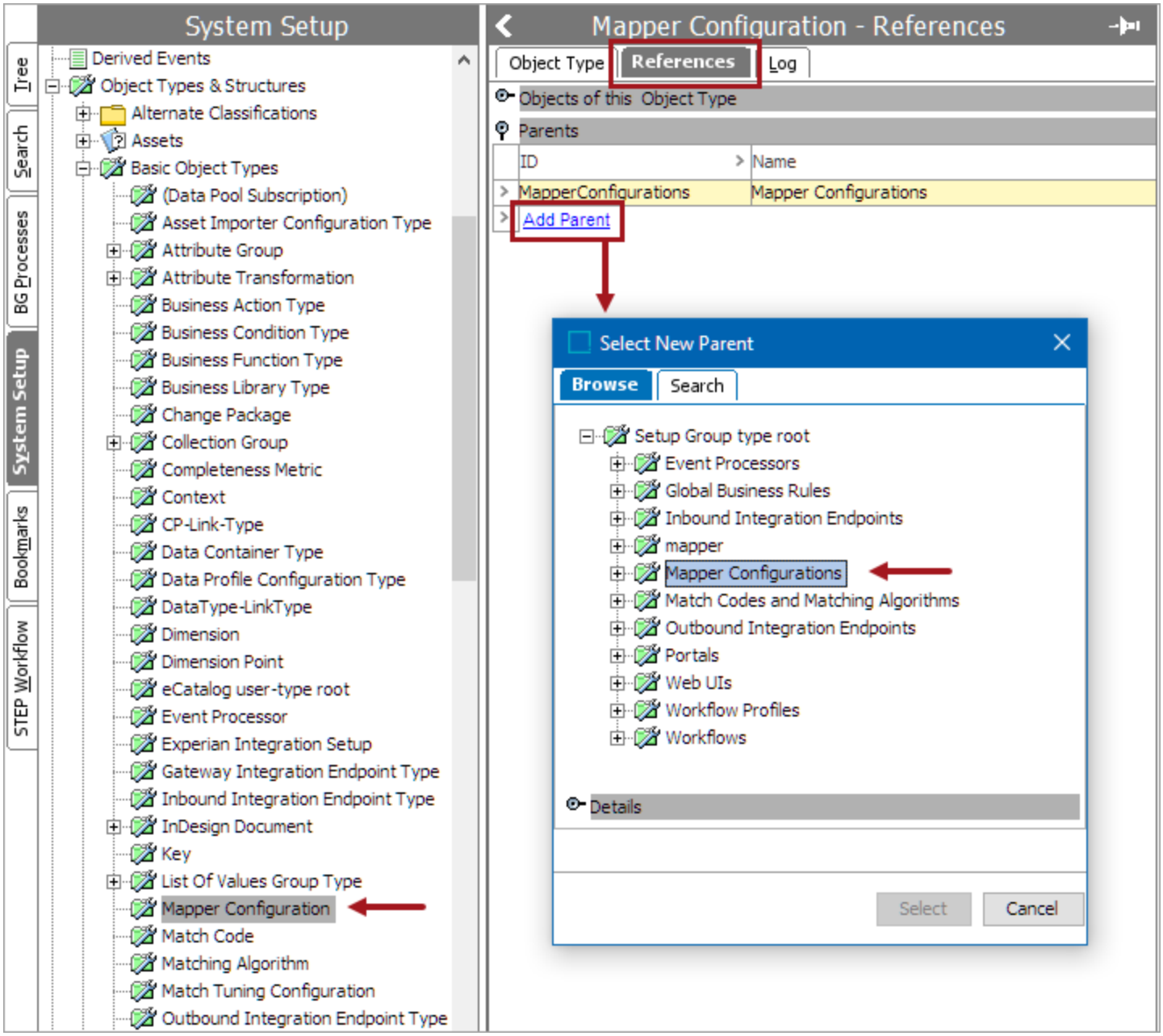
The sample below illustrates a hierarchy of setup group object types. A Setup Group named 'Mapper Configurations' has been created.




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 Configuration**, and click the **References** tab.

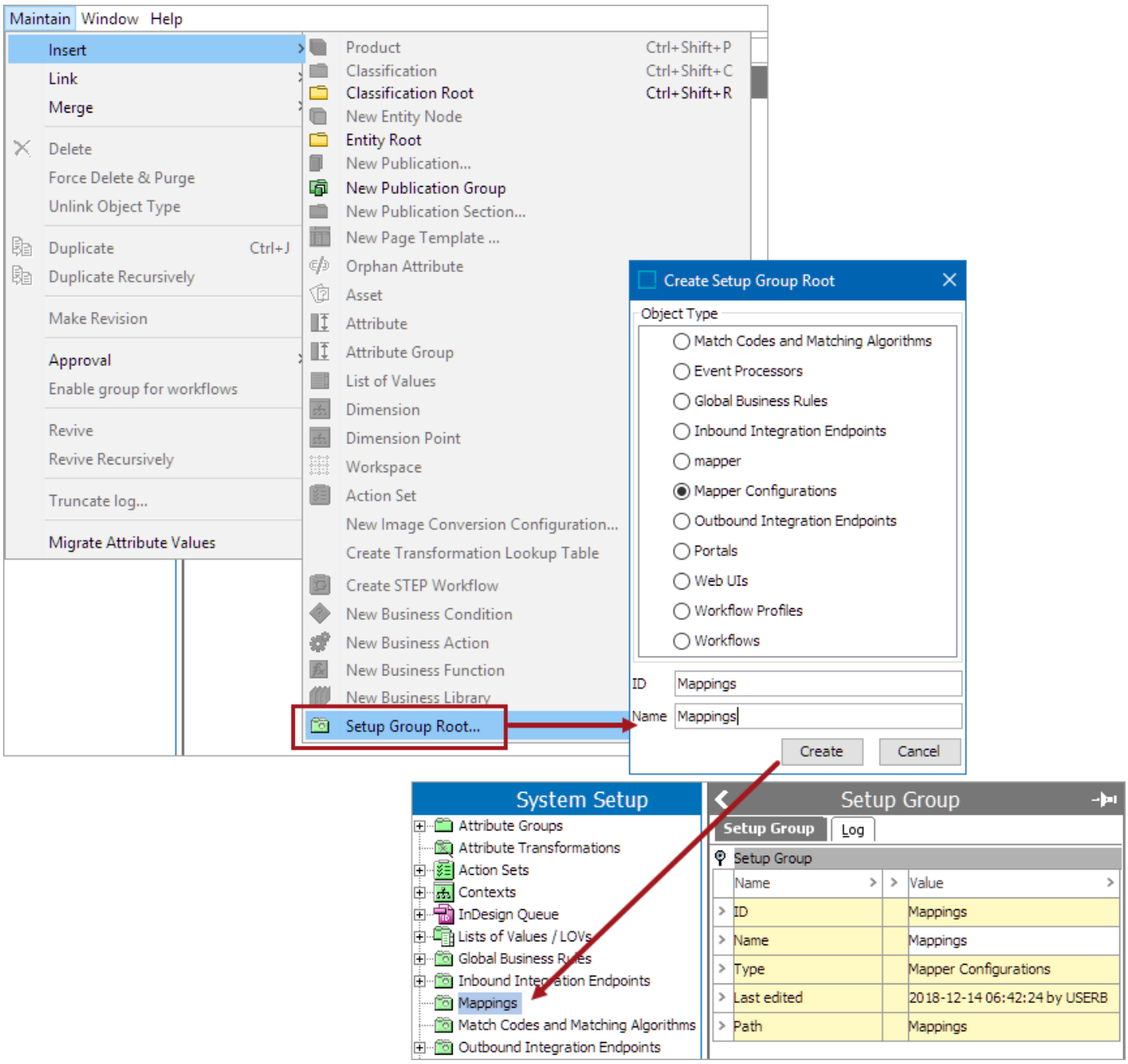
   Under the **Parents** flipper, click **Add Parent**, and in the node selector dialog, specify the **Mapper Configurations** node that was created in the above step.




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**.

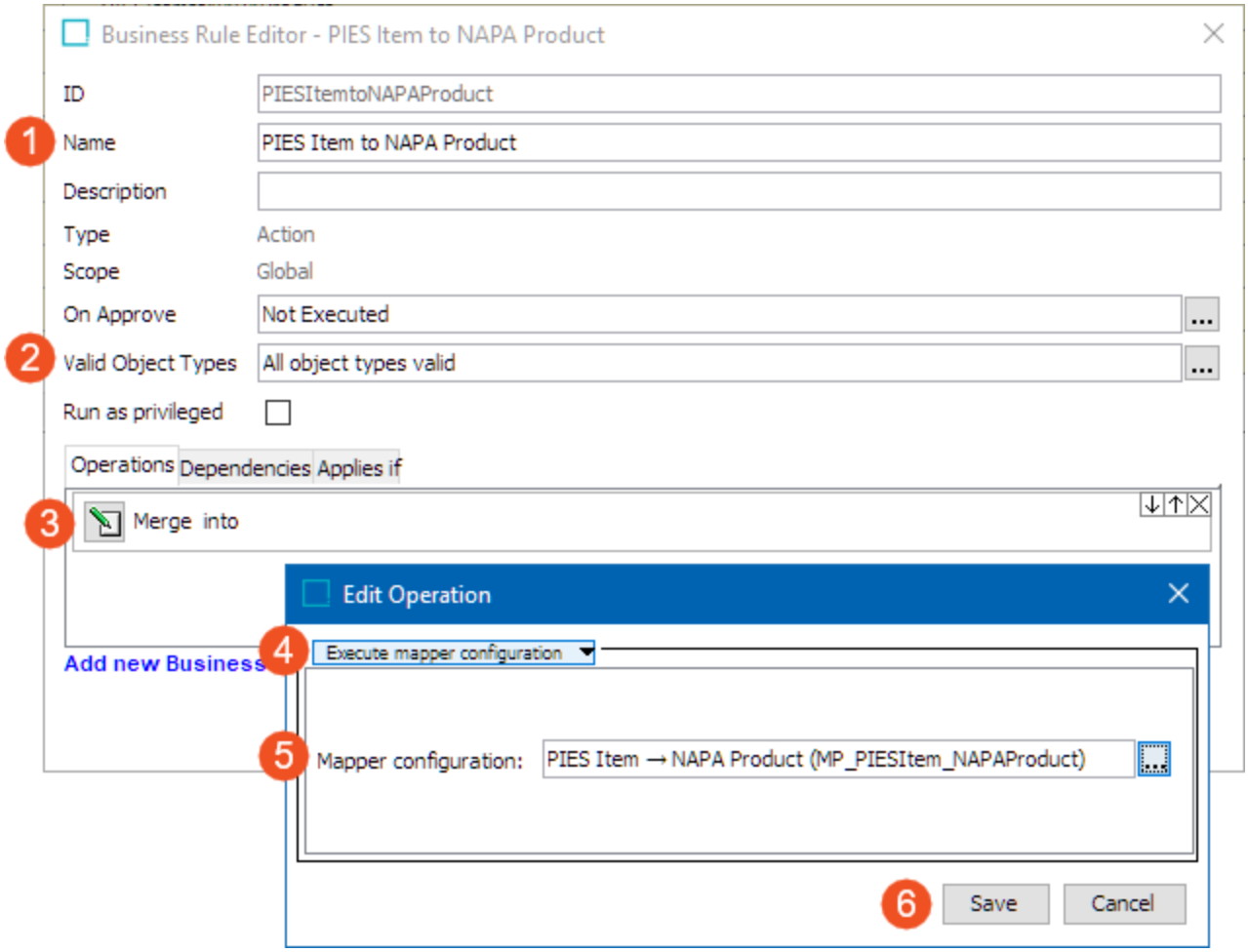



The setup group called 'Mappings' of the object type 'Mapper Configurations' is created. From this newly created node, users can right-click and add new Mapper Configurations.

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 / Offboarding data in Web UI.

## Prerequisites

Designers need to create and configure an Onboarding Mappings Details Screen, to configure Global Navigation 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:

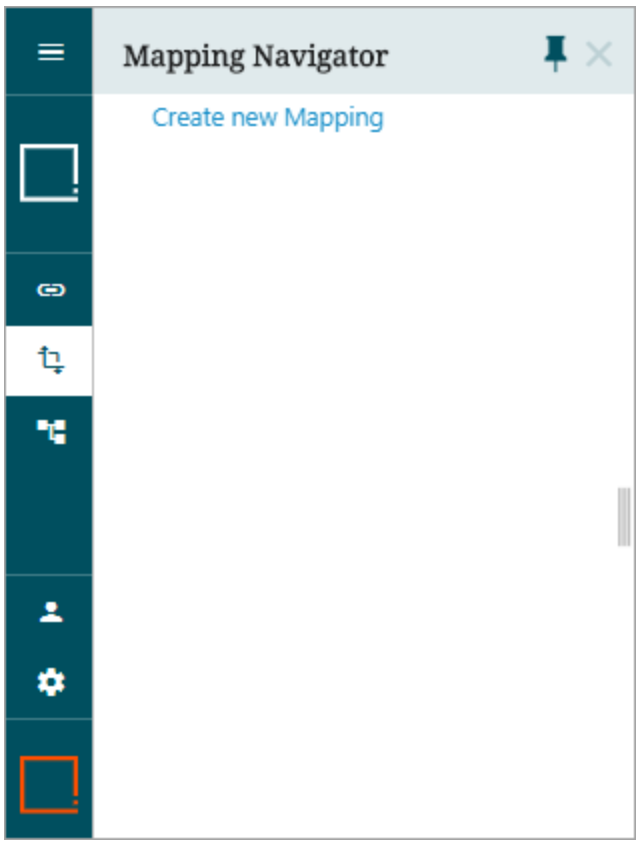
- Onboarding Mappings Details Screen
- Configuring Mapper Configuration Setup Entity
- Using an Onboarding Mappings Details Screen

## 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, refer to **Using an Onboarding Mappings Details Screen** topic.

## Mapper Configurations - Global Navigation Panel Component

The Global Navigation 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 Global Navigation 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 within the Mappings Navigator component of the Global Navigation Panel. For more information, refer to **Onboarding Mappings Details Screen** topic.

## Onboarding Mappings Details Screen

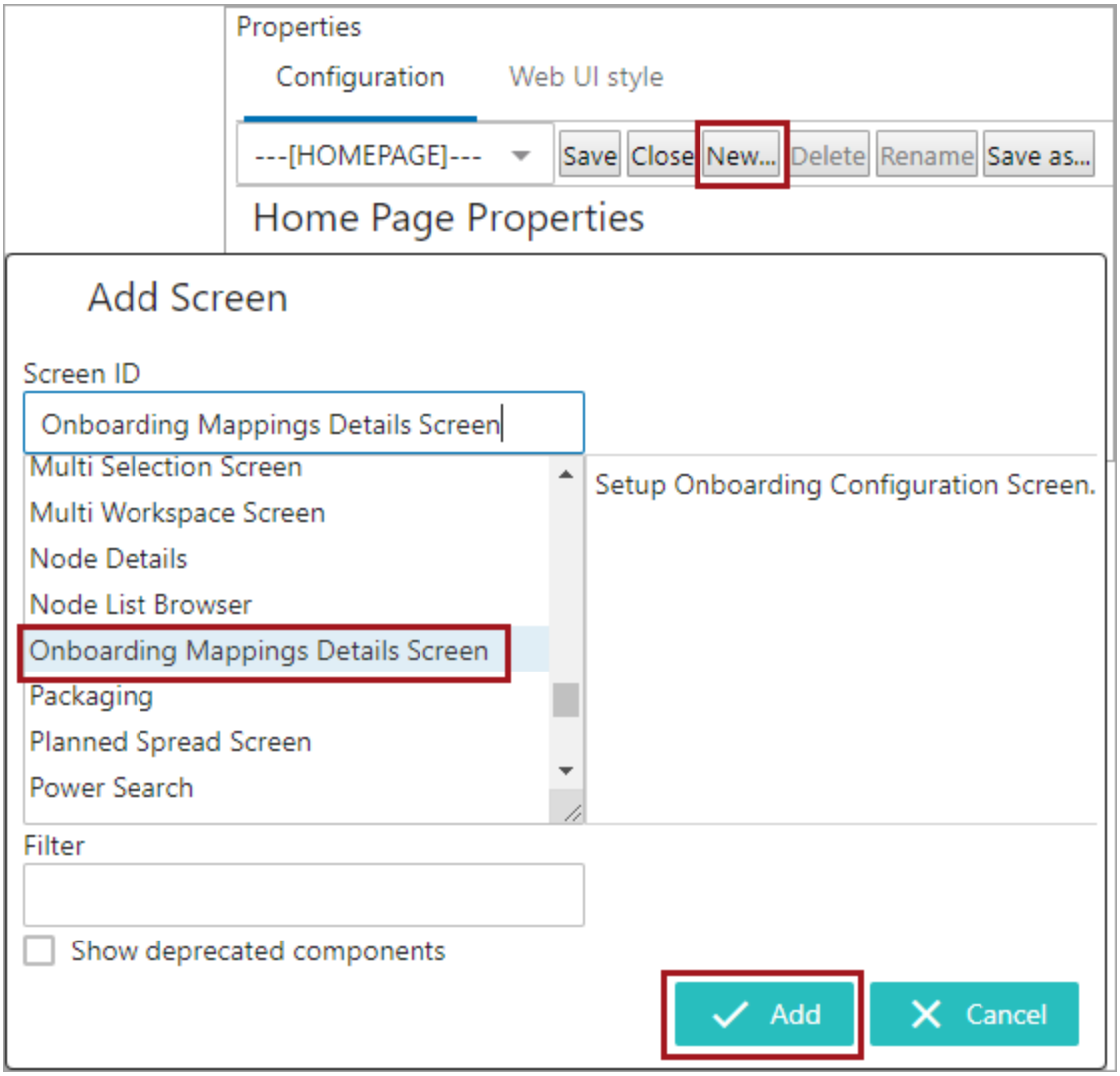
The Onboarding Mappings Details screen allows users to quickly display mapping configurations. The **Onboarding mappings navigator** component functions as a Global Navigation Panel component. For more information about Global Navigation Panel, refer to the Global Navigation Panel topic in the Web User Interfaces documentation.

### Configuration Prerequisites

It is expected that anyone configuring 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 documentation.

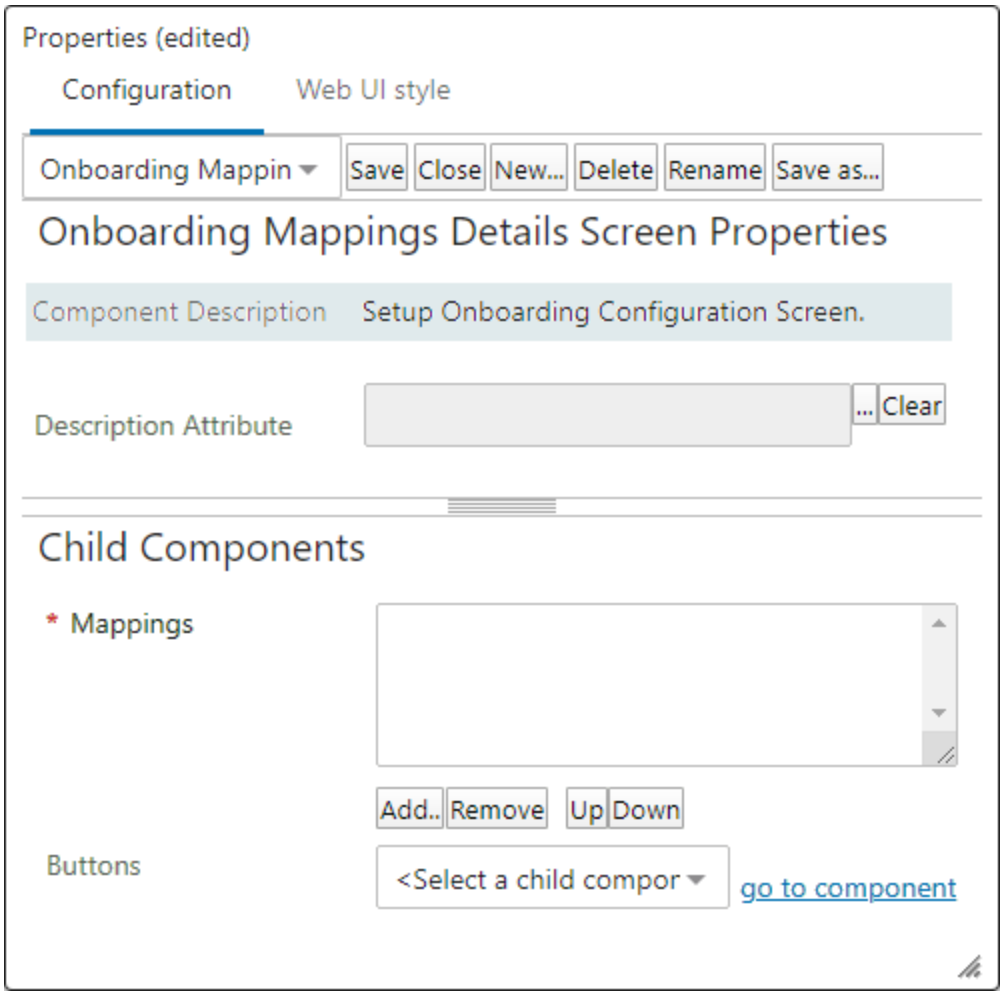
### Configuration Process

1. In the designer, create a new screen by selecting '**New**.'

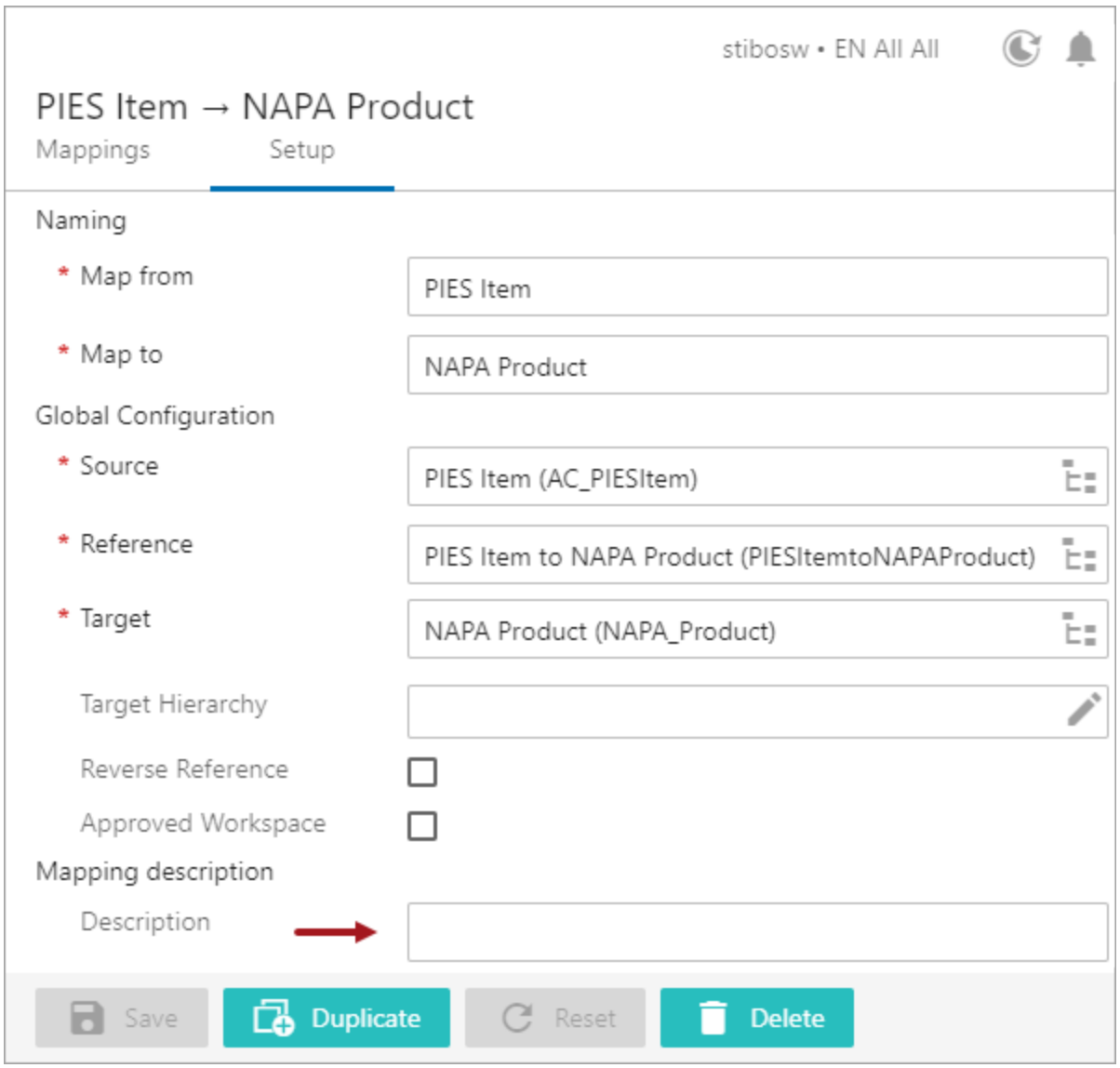


2. Choose '**Onboarding Mappings Details Screen**' from the options.
3. Assign a meaningful name to the screen in the 'Screen ID' field. In the example below, the Screen ID is 'Onboarding Mappings Details Screen'.
4. Complete the process by clicking '**Add**.'

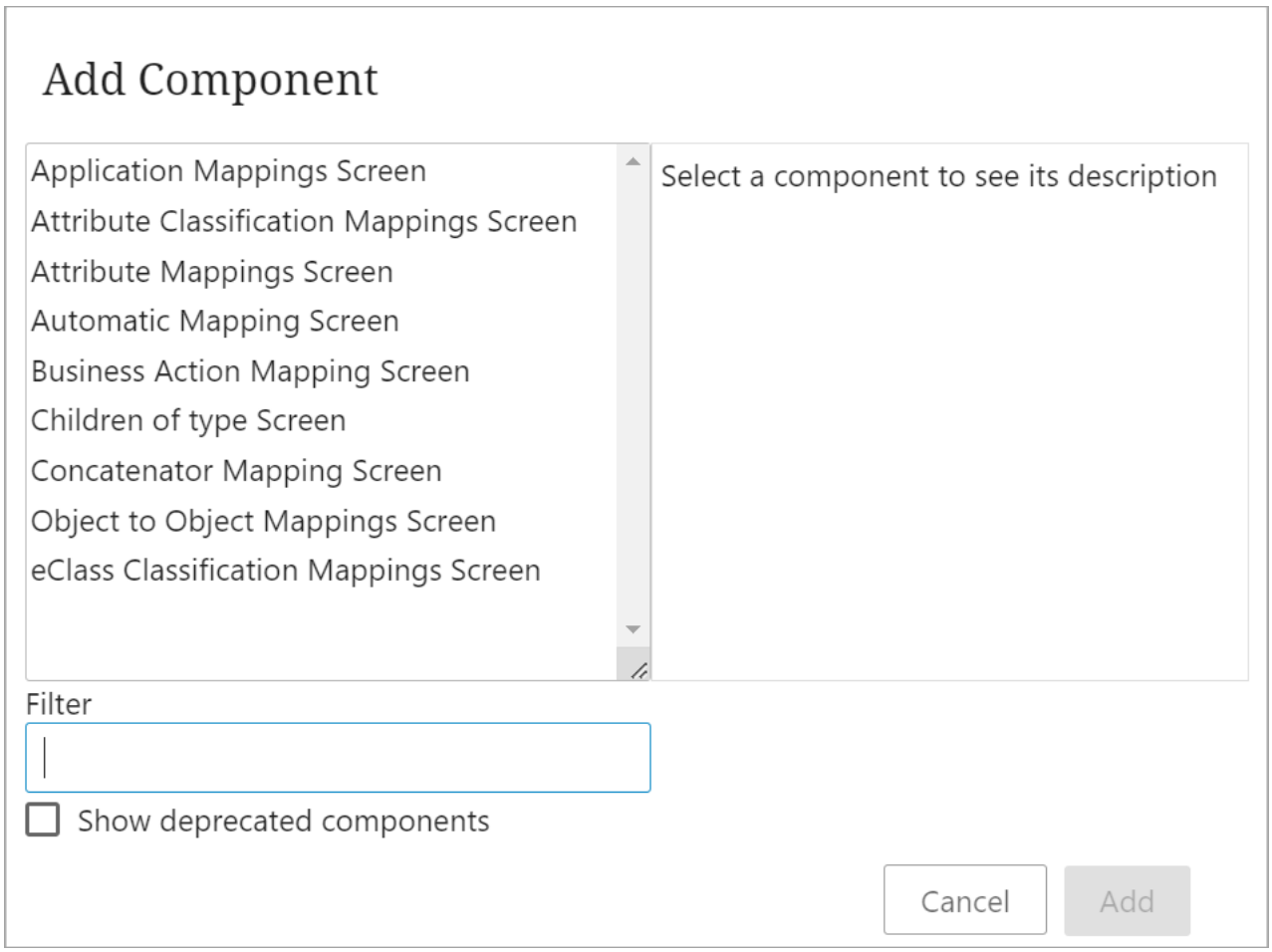
After the creation of the screen, navigate to the 'Onboarding Mappings Details Screen Properties' designer screen to find the following parameters available for configuration:



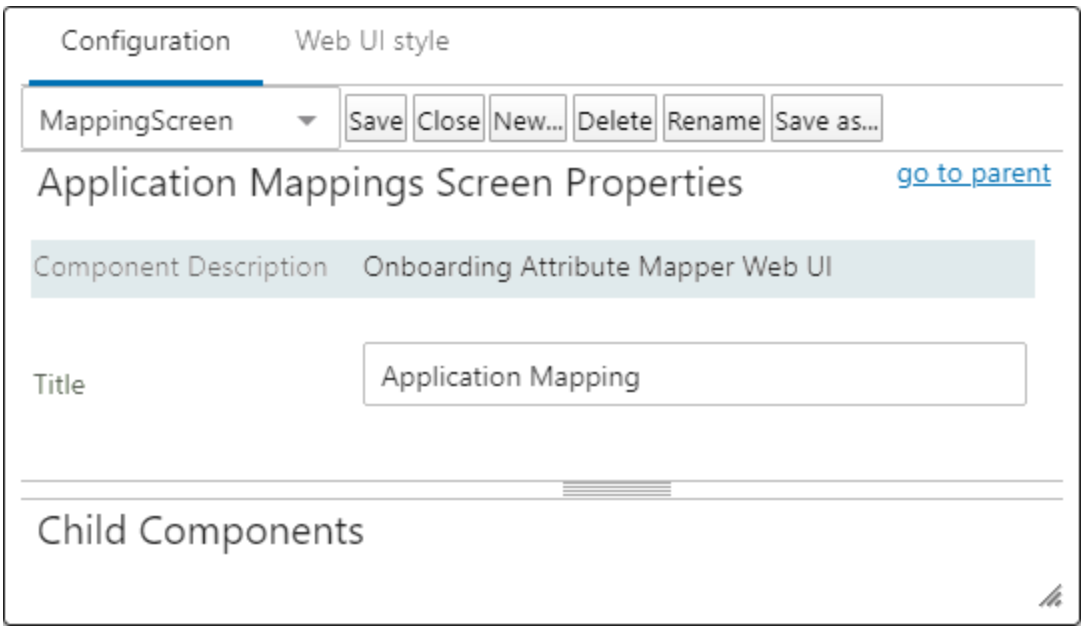

**Description Attribute:** This parameter enables users to set an attribute designated for the description text in the Mapper Configuration setup entity. The chosen attribute, valid for the Mapper Configuration object type, stores the text entered in the 'Description' field on the 'Setup' tab of the Onboarding Mappings Details screen.



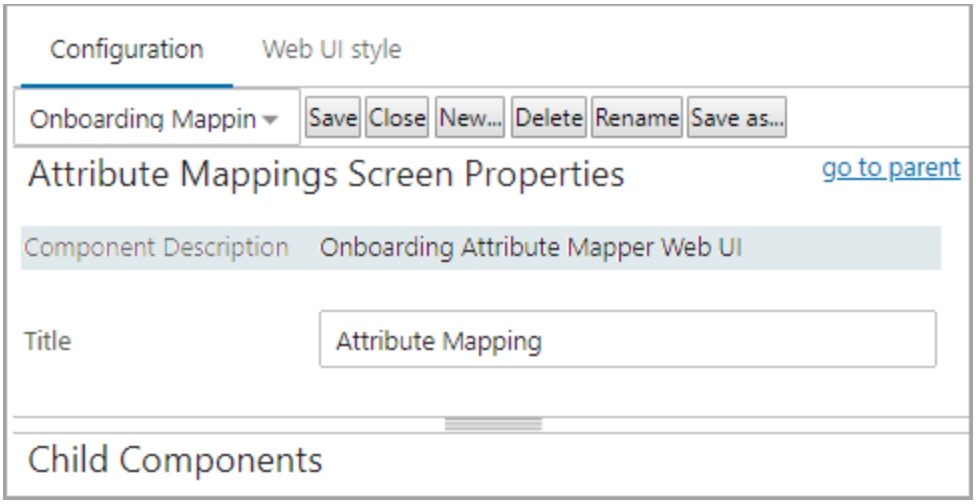

**Mappings:** By default, this required parameter is blank. Click on the 'Add' button to populate it with mapping plugins. This allows users to configure a Mapper Configuration setup entity using the plugins that have been added.



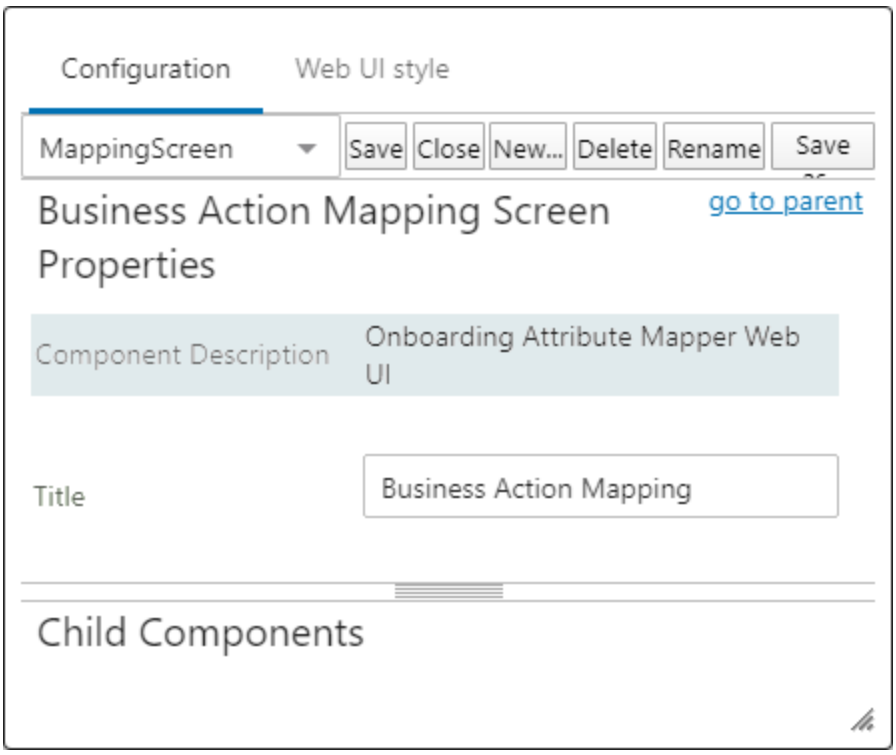

**Application Mappings Screen:** This specific mapping plugin is tailored for use within the PMDM for Automotive solution. When configured in a Mapper Configuration setup entity, it facilitates the extraction of data from applications within the Source object to generate new applications within the Target object. Additionally, it establishes references between the Target applications and both Target vehicles and Target part types. To modify the plugin name, if needed, double-click on the Application Mappings Screen in the Mappings field and adjust the value in the Title parameter.



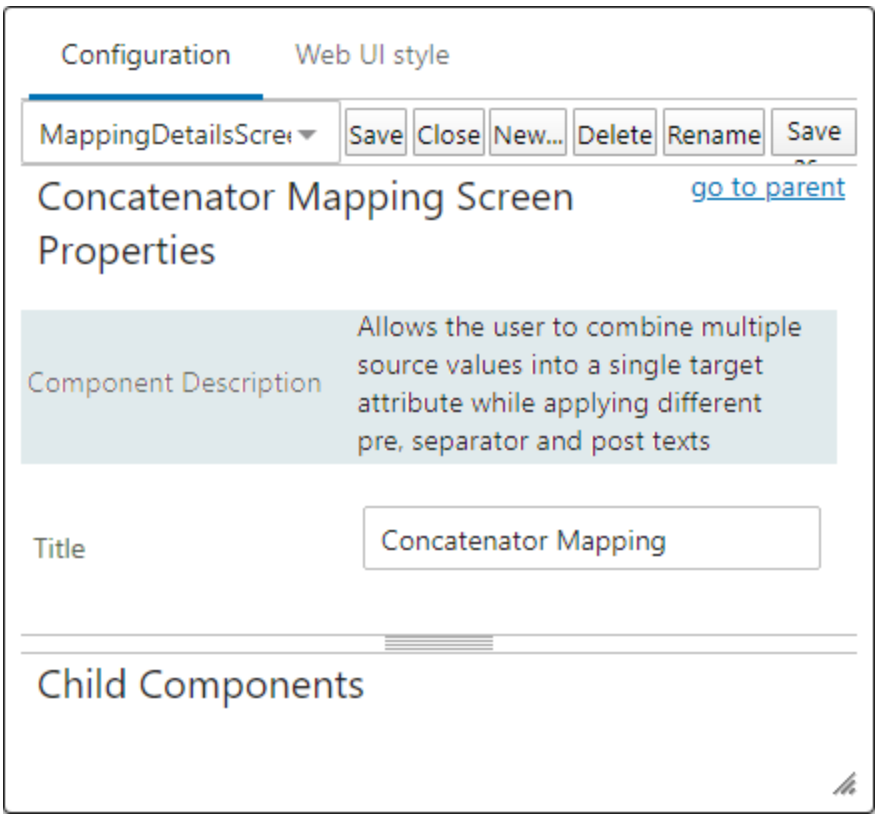

**Attribute Mappings Screen:** When configured into a Mapper Configuration setup entity, this plugin retrieves attribute values from the Source object and populates the corresponding attributes in the Target object. To modify the plugin name, if needed, double-click on the Attribute Mappings Screen in the Mappings field and adjust the value in the Title parameter.




**Business Action Mappings Screen:** This plugin, when configured in the Mapper Configuration, provides an extended flexibility for users to configure and solve unique mapping cases which cannot be achieved by the other Mapping plugins. The Business Action Mapping plugin takes a business action as a configuration and when executed, executes the business action on the target object.




**Concatenator Mappings Screen:** This plugin, when configured in the Mapper Configuration, it allows the user to copy and combine multiple source values into a single target attribute. This plugin is mostly used while applying different pre, separator and post texts.




**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.

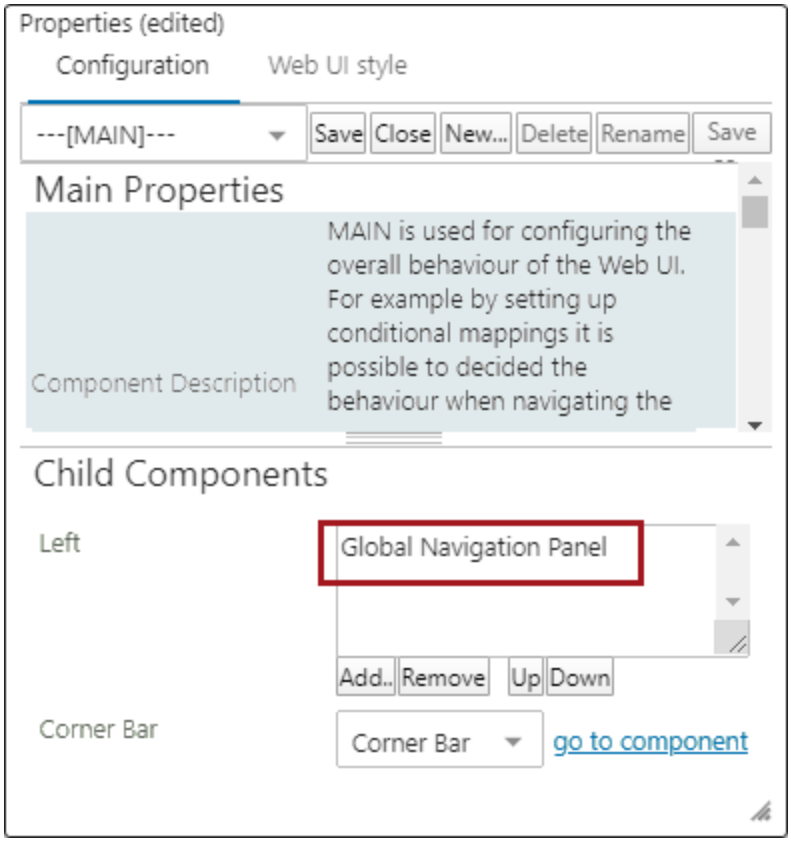
### Implementing the Onboarding Mappings Details Screen

Once the 'Onboarding Mappings Details Screen' is created, implementation of the screen requires the following setup:

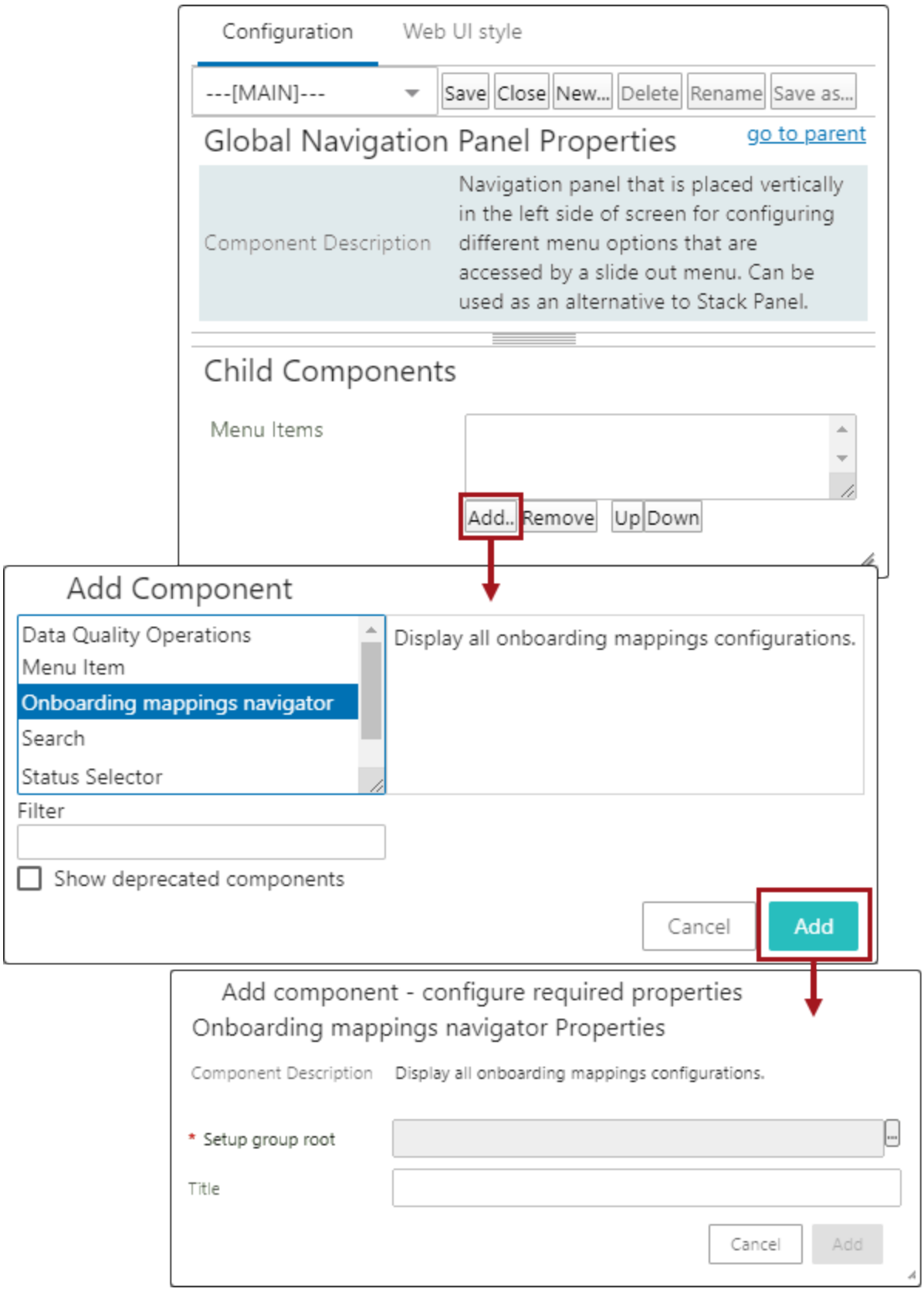
- A Global Navigation Panel needs to be configured to access the Mapper Configurations
- A Mapper Configuration object type must be configured to display the 'Onboarding Mappings Details Screen' when a Mapper Configuration is selected

**Configuring the Global Navigation Panel**

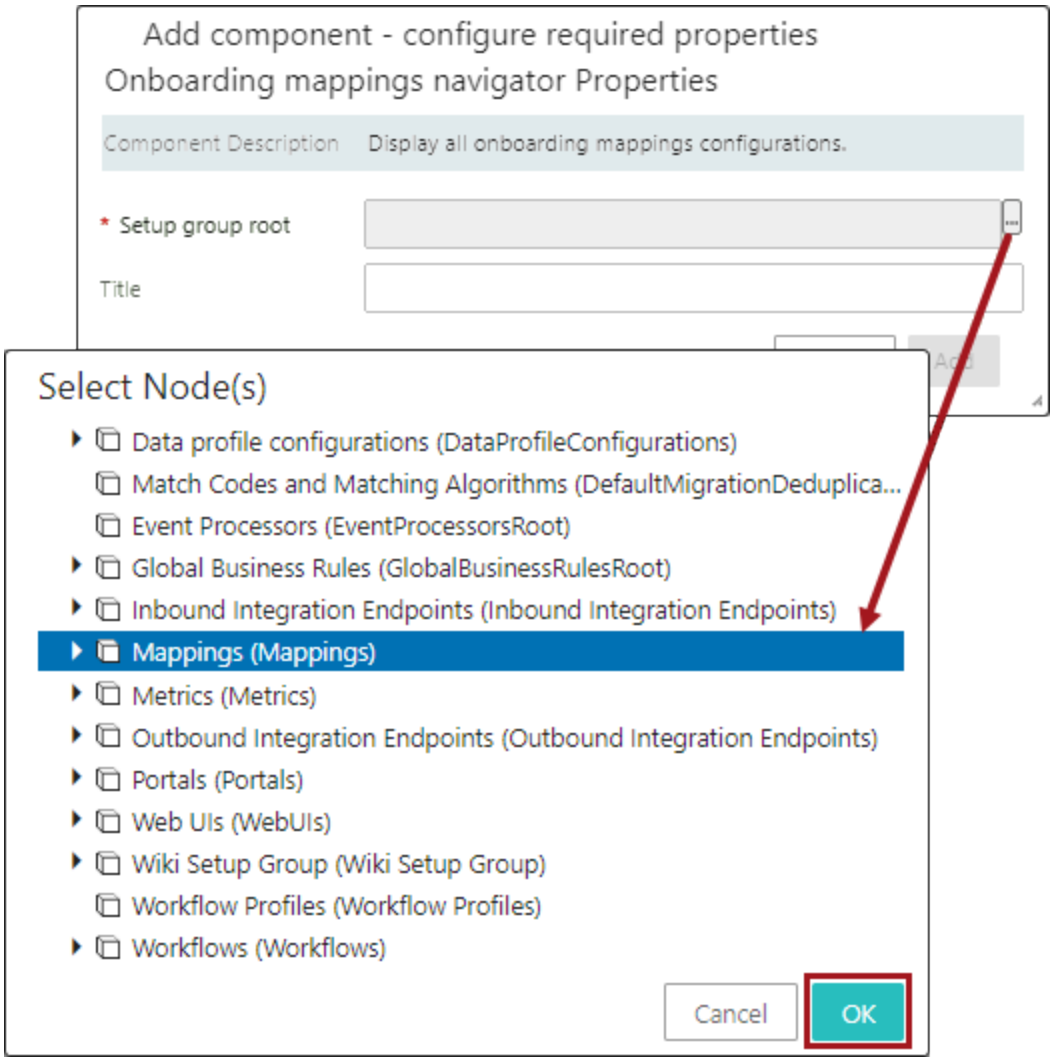
1. From the **Main Properties** screen, add a 'Global Navigation Panel' to the 'Left' section of the child components. Once added, double-click the Global Navigation Panel component. For Information on how to add the Global Navigation Panel, refer to the **Global Navigation Panel** topic in the **Web User Interfaces** documentation of the **STEP Online Help**.




2. From the Global Navigation Panel Properties, select 'Add' in the Menu Items field of the Child Components section. Select 'Onboarding mappings navigator' from this dialog and then click 'Add.' The 'Onboarding mappings navigator Properties' designer screen will be displayed.


Configuration
Web UI style
---[MAIN]---
Save
Close
New...
Delete
Rename
Save as...
Global Navigation Panel Properties
go to parent
Component Description
Navigation panel that is placed vertically in the left side of screen for configuring different menu options that are accessed by a slide out menu. Can be used as an alternative to Stack Panel.
Child Components
Menu Items
Add..
Remove
Up
Down
Add Component
Data Quality Operations
Menu Item
Onboarding mappings navigator
Search
Status Selector
Display all onboarding mappings configurations.
Filter
Show deprecated components
Cancel
Add
Add component - configure required properties
Onboarding mappings navigator Properties
Component Description
Display all onboarding mappings configurations.
* Setup group root
Title
Cancel
Add

3. 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.

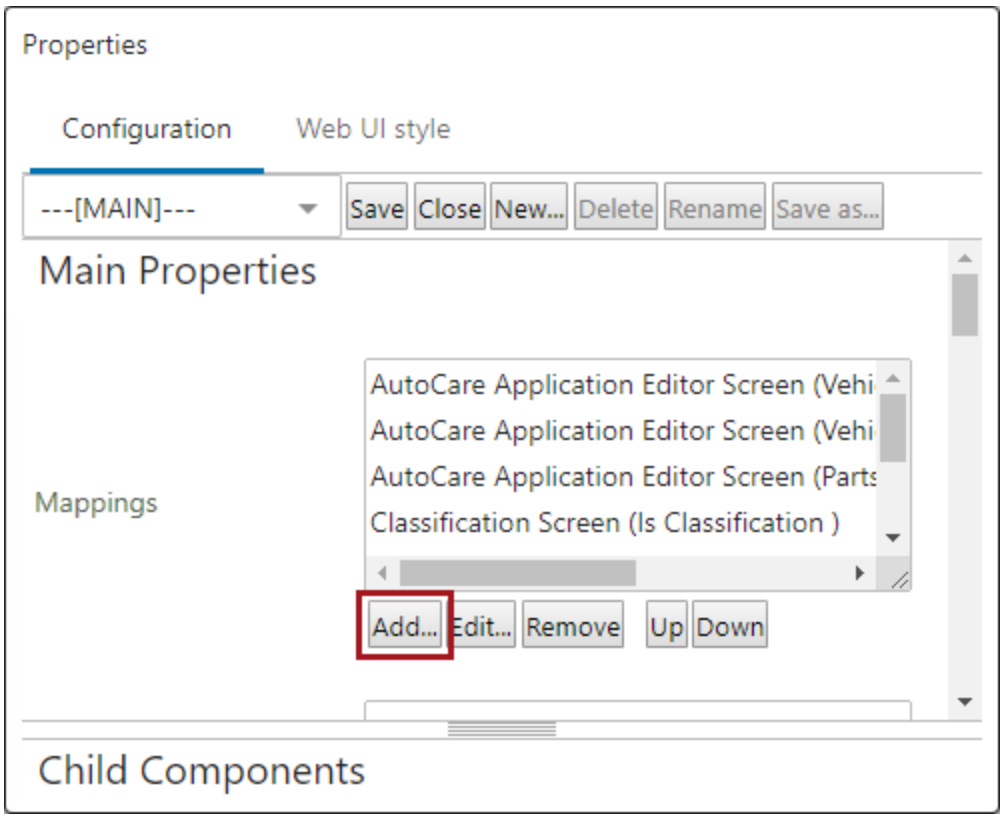



4. If required, type in a suitable title within the Title parameter that displays the 'Onboarding mappings navigator' component in the Global Navigation Panel. If left blank, the default value is set as 'Mapping Navigator'.

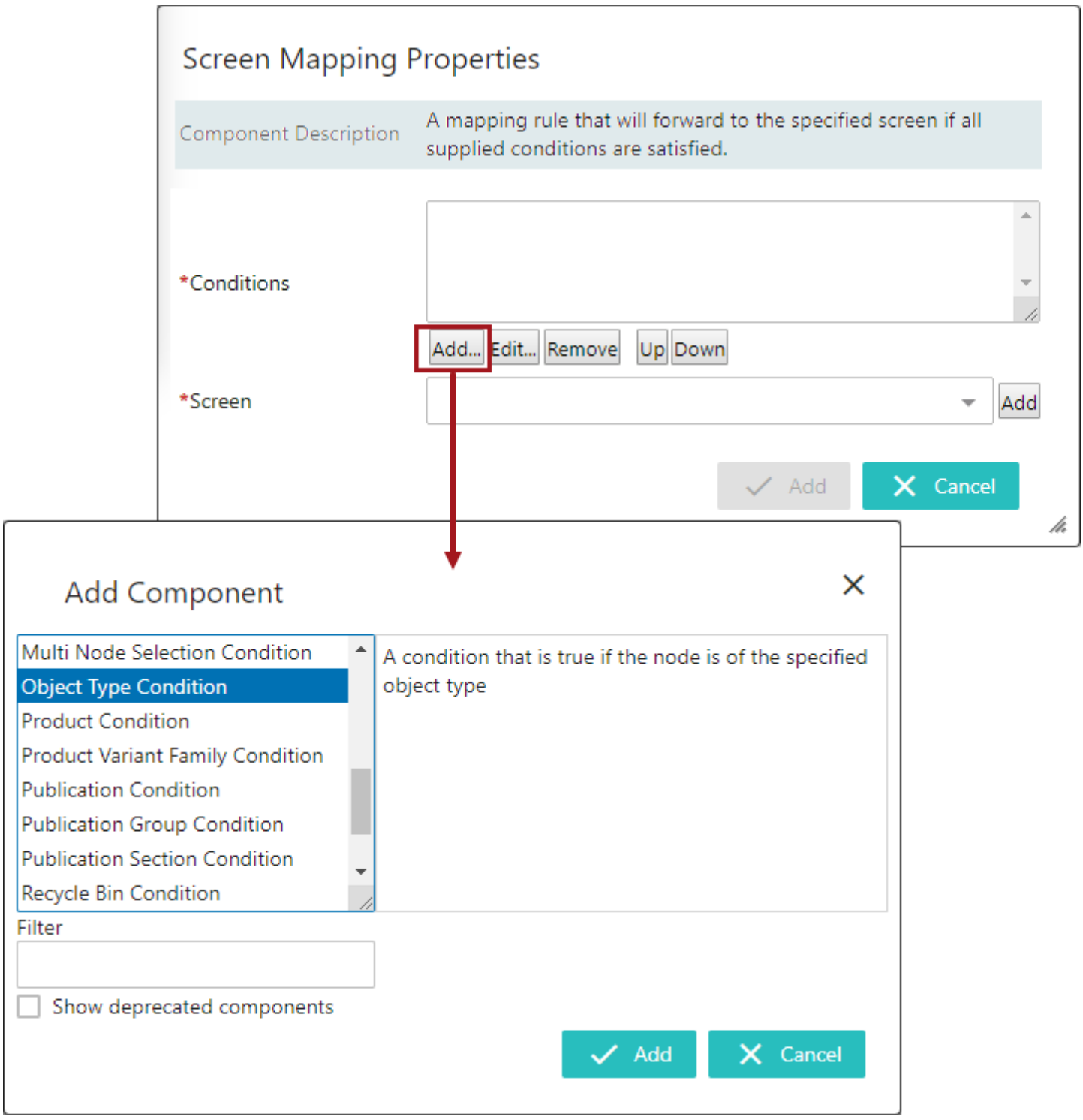
When finished, click **Save** in the designer to save the current settings. Do not close the designer unless you have finished configuring all the other parameters (outlined in the rest of this topic).

#### Configuring the Mapper Configurations to Display Onboarding Mappings Details Screen

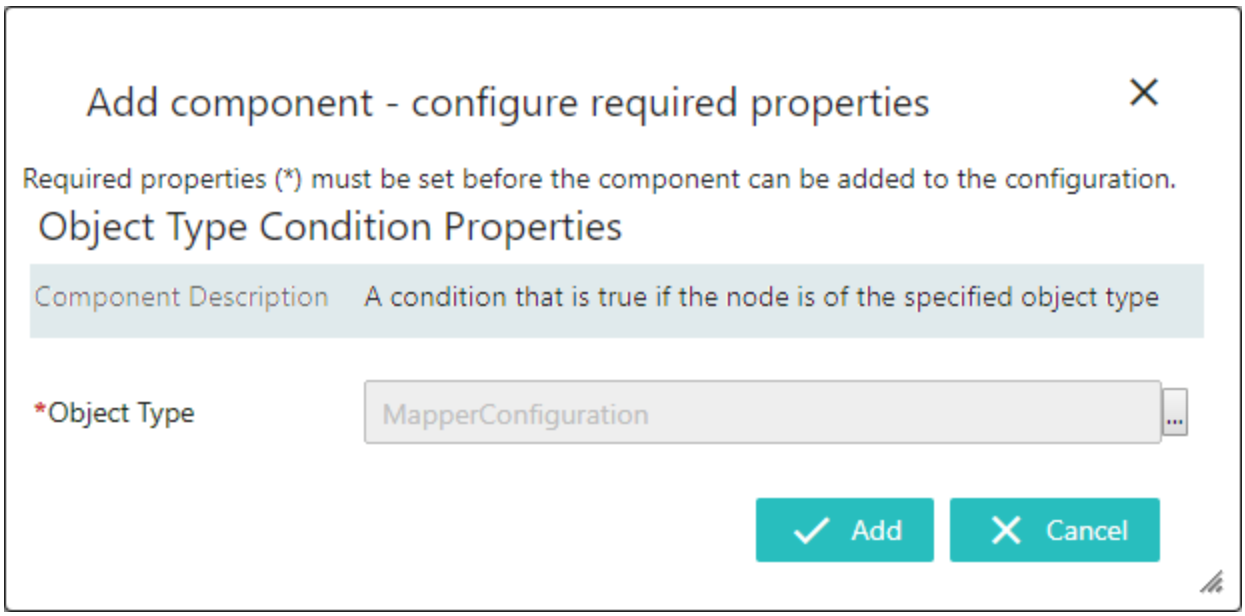
1. From the Main Properties, click the 'Add...' button for the Mappings parameter, and the Screen Mapping Properties will display.



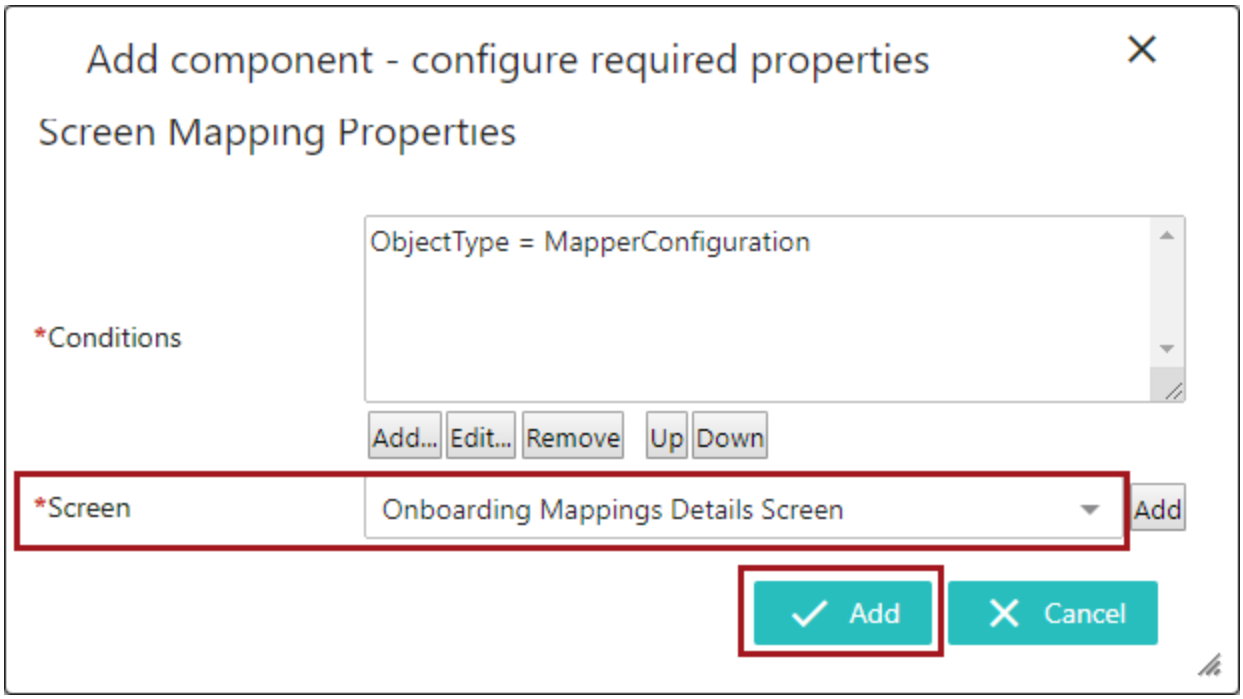

2. Add a condition by clicking the **Add...** button next to the 'Conditions' parameter, and the Add Component dialog will display.



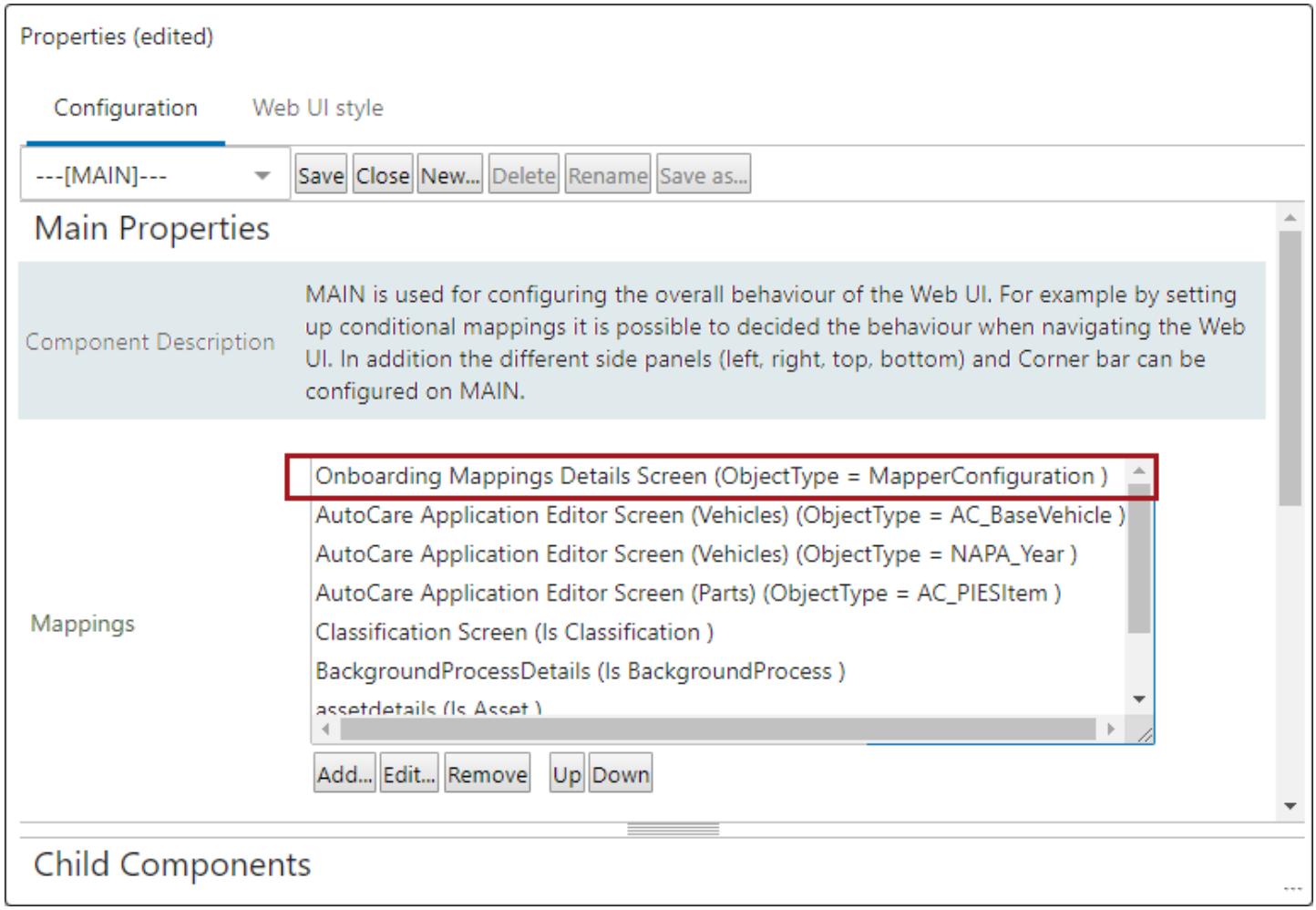

3. Within the Add Component dialog, select the Object Type Condition component and then click **Add**. In the Object Type Condition Properties, click the ellipsis button (...) next to the Object Type parameter, expand Basic Object Types, and select the Mapper Configuration (MapperConfiguration) object 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.




4. In the Screen Mapping Properties, select the previously created Onboarding Mappings Detail Screen using the dropdown selector next to the 'Screen' parameter.




5. 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 )
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.

Setup entity definitions can be exported as comments and submitted to an external source control system for comparison purposes. For details, refer to the Configuration Management documentation.

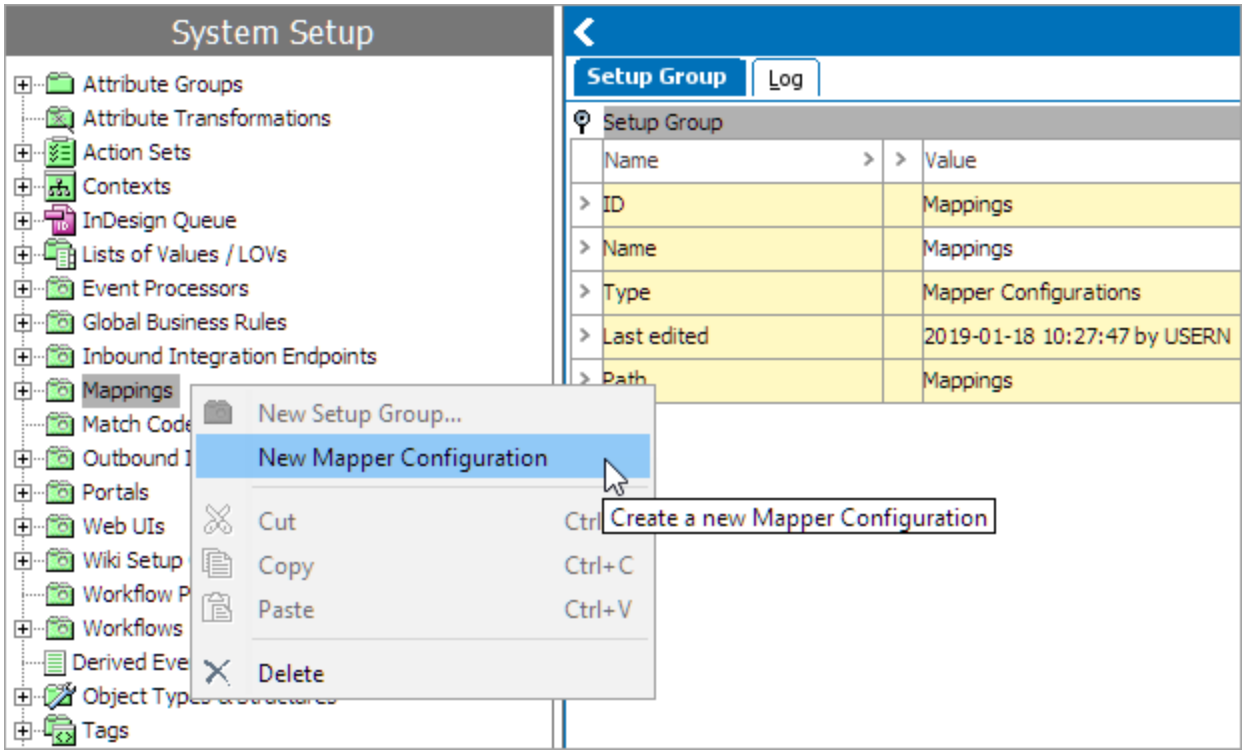
### Prerequisites

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, refer to the **Data Onboarding Automation Solution Initial Setup** topic within this section.

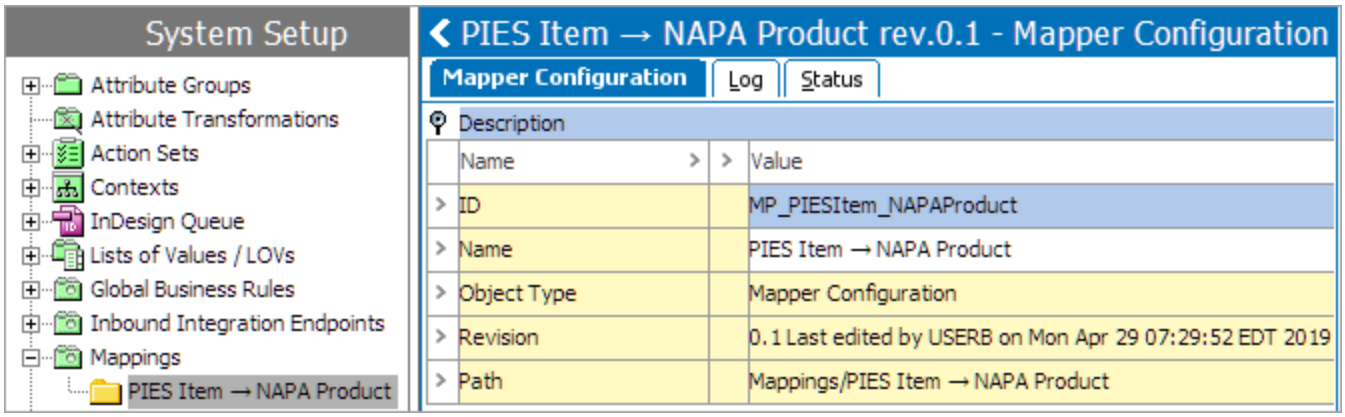
Since basic concepts for working with the designer are not covered in this section, a user configuring the Mapper Configuration setup entity within a Web UI is expected to be familiar with the Web UI Designer. In addition, the user must have appropriate privileges to access the designer. For more information, refer to the **Designer Access** topic within the **Web User Interfaces** section of the **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.

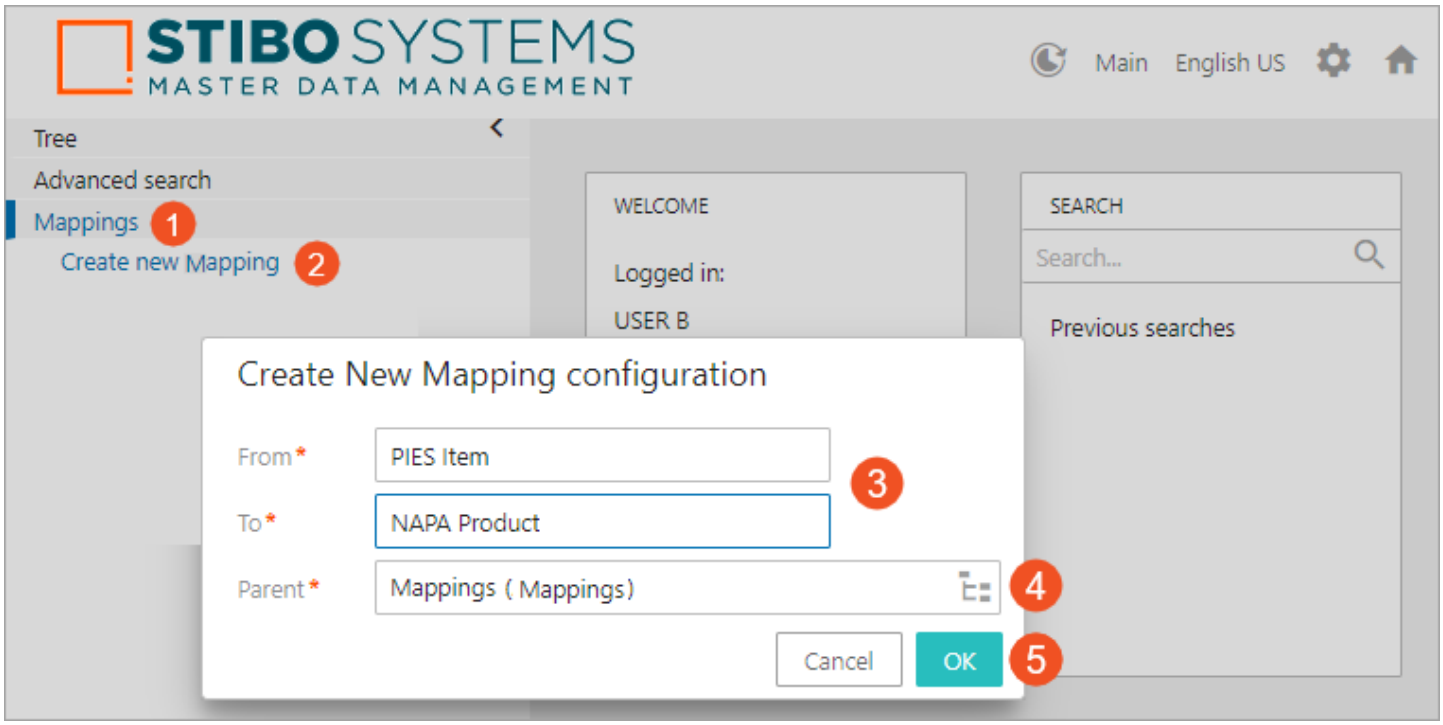


2. In the dialog that appears, type in a suitable ID and Name, and then click 'Create.' In the screenshot below, a Mapper Configuration named 'PIES Item → NAPA Product' is created under setup group called 'Mappings.'



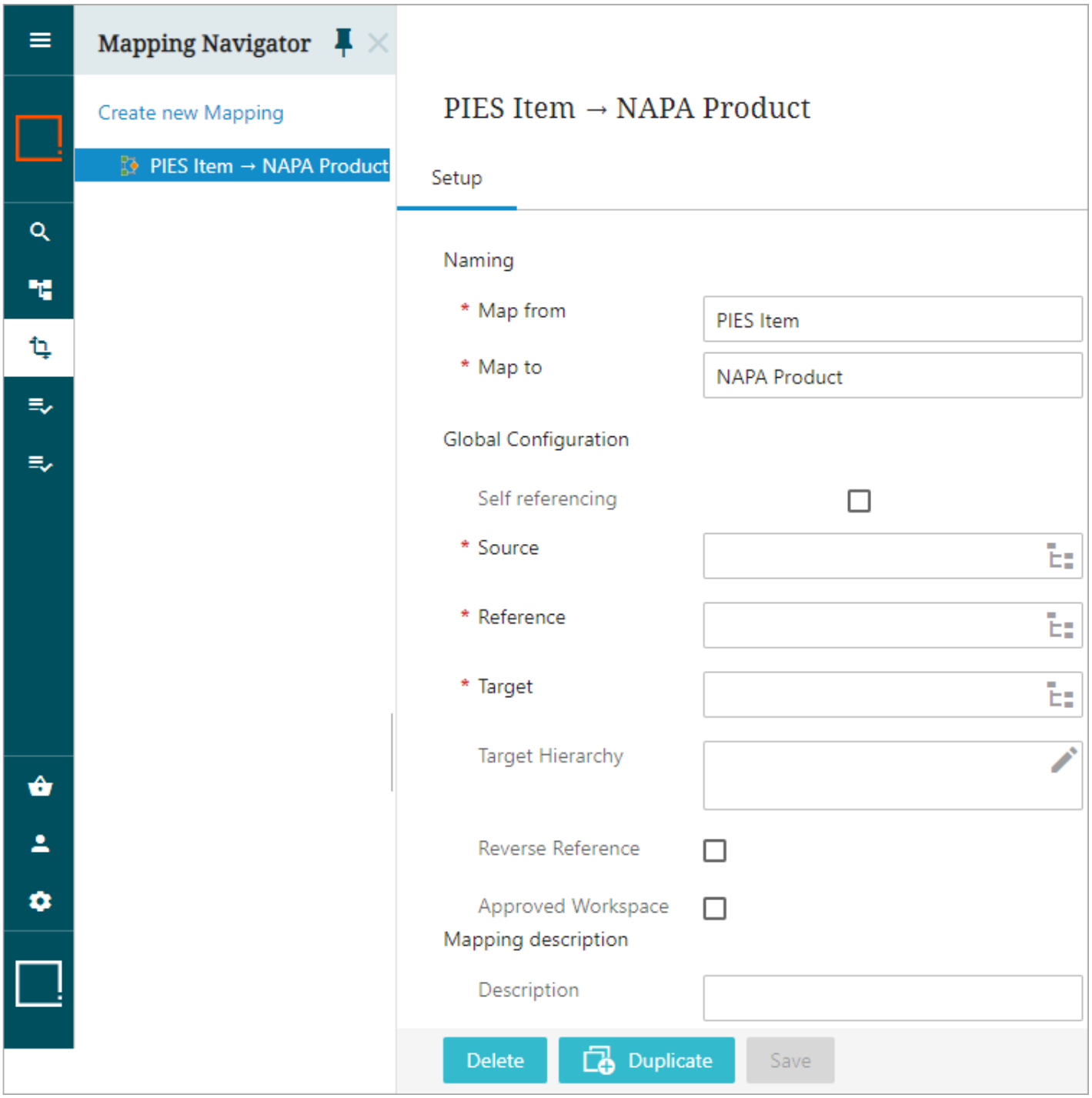

### Creating Mapper Configuration Setup Entity in Web UI

1. From the Web UI homepage, select the 'Mappings Navigator' component from the Global Navigation Panel.




2. Click on the 'Create new 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. Select the parent node (Setup Group) under which the newly created mapper configuration will reside.
5. Click **OK** and the new mapper configuration named 'PIES Item → NAPA Product' is created under setup group 'Mappings' in the Global Navigation 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.



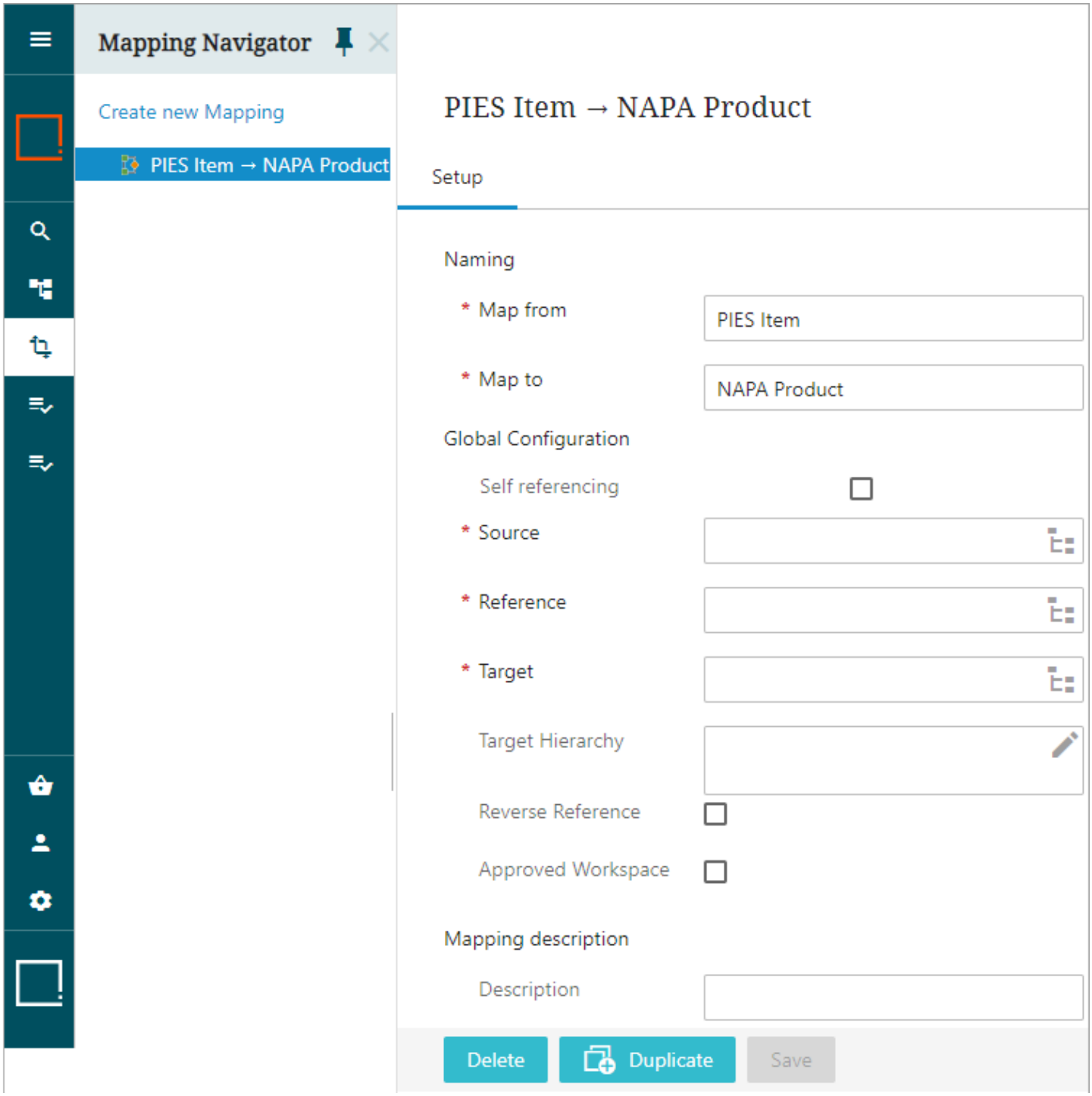

**Important:** If the Mapper Configuration is deleted in Web UI, then it will get moved to the Recycle Bin folder in System Setup. If the user tries to create a new mapping with the same text string in the 'from' and 'to' field, then the user will get the Unexpected error. If users want to reuse the 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).




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

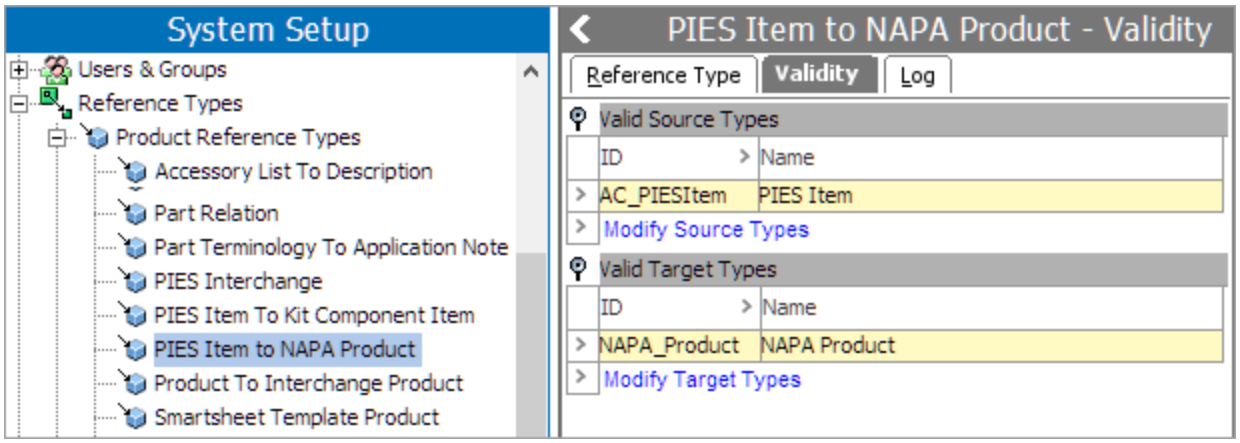
workbench and needs to be populated with an identifiable name of the source object type (this can be any string that the user wants to enter, e.g., PIES Item). Optionally, if a different name is desired, edit the name to be something more suitable.

**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.

**Self referencing:** By default, this remains unchecked. Selecting this checkbox allows users to onboard (move / copy) data within the object, thereby allowing users to execute onboarding that is defined as a self reference mapping. Selecting this checkbox makes the source and target to be the same object. Users only get to determine the source object within the Source parameter, and the Reference, Target, Target Hierarchy, and Reverse Reference parameters will be grayed out upon selecting this parameter.

**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).

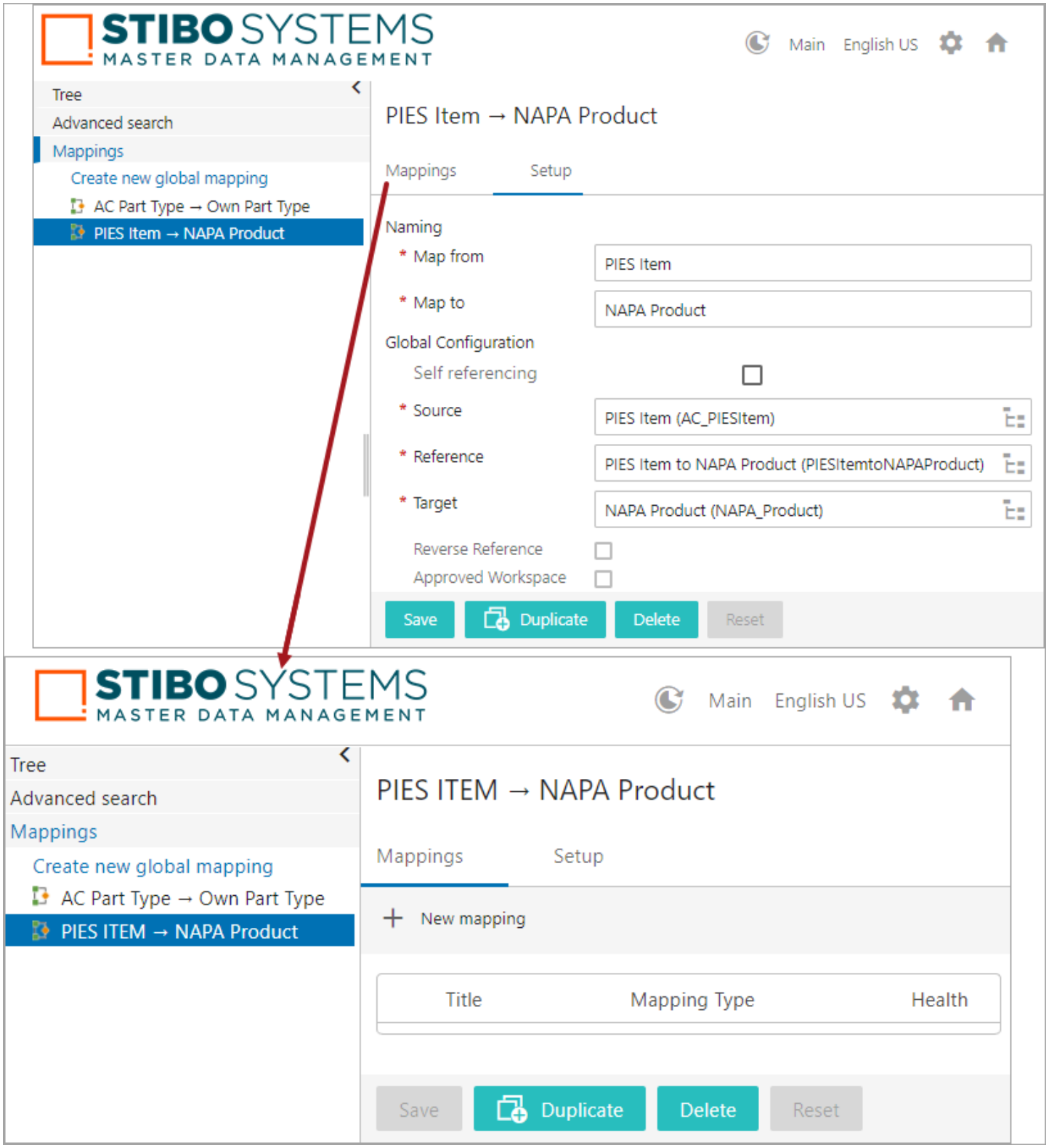
**Target Hierarchy:** By default, this optional parameter is blank. This parameter lets the user define an automatic creation of the Target object, parent to the Target object, and if required, a whole new hierarchy where the Target object should be created below. Click the Edit icon ( ) available within the right of the field

to open the Target Hierarchy Editor dialog. The way the name and ID structure of the newly created object is to be populated is defined within this dialog. The Target Hierarchy parameter checks for an existing object in the system, and a new object is created only if there are no similar object available in the system. The system will place the Target object in the suitable existing object if the object is already existing in the system. If not, it will create the parent object or a whole new hierarchy as defined in the Target Hierarchy Editor dialog. This ensures that there is no duplicate object created. For more information about Target Hierarchy Editor dialog, refer to the **Target Hierarchy Editor Dialog** topic within this section.

**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.

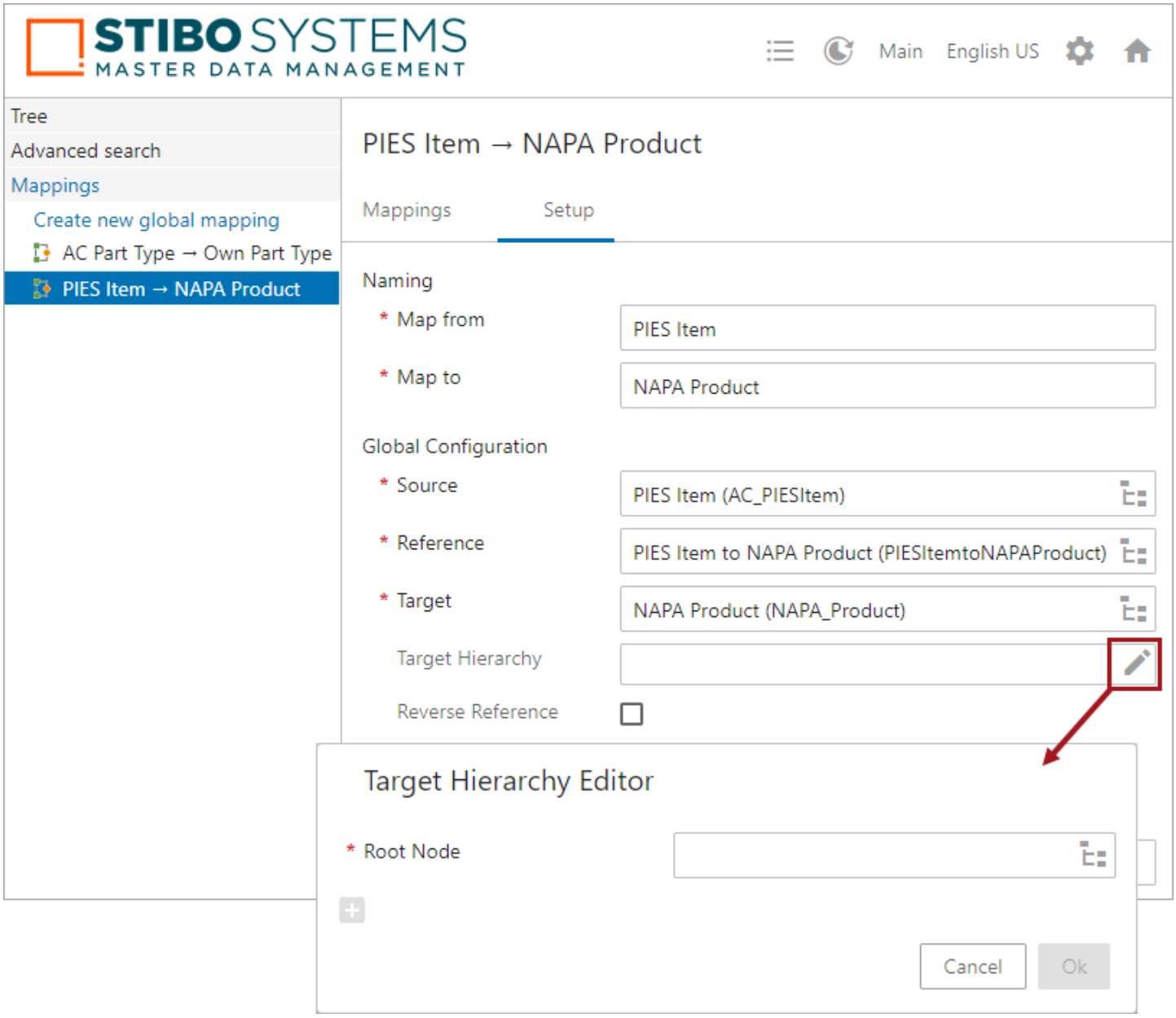

STIBO SYSTEMS
MASTER DATA MANAGEMENT
Main
English US
Tree
Advanced search
Mappings
Create new global mapping
AC Part Type → Own Part Type
PIES Item → NAPA Product
PIES Item → NAPA Product
Mappings
Setup
Naming
* Map from
PIES Item
* Map to
NAPA Product
Global Configuration
Self referencing
* Source
PIES Item (AC_PIESItem)
* Reference
PIES Item to NAPA Product (PIESItemtoNAPAProduct)
* Target
NAPA Product (NAPA_Product)
Reverse Reference
Approved Workspace
Save
Duplicate
Delete
Reset
STIBO SYSTEMS
MASTER DATA MANAGEMENT
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
New mapping
Title
Mapping Type
Health
Save
Duplicate
Delete
Reset

### Target Hierarchy Editor Dialog

When mapping data, not every Source object in the system will have an existing Target object to run the Mapper Configuration on. If the Target object does not exist, the Mapper should be able to create a Target object and place the Target object in the right hierarchy (referred to as Target object hierarchy). To automate the process of creating the Target object (and its hierarchy), the user has to configure on how the Target object hierarchy is to be automatically created when the Mapper Configuration setup entity is executed.

When the term 'Target object hierarchy' is mentioned in this topic, it is to be noted that the hierarchy could be in its simplest form that consists of just the Target object itself, or it can be a complex hierarchy that includes multi-level folders narrowing down to house the Target object.

When clicking the Edit icon ( ) that is displayed to the right of the **Target Hierarchy** parameter (available on the Setup tab of the Onboarding Mappings Details Screen), the Target Hierarchy Editor dialog appears. This dialog allows the user to define an automatic creation of the Target object, parent to the Target object, and if required, a whole new hierarchy where the Target object should be created below. For more information on configuring the Setup tab of the Onboarding Mappings Details Screen, refer to the **Configuring Mapper Configuration Setup Entity** topic of the **Data Onboarding** documentation.




Configuring the Target Hierarchy parameter is only necessary if the parent node of the Target object does not exist.

For example, if application records are to be created in the Target Hierarchy and the part that the applications belong to already exists in the Target Hierarchy, and the relationship between the source part and target part have already been established, then nothing needs to be defined in the Target Hierarchy parameter.

But if the part (parent node) for the applications do not exist, then users can use the Target Hierarchy parameter to configure the hierarchy that the Target object should be created below by defining the root node and its parent object types. Once the root node and parent object types have been defined, then the **OK** button will be enabled and the parent objects will get created along with the applications when the mapping configuration is executed.

The Target Hierarchy parameter has been designed to check for an existing Target object hierarchy in the system, and a new hierarchy is created only if there are no similar hierarchy available in the system. The system would place the Target object in the suitable hierarchy if the similar hierarchy is already existing in the system. This ensures that there is no duplicate hierarchy created. The way the name and ID structure of the newly created object is to be populated can be well defined in the Target Hierarchy Editor dialog.

**Note:** The Target Hierarchy parameter is designed to check for existing Target objects within the system to ensure that duplicate objects are not created. If existing objects are encountered, then they will be skipped and only new objects will get created.

### Prerequisites

Before configuring the Target Hierarchy Editor dialog, the Source, Reference, and the Target parameters that are available within the Setup tab of the Onboarding Mappings Details screen have to be correctly populated.

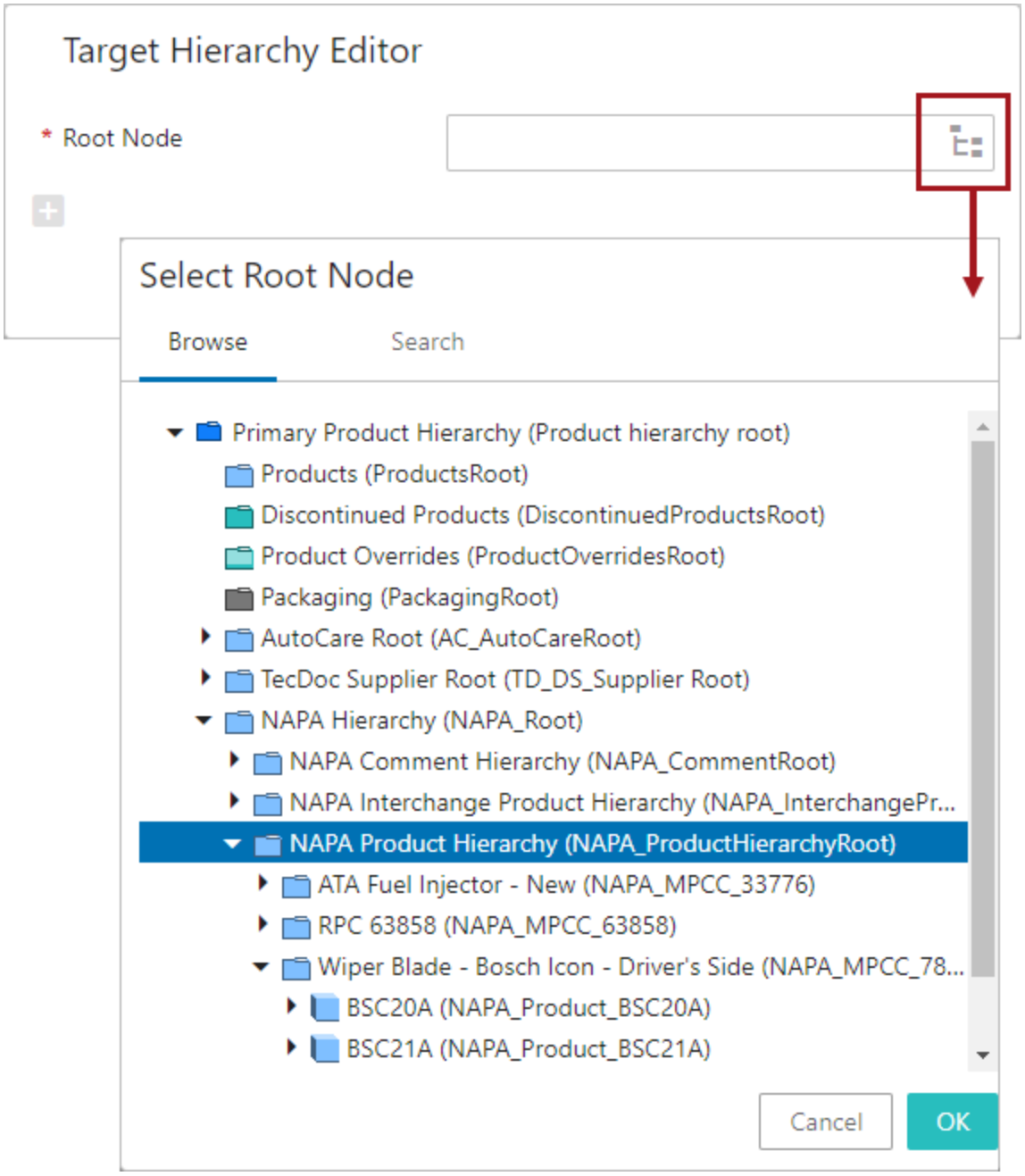
#### Configuring the Target Hierarchy Editor Dialog

To configure the Target Hierarchy Editor dialog, follow these steps:

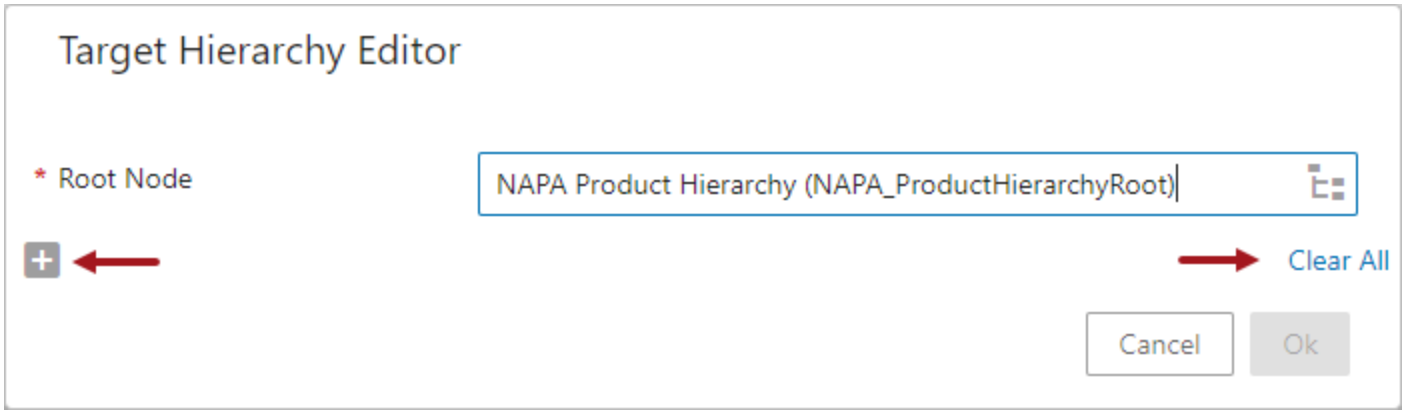
1. Click on the Edit icon ( ) on the Target Hierarchy parameter to open the Target Hierarchy Editor Dialog (as shown in the picture above).
2. In the Target Hierarchy Editor dialog, click on the Node Picker icon available next to the Root Node parameter to select the root node object. This selected object will be considered as the first level of the Target object hierarchy (root node) where the Target object should be created somewhere down below.

**Note:** Only the parent node of the node holding the object type defined in the 'Target' parameter will be valid to be selected.

In this example below, the NAPA Product Hierarchy is selected to be the root node where the new Target object should be created below.

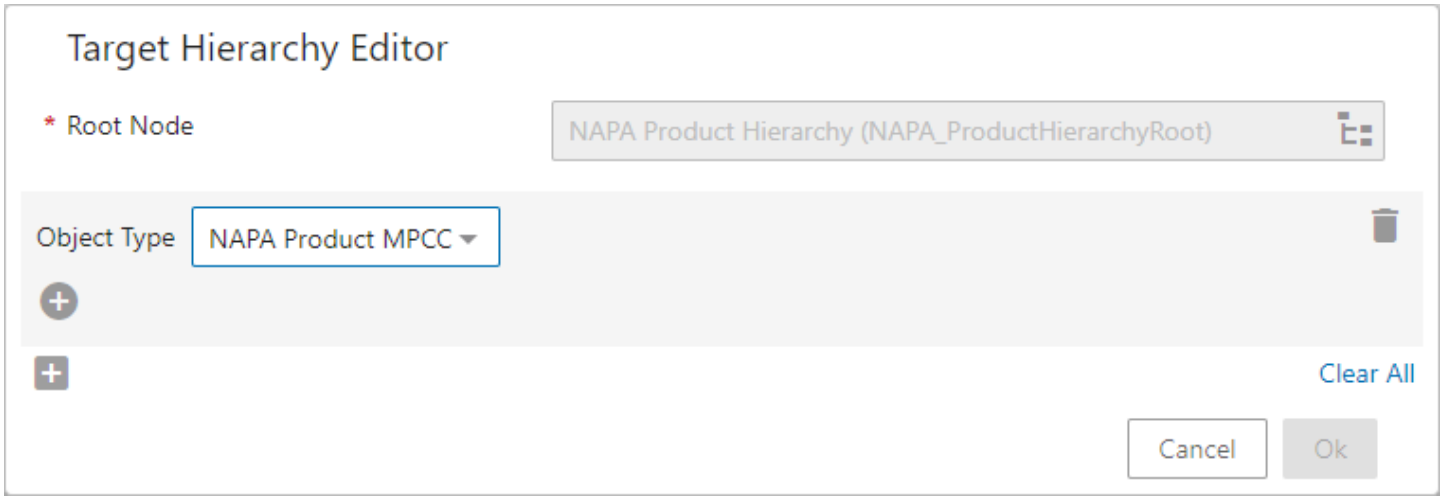



Once selected, the root node will be displayed within the Root Node parameter, and the 'Add Object Type' icon ( + ) along with the 'Clear All' link gets activated within the dialog. The 'Clear All' link will clear all of the values that are populated within the Target Hierarchy Editor dialog.

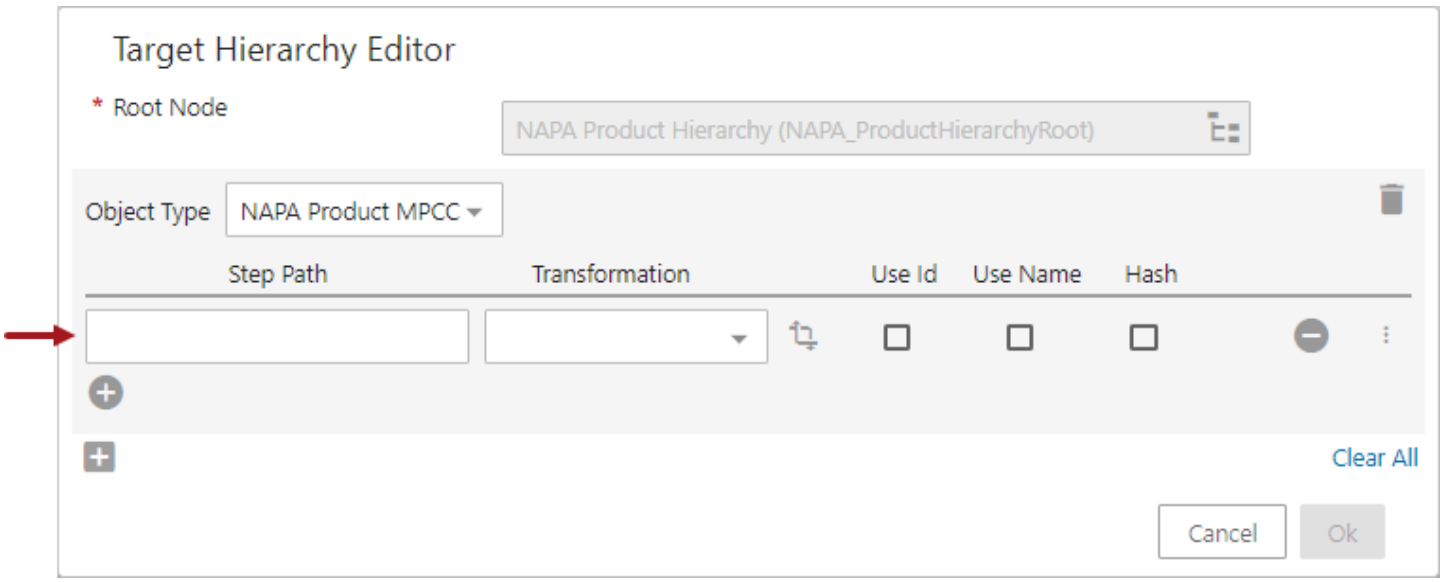



3. Click the 'Add Object Type' icon ( ) to further narrow down the process of defining the Target object hierarchy. This provides an option to select the object type using the dropdown next to the Object Type field. Only the object types that are valid under the root node object type (defined above) are available as options within the dropdown.

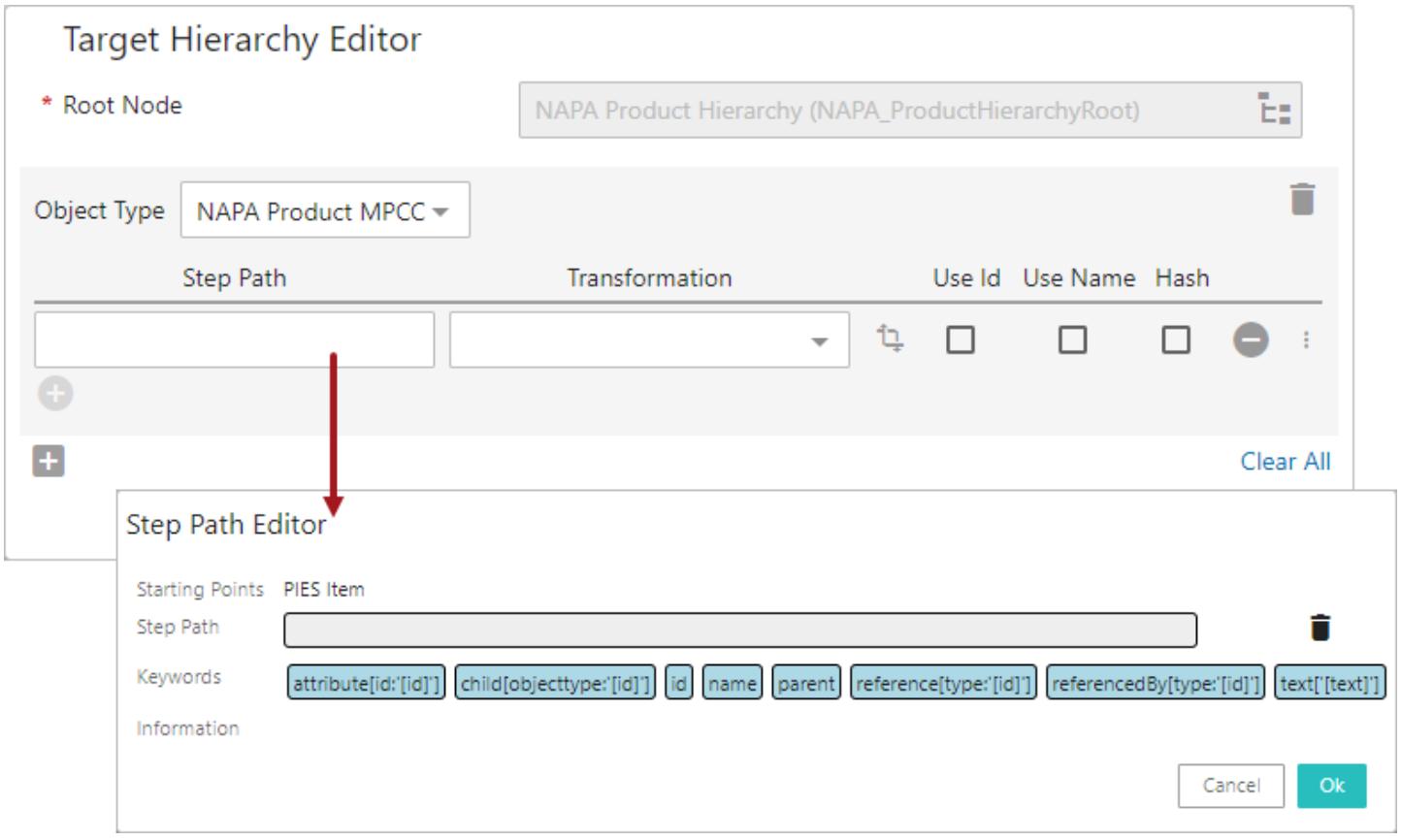
   In the screenshot below, the NAPA Product MPCC object type is selected.




4. By default, the ID of the new objects that are created using the above mentioned object type will have the system generated STEP ID. However, to create an object with a user specific ID and Name, click the 'Add' icon ( ). This provides an option for the user to define a criteria for the ID and Name of the new object that gets created on this level of the hierarchy.




5. Click on the STEP Path field to open the Step Path Editor dialog that the user can use to define which information (ID, Name, attribute values, or references) from the Source object needs to be retrieved. 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, refer to the **Mapping Validation Path Functionality** topic within this guide.



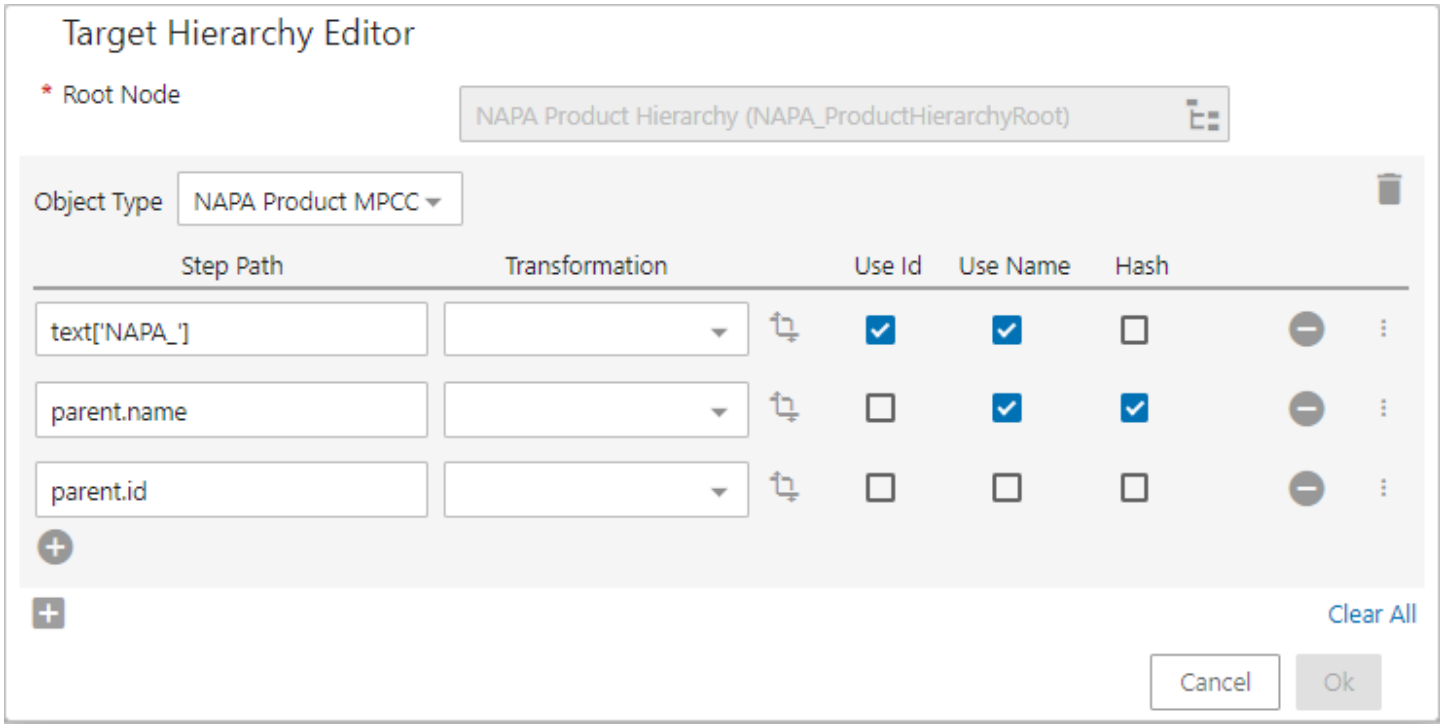
The retrieved information can be used to populate the ID or Name field of the newly created object by selecting the 'Use Id' or 'Use Name' checkbox respectively. Select the 'Hash' checkbox if the STEP Path value should be included in the hash value of the object's ID to calculate the object's uniqueness. The ID is allowed to have a maximum of 40 characters with the free text and hash value combined.

If necessary, use the dropdown in the Transformation column to review the list of available attribute transformations in the system that can be applied while defining the ID and Name of the new object that gets created on this level of the hierarchy. Also, users can click on the 'Select and modify transformation' icon ( ) to create a new or to edit an existing attribute transformation. For more information on creating or editing an attribute transformation via the Target Hierarchy Editor window, refer to the **Attribute Transformation in Mapper Configuration Setup Entity** topic within this guide.

If required, click the three-dotted icon ( ) that is available in the right of the criteria to display the rearranging dialog. The user can select the Top, Up, Down, or Bottom button to rearrange the criteria accordingly.

For this example, it is considered that the Name and ID values from the parent (PIES PCdb Part Terminology) of the Source object will be used in the Name and ID field of the new object (NAPA Product MPCC). Also, it is considered that the new objects will have a prefix value 'NAPA_'. To achieve this considerations the following setup is used.

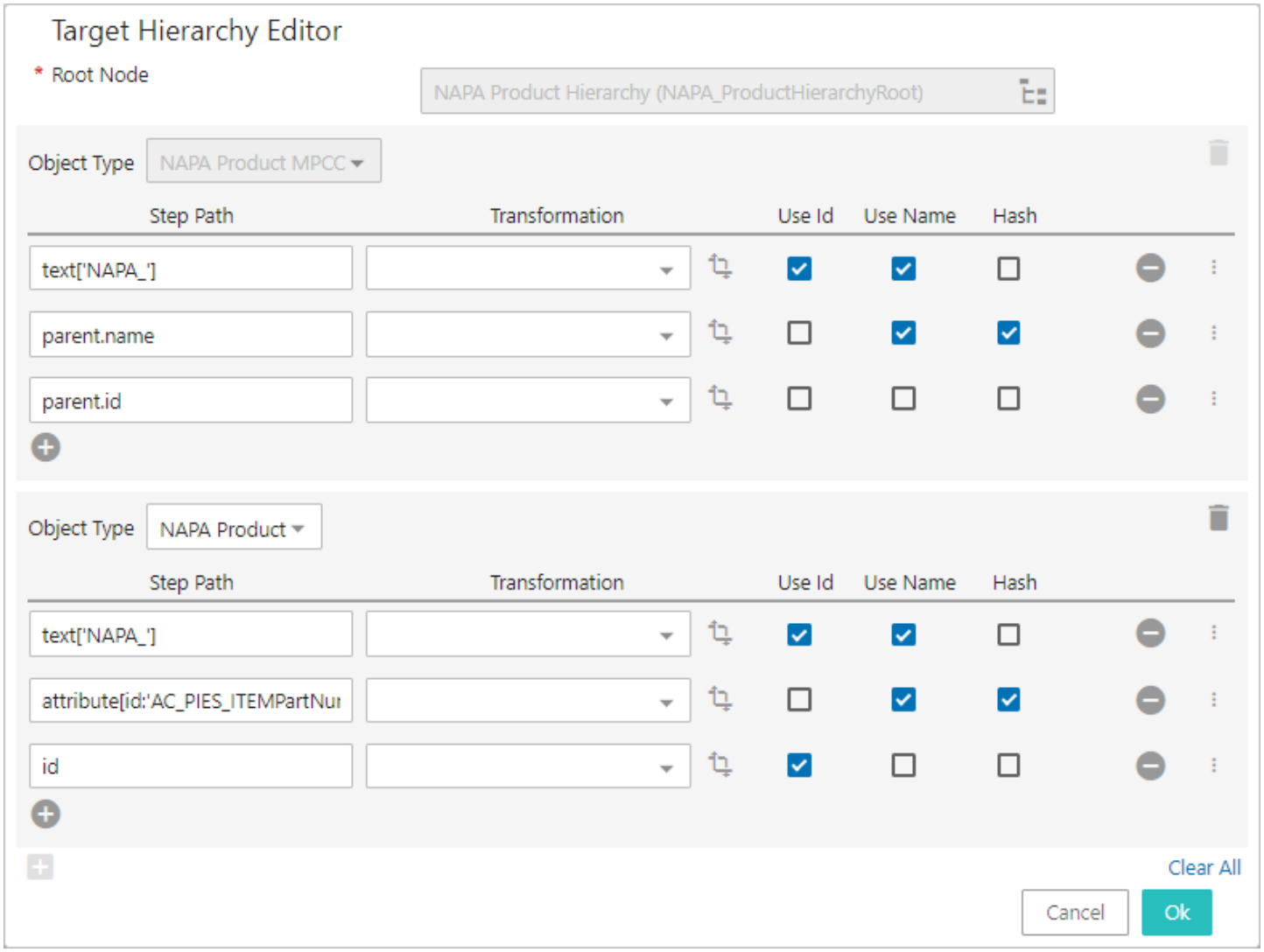
- The **text['[text]']** element is used to add 'NAPA_' as the prefix.
- The **parent** and the **name** elements are used to follow parent of the Source object and retrieve Name value from the parent of the Source object.
- Similarly, The **parent** and the **id** elements are used to follow parent of the Source object and retrieve ID value from the parent of the Source object.
- The 'Use ID,' 'Use Name,' and 'Hash' checkboxes are selected accordingly to use in the Name and the ID of the new object being created, as shown below.



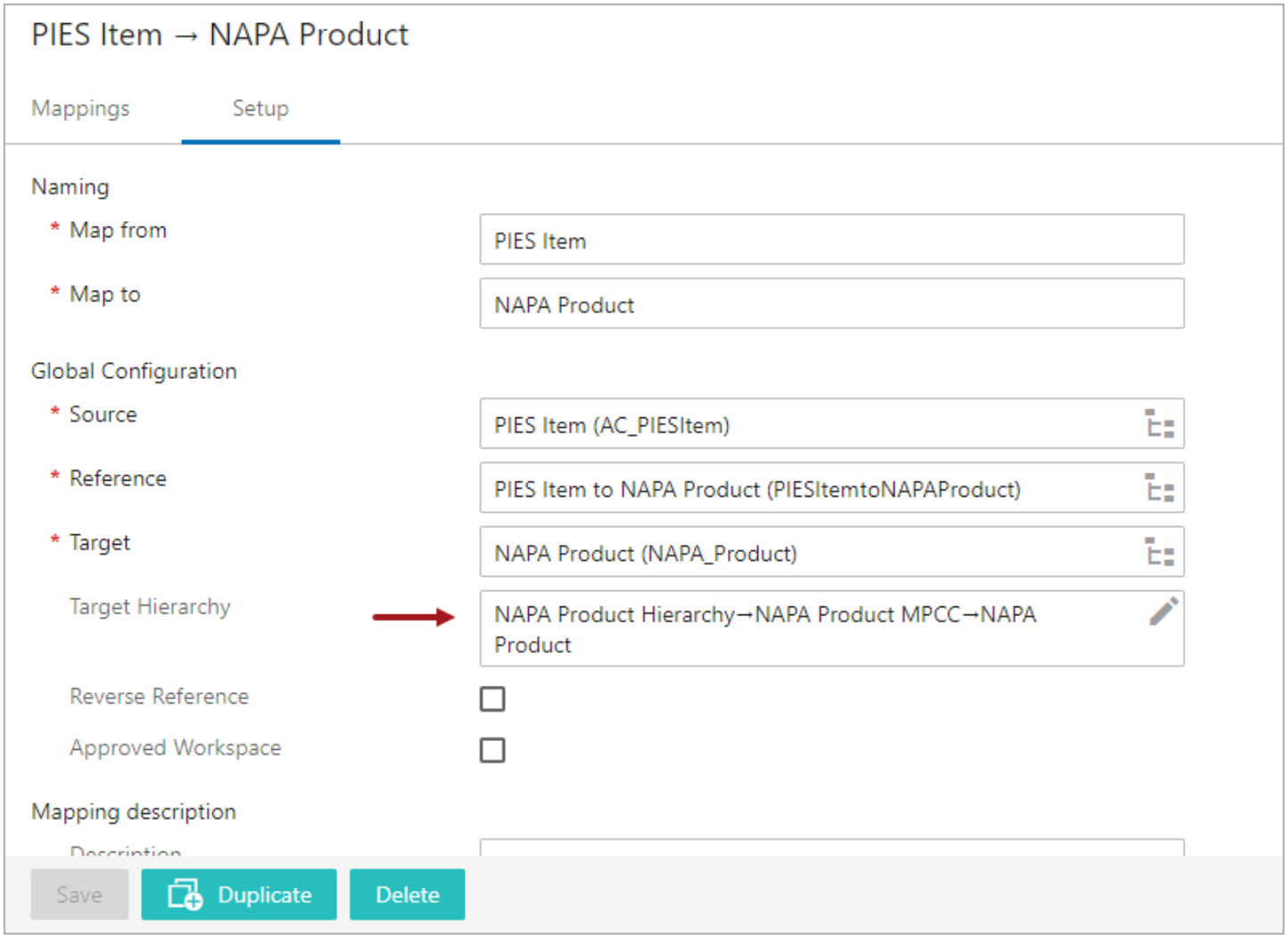

6. Similarly, add and define all of the objects that fall in the Target object hierarchy until the last object i.e., the Target object itself is defined. The **OK** button will be enabled when the Target object is defined.

   For this example, it is considered that the values of the Part Number attribute and ID from the Source object (PIES Item) will be used in the Name and ID field of the Target object (NAPA Product). Also, it is considered that the newly created Target objects will have a prefix value 'NAPA_'. To achieve this considerations the following setup is used.

   - The **text['[text]']** element is used to add 'NAPA_' as the prefix.
   - The **attribute[id:'[id]']** element is used to follow the Source object and retrieve the value of the attribute Part Number from the Source object.
   - Similarly, The **id** element is used to follow the Source object and retrieve ID value from the Source object.
   - The 'Use ID,' 'Use Name,' and 'Hash' checkboxes are selected accordingly to use in the Name and the ID of the Target object being created, as shown below.




7. Click **OK** to close the Target Hierarchy Editor dialog. All the objects in the Target object hierarchy with a direction arrow pointing from Target root node to Target object will be displayed in the Target Hierarchy parameter.


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)
Target Hierarchy
NAPA Product Hierarchy→NAPA Product MPCC→NAPA Product
Reverse Reference
Approved Workspace
Mapping description
Save
Duplicate
Delete

## 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:

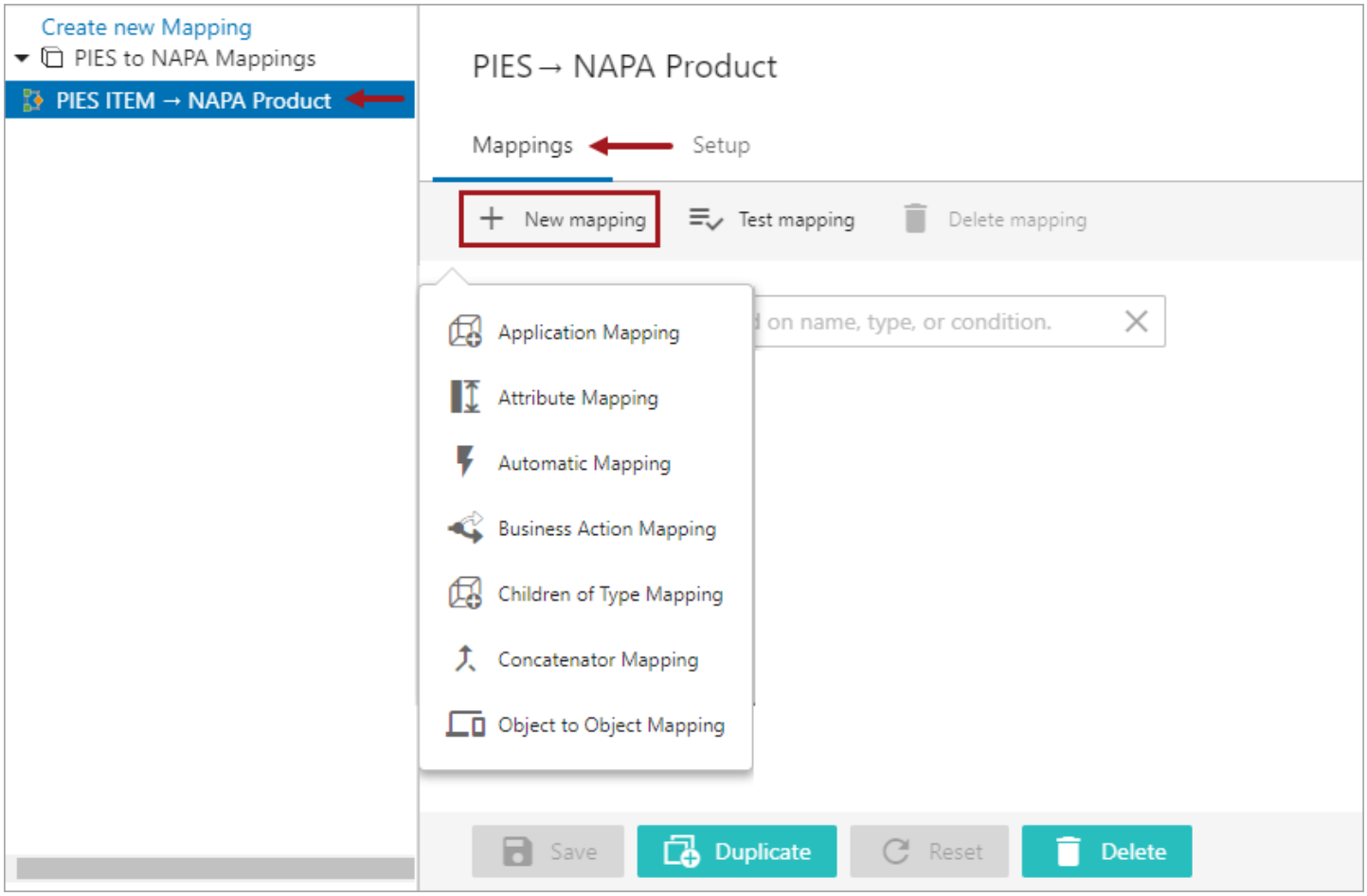
- Can view existing mapping plugin and its configuration
- Can create new mapping plugins
- Can define global configurations for the selected Mapper Configuration setup entity

Mapper Configuration setup entity can be configured to meet different Onboarding and Offboarding requirements in Onboarding Mappings Details screen.

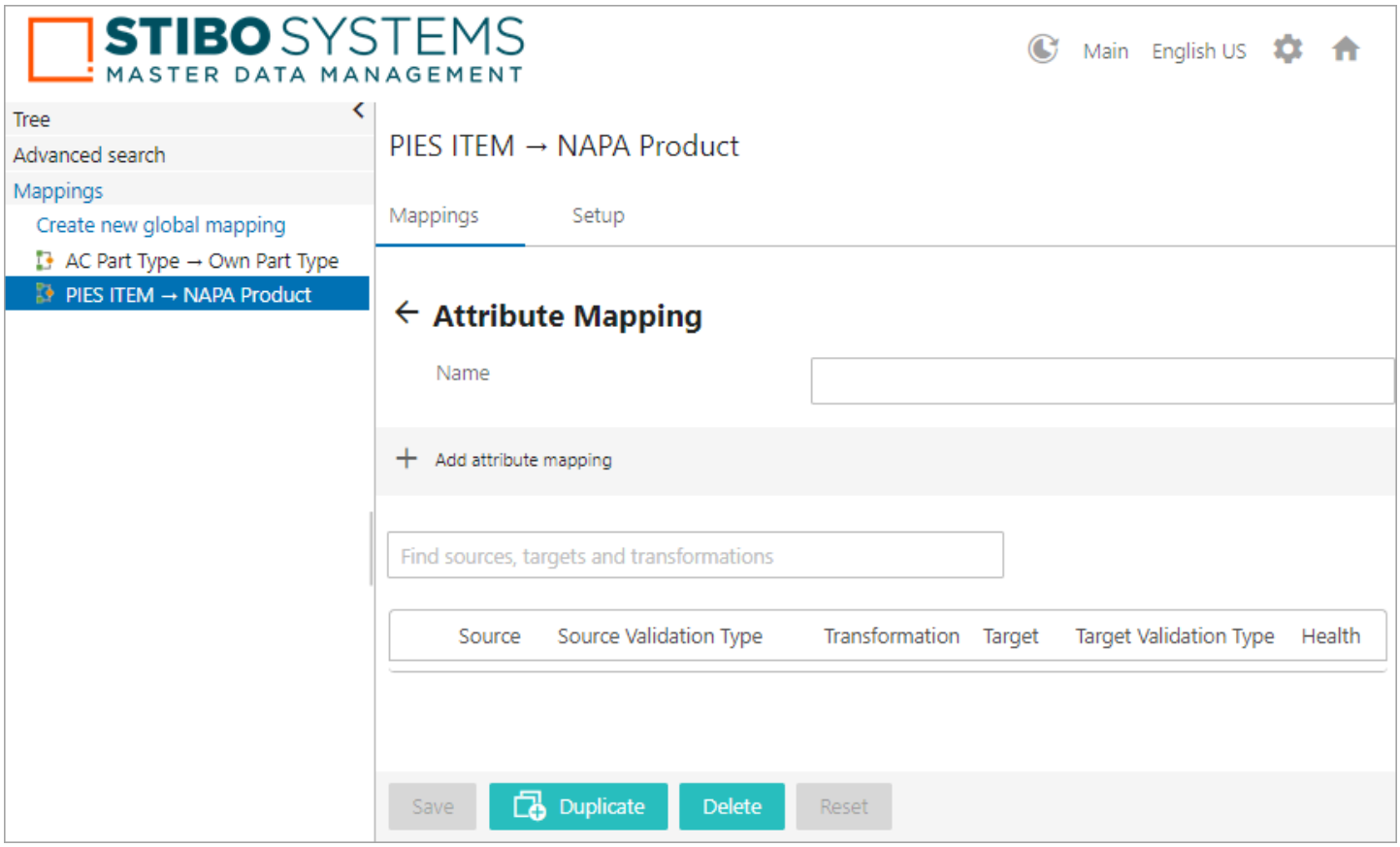
### Creating Mapping Plugins

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, refer to 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.

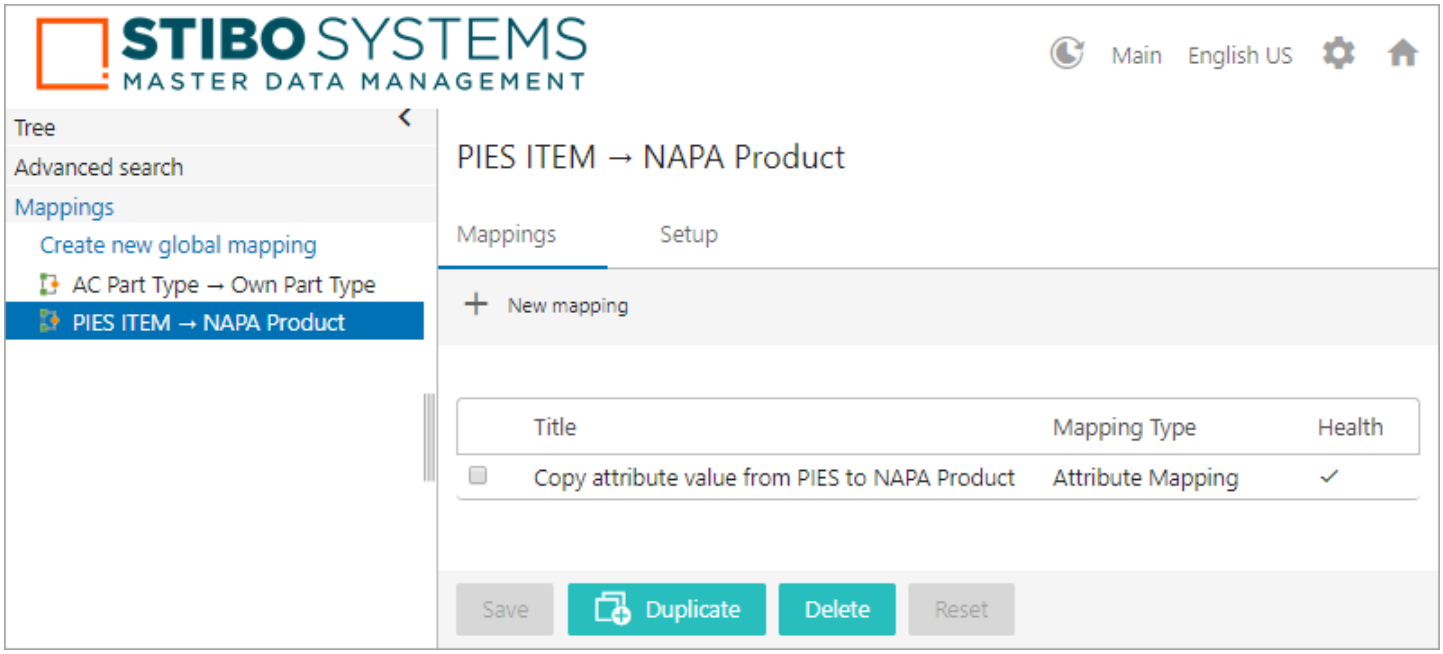


2. Select the desired type of mapping plugin, and a screen is displayed prompting the user to further configure the plugin. For example, in the screenshot below, a screen is displayed when the Attribute Mapping plugin is selected.




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.

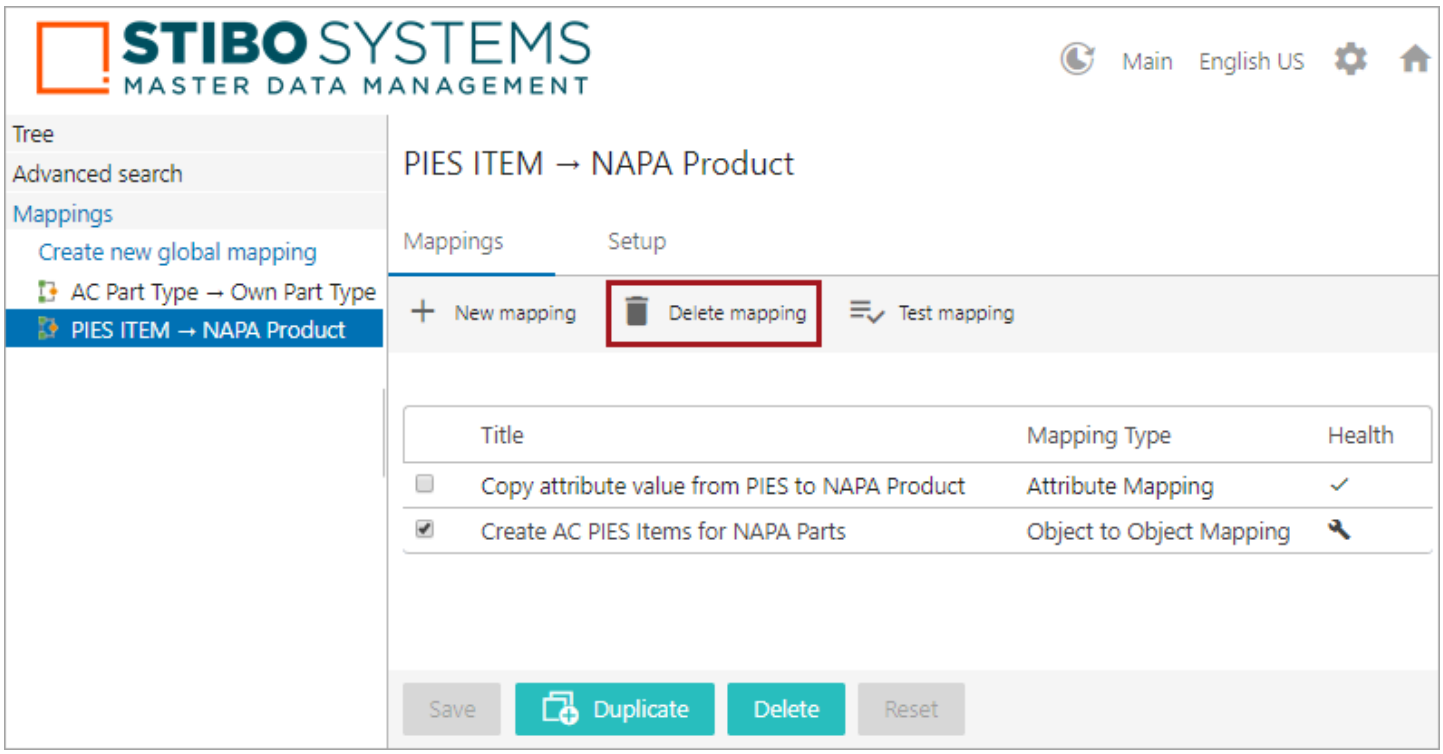



## Managing Mapping Plugins

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).




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 executing through Mapper Action button (this is the recommended method in the Web UI as it will be faster)
- By executing through Stacked Mapper Action button (this is being used when multiple Mapper Configuration Setup Entity has to be accessed standing on the object)
- By executing through the Onboarding Comparison Screen
- By a background process with an automated import
- By configuring it in a business rule and executing the business rule on the source object

Setup entity definitions can be exported as comments and submitted to an external source control system for comparison purposes. For details, refer to the Configuration Management documentation.

## 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, refer to 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, refer to the **Data Onboarding Automation Solution Initial Setup**,and the **Configuring Web UI for Data Onboarding Automation solution** topics within this section.

## Executing Through Mapper Action Button

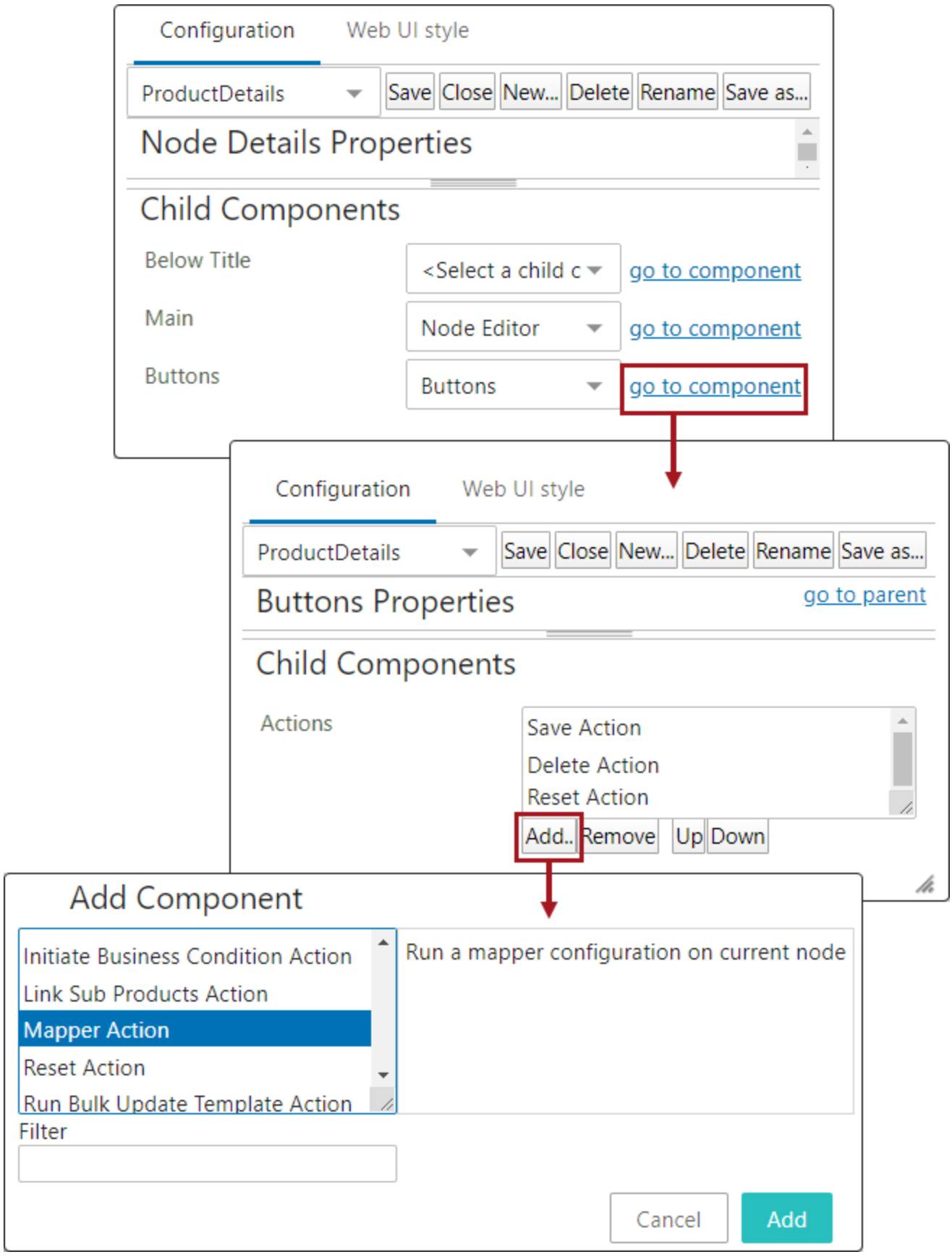
This process involves defining an action button called **Mapper Action** to hold the Mapper Configuration setup entity, and configuring the action button in required screens. Clicking on the action button will initiate the Mapper Configuration setup entity on the selected node.

The **Mapper Action** button can be configured to be available within the Source object (or Target object) screen, or any other places that allows the action button to be executed (e.g., run as a bulk update on a Collection, or initiate the action button with a node selected in the Search by Card screen). For the ease of understanding, this document depicts an example of configuring the Mapper Action button in the Node Detail screen that is configured to view the Source object.

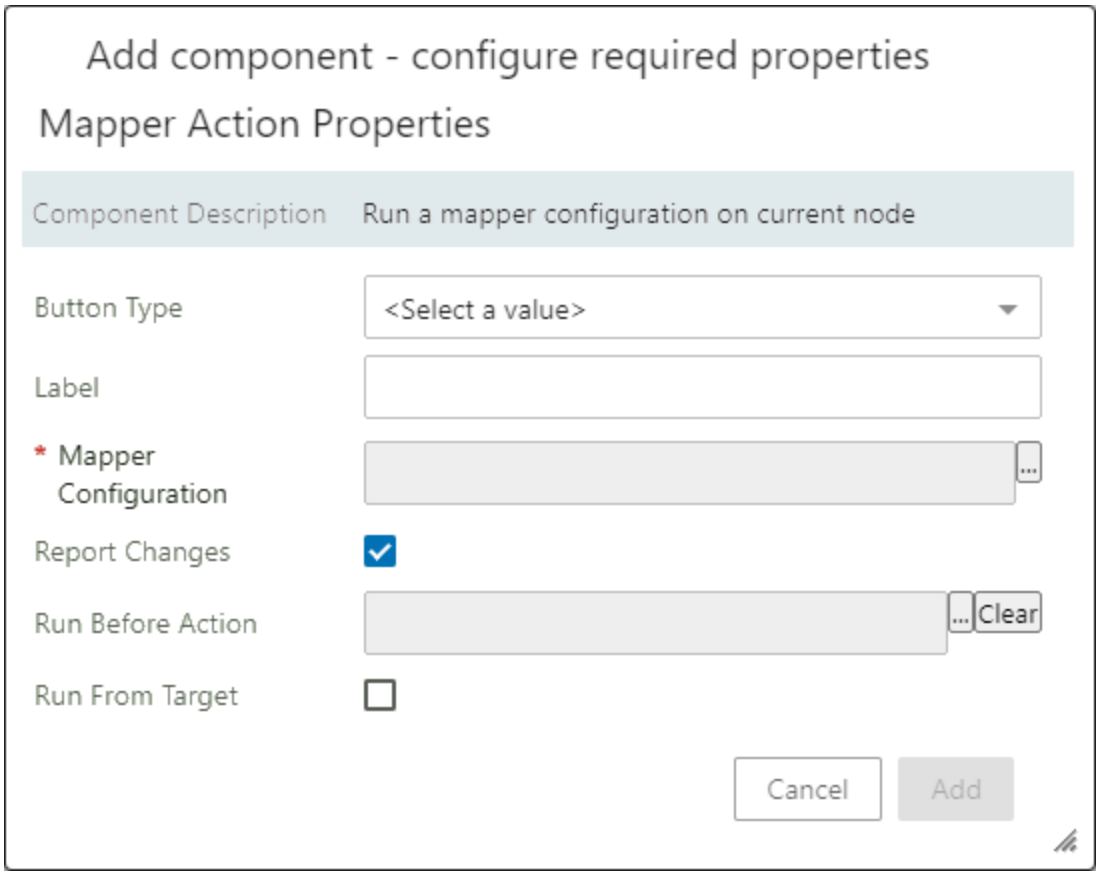
**Note:** If the Mapper Action button is to be configured in the Target object screen, then it is required that the user selects the Run From Target parameter as explained in this topic below.

Following are the steps to configure the Mapper Action button.

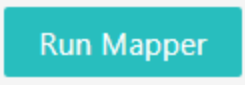
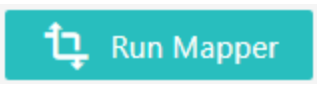
1. In the Web UI designer, navigate to the screen that requires the Mapper Action button to be configured > Buttons > go to component > Button Properties.
2. Click **Add** next to the Actions field that is available within the Buttons Properties to access the Add Component dialog.


Configuration
Web UI style
ProductDetails
Save
Close
New...
Delete
Rename
Save as...
Node Details Properties
Child Components
Below Title
<Select a child c
go to component
Main
Node Editor
go to component
Buttons
Buttons
go to component
Configuration
Web UI style
ProductDetails
Save
Close
New...
Delete
Rename
Save as...
Buttons Properties
go to parent
Child Components
Actions
Save Action
Delete Action
Reset Action
Add..
Remove
Up
Down
Add Component
Initiate Business Condition Action
Link Sub Products Action
Mapper Action
Reset Action
Run Bulk Update Template Action
Run a mapper configuration on current node
Filter
Cancel
Add

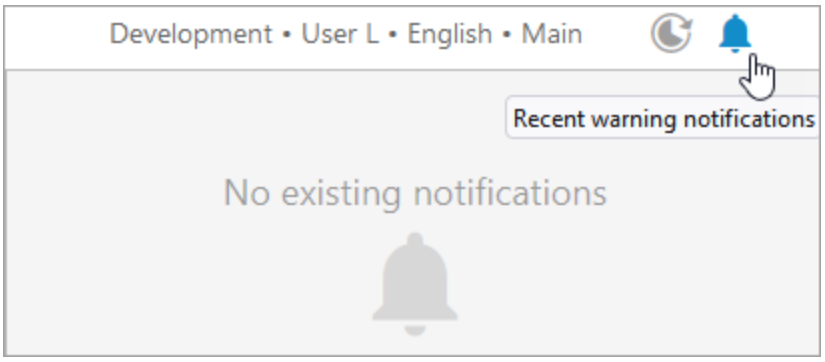
3. Find **Mapper Action** and click **Add**, and the Mapper Action Properties dialog will be displayed.




4. Populate the parameters as detailed below:
   - **Button Type:** Select from ICON_AND_TEXT, ICON, and TEXT using the Button Type dropdown. If the user choose to select TEXT or ICON_AND_TEXT and the Label parameter is empty, the default text 'Run Mapper' will display.

| Button Types | Button Displayed As |
|---|---|
| ICON | |
| TEXT |  |
| ICON_AND_TEXT |  |

- **Label:** This parameter defines the text that the user wants to appear in the button that is displayed on screen.
- **Mapper Configuration:** Required parameter to specify the Mapper Configuration setup entity that needs to be executed. Click the ellipsis button (...) to find and select the Mapper Configuration setup entity that is available in the system.
- **Report Changes:** This parameter is checked by default. Having this parameter checked ensures that the changes effected by the execution of the associated Mapper Configuration setup entity is displayed in the message dialog (explained below) and also in the 'Recent warning notifications' panel, located in the upper right corner of Web UI.

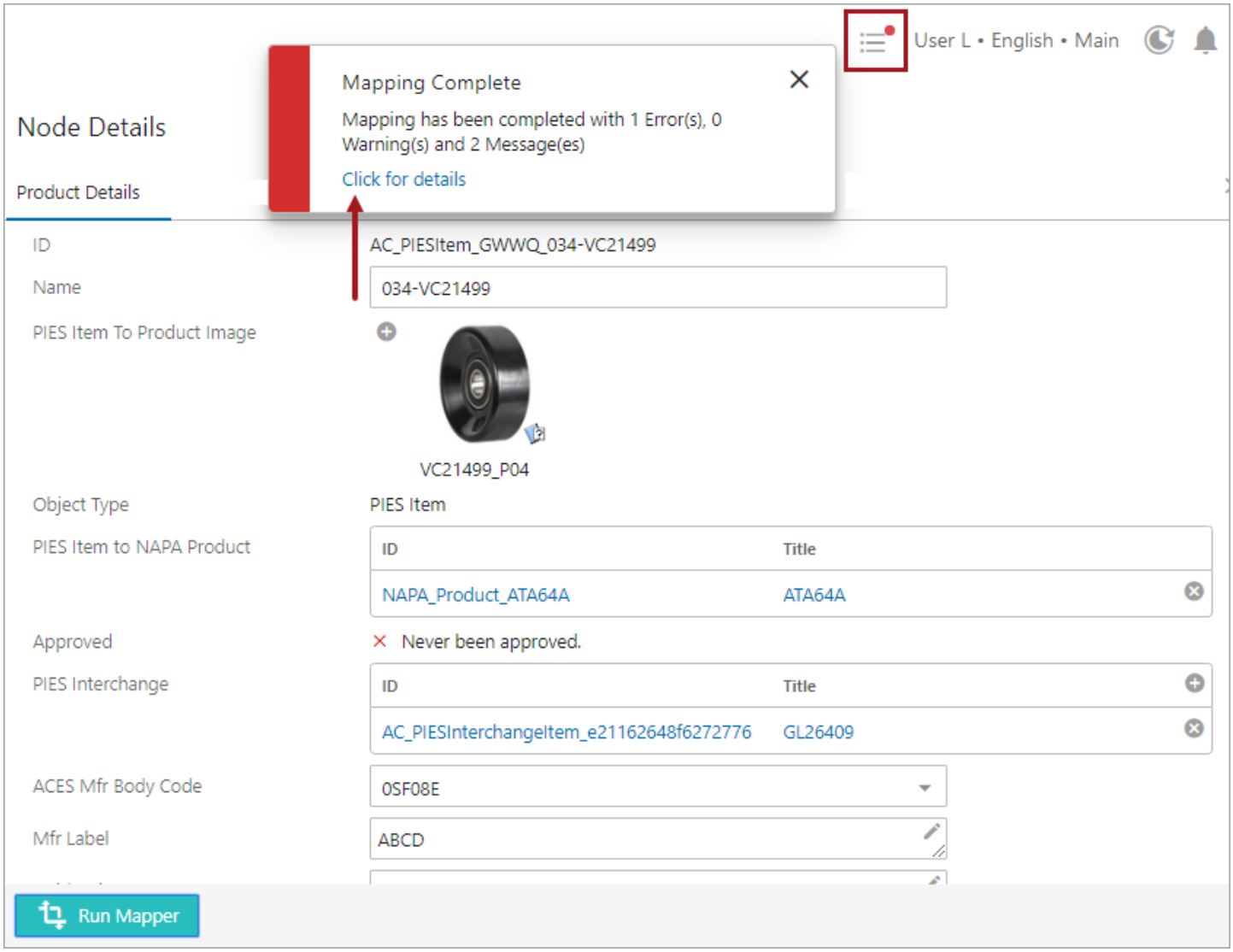



With the parameter checked, the "Mapping Complete - Mapping has been completed with X Error(s), X Warning(s) and X Message(es)" message with the 'Click for details' link will display in the message dialog (as shown below). If this parameter is left unchecked, any errors and warnings encountered in the execution of the Mapper Configuration setup entity will not be displayed, and the message dialog displays as 'Mapping Complete.'

- **Run Before Action:** This parameter allows the user to configure a business action. The configured business action gets executed before the Mapper Configuration setup entity is executed.
- **Run From Target:**This parameter allows the user to execute the Mapper Configuration from the target object. When the Run From Target parameter is selected, users can initiate the onboarding from the target object. The onboarding of the data will continue the same way as it was run from the source object. It is required that the selected target object has a valid source object.

5. Click the **Add** button to close the Mapper Action Properties designer screen.
6. Click the **Save** button in the designer and **Close**.

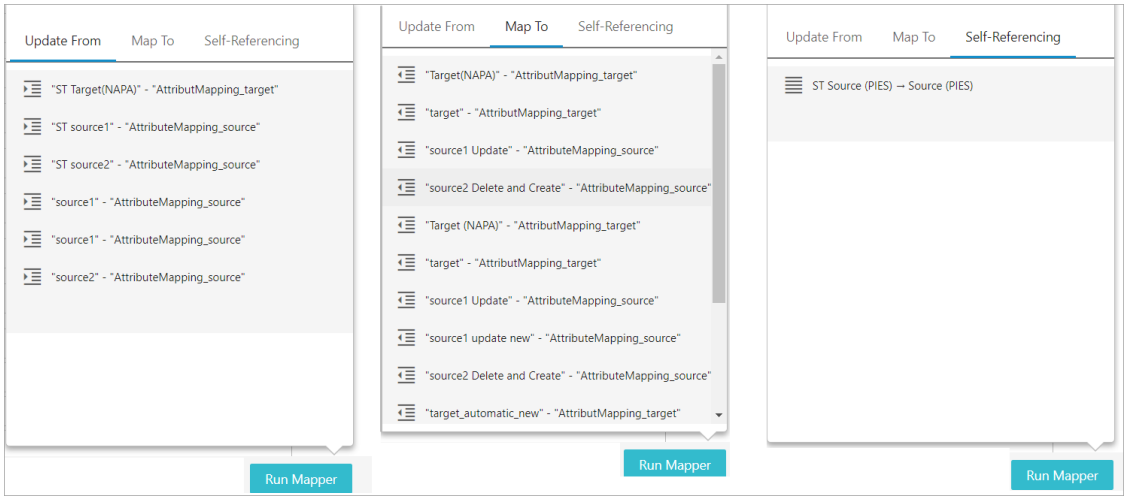
A status message dialog will be displayed when the user clicks on the action button that is configured in any screen. This dialog shows the execution status of the Mapper Configuration and configured business action execution in the top center of the screen. More details on the status of the execution can be found either by clicking the 'Click for details' link in the message dialog, or by accessing it through the 'Recent warning notifications' side panel located in the upper right corner of Web UI.




## Executing Through Stacked Mapper Action Button

This process involves defining an action button called **Stacked Mapper Action** to hold multiple Mapper Configuration setup entities, and configuring the action button in required screens. Unlike the Mapper Action button described above, this action button provides an option to configure multiple Mapper Configuration setup entity, thereby allowing users to select a Mapper Configuration setup entity standing on the object.

Clicking on the action button will display a dialog (with one or more tabs depending on the Mapper type configured) where users can select a Mapper Configuration setup entity and on the selected node. The Map To tab contains the Mapper Configuration setup entities that are to be run on the source object, the Update From tab contains the Mapper Configuration setup entities that are to be run from the target object, and the Self-Referencing tab contains the Mapper Configuration setup entities that are being self referenced.




The **Stacked Mapper Action** button can be configured to be available within the Source object (or Target object) screen, or any other places that allows the action button to be executed (e.g., run as a bulk update on a Collection, or initiate the action button with a node selected in the Search by Card screen). For the ease of understanding, this document depicts an example of configuring the Stacked Mapper Action button in the Node Detail screen that is configured to view the Source object.

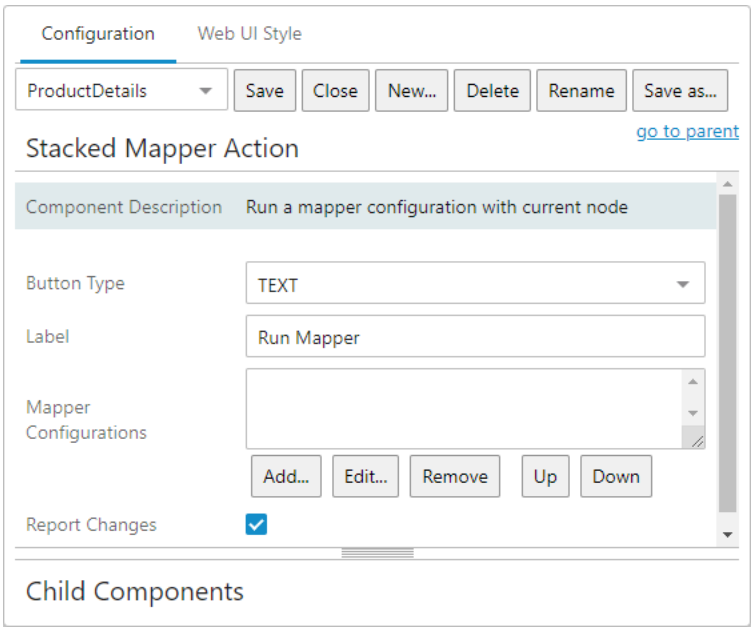
**Note:** If the Mapper Action button is to be configured in the Target object screen, then it is required that the user selects the appropriate option in the Run On parameter as explained in this topic below.

Following are the steps to configure the Mapper Action button.

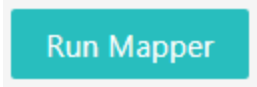
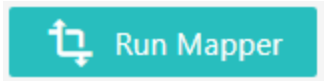
1. In the Web UI designer, navigate to the screen that requires the Mapper Action button to be configured > Buttons > go to component > Button Properties.
2. Click **Add** next to the Actions field that is available within the Buttons Properties to access the Add

Component dialog.

3. Find **Stcked Mapper Action** and click **Add**, and the Stacked Mapper Action Properties dialog will be displayed.

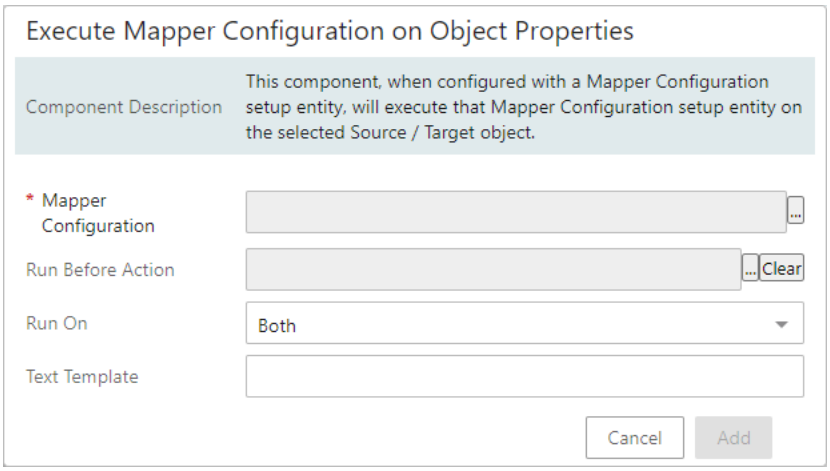


4. Populate the parameters as detailed below:
   - **Button Type:** Select from ICON_AND_TEXT, ICON, and TEXT using the Button Type dropdown. If the user choose to select TEXT or ICON_AND_TEXT and the Label parameter is empty, the default text 'Run Mapper' will display.

| Button Types | Button Displayed As |
|---|---|
| ICON | |
| TEXT |  |
| ICON_AND_TEXT |  |

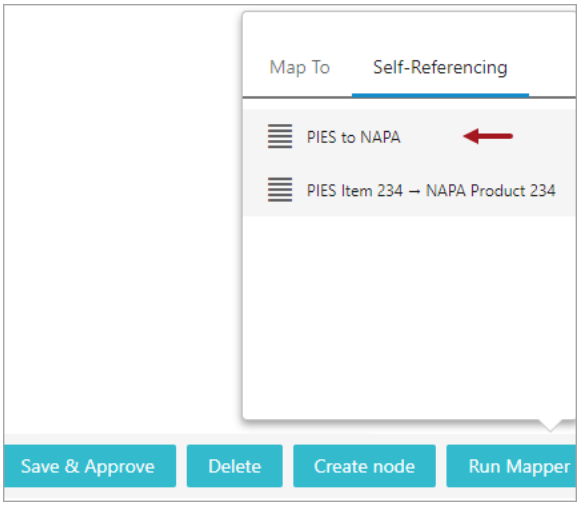


- **Label:** This parameter defines the text that the user wants to appear in the button that is displayed on screen.
- **Mapper Configurations:** Required parameter to specify the Mapper Configuration setup entity(ies) that needs to be executed. Click the Add button to display the 'Execute Mapper Configuration on Object Properties' dialog where users can find and select the Mapper Configuration setup entity. Further configuration is required within the 'Execute Mapper Configuration on Object Properties' dialog as explained in detail below:

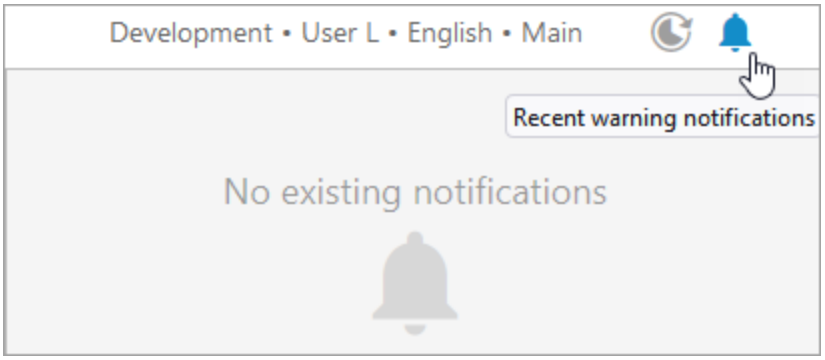



  - **Mapper Configuration:** Required parameter to specify the Mapper Configuration setup entity that needs to be executed. Click the ellipsis button (...) to find and select the Mapper Configuration setup entity that is available in the system.

- **Run Before Action:** This parameter allows users to configure a business action. The configured business action gets executed before the Mapper Configuration setup entity is executed.
- **Run On:** This parameter contains options that determine whether the Mapper Configuration setup entity is to be executed on the source object, target object, or both. The available options are Source, Target, and Both.
- **Text Template:** This field allows users to determine how the Mapper Configuration setup entity has to be displayed in the dialog when the Stacked Mapper action button is clicked.



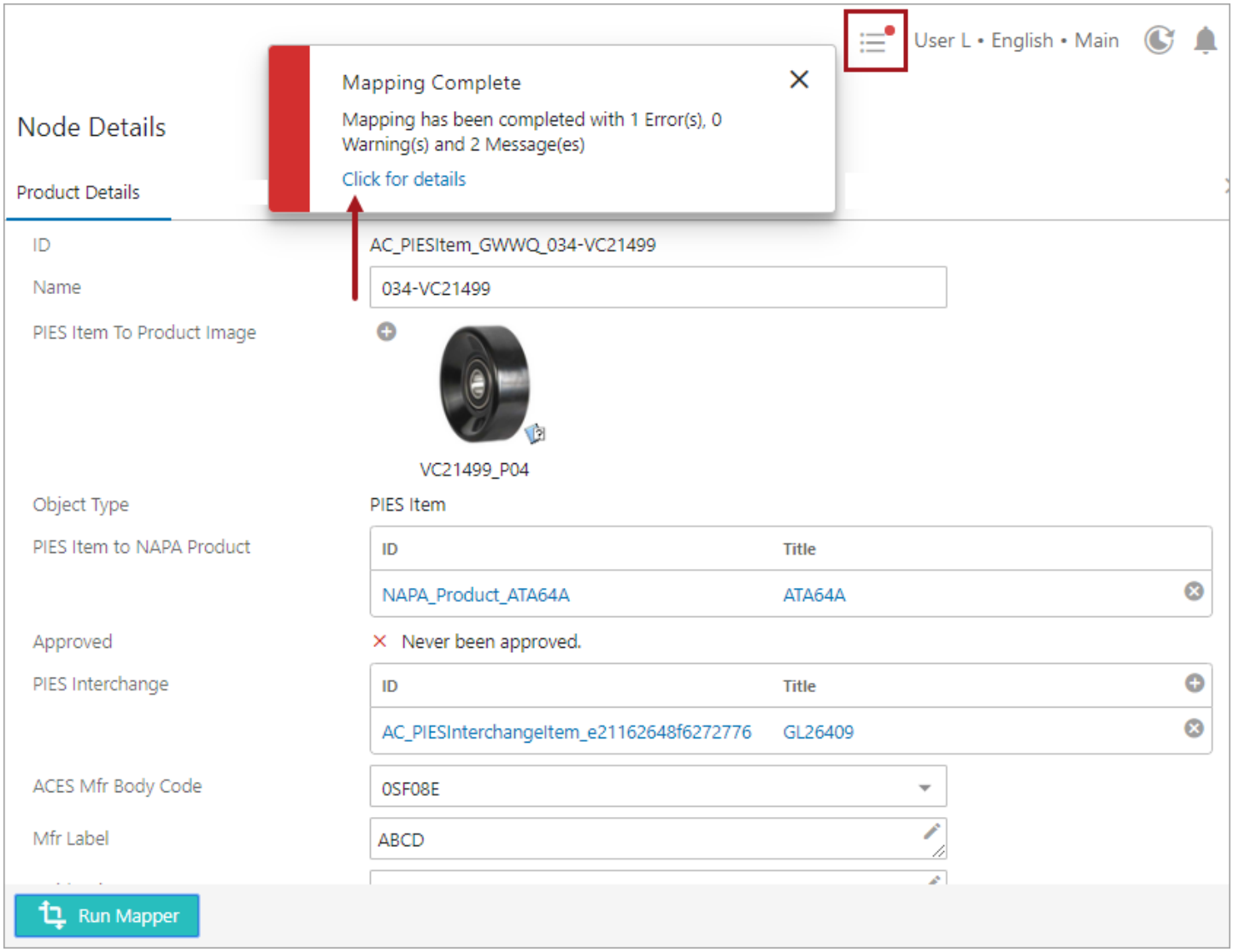

- **Report Changes:** This parameter is checked by default. Having this parameter checked ensures that the changes effected by the execution of the associated Mapper Configuration setup entity is displayed in the message dialog (explained below) and also in the 'Recent warning notifications' panel, located in the upper right corner of Web UI.




With the parameter checked, the "Mapping Complete - Mapping has been completed with X Error(s), X Warning(s) and X Message(es)" message with the 'Click for details' link will display in the message dialog (as shown below). If this parameter is left unchecked, any errors and warnings encountered in the execution of the Mapper Configuration setup entity will not be displayed, and the message dialog displays as 'Mapping Complete.'

5. Click the **Add** button to close the Mapper Action Properties designer screen.
6. Click the **Save** button in the designer and **Close**.

A status message dialog will be displayed when the user clicks on the action button that is configured in any screen. This dialog shows the execution status of the Mapper Configuration and configured business action execution in the top center of the screen. More details on the status of the execution can be found either by clicking the 'Click for details' link in the message dialog, or by accessing it through the 'Recent warning notifications' side panel located in the upper right corner of Web UI.




## Executing by Onboarding Comparison Screen

This process involves configuring an Onboarding Comparison Screen to hold the Mapper Configuration and then access this screen for the Source object through multiple ways. This type of execution provides an option for the user to preview the changes affecting the Target object even before the data is onboarded. The Onboarding

Comparison Screen allows users to assess each of the changes, and gives the user an option to accept or reject the changes that will affect the Target object. Additionally, this screen also provides users an option to exclude any Mapping plugin being executed for any required Source object on its subsequent import.

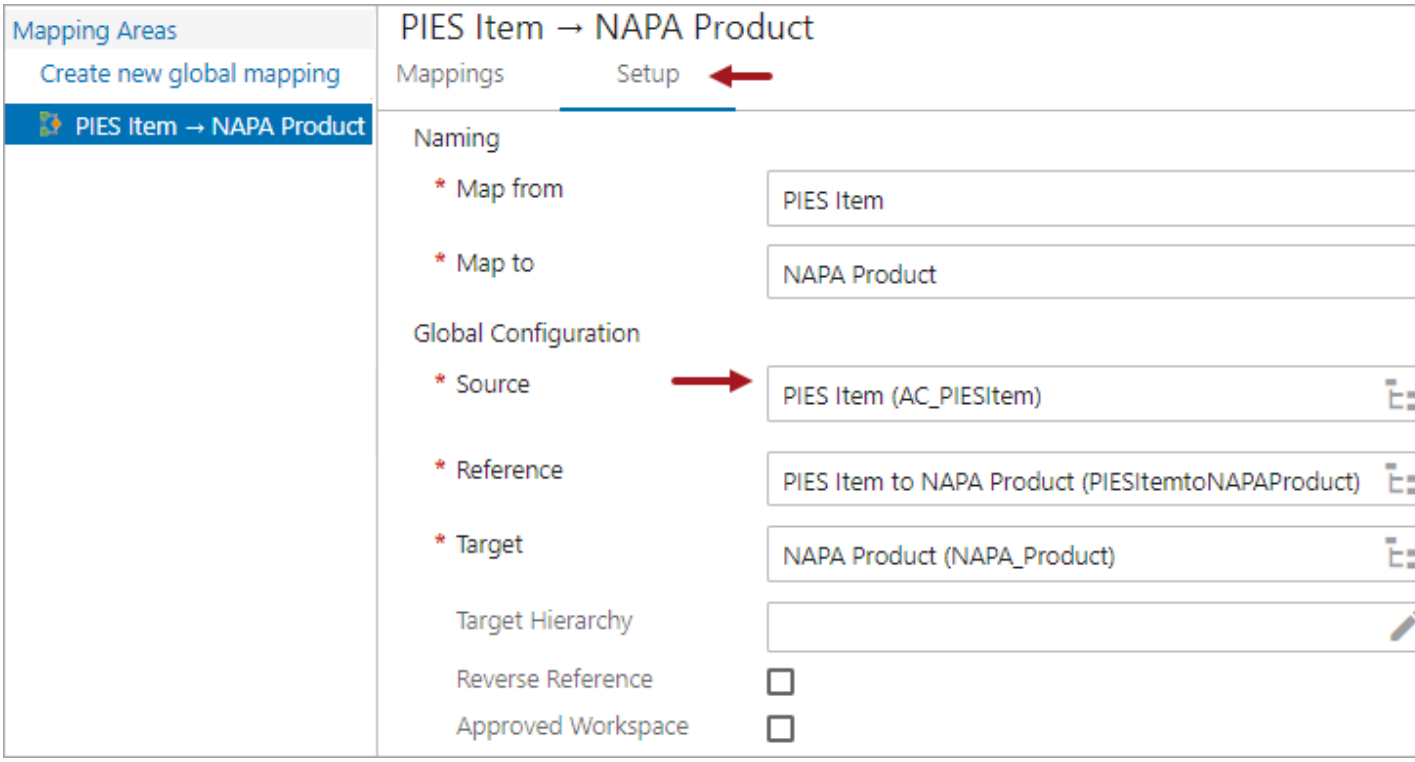
1. Create and configure a Onboarding Comparison Screen. For the full set of instructions on configuring the Onboarding Comparison Screen, refer to the **Configuring Onboarding Comparison Screen** documentation.
2. Review and implement the most common ways to use Onboarding Comparison Screen as described in **Using Onboarding Comparison Screen Interface** topic within this guide.

## Executing by a Background Process

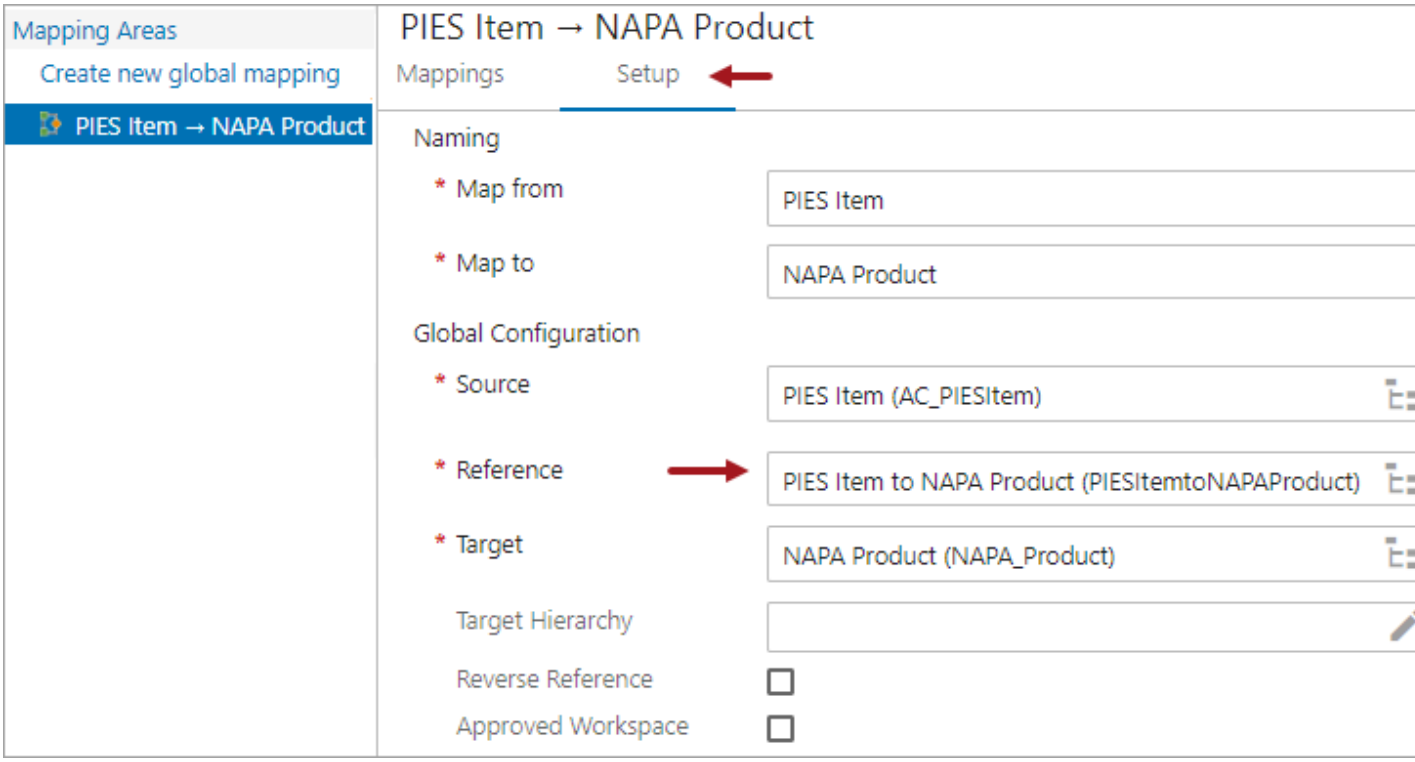
This method of execution is also considered as the automatic onboarding method. This process involves creating a new state in the Import workflow to hold a Mapper Configuration setup entity that gets automatically executed whenever there is a new import. After the file is imported in the system, the Onboarding background process runs a drill down search for the following:

- To identify *all* the **Source objects** in the system that has the change flag attribute value set to 'true.' The change flag attribute is configured in the 'Changed attribute' parameter (explained below). The Source object (s) belong to the object type configured in the 'Source' parameter that is available within the 'Setup' tab of the Onboarding Mappings Details screen for the selected Mapper Configuration setup entity (an example shown below).

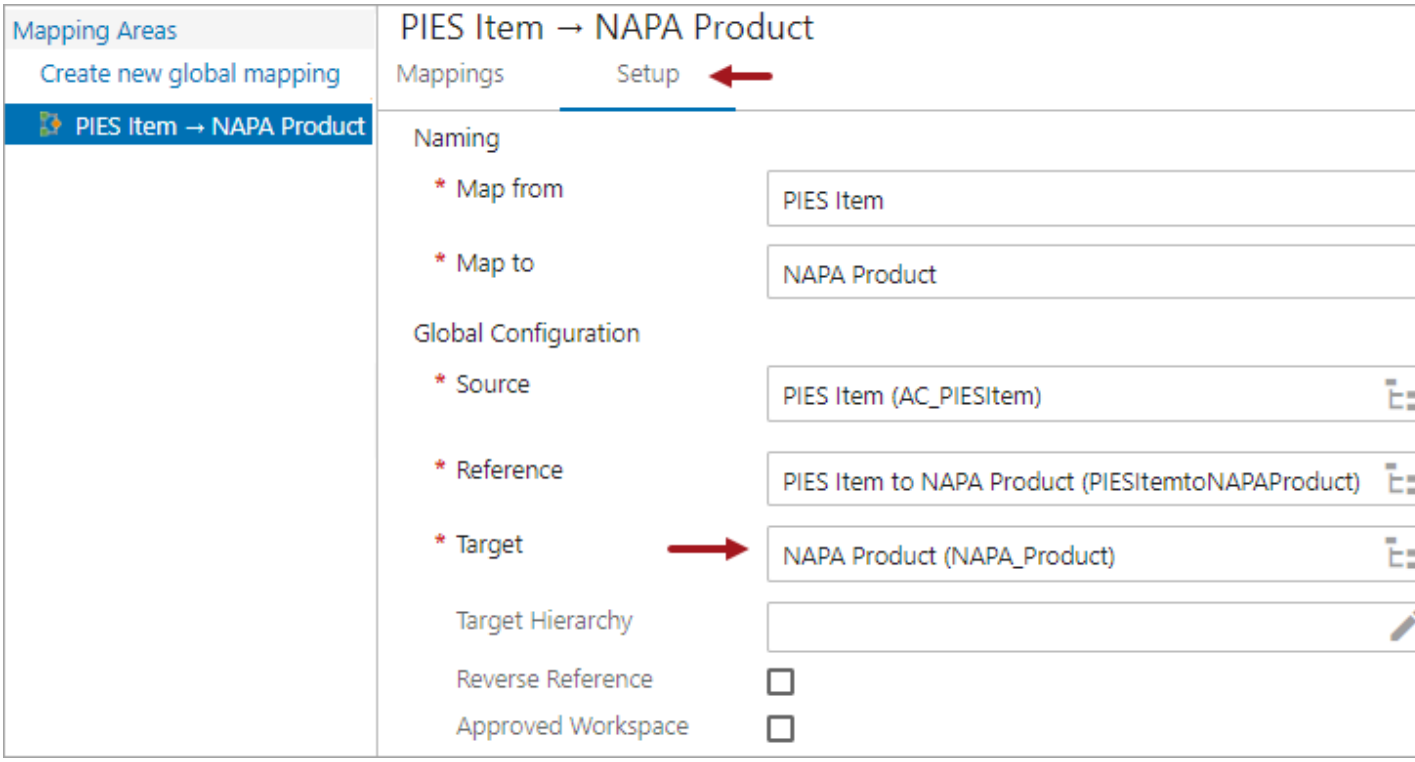
**Note:** When executing through this method, the background process executes the Mapper Configuration setup entity on *every source objects residing in the system* that has the change flag attribute value set to 'true.' *Not* just the source objects in the importing file.



- To identify the **Reference type** configured in the 'Reference' parameter which is available within the 'Setup' tab of the Onboarding Mappings Details screen for the selected Mapper Configuration setup entity (an example shown below).




- To identify the **Target object** belonging to the object type configured in the 'Target' parameter which is available within the 'Setup' tab of the Onboarding Mappings Details screen for the selected Mapper Configuration setup entity (an example shown below). If the Target object does not exist to run the Mapper Configuration on, then it can be made possible to create a Target object based on the condition defined in the 'Target Hierarchy' Parameter.




Once the Onboarding background process identifies the Source object, Target object, and the reference linking the Source object with the Target object, then the Mapper Configuration setup entity will be executed only on the identified results. This way, Onboarding background process eliminates execution of the Mapper Configuration on all other objects in the system than those that actually requires to be updated.

To execute the Mapper Configuration by a background process, the following configuration is needed:

These levels of configuration are explained in detail below. For more information on creating / editing the business action and workflows, refer to the **Business Action** section, and the **Workflows** section of the **STEP Online Help**.

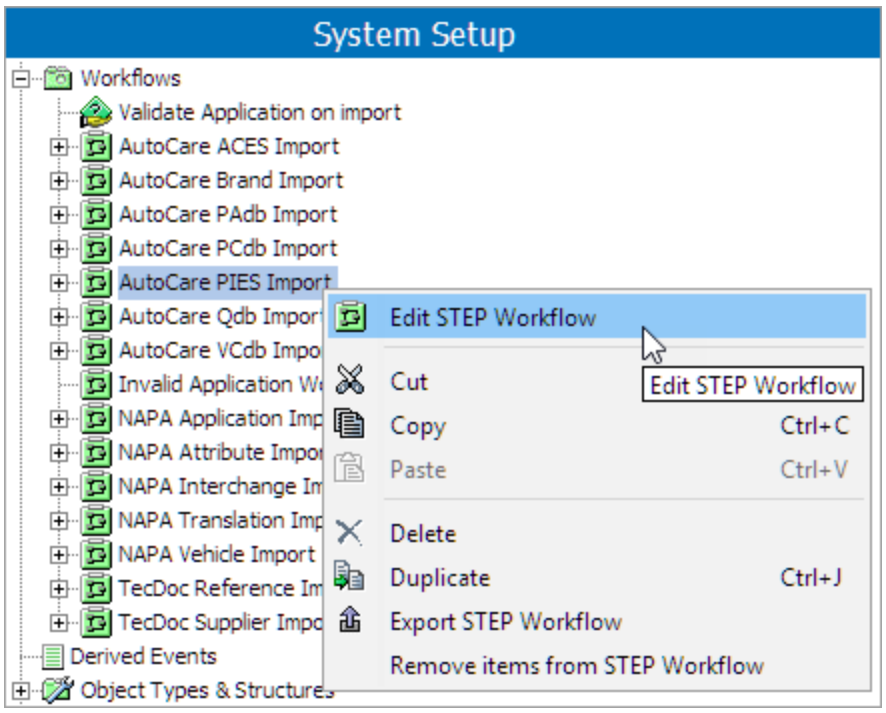
### Configuring the Onboarding state to execute Mapper Configuration setup entity in the Workflow

The Mapper Configuration setup entity *cannot* be executed on any existing state in the workflow. To achieve this, a new state after the Import state needs to be created, and a business action that triggers a background process to execute the Mapper Configuration setup entity is added to the newly created onboarding state in the import workflow. Below are the steps to create and update a workflow to execute the Mapper Configuration setup entity.

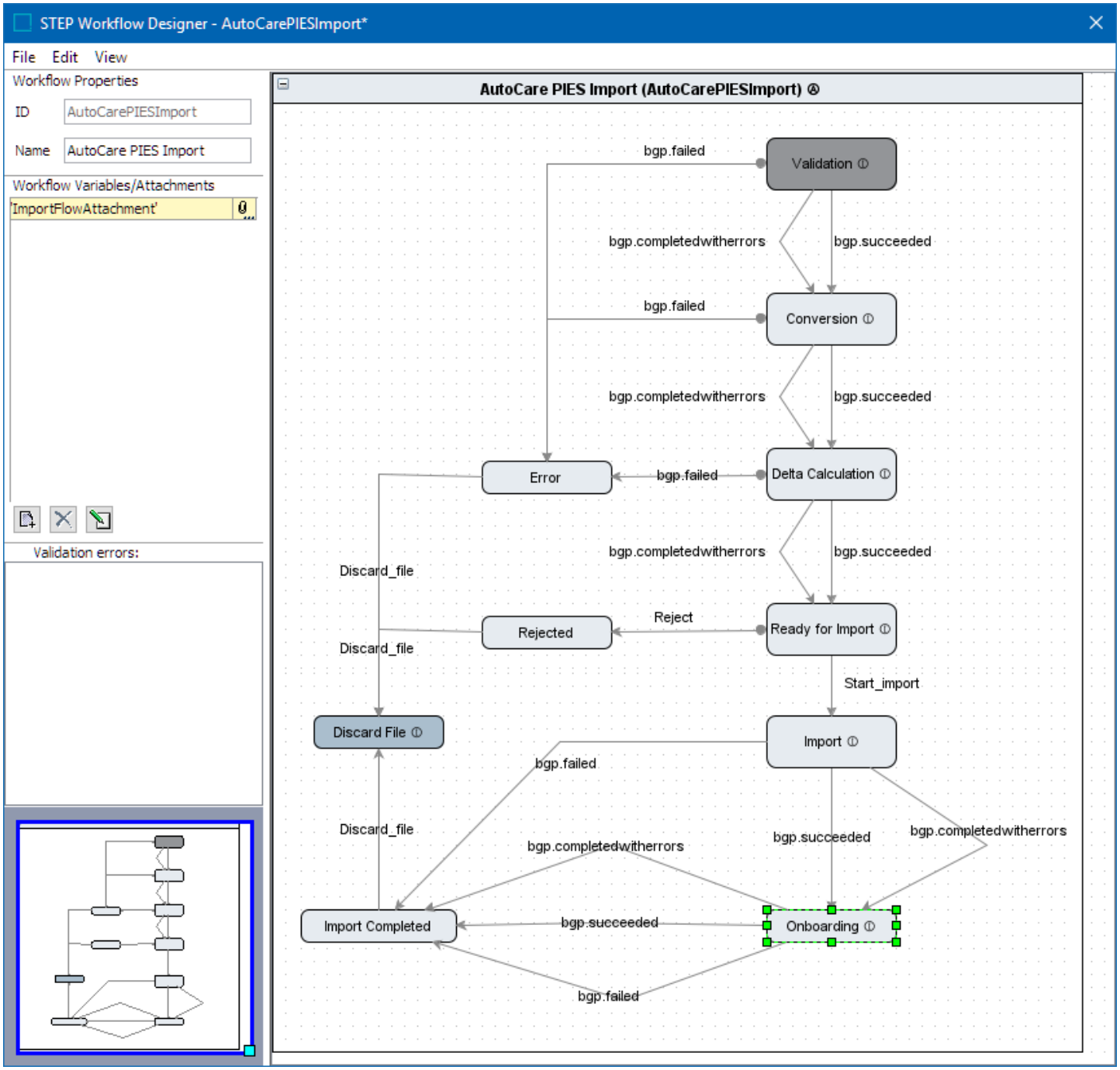
1. Go to System Setup > Workflows > Select the workflow used by the importer.

   For this example, AutoCare PIES Import workflow is used.

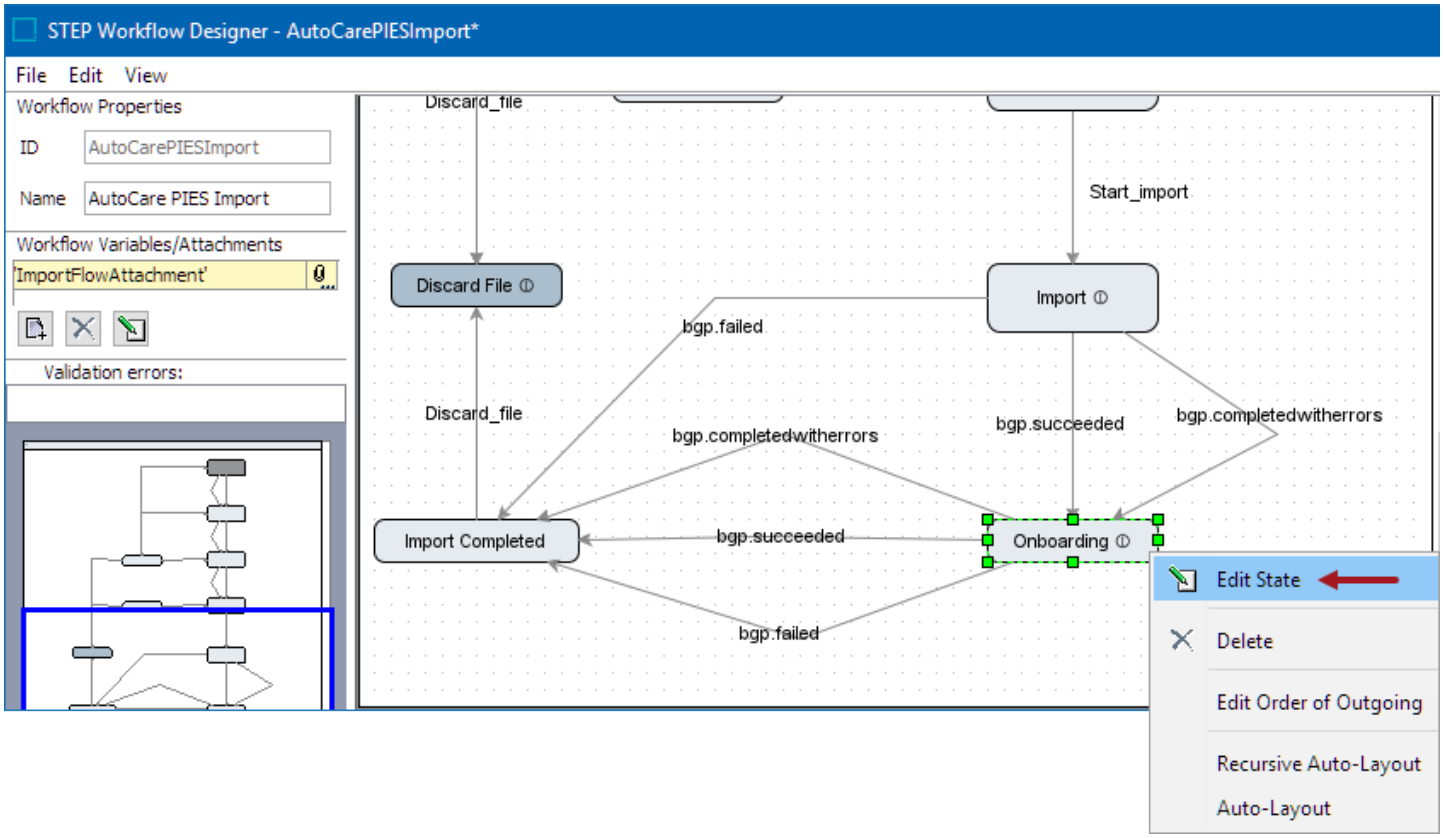
2. Right-click the respective workflow, select **Edit STEP Workflow**, and the STEP Workflow Designer will display.



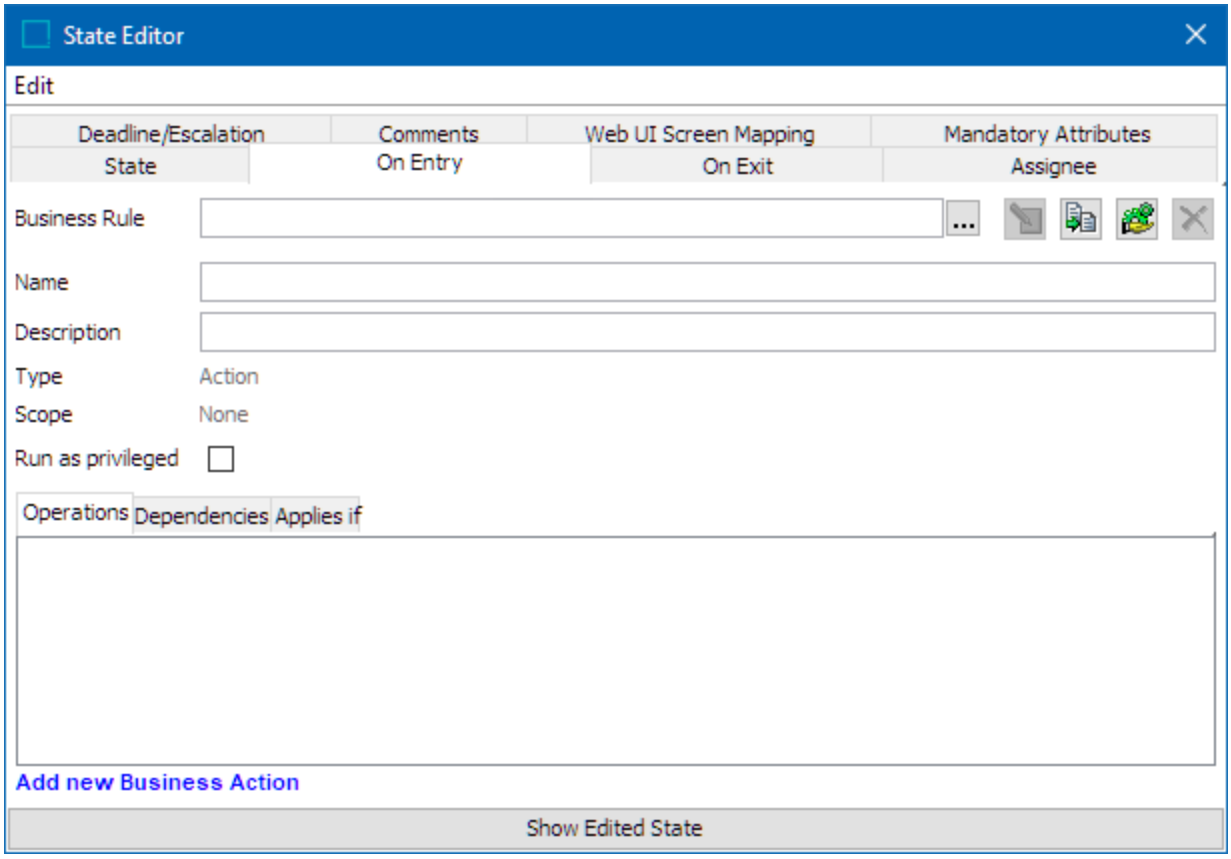

3. Within the STEP Workflow Designer, create a new state between the Import state and Import Completed state. In this example, a new state called 'Onboarding' is created (as shown below). For detailed instructions on creating States and Transitions, refer to **Creating a Workflow** topic of the **Workflows** section of the **STEP Online Help**.



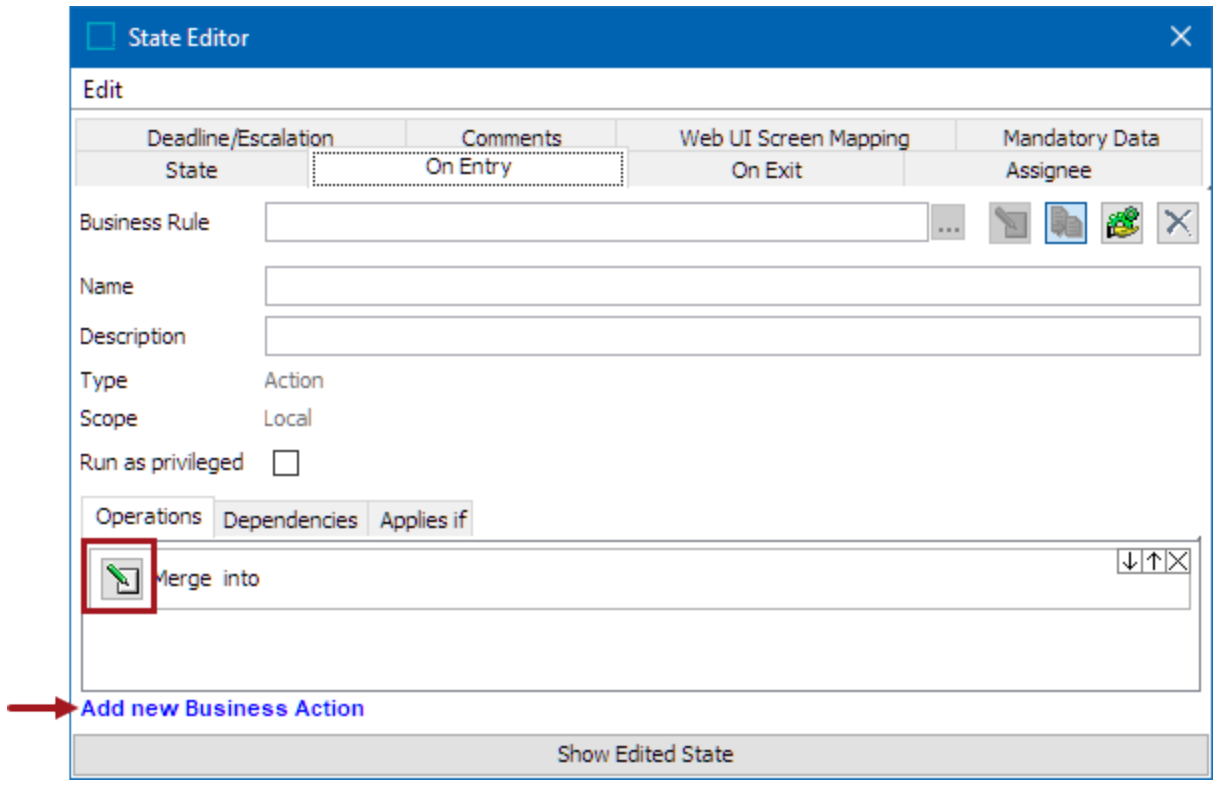

4. Right-click the newly created state (in this case it is 'Onboarding' state), select **Edit State**, and the State Editor will display.



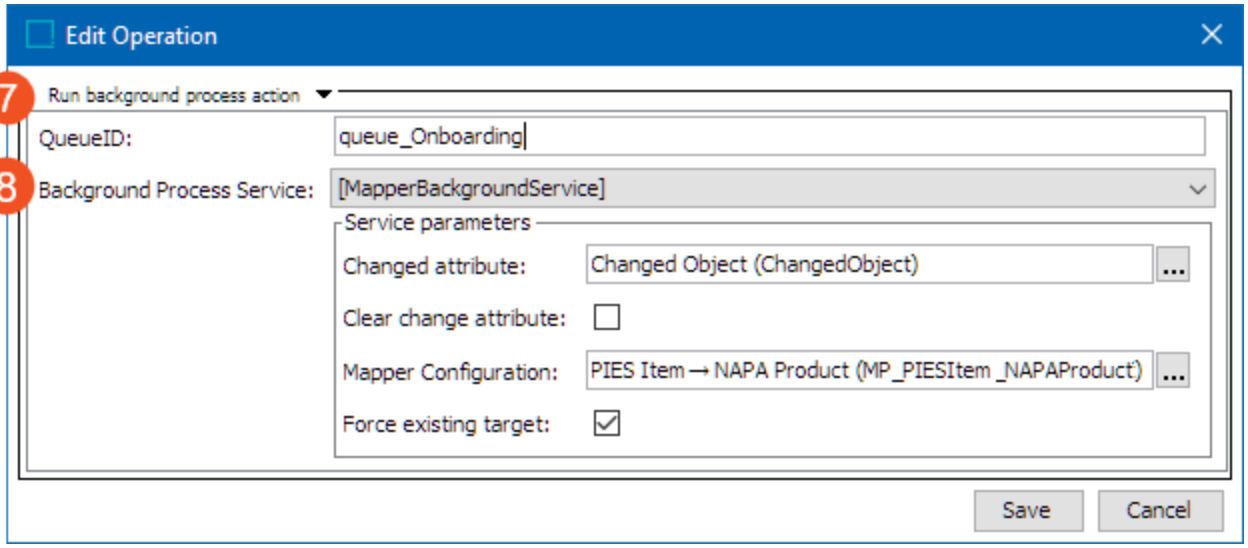

5. Within the State Editor, click the **On Entry** tab (as shown below).




6. Click on the **Add new Business Action** link to create a local action and click the edit button ( ) to open the Edit Operation dialog. For more information on creating a business rule, refer to the **Creating a Business Rule, Function, or Library** topic of the **Business Rules** section of the **STEP Online Help**.



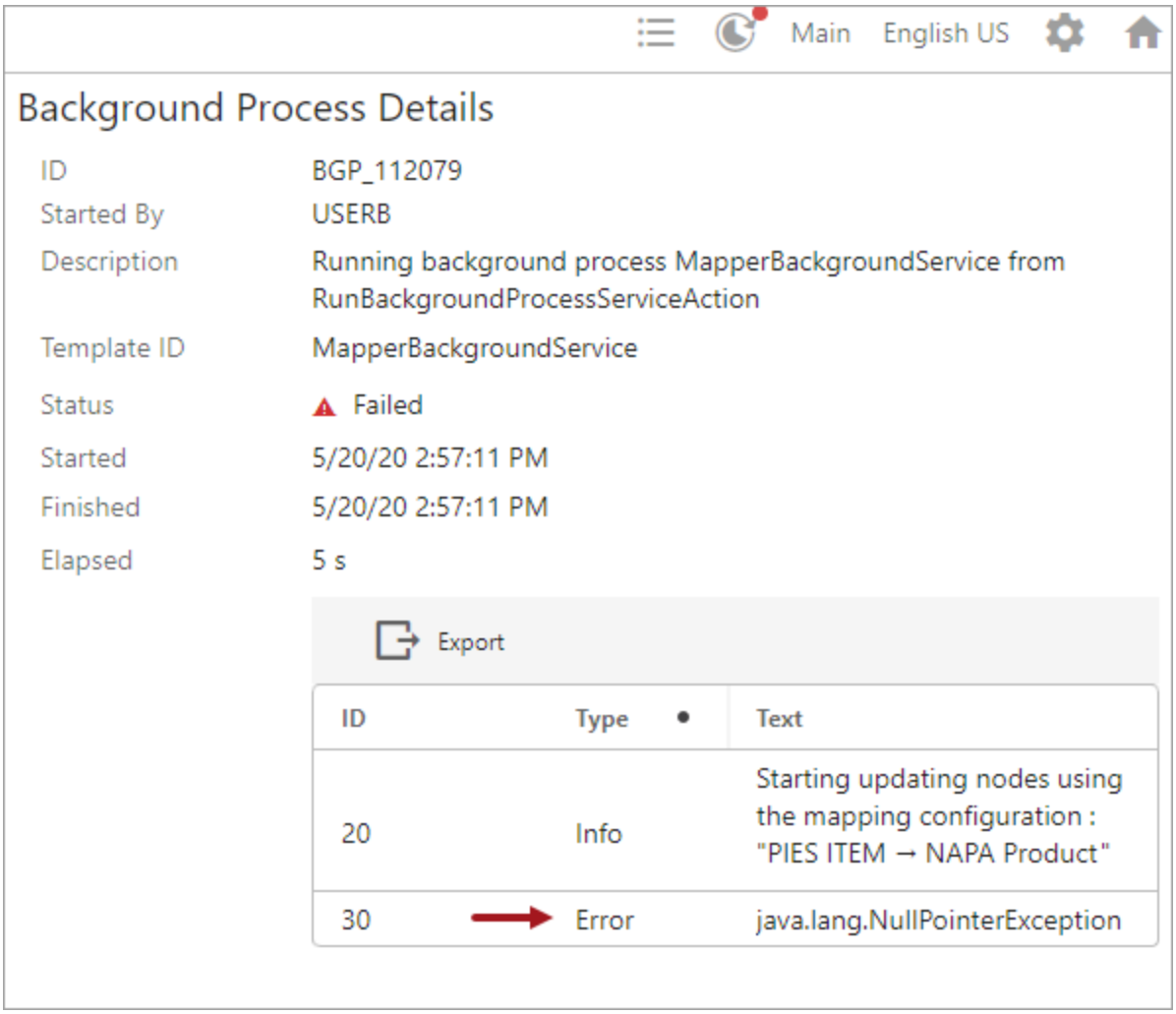

7. Use the dropdown menu within the Edit Operation dialog to select **Import flow** > **Run background process action** operation.




8. 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:** This is the required parameter that must be populated with the same attribute that is configured within the 'Updated values attribute' parameter in the 'Set import status attributes' operation. Click the ellipsis button (...) to find and select an attribute that is used to identify when an object is changed while importing. When the attribute is configured, the system will search for all the source 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, refer to the **Business Action: Set Import Status Attributes** topic of the **Automotive Reference Guide** found within the **Solution Enablement** section of **STEP Online Help**.

If the 'Changed attribute' parameter is not populated with the right attribute value, then the [MapperBackgroundService] will fail with an exception error. The screenshot below displays the Background Process of the onboarding state failed with the NullPointerException error.



- **Clear change attribute:** Selecting this parameter will clear the flag set in the attribute configured within the 'Changed attribute' parameter after the Mapper Configuration setup entity is executed. Checking this parameter prevents the Mapper Configuration setup entity being executed multiple times on the same Source object over subsequent imports.

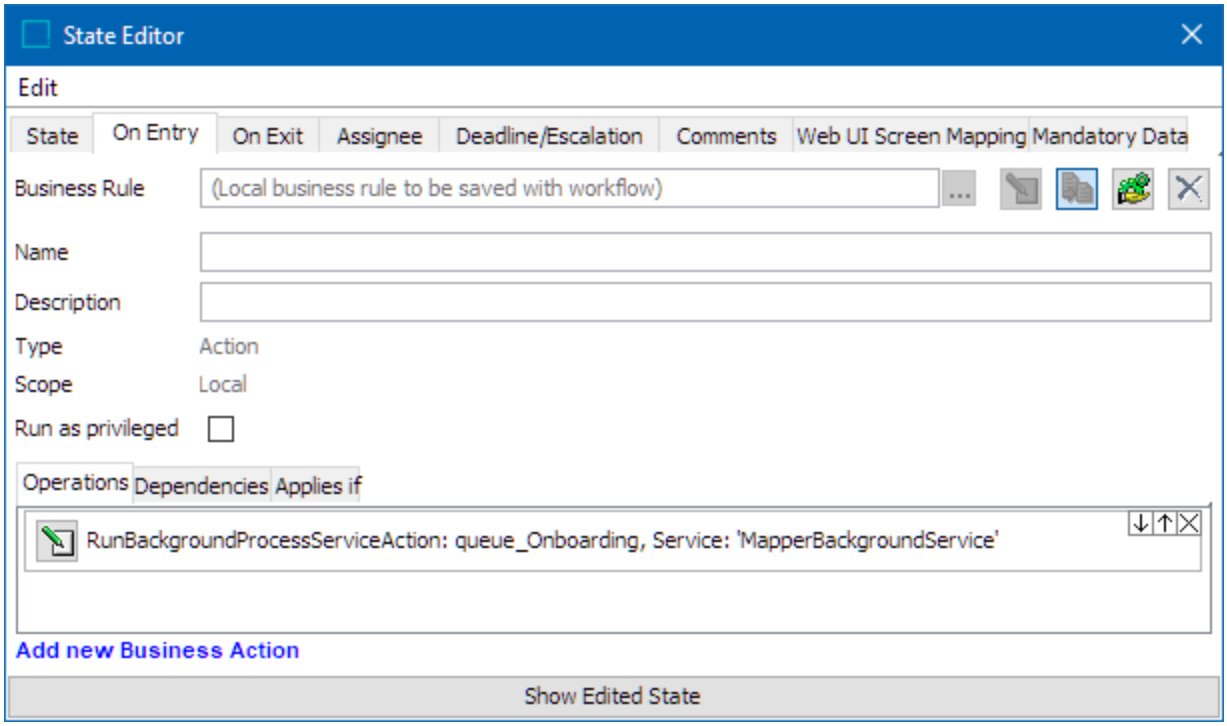
  **Note:** The 'Clear change attribute' parameter has to checked only when it is decided that the flags marked in the attribute will be of no more use by any other functionalities. If there are multiple Mapper Configuration setup entity being executed in the same workflow state, then this parameter is to be only checked in the last Mapper Configuration setup entity that gets executed.

- **Mapper Configuration:** Required parameter to specify the Mapper Configuration that needs to be executed.
- **Force existing target:** Selecting this parameter will allow onboarding the data only if the source object has an existing target object. Thus, it will not allow creation of new targets irrespective of the presence / absence of the target hierarchy defined in the Target Hierarchy parameter that is available within the 'Setup' tab of the Onboarding Mappings Details screen. Deselecting this parameter will allow creation of the target object if necessary, provided that the target hierarchy is defined in the Target Hierarchy parameter that is available within the 'Setup' tab of the Onboarding Mappings Details screen.

9. Click the **Save** button to save and close the configuration.

   **Note:** Alternatively, users can also create a global business action and then configure that action in the 'Business Rule' parameter by clicking the ellipsis button (...) to find and select the business action created to set the Mapper Configuration after import. But, this method *hinders the performance* of the onboarding process. It is recommended that a local action is created in the state and configured accordingly.

   Below is an example of how the State Editor should look prior to saving:




10. Close the State Editor, and save and close the STEP Workflow Designer.

**Important:** Users *cannot* create multiple local actions to execute multiple Mapper Configuration setup entity in one single state. If multiple Mapper Configuration setup entity is to be executed, then the user has to create multiple onboarding states for each Mapper Configuration setup entity that is to be executed.

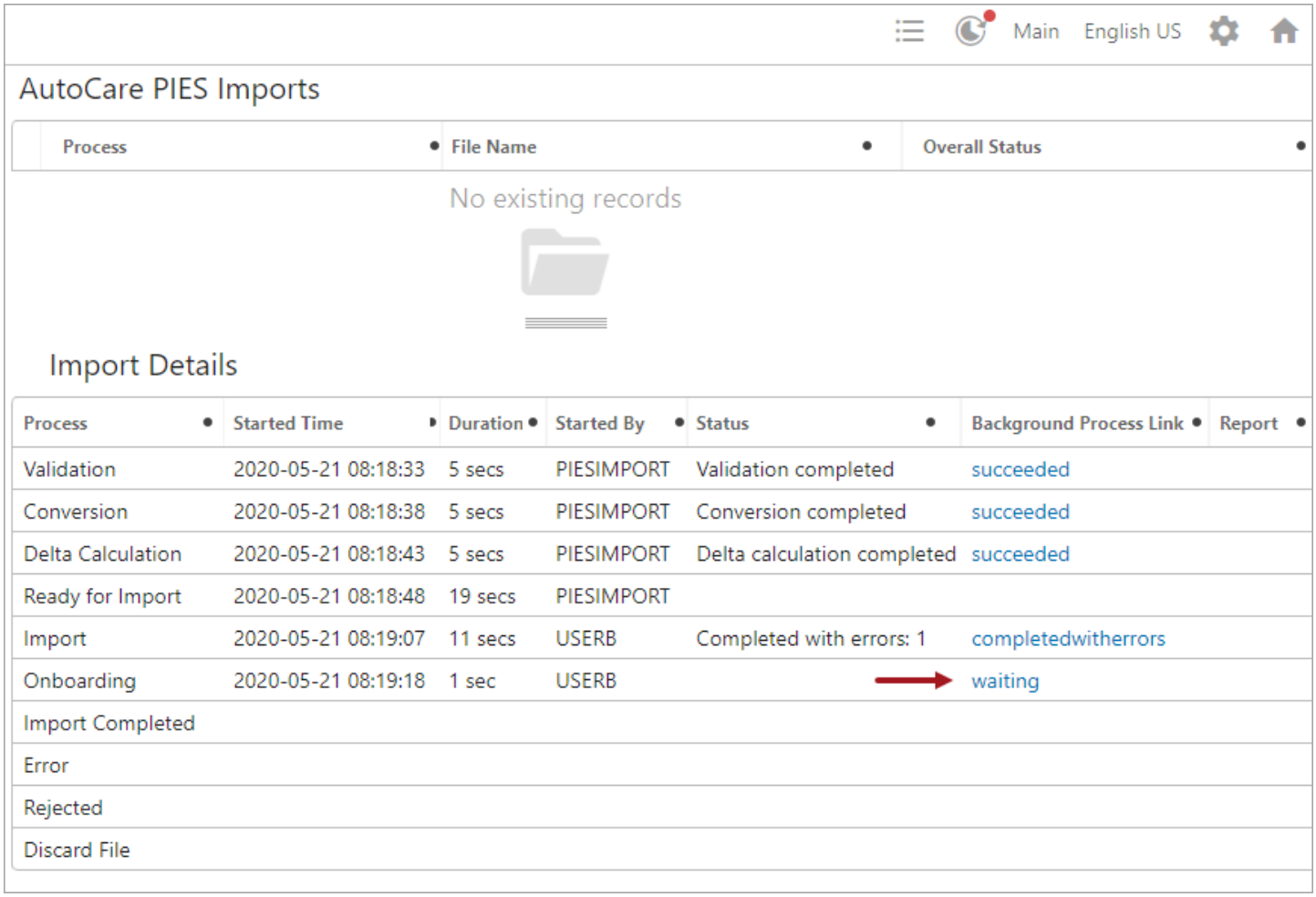
### Monitoring the automated Onboarding process

Monitoring an Onboarding process allows users to view the onboarding process' activity and status, and analyze the execution report for details.

Each onboarding state can have only one business action (Run background process action). For executing multiple Mapper Configuration setup entity, user has to create multiple onboarding states for each Mapper Configuration setup entity. Presence of multiple onboarding states allows each business actions within them to run as individual background processes and they all shares the common background process service (MapperBackgroundService).

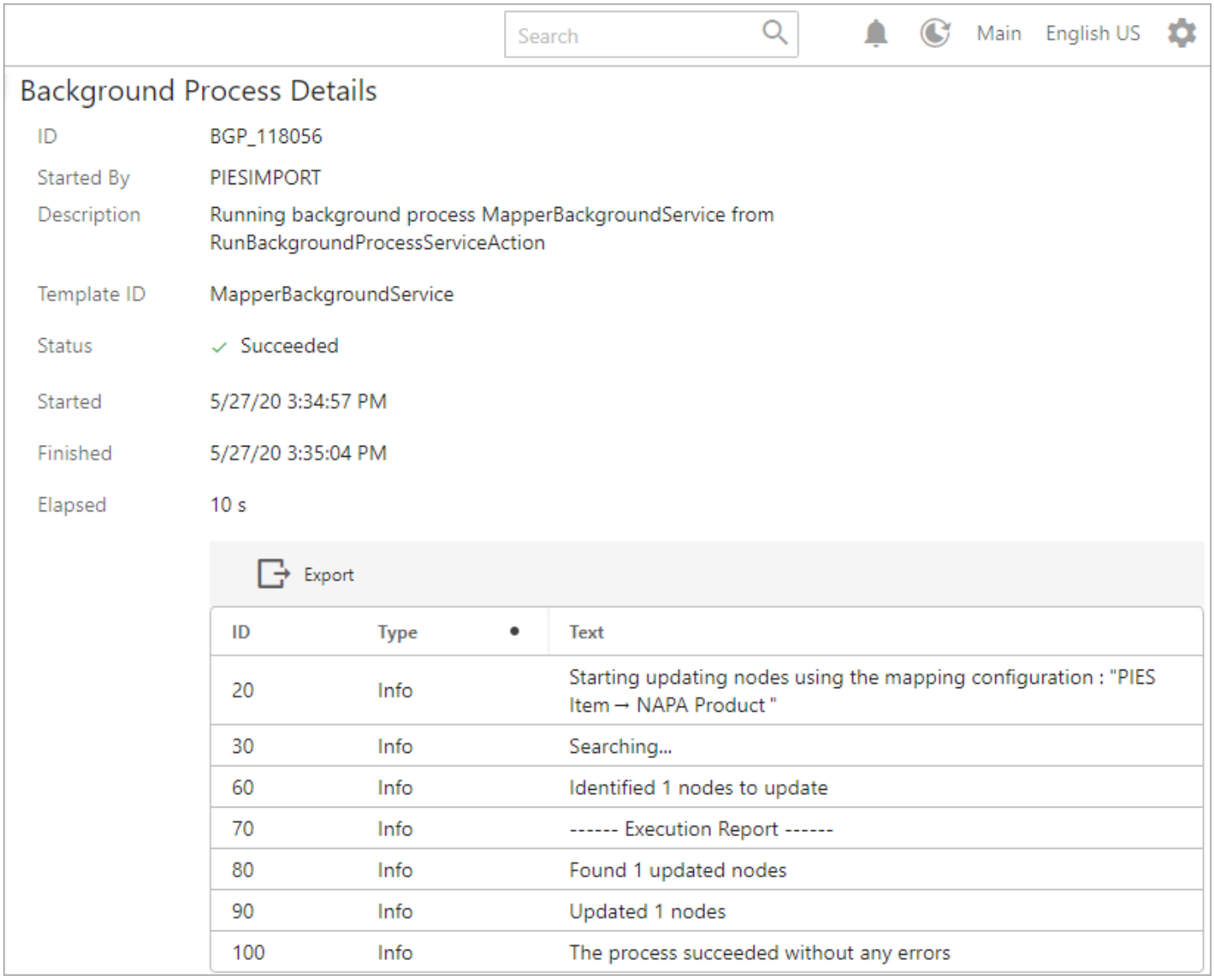
As the file is uploaded for an import in the Web UI, the Controller Entity moves through their associated Import workflow. Users can monitor the progress using their associated Status Selector Homepage Widget on the Web UI homepage and/or the left side panel. Clicking the Controller icon displays the Import Details below the Controller list. For information on how to access the Import Details screen, refer to the associated importer topics under the **Importing Automotive Data** section.

When the Controller reaches the Onboarding state in the workflow and executes the Mapper Configuration setup entity on all the Source objects with Changed attribute value = true, a background process (BGP) will be generated. Additionally, the Background Process Link column displays a link to the Background Process Details screen as 'waiting.'




Users can click the 'waiting' Background Process Link (shown above) to view the details of the onboarding process within the Background Process Details screen.

In the example below, the ID 'BGP_118056' displays the status as 'Succeeded' and has updated one object.




Following are the BGP Log output considerations for the automated onboarding process:

- Progress of the objects updated is reported per 100 updated objects
- Any direct failure to onboard an object is reported as an exception (this does not stop onboarding of remaining objects)

## Executing by a Business Rule

**Note:** This is an outdated method that is still being used for execution in Web UI. Users are suggested to use the Mapper Action button instead (as detailed above in this topic).

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 can occur with a 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, refer to 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**.

# 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, refer to the **Using the Onboarding Mappings Details Screen** topic.

## Application Mapping Plugin

The Application Mapping plugin will copy application records from the source object to the target object. In addition, it also designates vehicle and part type references to the newly created target application records. The Application Mapping plugin is added and configured in the Onboarding Mappings Details Screen for the selected Mapper Configuration.

## Prerequisites

There are terminology variations within the Application Mapping plugin for each standard.

For example:

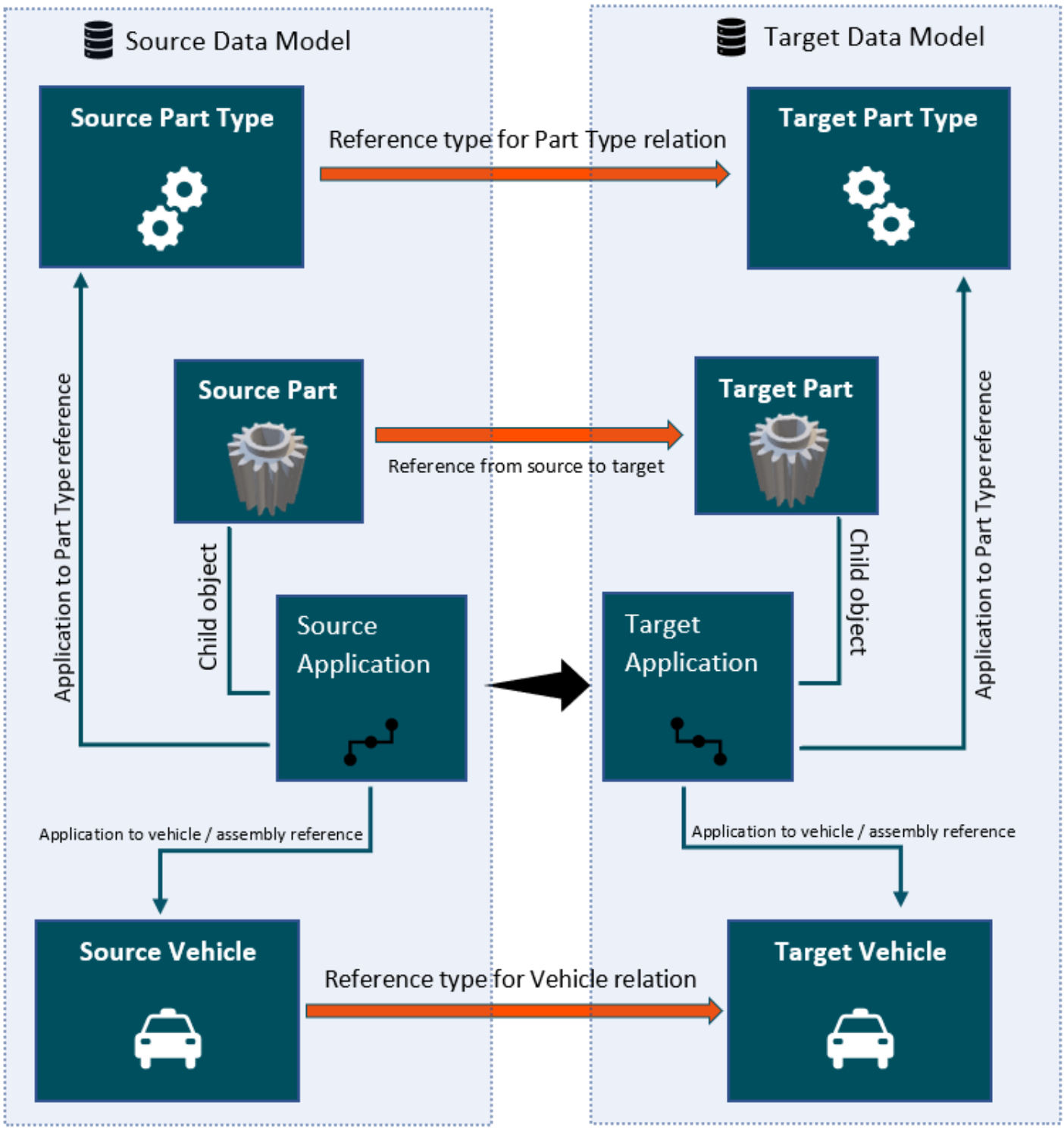
- The AutoCare addresses terminologies like Vehicle, Part Terminology, and Part
- The TecDoc addresses terminologies like Assembly, Generic Article, and Article
- Some users will have their application records applicated to vehicles, assembly, or engines, which they commonly address as targets

Despite these differences, the overall functionality that allows users to onboard application records remains the same. For ease of understanding, this topic is documented with an example where the user onboards application records from AutoCare standard to the NAPA standard. For more information about variations across different standards, refer to the **PMDM for Automotive Terminology** topic within the Automotive Reference Guide.

Before creating the Application Mapping plugin, the following items must be done:

- Ensure that there is a reference established between the object type of the source application record and the object type of the vehicle
- Ensure that there is a reference established between the object type of the source application record and the object type of the part type
- Establish a reference between the object type of the source part and the object type of the target part

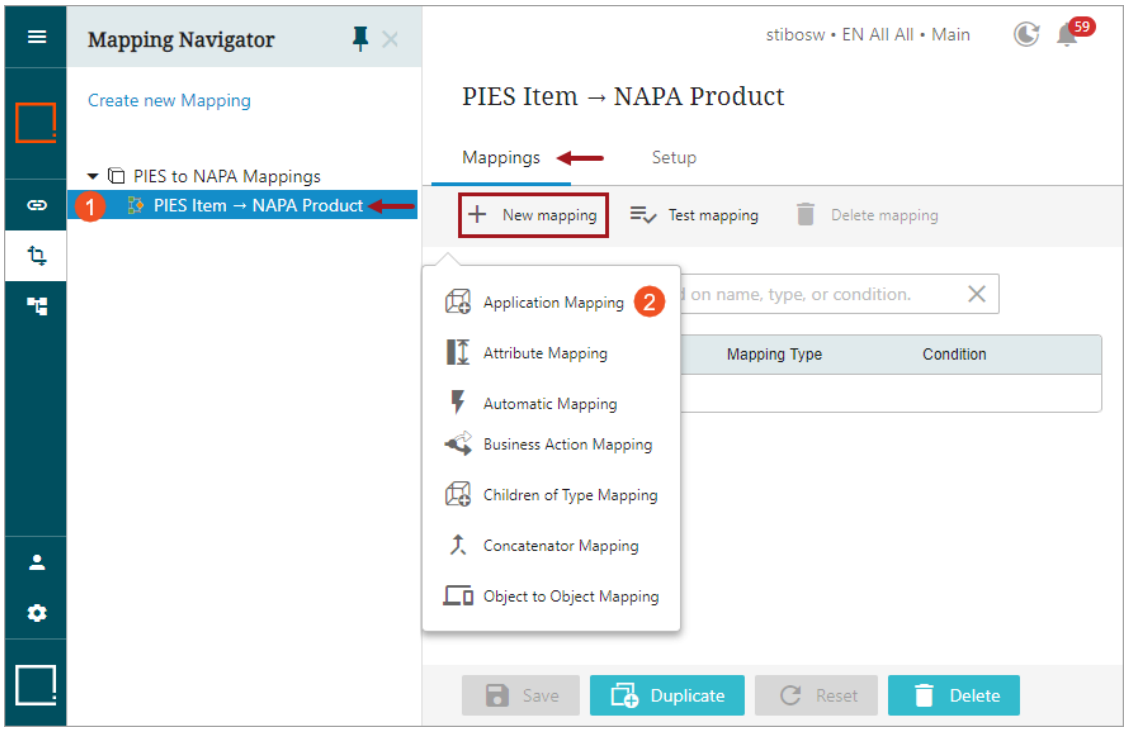
- Establish a reference between the object type of the source vehicle and the object type of the target vehicle
- Establish a reference between the source part type and the target part type



## Configuring Application Mapping Plugin

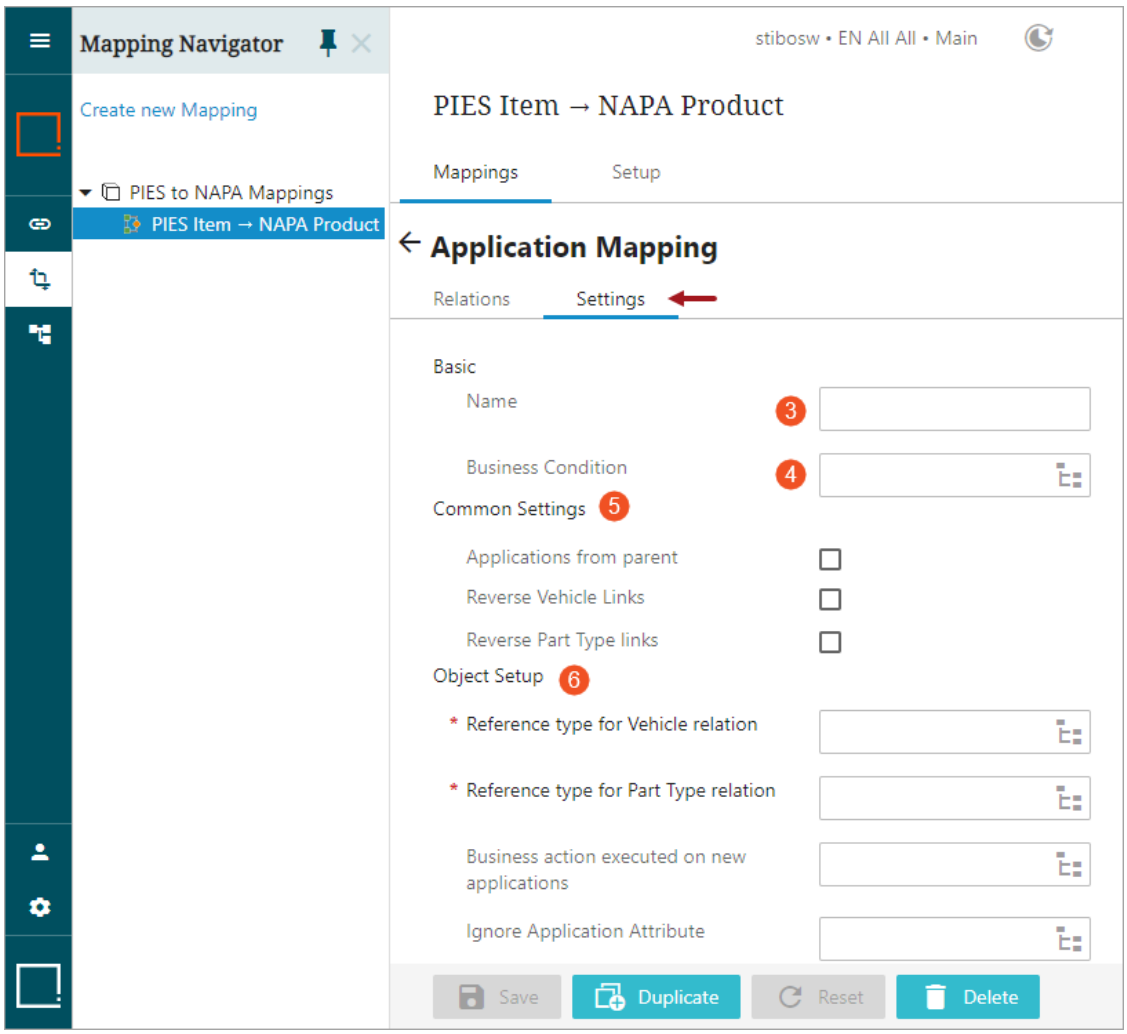
To configure the Application Mapping plugin, follow these steps:

1. With the required Mapper Configuration selected, click on the **New mapping** button available within the Mappings tab of the Onboarding Mappings Details screen.



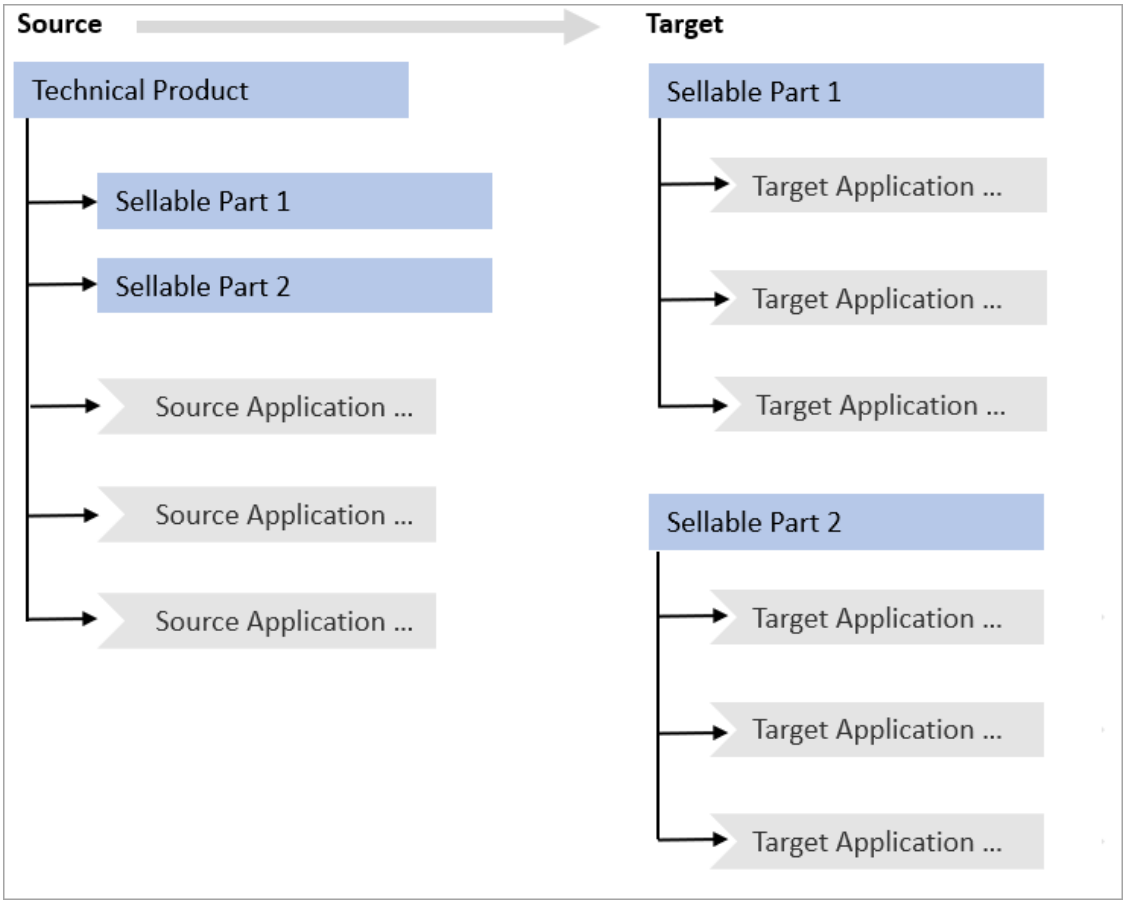

2. Selecting the **Application Mapping** will display a layout split into two tabs called 'Relations' and 'Settings.' While the Relations tab allows users to add new mapper rows, the Settings tab comprises all of the parameters that belong to the Application Mapping plugin. Users should first configure the necessary parameters in the Settings tab before adding any mapper rows in the Relations tab.

   Below is the screenshot of the Settings tab for the Application Mapping plugin in the Onboarding Mappings Details screen. The steps required to configure the Application 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.

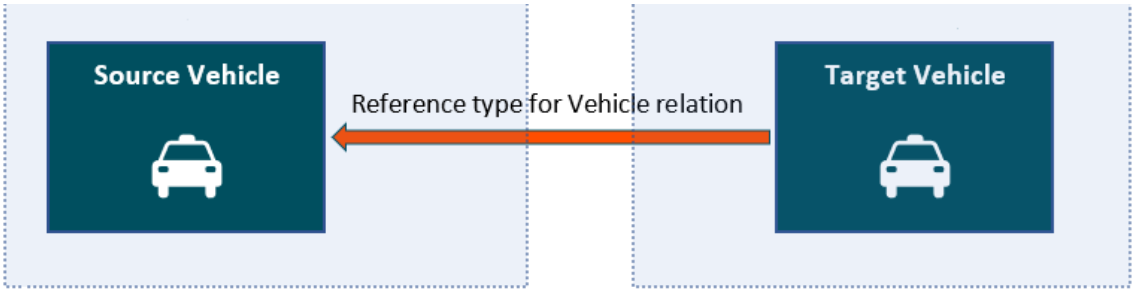



3. Type in a suitable name next to the Name field. This could be any unique name that describes the mapping functionality.
4. The Business Condition parameter allows users to select a business condition. The selected business condition runs the mapper plugin only on the object if the condition is true.
5. Populate the parameters available under the Common Settings field as explained below:
   - **Applications from parent:** This parameter is to be enabled only if the source part and the application is on the same level in the hierarchy. Meaning, source application records reside directly below the parent of the source part. However, the execution of such a Mapper Configuration will create the application records below the target part object itself.

     Below is an example of such a use case where the sellable parts which inherit all relevant technical product information are modeled to reside as children of the 'Technical Product' ('Technical Product' is an object where all technicalities of a product are maintained). The 'Technical Product' object is applicated, and all application records created therefore become the siblings of the sellable parts. When the application records from the source are onboarded with the 'Applications from parent' parameter selected, the application records are created below the target sellable part (as shown below).



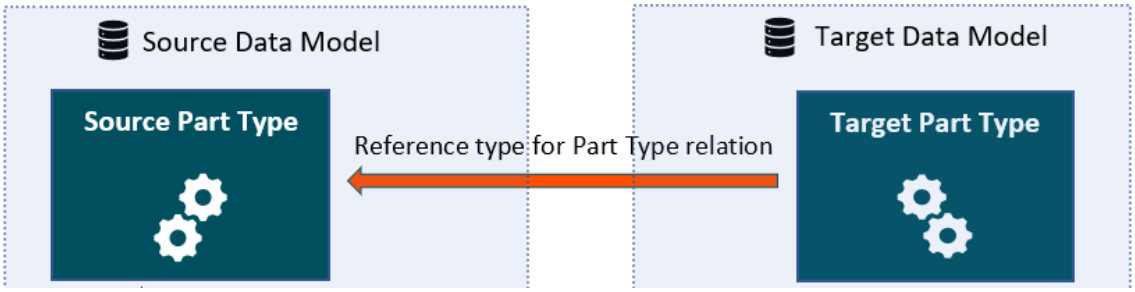

- **Reverse Vehicle Links:** This option is selected if the reference configured within the 'Reference type for Vehicle relation' field is in the direction from the target to source vehicle (as shown below) and the user wants to change the direction of the reference. This is the same as selecting the target vehicle and using the [Referenced By] option.




Consider an example mapping setup where the application records belonging to the source object 'PIES Item' are to be onboarded to the target object 'NAPA Product.' Usually, the reference type that establishes the vehicle relationship will be directed from source vehicle 'Base Vehicle' to target vehicle 'NAPA Year.'

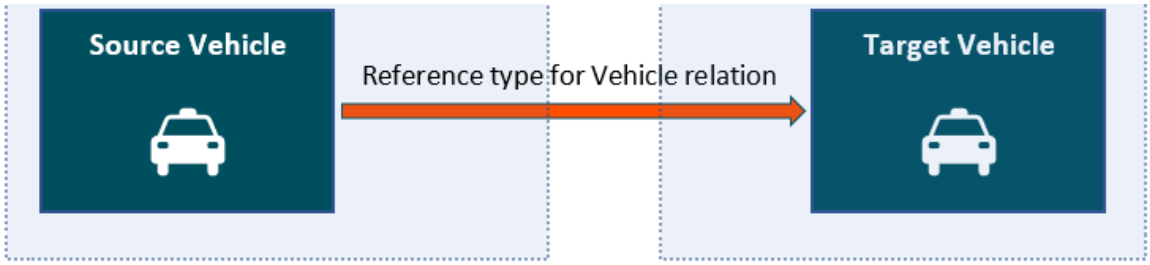
However, if the reference type is directed from target vehicle 'NAPA Year' to source vehicle 'Base Vehicle,' then instead of creating a new vehicle relation reference type, users can select the Reverse Vehicle Links field to establish the right direction of vehicle relationship.

- **Reverse Part Type Links:**This option is selected if the reference configured within the 'Reference type for Part Type relation' field is in the direction from the target to source part type (as shown below) and the user wants to change the direction of the reference. This is the same as selecting the target part type and using the [Referenced By] option.



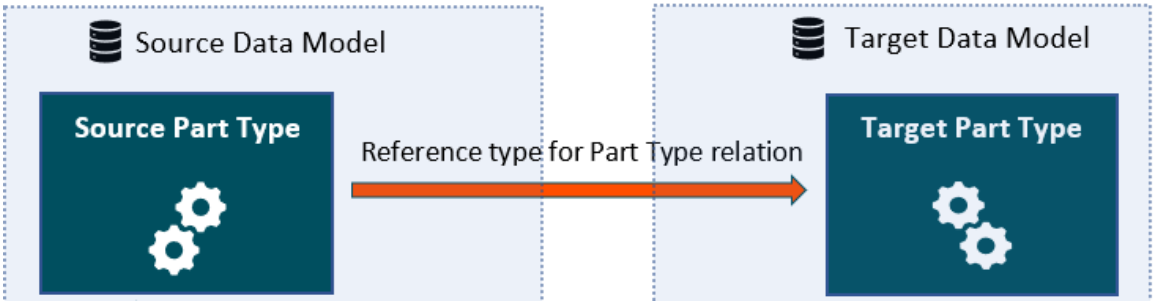

For example, if there is a mapping setup where the application records belonging to the source object 'PIES Item' are to be onboarded to the target object 'NAPA Product.' Usually, the reference type that establishes the part type relationship will be directed from source part type 'Part Terminology' to target part type 'NAPA MPCC.' However, if the reference type is directed from target part type 'NAPA MPCC' to source part type 'Part Terminology,' then instead of creating a new part type relation reference, users can select the Reverse Vehicle Links field to establish the right direction of part type relationship.

6. Populate the parameters available under the Object Setup field as explained below:
   - **Reference type for Vehicle relation:** Defines which reference type to use to look for the vehicle relationship from the source to the target vehicle.




For example, suppose the source object is PIES Item and applications belonging to this source object need to be mapped to target object 'NAPA Product.' In that case, the reference type for vehicle relationship is referenced from source object vehicle = Base Vehicle to target object vehicle = NAPA Year.

   - **Reference type for Part Type relation:** Defines which reference type to use to look for the part type relationship from the source to the target part type.




For example, if the source object is PIES Item and applications belonging to this source object need to be mapped to target object 'NAPA Product,' ,then the reference type for part type relationship is referenced from source object part type 'AutoCare Part Terminology' to target object part type 'NAPA MPCC.'

- **Business action executed on new applications:** This field allows the user to select a Business Action. The selected business action will be executed only on the new application records created while running the Mapper Configuration.

  While creating a business rule to be used in this parameter, it is recommended that the following binds be used as necessary along with the other standard binds as required:

| Bind | Functionality |
|---|---|
| **Mappings Manager** | When the Mapper Configuration is being executed, it is quite likely the actual new child node is not created in the target yet. This means the business rule *cannot* use the Current Object bind. Instead, the business rule must use the Mappings Manager bind that has only one method<br>addValue(DataType datatype, String Value).<br>This allows the business rule to create new references and set attribute values on the child object. It is not possible to create new objects through the Mappings Manager bind. |
| **Source Object** | Used to retrieve data specific points from the source object |
| **Source Children** | The set of children (objects) that are merged into the new target child. The set always contains at least one object. |

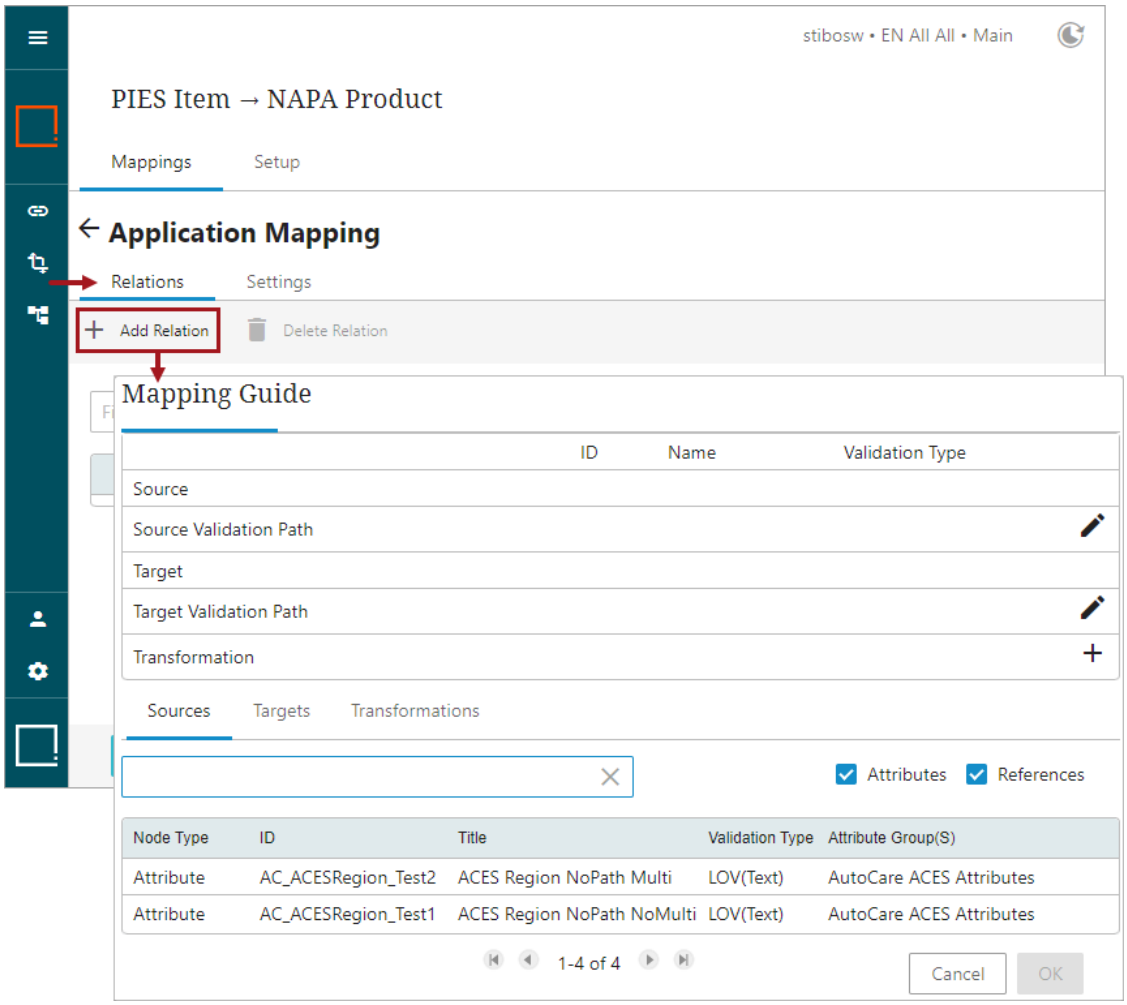
- **Ignore Application Attribute:** This field allows users to select an attribute that is valid for the source application record. The selected attribute determines if the application record needs to be onboarded when the Mapper Configuration setup entity is executed on the source object. If the selected attribute belonging to the source application record holds the value 'true', then such application records will be skipped from being onboarded.

For example, if the user wants to skip the application records from being onboarded that are marked for deletion, then the attribute used for flagging the deletion of the application record is configured in this parameter.

Setup entity definitions can be exported as comments and submitted to an external source control system for comparison purposes. For details, refer to the Configuration Management documentation.

7. Once all the necessary parameters in the Settings tab are populated, the users must define a relation for retrieving information from the source and copy it to the target. Users can navigate to the Relations tab and add new mapper rows to the Application Mapping plugin to define a relation.

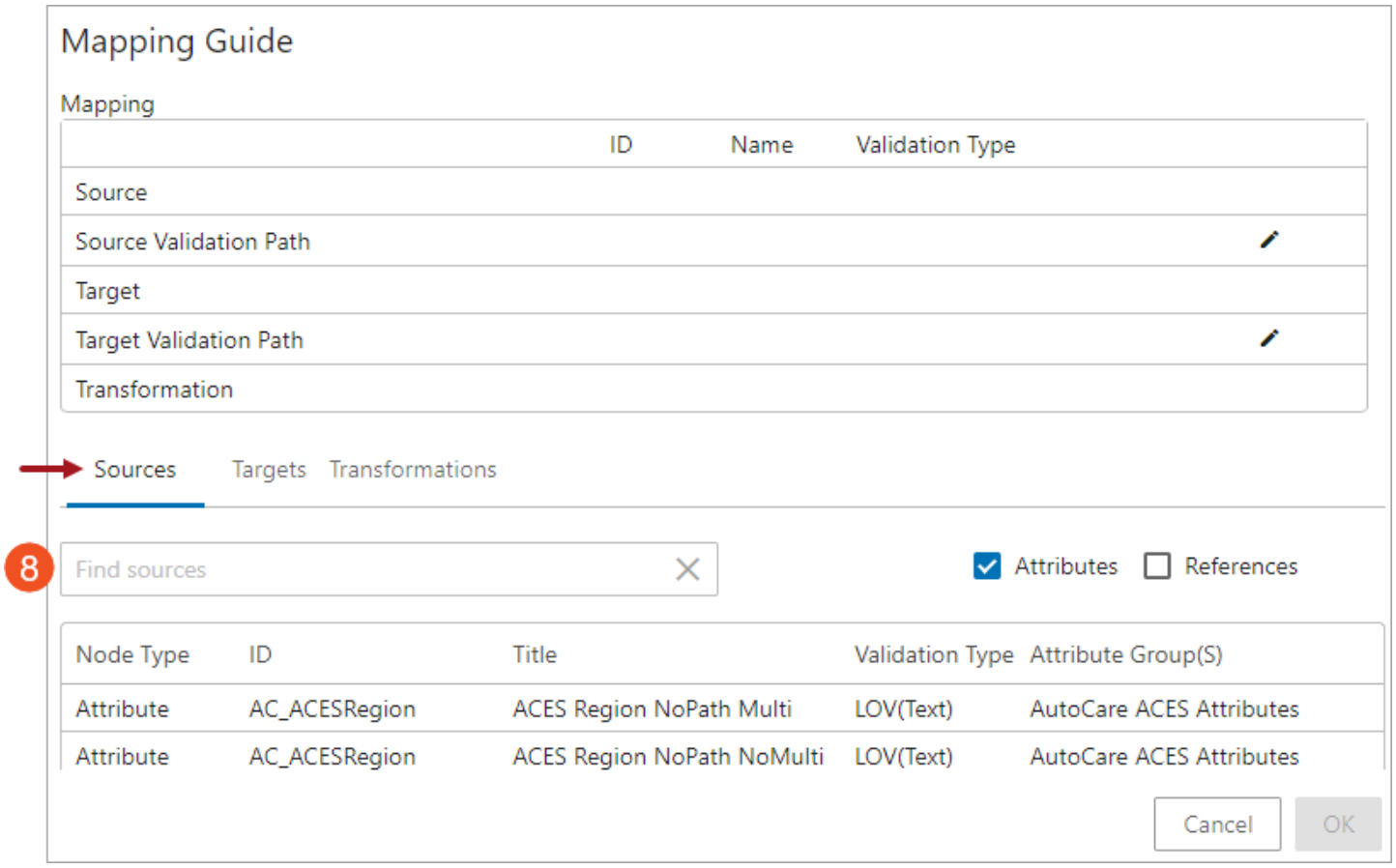
   Click the '**Add Relation**' icon, and the Mapping Guide window will display (as shown below).



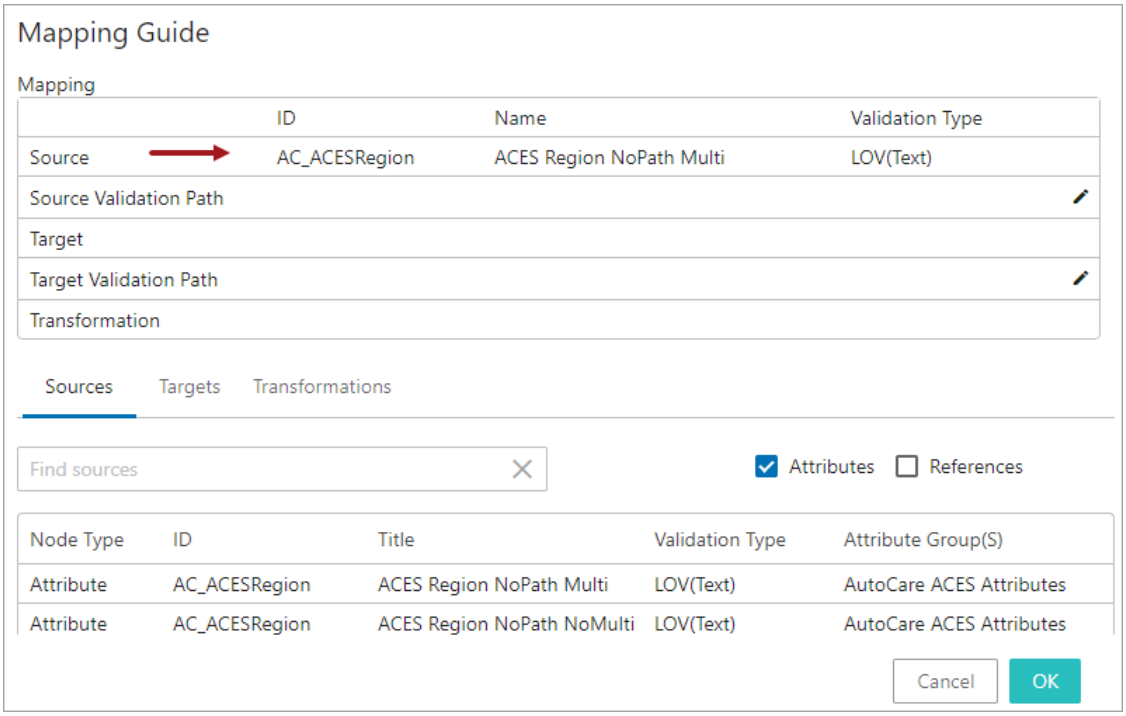

   Adding a relation can be as simple as copying an attribute value from the source application over to the target application. Or for any condition(s) that can be retrieved on the source object by following references or attributes can also be added as a relation to the target object in order to map the applications (only if the valid relations are covered).

8. To retrieve an attribute value of the source application record, select the **Sources** tab, and with the **Attributes** checkbox selected, start typing the initial letters of the attribute name or ID in the search field. This

filters down and displays attributes that are valid for the source object's application records. Select the required attribute from the result list displayed below the search bar.

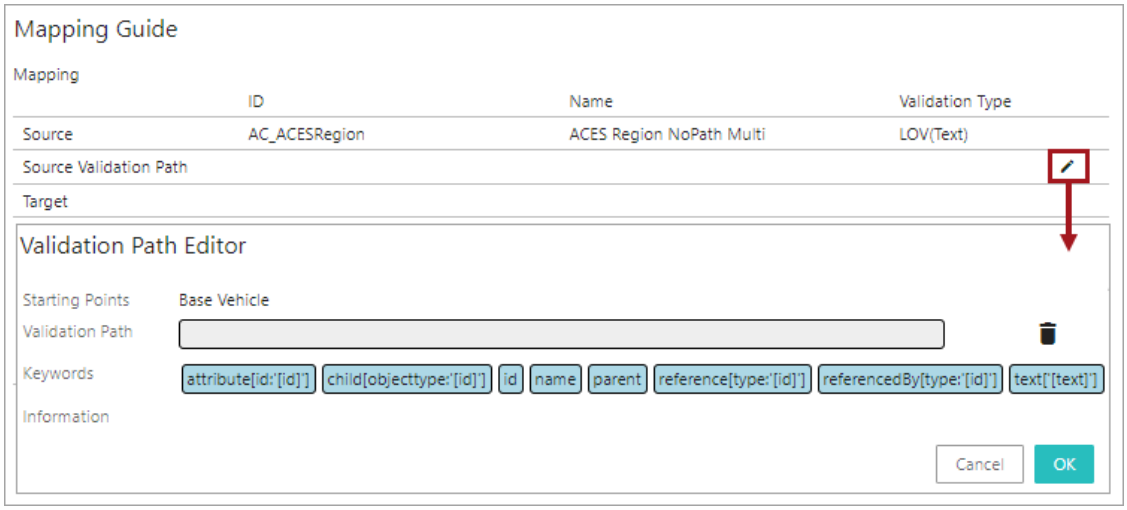



Once selected, the attribute will be populated in the Source field (as shown below)

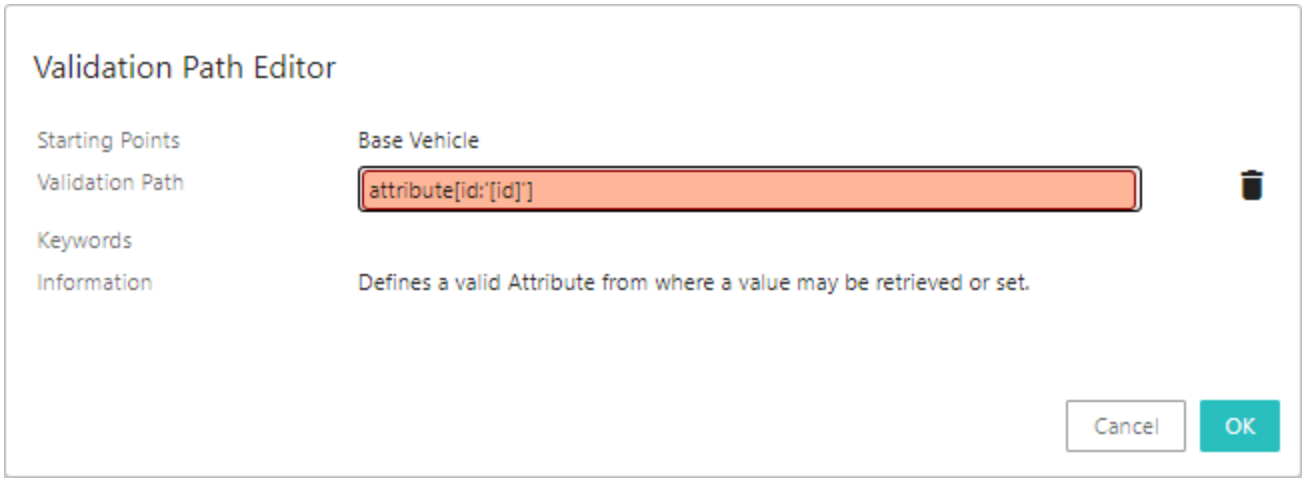



9. Optionally, if any additional information (ID, Name, attribute values, references, or links) from the source object's application record needs to be retrieved, click on the Edit icon ( ) in the **Source Validation Path** field, and the Validation Path Editor window will display (as shown below). If the user does not want to retrieve additional information, skip populating the Source Validation Path field and proceed with step 11 within this topic.

   Within the Validation Path Editor window, the user can select specific data points from the source model 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, refer to the **Mapping Validation Path Functionality** topic within this guide.

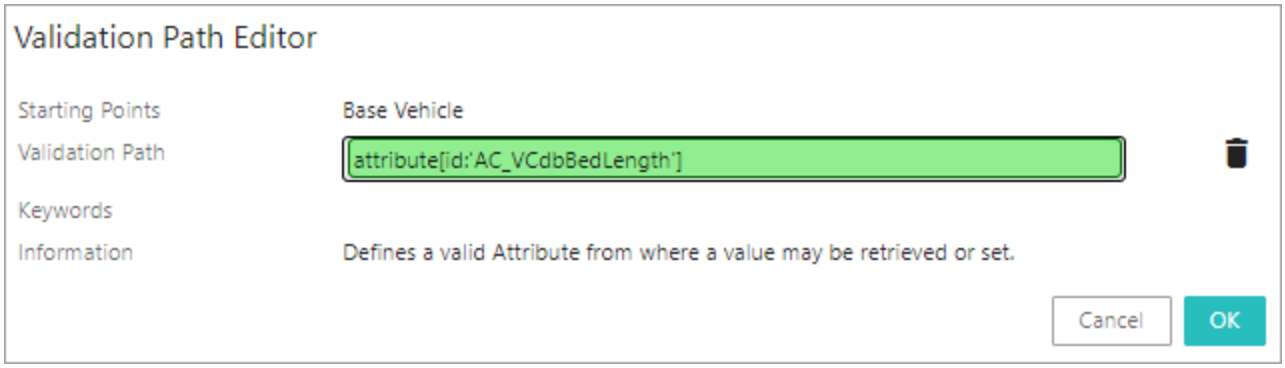


10. Double click on the required element(s) from the Keywords field. The selected element will be populated in the Validation Path field (as shown below). The specific combinations of elements are chosen depending on the unique requirements for the data being retrieved. Selecting the elements will define the source STEP Path, which describes the path from which the source data must be retrieved. Depending on the element that is selected, further configurations may be required. In the example below, element **attribute[id:'[id]']** is selected and displayed in the Validation Path field. The red colored field indicates that additional configuration is required.

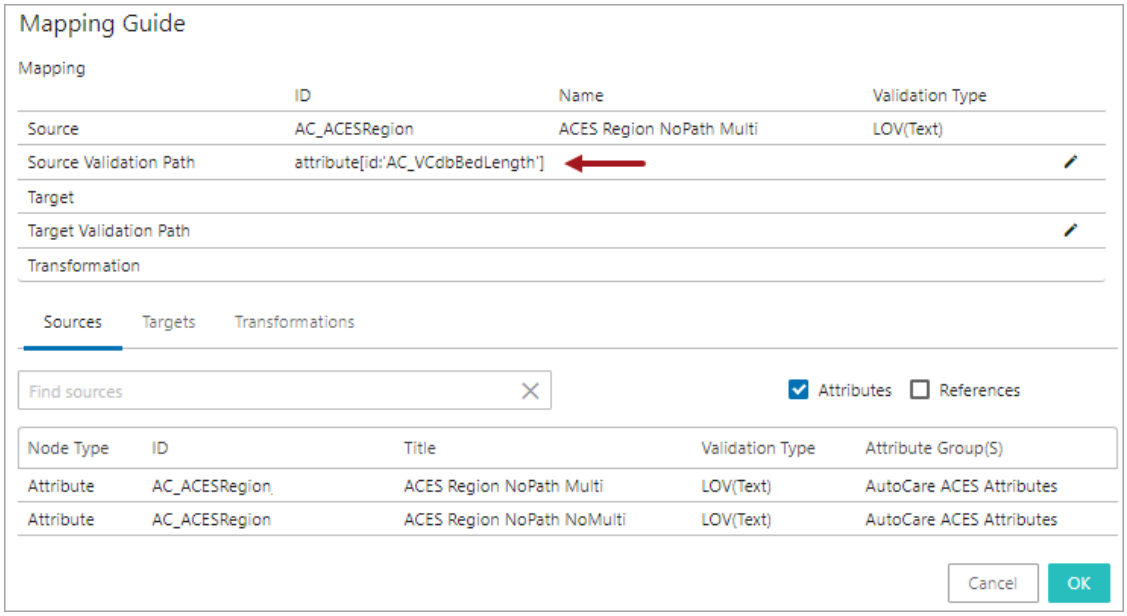



Users can add one element or a combination of multiple elements to create a source STEP Path. For more information on creating the source STEP Path, refer to the **Mapping Validation Path Functionality** topic within this guide.

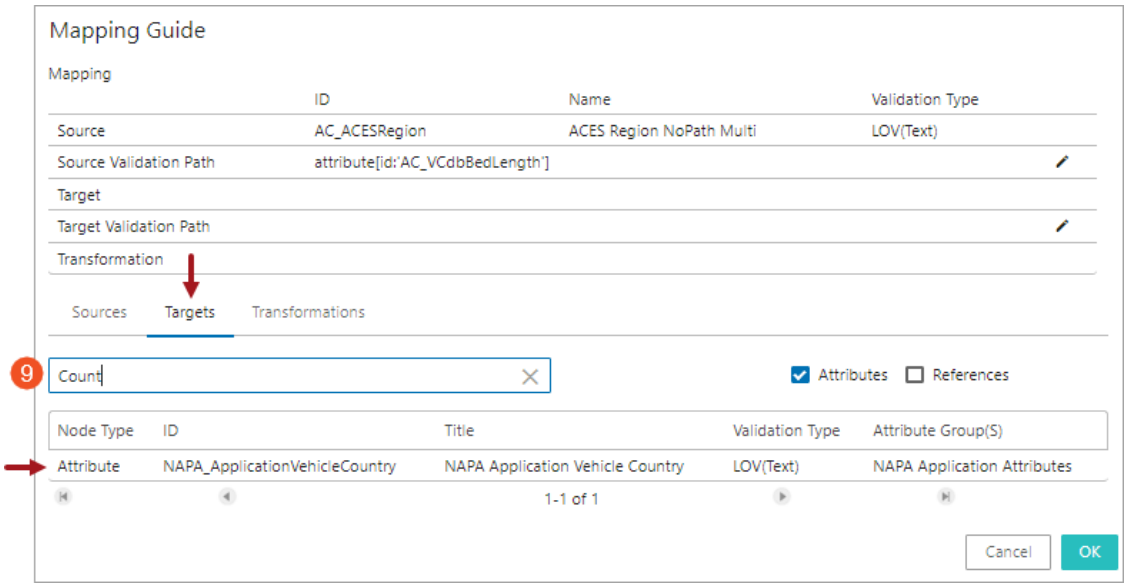
In this example, the selected element '**attribute[id:'[id]']**' is configured to retrieve the value from attribute with ID 'AC_VCdbBedLength'.



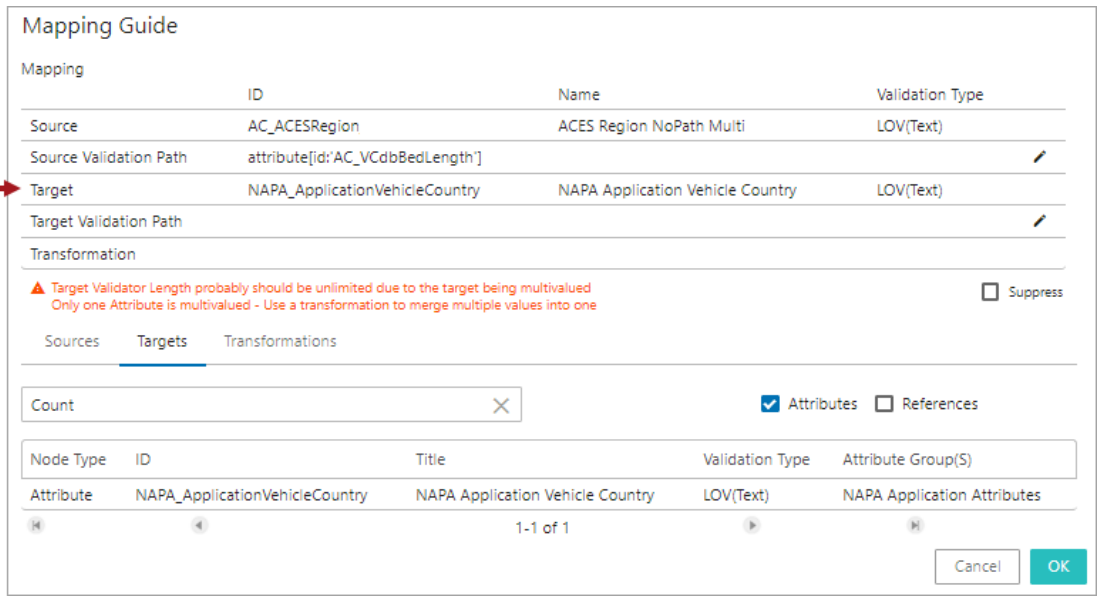

Once selected, the source STEP Path with the defined data will be populated in the source Validation Path field.




11. 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.

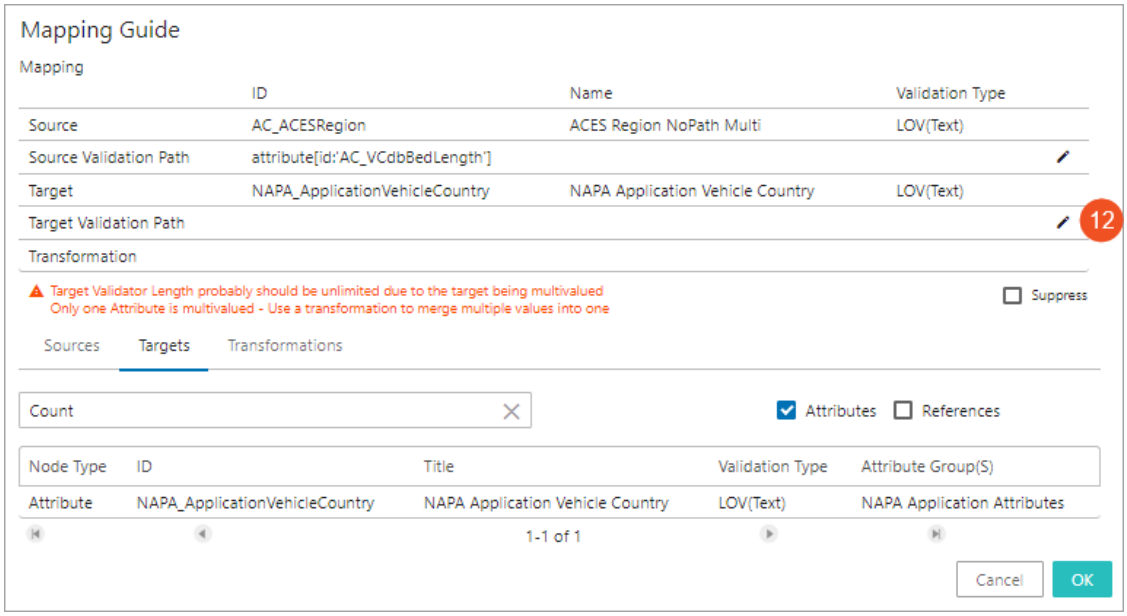



Once selected, the selected attribute / reference will be populated in the Target field (as shown below).

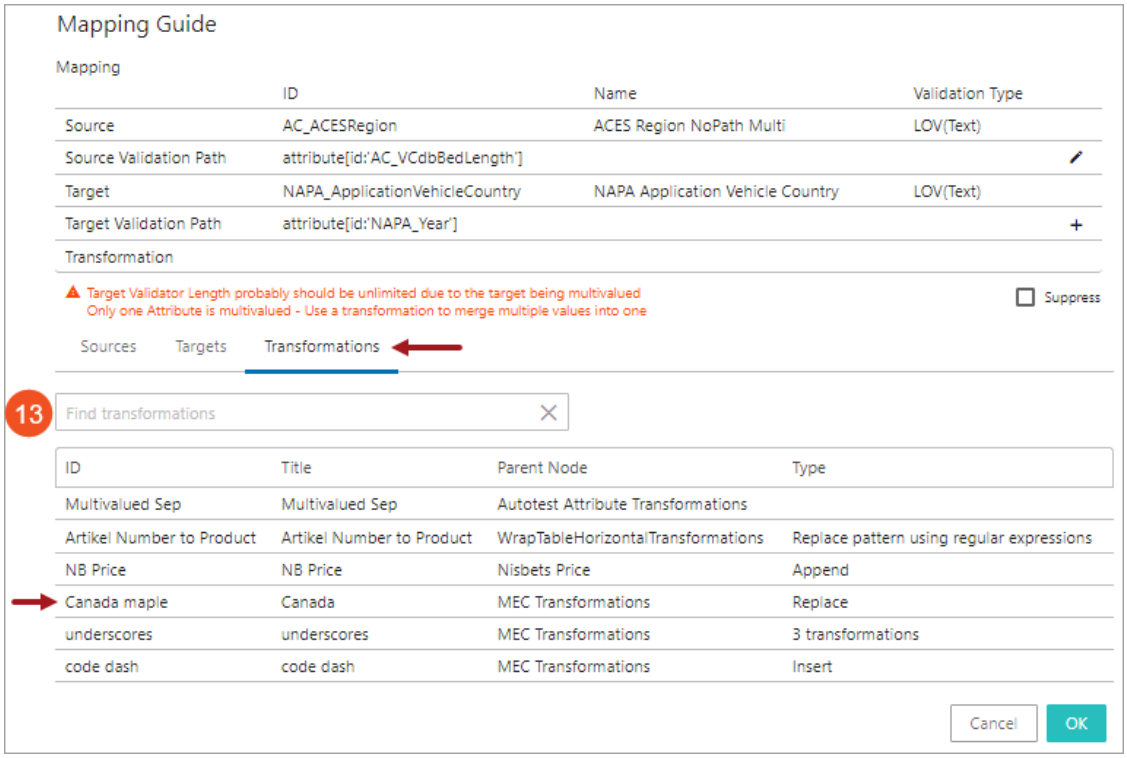



12. Optionally, if the retrieved information from the source object needs to be populated in the specific data points of the target object, click on the Edit icon ( ) in the Target Validation Path field, and the Validation Path Editor window will display(as shown below). If the user does not want to retrieve additional information, skip populating the Target Validation Path field and proceed with step 13 of this topic.

    Within the Validation Path Editor window, the user can select specific data points from the target model 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 target STEP Path, refer to the **Mapping Validation Path Functionality** topic within this guide.

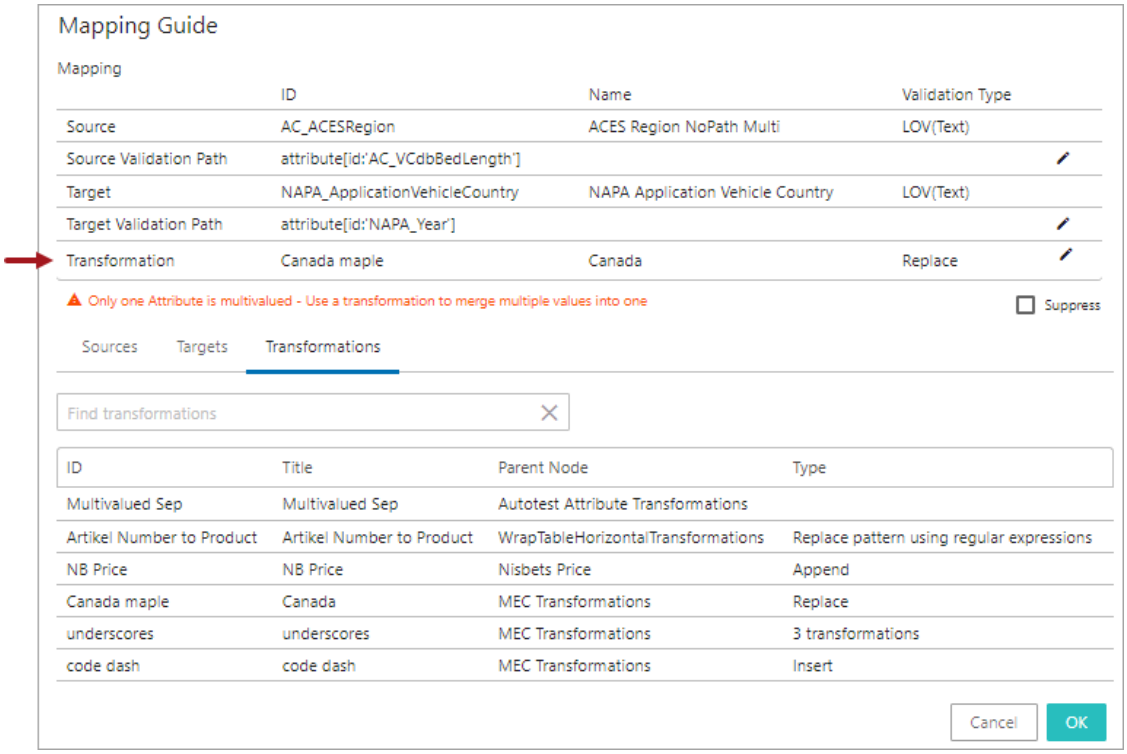


13. If transformations are needed, users can either use an existing attribute transformation in the system or create a new one by clicking the 'Create new transformation' icon ( + ) available within the Transformation field. To configure an existing attribute transformation, users should select an existing attribute transformation listed in the 'Transformations' tab displayed in the lower half of the window (as shown below).

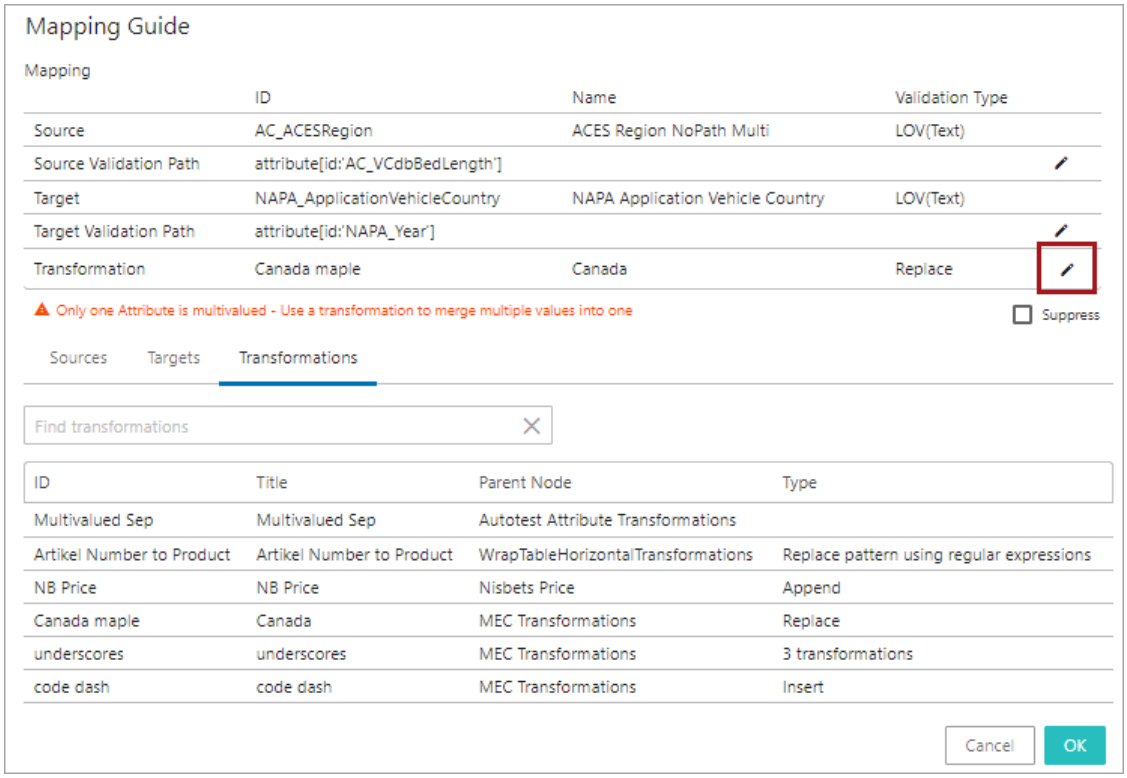



Clicking on the 'Create new transformation' icon ( + ) available within the Transformation field will open the Transformation Overview window where the user can create a new attribute transformation. For details about creating an attribute transformation standing on a Mapping Guide window, refer to **Creating Attribute Transformations Through Mapping Guide Window** topic within this guide.

The selected transformation will be populated in the Transformation field (as shown below).

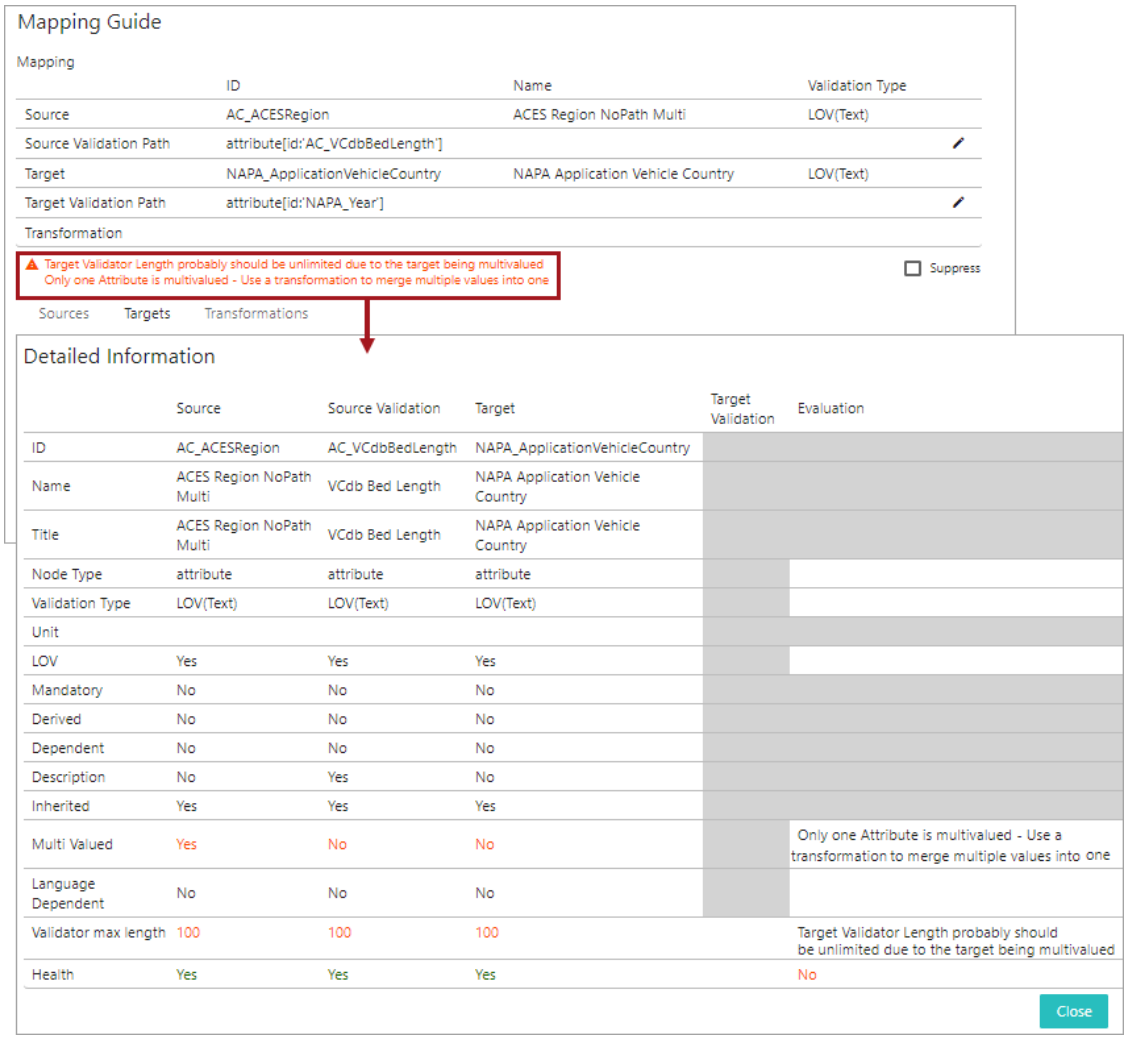



The populated transformation can be edited / removed by clicking the edit icon ( ) in the Transformation field.



**Note:** Deleting an attribute transformation from the Mapping Guide window means removing it from the current mapping setup; however, editing it impacts all of the other mappings that have used the particular attribute transformation. Users are advised to be cautious before editing the attribute transformation. The Mapping Guide in the Web UI does not analyze if the transformation is applied to any other mapping.

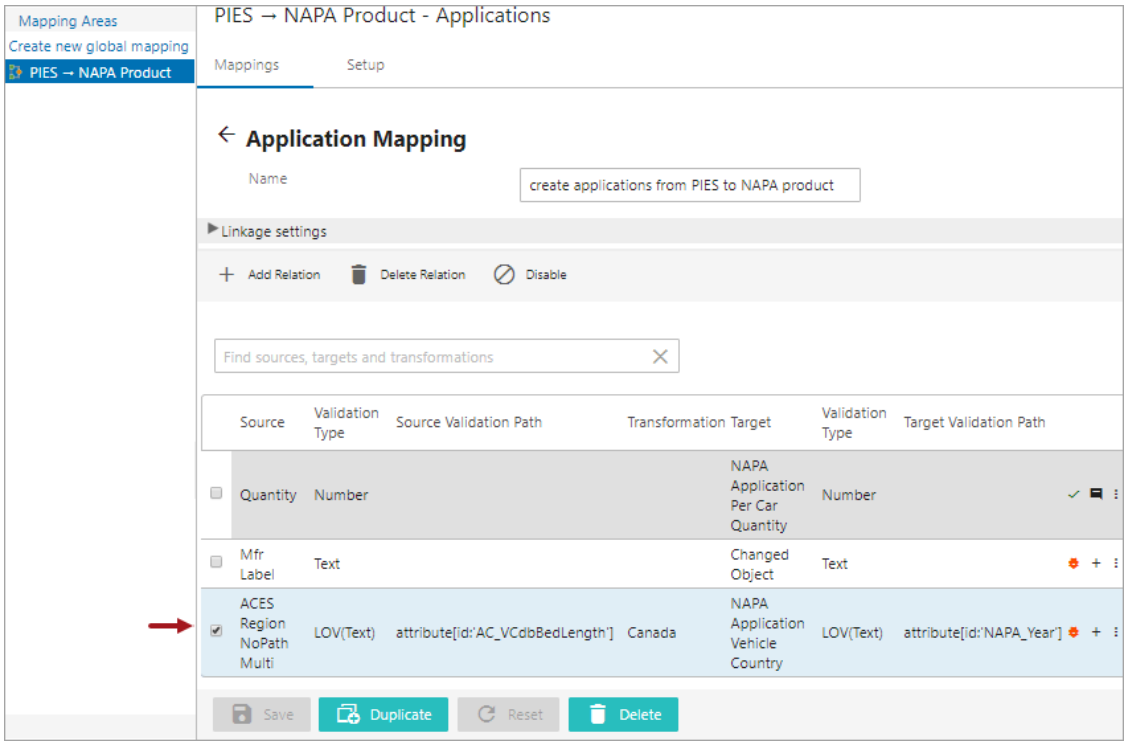
After the source and target data points are defined, the system evaluates the validity match between the source and the target data points. A hyperlink text explaining the reason for the validity match / mismatch will be displayed. Clicking on the hyperlink opens the '**Detailed Information**' dialog which displays the validity match / mismatch information of the Application mapping configuration. Below is an example of an invalid match between the source and the target attribute that is displayed in the 'Detailed Information' dialog.




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.

14. 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 application mapping row will be displayed. To edit the application

mapping, click on the row to open the **Mapping Guide** window in which edits can be made.




15. Repeat the above steps 7 to 14 to add more mapper rows for the same plugin.

**Important:** Users must save the changes before exiting the Mapper Configuration. If the user fails to click **Save**, then the mapping will be lost once the user leaves the Mapper Configuration.

Users can add any number of mapper rows (relations) 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 towards the right of each 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. For more information on handling the mapper rows, refer to topic **Modifying Mapper Rows on the Onboarding Mapping Details Screen** within this guide.

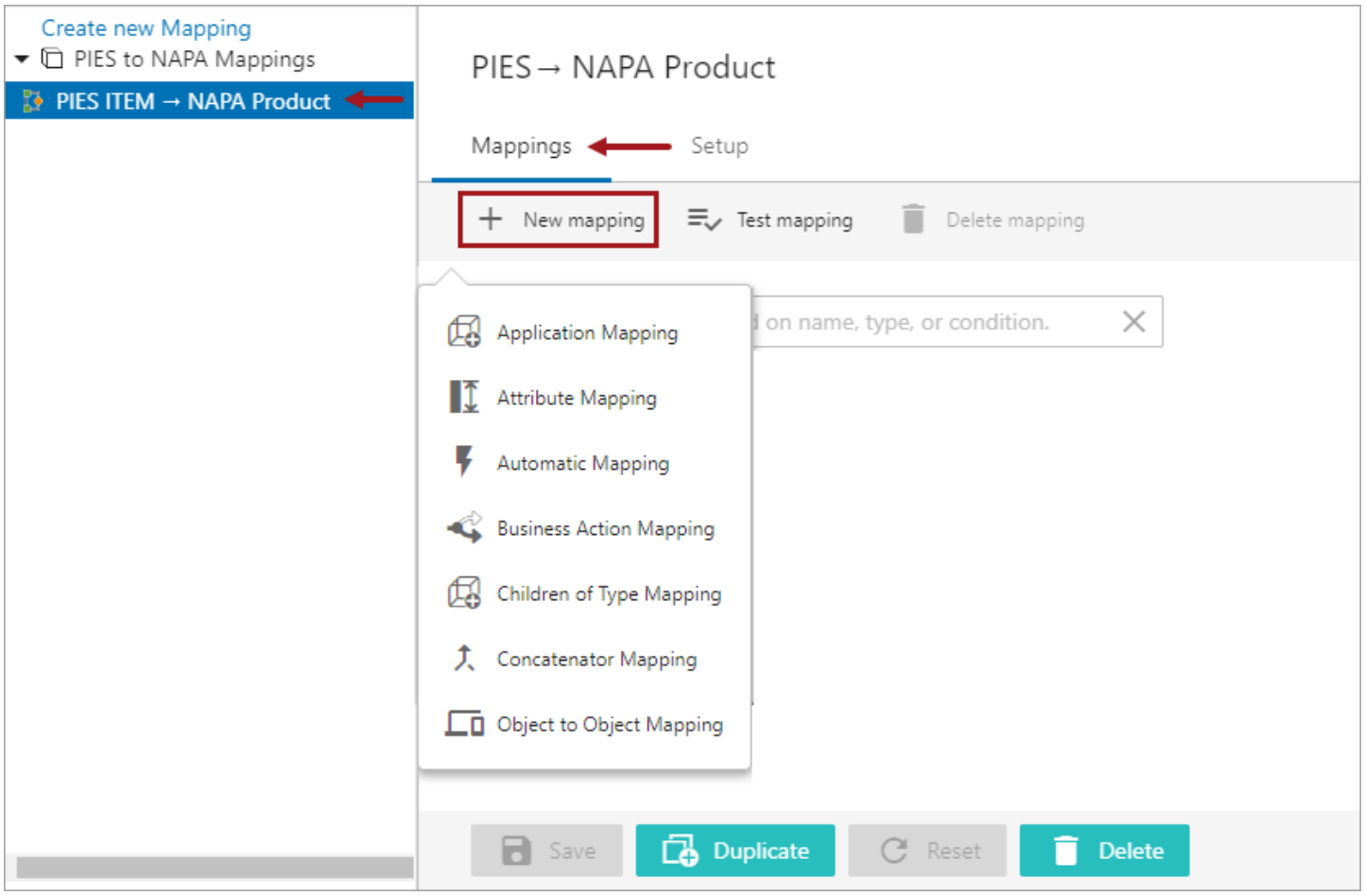
## Attribute Mapping Plugin

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.

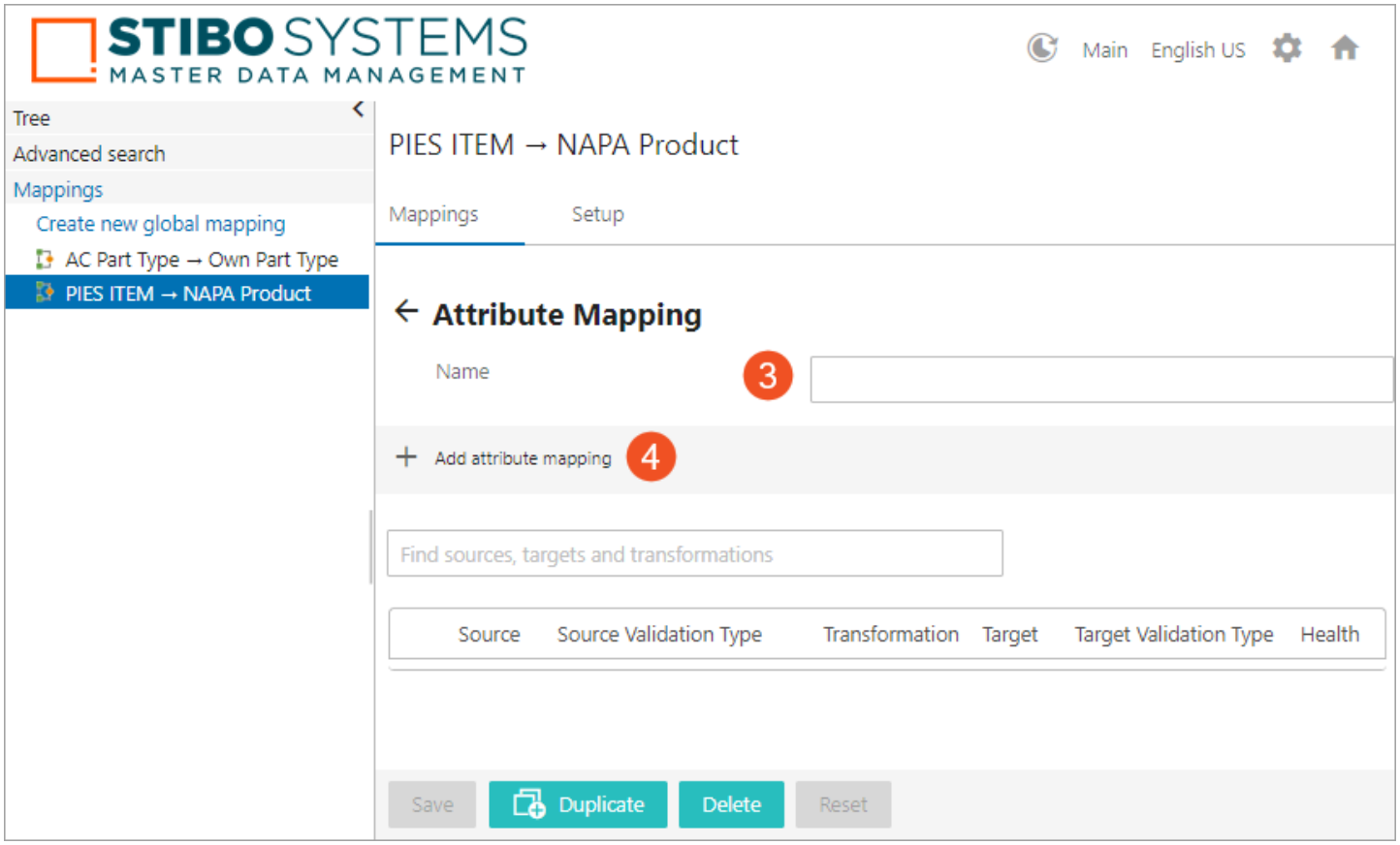
**Note:** All settings are not necessary for all attributes. The specific combinations chosen are dependent on the unique requirements for the attribute being configured. The settings contained within this plugin can include transformations. The plugin can be configured to either Onboard / Offboard the data or in some cases can be configured to work along with transformations too.

To configure the Attribute Mapping plugin, follow these steps.

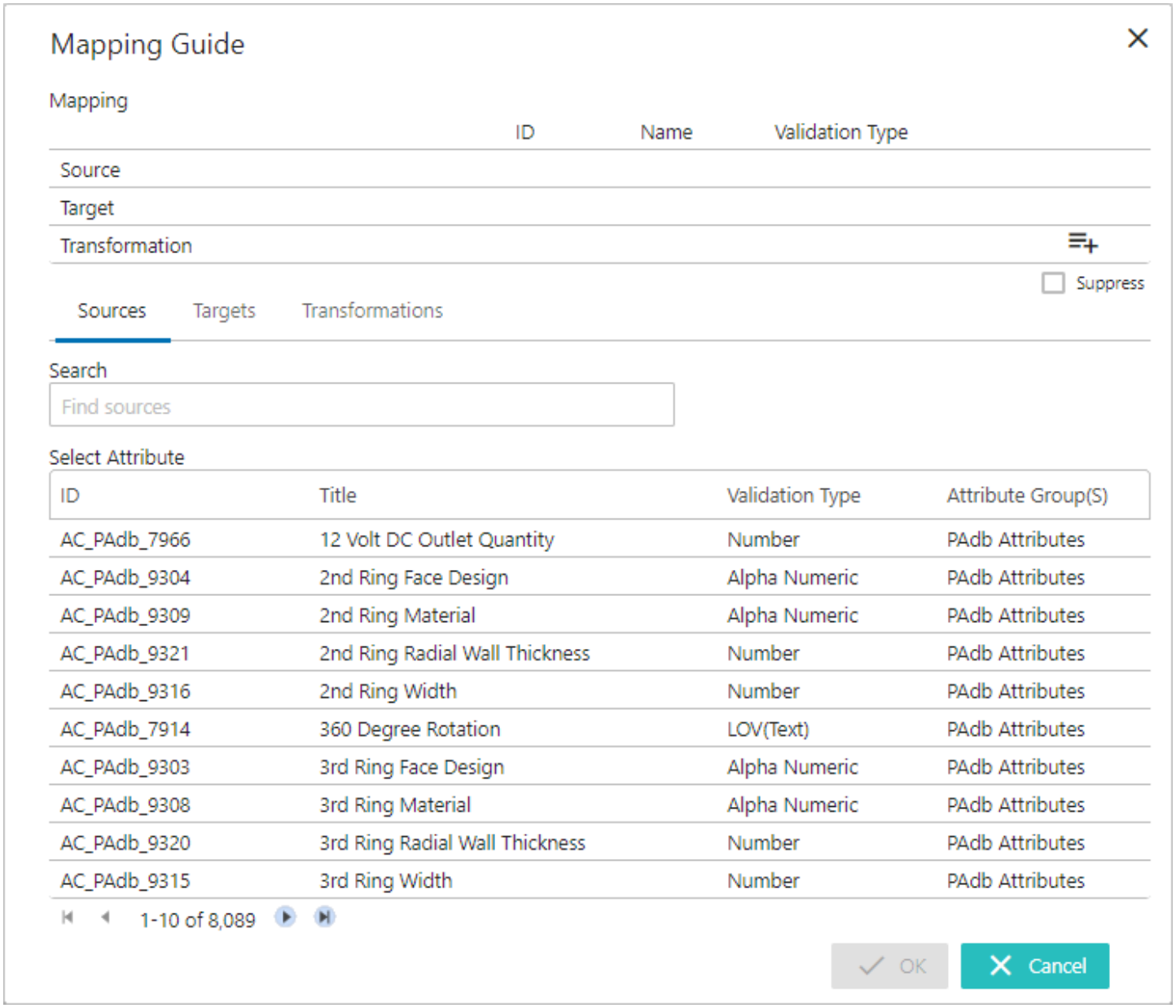
1. With the required Mapper Configuration selected, click on **New mapping** available within the Mappings tab of the Onboarding Mappings Details screen.



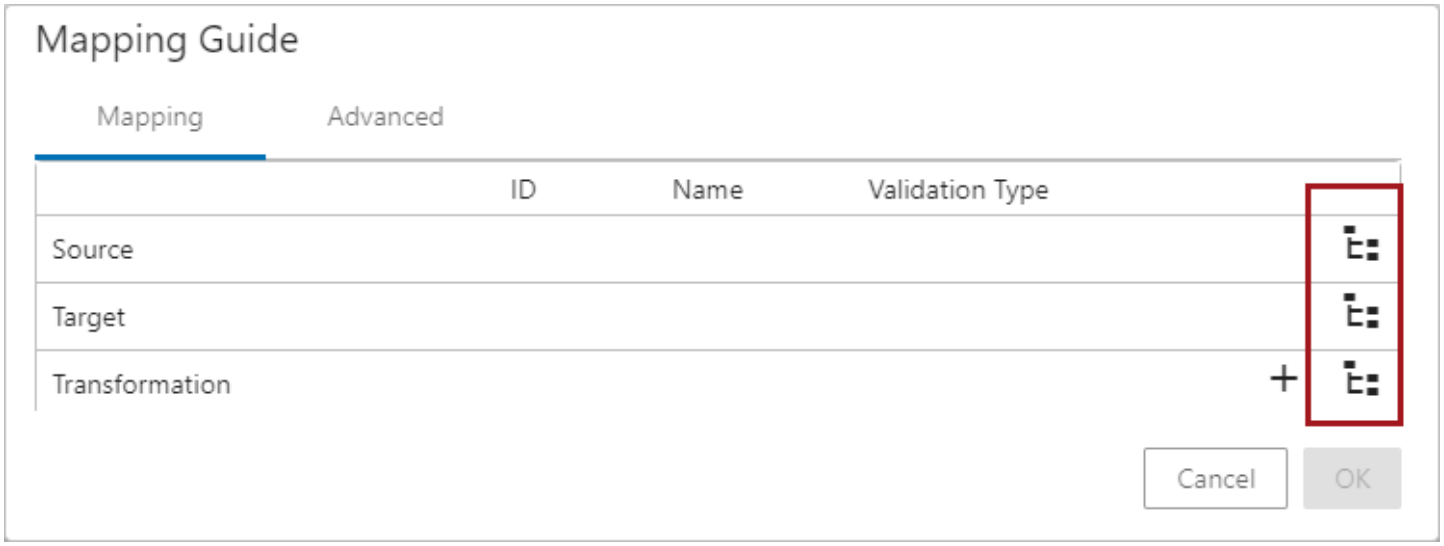

2. Select Attribute Mapping, and a screen is displayed prompting the user to further configure the plugin (as shown below).



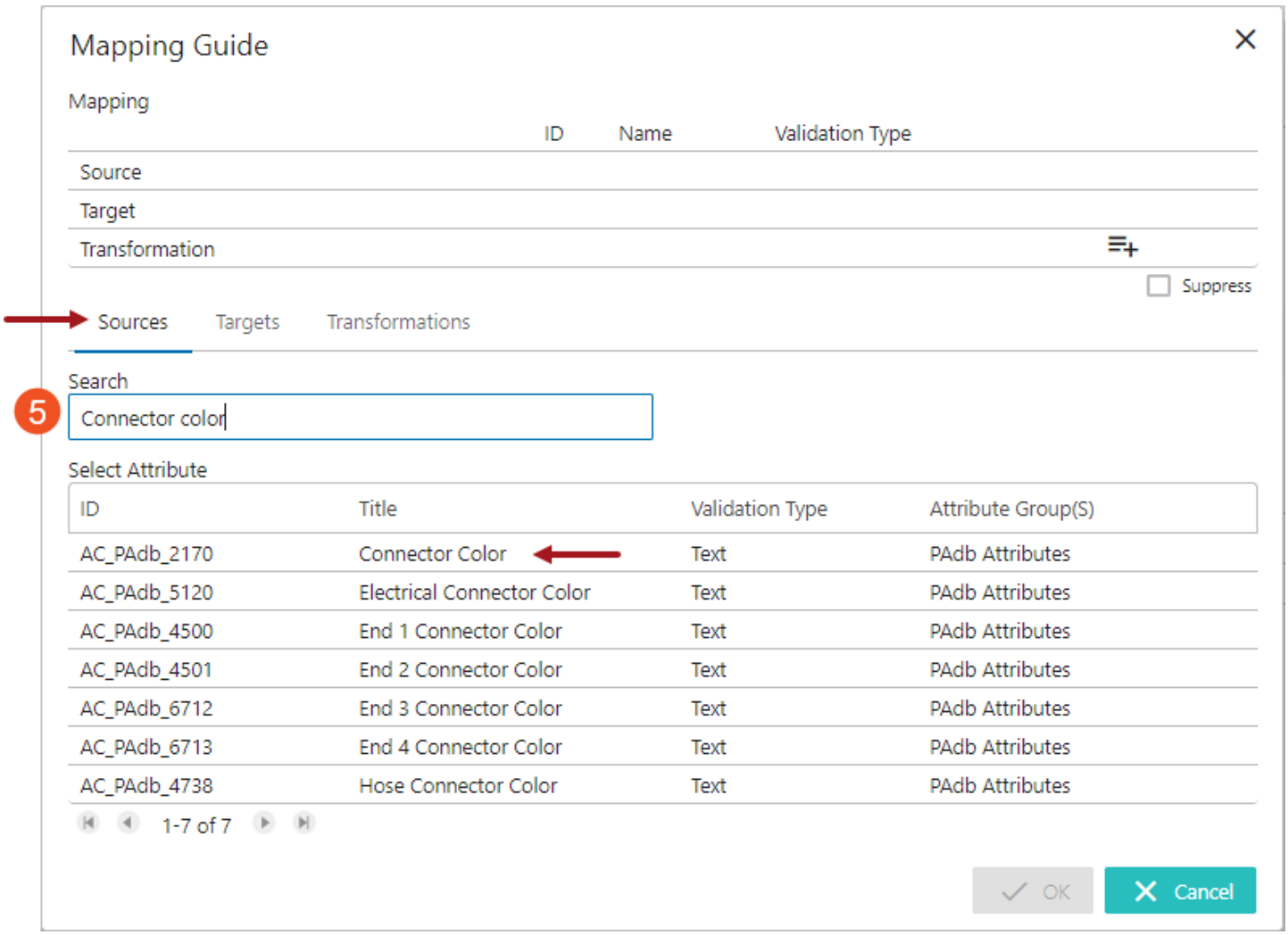

3. Type in a suitable name next to Name field. This could be any unique name that clearly describes the mapping functionality.
4. Click the 'Add attribute mapping' icon, and the Mapping Guide window will display (as shown below).



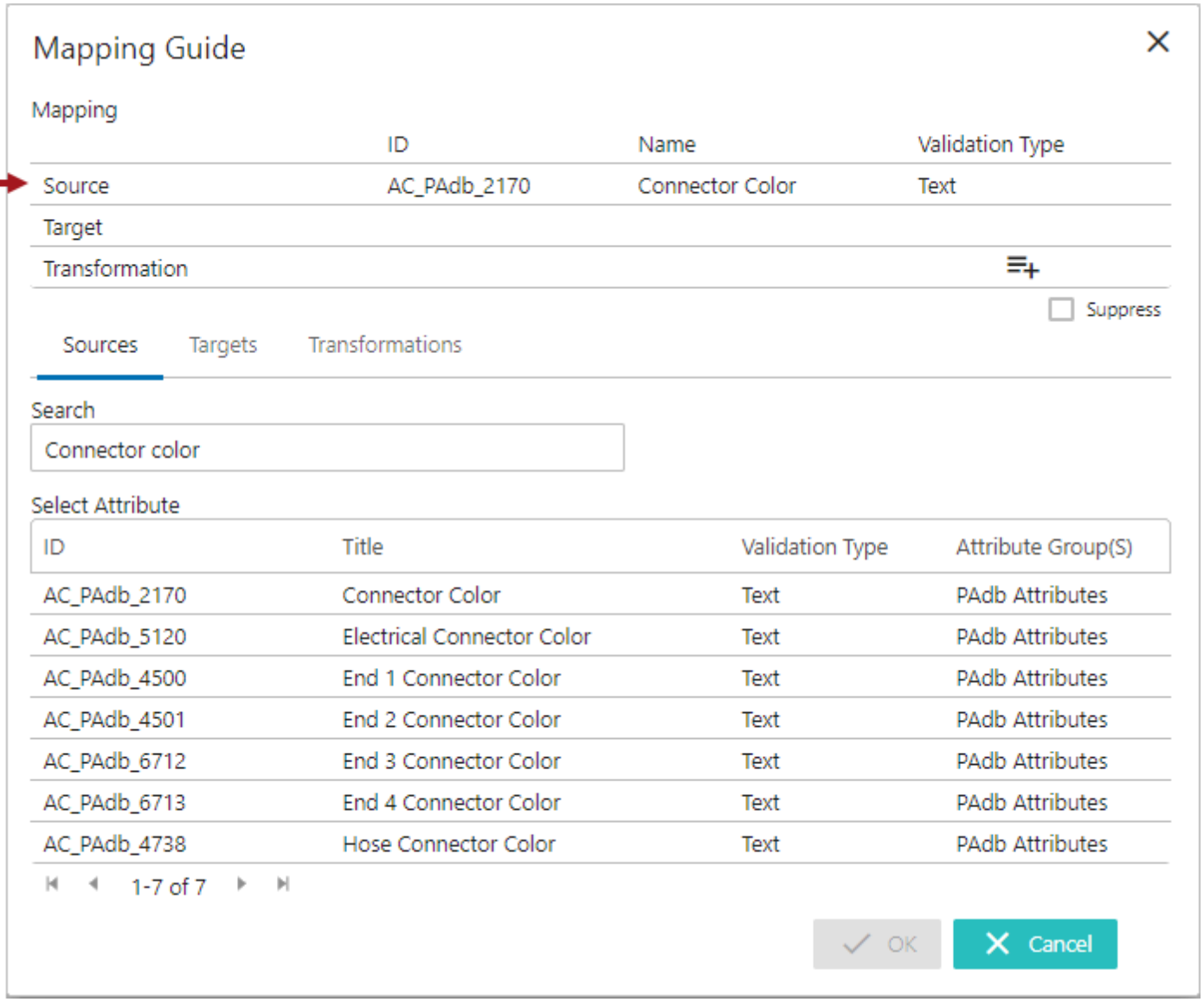

**Note:** It is possible in the Mapping Guide window to display the regular Node Pickers (as shown below) instead of the typeahead field. In order to use the Node Pickers in the Mapping Guide window, users need to set the following property in the sharedconfig.properties file:
`Mapper.MappingGuideUseStandardComponents=true`.



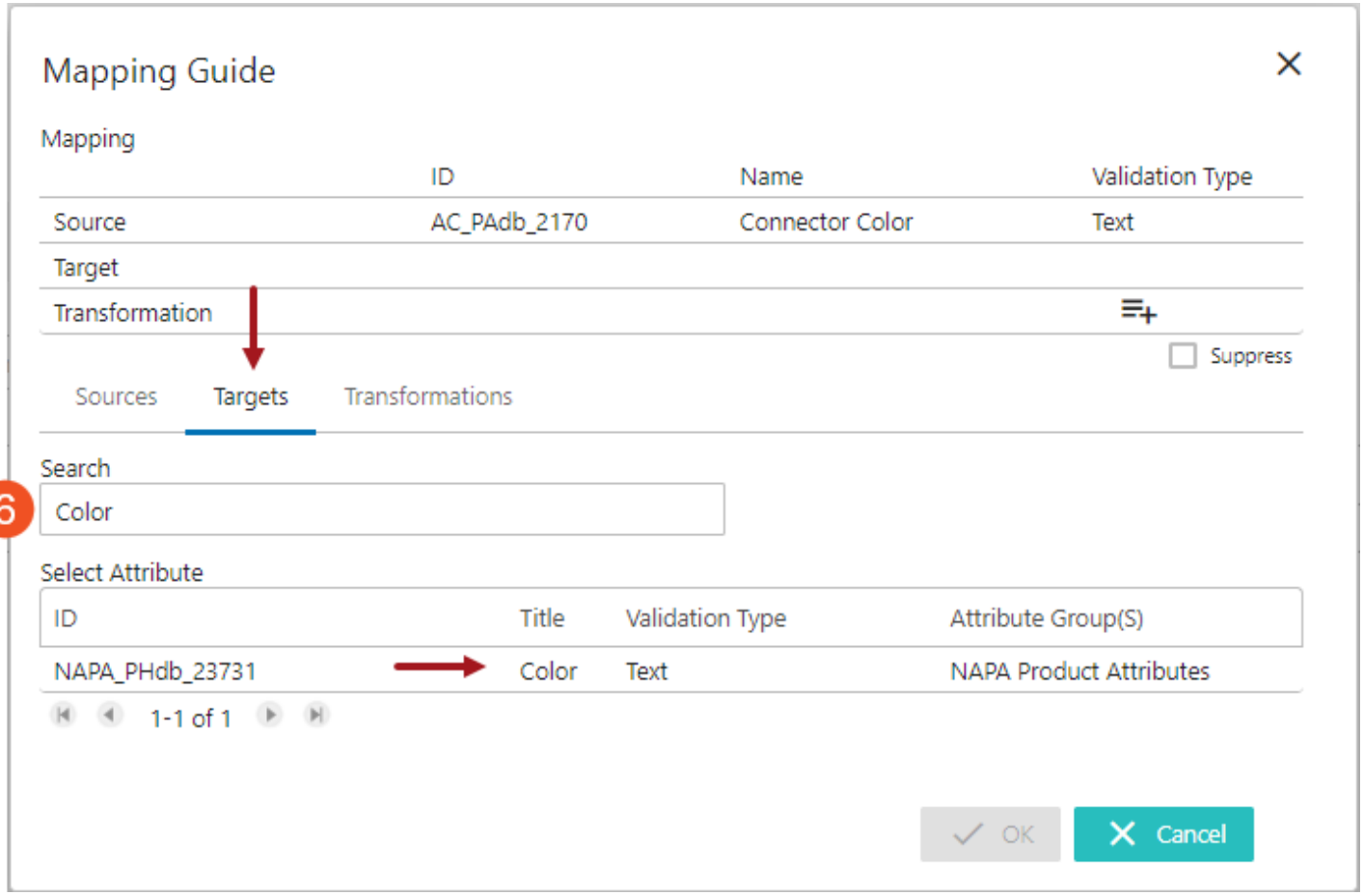

5. With the Sources tab selected, click in the Search field and start typing the initial letters of the attribute name or ID. This brings up a dropdown of typeahead search results listing the attributes of the Source object available in the system from where the attribute value could be retrieved. Select the attribute from the result list displayed below the search bar.



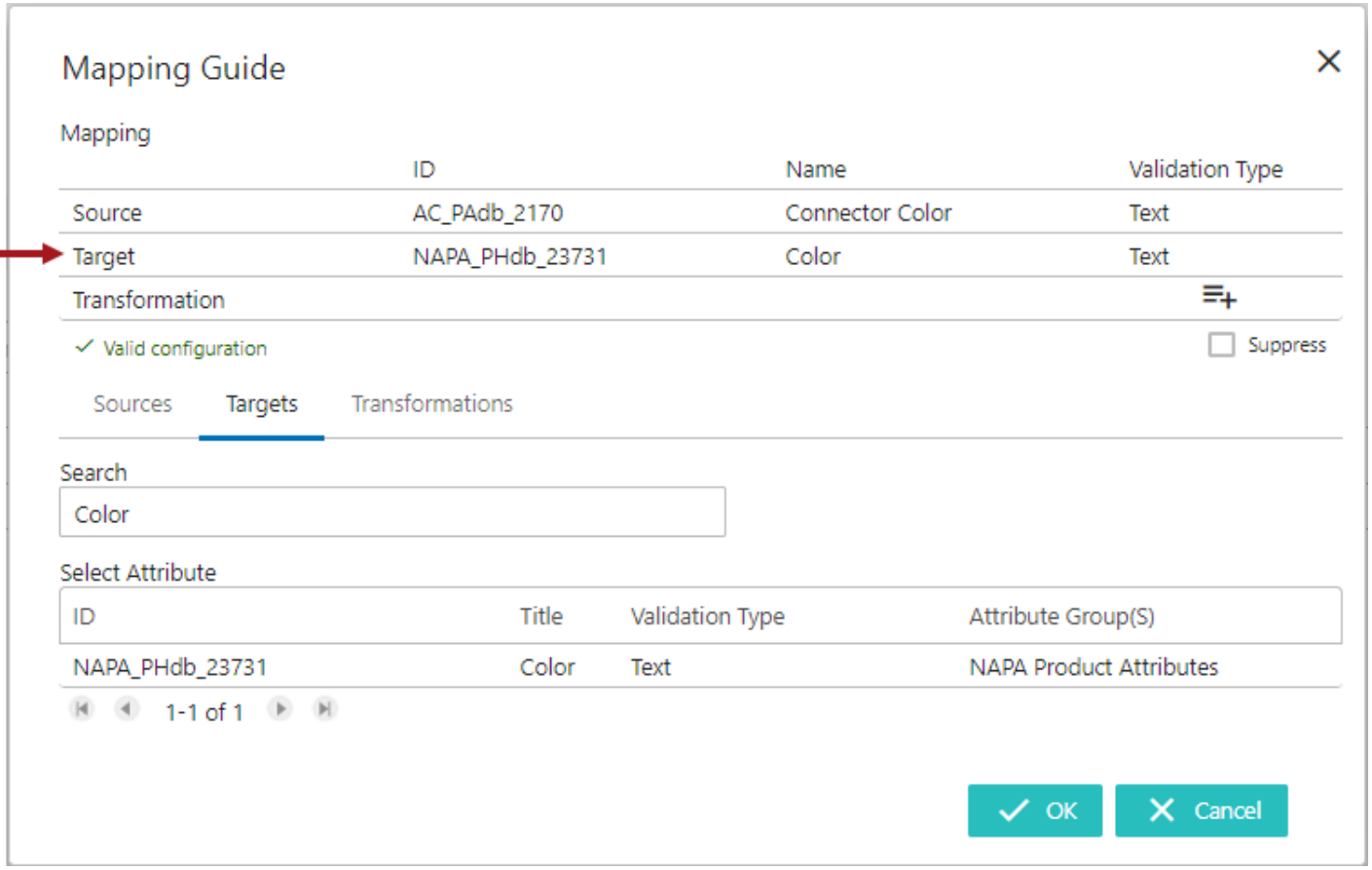

Once selected, the attribute will be populated in the Source field (as shown below)



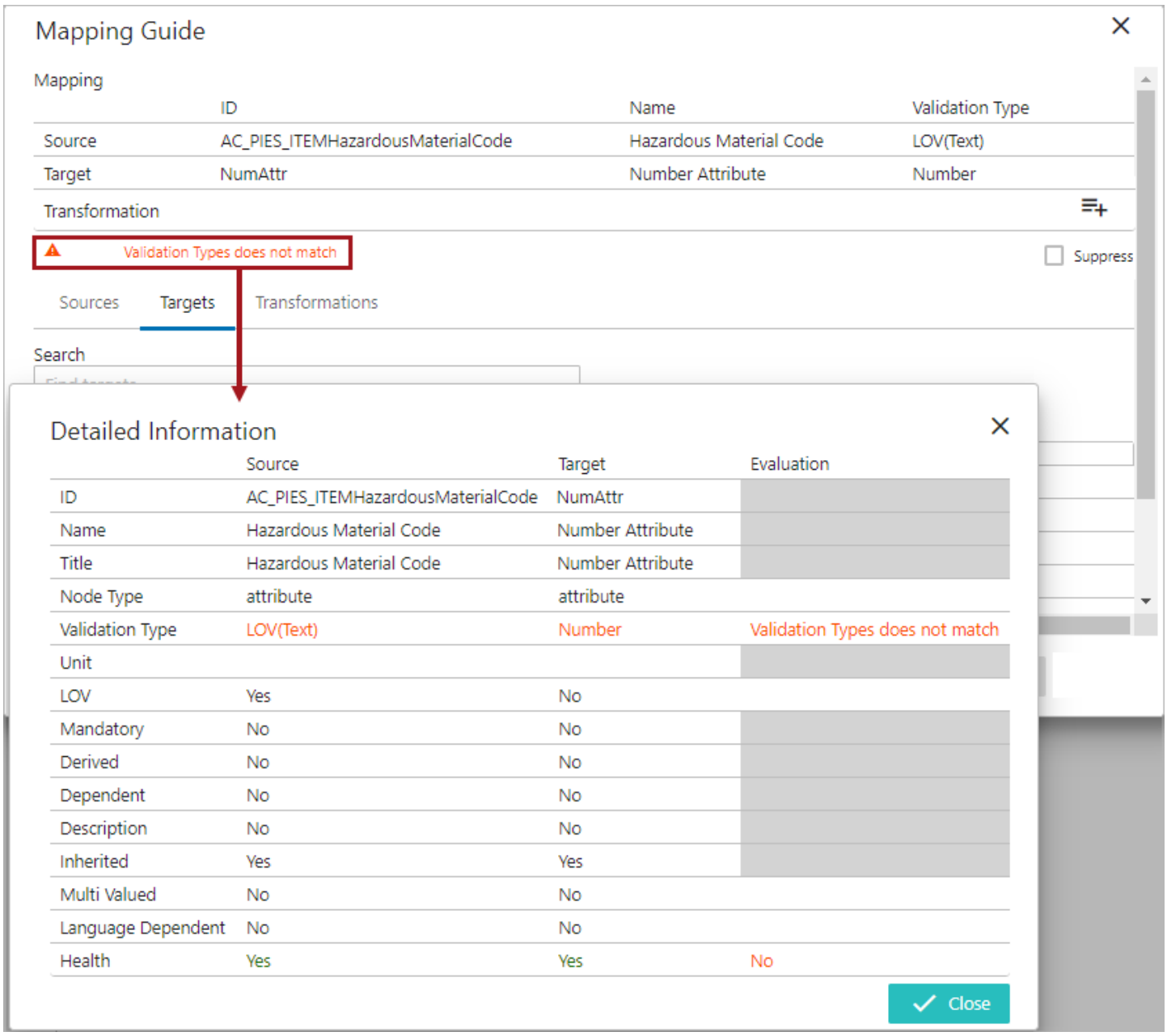

6. With the Targets tab selected, click in the Search field and start typing the initial letters of the attribute name or ID. This searches for the attribute from the Target object where the retrieved value should be copied to. Select the attribute from the result list displayed below the search bar.



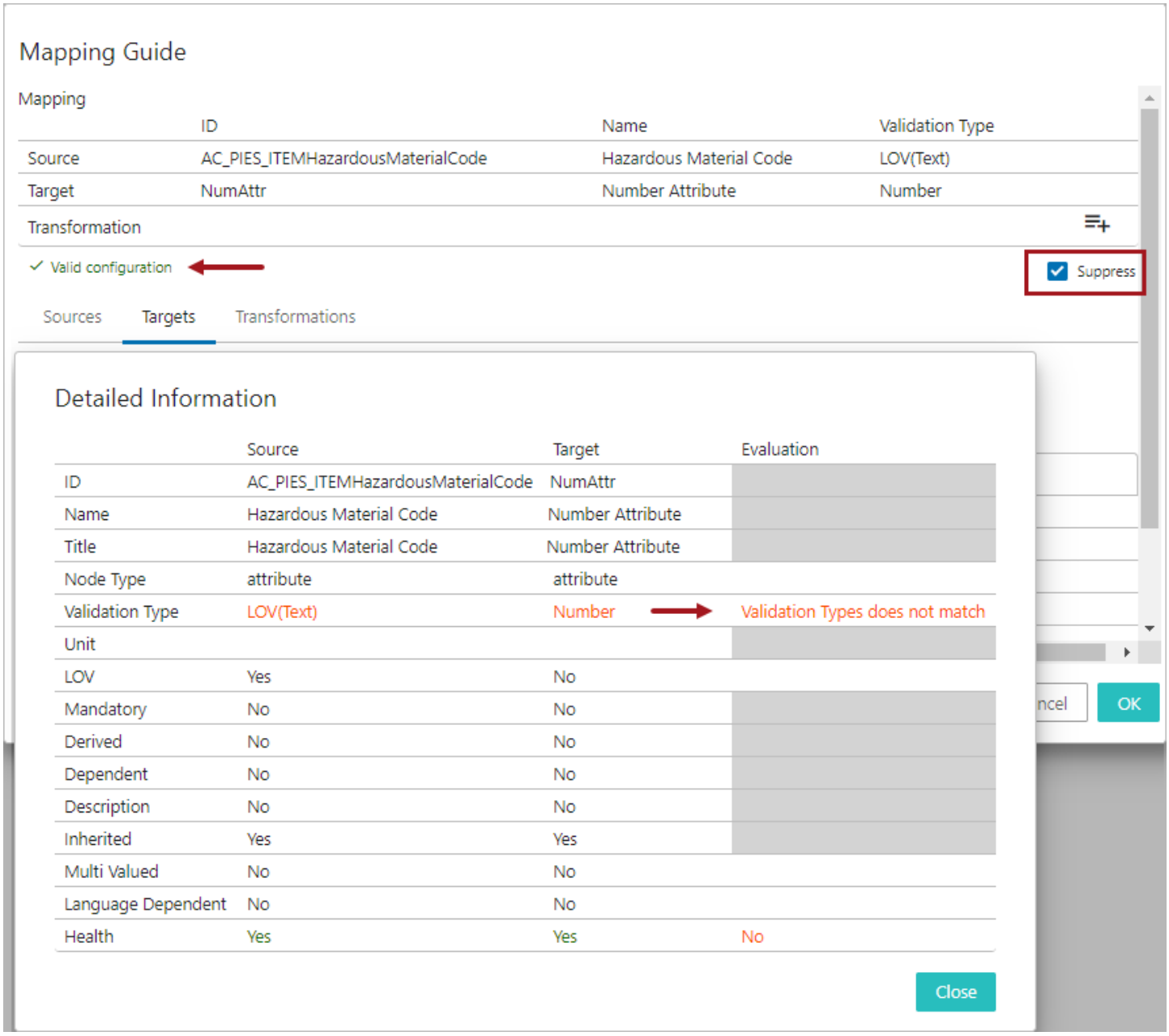

Once selected, the attribute will be populated in the Target field (as shown below).



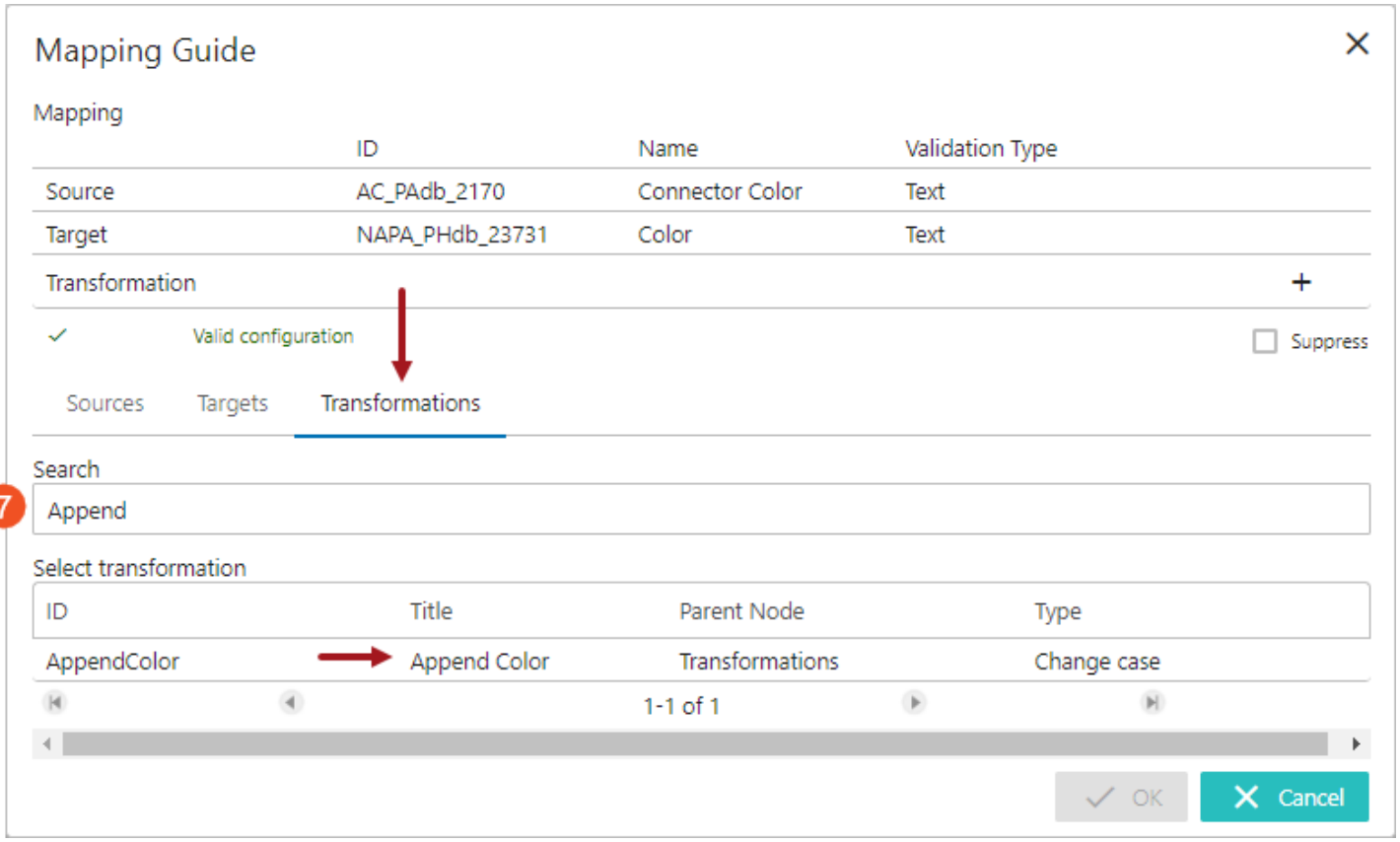

After the source and target attributes are defined, the system evaluates the validity match between the source and the target attributes. A hyperlink text explaining the reason for the validity match / mismatch gets displayed. Clicking on the hyperlink will open the 'Detailed Information' dialog and 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:




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.

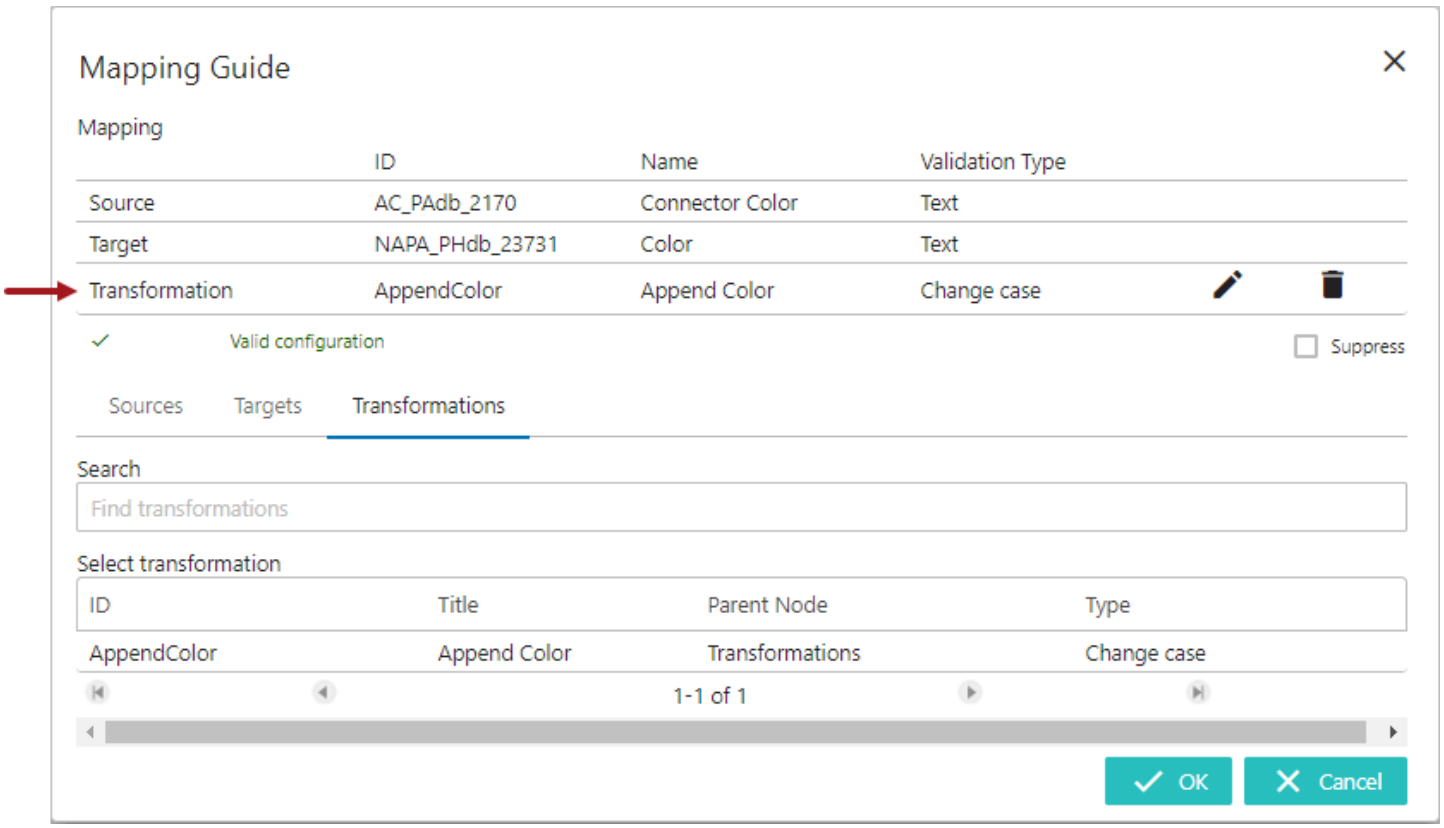



7. If transformations are needed, users can either use an existing attribute transformation in the system or create a new one by clicking the 'Create new transformation' icon ( + ) available within the Transformation field. To configure an existing attribute transformation, users should select an existing attribute transformation listed in the 'Transformations' tab displayed in the lower half of the window (as shown below).

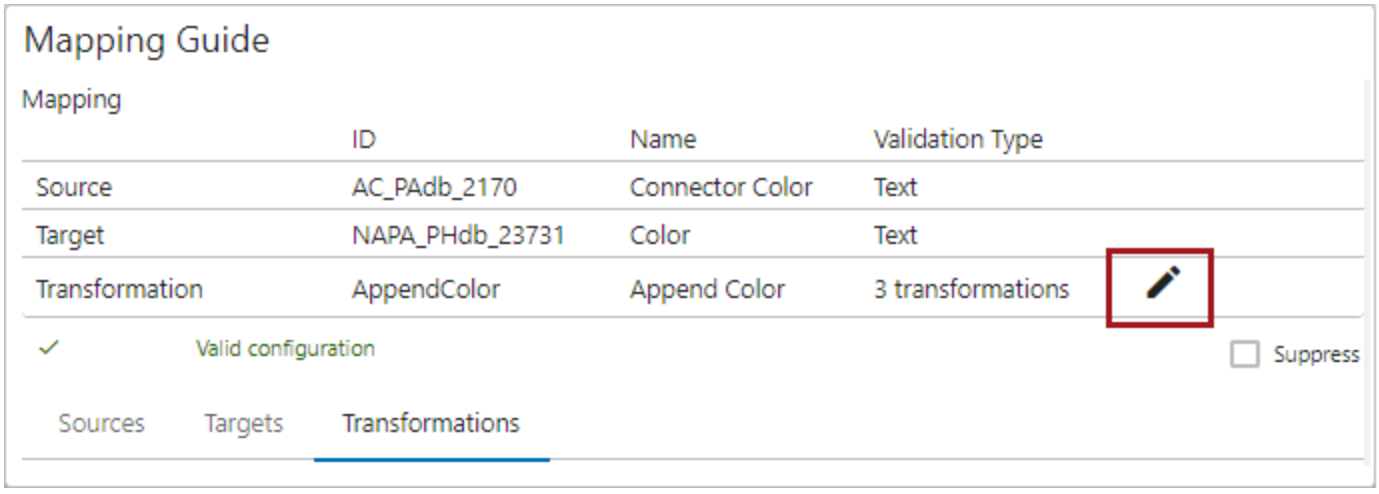



Clicking on the 'Create new transformation' icon ( + ) available within the Transformation field will open the Transformation Overview window where the user can create a new attribute transformation. For details about creating an attribute transformation standing on a Mapping Guide window, refer to **Creating Attribute Transformations Through Mapping Guide Window** topic within this guide.

The selected transformation will be populated in the Transformation field (as shown below).

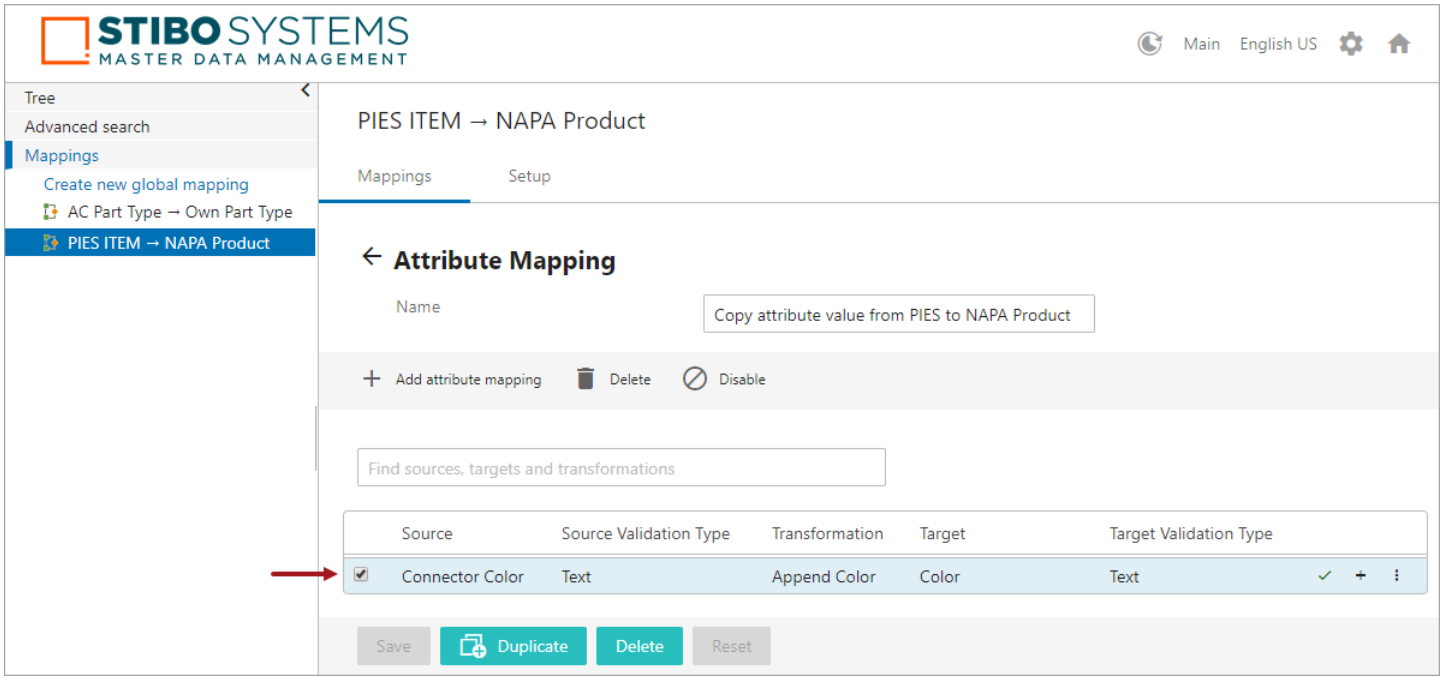



The populated transformation can be edited / removed by clicking the edit icon ( ) in the Transformation field.



**Note:** Deleting an attribute transformation from the Mapping Guide window means removing it from the current mapping setup; however, editing it impacts all of the other mappings that have used the particular attribute transformation. Users are advised to be cautious before editing the attribute transformation. The Mapping Guide in the Web UI does not analyze if the transformation is applied to any other mapping.

8. Click **OK** to close the Mapping Guide window and then click **Save** to save the changes. The newly added attribute mapper 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.




9. Repeat the above steps 4 to 8 to add more attribute mapping rows for the same plugin.

**Important:** Users must save the changes before exiting the Mapper Configuration. If the user fails to click **Save**, then the mapping will be lost once the user leaves the Mapper Configuration.

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 mapper row is displayed next to each mapper 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. For more information on handling the mapper rows, refer to topic **Modifying Mapper Rows on the Onboarding Mapping Details Screen** within this guide.

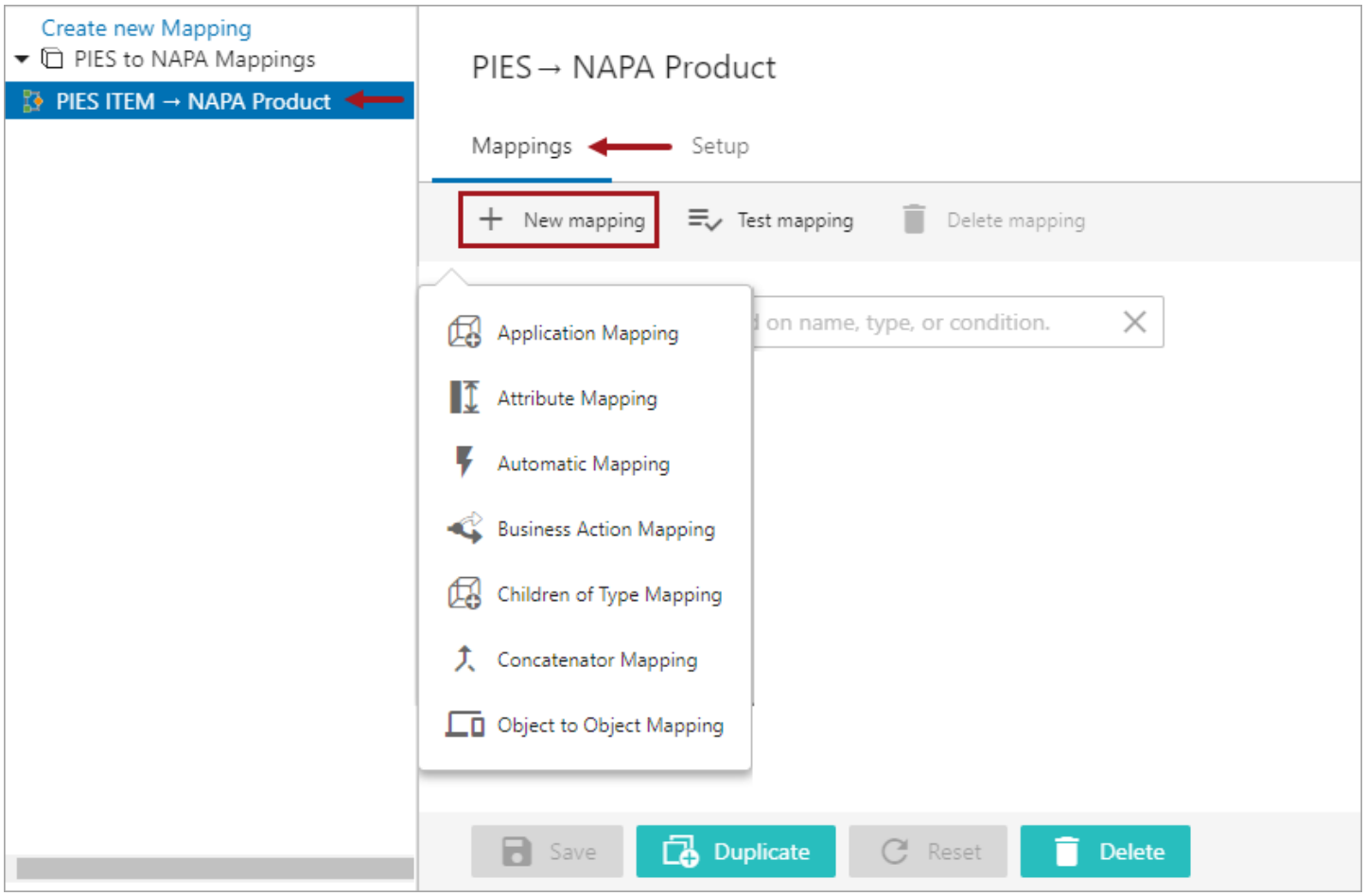
## Automatic Mapping Plugin

The Automatic Mapping plugin will automatically identify shared attributes and references between the source and target objects and will move the attribute values and references from the source over to the target. The plugin must be added to the Onboarding Mappings Details Screen first in order for it to be selectable within a Mapper Configuration.

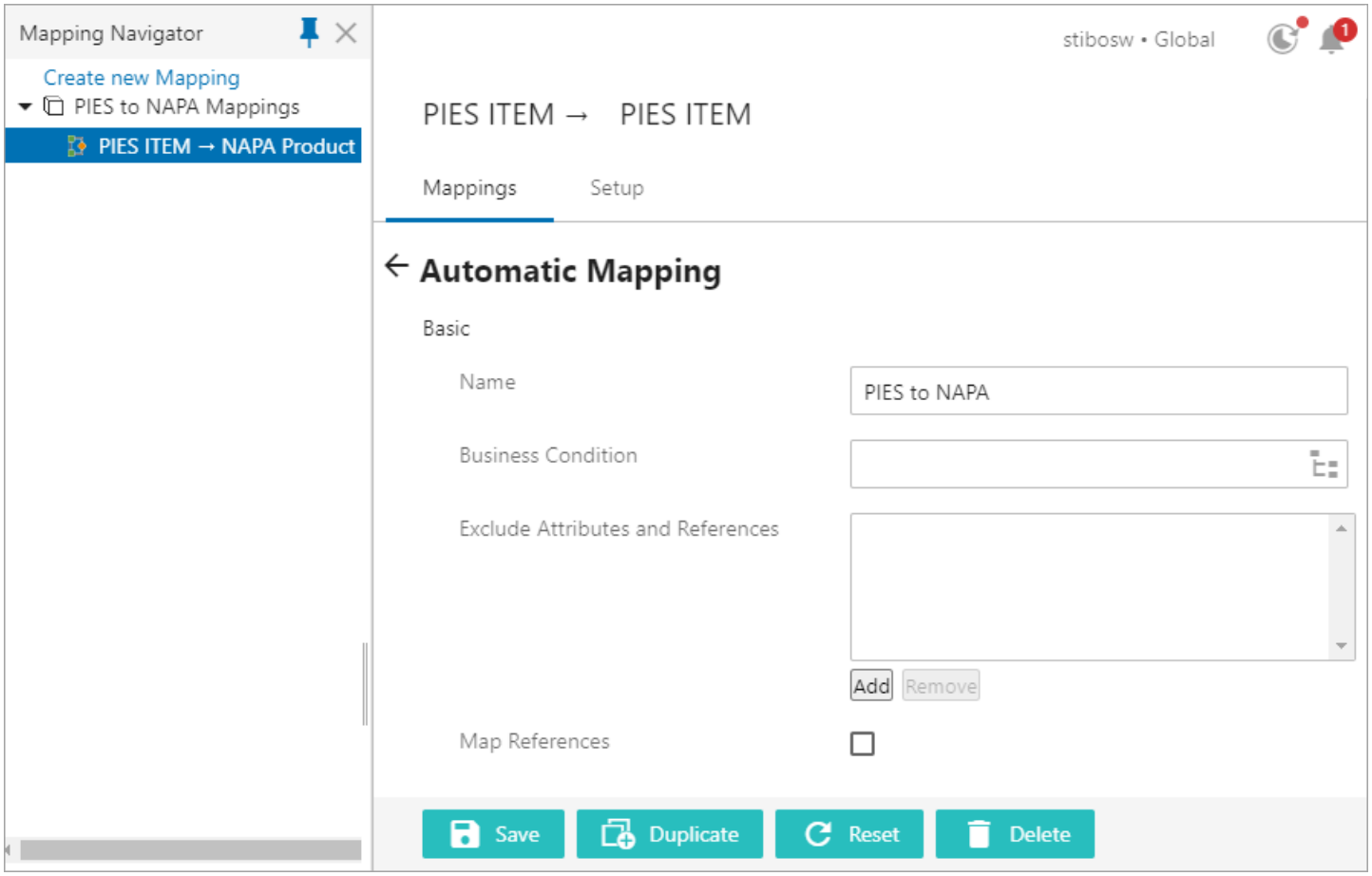
**Note:** All shared attribute values and references will be copied from the source to the target object except for the attribute(s) / reference(s) that is added in the 'Exclude Attributes and References' parameter (as explained below). The settings contained within this plugin cannot include any transformations.

To configure the Automatic Mapping plugin, follow these steps:

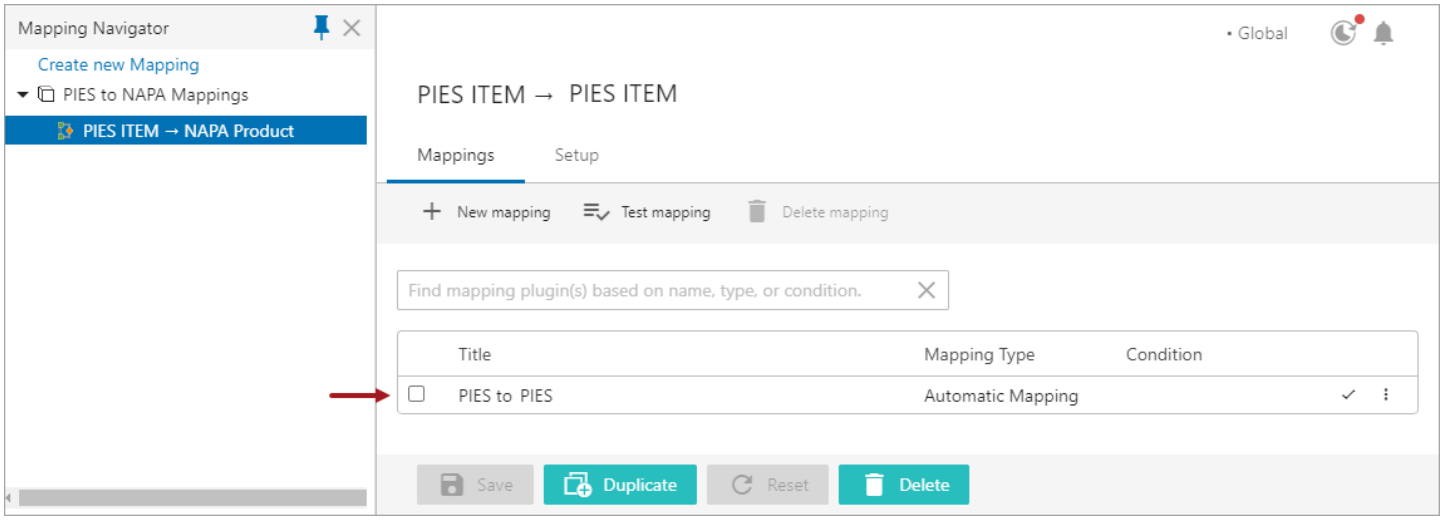
1. With the required Mapper Configuration selected, click on **New mapping** that is available within the Mappings tab of the Onboarding Mappings Details screen.




2. Select **Automatic Mapping**, and a screen is displayed, prompting the user to configure the plugin further (as shown below).




3. Type in a suitable name next to the Name field. This could be any unique name that clearly describes the mapping functionality.
4. In the optional Business Condition parameter, populate a business condition that determines any condition set to execute the mapping plugin. The mapping plugin will be executed only when the defined condition is met.
5. In the 'Exclude Attributes and References' parameter, click the Add button to open the node picker. Then select the desired attributes and / or references that are to be excluded from being onboarded.
6. Select the 'Map References' parameter only if the source object's references are to be onboarded to the target object. Only the references from the shared reference type will be copied to the target object.
7. Select the 'Ignore Attribute Links on Target' parameter only if the Automatic Mapping plugin should include the orphan attributes for onboarding. Leaving this parameter unselected will exclude the orphan attributes from being onboarded.
8. Click **Save** to save the changes. The newly added Automatic Mapping will be listed as shown below.



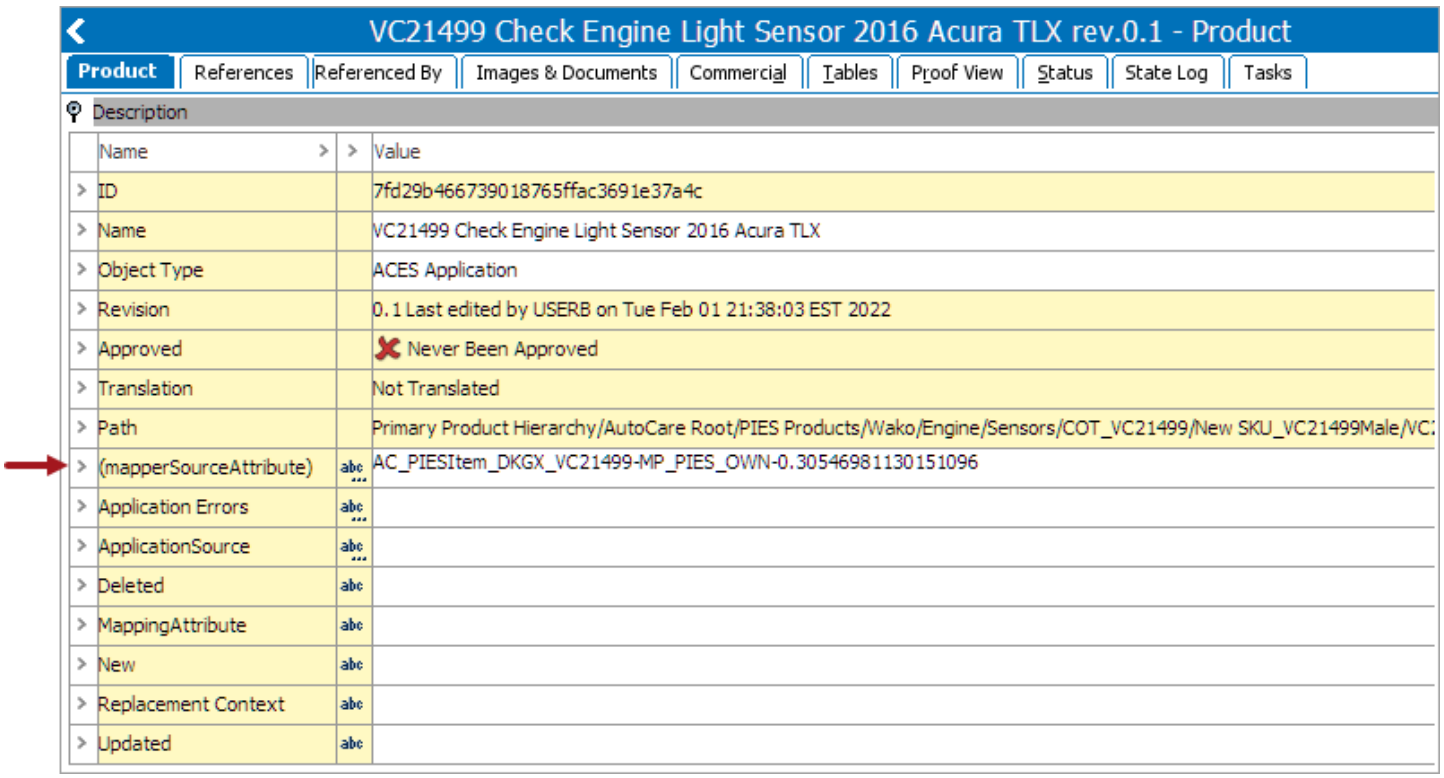

**Important:** Users must save the changes before exiting the Mapper Configuration. If the user fails to click **Save**, then the mapping will be lost once the user leaves the Mapper Configuration.

The health of the Automatic 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 checkmark shown in the image above indicates that the mapping plugin is healthy.

## Children of Type Mapping Plugin

The Children of Type Mapping plugin automatically identifies child object types shared between the source and target objects, and then copies those child objects, along with their data, from the source to the target object. In order for the Onboarding Mappings Details Screen to be selectable within a Mapper Configuration, the plugin must first be added to a Mapper Configuration.

When the Mapper Configuration is run for the first time on the source object, all the source object's eligible child objects are copied to the target object. The child objects created under the target object by the mapper are assigned a unique code that populates the attribute 'mapperSourceAttribute'. This code distinguishes the mapper-created child object from the child objects created by other external methods (for example, child objects created by an import). The mapper later uses this identification whenever an update is run on the existing child objects in the target.

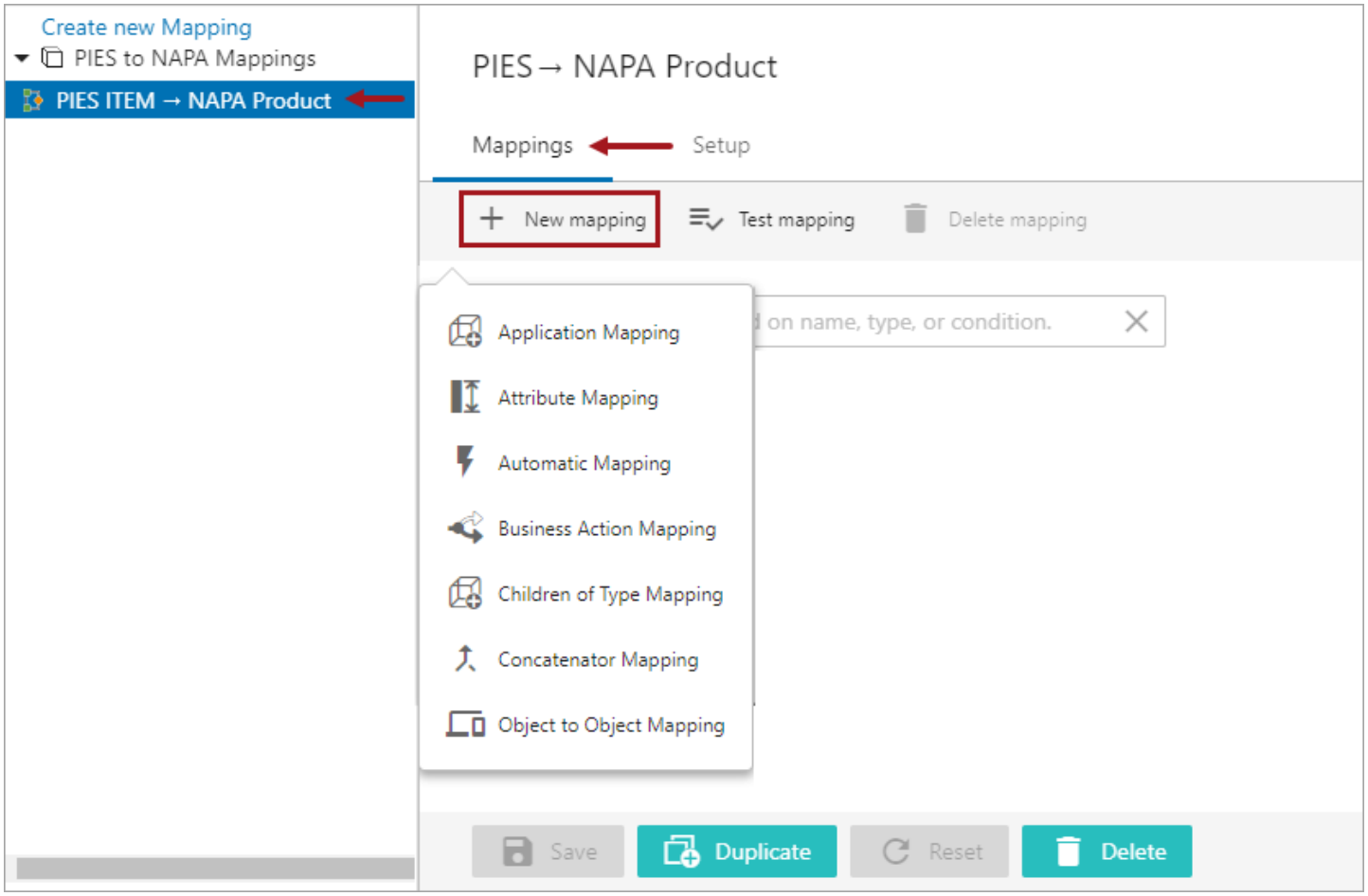



**Note:** If a child object on the target is created by an external source and not the mapper, then the mapper will not run an update on that child object even if an identical child object exists on the source. Updates to existing child objects on the target are run only if that child object was created by the mapper and not by any other method.

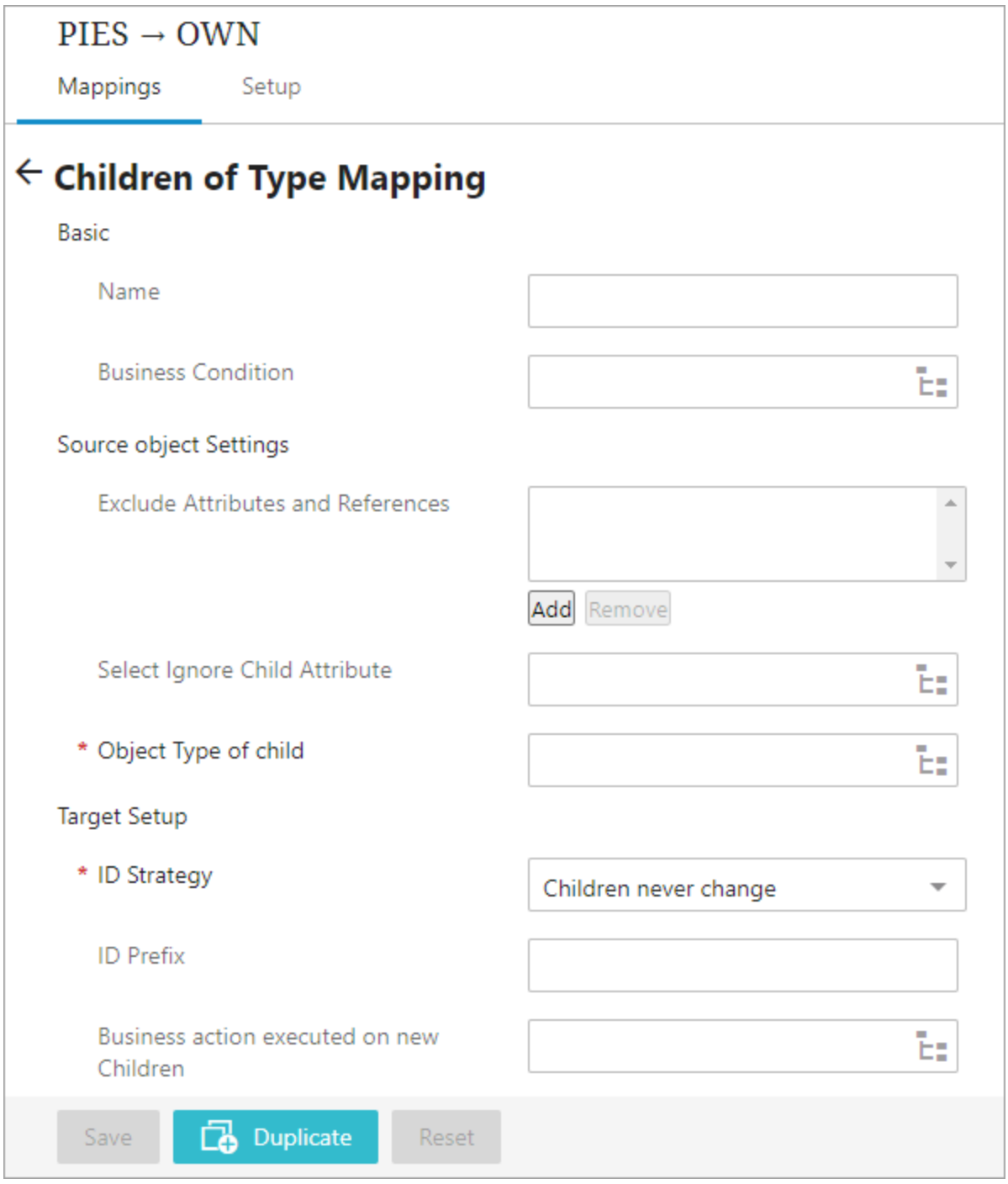
**Note:** This plugin disallows application of transformations to attributes when copying child objects from the source to the target objects.

To configure the Children of Type Mapping plugin, follow these steps:

1. With the required Mapper Configuration selected, click on **New mapping** that is available within the Mappings tab of the Onboarding Mappings Details screen.




2. Select **Children of Type Mapping** from the dropdown that displays. This prompts display of a screen that enables the user to configure the plugin further (as shown below).




3. Type in a suitable name next to the Name field. This could be any unique name that clearly describes the mapping functionality.
4. In the optional Business Condition parameter, populate a business condition that determines under what conditions the mapping will be run. The defined business condition will run on the source object and not the child object. The mapping plugin will be executed only when the defined condition is met.
5. Populate the parameters displaying in the 'Source object Settings' section:

- **Exclude Attributes and References:** This parameter allows users to add attributes and references to be excluded when onboarding target child objects. Clicking the Add button opens a node picker dialog from which users can select attributes and references. It is important to note that users must select attributes and references belonging to the child object only, not the source object. Users cannot select inherited attributes.
- **Select Ignore Child Attribute:** This parameter allows users to select an attribute that is valid for the child of the source object. The attribute selected for this parameter determines whether a child object is to be onboarded or not when the Mapper Configuration is executed on the source object. If the specified attribute belonging to the child object holds the value 'true,' then such child objects will be ignored on onboarding.

  For example, if the user wants to skip onboarding the application records that are marked for deletion, then the attribute used for flagging the deletion of the application record is configured in this parameter. So if there is an application record that has the delete attribute marked 'True,' then that application record will not be onboarded.
- **Object Type of Child:** In this required parameter, users define the object type of the child object to be onboarded.

6. Populate the parameters available in the Target Setup section:
   - **ID Strategy:** The options available in this parameter define a strategy to handle the ID for the child in the target object during an update. The option selected in this parameter determines how an existing target child object is identified, handled, and updated.

     If a target child object has similar data to the source child object, the mapper will verify if the target child object is created by the mapper based on the unique identification code populated in the 'mapperSourceAttribute' attribute. If the target child object is created by the mapper, then the mapper handles the existing child object based on the following options selected within the parameter.

     - **Children never change:**

       When this option is selected, the mapper looks for an existing child object (as described above) and handles the mapper-created child objects as follows:

       - If an existing target child object is eligible for an update, the mapper will delete and replace it with a newly updated child object (with a new object ID). However, new child object will retain the same 'mapperSourceAttribute' attribute value (unique code) as the replaced and deleted child object.
       - If an existing target child object is not eligible for an update, then the mapper will skip those objects. Thereby the objects remain untouched in the target.

       With this option selected, the ID of new objects are created using the following pattern:

       ```
       [prefix]MD5(DataType.ID="value",DataType.ID="value"…)
       ```

       The list id="value" is alphabetically sorted. If the values are multivalued, they will be separated by the tag "<multisep/>". The values will be listed in the order supplied by the source.

- **Children can have updates:** With this option selected, whenever an update is run over an existing child object in the target, the mapper does not delete and recreate the child object but instead updates the existing child object. Meaning, the ID of the target child object remains the same even after multiple updates.

  This strategy works fine for most updates, but in some cases, it will end up deleting the target object and creating a new one if a target object has multiple source objects and the primary source object is deleted. Hence the ID pattern can no longer be used to identify the update.

  With this option, the ID pattern is based on the source object:

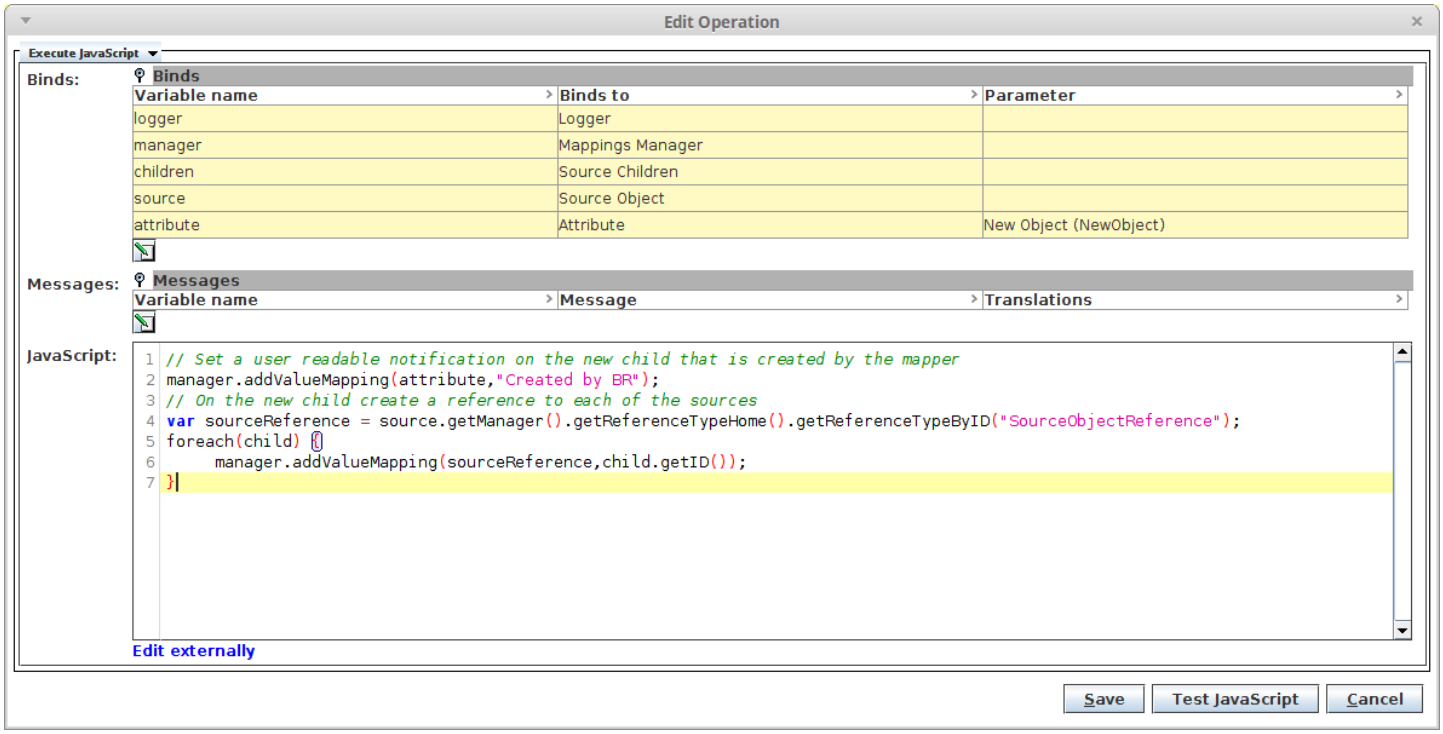
```
[prefix]MD5(source.ID)
```

- **ID Prefix:** If required, users can define an ID prefix for the new child objects created under the target object.
- **Business action executed on new Children:** This parameter allows users to add a business rule that will be run on the target child objects created by the mapper.

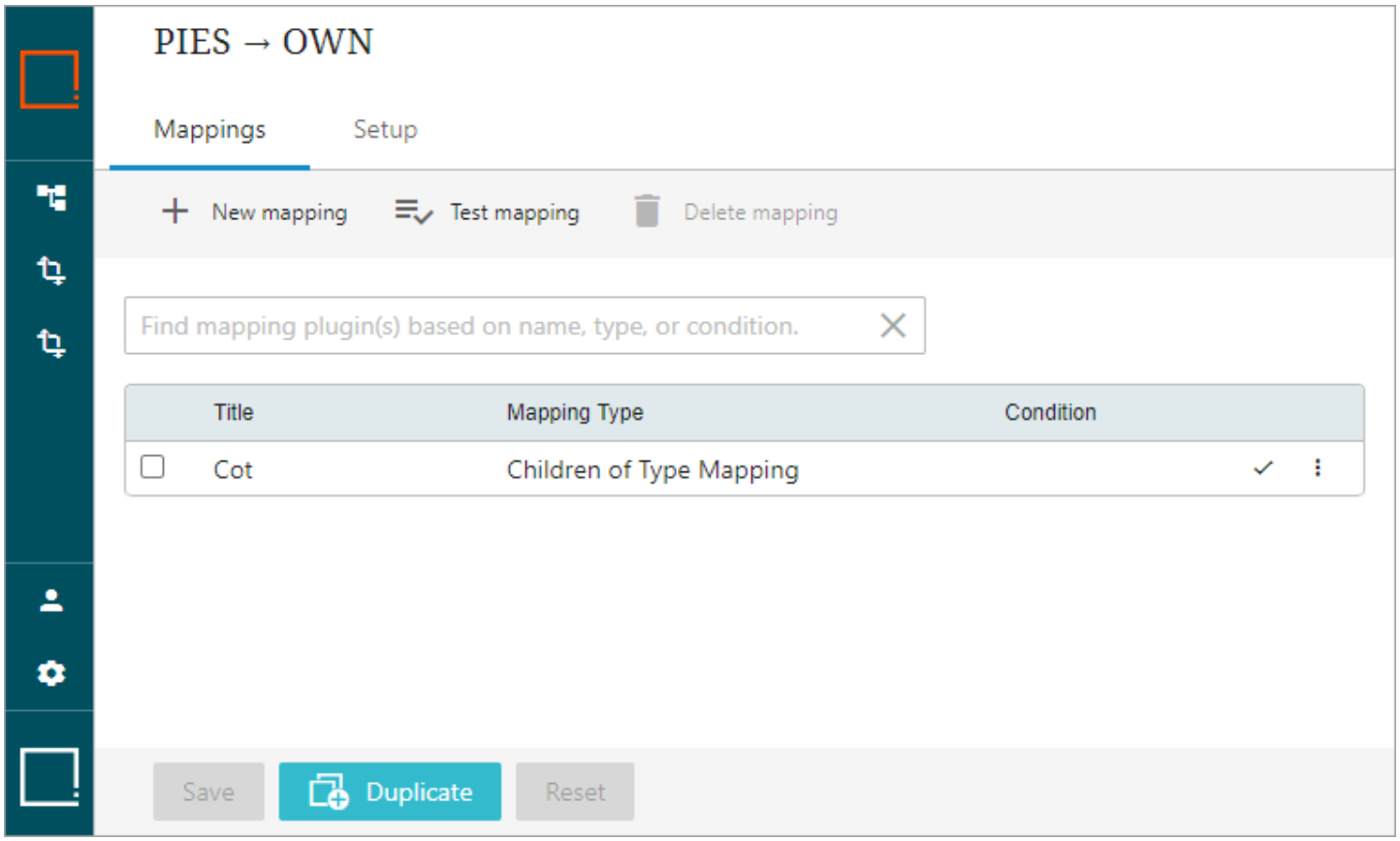
  While configuring a business rule to be used in this parameter, it is recommended that the following binds be used as necessary along with the other standard binds as required:

| Bind | Functionality |
|---|---|
| **Mappings Manager** | When the Mapper Configuration is being executed, it is quite likely the actual new child object is not created in the target yet. This means the business rule *cannot* use the Current Object bind. Instead, the business rule must use the Mappings Manager bind that has only one method<br>addValue(DataType datatype, String Value).<br>This allows the business rule to create new references and set attribute values on the child object. It is not possible to create new objects through the Mappings Manager bind. |
| **Source Object** | Used to retrieve data specific points from the source object |
| **Source Children** | The set of children (objects) that are merged into the new target child. The set always contains at least one object. |

Below is an example of a business rule using the mapper-specific binds.




7. Click **Save** to save the changes. The newly added Children of Type Mapping will be listed as shown below.



**Important:** Users must save the changes before exiting the Mapper Configuration. If the user fails to click **Save**, the mapping will be lost once the user leaves the Mapper Configuration.

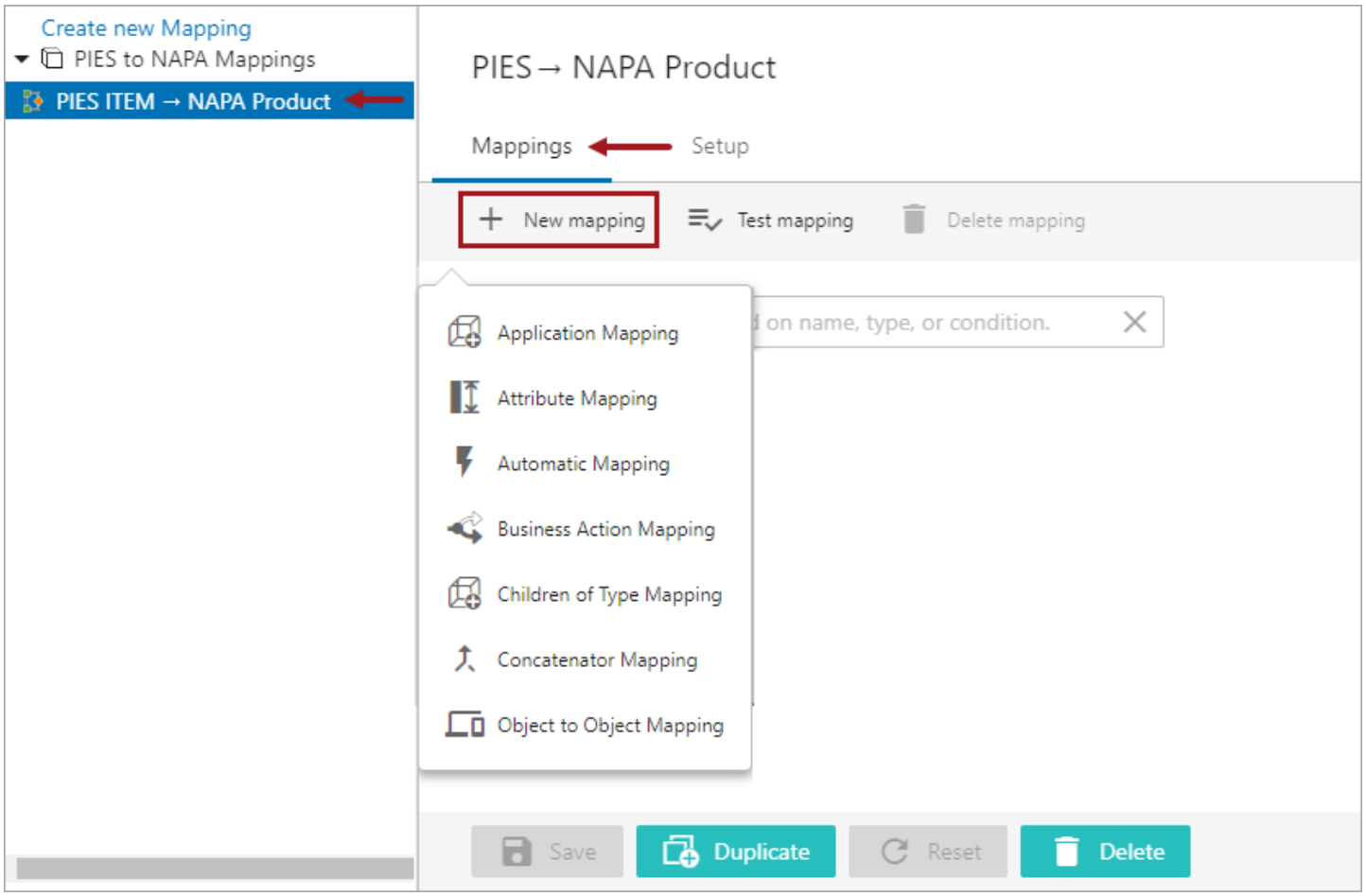
The health of the Children of Type Mapping plugin is displayed next to each Mapping plugin. The user has the flexibility to disable, delete, or rearrange the listing order of the Mapping plugins. The checkmark shown in the image above indicates that the mapping plugin is healthy.

## Concatenator Mapping Plugin

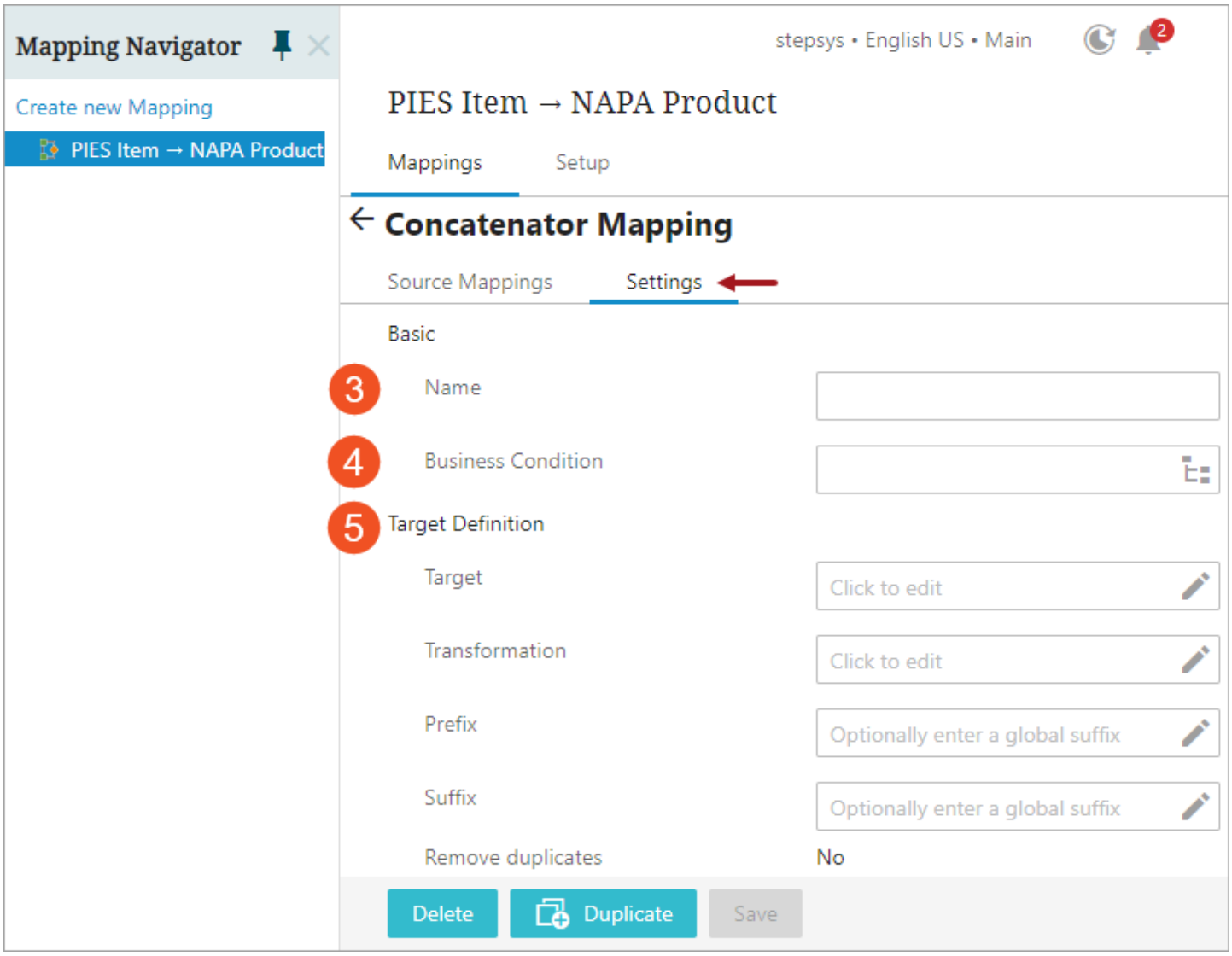
The Concatenator Mapping plugin allows users to combine multiple source values into a single target attribute while applying different texts that can be prepended or appended to the concatenated values, or used to separate those. The plugin is configured on the Onboarding Mappings Details Screen for the selected Mapper Configuration.

Before following the steps described below to configure this plugin, users must first determine which target attribute will be populated with the concatenated value, and then define the source attributes. So, it is advised to follow the steps mentioned below to avoid confusion. To configure the Concatenator Mapping plugin, follow these steps:

1. With the required Mapper Configuration selected, click the **New mapping** button found at the top of the Mappings tab on the Onboarding Mappings Details screen.

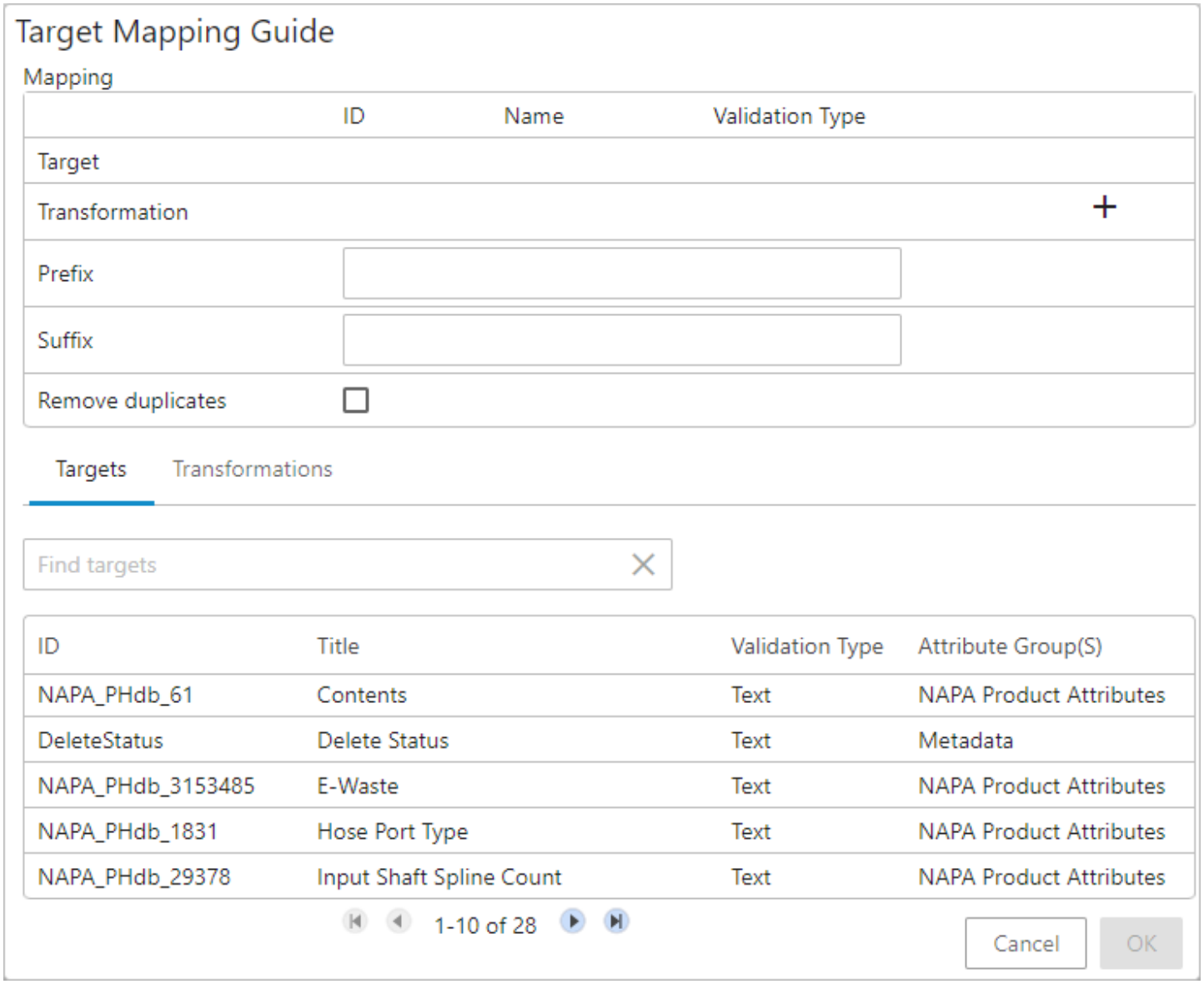


2. Select Concatenator Mapping from the New mapping button's dropdown, and a screen displays with two tabs, Source Mappings and Settings.




3. On the Settings tab, type the desired name in the Name field. This could be any unique name that describes the mapping functionality.
4. If needed, add a business condition in the 'Business Condition' parameter that defines running the mapper plugin only on the object if the condition is true.
5. Populate the parameters in the Target Definition section.

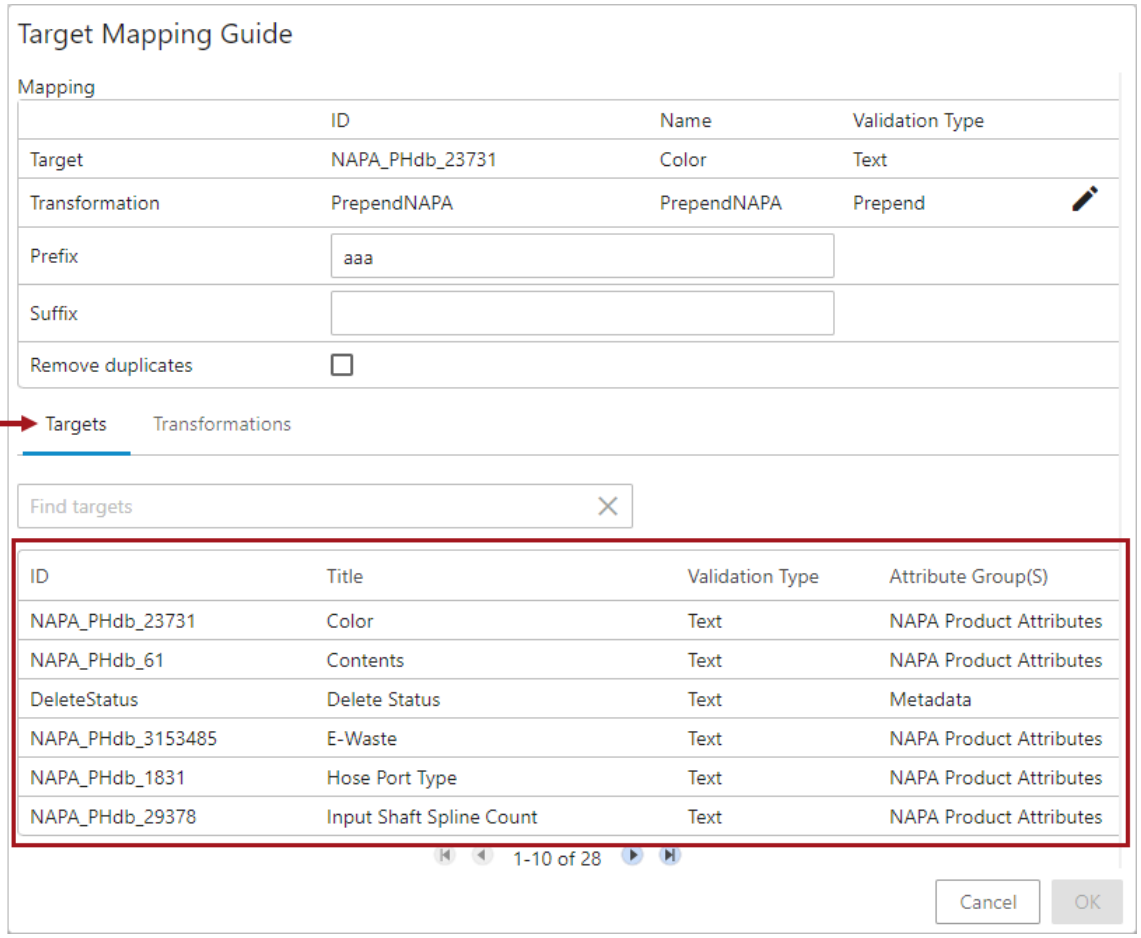
   To configure the five parameters available under the Target Definition field (Target, Transformation, Prefix, Suffix, and 'Remove duplicates'), users must access the Target Mapping Guide window. Click on the edit icon ( ) available next to any of the parameters to access this window. The same parameters will be available for editing within the Target Mapping Guide window, as shown in the screenshot below.




Below are the details about these parameters:

- **Target:**Displaying at the top of the Mapping table, the Target parameter displays the ID, Name. and Validation Type of the Target object attribute. Multiple values retrieved from the source object are concatenated and copied to this attribute. To display a target attribute in the Mapping table, users must select an attribute listed in the table on the 'Targets' tab displayed in the lower half of the window (as shown below). A search bar above the table enables typeahead searching, which allows users to type in the few characters of the attribute name, and then select the desired target attribute from the results.

  It is mandatory to have an attribute defined in the Target parameter.

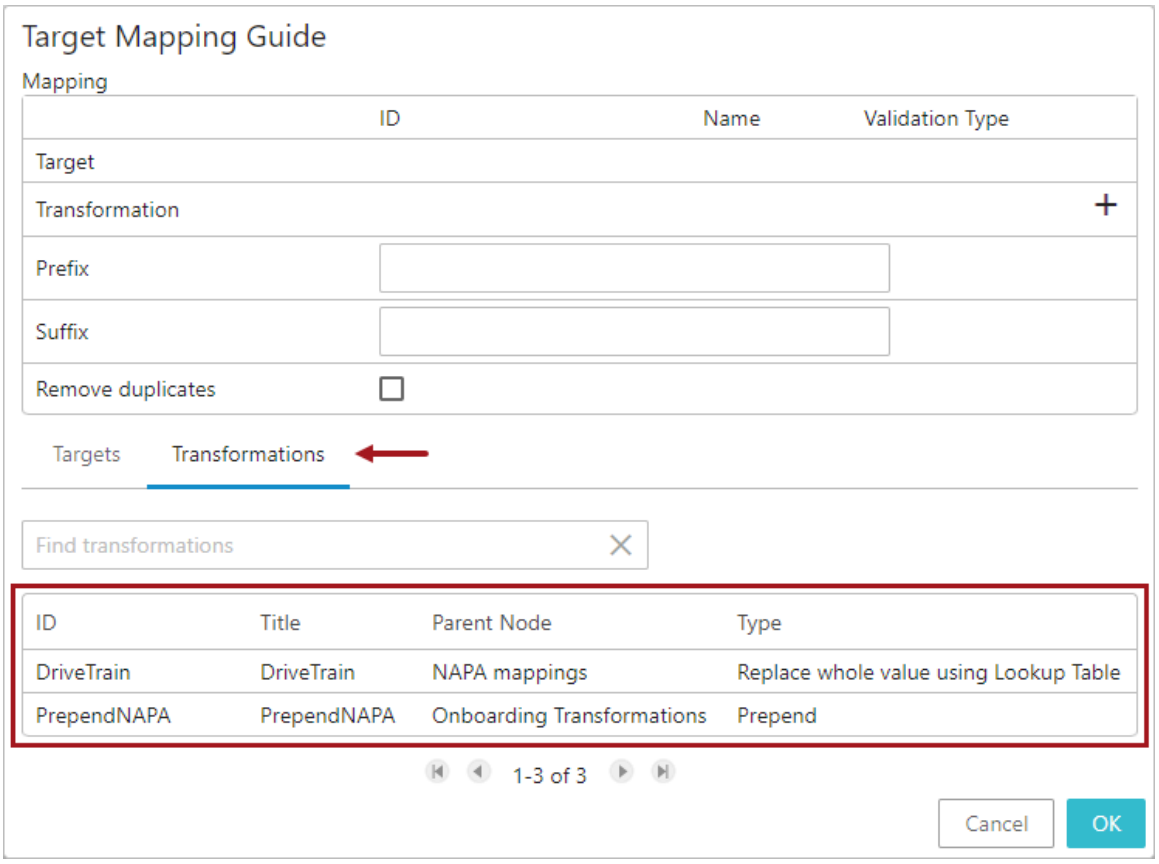



**Note:** The Concatenator Mapping plugin can have multiple source attributes but not multiple target attributes.

- **Transformation:** This field allows users to add an attribute transformation. The attribute transformation added within this parameter will be applied to the final concatenated value and not the individual values retrieved from multiple source attributes. Users can either use an existing attribute transformation in the system or create a new one by clicking the 'Create new transformation' icon (+) available within the Transformation field. To configure an existing attribute transformation, users should select an existing attribute transformation listed in the 'Transformations' tab displayed in the lower half of the window (as shown below).

  Clicking on the 'Create new transformation' icon (+) available within the Transformation field will open the Transformation Overview window where the user can create a new attribute transformation. For details about creating an attribute transformation standing on a Mapping Guide window, refer to **Creating Attribute Transformations Through Mapping Guide Window** topic within this guide.

The populated transformation can be edited / removed by clicking the edit icon ( ) in the Transformation field.

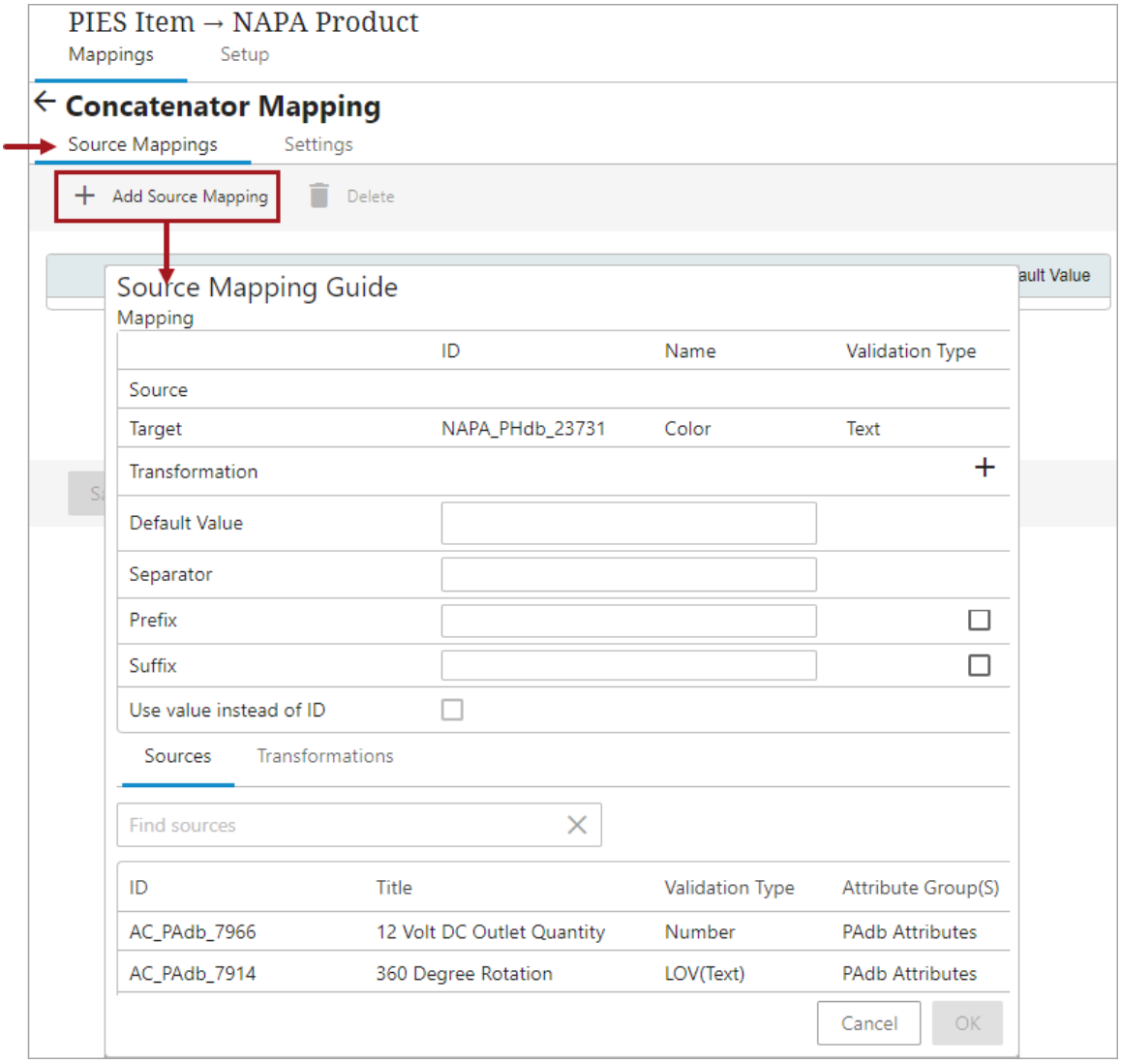



- **Prefix:** If the onboarding process requires any prefix to be prepended to the target attribute value, the user can enter a prefix into the Prefix field. The specified prefix will be applied to the final concatenated value and not on the individual values retrieved from multiple source attributes.
- **Suffix:** If the onboarding process requires any suffix to be appended to the target attribute value, the user can enter the suffix to the Suffix field. The specified suffix will be applied to the final concatenated value and not on the individual values retrieved from multiple source attributes.
- **Remove Duplicates:** Checking the box for this parameter will ensure that only one value of any identified duplicates is populated in the target attribute.

**Important:** As the Concatenator Mapping plugin allows users to retrieve values from multiple source attributes and copy them to one target attribute, any configuration settings (transformation, prefix, etc.) made within the Target Mapping Guide will only be applied to the final concatenated value on the target attribute and *not* on the individual values retrieved from the source attributes.

Once all the parameters within the Target Mapping Guide window are populated, click the **OK** button at the bottom right of the window to close the window. All the values populated within the Target Mapping Guide window will be reflected accordingly within the five parameters (Target, Transformation, Prefix, Suffix, and 'Remove duplicates') available under the Target Definition field.

6. Once all the necessary parameters in the Settings tab are populated, click the Source Mappings tab to define the source attributes. Then click the Add Source Mapping button to open the Source Mapping Guide window.

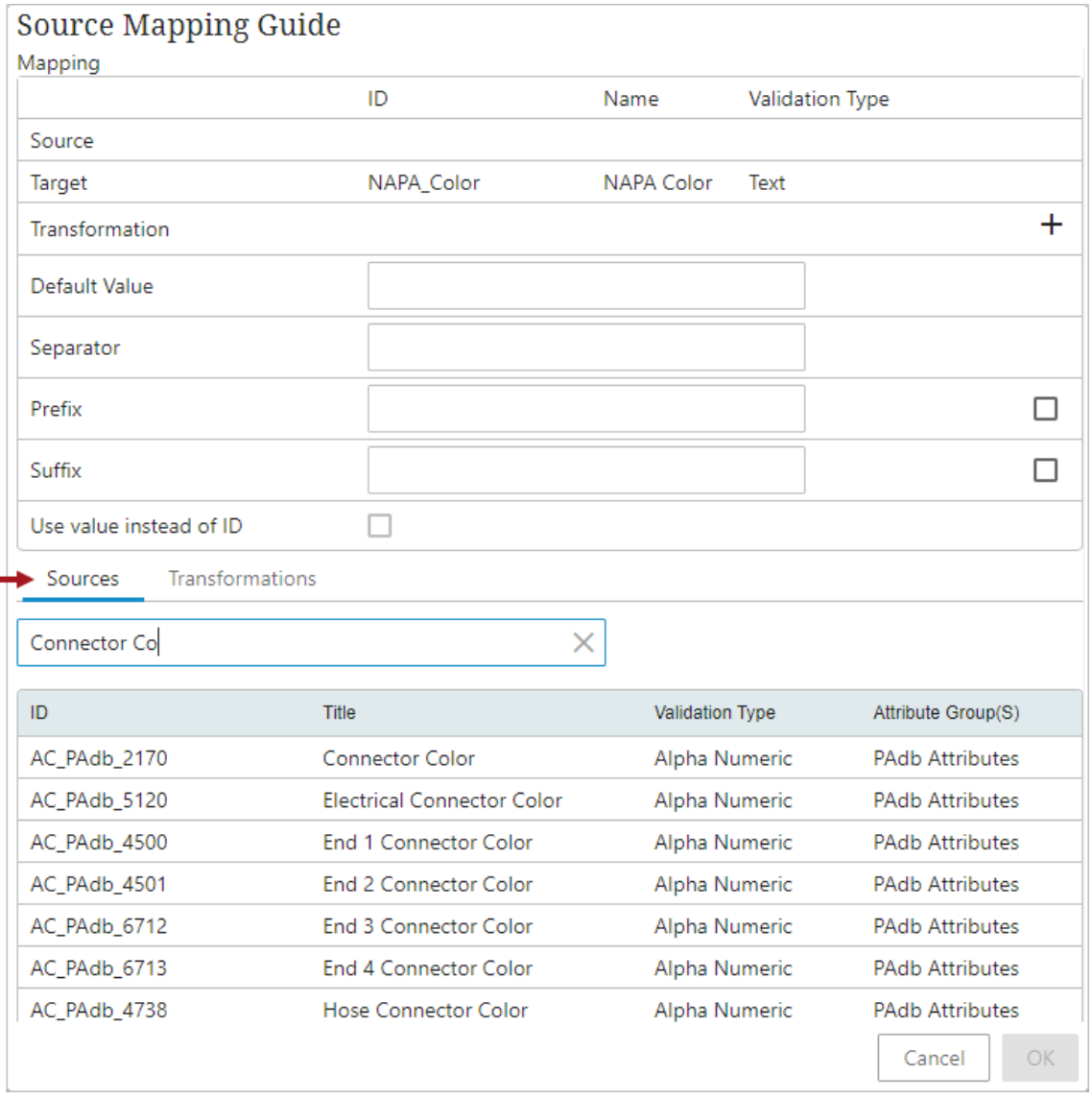



The Source Mapping Guide window allows users to define a source attribute, and configure the value retrieved from the defined source attribute.

**Important:** As the Concatenator Mapping plugin allows users to retrieve values from multiple source attributes and copy them to one target attribute, any configuration settings made within the Source Mapping Guide will only apply to the individual values retrieved from the source attribute and *not* on the final concatenated value expected to be on the target attribute.

Below are the details about these parameters:

- **Source:** The Source row displays the attribute belonging to the Source object from which the value is retrieved. To configure a source attribute to this field, users should select an attribute from the table on the 'Sources' tab found in the lower half of the window (as shown below). A search bar above the table enables typeahead searching, which allows users to type in the first few characters of the desired source attribute name, and then select the desired source attribute from the results. Further, users can take the help of the typeahead field to type in the few characters of the attribute name to find their appropriate source attribute. It is mandatory to have an attribute defined in the Source parameter.

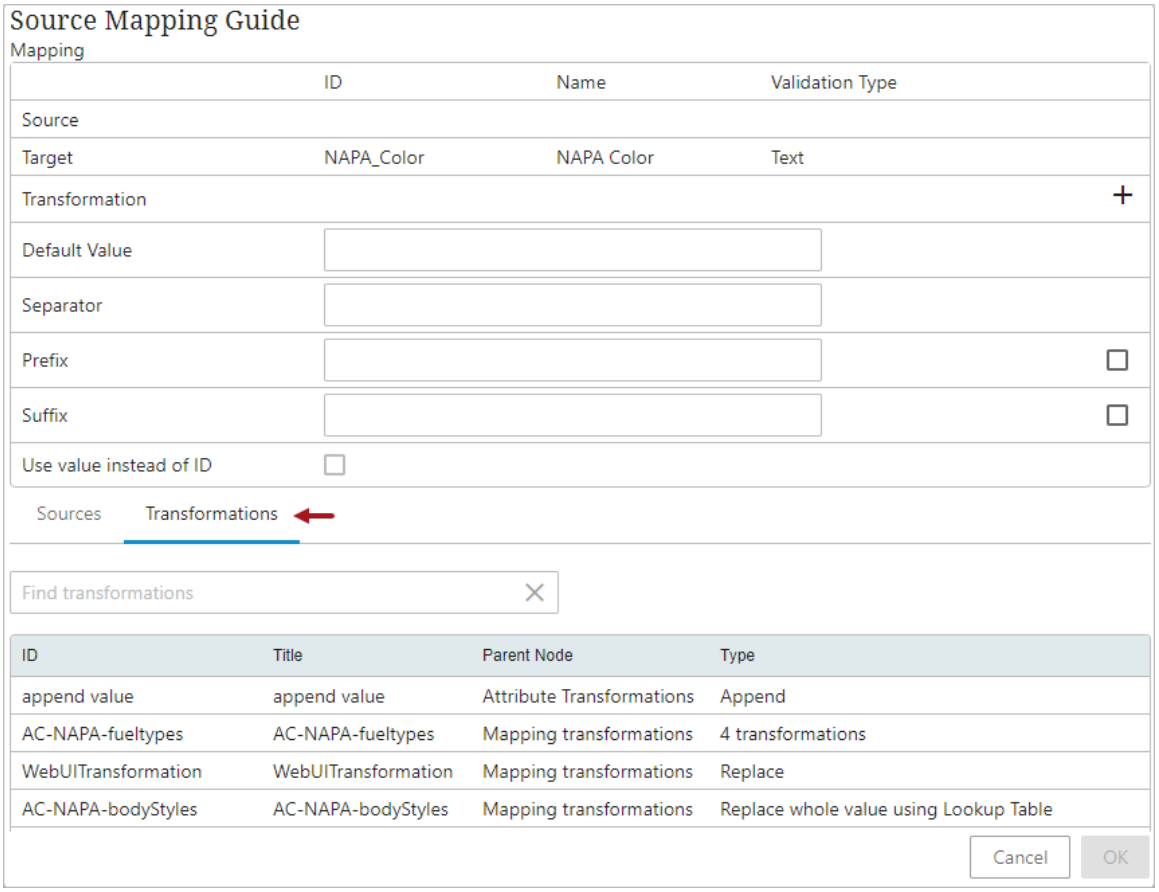



- **Target:** This field displays the target attribute where the retrieved value is copied to. Users should access the Target Mapping Guide window as defined in step 5 of this document to edit the target attribute.
- **Transformation:** This field allows users to add an attribute transformation. The attribute transformation added within this parameter will be applied to the selected source attribute's value and not the final concatenated value. Users can either use an existing attribute transformation available in the system or

create a new one by clicking on the 'Create new transformation' icon ( + ) available within the Transformation field. To configure an existing attribute transformation, users should select an attribute transformation that is listed in the 'Transformations' tab displayed in the lower half of the window (as shown below).

Clicking on the 'Create new transformation' icon ( + ) available within the Transformation field will open the Transformation Overview window where the user can create a new attribute transformation. For details about creating an attribute transformation standing on a Mapping Guide window, refer to **Creating Attribute Transformations Through Mapping Guide Window** topic within this guide.

The populated transformation can be edited / removed by clicking the edit icon ( ) in the Transformation field.

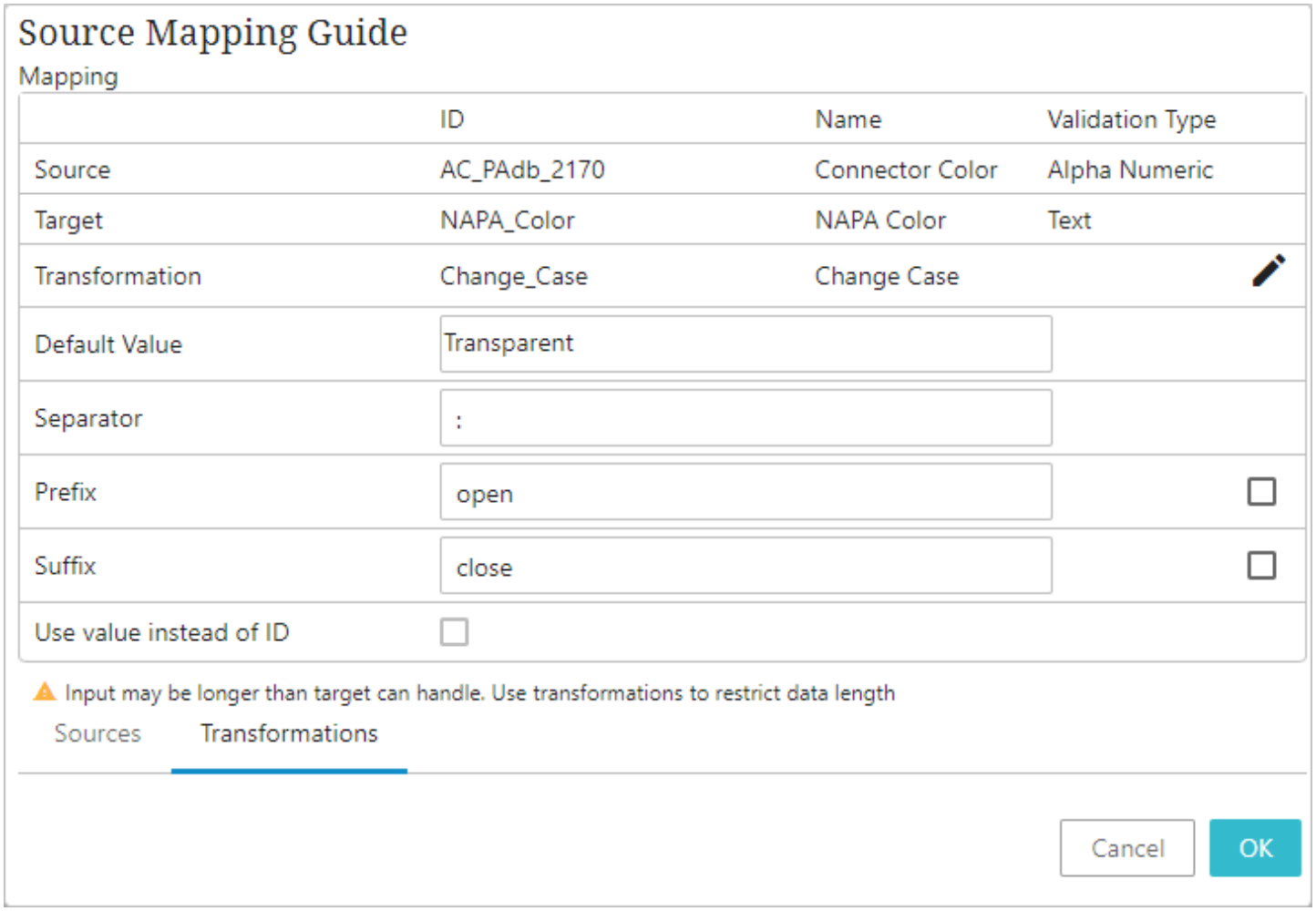



- **Default Value:** This field allows users to enter a text which needs to be populated within the target attribute value when the source attribute remains empty for the executed source object. This gets only relevant if the source attribute you are mapping is empty. Otherwise, the default value will not function.
- **Separator:** Any character entered in this field will be a value delimiter when the Concatenator Mapping plugin has multiple source attributes or a single source attribute with multiple values.
- **Prefix:** Any character entered in this field will be placed before the retrieved source attribute value. This gets only relevant if the source attribute you are mapping has any value in it. Otherwise, the prefix will not

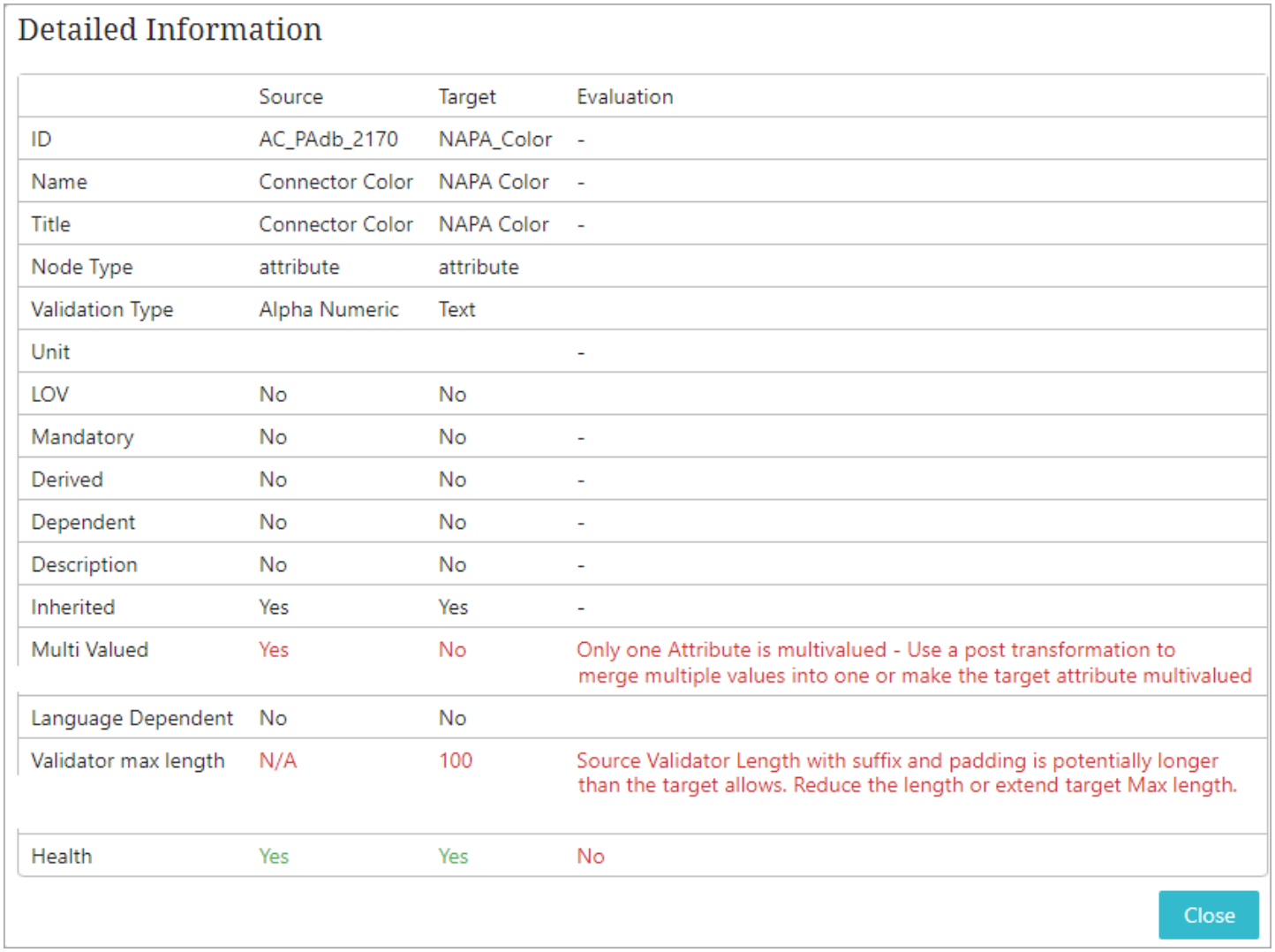
be set on an empty attribute. In addition, if the Default Value parameter includes any characters, then the prefix will be set on the default value.

- **Suffix:** Any character entered in this field will be placed after the retrieved source attribute value. This gets only relevant if the source attribute you are mapping has any value in it. Otherwise, the suffix will not be set on an empty attribute. However, if the Default Value parameter includes any characters, then the suffix will be set on the default value.
- **Use Value instead of ID:** This option gets only relevant if the source attribute is of LOV validation type. Otherwise, the checkbox will be grayed out. The default is set to retrieve the ID of the populated LOV from the source attribute and not the LOV value. However, selecting this checkbox will retrieve the LOV value from the source attribute for mapping.

Below is an example screenshot of the Source Mapping Guide window when the parameters are populated:




After the source and target attributes are defined in the Source Mapping Guide window, 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 in the middle of the window. Clicking on the hyperlink opens the 'Detailed Information' dialog (shown below) and displays the validity match / mismatch information of the mapper. Below is an example of a validation mismatch between the source and the target attribute displayed in the 'Detailed Information' dialog:

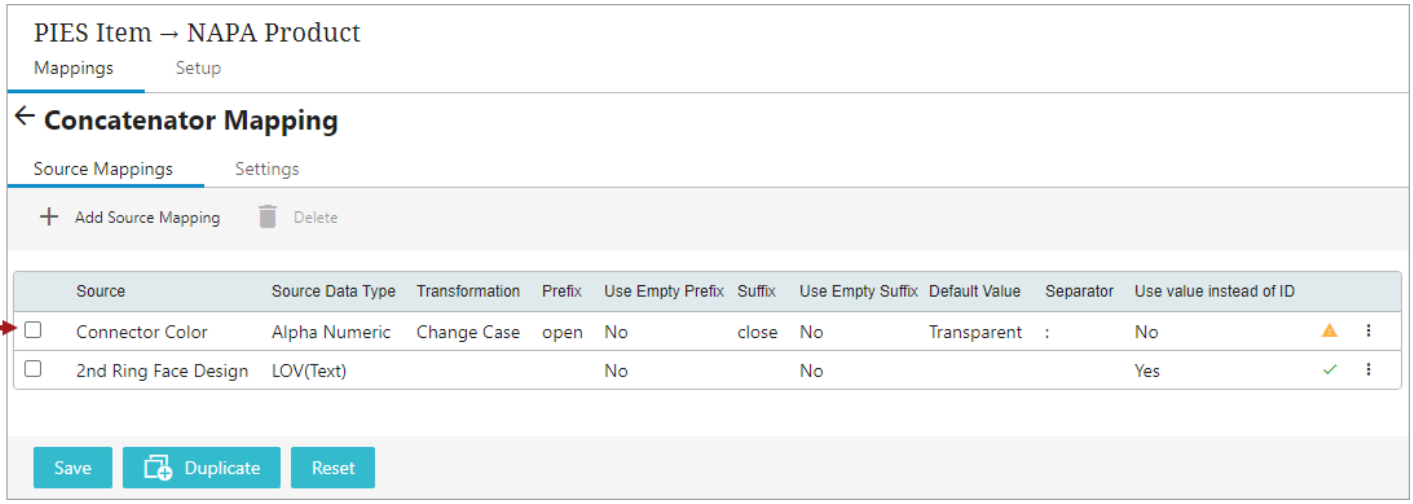


Detailed Information

| | Source | Target | Evaluation |
|---|---|---|---|
| ID | AC_PAdb_2170 | NAPA_Color | - |
| Name | Connector Color | NAPA Color | - |
| Title | Connector Color | NAPA Color | - |
| Node Type | attribute | attribute | |
| Validation Type | Alpha Numeric | Text | |
| Unit | | | - |
| LOV | No | No | |
| Mandatory | No | No | - |
| Derived | No | No | - |
| Dependent | No | No | - |
| Description | No | No | - |
| Inherited | Yes | Yes | - |
| Multi Valued | Yes | No | Only one Attribute is multivalued - Use a post transformation to merge multiple values into one or make the target attribute multivalued |
| Language Dependent | No | No | |
| Validator max length | N/A | 100 | Source Validator Length with suffix and padding is potentially longer than the target allows. Reduce the length or extend target Max length. |
| Health | Yes | Yes | No |

Close

Once all the necessary parameters within the Source Mapping Guide window are populated, click **OK** to close the Source Mapping Guide window and then click **Save** to save the changes. All the values populated within the Source Mapping Guide window parameters will be displayed as a mapper row in the table within the Source Mappings tab (as shown below). To edit the source attribute selections, click on the row, and it will open up the Source Mapping Guide window in order to make edits.




**Note:** After the user edits and closes the Source Mapping Guide / Target Mapping Guide window, the **Save** button remains enabled until the user clicks on the **Save** button. If the user fails to click **Save**, then the mapping changes will be lost if the user navigates out of the selected Mapper Configuration.

7. Repeat step 6 to add more source attributes to concatenate the values for the defined target attribute.

Users can add any number of mapper rows (source attributes) 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.

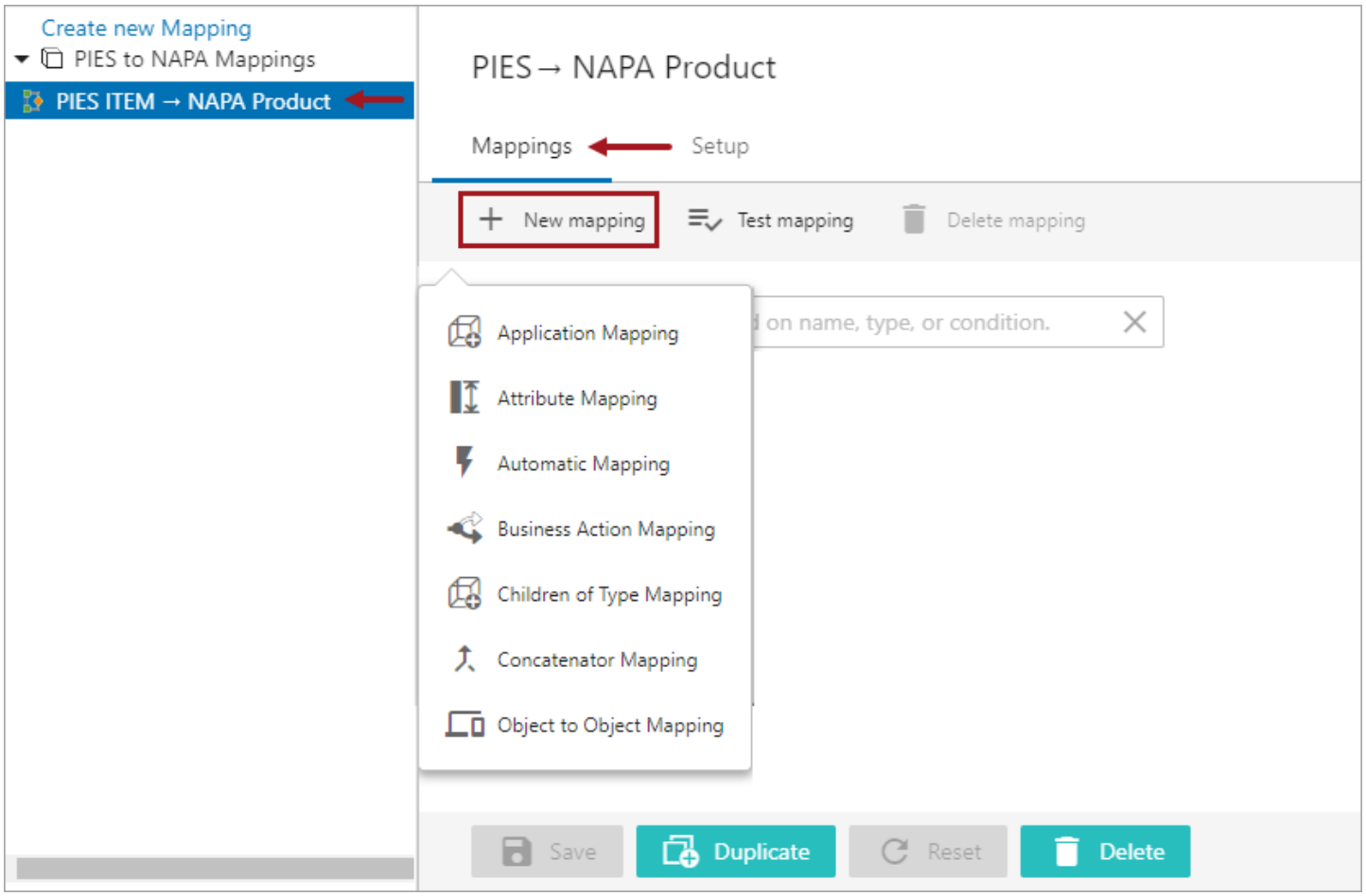
The health of the mapper row is displayed next to each mapper row. Users can also add additional information describing each of the mapper rows. The user has the flexibility to disable, delete, or rearrange the listing order of the mapper rows. For more information on handling the mapper rows, refer to the topic **Modifying Mapper Rows on the Onboarding Mapping Details Screen** within this guide.

## Object to Object Mapping Plugin

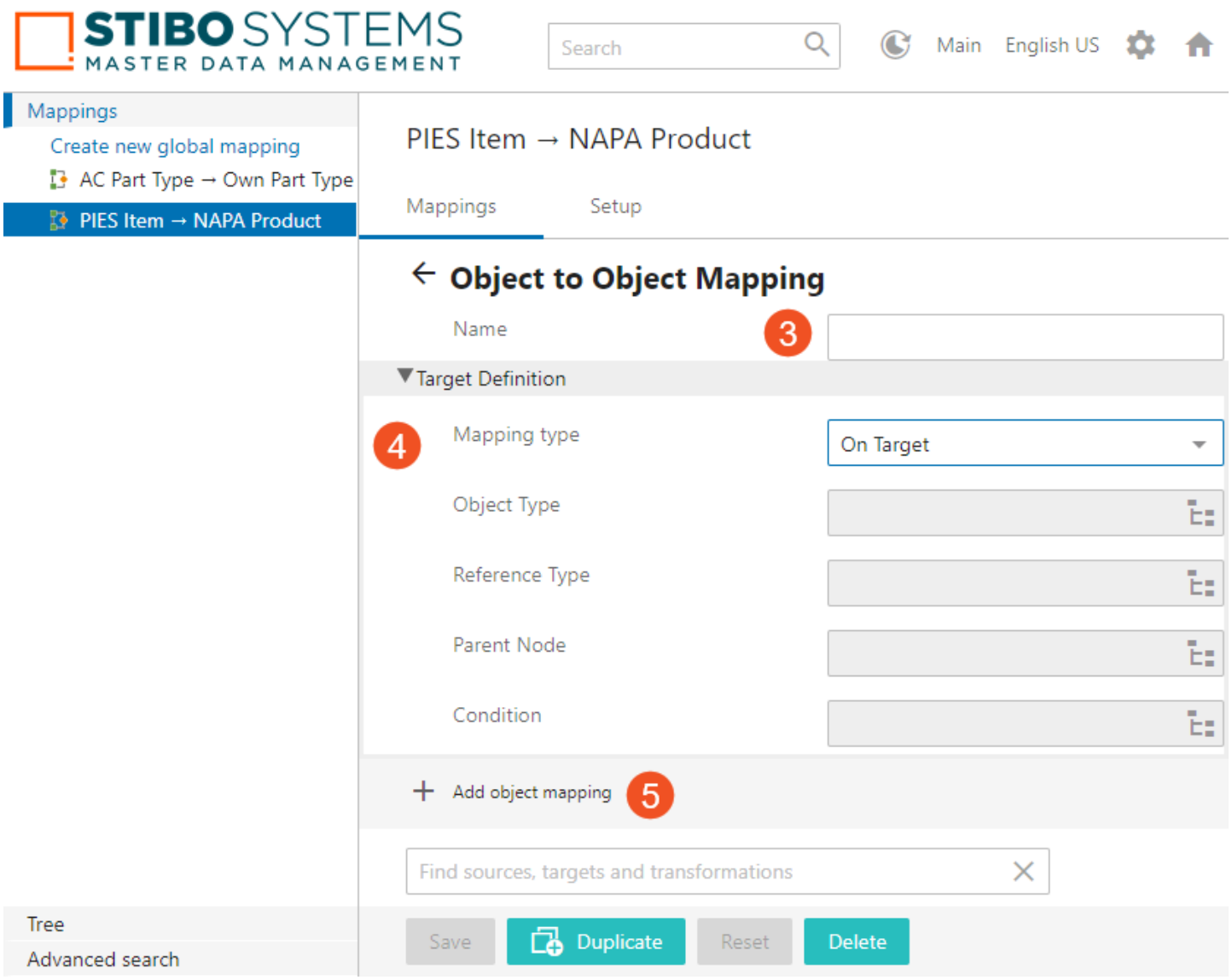
Object to Object Mapping plugin will copy the information present on the Source object or from its related object (Source STEP Path) to the Target object or to its related object. The information retrieved from the Source object could be the ID, name, attribute values, or references. The information 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.

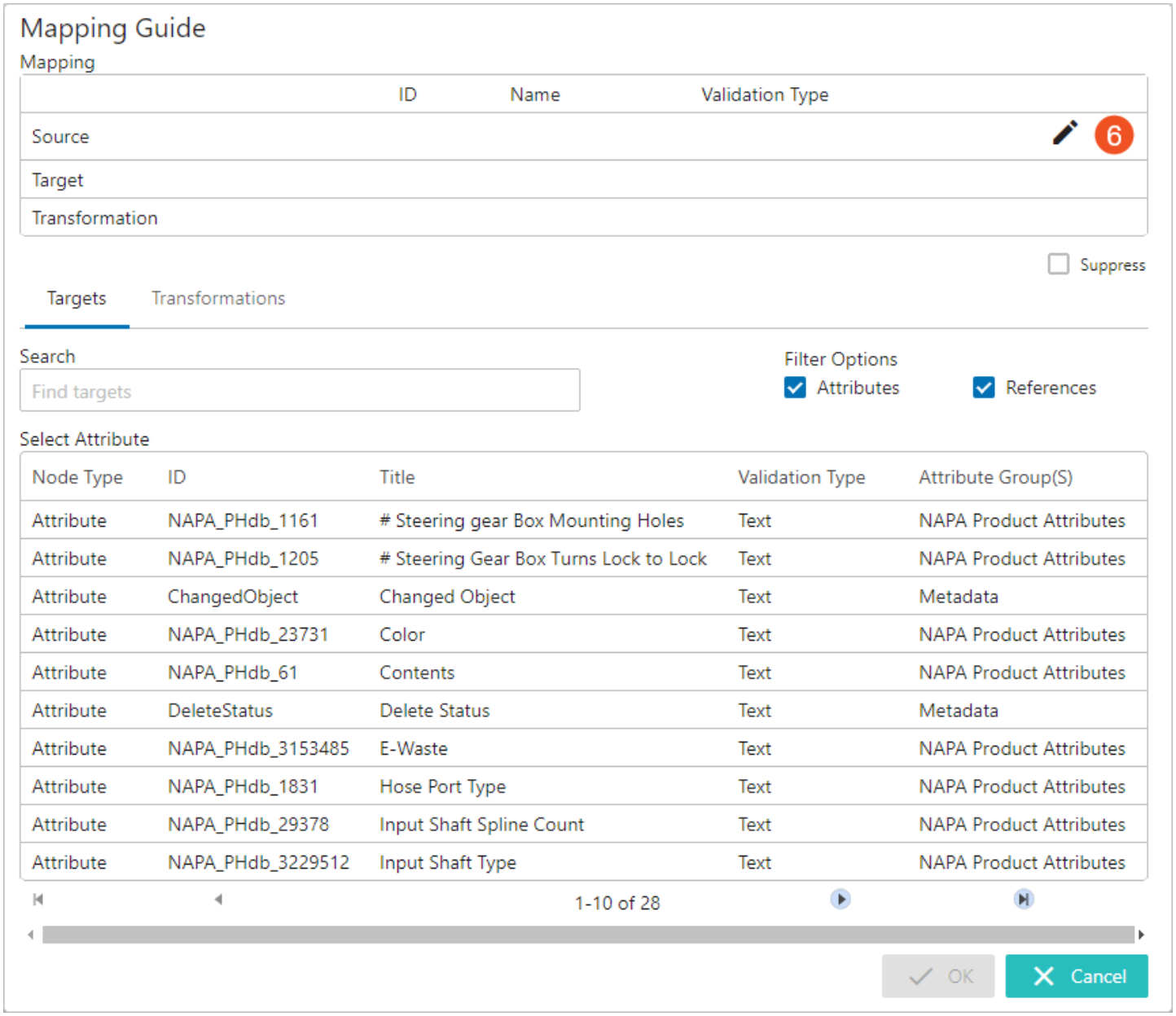
- **Child:** Selecting this option will evaluate the Source STEP Path in order to get the data to be used to create child objects below the Target object. The child objects that are created can either be Classification or Product objects. The data that is retrieved from the Source object can be used 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.

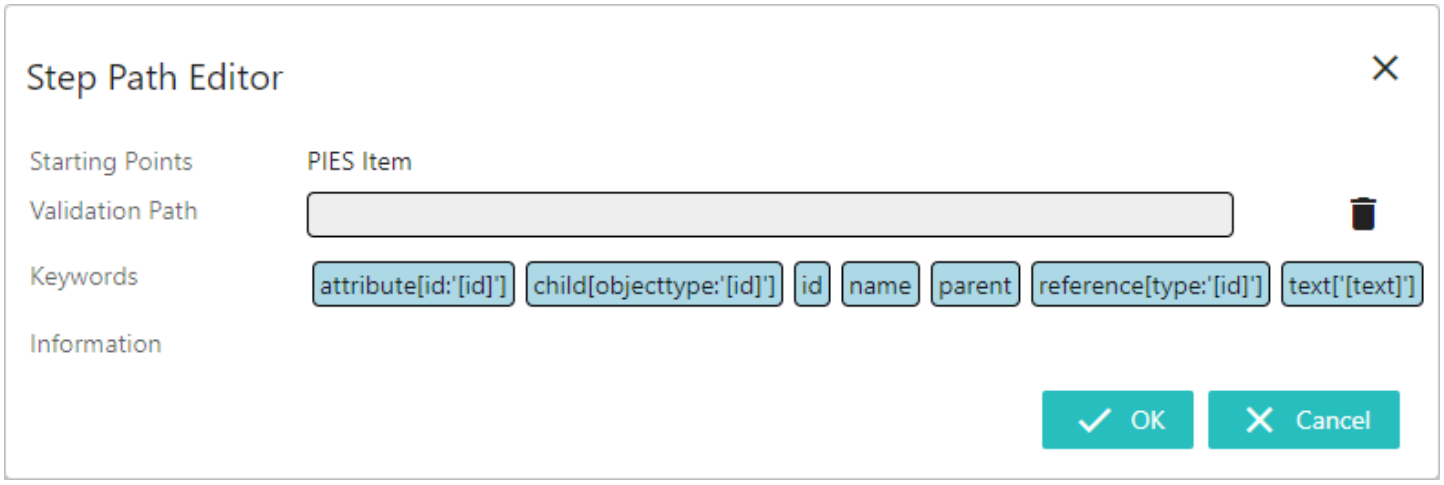
- **Reference:** Selecting this option will evaluate the STEP Path on the Source object to create new objects below another object, and then establish a reference between the Target object and the newly created object. The data that is retrieved from the Source object can be used in the new objects as 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.
- **Condition:** This parameter is used when either the Child or Reference Mapping option is selected within the 'Mapping type' parameter. The user can select a business condition that gets executed when the associated Mapper Configuration setup entity is executed. This parameter will most often be used to define a business condition that decides whether the objects related to Target is to be created or not.

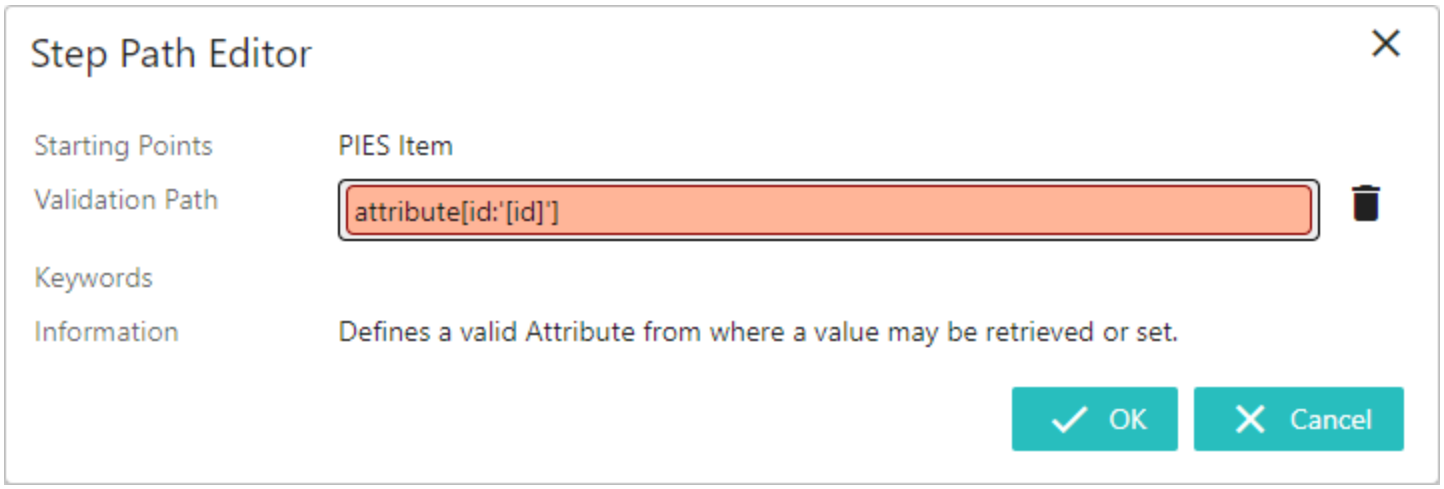
5. Click the ‘Add object mapping’ icon, and the Mapping Guide window will display.




6. 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, refer to the Mapping Validation Path Functionality topic within this guide.

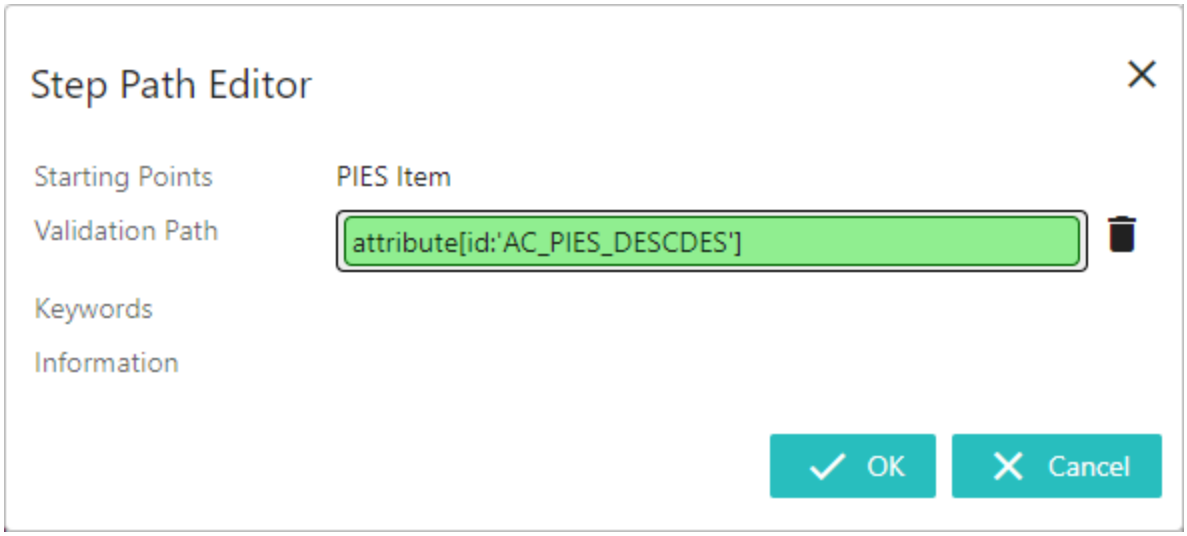



7. Double-click on the required element(s) from the 'Keywords' field. The selected element will be populated in the 'Validation Path' field (as shown below). The specific combination of elements are chosen depending on the unique requirements for the data being retrieved. Selecting the elements will 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.

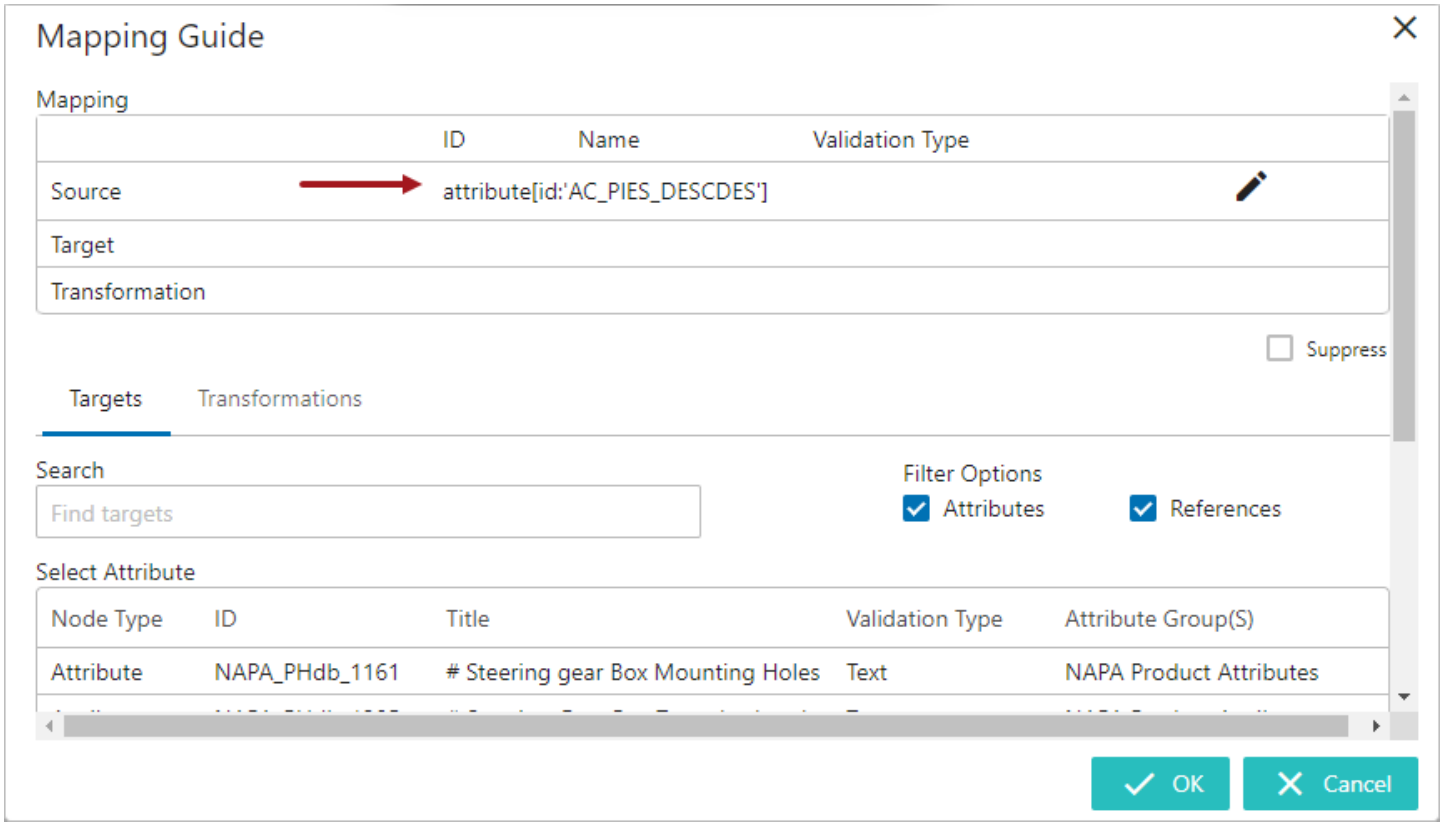



Users can add one element or a combination of elements to create a Source STEP Path. For more information on creating the Source STEP Path, refer to the Mapping Validation Path Functionality topic within this guide.

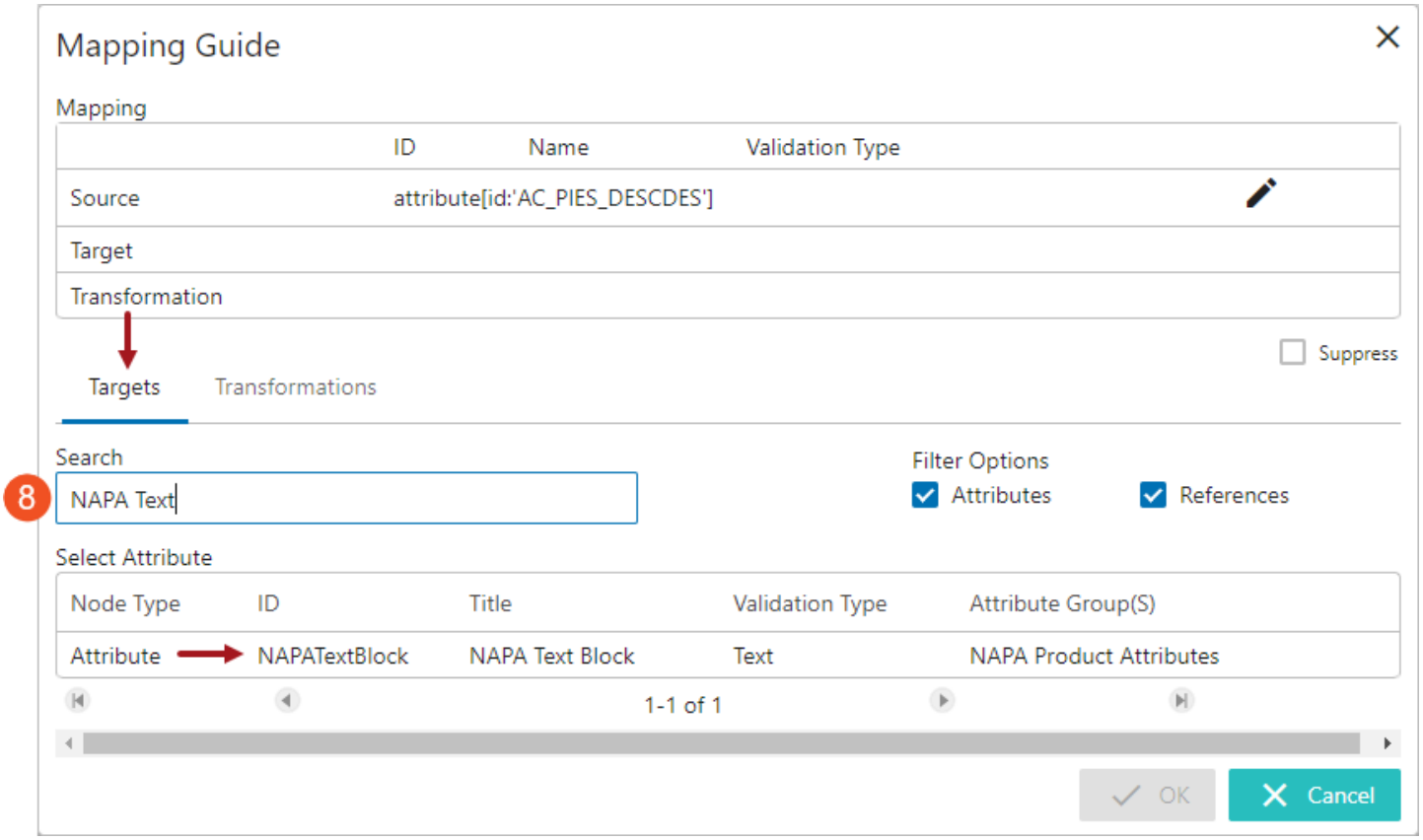
In this example, the selected element 'attribute[id:'[id]']' is configured to retrieve the value from attribute with ID 'AC_PIES_DESCDES'.



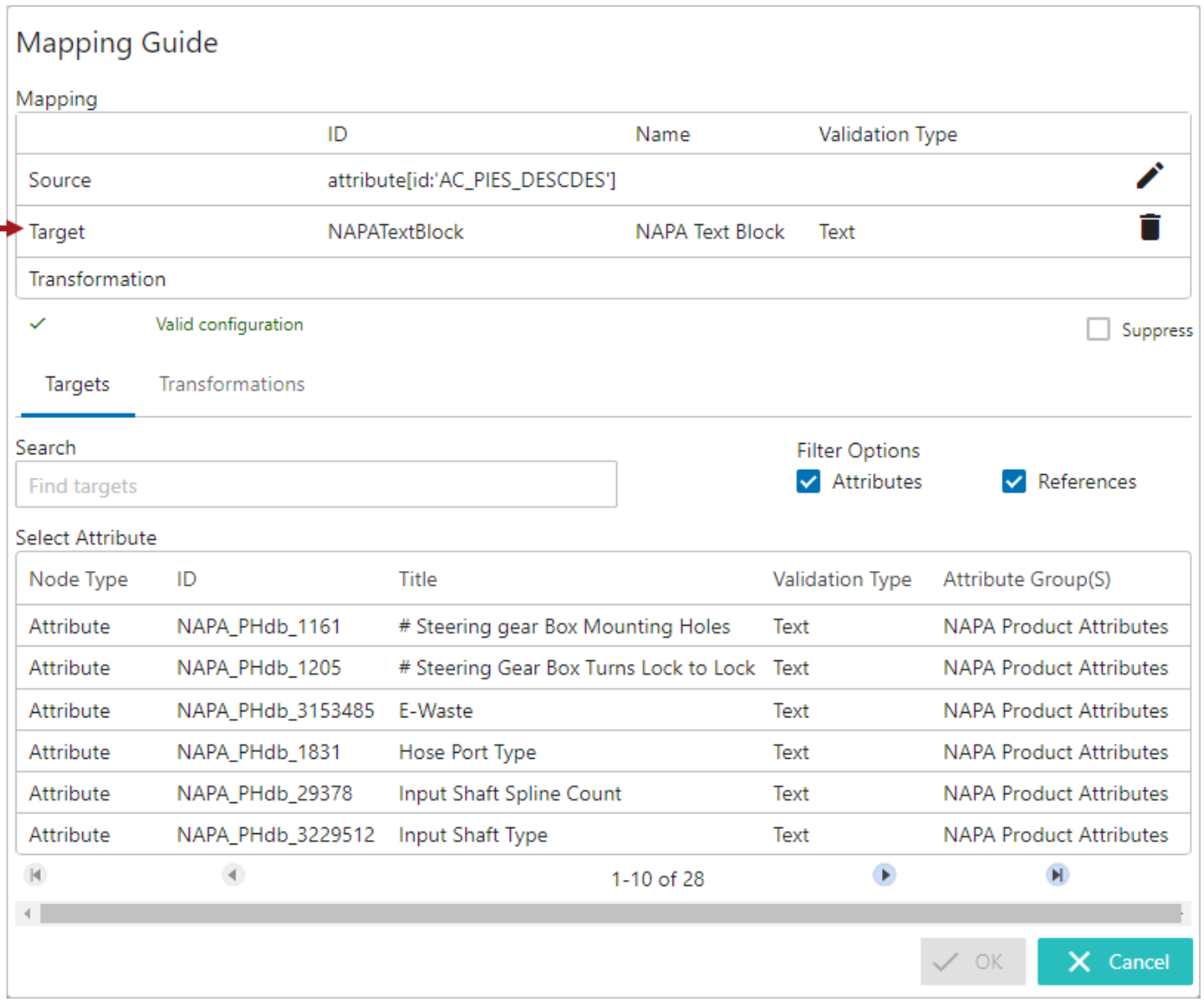

Once selected, the Source STEP Path with the defined data will be populated in the Source field.




8. 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.

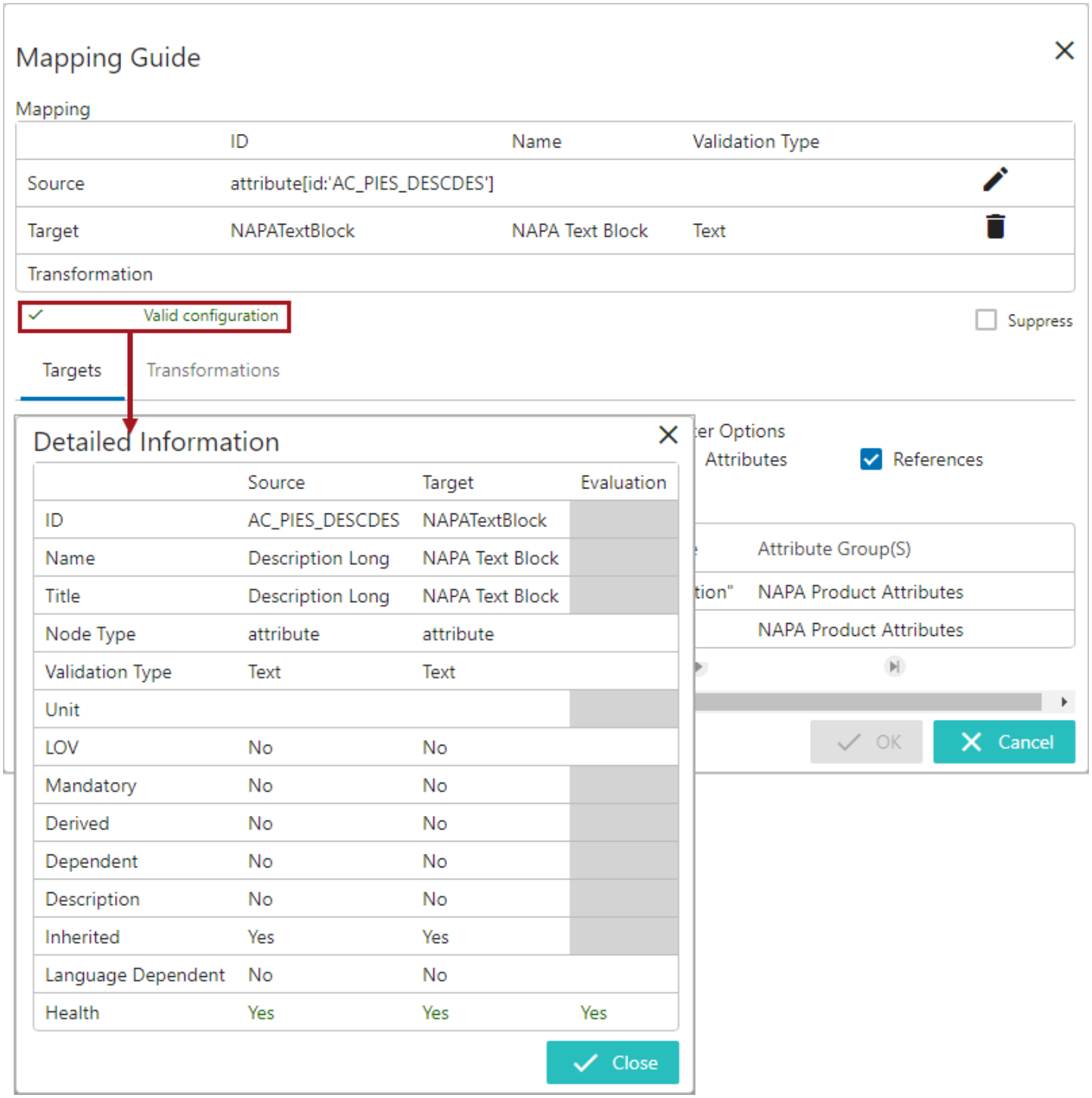



Once selected, the attribute / reference will be populated in the Target field.




**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.

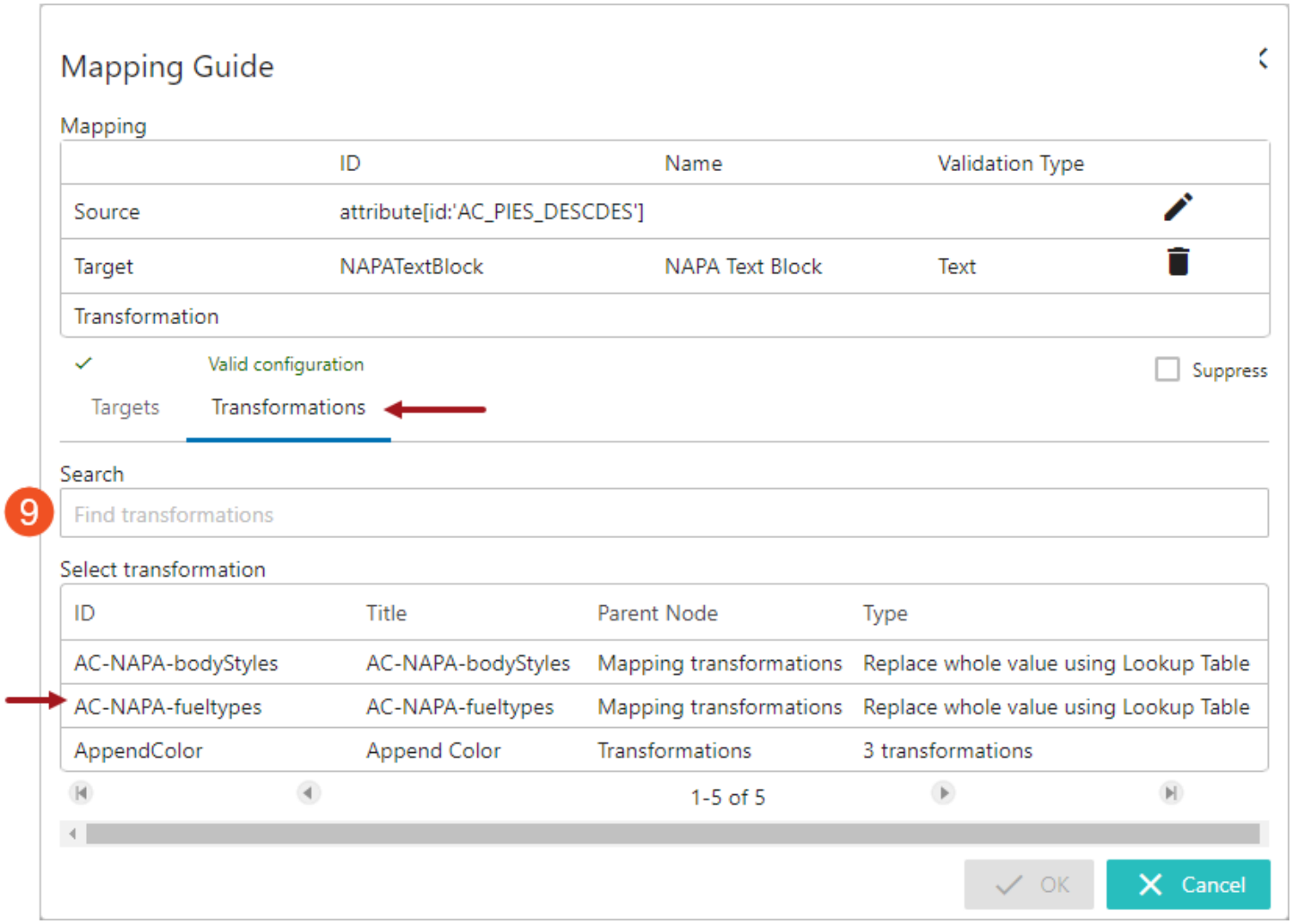



Users can suppress the validation mismatch warning message by clicking on the 'Suppress' checkbox in the Mapping Guide window. Selecting the 'Suppress' checkbox *only* removes the data type mismatch warnings displayed on the Mapping Guide window and does not resolve the mismatch irregularities.

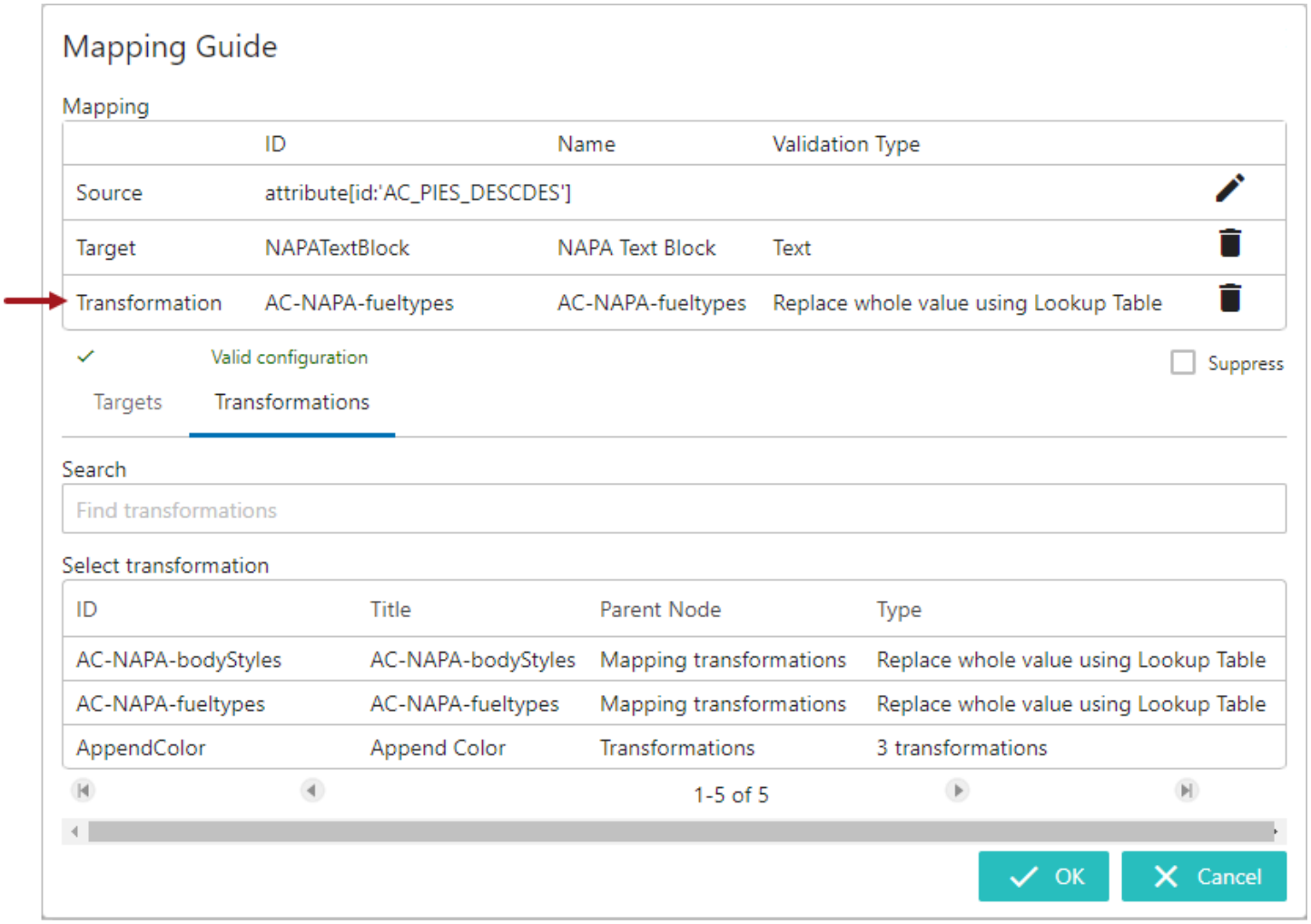
9. With the 'Transformations' tab selected, click in the search field and start typing the initial letters of the transformation name or ID. This brings up a dropdown of typeahead search results listing the attribute

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.

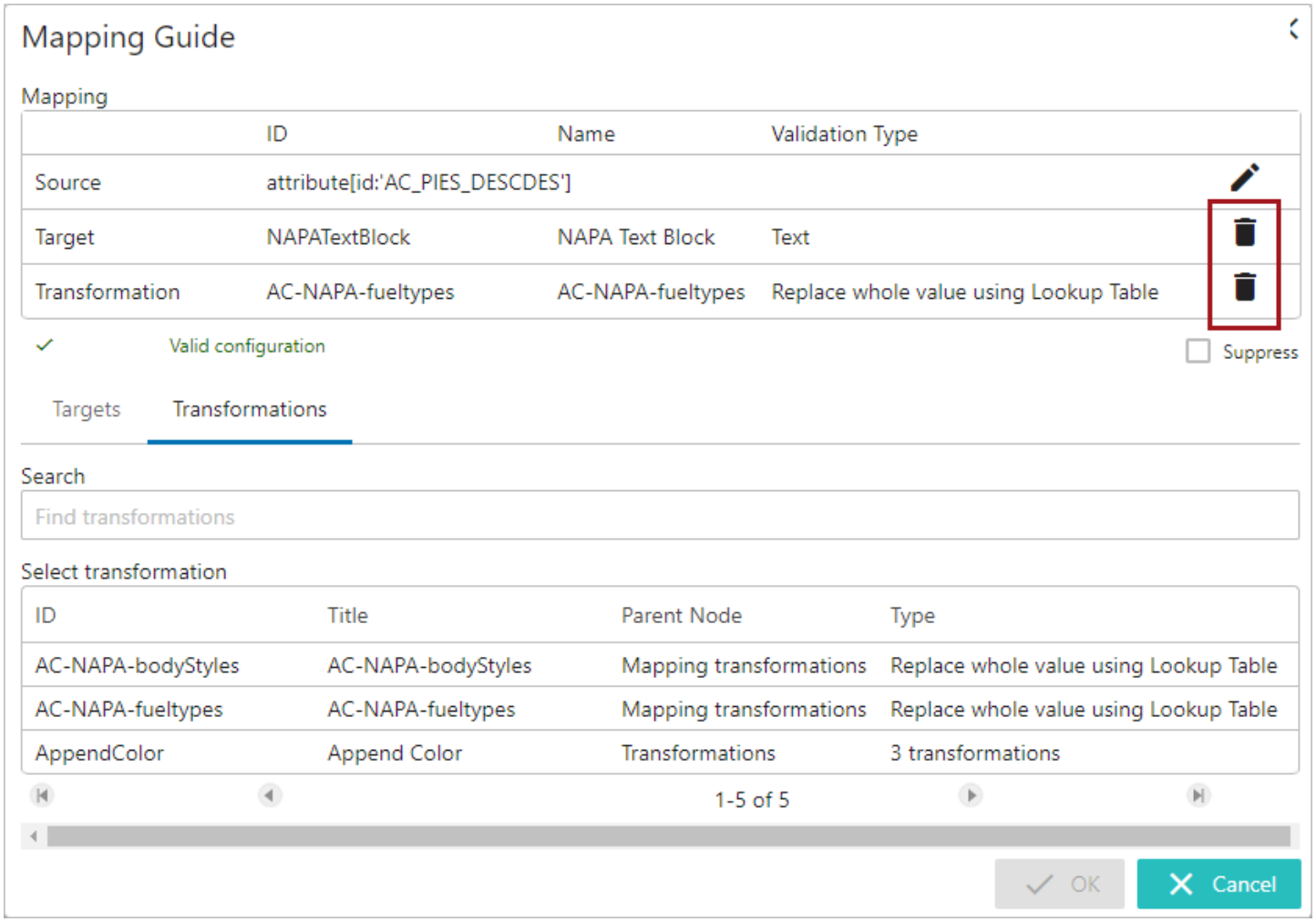



The selected transformation will be populated in the 'Transformation' field (as shown below).



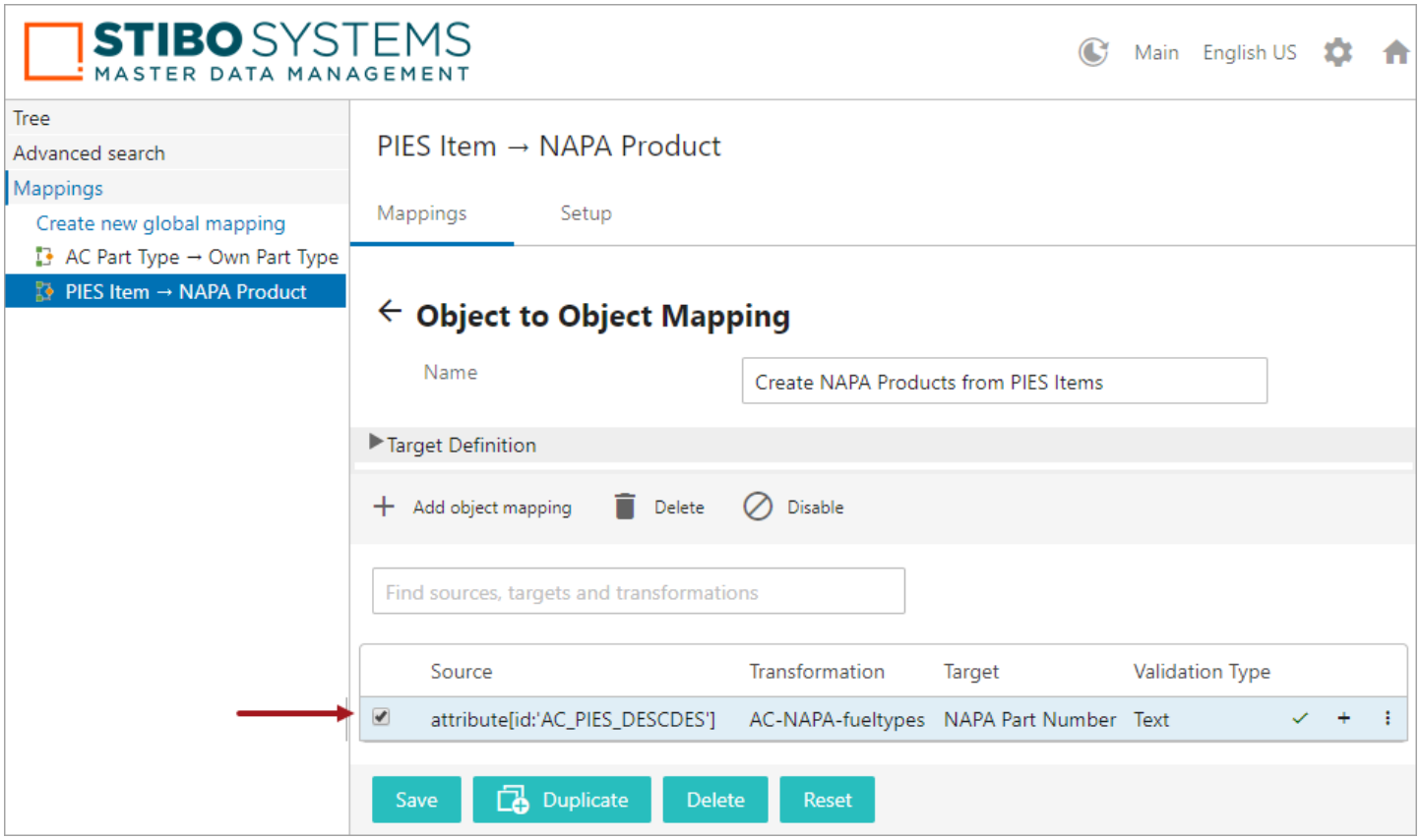

For information on configuring the attribute transformations, refer to the **Attribute Transformations** topic within the **System Setup** section of **STEP Online Help**. And for details on each of the available transformation options, refer to the **Transformations** topic in the **Resource Materials** 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.




10. 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.




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. For more information on handling the mapper rows, refer to topic **Modifying Mapper Rows on the Onboarding Mapping Details Screen** within this guide.

### Mapping Validation Path Functionality

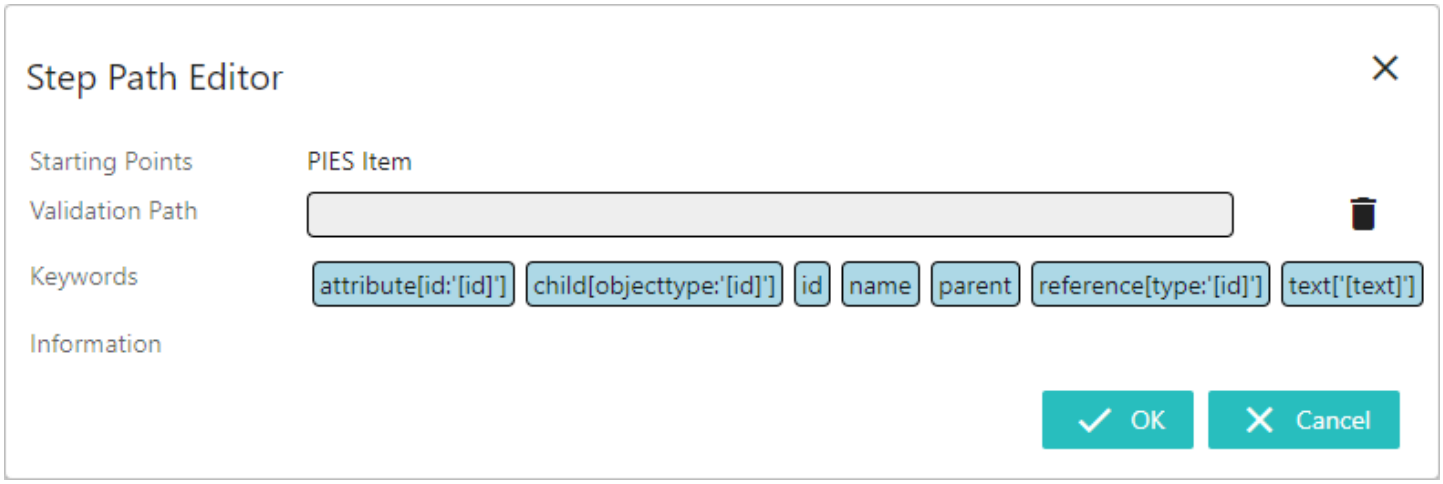
The Mapping Validation Path functionality in the STEP Path Editor dialog ensures that users are able to select valid Source and Target objects to define their mapping setup in Object to Object Mapping or Application Mapping configurations. The premise behind this concept is that when some level of valid 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 helps to retrieve the attribute value(s). 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 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 **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, if the user needs to retrieve the value from ACES Body Num Doors attribute (AC_ ACESBodyNumDoors) stored at the child level object (object type = AC_ACESApplication) 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:

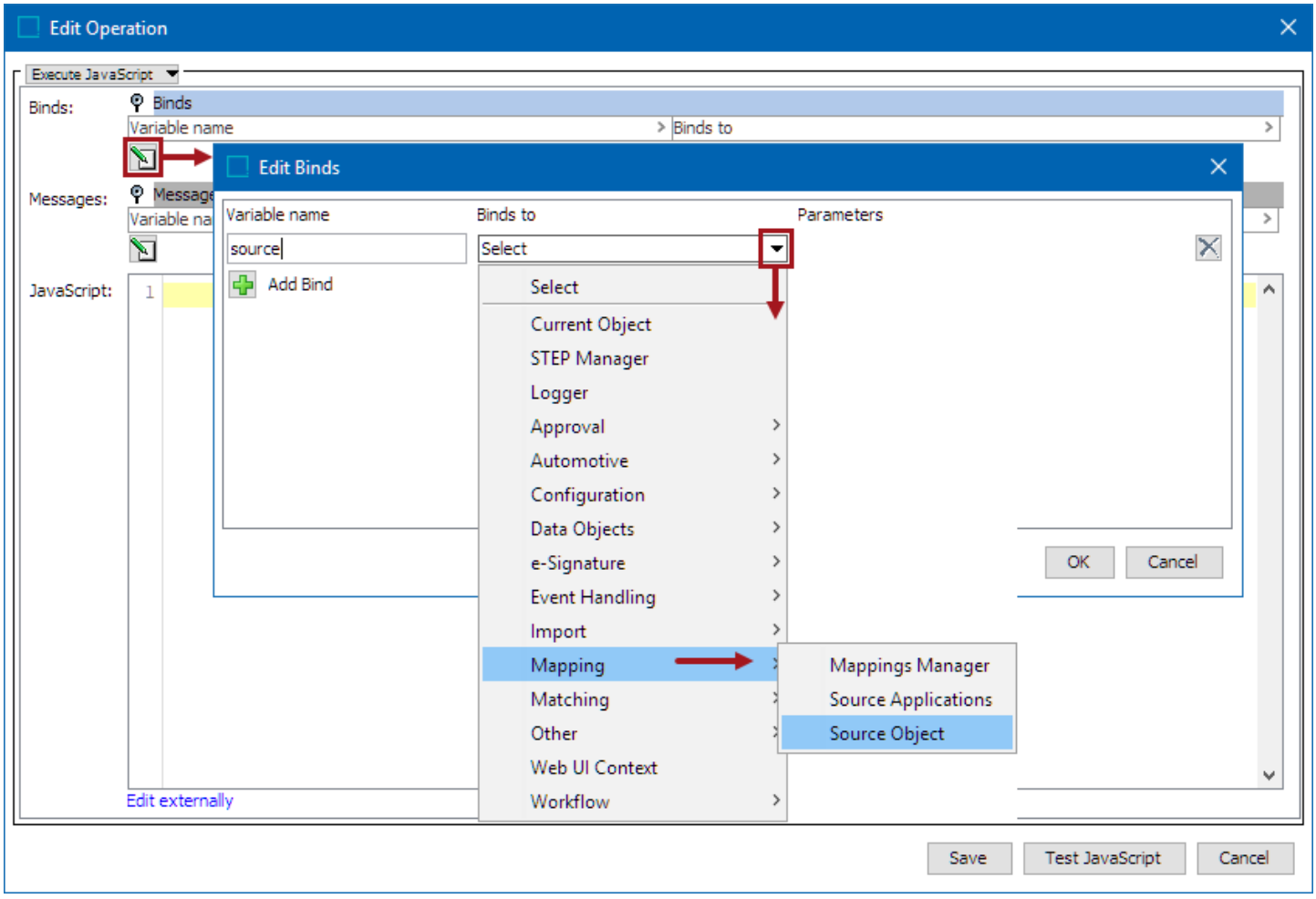
- Identifying the source object.
- Examining all children of the source object with object type AC_ACESApplication.
- Evaluating the value of the ACES Body Num Doors attribute (AC_ACESBodyNumDoors) that is stored on the child object.

## 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 that is copied from a Source object is determined by the business action that is defined in 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, refer to the **Source Object Bind** topic within the **Resource Materials** section of STEP Online Help.

For more information on adding other binds, refer to **JavaScript Binds** in the **Resource Materials** online help.

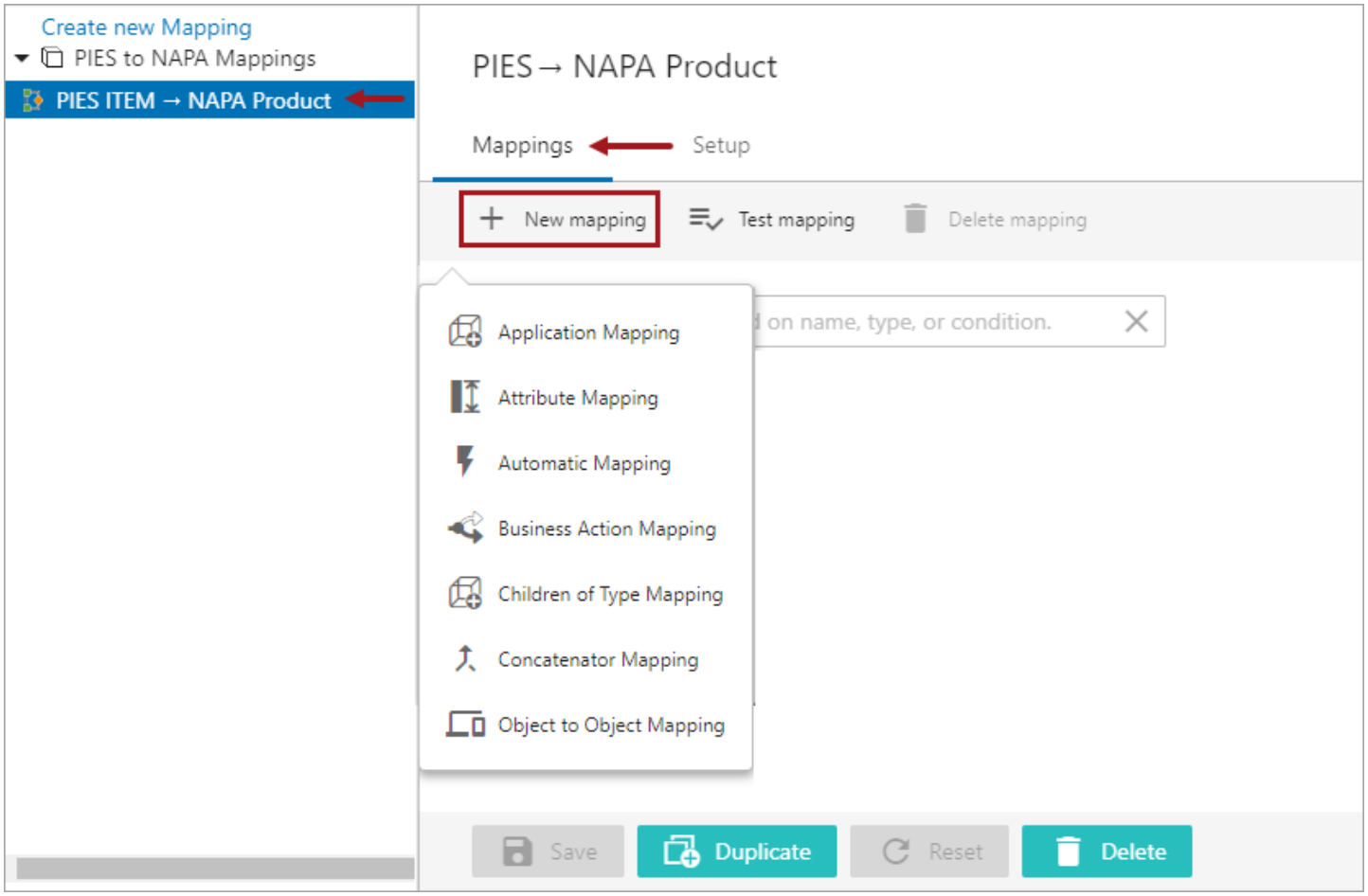
## Configuration

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.

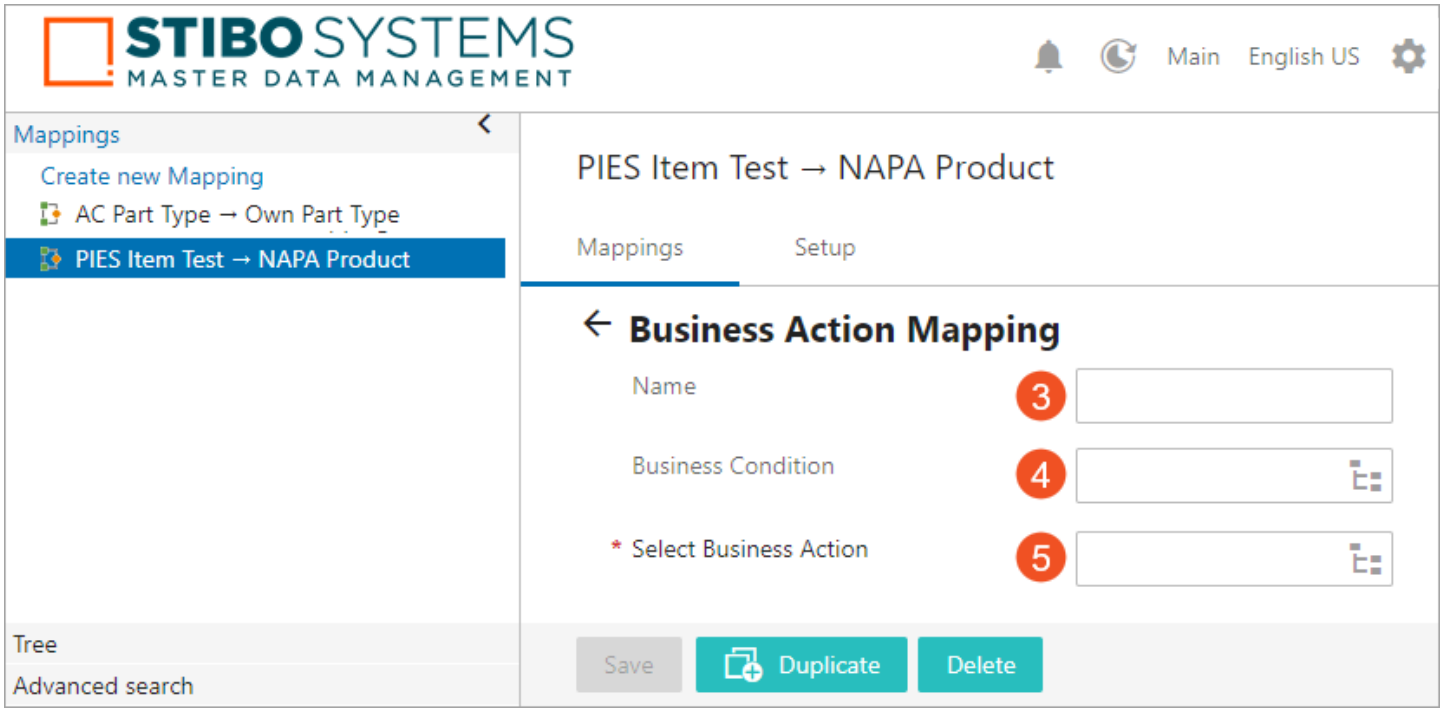
**Note:** A setup entity definition can be exported as comments and submitted to an external source control system for comparison purposes. For details, refer to the **Configuration Management** documentation.

To configure the Business Action Mapping plugin, follow these steps:

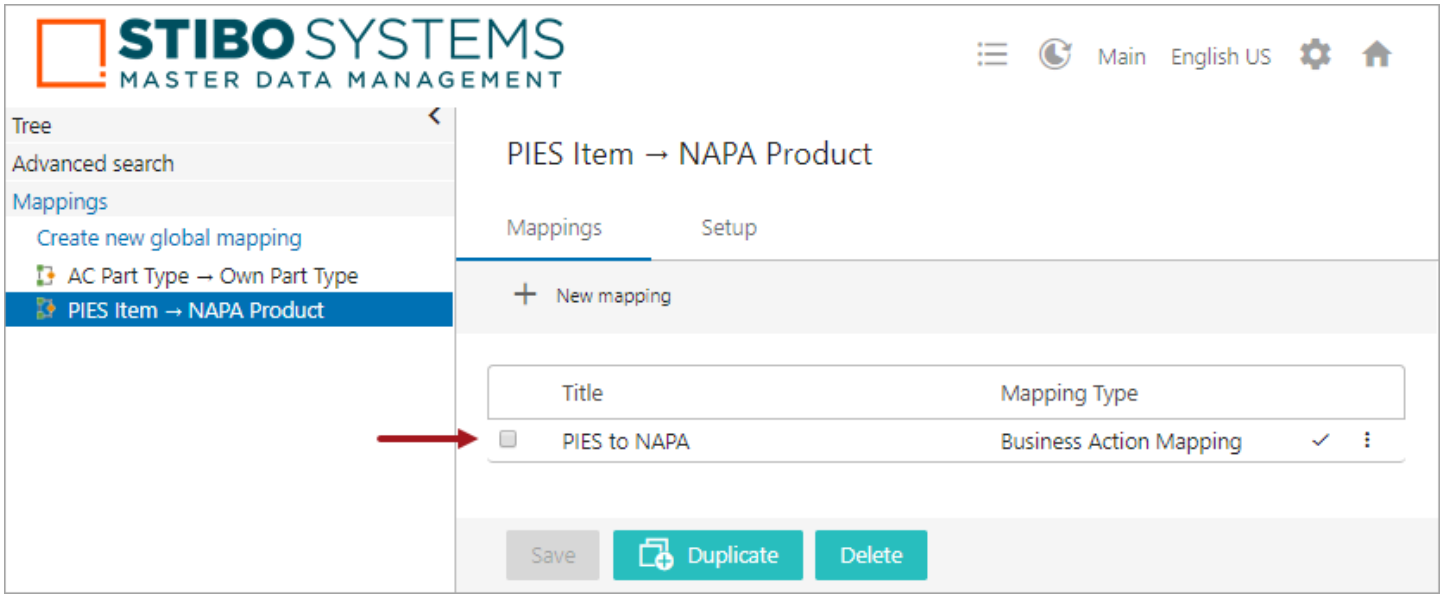
1. With the required Mapper Configuration selected, click on the **New mapping** option that is available within the Mappings tab of the Onboarding Mappings Details screen.




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 optional Business Condition parameter, populate a business condition that defines to run the mapping plugin only on the object if the condition is true.
5. 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.
6. 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.

7. If multiple business actions are required, repeat the steps above to add multiple Business Action Mapping plugins to the Mapper Configuration setup entity.

## Attribute Transformation in Mapper Configuration Setup Entity

Attribute transformations within the onboarding process can be used to transform attribute values, names, and IDs of the target object while onboarding data from the source object to the target object. When the Mapping plugin is defined with the source and the target attributes, users can add an attribute transformation if necessary. As with all transformations, the source object data is not modified, but the target object data is altered by the transformations applied. Attribute transformation can be used to make a set of transformations, and then apply it in the Mapper rows of the Mapping plugins. Attribute transformations can be applied only in the following Mapping plugins:

- Application Mapping plugin
- Attribute Mapping plugin
- Concatenator Mapping
- Object to Object Mapping plugin

Apart from being used in the Mapping plugins, the attribute transformations can be also applied while defining the ID and name for the target object hierarchy that are created through the Target Hierarchy field in the Setup tab of the Onboarding Mappings Details screen.

Users can perform the following functions with regards to the attribute transformations in the Target Hierarchy Editor window and the Mapping Guide window:

- Create a new attribute transformation and define transformation function(s)
- Add, edit, delete, or rearrange the listing order of the transformations for an attribute transformation
- Test the outcome of the added transformation

**Note:** The modifications made to the attribute transformations by accessing it through the Mapping Guide window / Target Hierarchy Editor window is global and affects the attribute transformations residing in System Setup. Meaning, changes made through the Mapping Guide window will have an impact on every place in the system where the changed attribute transformation is applied / used on. Before editing any of the existing attribute transformations, its effect on other places it is being used must be properly assessed.

This topic explains how to access and create / edit attribute transformations in Mapper Configuration setup entity. For information on how to create and configure an attribute transformation in workbench, refer to the **Attribute Transformations** section of the **System Setup** documentation of **STEP Online Help**.

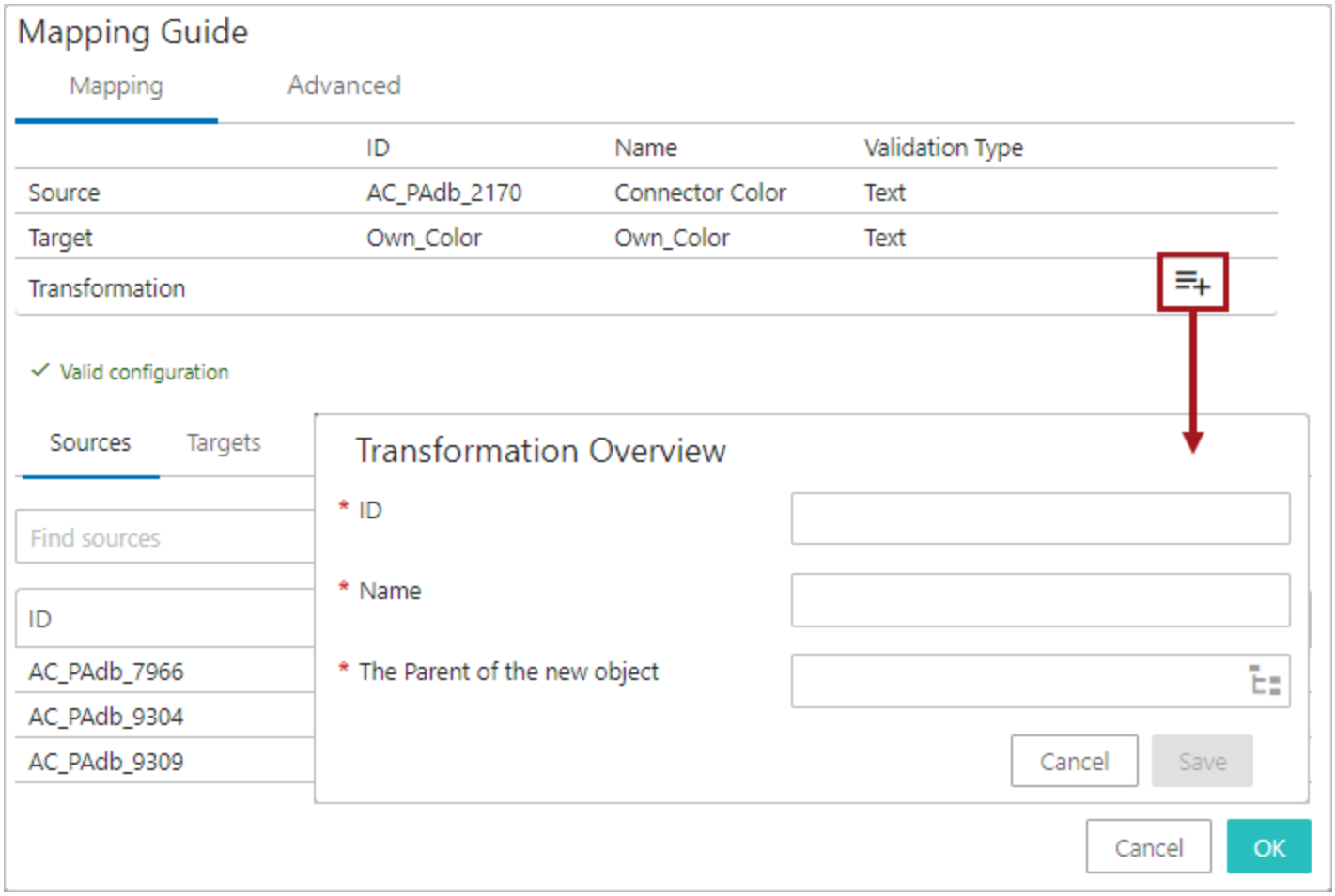
**Note:** A setup entity definition can be exported as comments and submitted to an external source control system for comparison purposes. For details, refer to the **Configuration Management** documentation.

## Creating Attribute Transformations Through Mapping Guide Window

Users can create an attribute transformation and define one or more transformations to it through a Mapping Guide window and the Target Hierarchy Editor window. The created attribute transformation will reside in System Setup under the Attribute Transformations node and will be available to be used in all other places where the attribute transformations can be applied on.

This topic explains how to create an attribute transformation through a Mapping Guide window in the Web UI. For information on how to create and configure an attribute transformation through the Target Hierarchy Editor window, refer to the **Creating Attribute Transformations Through Target Hierarchy Editor Window** topic within this guide. For information on how to create and configure an attribute transformation in workbench, refer to the **Attribute Transformations** section of the **System Setup** documentation of **STEP Online Help**.

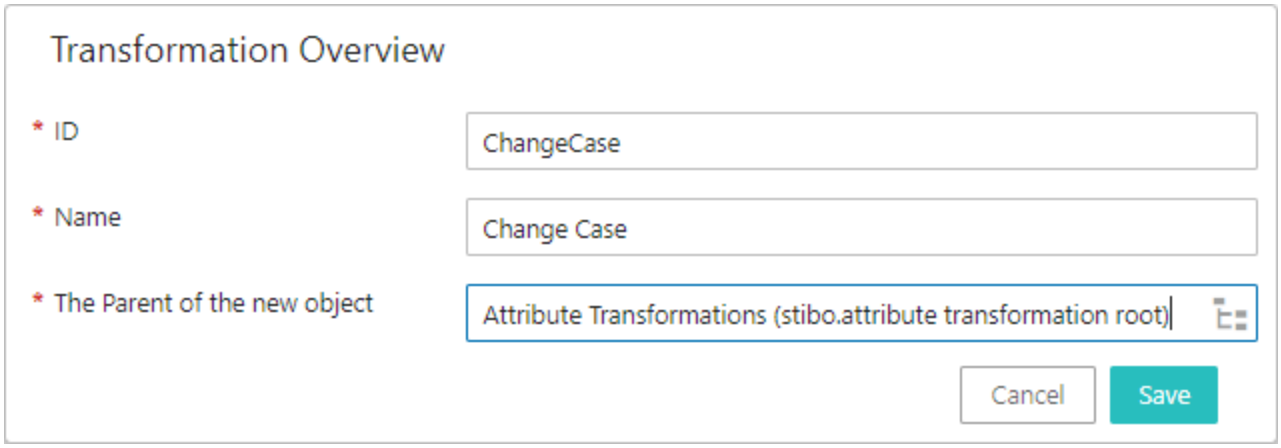
For any Mapping Guide window, if no attribute transformation is applied, then a create button ( ) is displayed within the Transformation field. Clicking the create button( ) will open the Transformation Overview dialog with the parameters explained as follows:



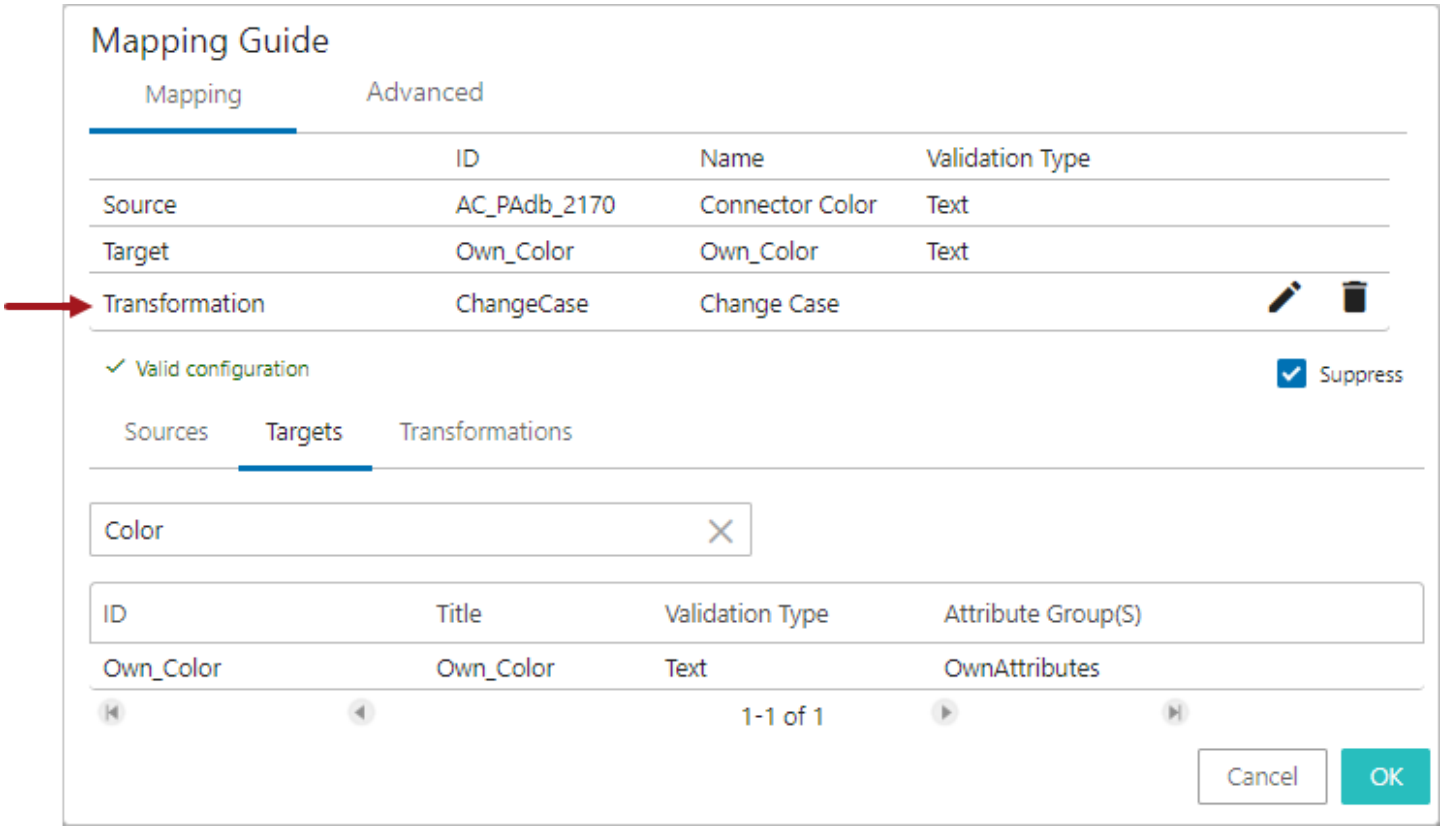

- **ID:** Type an ID for the attribute transformation.
- **Name:** Type a suitable name that describes the attribute transformation functionality.

- **The Parent of the new object:** The user can select the parent node under which the newly created attribute transformation will reside.

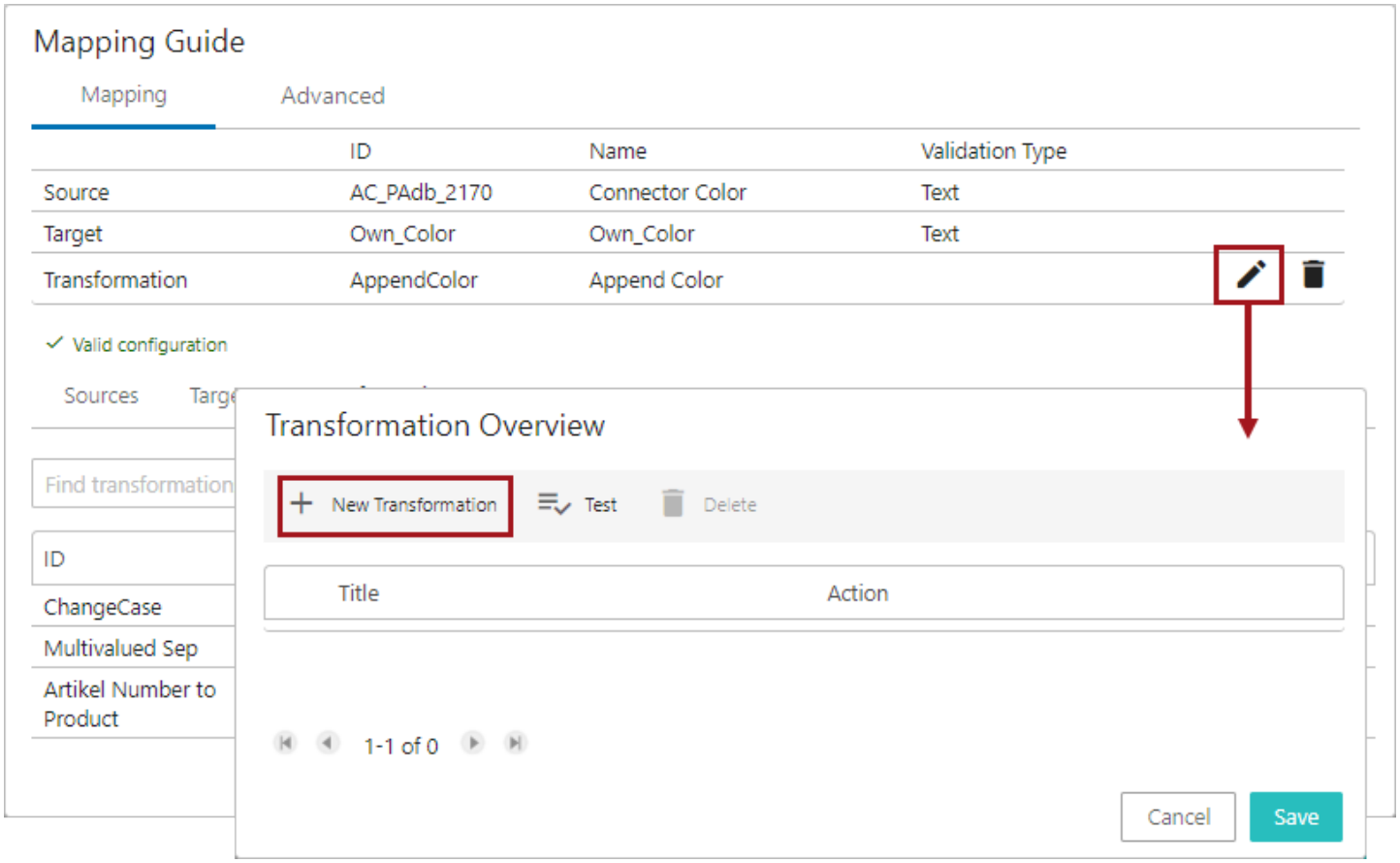
In the example below, an attribute transformation named 'Change Case' (ID=ChangeCase) is created to reside in the Attribute Transformations folder.



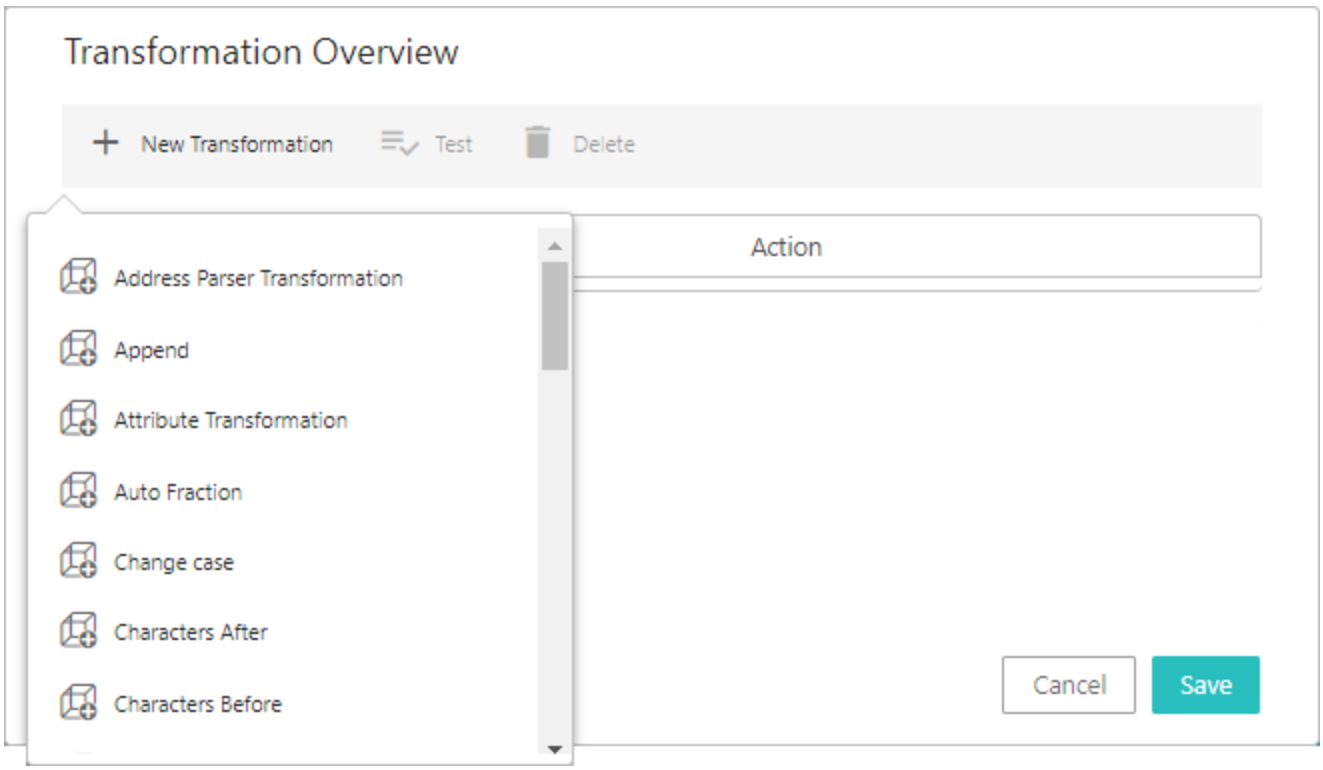

Once all parameters are populated, clicking on the **Save** button will save and close the Transformation Overview dialog, and the newly created attribute transformation will be populated within the Transformation field of the Mapping Guide window. Should changes need to be made, an edit ( ) and a delete ( ) button will be displayed next to the attribute transformation placed in the Transformation field.



The newly created attribute transformation will not have any transformations defined in them. To define one or more transformations, click the edit button ( ) to open the Transformation Overview dialog and add a transformation by clicking the New Transformation button (as shown below).



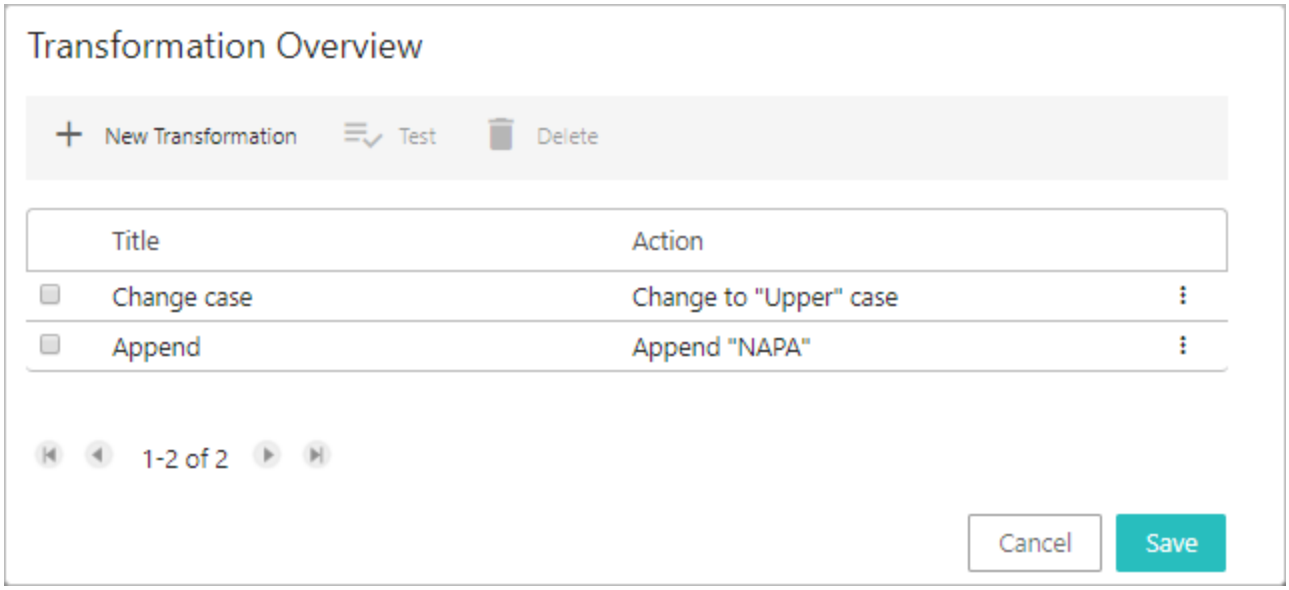

Clicking on **New Transformation** displays the list of rules to be used. Select the relevant transformation.



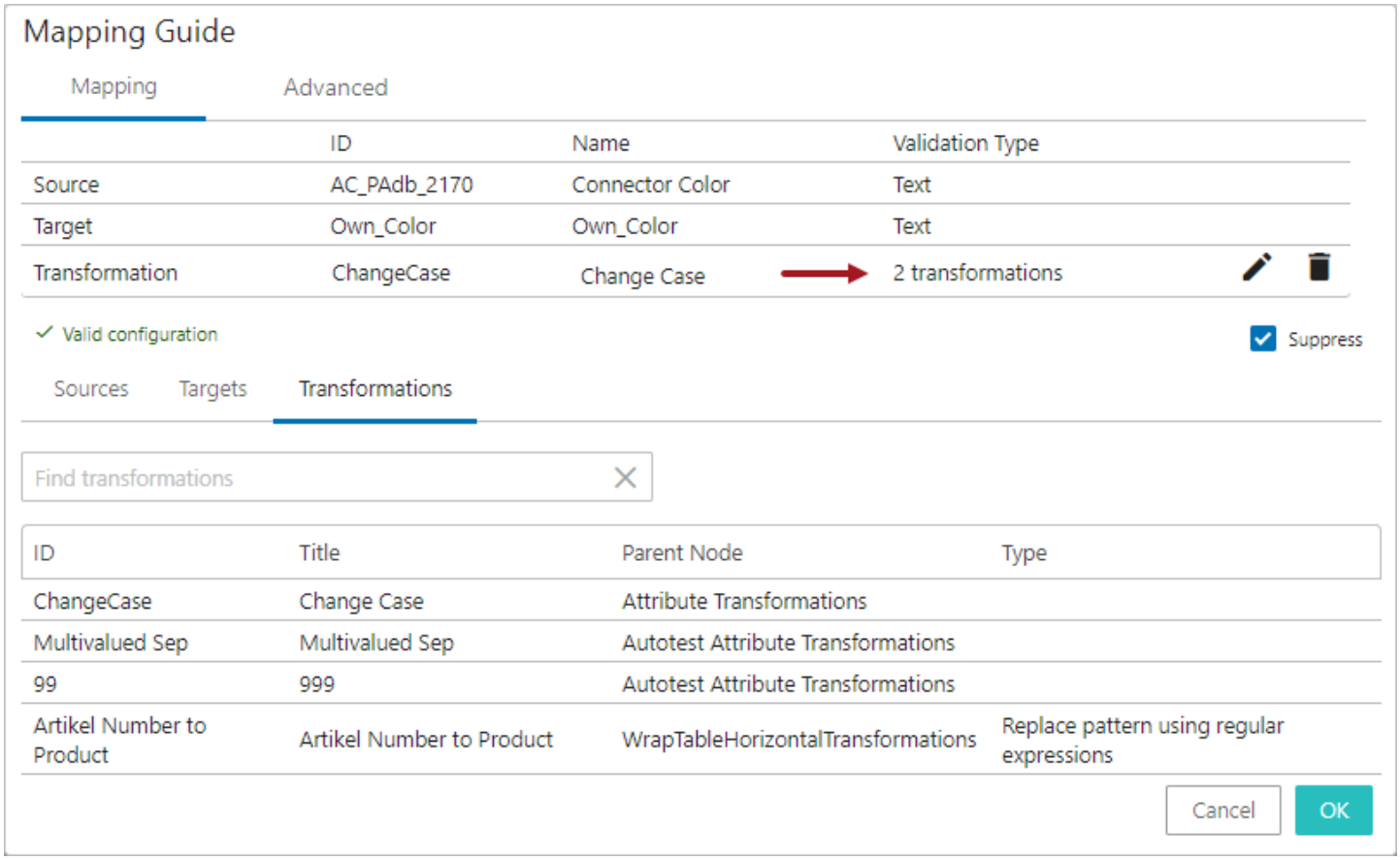

If the selected transformation requires any further configuration, then the **Edit Transformation** dialog will display. Enter the necessary values and click **Save** to save and close the Edit Transformation dialog. The selected transformation will get listed in the Transformation Overview dialog (as shown below). For a description of the available transformations, refer to the **Transformations** topic in the **Resource Materials** online help.

**Note:** It is possible to create multiple sequential transformations as a rule, but the user must to be careful about the order in which the transformations are applied.

In the example below, 'Change case' and 'Append' transformations are used.




Click **Save** to save and close the Transformation Overview dialog. The Validation Type column within the Mapping Guide window will display the total number of transformations available in the selected attribute transformation.



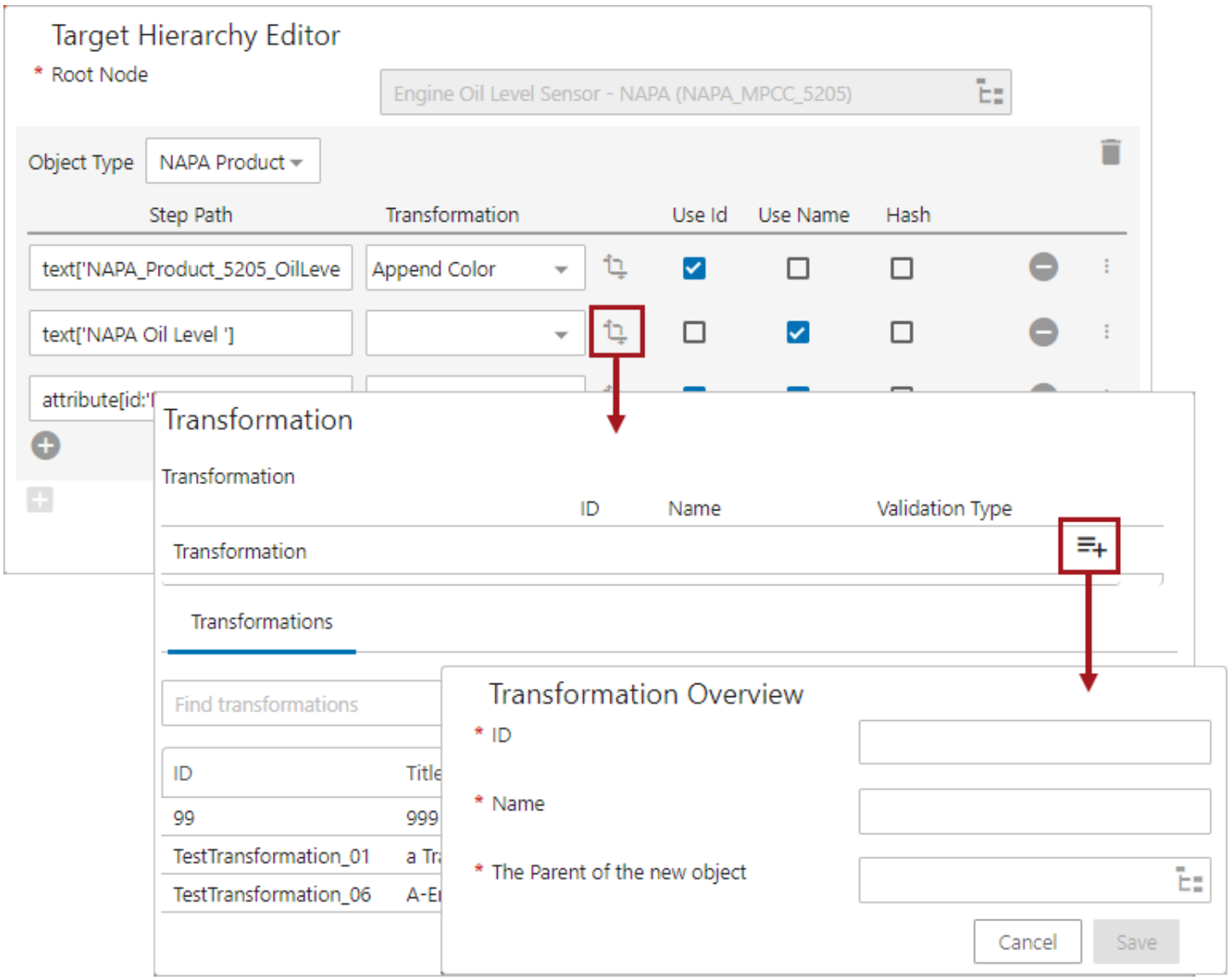
## Creating Attribute Transformations Through Target Hierarchy Editor Window

Users can create an attribute transformation and define one or more transformations to it through a Mapping Guide window and the Target Hierarchy Editor window. The created attribute transformation will reside in System Setup under the Attribute Transformations node and will be available to be used in all other places where the attribute transformations can be applied on.

This topic explains how to create an attribute transformation through a Mapping Guide window in the Web UI. For information on how to create and configure an attribute transformation through the Target Hierarchy Editor window, refer to the **Creating Attribute Transformations Through Mapping Guide Window** topic within this guide. For information on how to create and configure an attribute transformation in workbench, refer to the **Attribute Transformations** section of the **System Setup** documentation of **STEP Online Help**.

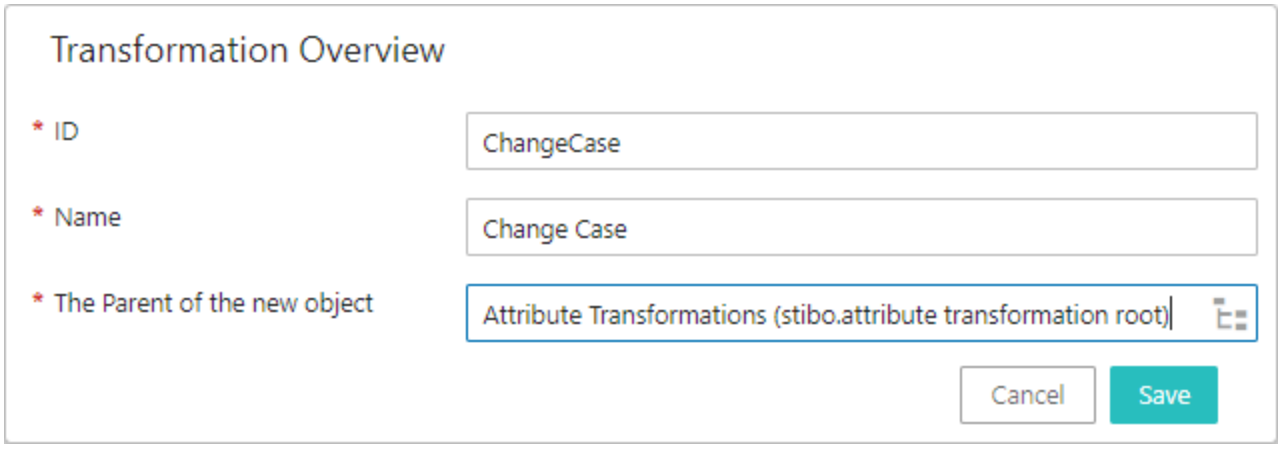
When clicking the Edit icon ( ) that is displayed to the right of the **Target Hierarchy** parameter (available on the Setup tab of the Onboarding Mappings Details Screen), the Target Hierarchy Editor window displays.

Clicking on the 'Select and modify transformation' icon ( ) will open the Transformations dialog. For any Transformation dialog, if no attribute transformation is applied, then a create button ( ) is displayed within the Transformation field. Clicking the create button( ) will open the Transformation Overview dialog with the parameters explained as follows:

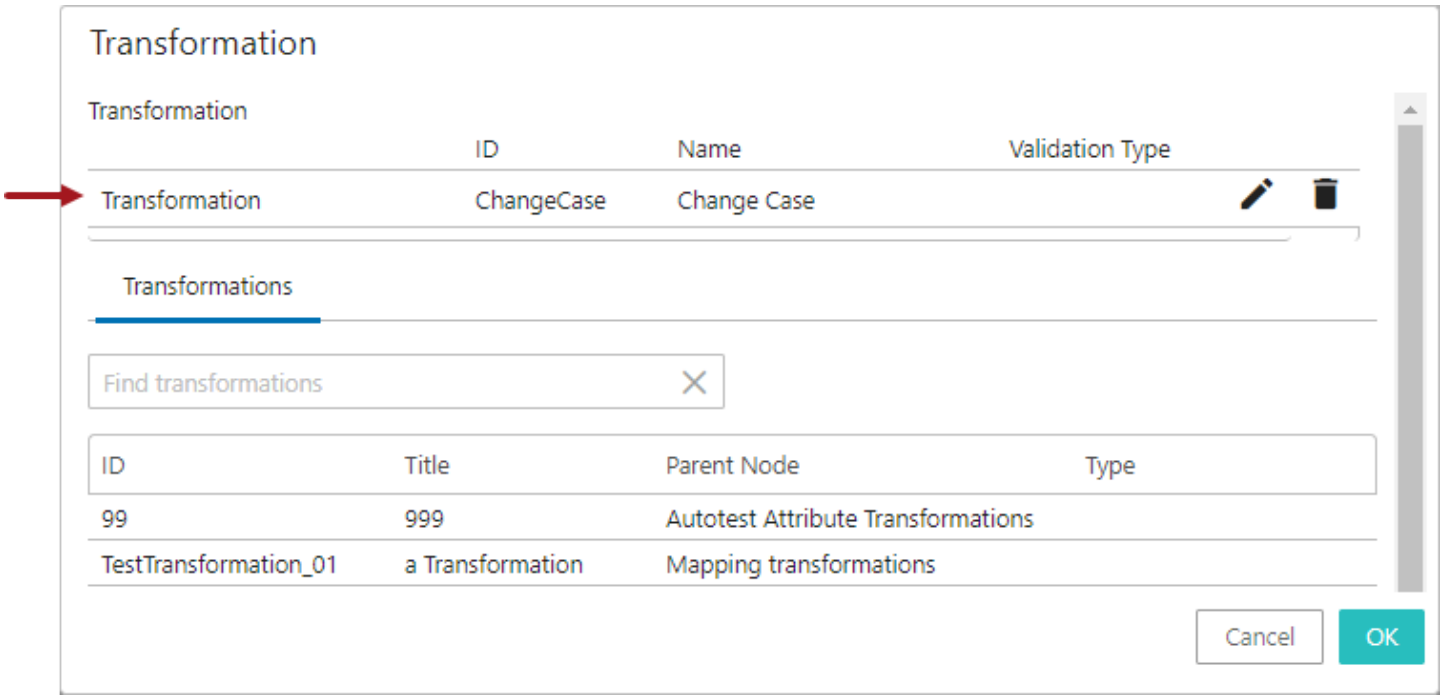



- **ID:** Type an ID for the attribute transformation.
- **Name:** Type a suitable name that describes the attribute transformation functionality.
- **The Parent of the new object:** The user can select the parent node under which the newly created attribute transformation will reside.

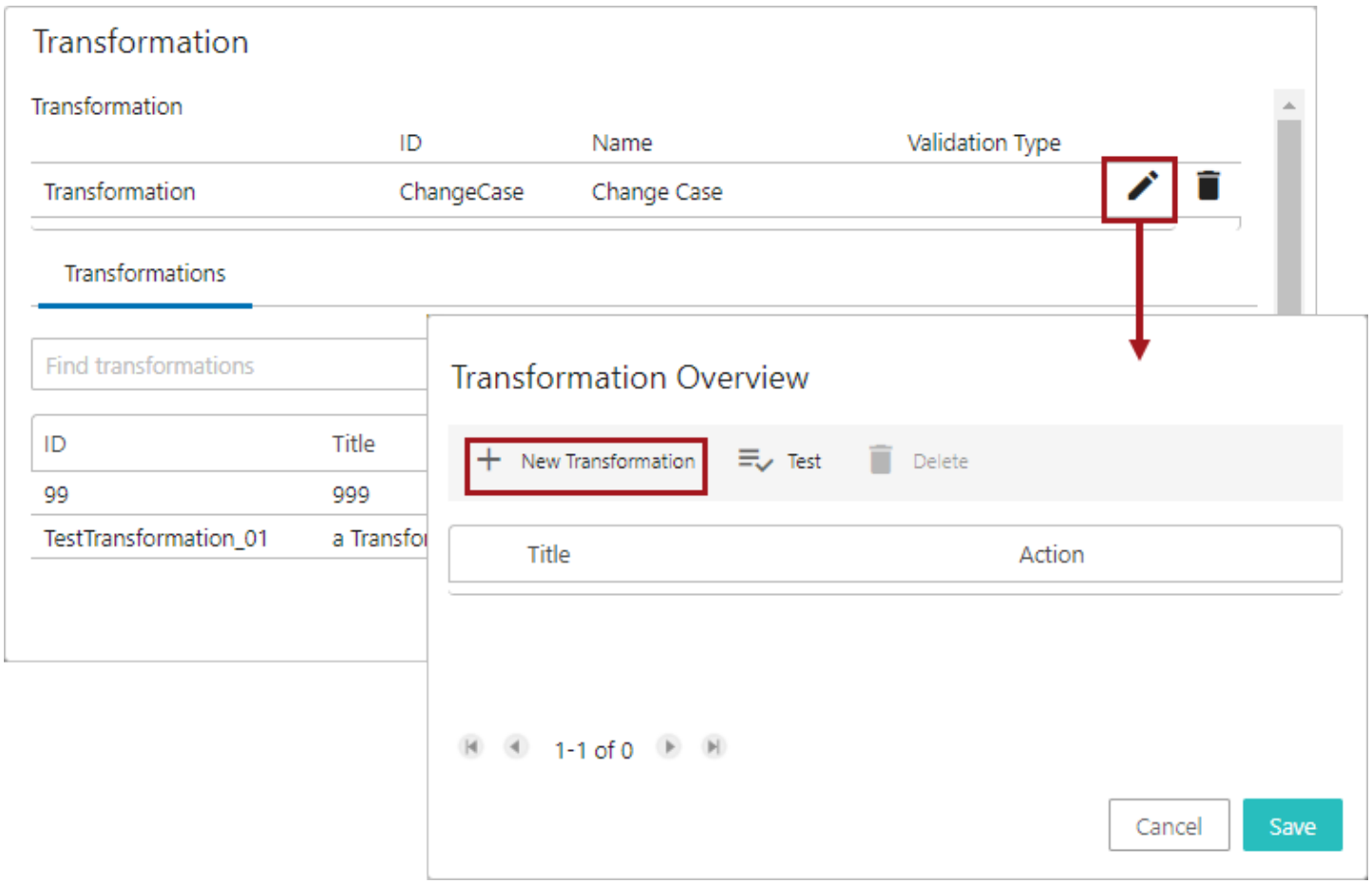
In the example below, an attribute transformation named 'Change Case' (ID=ChangeCase) is created to reside in the Attribute Transformations folder.



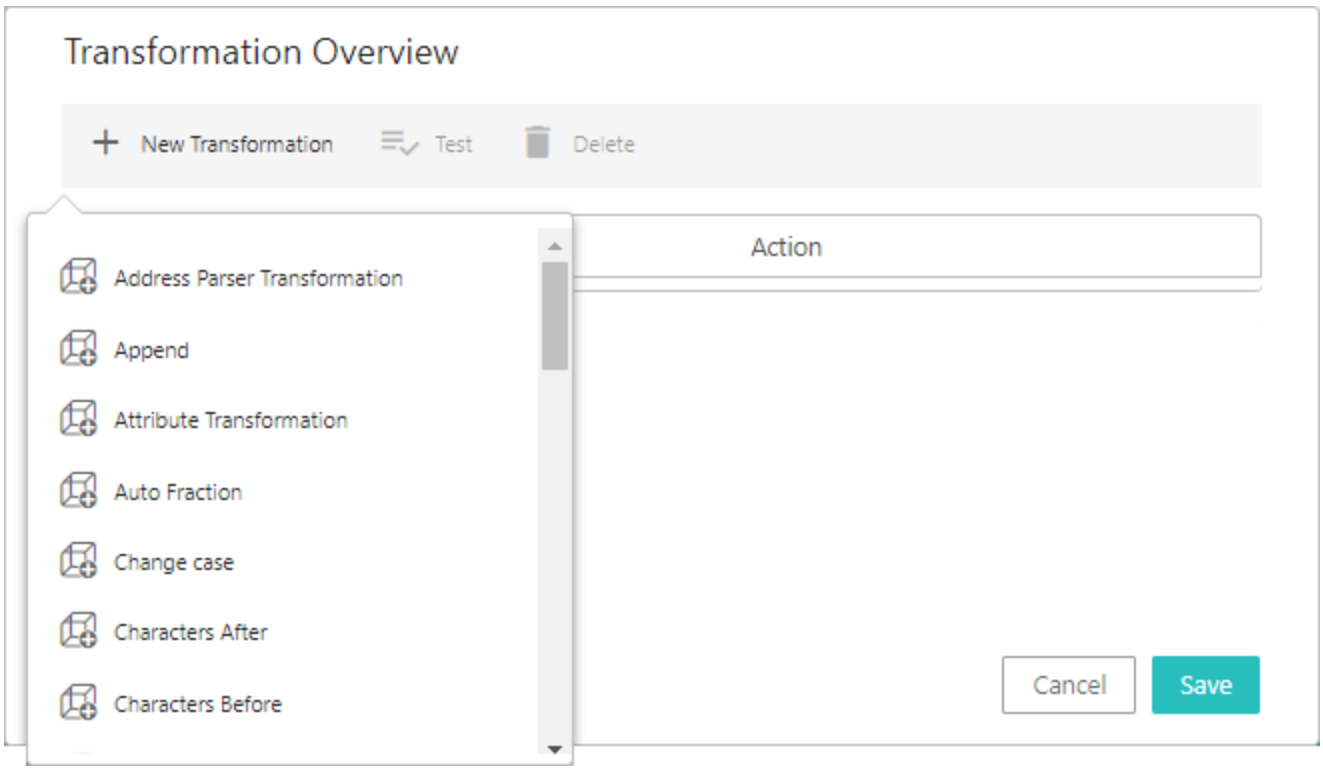

Once all parameters are populated, clicking on the **Save** button will save and close the Transformation Overview dialog, and the newly created attribute transformation will be populated within the Transformation field of the Transformation window. Should changes need to be made, an edit ( ) and a delete ( ) button will be displayed next to the attribute transformation placed in the Transformation field.




The newly created attribute transformation will not have any transformations defined in them. To define one or more transformations, click the edit button ( ) to open the Transformation Overview dialog and add a transformation by clicking the New Transformation button (as shown below).



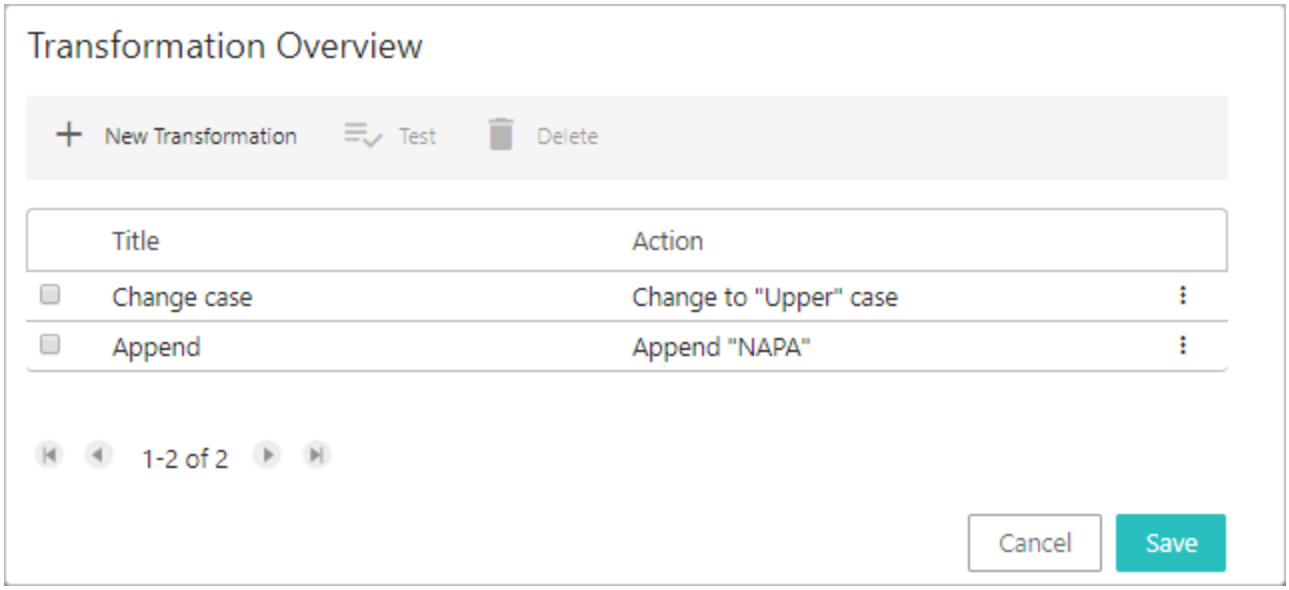

Clicking on **New Transformation** displays the list of rules to be used. Select the relevant transformation.



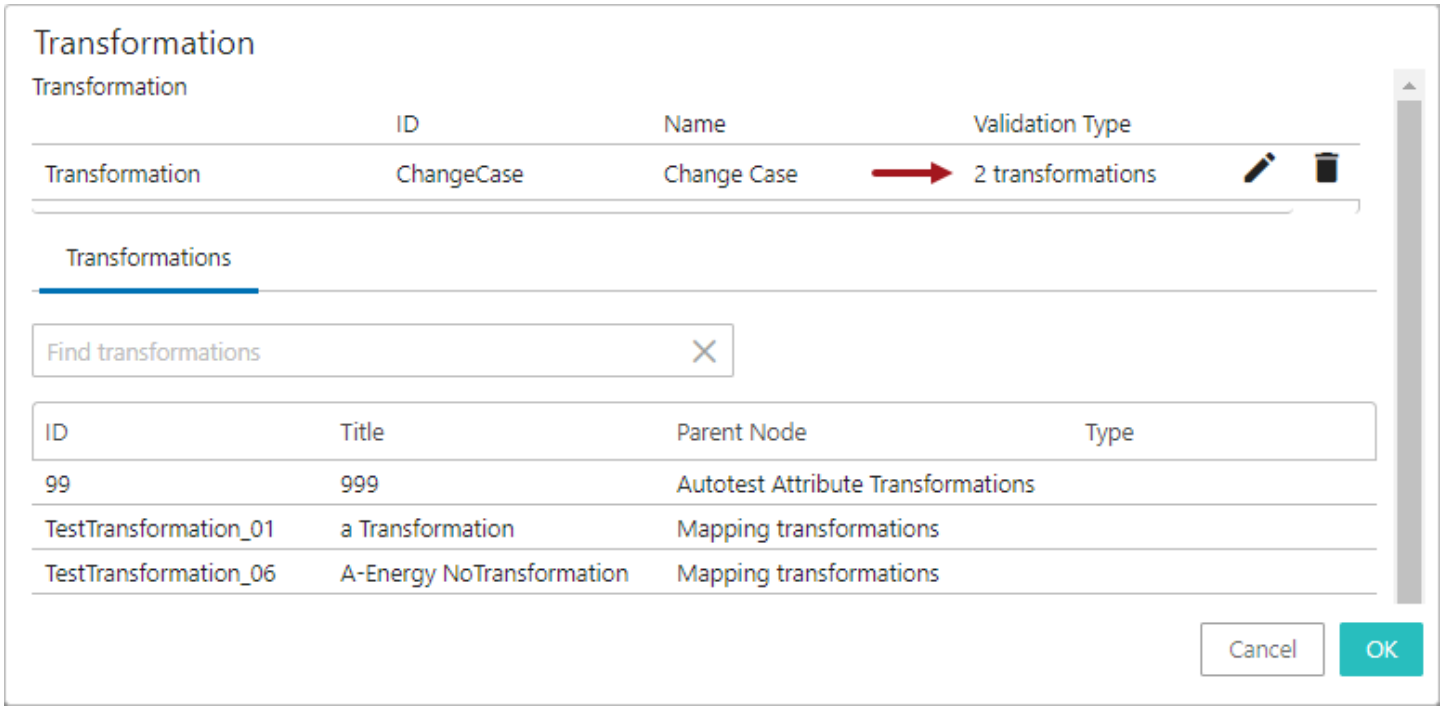

If the selected transformation requires any further configuration, then the **Edit Transformation** dialog will display. Enter the necessary values and click **Save** to save and close the Edit Transformation dialog. The selected transformation will get listed in the Transformation Overview dialog (as shown below). For a description of the available transformations, refer to the **Transformations** topic in the **Resource Materials** online help.

**Note:** It is possible to create multiple sequential transformations as a rule, but the user must to be careful about the order in which the transformations are applied.

In the example below, 'Change case' and 'Append' transformations are used.




Click **Save** to save and close the Transformation Overview dialog. The Validation Type column within the Mapping Guide window will display the total number of transformations available in the selected attribute transformation.

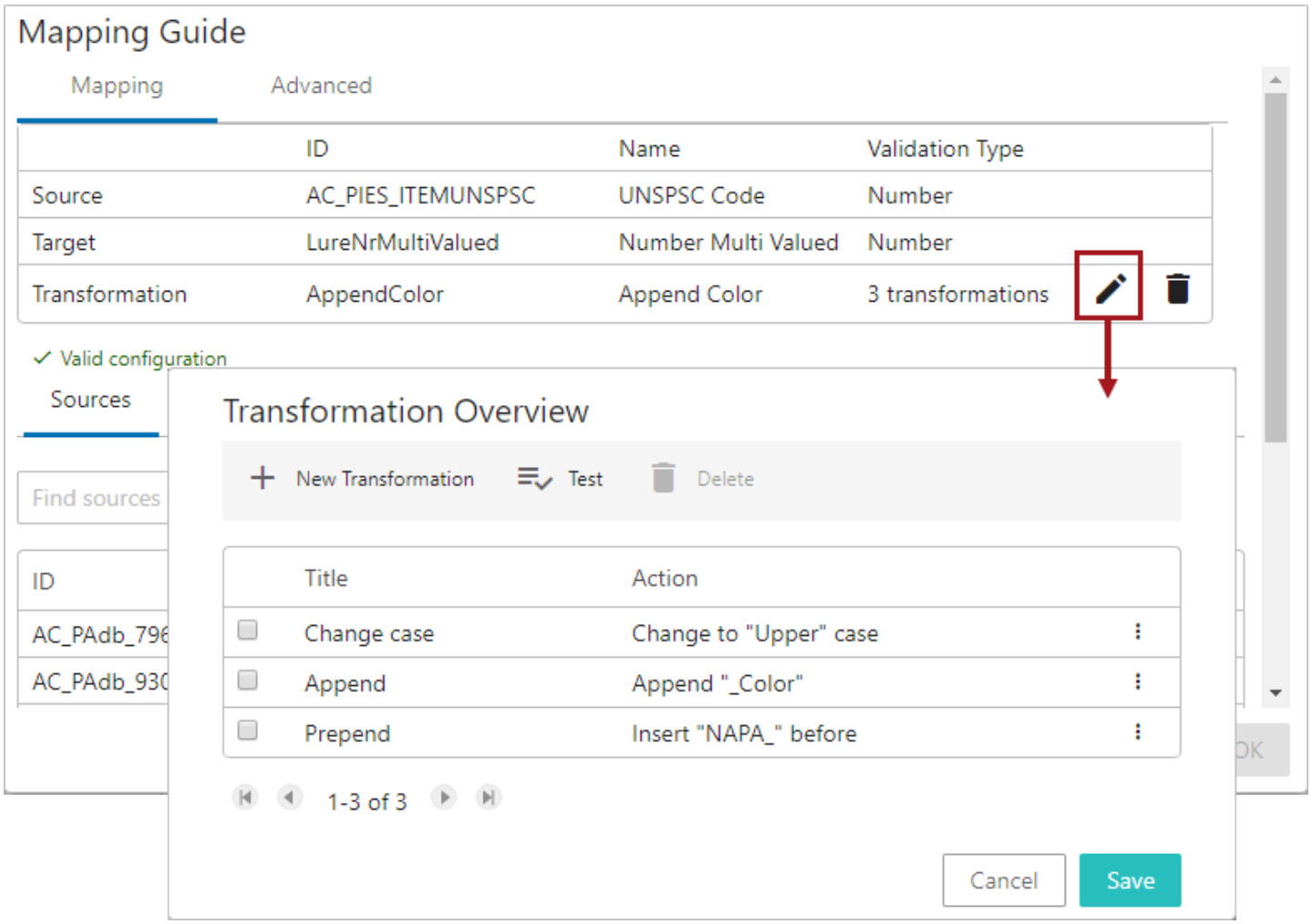


## Modifying Transformations on the Transformation Overview Dialog

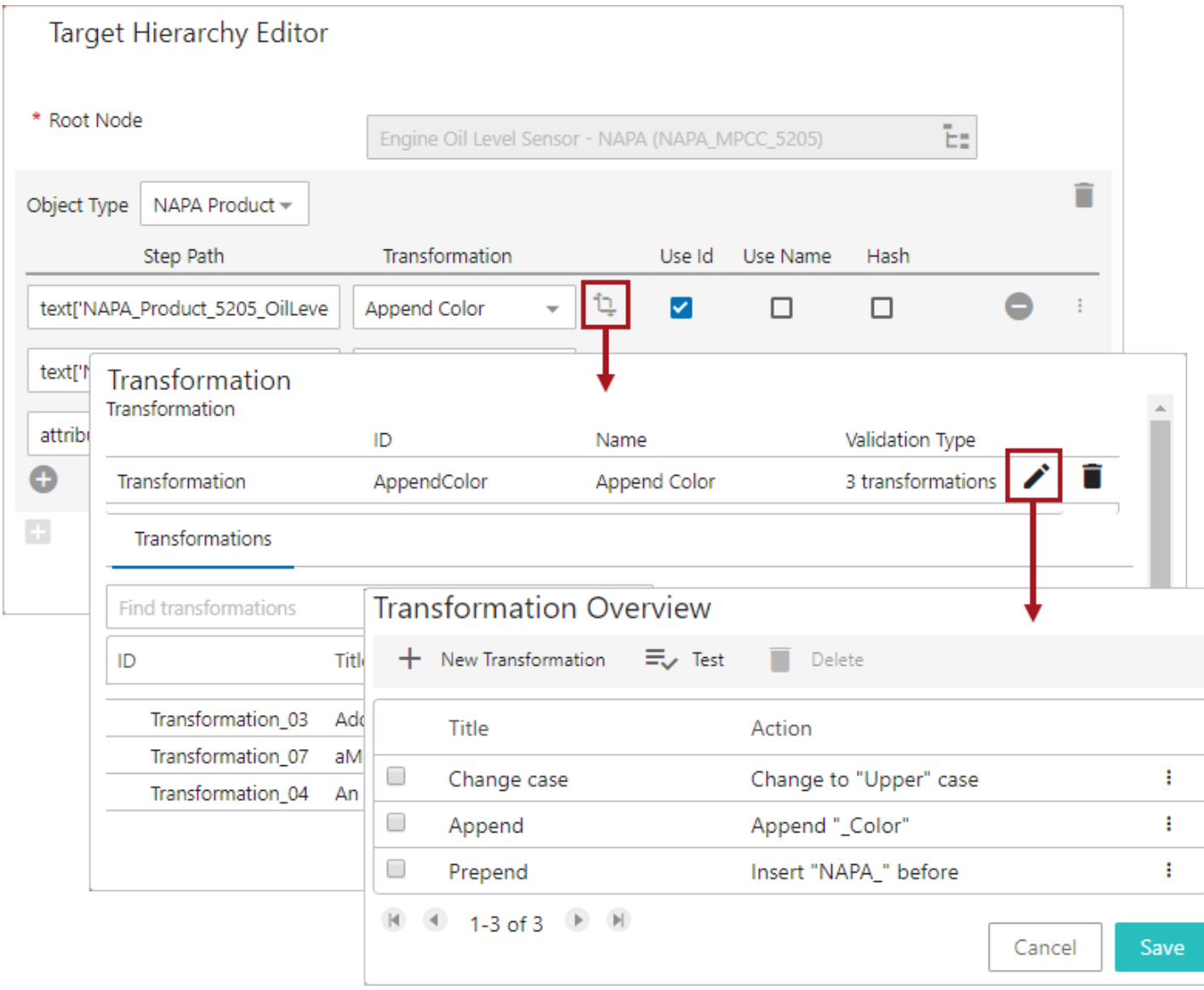
On the Transformation Overview dialog, the user has the flexibility to add, edit, delete, or rearrange the listing order of the transformations. When there are multiple transformations listed, the order of execution of each transformation is based on the order in which it is listed within the Transformation Overview dialog.

Following are the two ways that the Transformation Overview dialog can be accessed in the Web UI:

- Through the Mapping Guide window by clicking the edit button ( ) for the attribute transformation.

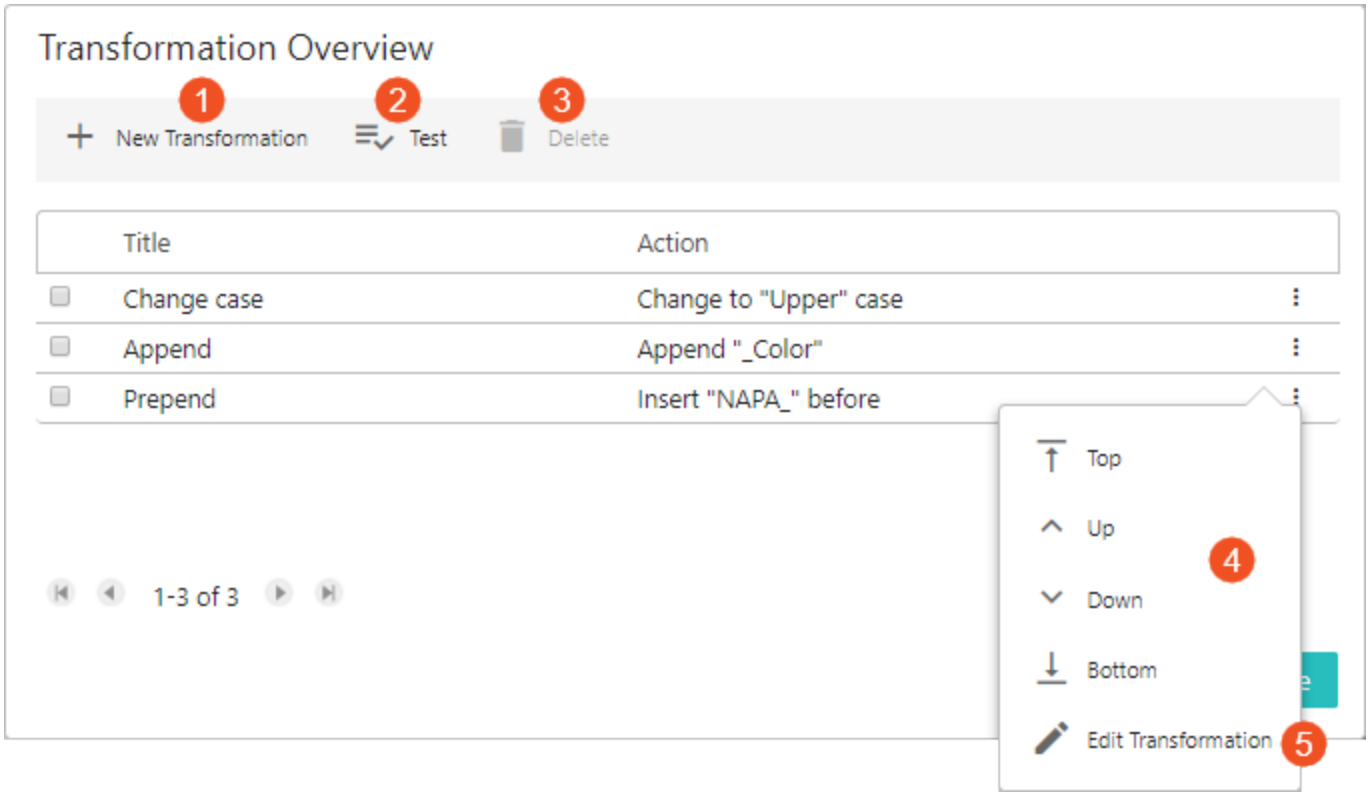



- Through the Transformation Hierarchy Editor window by clicking the 'Select and modify transformation' icon ( ), and then clicking the edit button ( ) for the attribute transformation.




In the following image, the various functionalities that are available in the Transformation Overview dialog are numbered.

In the image below, the attribute transformation is configured to include multiple transformations. Each of the transformation is displayed as rows.

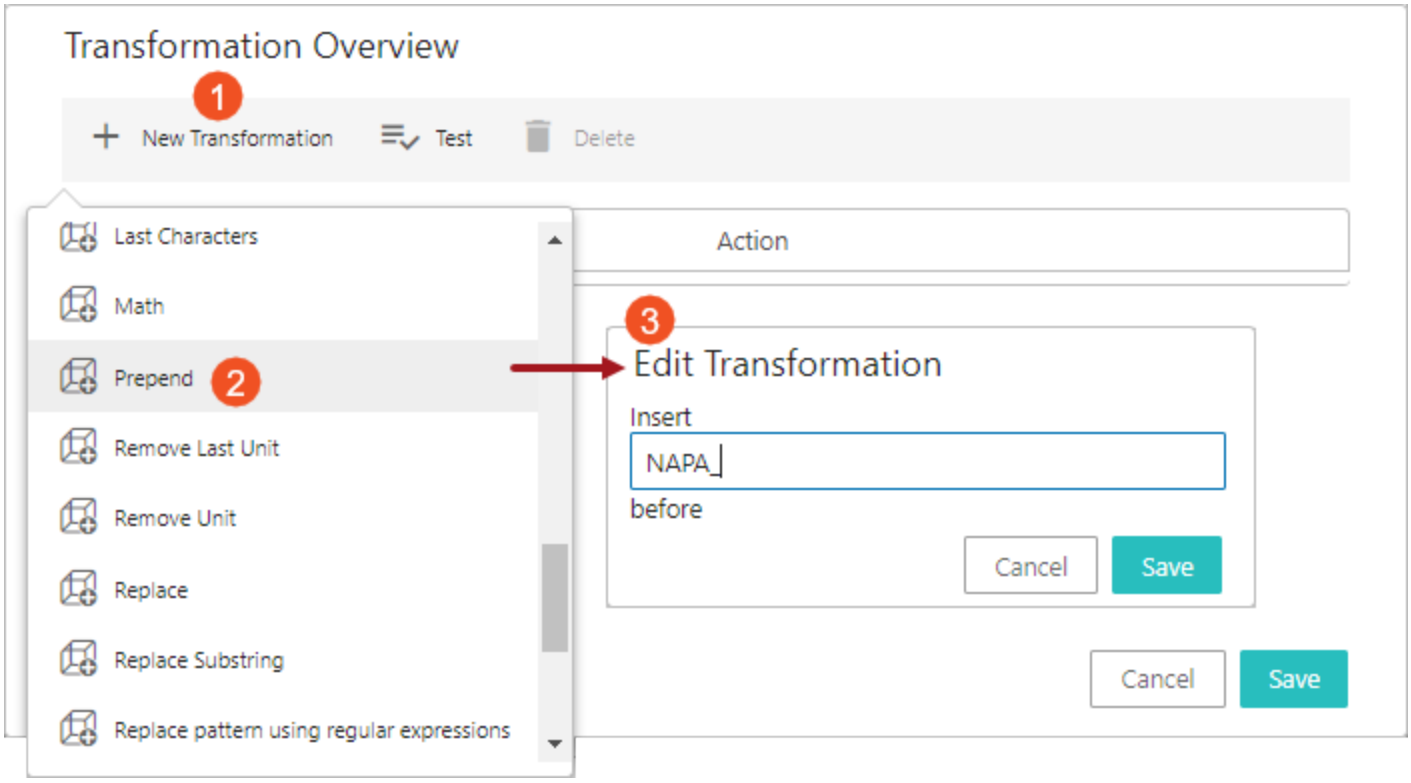



The available modifications defined below include:

1. Add a new transformation to the attribute transformation
2. Test the result of the attribute transformation
3. Delete an existing transformation
4. Change the listing order of the transformation
5. Edit an existing transformation

### Add a transformation

Adding a set of transformations to the attribute transformation will make a rule, and when applied in the Mapping plugin, it will affect the Target object name / ID / attribute value as defined within each of the transformations.




1. Click on the **New Transformation** button to display the list of transformations available.
2. Select the relevant Transformation. If the selected Transformation requires any further configuration, then the **Edit Transformation** dialog will display.
3. Define the transformation and click **Save** to save and close the Edit Transformation dialog.

The selected transformation is listed in the Transformation Overview dialog. For a description of the available transformations, refer to the **Transformations** topic in the **Resource Materials** online help.

### Test an attribute transformation

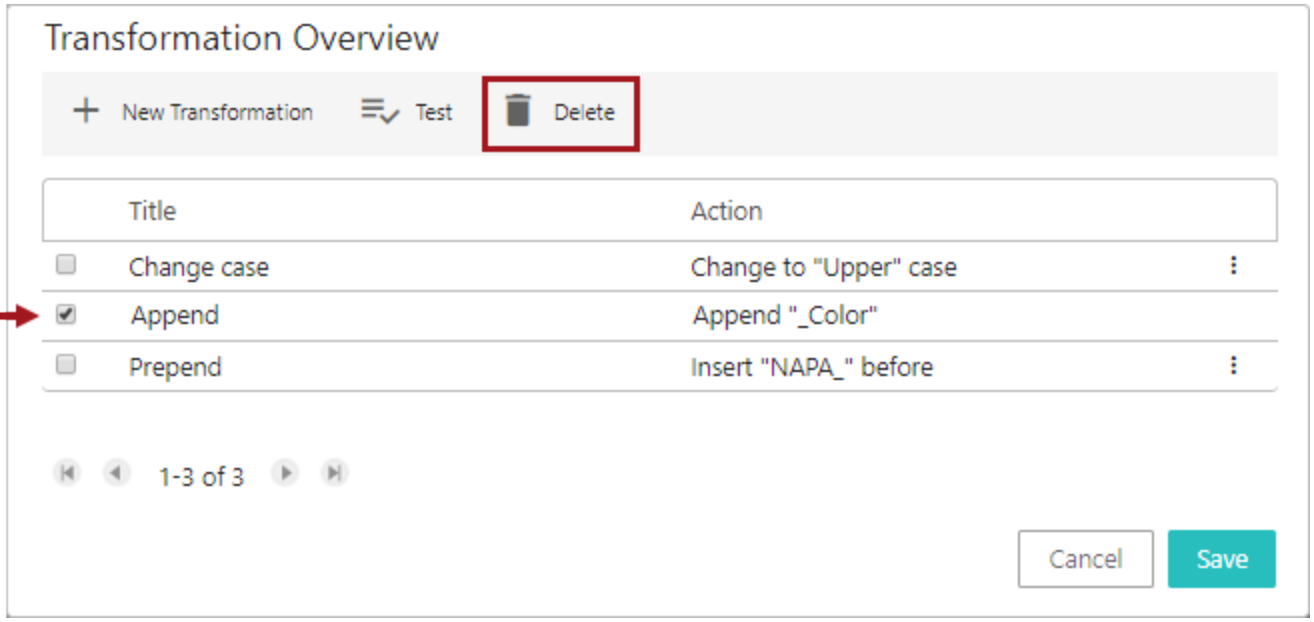
Details on how to test an attribute transformation configured in a Mapping plugin is explained in the **Testing Attribute Transformations Configured in Mapping Plugins** topic.

### Delete a transformation

Deleting a transformation will no longer display nor execute the transformation in the selected attribute transformation. It must only be deleted when it is no longer required. Deletion made through the Mapping Guide window impacts every other place where this transformation is applied / used.

1. In the Transformation Overview dialog, select the checkbox that is available on the left of the transformation that is intended to be deleted. This makes the **Delete** button (

) visible in the toolbar.

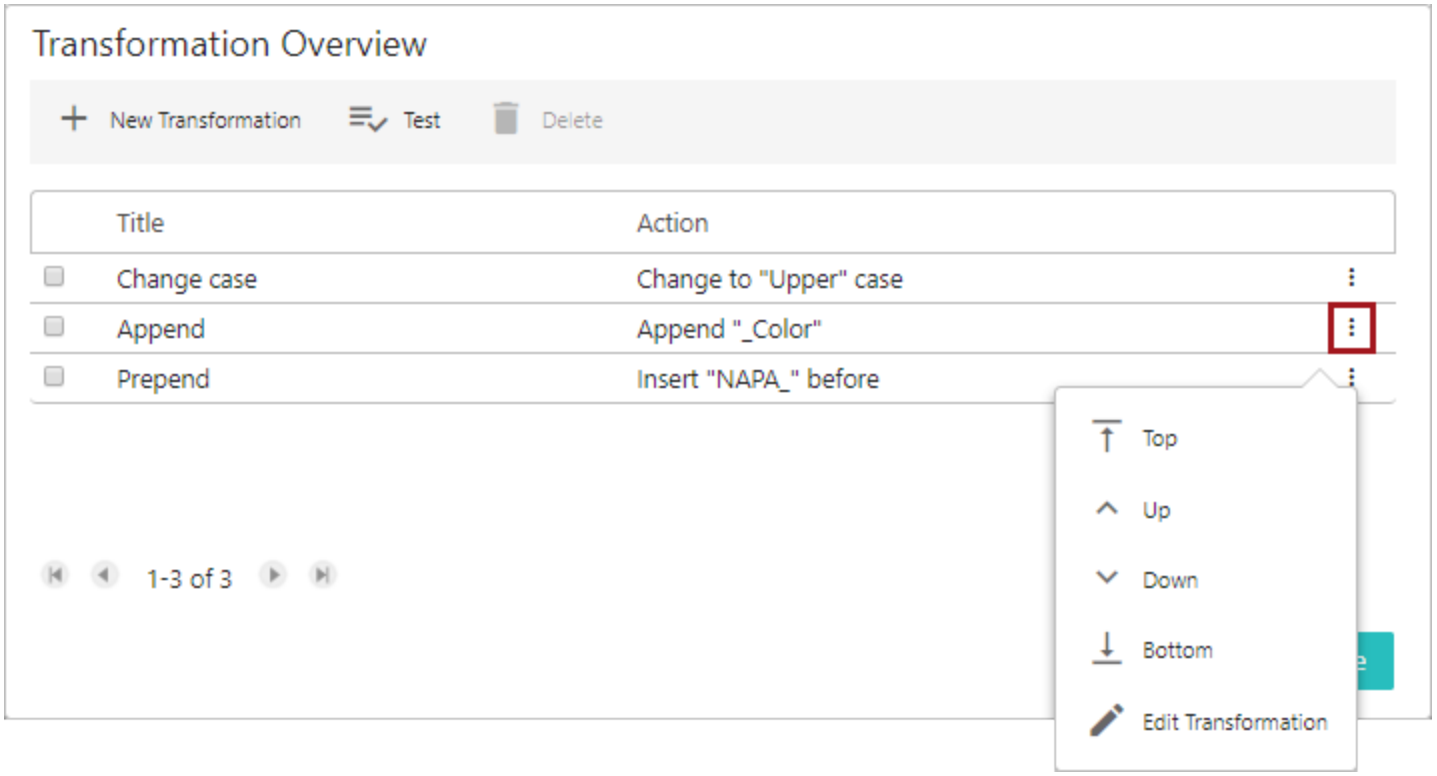



2. Click the **Delete** button. This removes the selected transformation(s) from the Transformation Overview dialog in Web UI and also removes the functionalities defined in that particular transformation.
3. Click the **Save** button to confirm that the transformation row should be deleted.

### Change the order of the transformations

When there are multiple transformations available within the attribute transformation, the transformations are applied in the order in which they are listed within the Transformation Overview dialog. Users can rearrange the order of the transformation as defined below.

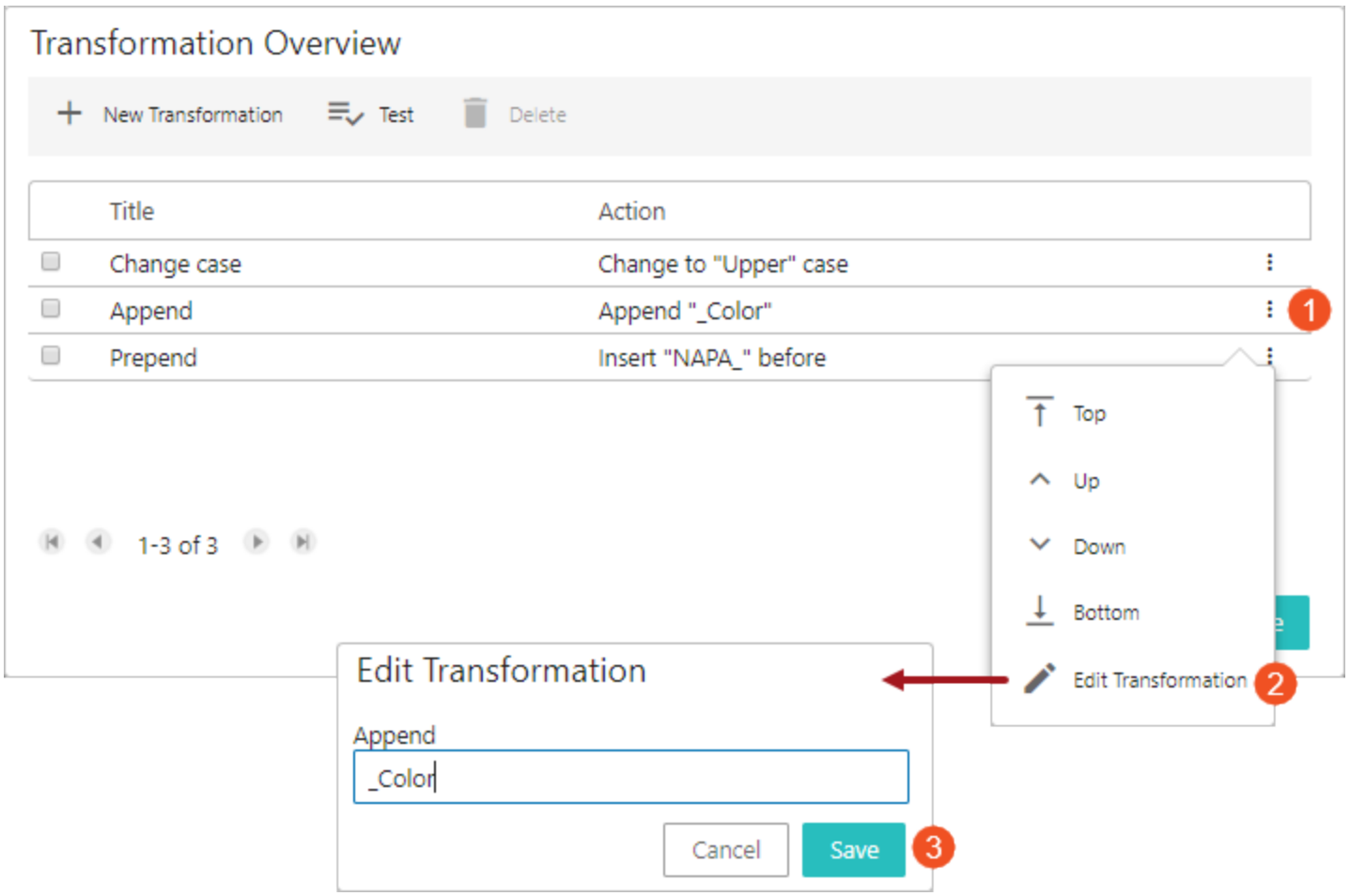
1. For the selected attribute transformation, access the Transformation Overview dialog.
2. Click the three-dotted icon ( ) that is available in the right of the transformation to display the rearranging dialog. The user can select the Top, Up, Down, or Bottom button to rearrange the transformation accordingly.




### Edit an existing transformation

Users can edit an existing transformations through the Transformation Overview dialog. Below is the procedure to edit an existing transformation.

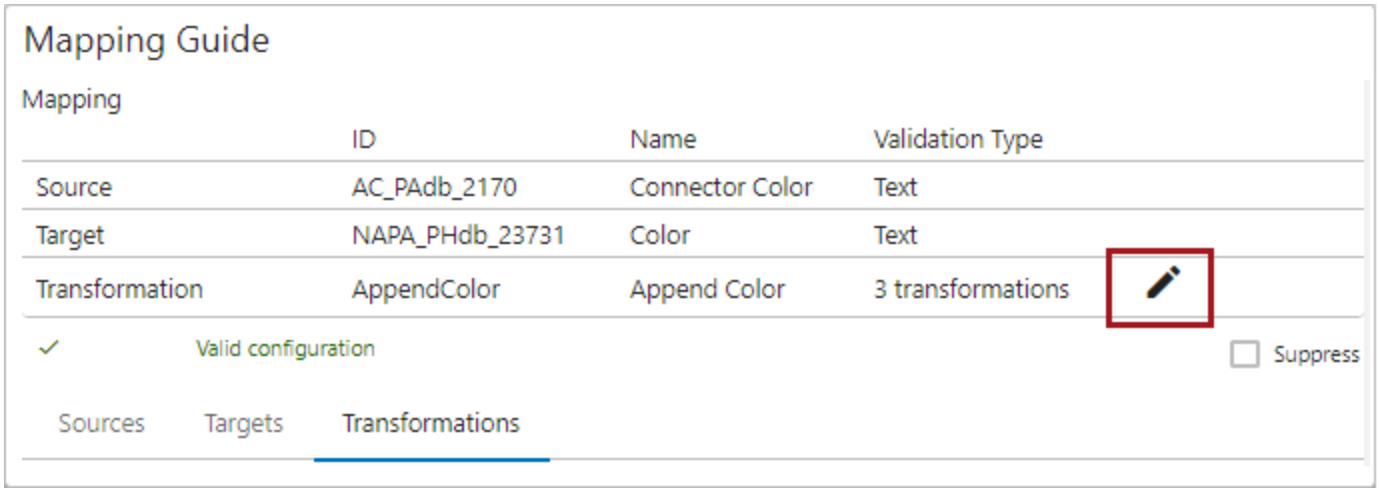
1. For the selected attribute transformation, access the Transformation Overview dialog, and click the three-dotted icon ( ) that is available in the right of the transformation to display the rearranging dialog.
2. Click the **Edit Transformation** button to open the Edit Transformation dialog.
3. Edit the transformation as needed and click **Save** to save and close the Edit Transformation dialog.


Transformation Overview
New Transformation
Test
Delete
Title
Action
Change case
Change to "Upper" case
Append
Append "_Color"
Prepend
Insert "NAPA_" before
1
1-3 of 3
Top
Up
Down
Bottom
Edit Transformation
2
Edit Transformation
Append
_Color
Cancel
Save
3

## Testing Attribute Transformations Configured in Mapping Plugins

Users can test the attribute transformation configured in the Mapping plugin in the Web UI by entering a sample test value as explained below. This helps to test the transformation by evaluating if the attribute value is transformed correctly without actually changing the data or moving the data from one attribute to the other.

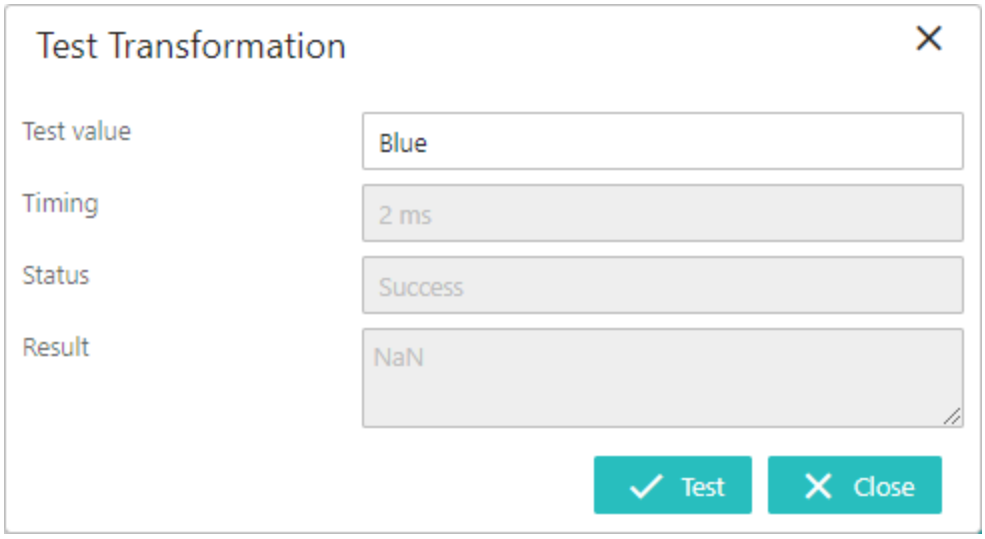
When an existing attribute transformation is added for any Mapping plugin, an edit icon ( ) is displayed within the Transformation field in the Mapping Guide window. Clicking the edit icon ( ) opens the Transformation Overview dialog and lets users test the functionality of the transformation via the Test button (as shown below).




Click the Test button in the Transformation Overview dialog to open the Test Transformation dialog with the parameters explained as follows:

- **Test Value:** This editable blank field lets users enter text to be used as the sample value for testing.
- **Timing:** This read-only field displays the time elapsed in transforming the 'Test value' field once the **Test** action button is clicked in the Test Transformation dialog.
- **Status:** This read-only field displays the success or failure status of the transformation when the **Test** action button is clicked in the Test Transformation dialog.
- **Result:** This read-only field displays the final transformed value of 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.'



# 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, refer to 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

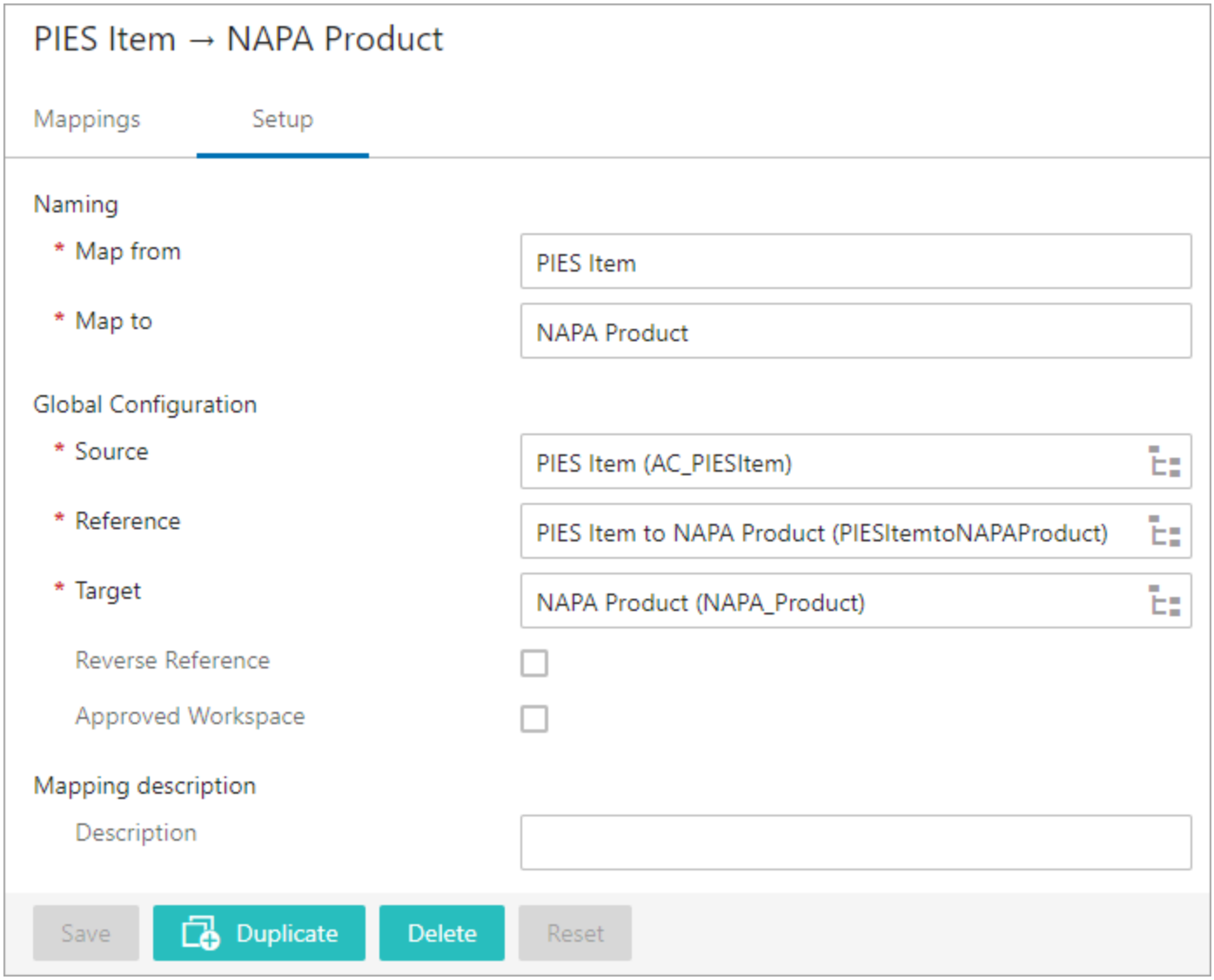
The requirement for this use case entails that the user has an attribute value in the Source object that needs to be copied over to the Target object. The user can do this either by using the Attribute Mapping plugin or by using the Object to Object Mapping plugin where the Source STEP path and the Target 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

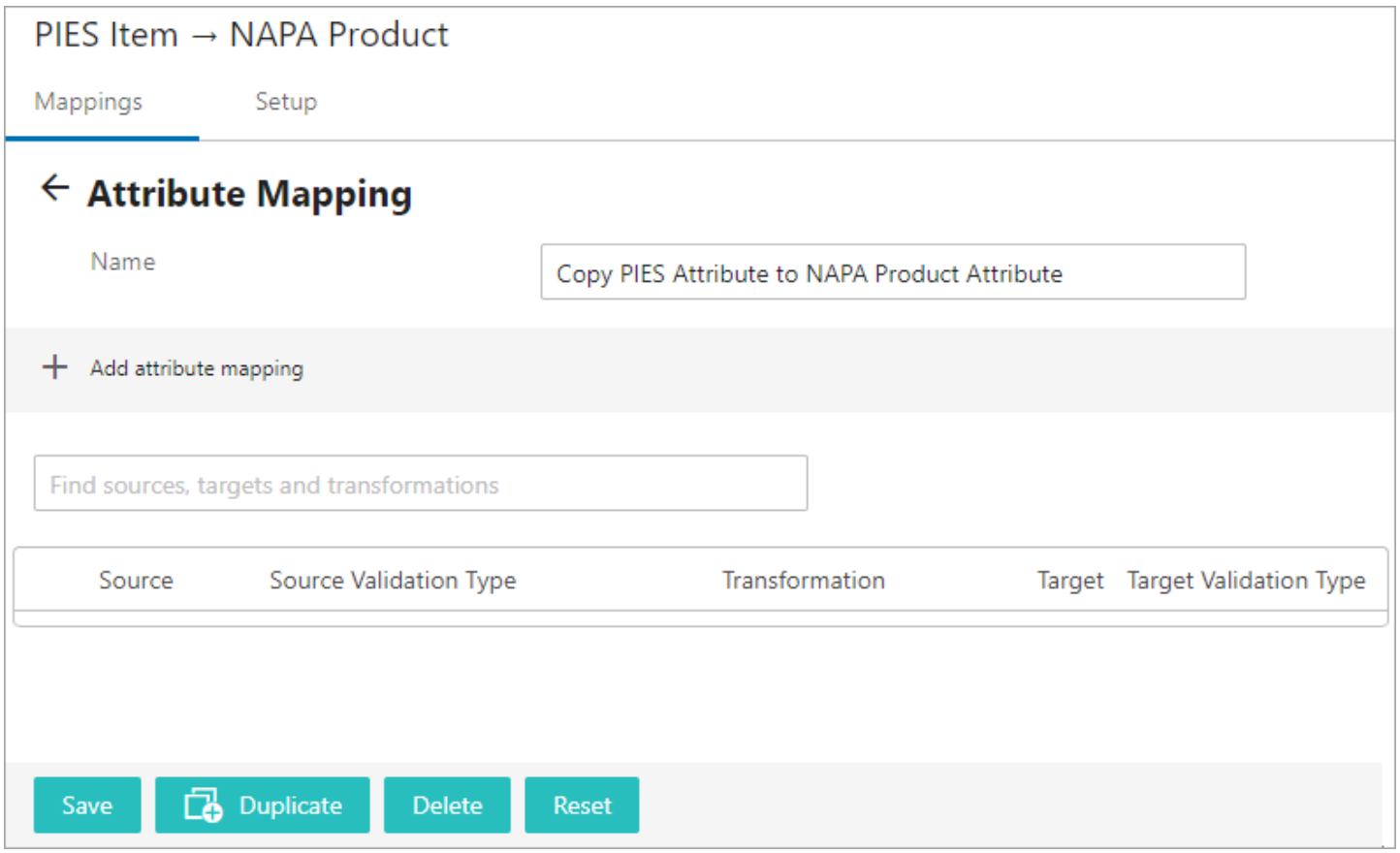
The validity of the Source and the Target attribute shall not have any mismatch.

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, refer to topic **Configuring Mapper Configuration Setup Entity** within this guide.

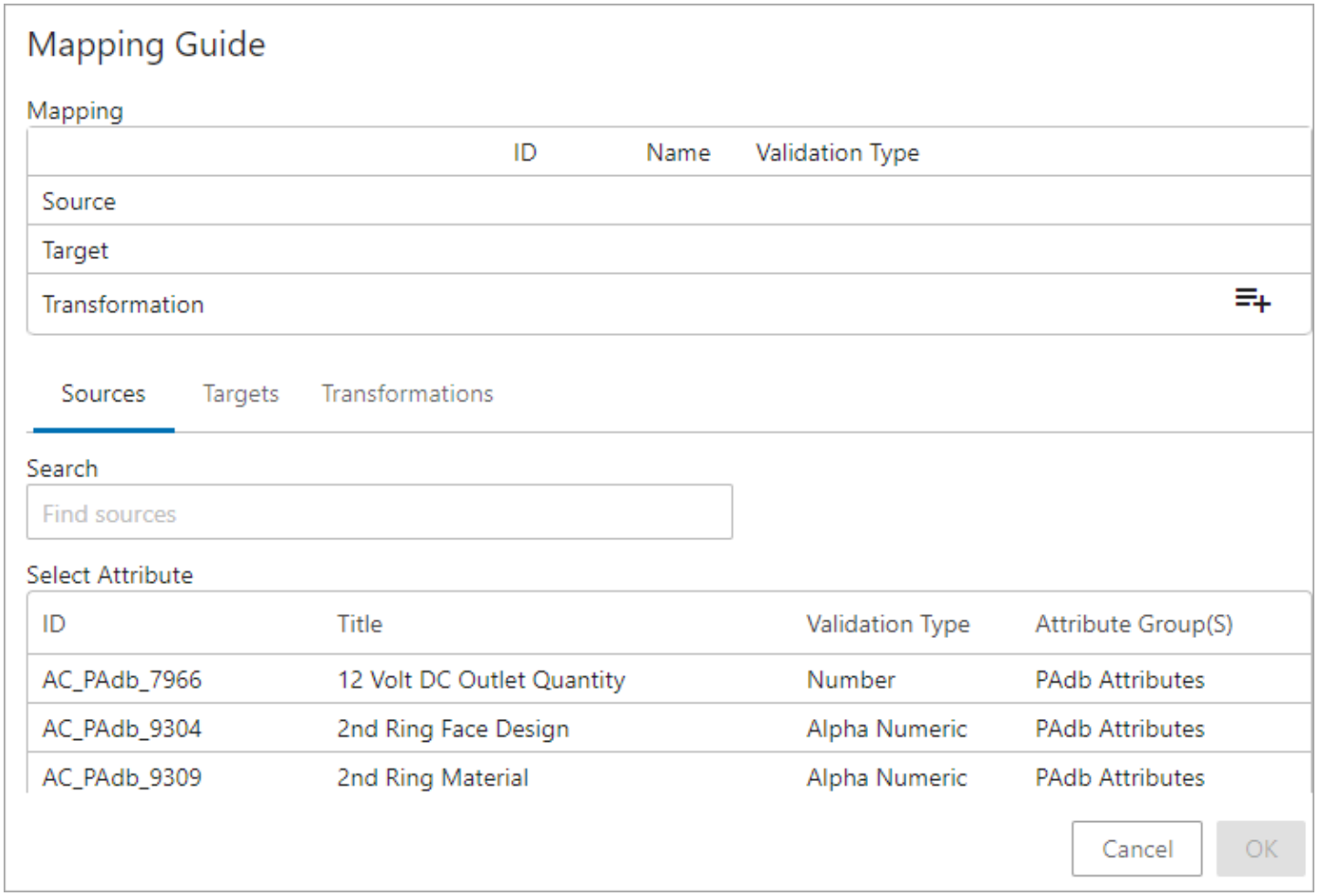



Configure an Initiate Business Action button within the Product Details screen of the Source object which shall execute the Mapper Configuration. For more information on executing the Mapper Configuration, refer to topic **Executing Mapper Configuration Setup Entity** within this guide.

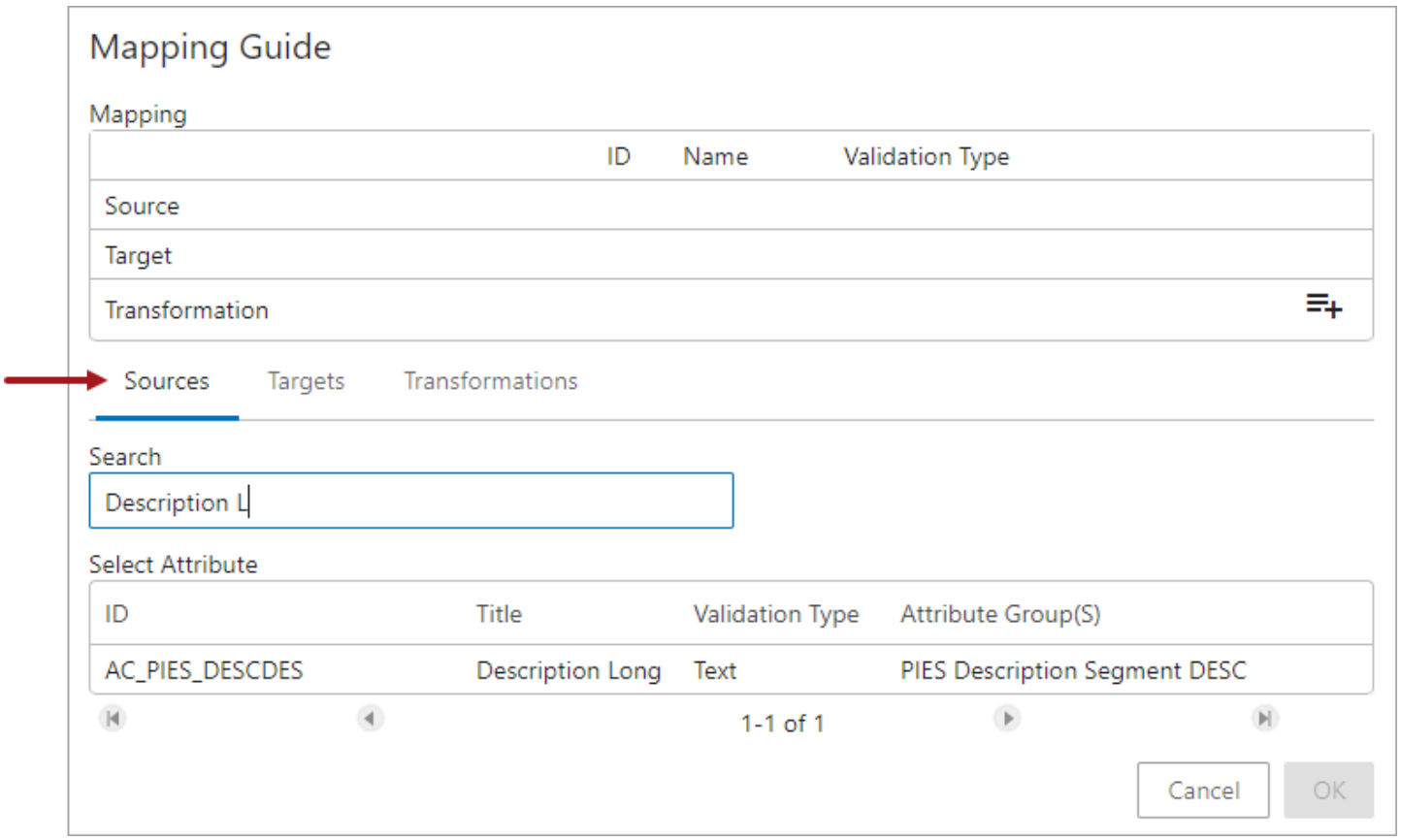
1. Add a new Attribute Mapping plugin (or edit the one that exists) in the required Mapper Configuration.
2. Type in a suitable name next to Name field. In this example, the Name field is entered with the text 'Copy PIES Attribute to NAPA Product Attribute'



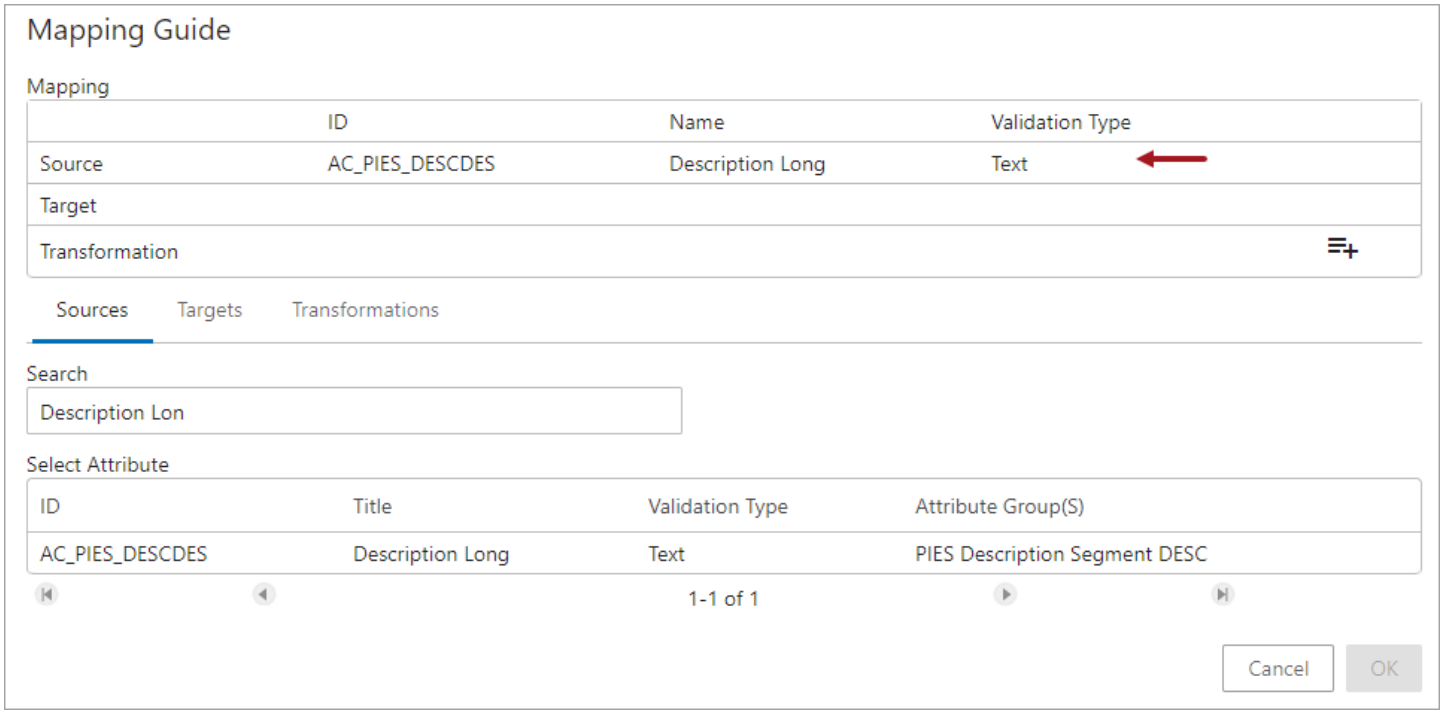

3. Click the ‘Add attribute mapping’ icon, and the Mapping Guide window will display (as shown below).



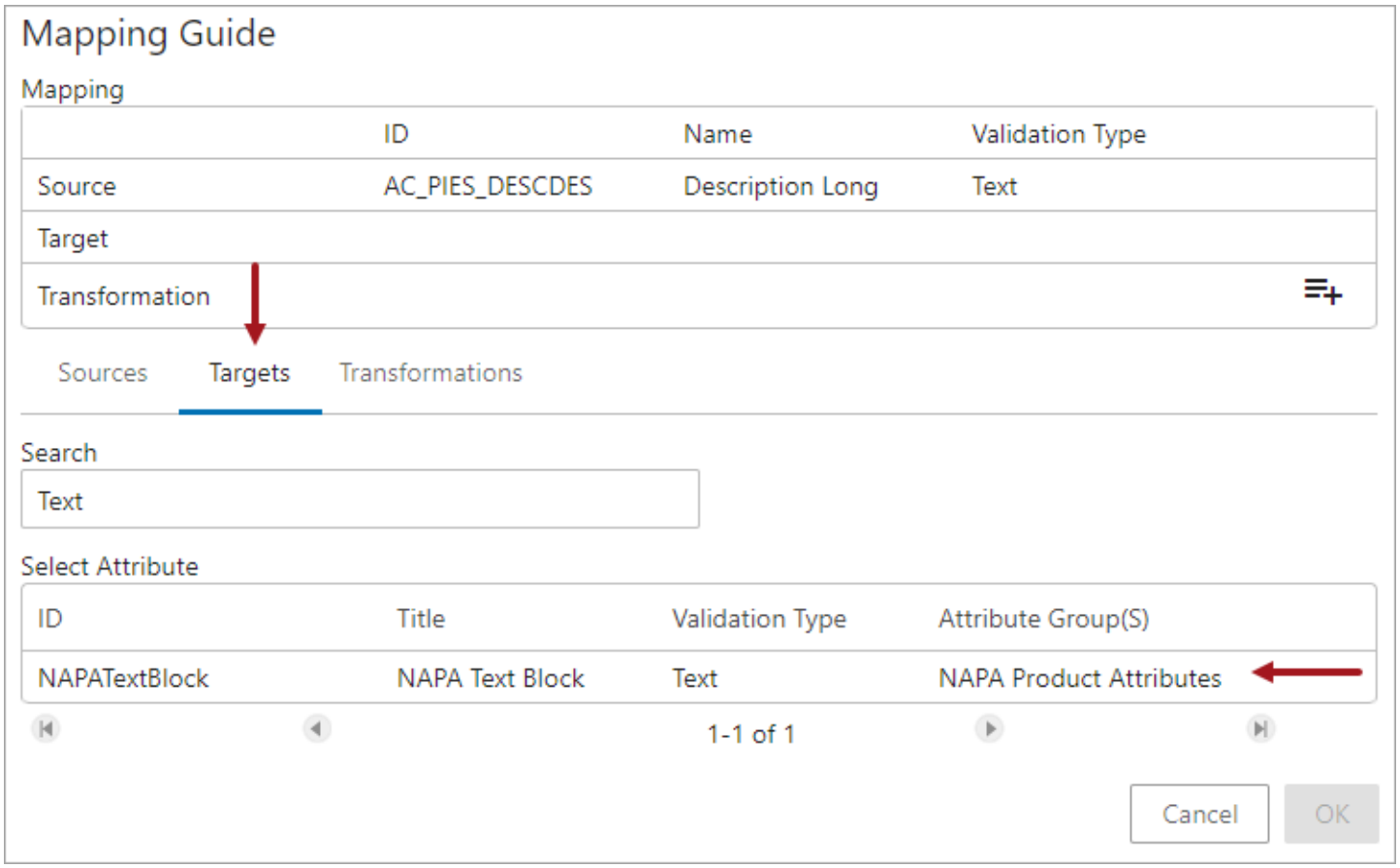

4. With the Sources tab selected, click in the search field and start typing the initial letters of the attribute name or ID. This searches for the attribute of the Source object from where the attribute value could be retrieved. Select the attribute from the result list displayed below the search bar. In this example, the attribute 'Description Long' is used as source attribute.




5. Click **OK** and the selected attribute will be populated in the Source field (as shown below).

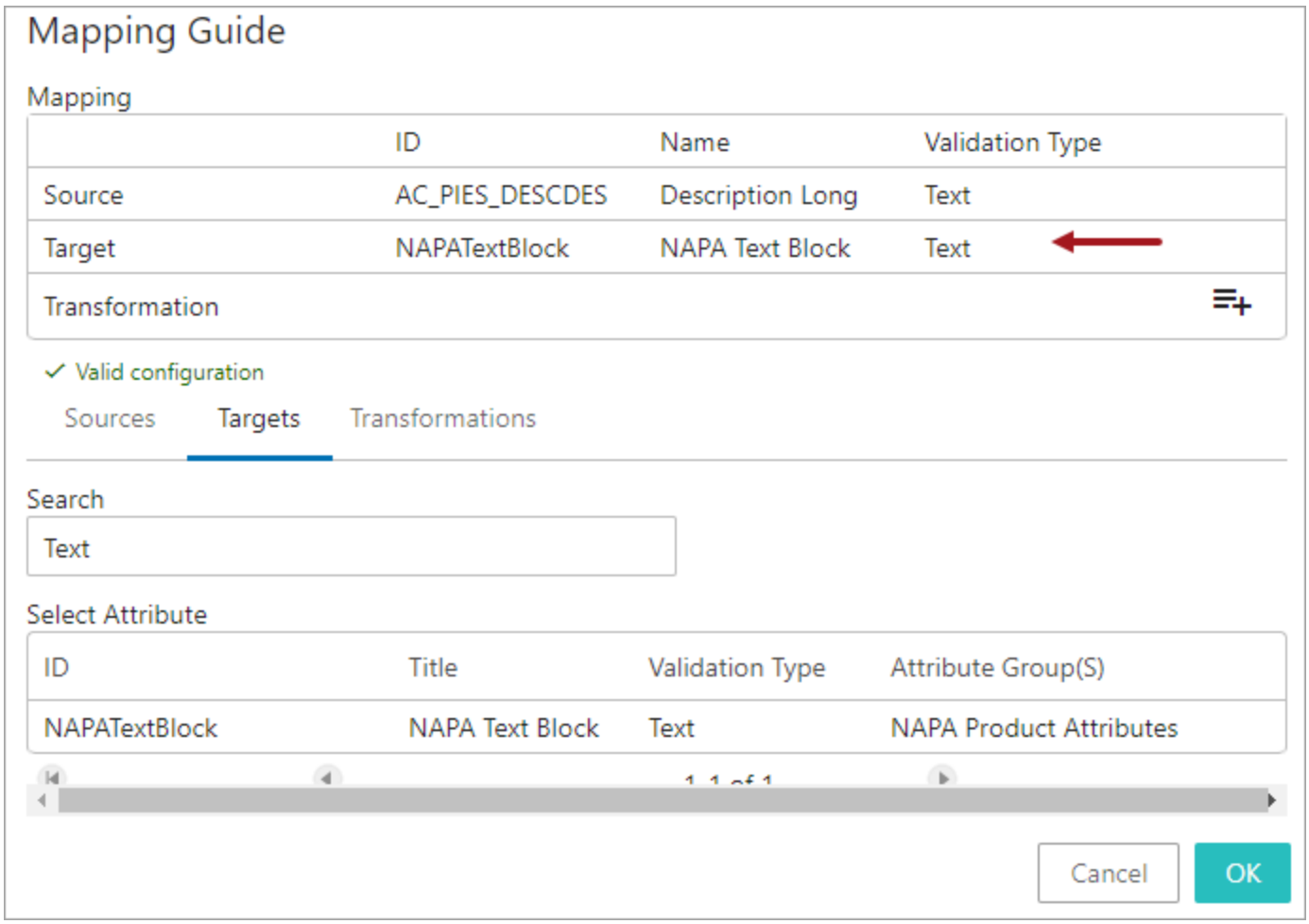


6. With the Targets tab selected, search and select the attribute of the Target object where the retrieved value from the Source object shall reside.

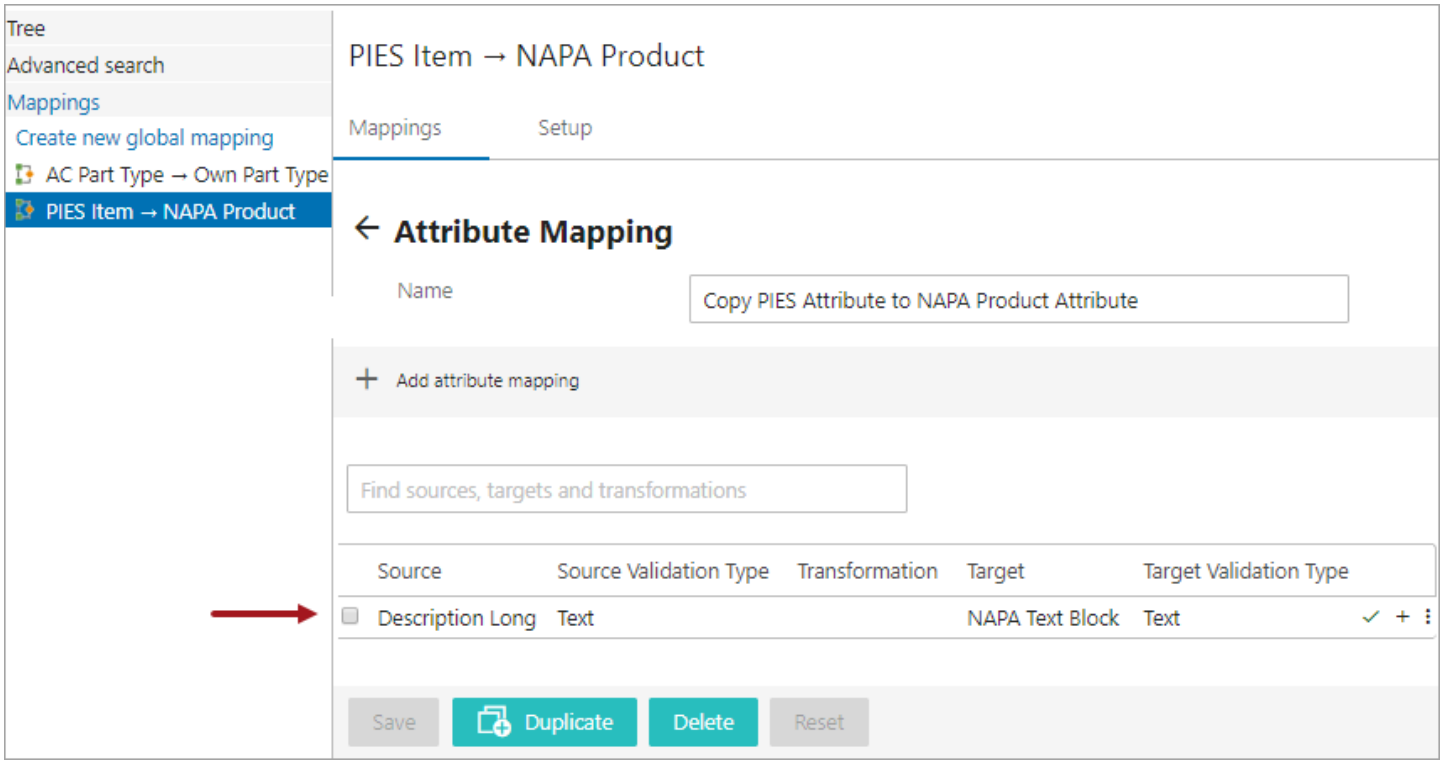



**Note:** The Source object attribute and the Target object attribute should have a validity match.

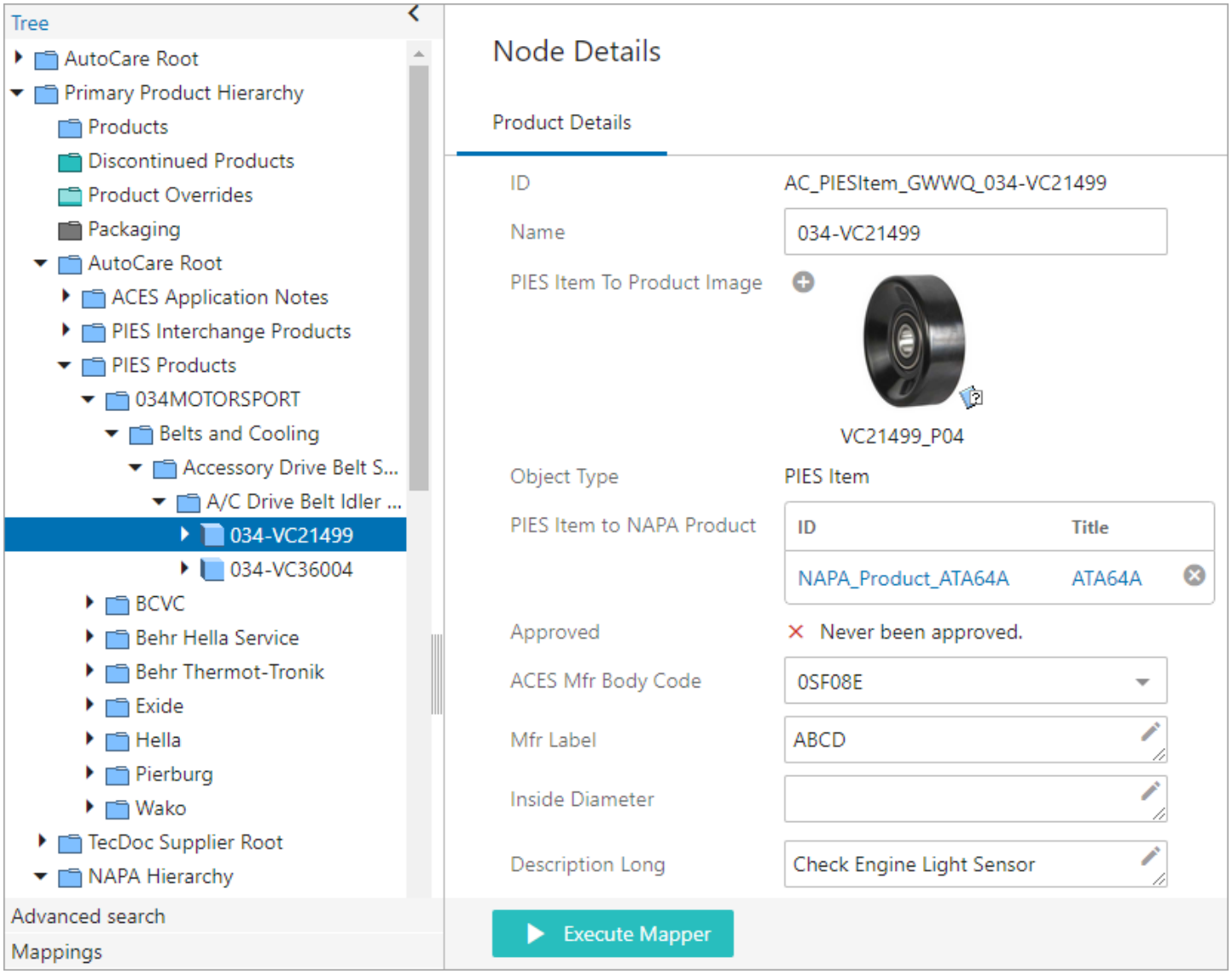
Once selected, the selected attribute will be populated in the Target field.




7. Click **OK** to save and close the Mapping Guide window. The newly added object mapping row will be listed as shown below.




8. 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, refer to topic **Executing Mapper Configuration Setup Entity** within this guide.




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

- AutoCare Root
- Primary Product Hierarchy
  - Products
  - Discontinued Products
  - Product Overrides
  - Packaging
  - AutoCare Root
    - ACES Application Notes
    - PIES Interchange Products
    - PIES Products
  - TecDoc Supplier Root
  - NAPA Hierarchy
    - NAPA Comment Hierarchy
    - NAPA Interchange Product Hi...
    - NAPA Product Hierarchy
      - ATA Fuel Injector - New
        - ATA64A
      - RPC 63858
      - Wiper Blade - Bosch Icon - ...
- TecDoc Supplier Price Lists
- Import Flow Root

Advanced search

Mappings

## Node Details

Product Details

| | |
|---|---|
| ID | NAPA_Product_ATA64A |
| Name | ATA64A |
| Object Type | NAPA Product |
| Approved | ✕ Never been approved. |
| Part Number To Image | VC21499_P04 |
| # Steering gear Box Mounting Holes | |
| Color | |
| NAPA Package INfo | |
| * NAPA Part Number | |
| NAPA Text Block | Check Engine Light Sensor |

### 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

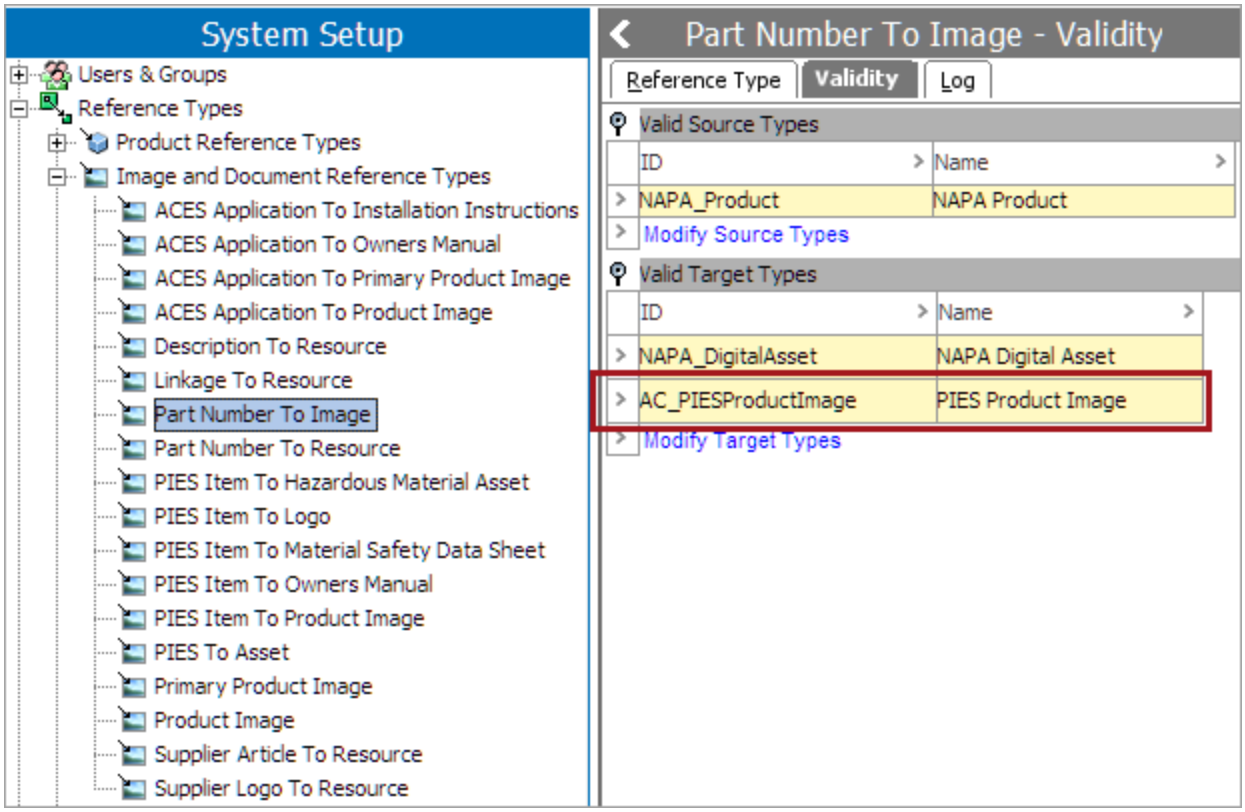
Consider an asset that is referenced as the Primary Product Image on the Source object (for example, PIES Item) and the user needs to also use the same asset on the Target object (for example, NAPA Product) as the Primary Product Image.

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 'PIESItemtoNAPAProduct' in the system.

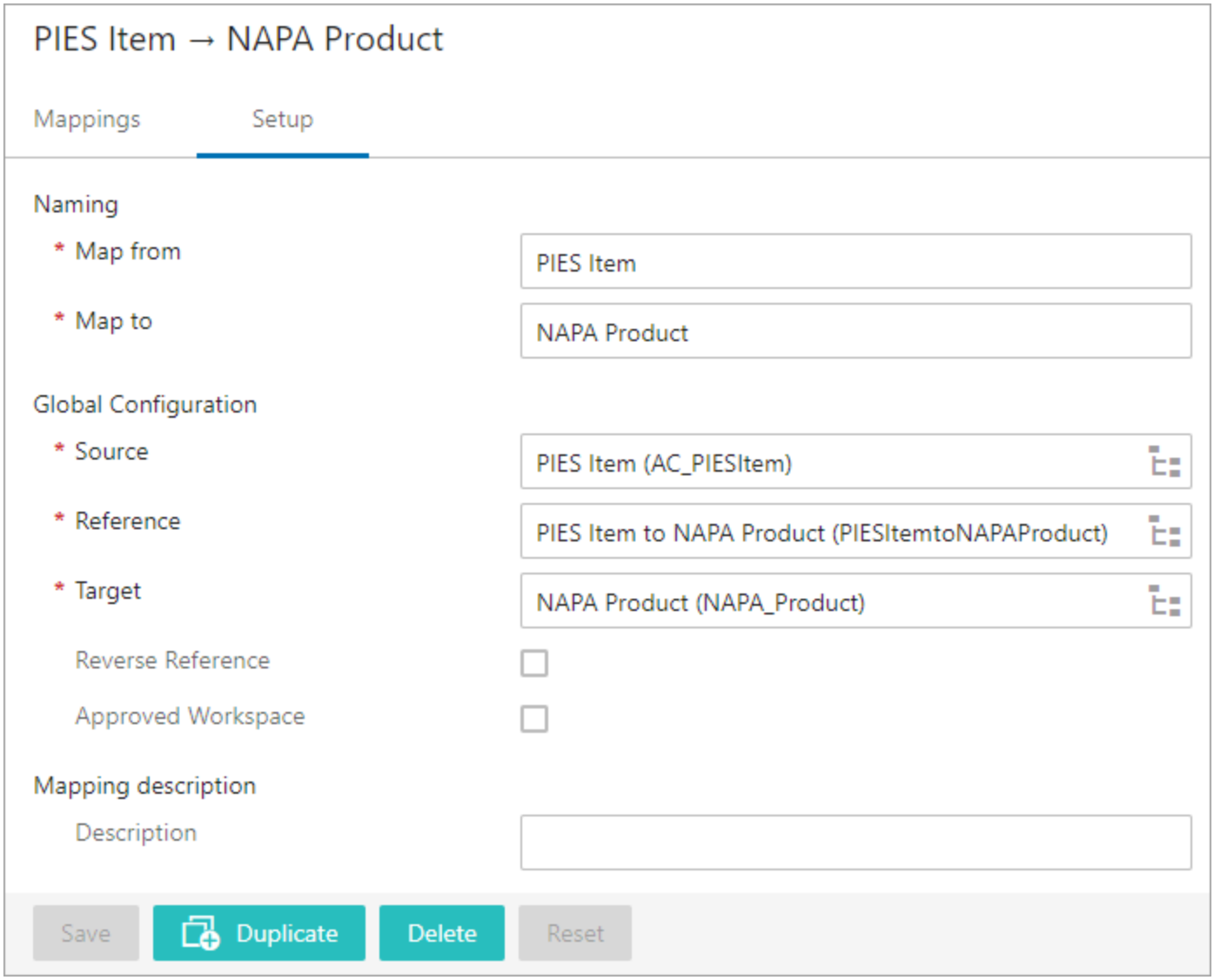
Following are the detailed steps describing how to configure the mapping to create the asset that is used as the Primary Product Image reference on a PIES Item to be used as the Primary Product Image reference on a NAPA Product, too.

## 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, refer to the **Reference Types** topic within the **System Setup** section of **STEP Online Help**.

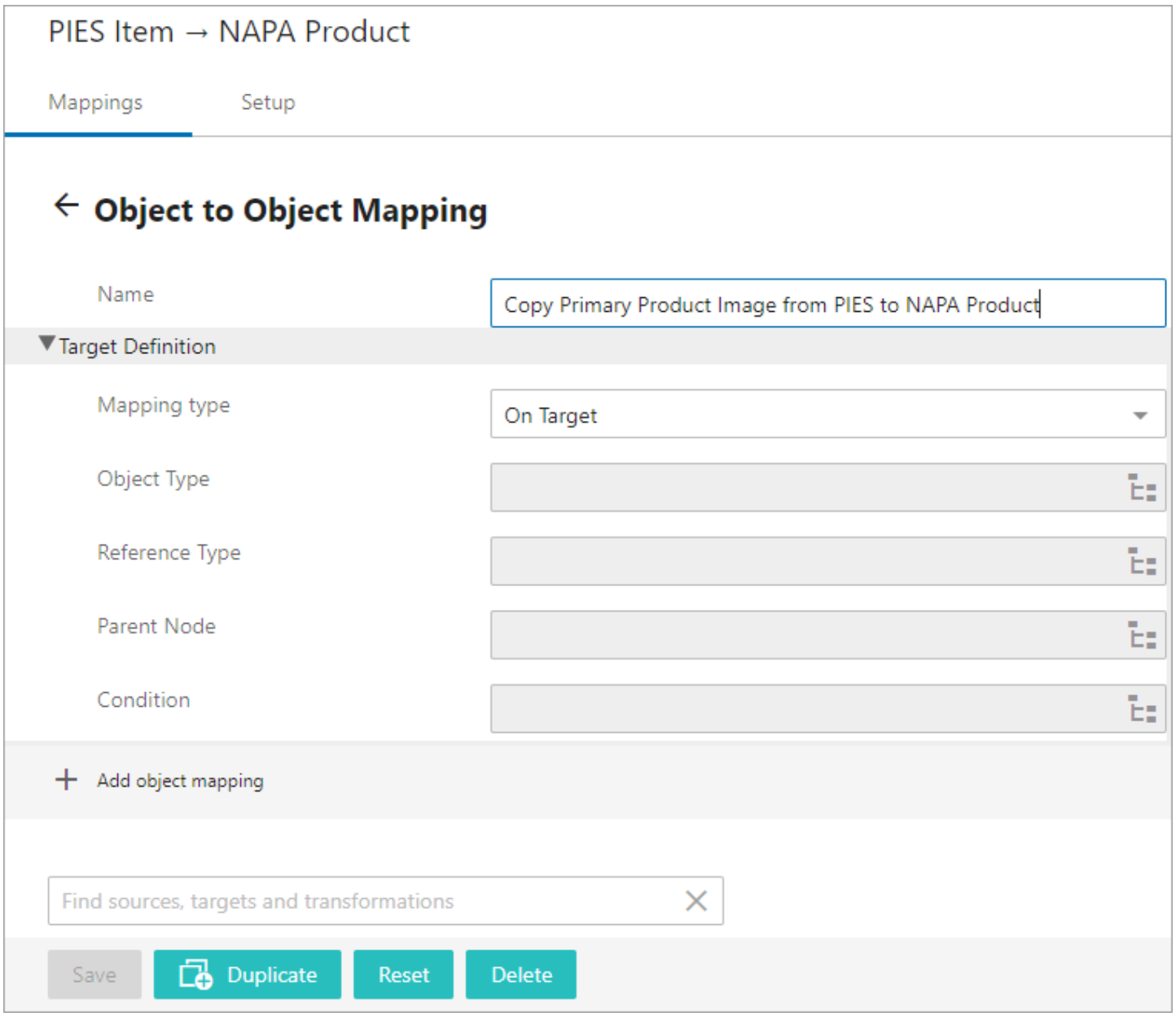



In the Web UI, create and configure the Mapper Configuration and define the Source and Target object types as well as the Reference type to follow to determine the relationship between the Source and Target object. For more information on configuring and setup procedure about the Mapper Configuration, refer to topic **Configuring Mapper Configuration Setup Entity** within this guide.

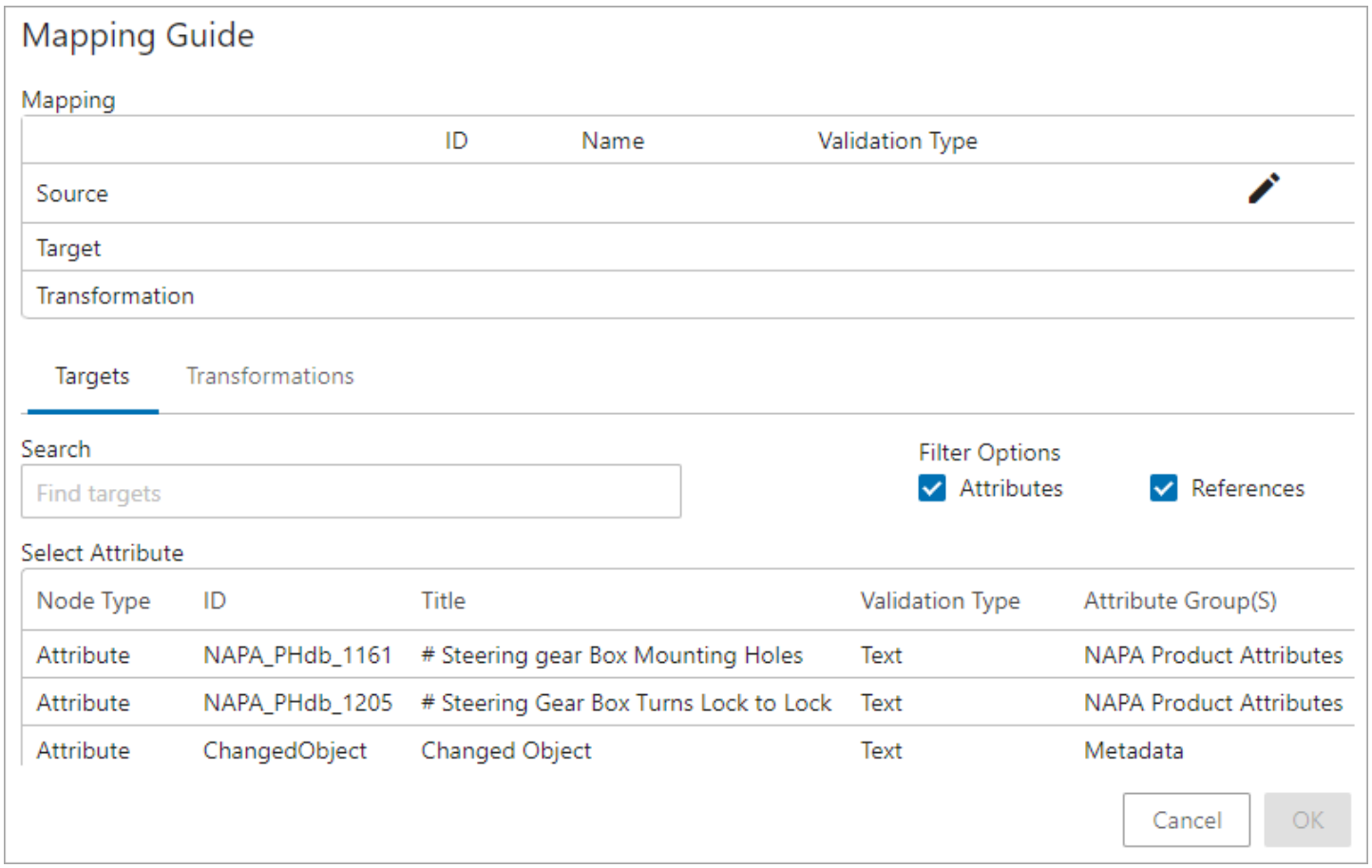



Configure an Initiate Business Action button within the Product Details screen of the Source object which shall execute the Mapper Configuration. For more information on executing the Mapper Configuration, refer to topic **Executing Mapper Configuration Setup Entity** within this guide.

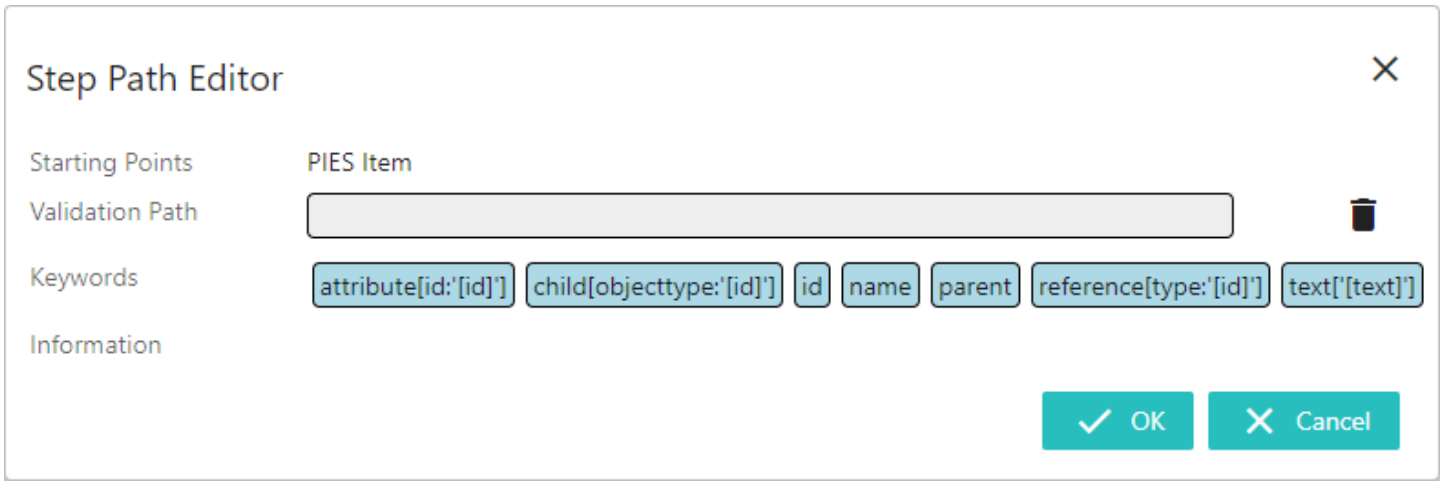
1. Add a new Object to Object Mapping plugin (or edit the one that exists) in the required Mapper Configuration.
2. Type in a suitable name next to Name field. In this example, the Name field is entered with the text 'Copy Primary Product Image from PIES to NAPA Product.'




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.

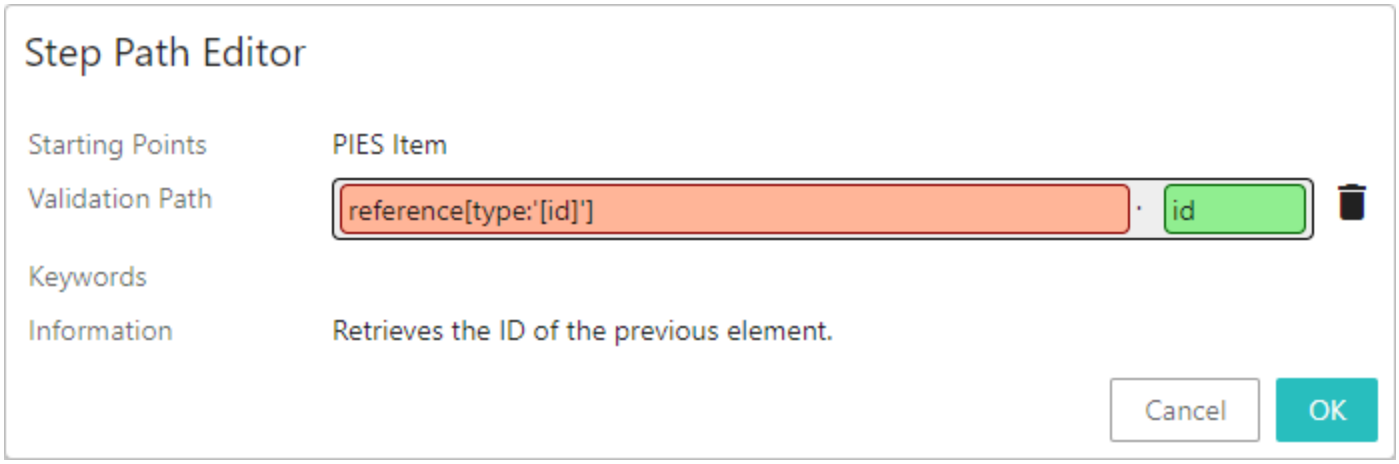



5. Click on the Edit icon ( ) in the Source field, and the Step Path Editor window will display (as shown below). For more information on defining the Source STEP path, refer to topic **Mapping Validation Path Functionality** within this guide.



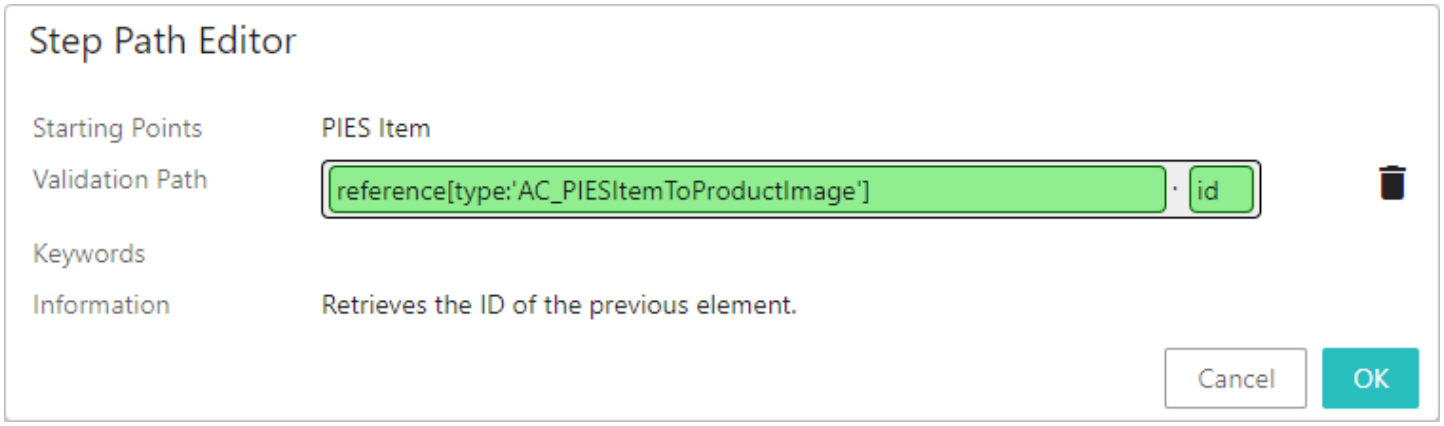

6. First, double click to add the **reference[type:'[id]']** element from the Keywords field. Also, the ID of the asset must be configured from the Source to use in the Target reference, so double click to add the **id** element from

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.

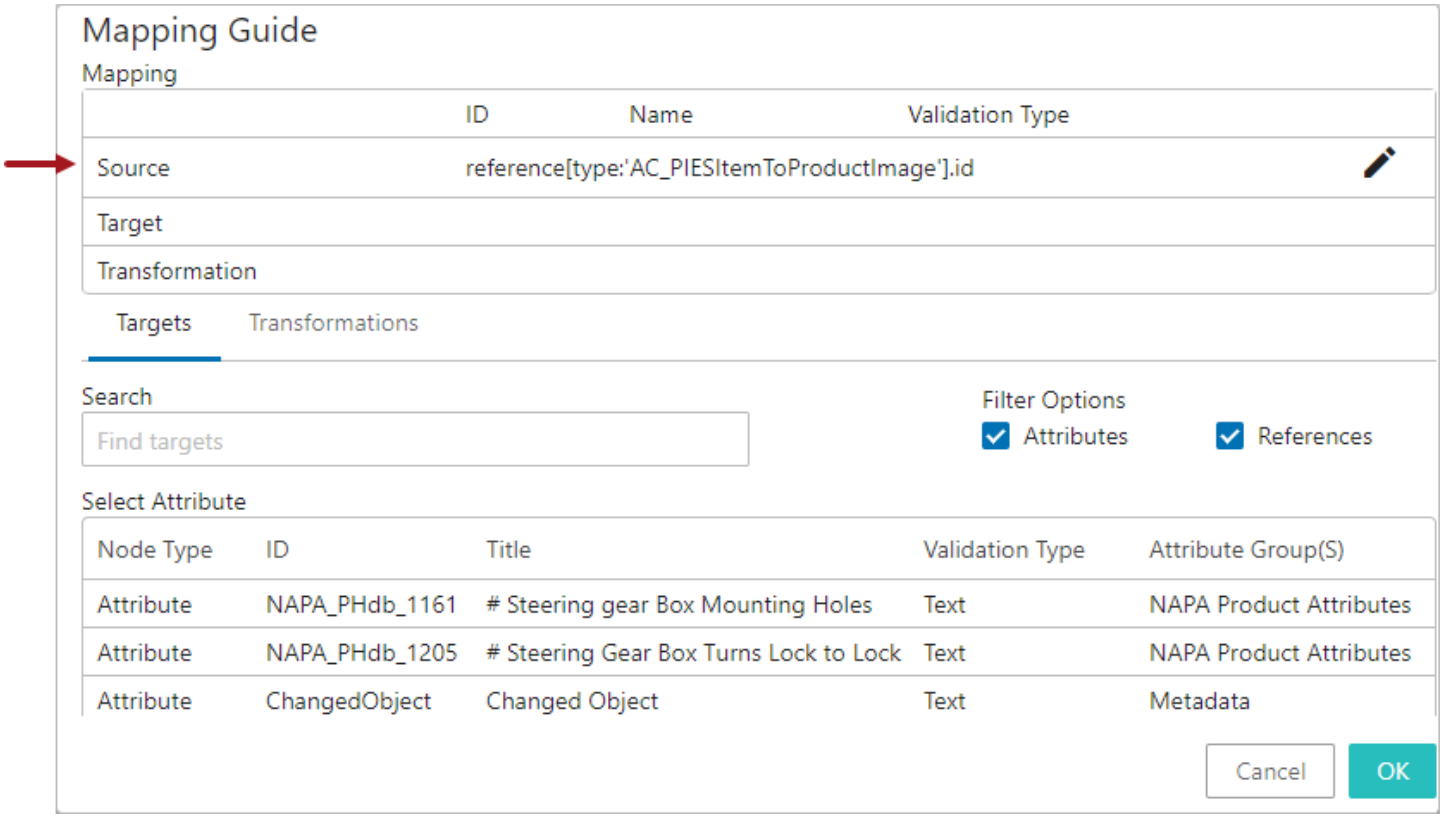



7. Click on the empty red colored **reference[type:'[id]']** element available within the Validation Path field and select the reference type that is referencing the Primary Product Image on the Source object. In this example, it is the reference type with id 'AC_PIESItemToProductImage.'

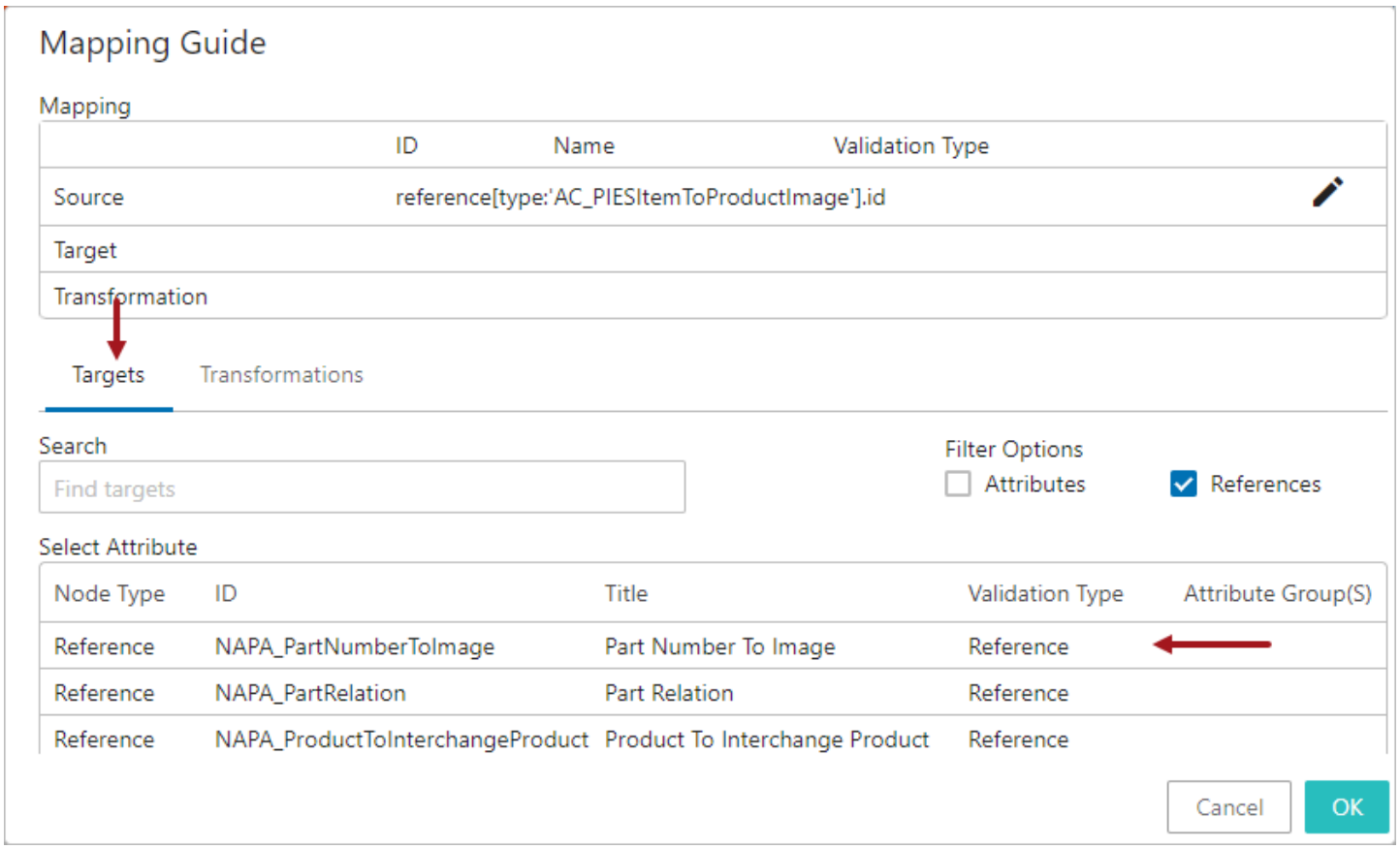
   Once populated, the elements in the Validation path turn green in color as displayed below.




8. Click **OK** and the Source STEP Path will be populated in the Source field.

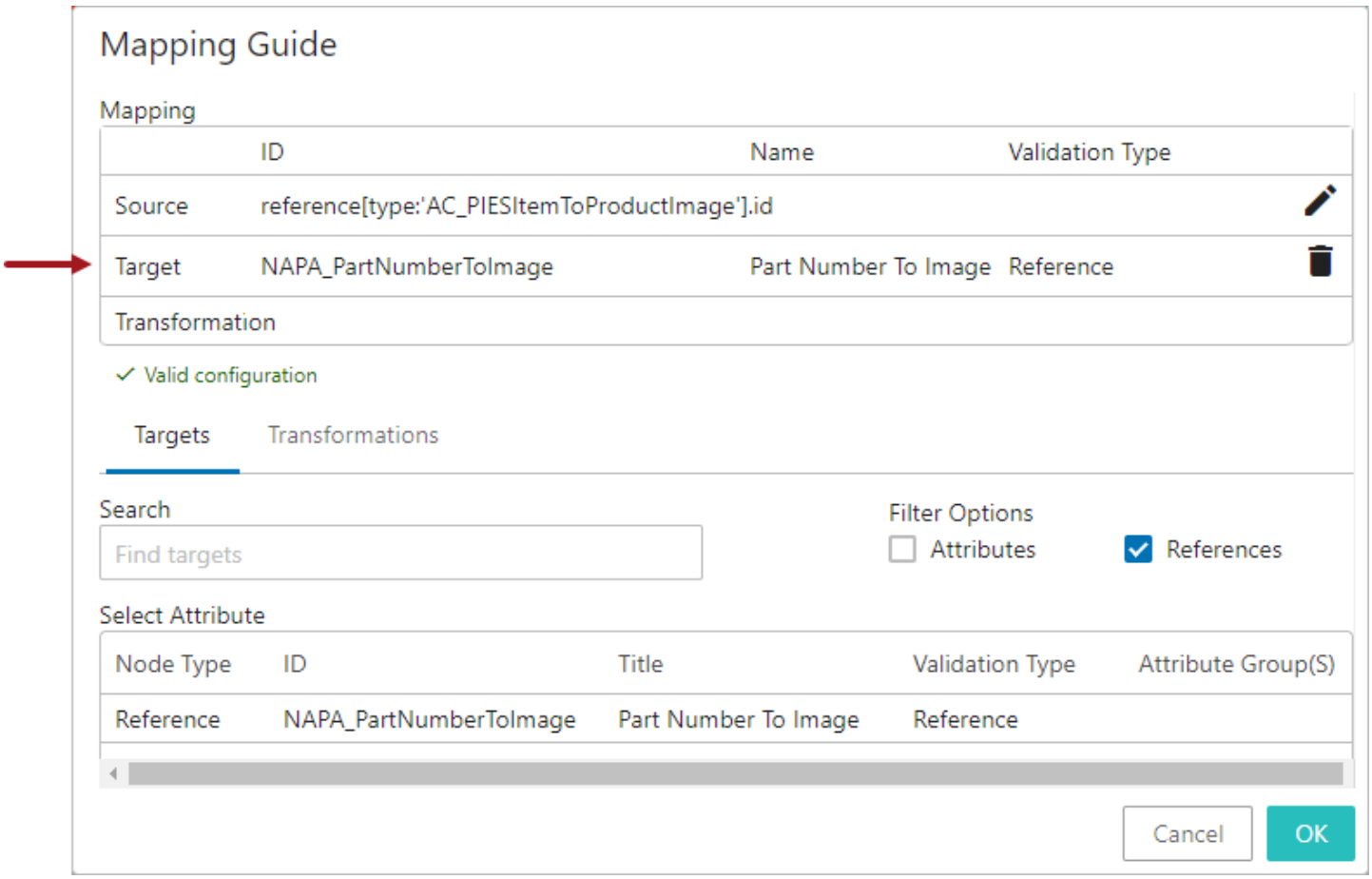



9. With the Targets tab selected, search and select the 'Image and Document Reference Types' of the Target object. In this reference type, the Primary Product Image of the Source object will reside.

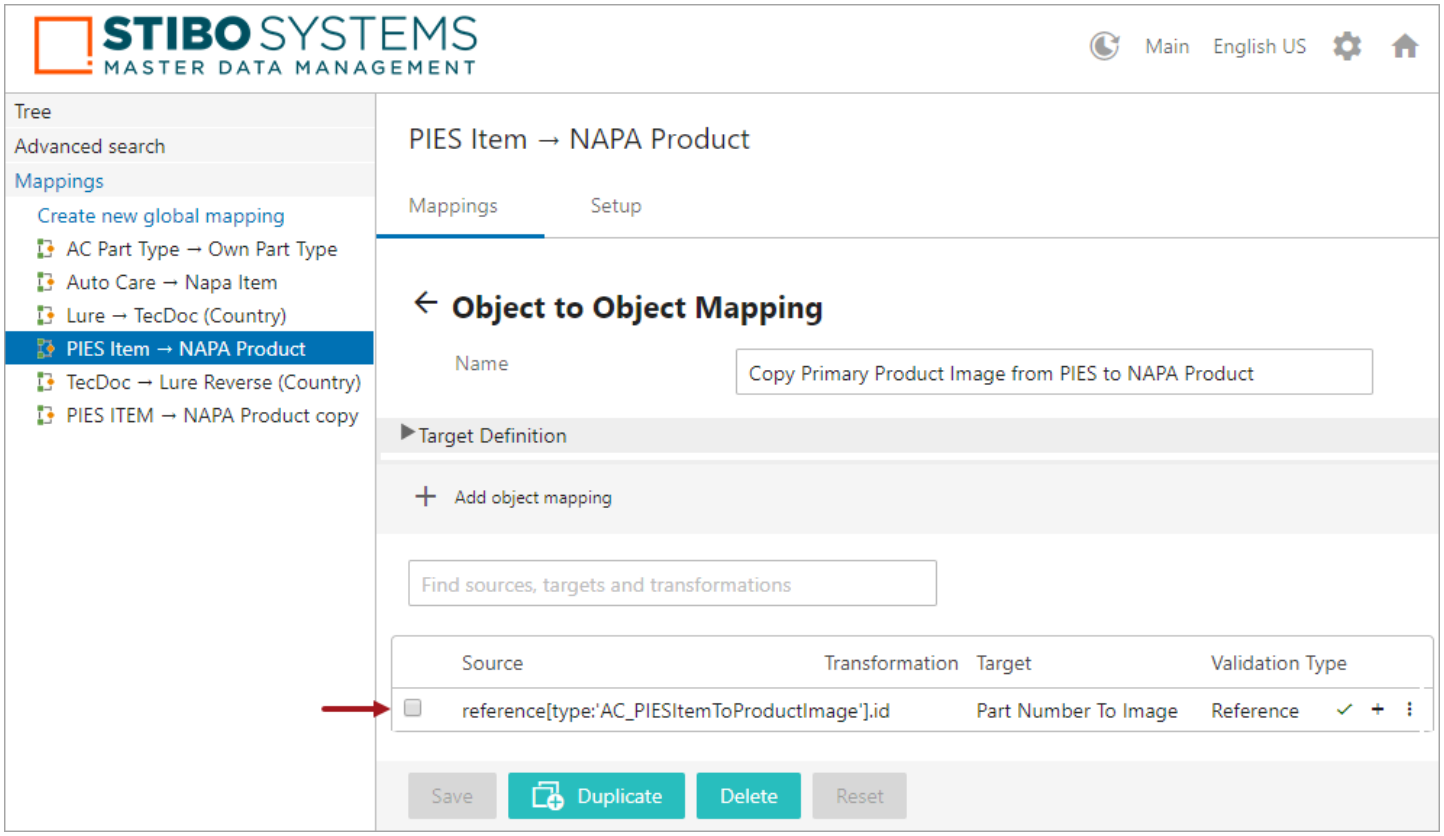



**Note:** The asset object type must be defined as the Valid Target Type for the Image Reference Type for both the Source and the Target object.

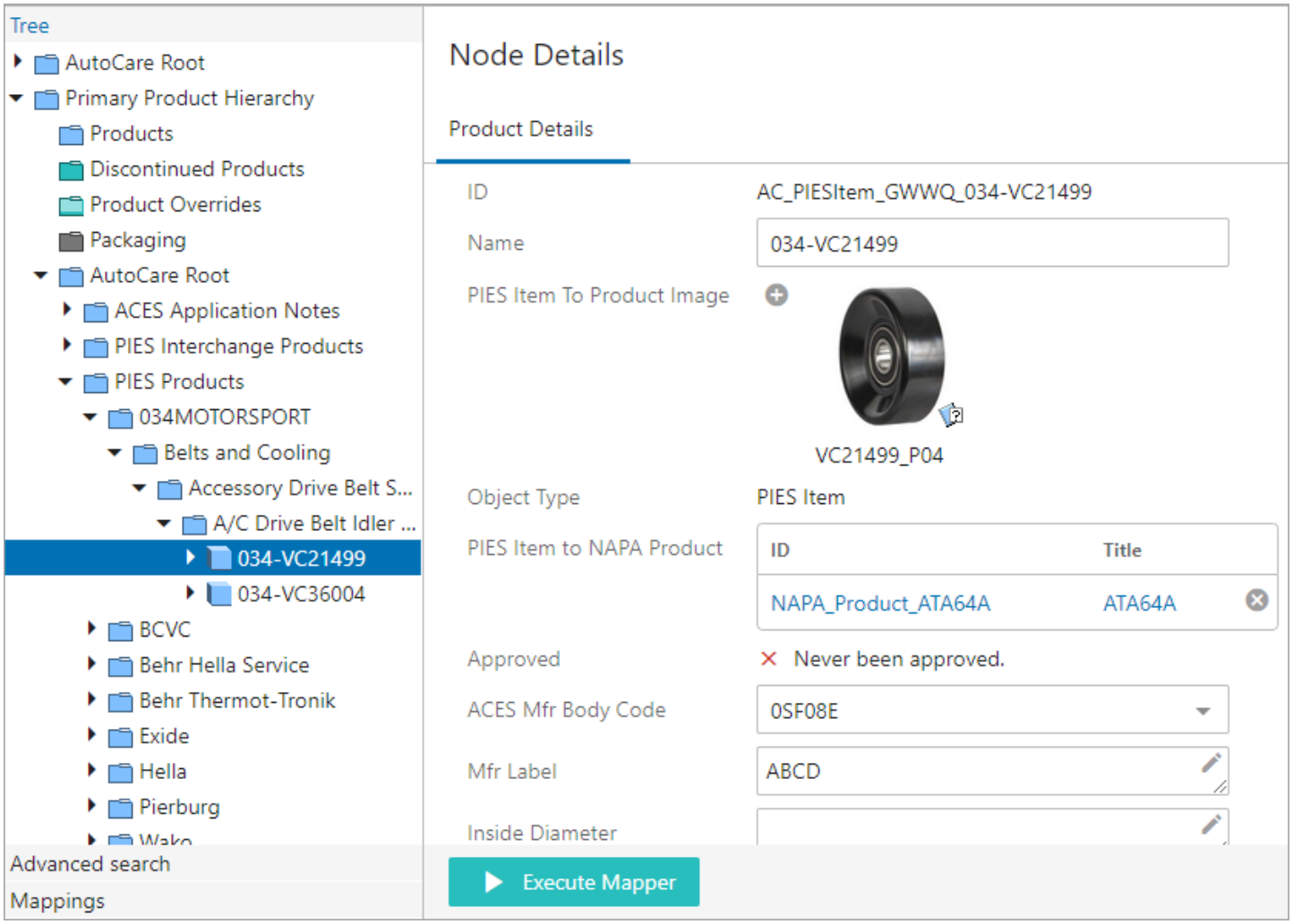
Once selected, the selected reference type will be populated in the Target field.



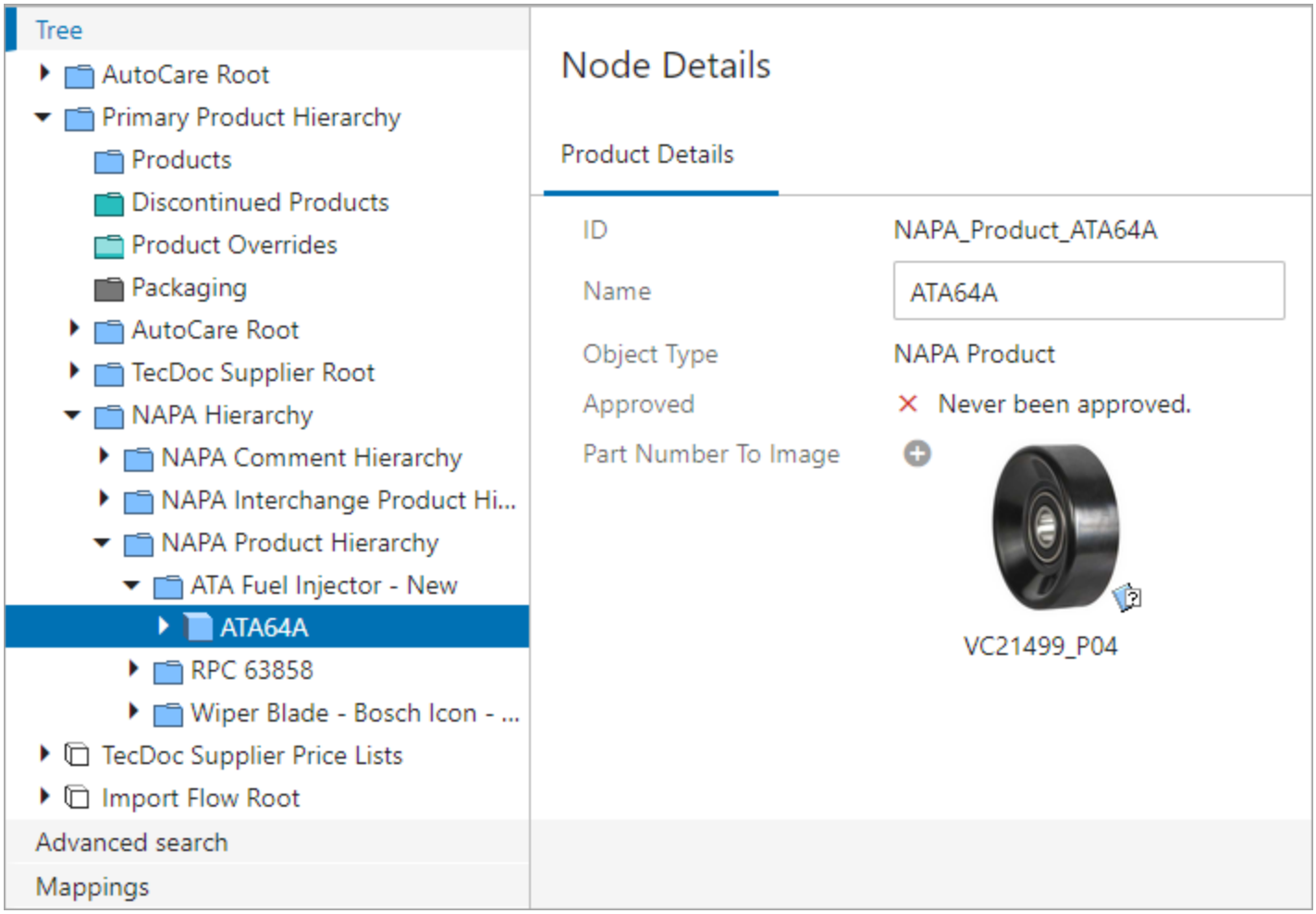

10. Click **OK** to close the Mapping Guide window and then click **Save** to save the changes. The newly added object mapping row will be listed.




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, refer to topic **Executing Mapper Configuration Setup Entity** within this guide.




After the user runs the mapping, the asset (VC21499_P04) that was used in the Source object is created as a reference on the Target object using the target reference type that is configured in the mapping.




**Example Two**

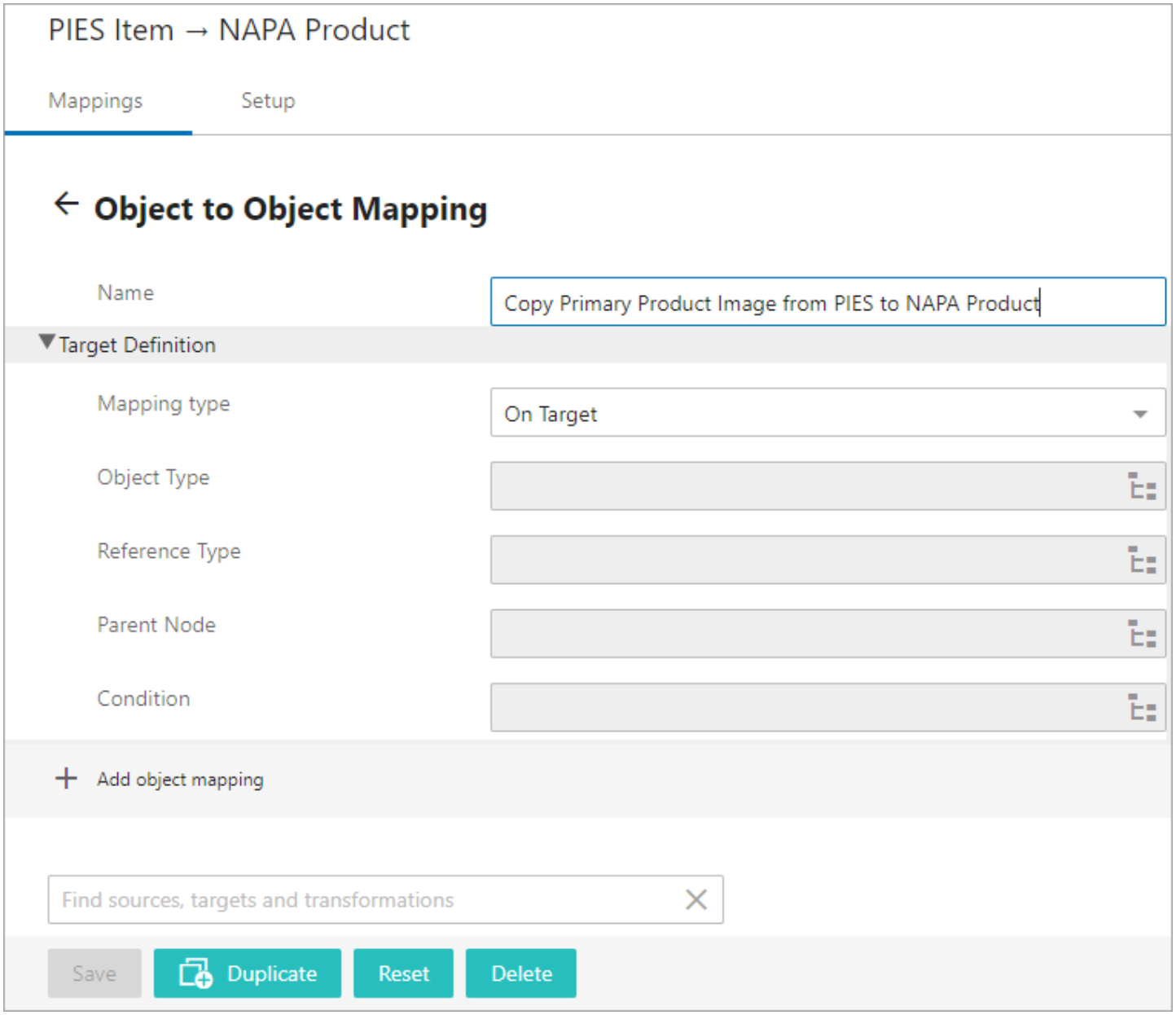
An additional requirement for this use case entails that the user has an attribute value in the Source object that needs to be copied over to the Target object. The user can do this either by using the Attribute Mapping plugin or the Object to Object Mapping plugin. As the main intention of this topic 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

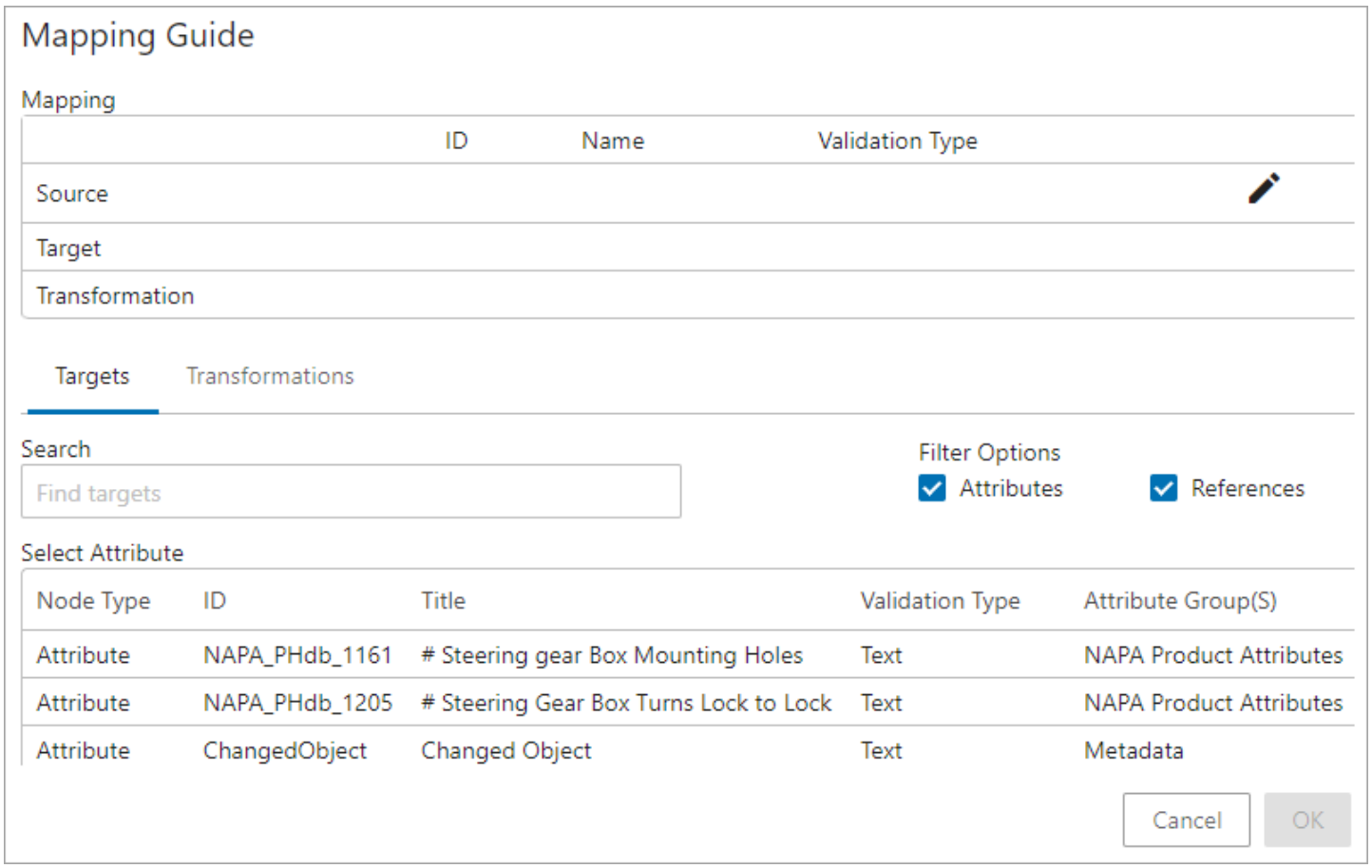
The validity of the Source and the Target attribute shall not have any mismatch.

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, refer to topic **Executing Mapper Configuration Setup Entity** within this guide.

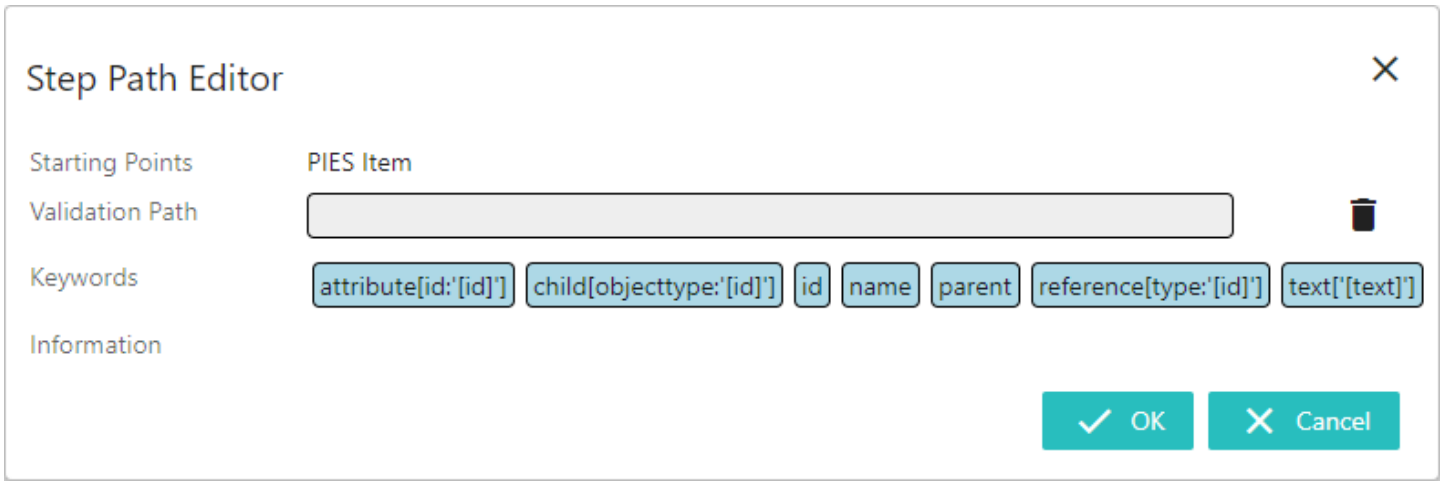
Within the Object to Object Mapping plugin, select the 'On Target' option in the 'Mapping type' field under the Target Definition flipper.




1. Click the ‘Add object mapping’ icon, and the Mapping Guide window will display.

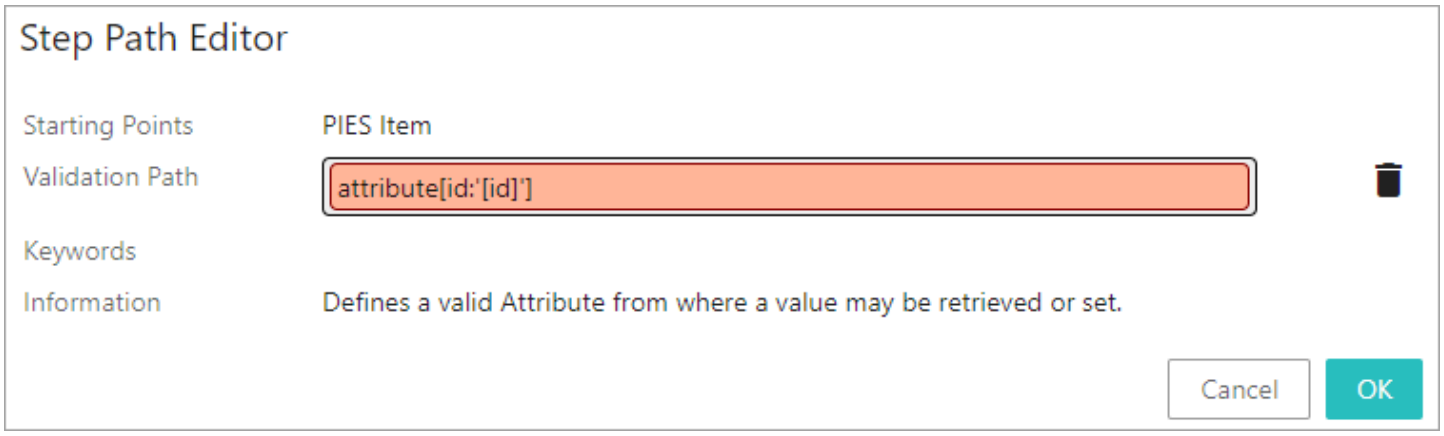



2. Click on the Edit icon ( ) in the Source field, and the Step Path Editor window will display. For more information on defining the Source STEP path, refer to topic **Mapping Validation Path Functionality** within this guide.



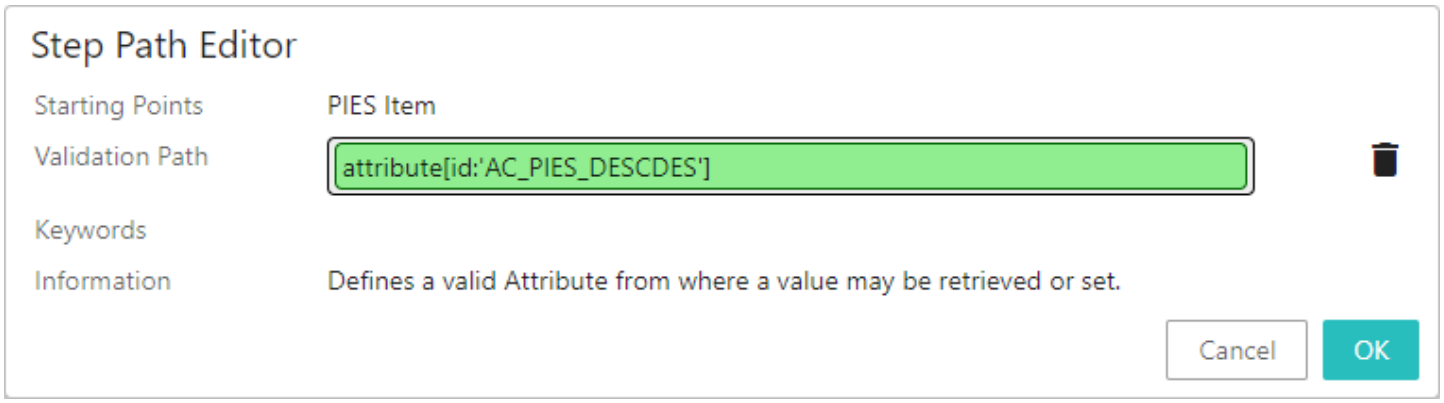

3. Double click on add the **attribute[id:'[id]']** element from the Keywords field. The selected element will be populated in the Validation Path field (as shown below). The red colored field depicts that further

configuration is required.

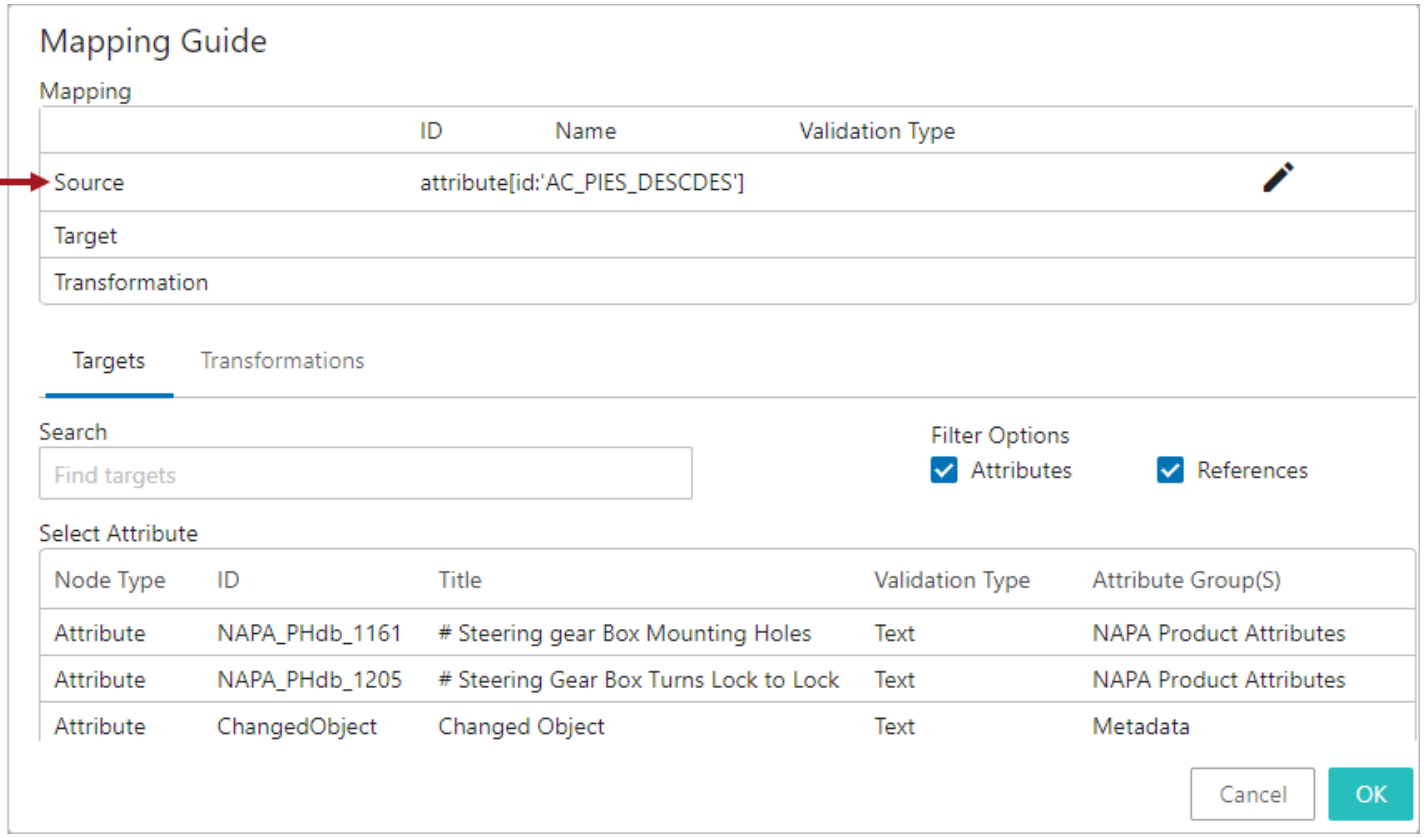



4. Click on the empty red colored **attribute[id:'[id]']** element available within the Validation Path field and select the Source attribute from which the value will be copied over to the Target object. In this example, the attribute with id 'AC_PIES_DESCDES' is selected.

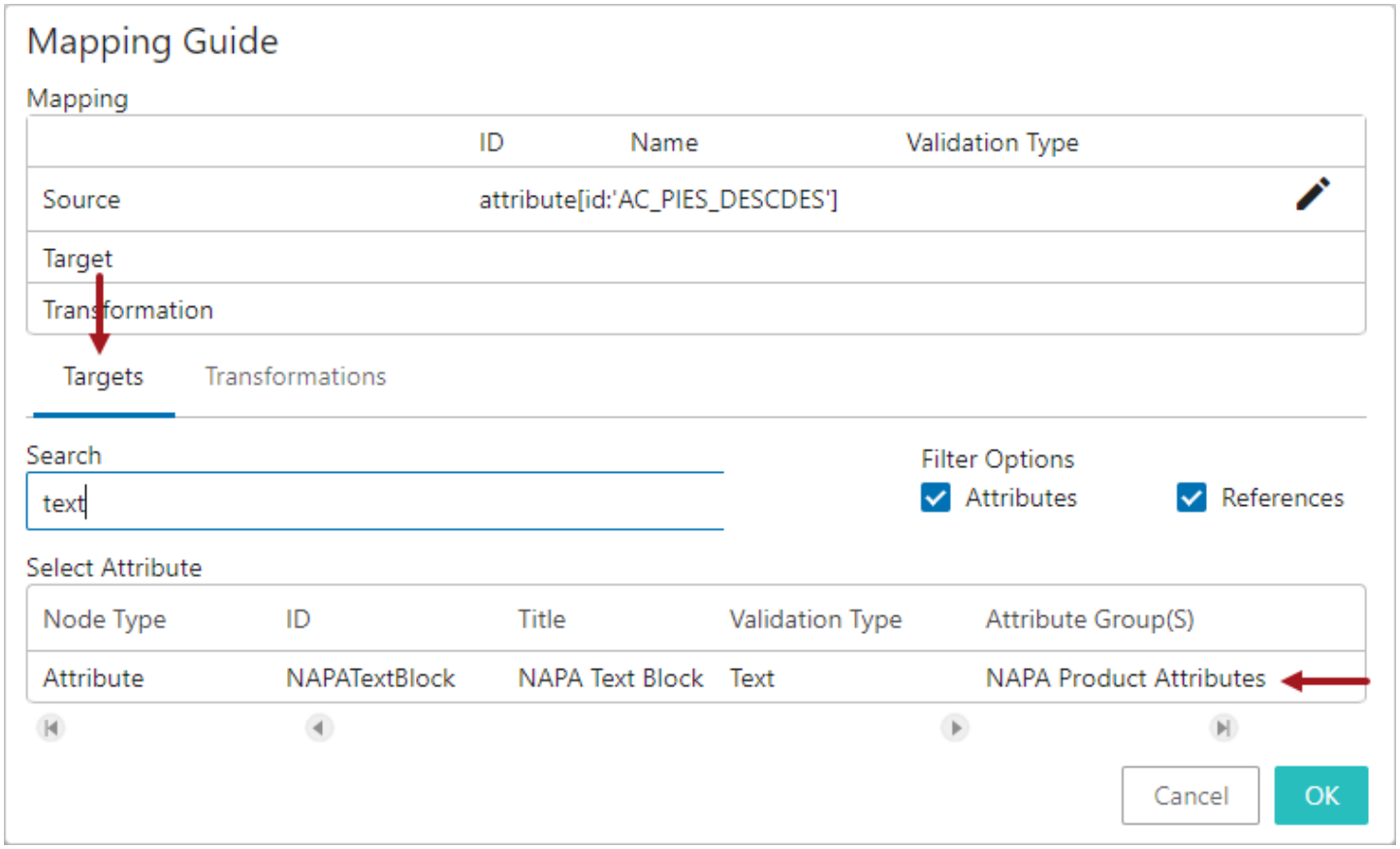
   Once populated, the elements in the Validation path turn green in color as displayed below.




5. Click **OK** and the selected element with the defined data will be populated in the Source field.

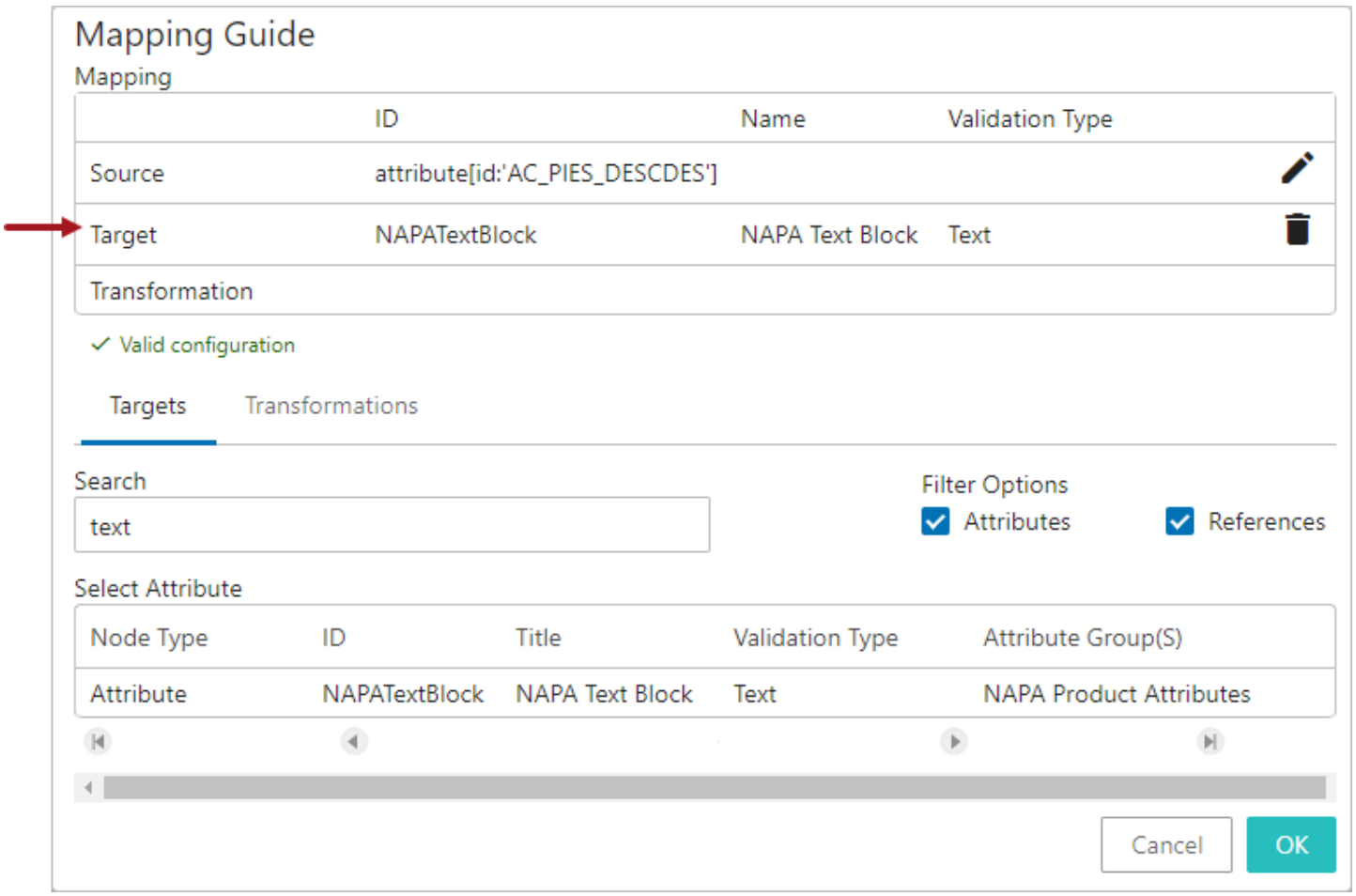



6. With the Targets tab selected, search and select the attribute of the Target object where the retrieved value from the Source object shall reside.

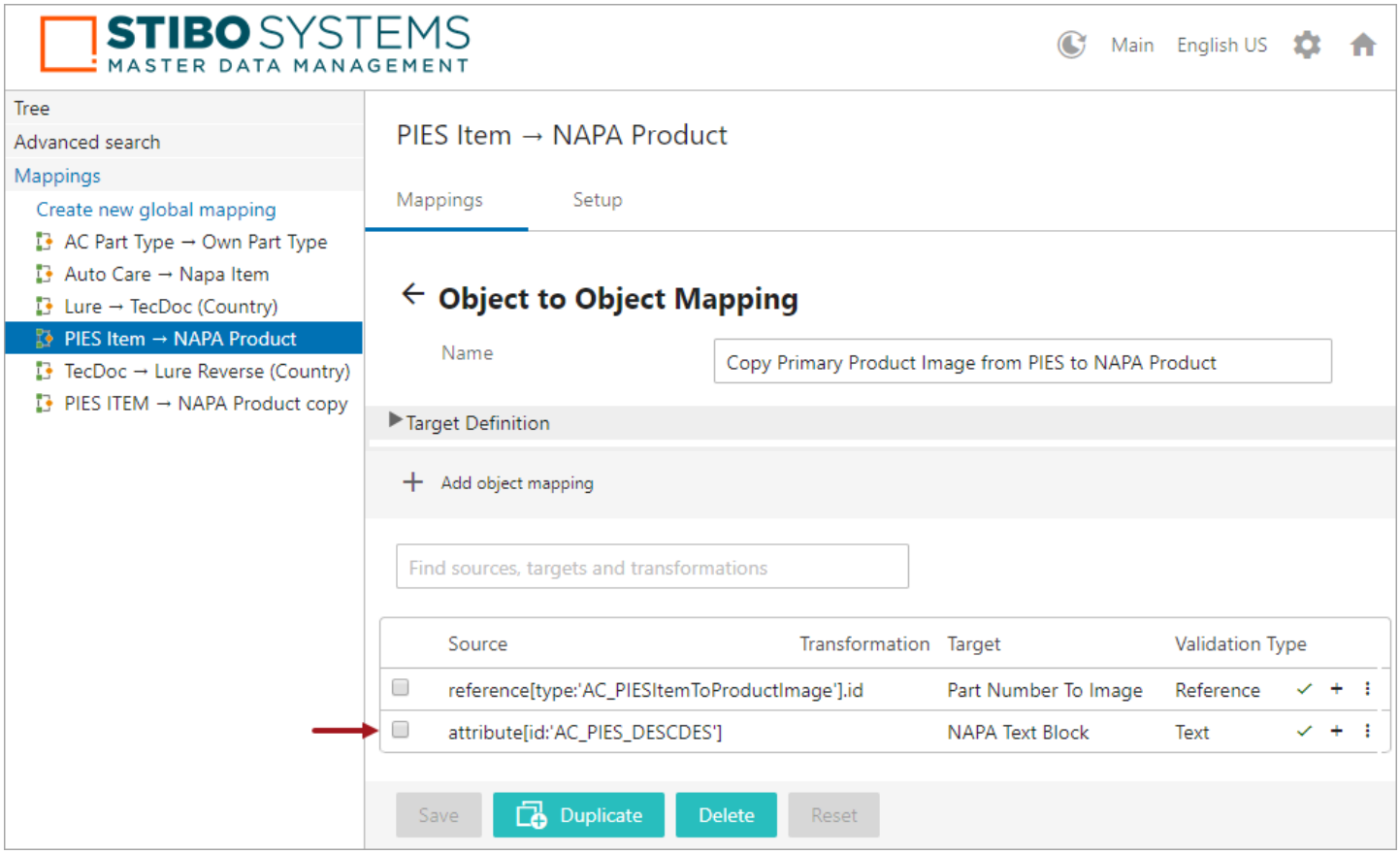



**Note:** The Source object attribute and the Target object attribute should have the validity match.

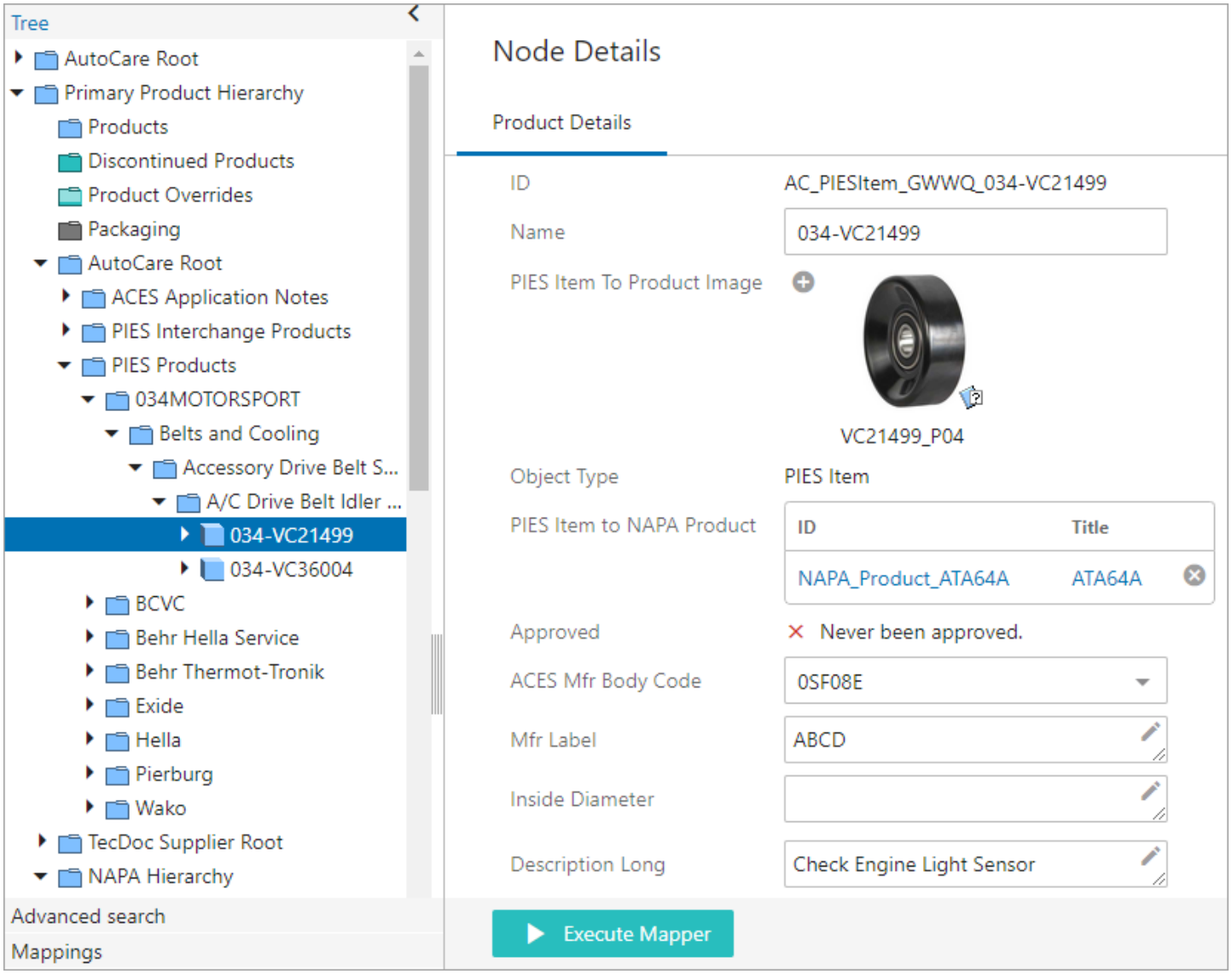
Once selected, the selected attribute will be populated in the Target field (as shown below).




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.




8. 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, refer to topic **Executing Mapper Configuration Setup Entity** within this guide.




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

- AutoCare Root
- Primary Product Hierarchy
  - Products
  - Discontinued Products
  - Product Overrides
  - Packaging
  - AutoCare Root
    - ACES Application Notes
    - PIES Interchange Products
    - PIES Products
  - TecDoc Supplier Root
  - NAPA Hierarchy
    - NAPA Comment Hierarchy
    - NAPA Interchange Product Hi...
    - NAPA Product Hierarchy
      - ATA Fuel Injector - New
        - ATA64A
      - RPC 63858
      - Wiper Blade - Bosch Icon - ...
- TecDoc Supplier Price Lists
- Import Flow Root

Advanced search

Mappings

## Node Details

Product Details

| | |
|---|---|
| ID | NAPA_Product_ATA64A |
| Name | ATA64A |
| Object Type | NAPA Product |
| Approved | ✕ Never been approved. |
| Part Number To Image | VC21499_P04 |
| # Steering gear Box Mounting Holes | |
| Color | |
| NAPA Package INfo | |
| * NAPA Part Number | |
| NAPA Text Block | Check Engine Light Sensor |

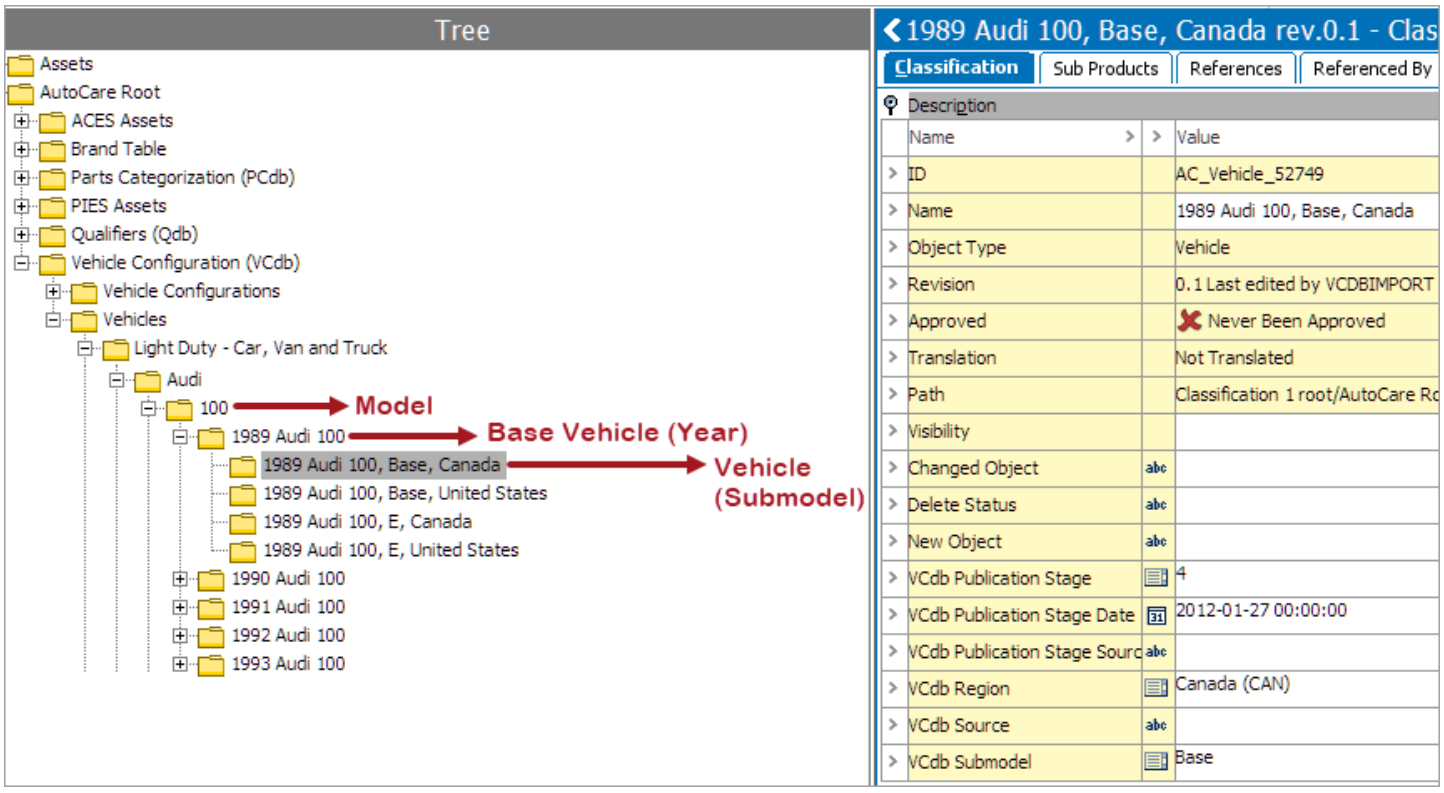
### Use Cases - Copy Data from One Object to Child Object of Another Object

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.

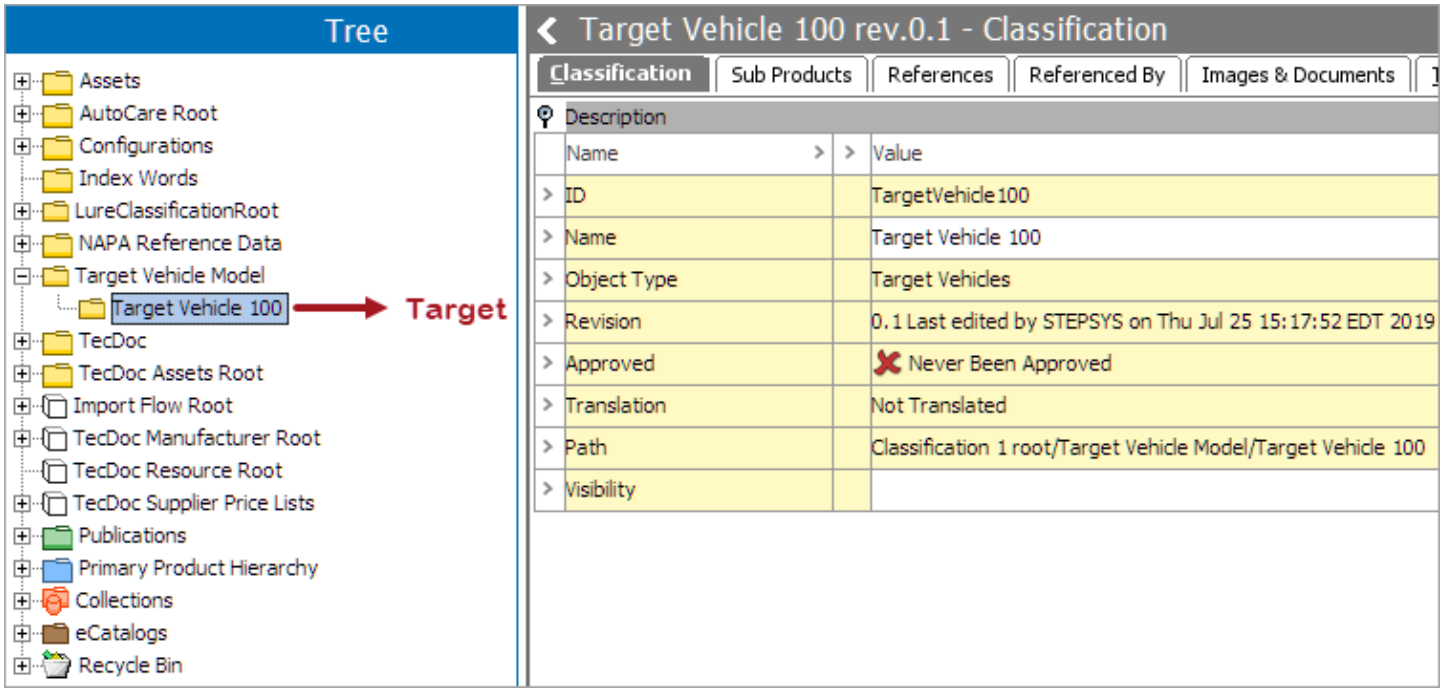
#### Example One

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).



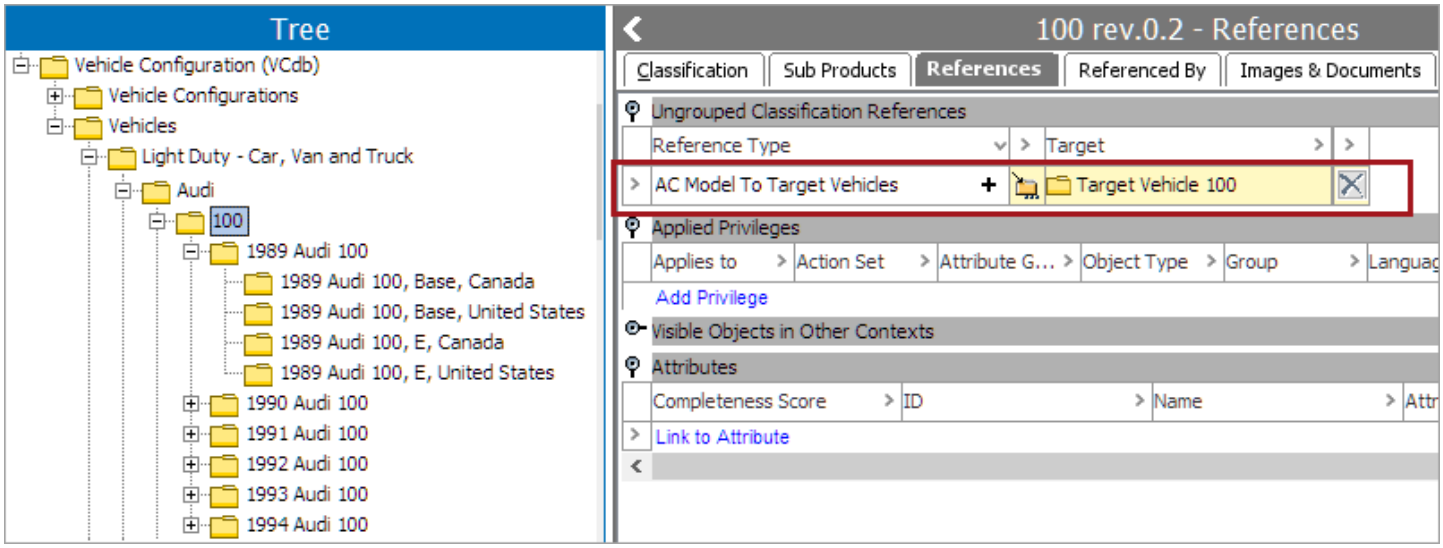

Below is the image displaying the Own hierarchy of Target object (Target Vehicle hierarchy).



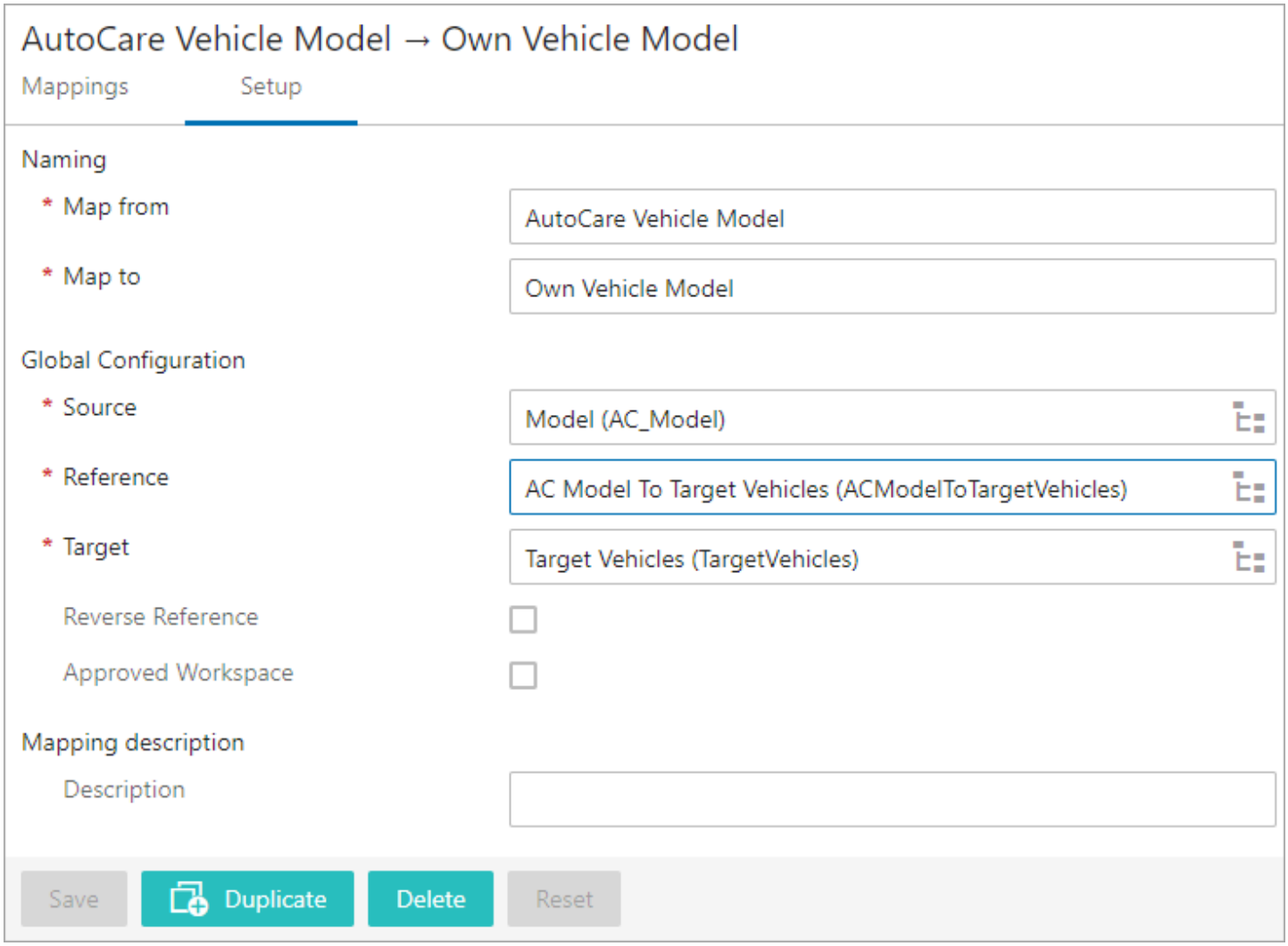

Following are the detailed steps describing to achieve the same via the mapping solution.

## 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, refer to the **Reference Types** topic within the **System Setup** section of **STEP Online Help**.

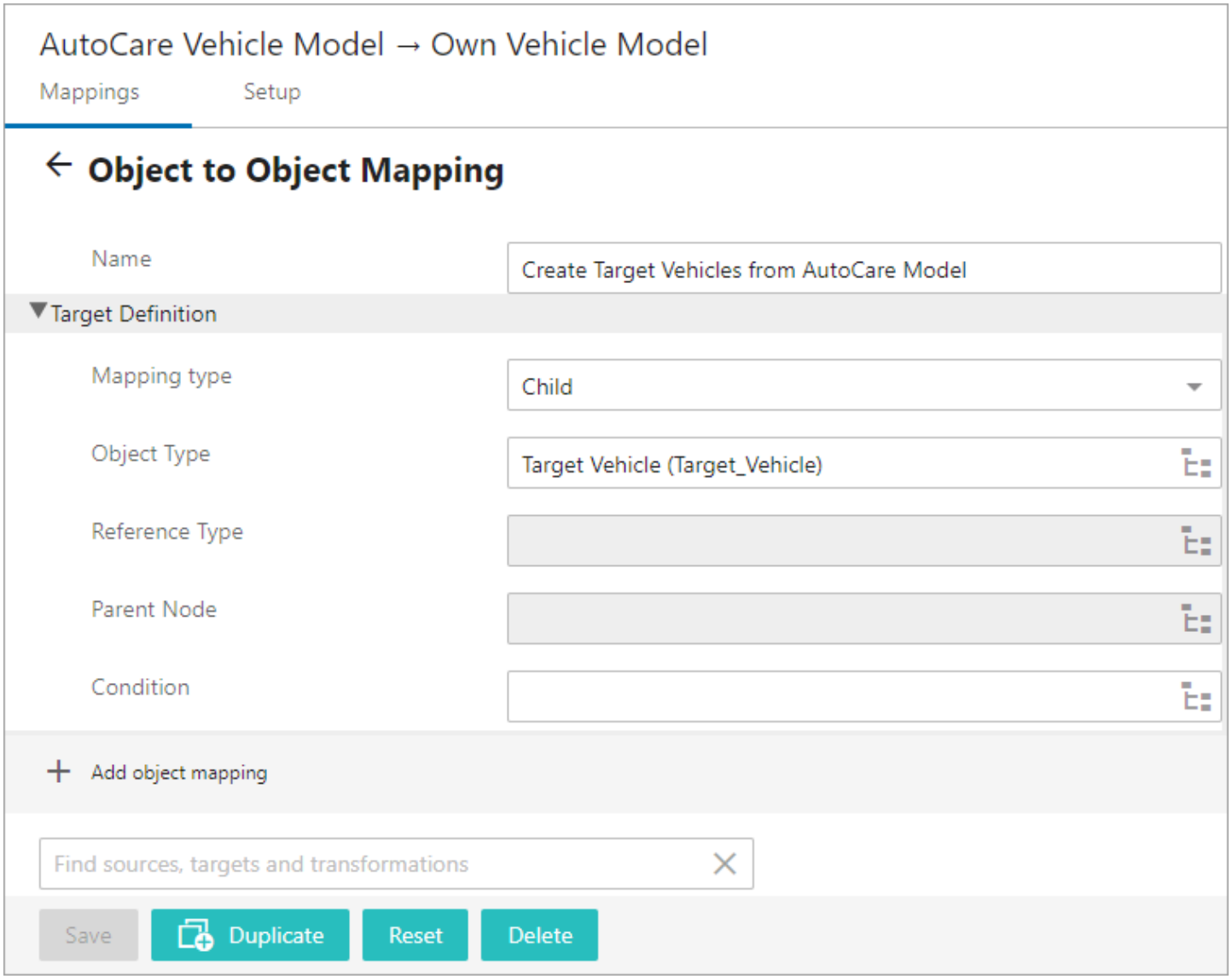


In the Web UI, create and configure the Mapper Configuration and define the Source and Target objects and the Reference type to follow to determine the relationship between the Source and Target object. For more information on configuring and setup procedure for the Mapper Configuration, refer to topic **Configuring Mapper Configuration Setup Entity** within this guide.

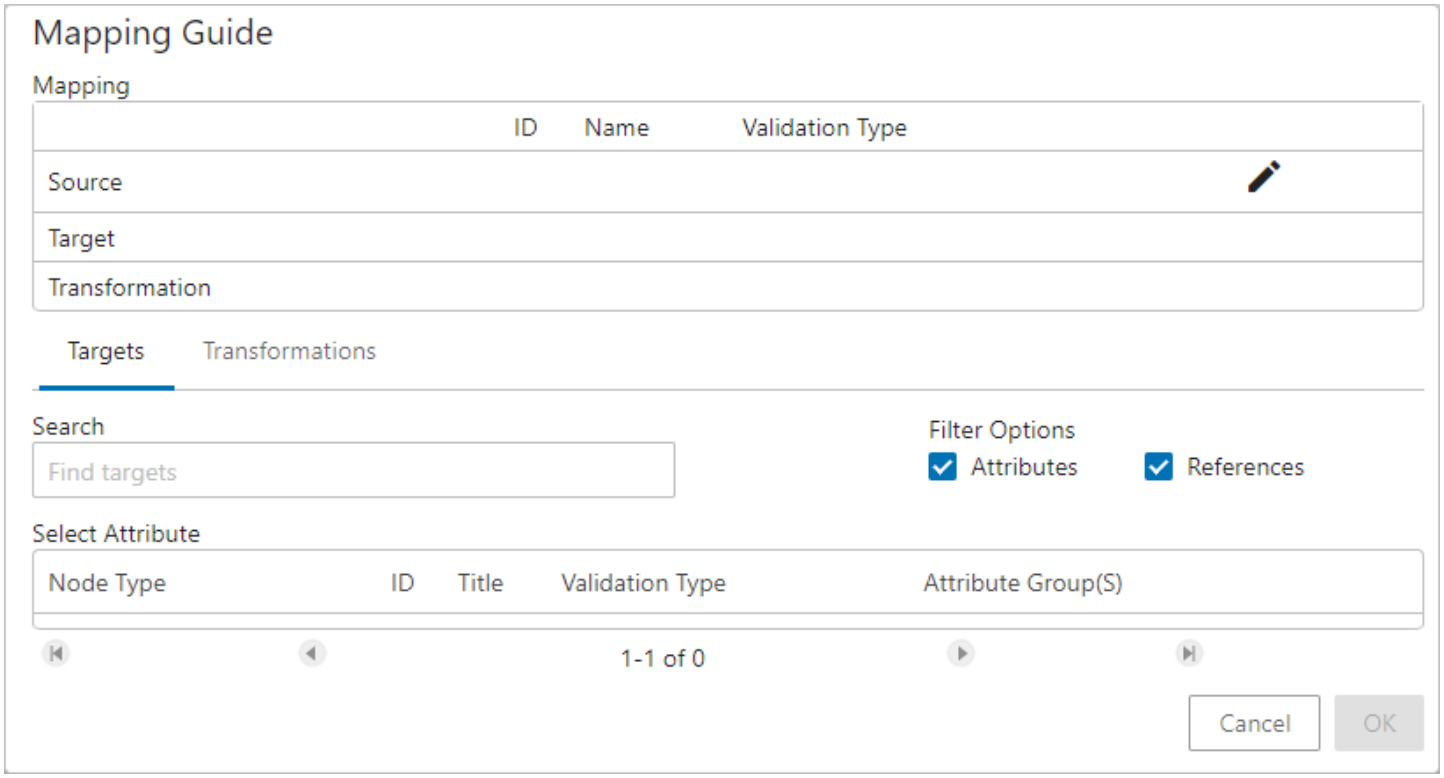



Configure an Initiate Business Action button within the Product Details screen of the Source object which shall execute the Mapper Configuration. For more information on executing the Mapper Configuration, refer to topic **Executing Mapper Configuration Setup Entity** within this guide.

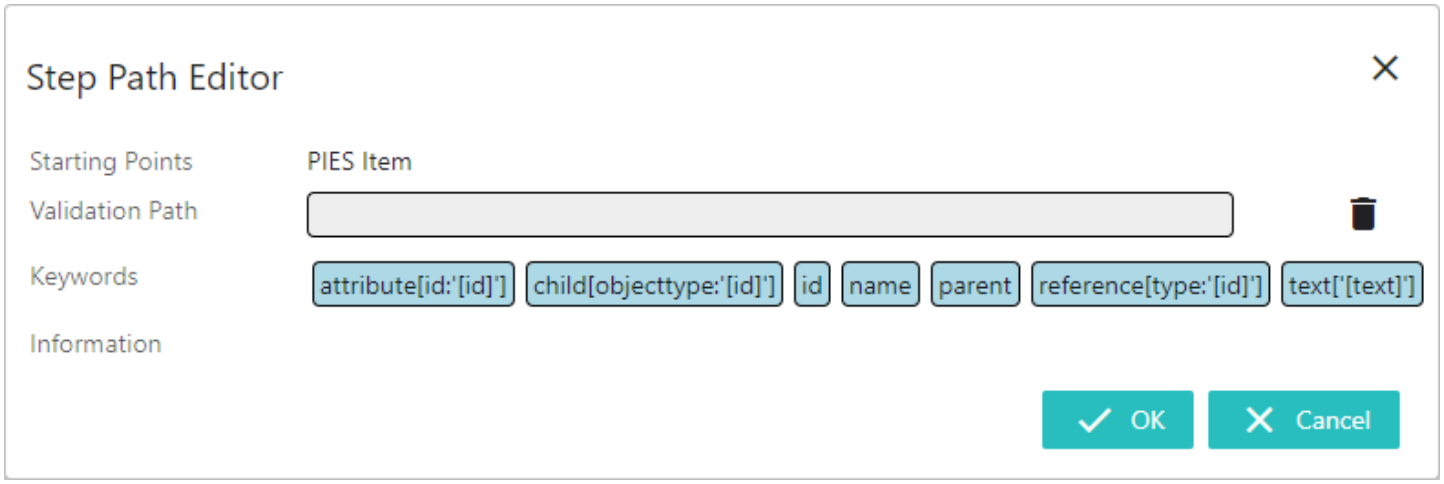
1. Add a new Object to Object Mapping plugin (or edit the one that exists) in the required Mapper Configuration.
2. Type in a suitable name next to Name field. In this example, the Name field is entered with the text 'Create Target Vehicles from AutoCare Model.'




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).

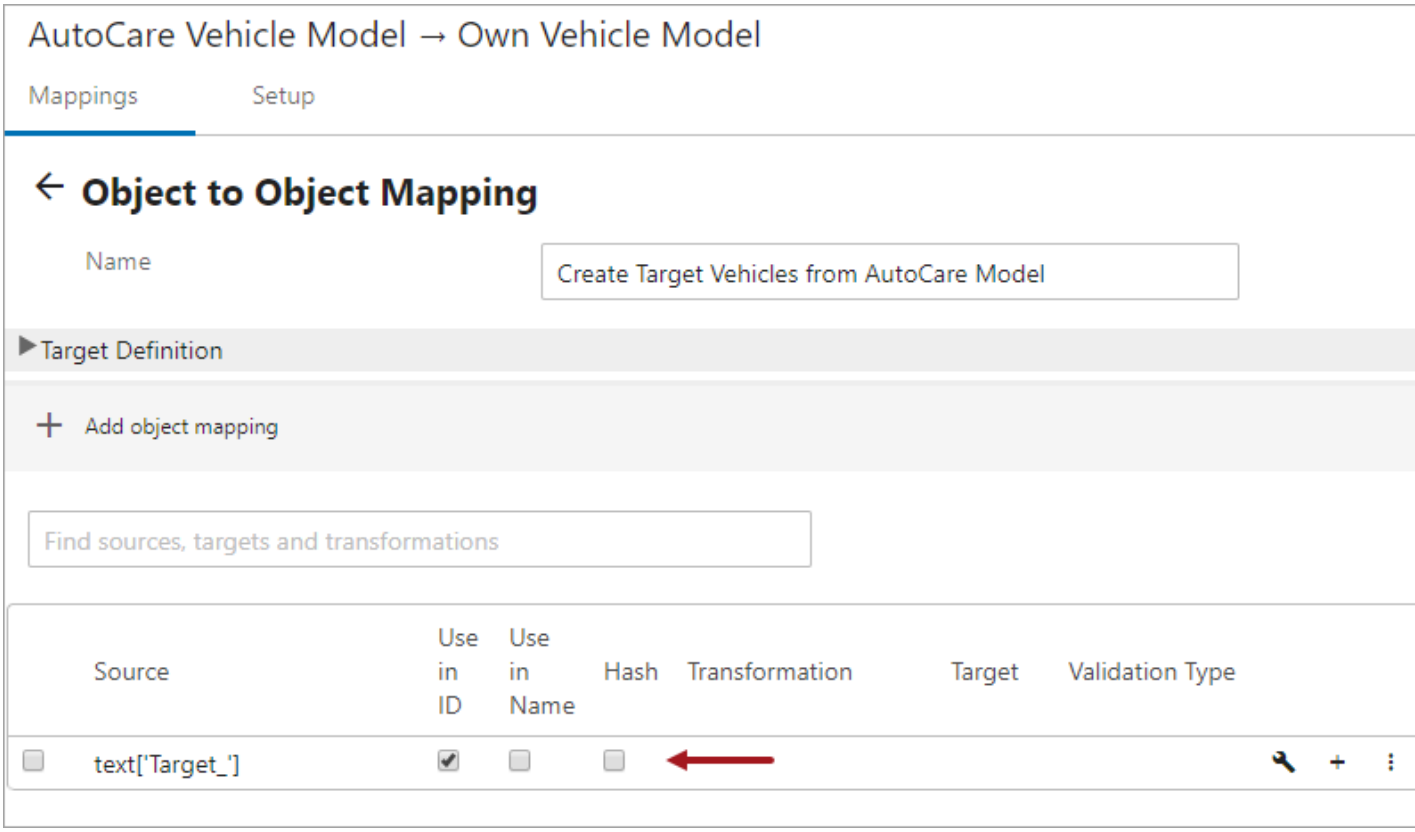



6. Click on the Edit icon ( ) in the Source field, and the Step Path Editor window will display. For more information on defining the Source STEP path, refer to topic **Mapping Validation Path Functionality** within this guide.



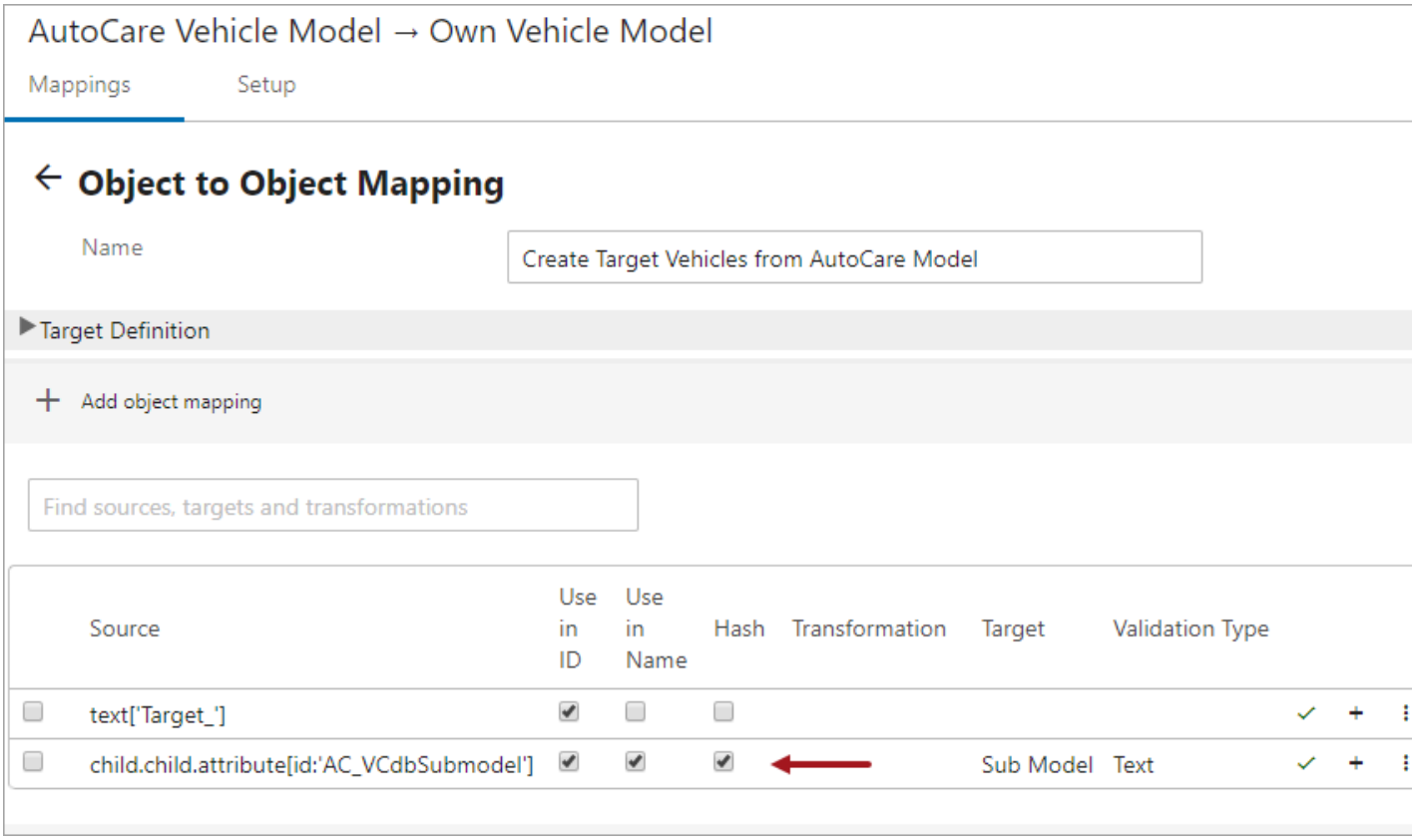

7. For this use case, since there are multiple functions required to be executed, the following logic is used to accomplish the requirement through the STEP path.
   - To define what to use for the ID of the new objects that will get created.

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.



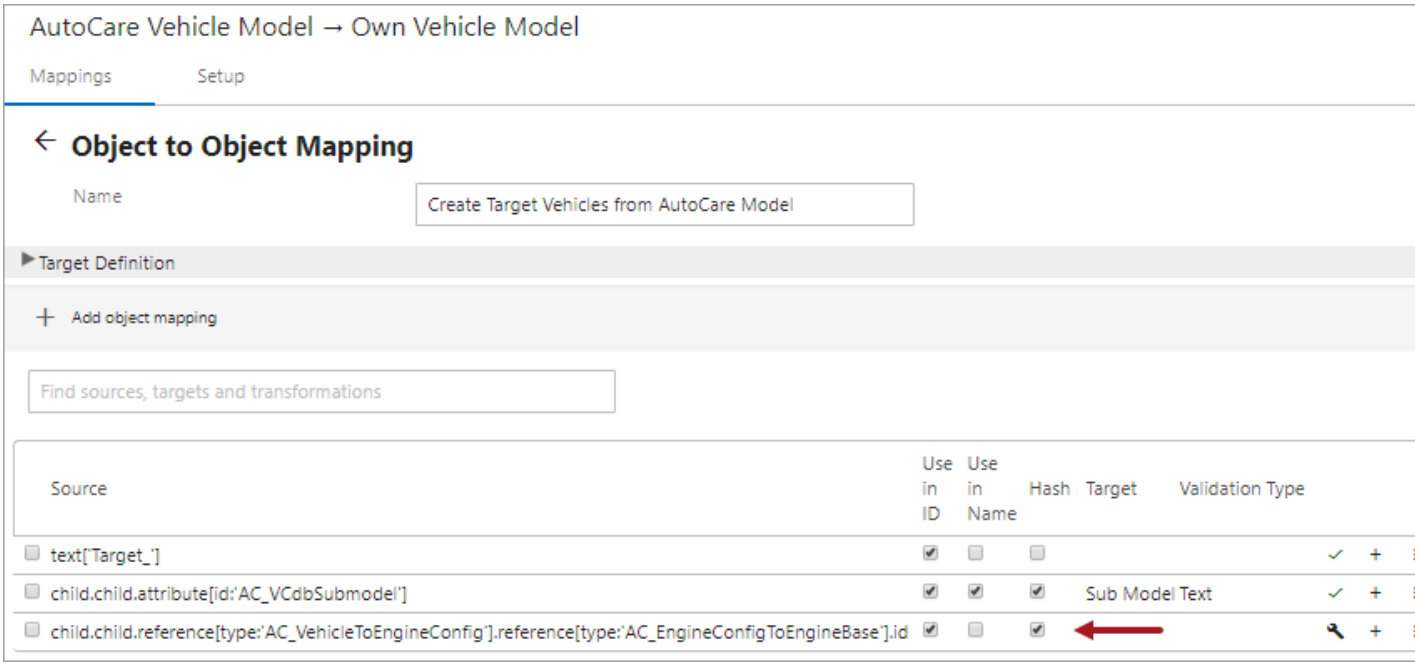

- To define what to use for the Name of the new objects that are getting created, as well as what values to use in the ID to make it unique.

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.



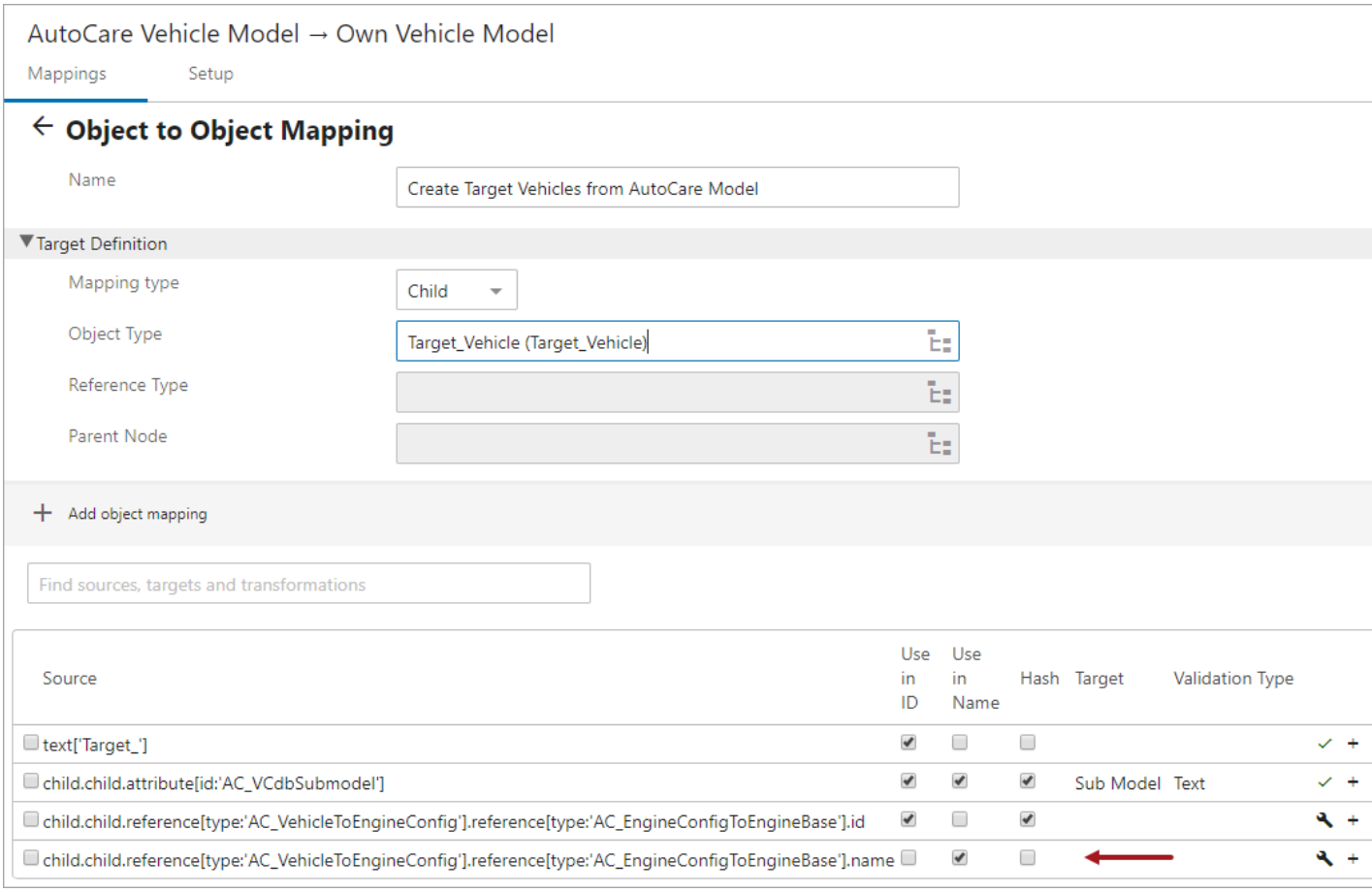

- The use case requires to use the Engine Base ID in the ID of the new objects.

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.



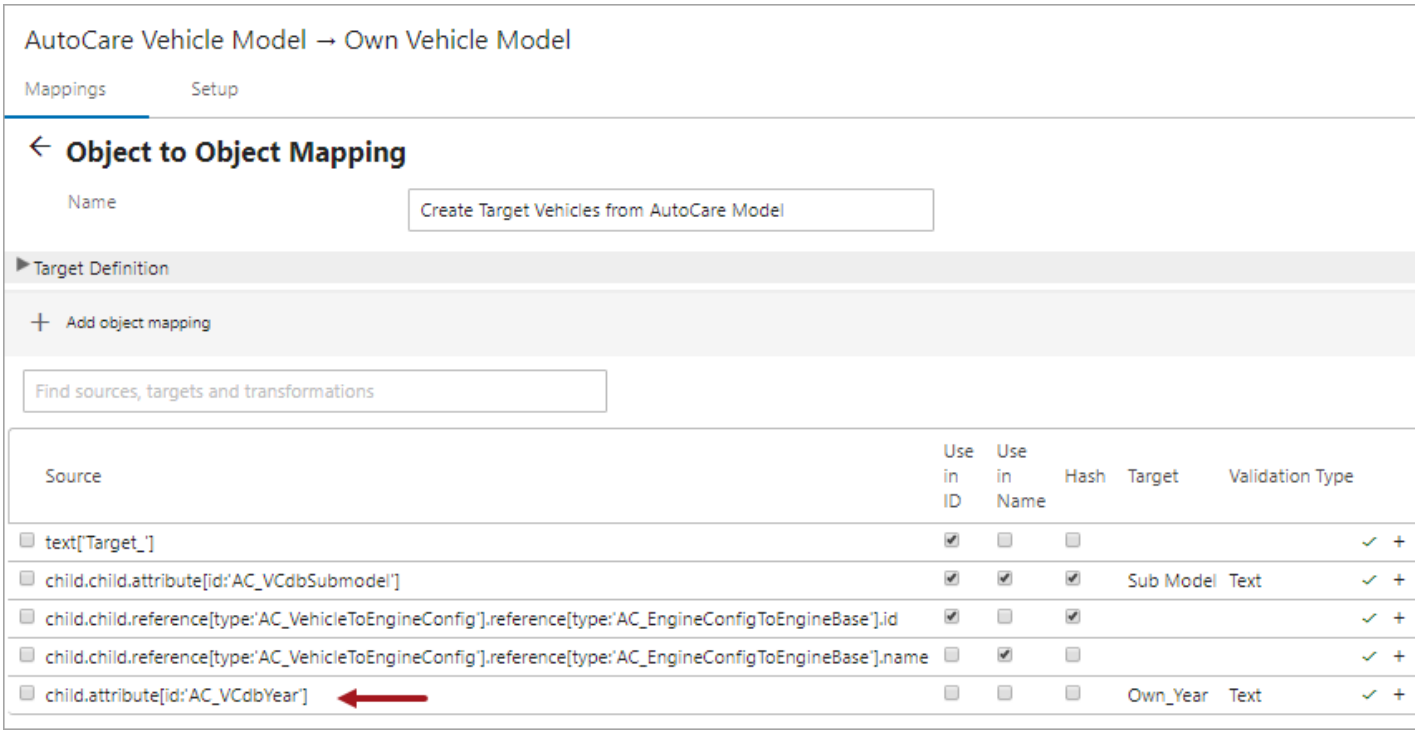

- The use case requires the use of the Engine Base Name in the Name of the new objects.

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).

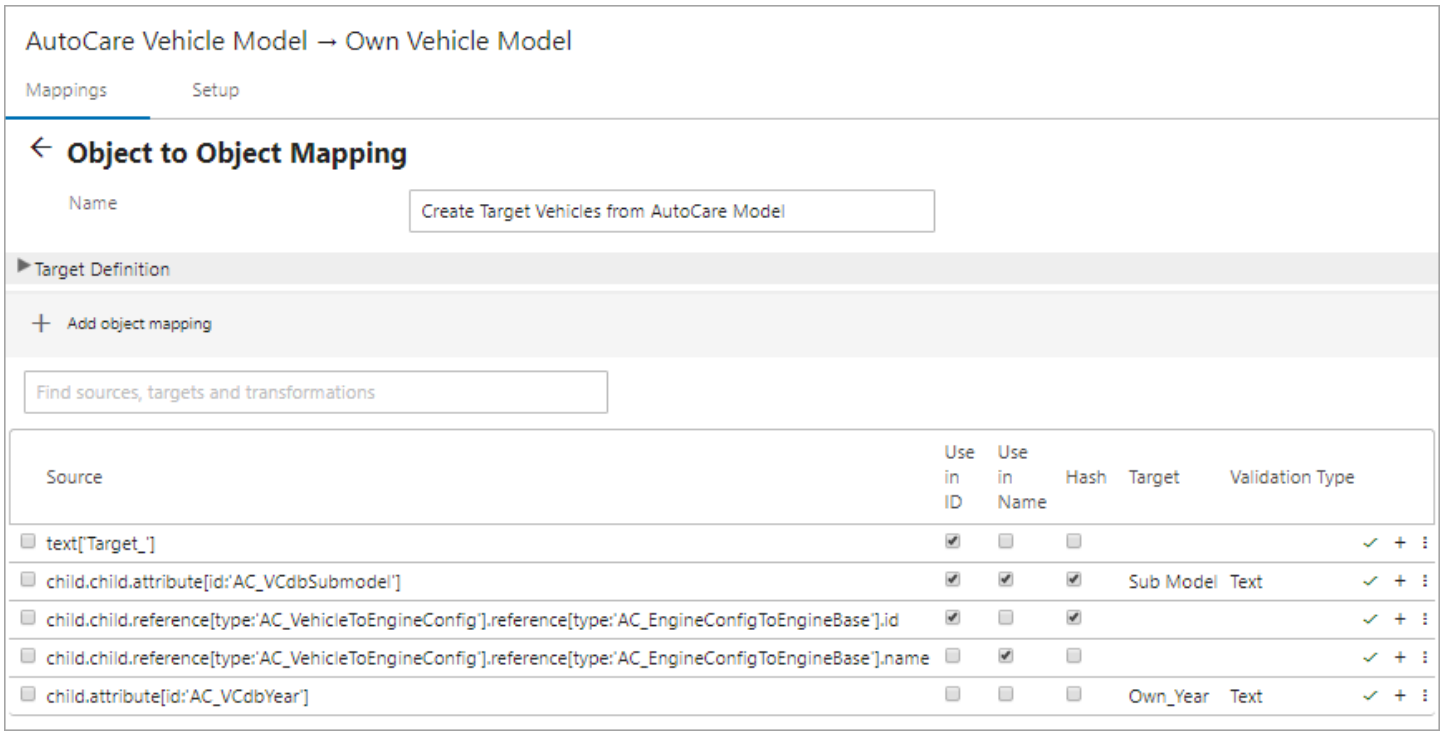



- The use case requires the use of the Year attribute value and store it as an attribute value on the new objects.

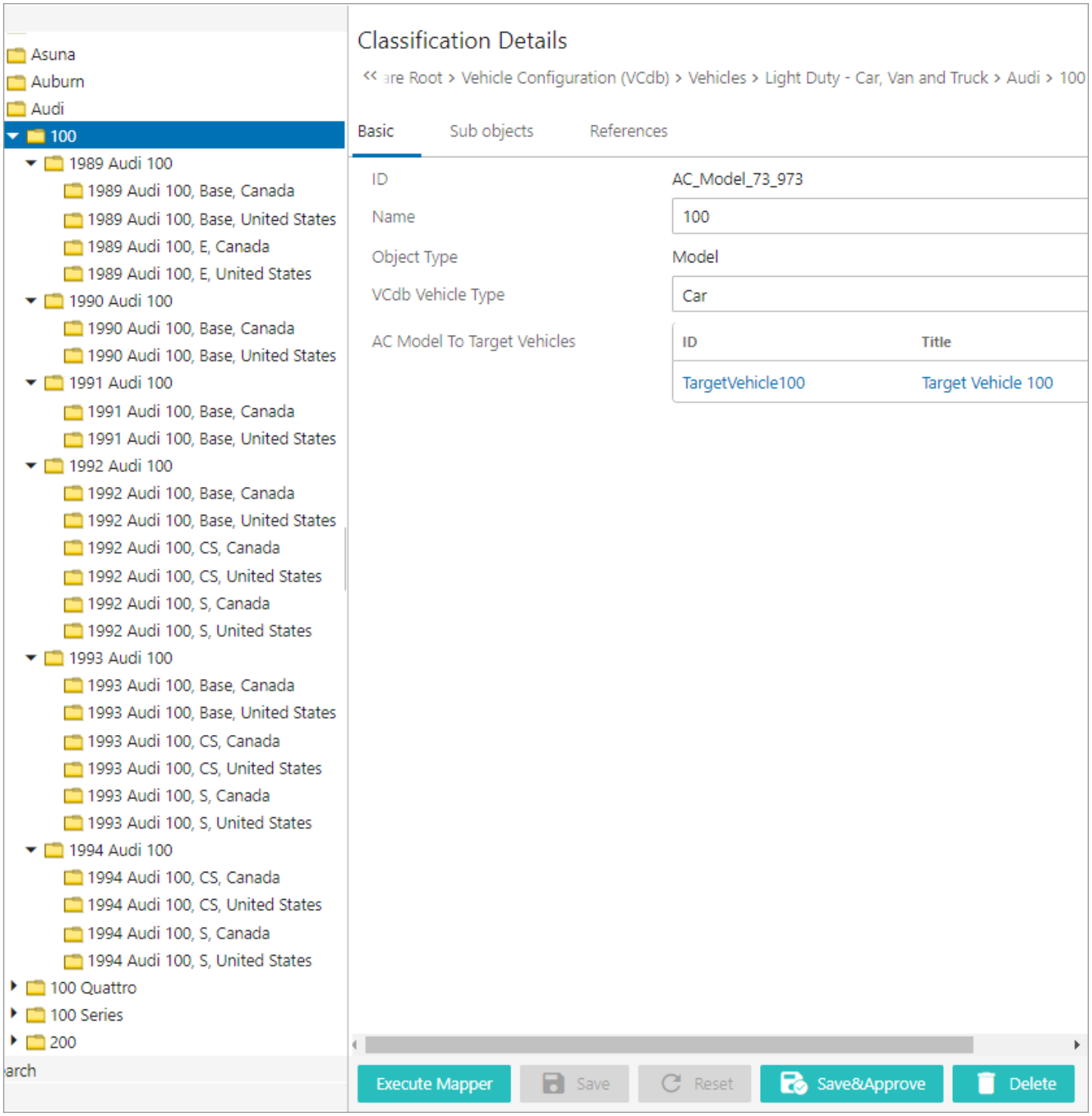
Using the **child[objecttype:'[id]']** and **attribute[id:'[id]']** elements in the STEP path to follow the child object of the Source object, the AC_VCdbYear attribute value is retrieved for Year. Also, the Target attribute of Year is selected to store the value in (as shown below).




8. Click **OK** to save and close the Mapping Guide window. The newly added object mapping row will be listed.



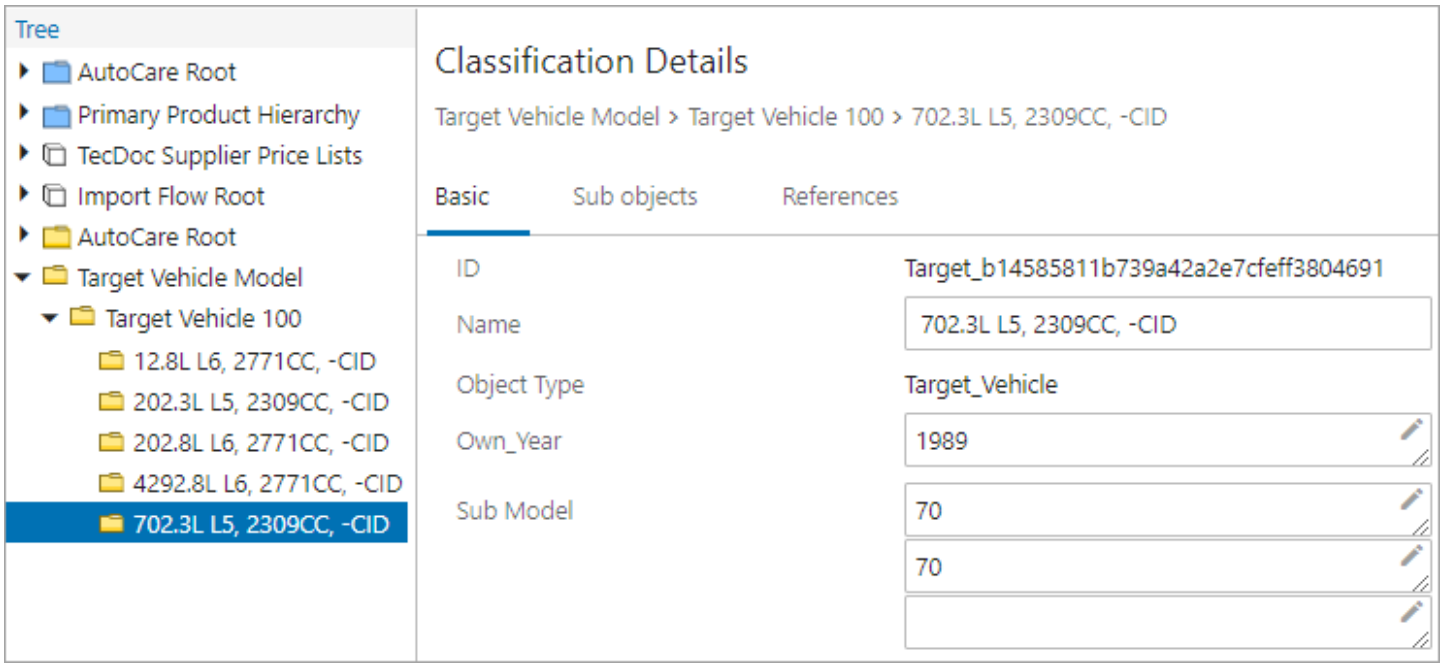

9. In the Web UI, standing on the Source object, execute the Mapper Configuration that holds the mapping plugin. For more information on executing the Mapper Configuration, refer to topic **Executing Mapper Configuration Setup Entity** within this guide.




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.



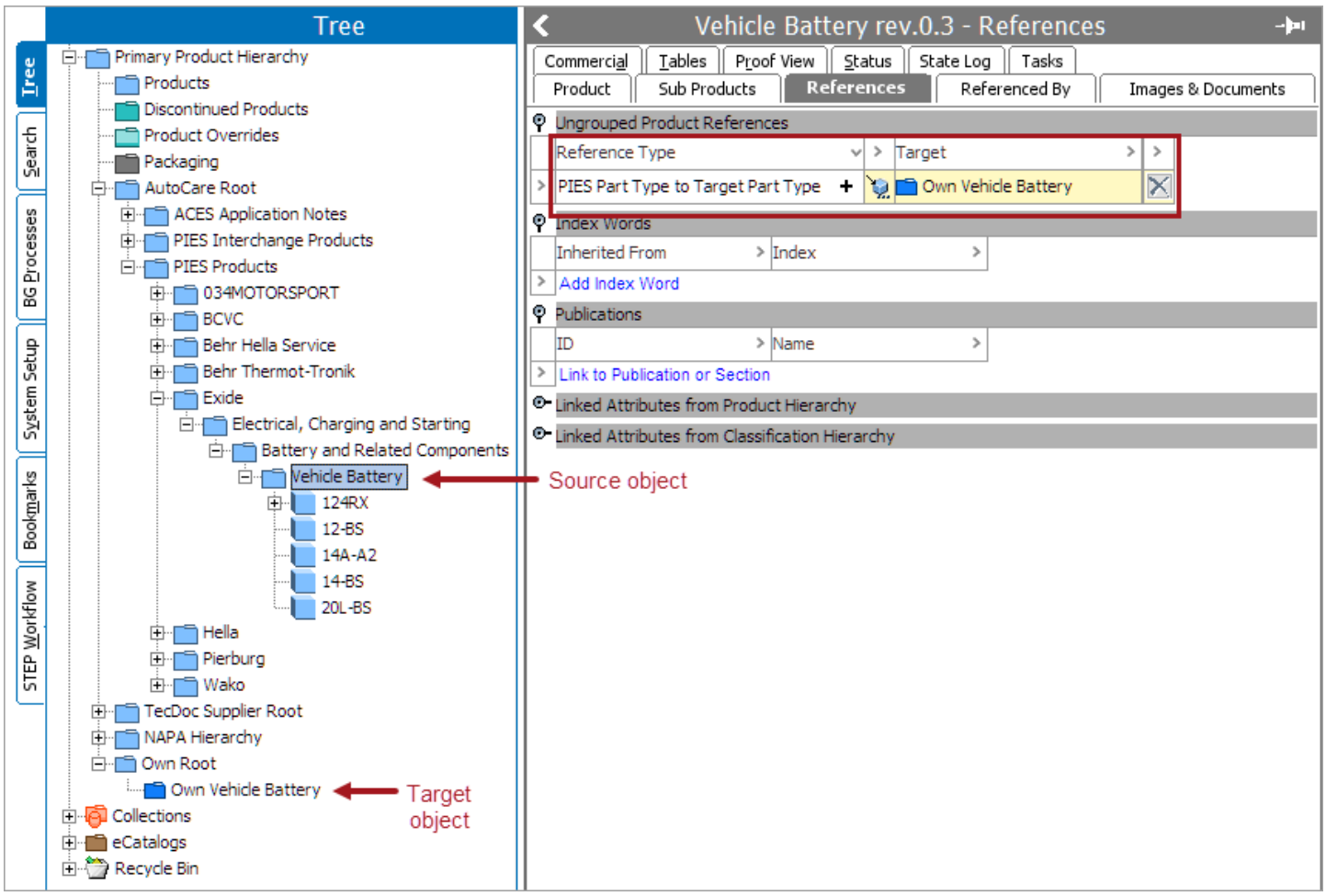

**Example Two**

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.

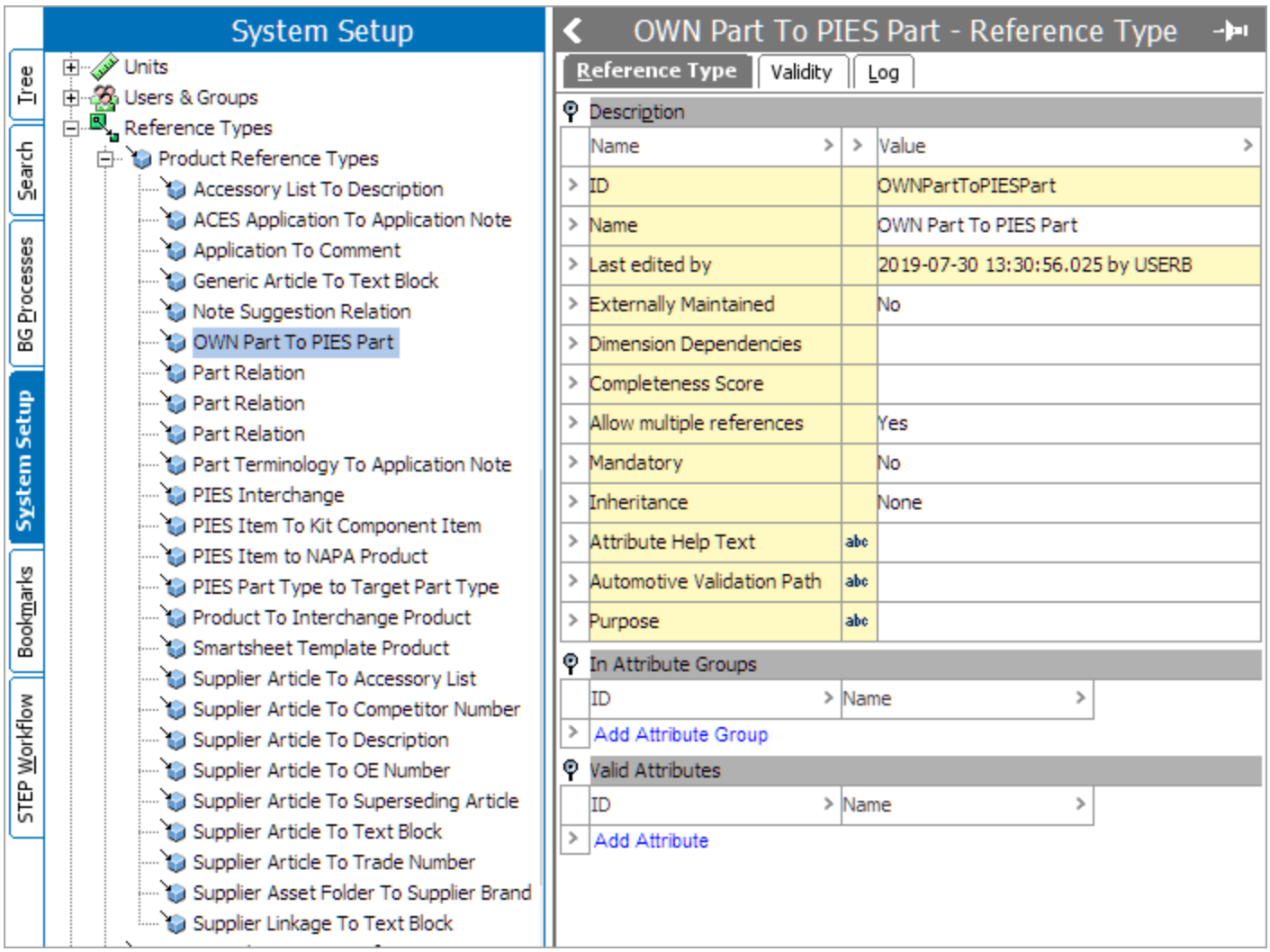
In this example, the AutoCare part type 'Vehicle Battery' is used as the Source object and all PIES Items below 'Vehicle Battery' are used to create new objects below the Own object (example 'Own Vehicle Battery' hierarchy).

## Prerequisites

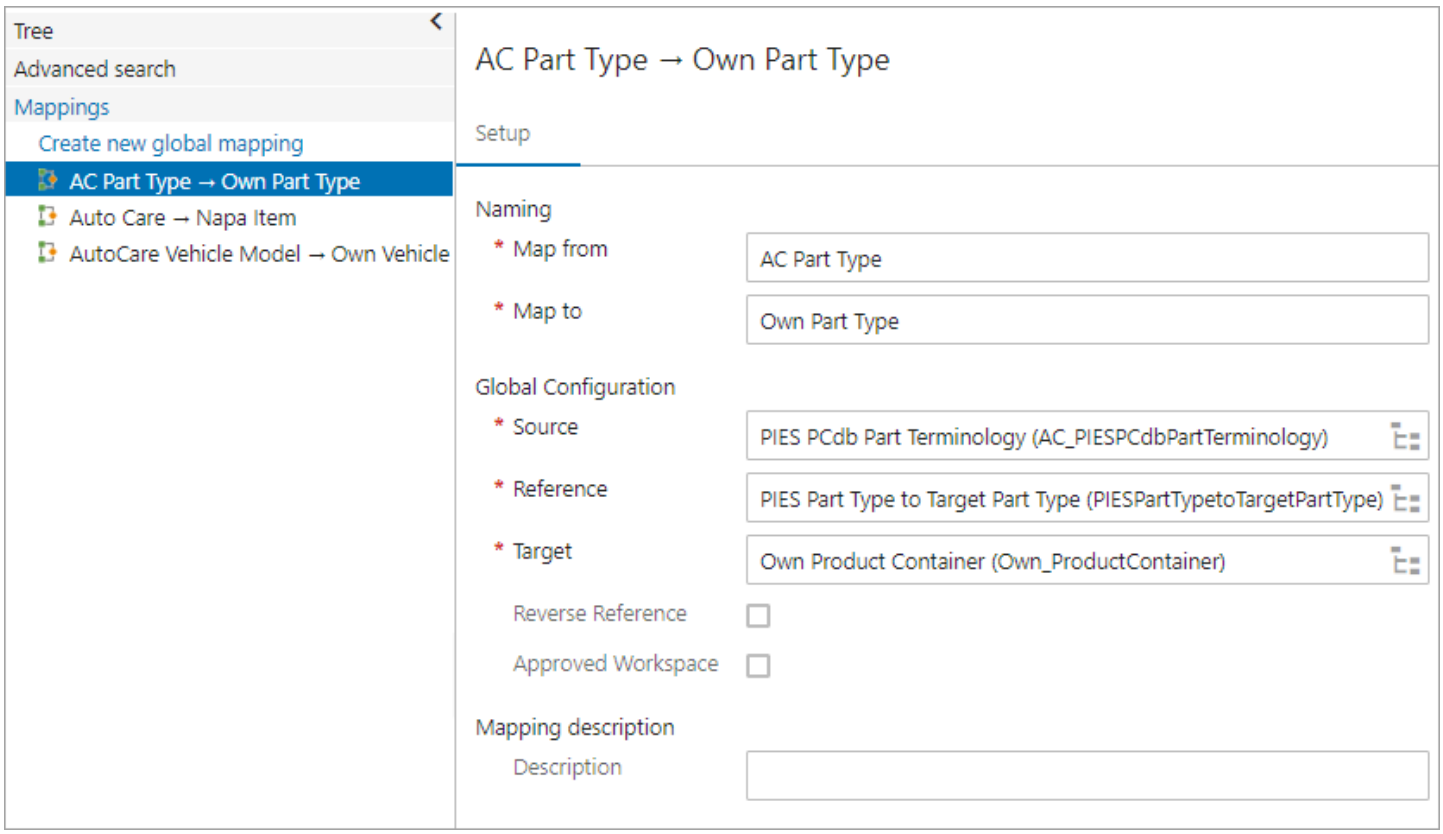
Create a new Product Reference Type to link the PIES Item parent object to the Target parent object and establish the reference between the PIES Item parent part type object to the Own parent part type object that user needs to create the new part numbers below. In this example, the Product Reference 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.




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, refer to the **Reference Types** topic within the **System Setup** section of **STEP Online Help**.

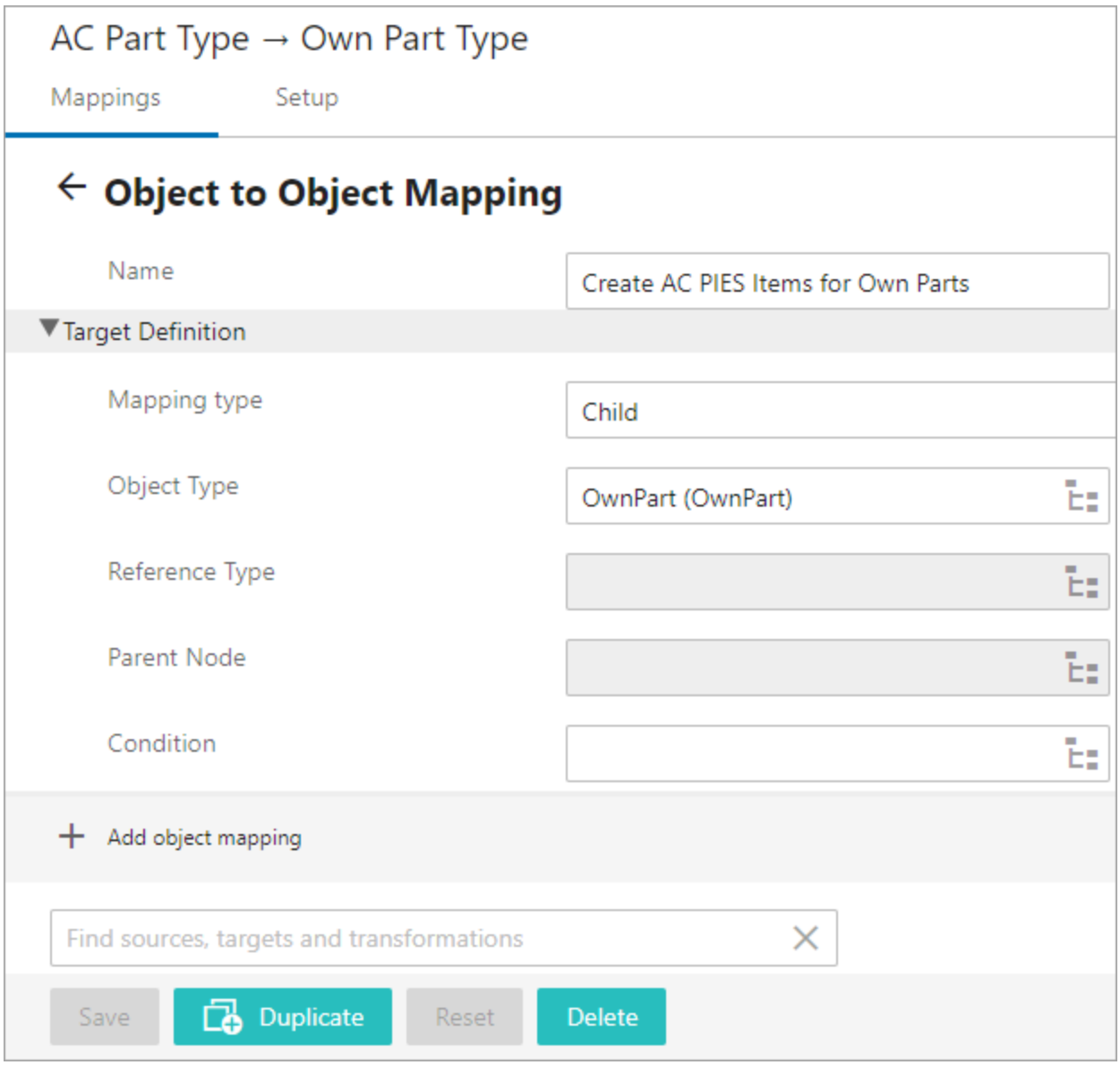



In the Web UI, create and configure the Mapper Configuration and define the Source and Target objects and the Reference type to follow to determine the relationship between the Source and Target object. For more information on configuring and setup procedure for the Mapper Configuration, refer to topic **Configuring Mapper Configuration Setup Entity** within this guide.

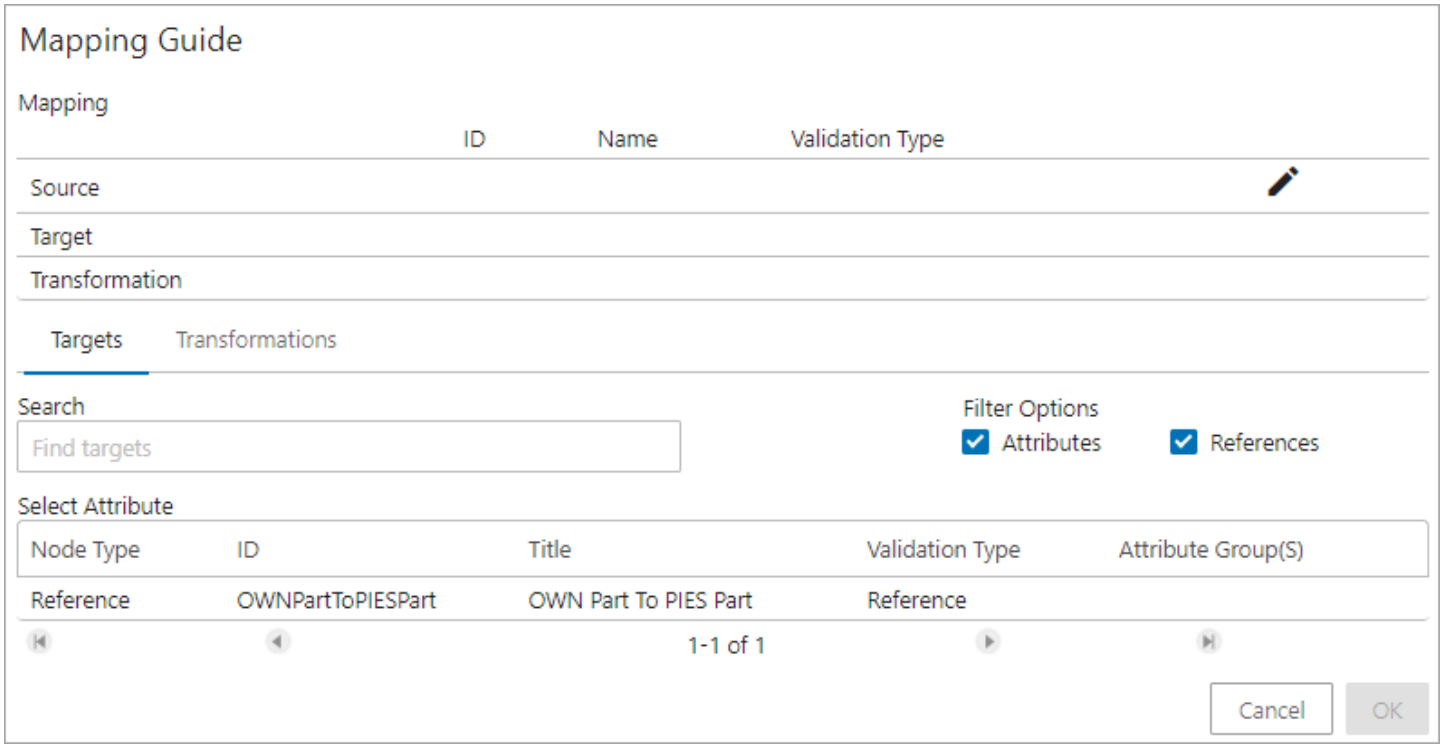



Configure an Initiate Business Action button within the Product Details screen of the Source object which shall execute the Mapper Configuration. For more information on executing the Mapper Configuration, refer to topic **Executing Mapper Configuration Setup Entity** within this guide.

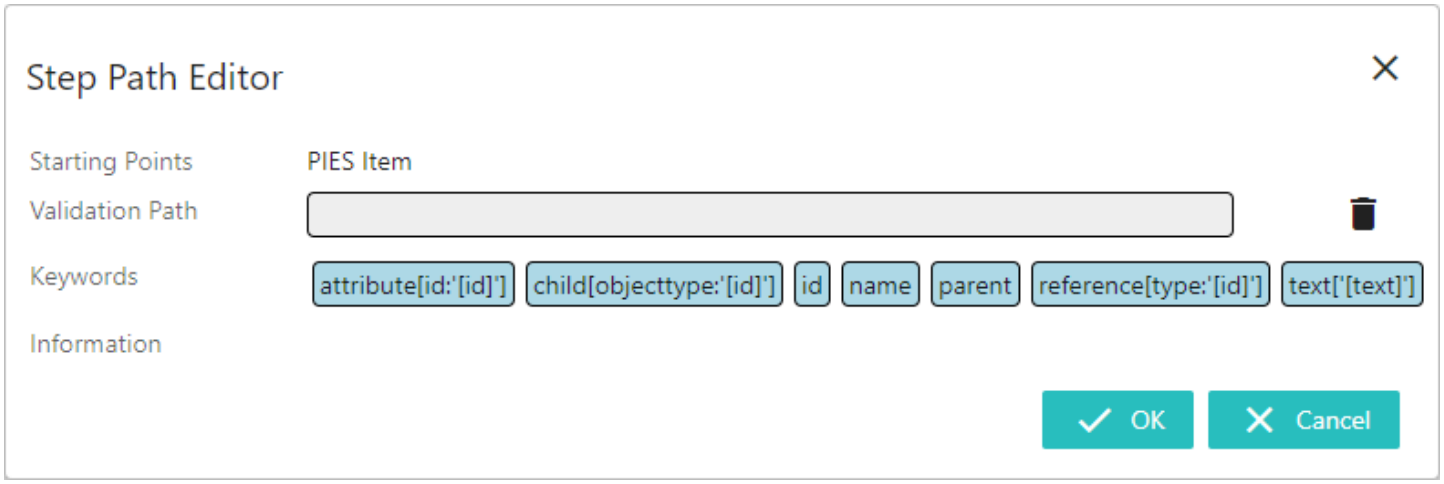
1. Add a new Object to Object Mapping plugin (or edit the one that exists) in the required Mapper Configuration.
2. Type in a suitable name next to Name field. In this example, the Name field is entered with the text 'Create AutoCare PIES Items for Own Parts.'




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 'Own_Partnumber' is selected.
5. Click the ‘Add object mapping’ icon, and the Mapping Guide window will display.

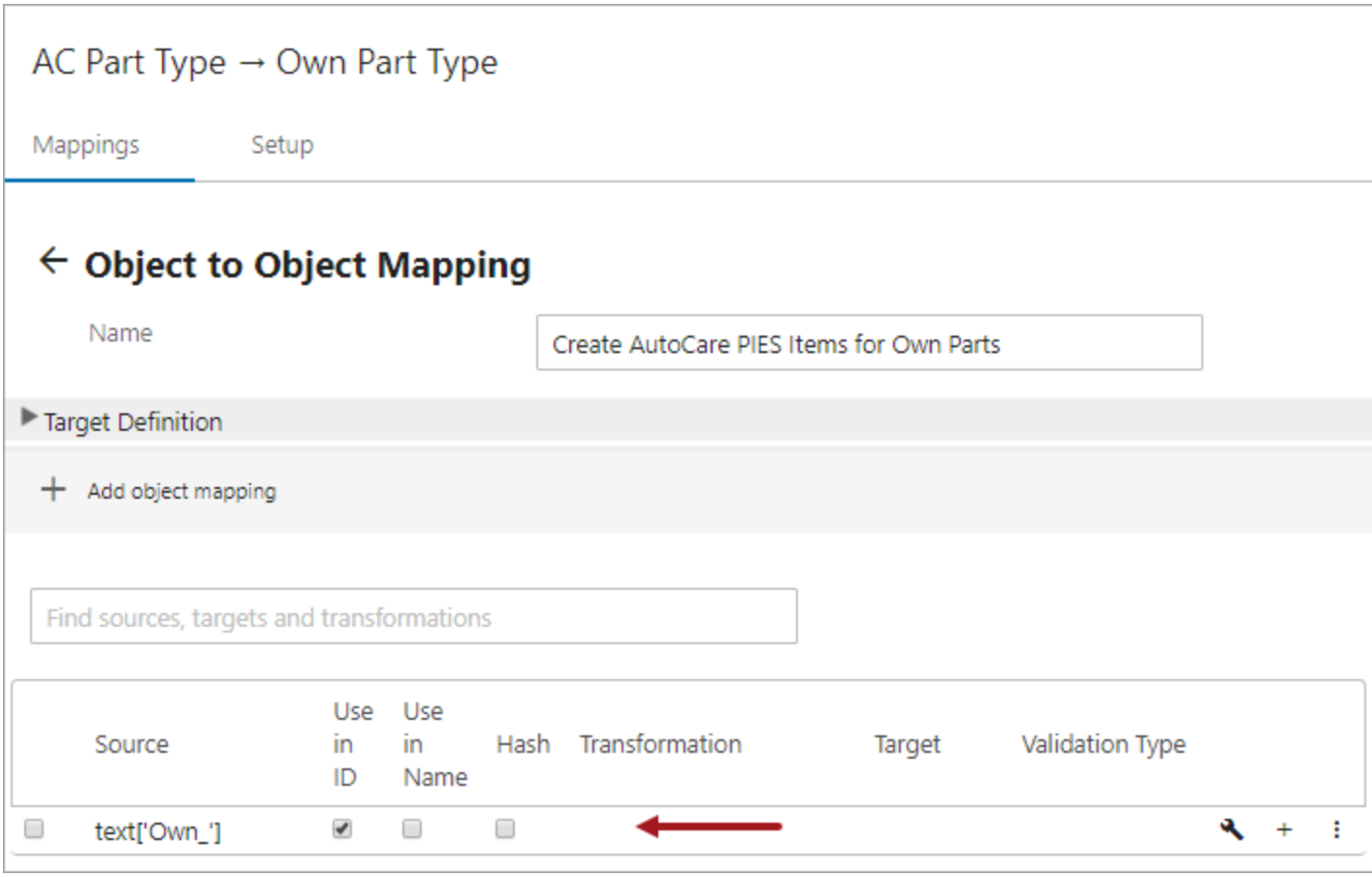



6. Click on the Edit icon ( ) in the Source field, and the Step Path Editor window will display. For more information on defining the Source STEP path, refer to topic **Mapping Validation Path Functionality** within this guide.



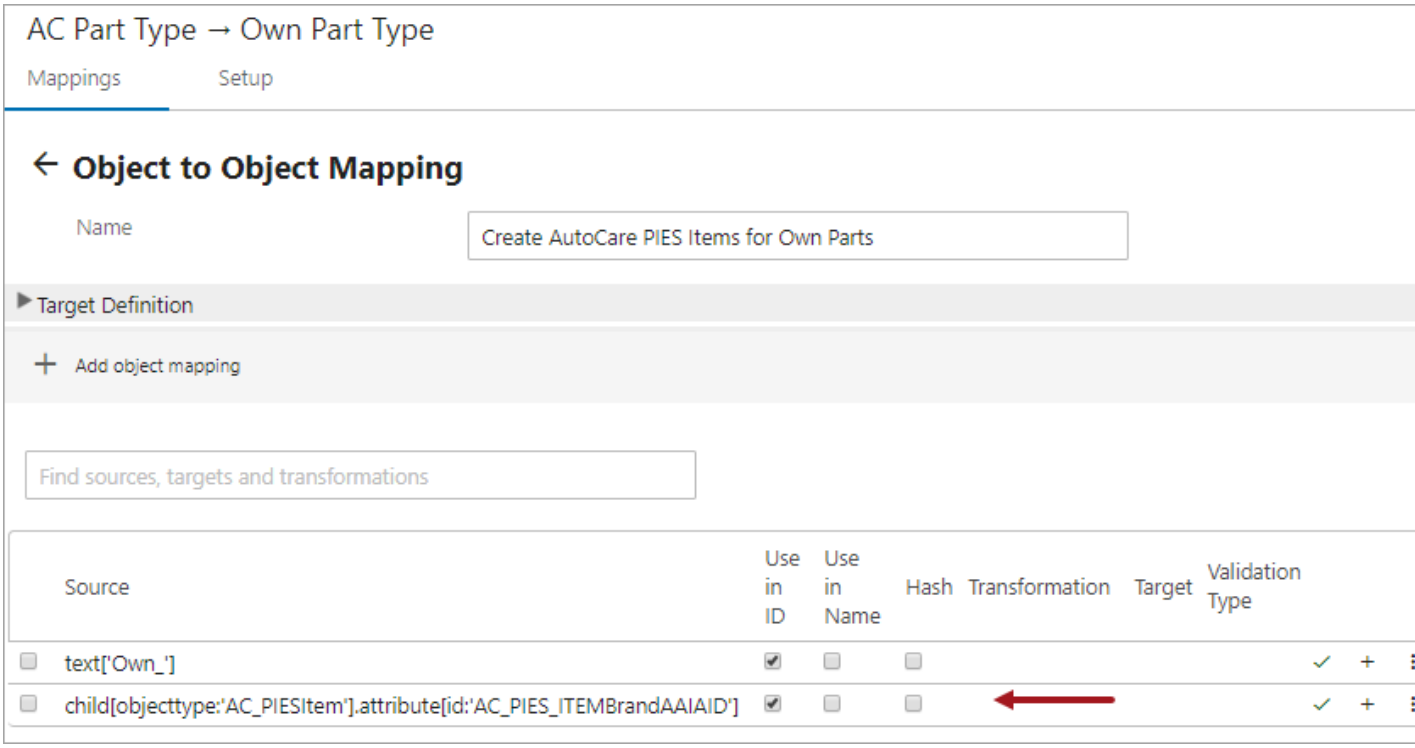

7. For this use case, since there are multiple functions required to be executed, the following series of logic is used to accomplish the requirement through the STEP path.
   - To define what to use for the ID of the new objects that will get created.

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).



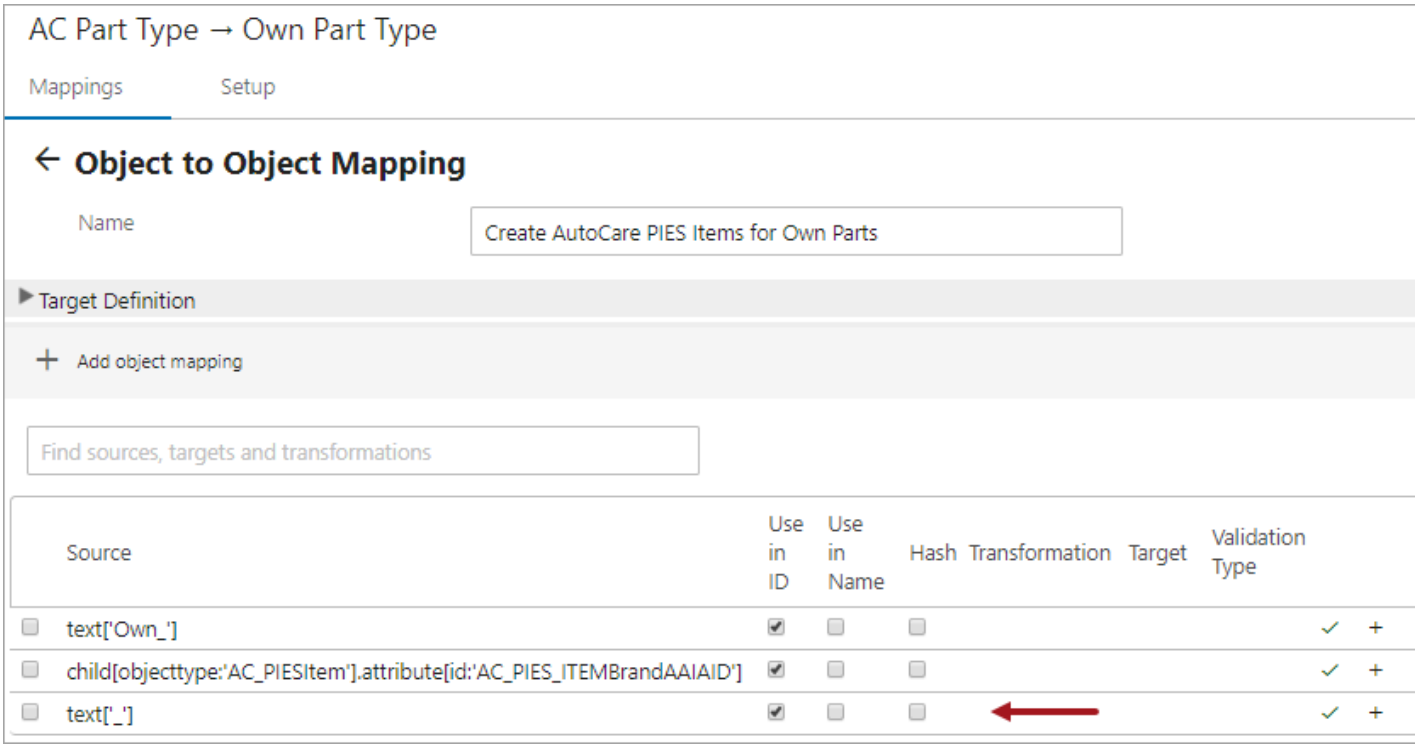

- To get the BrandAAIAID value in the ID of the new objects that will get created.

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).



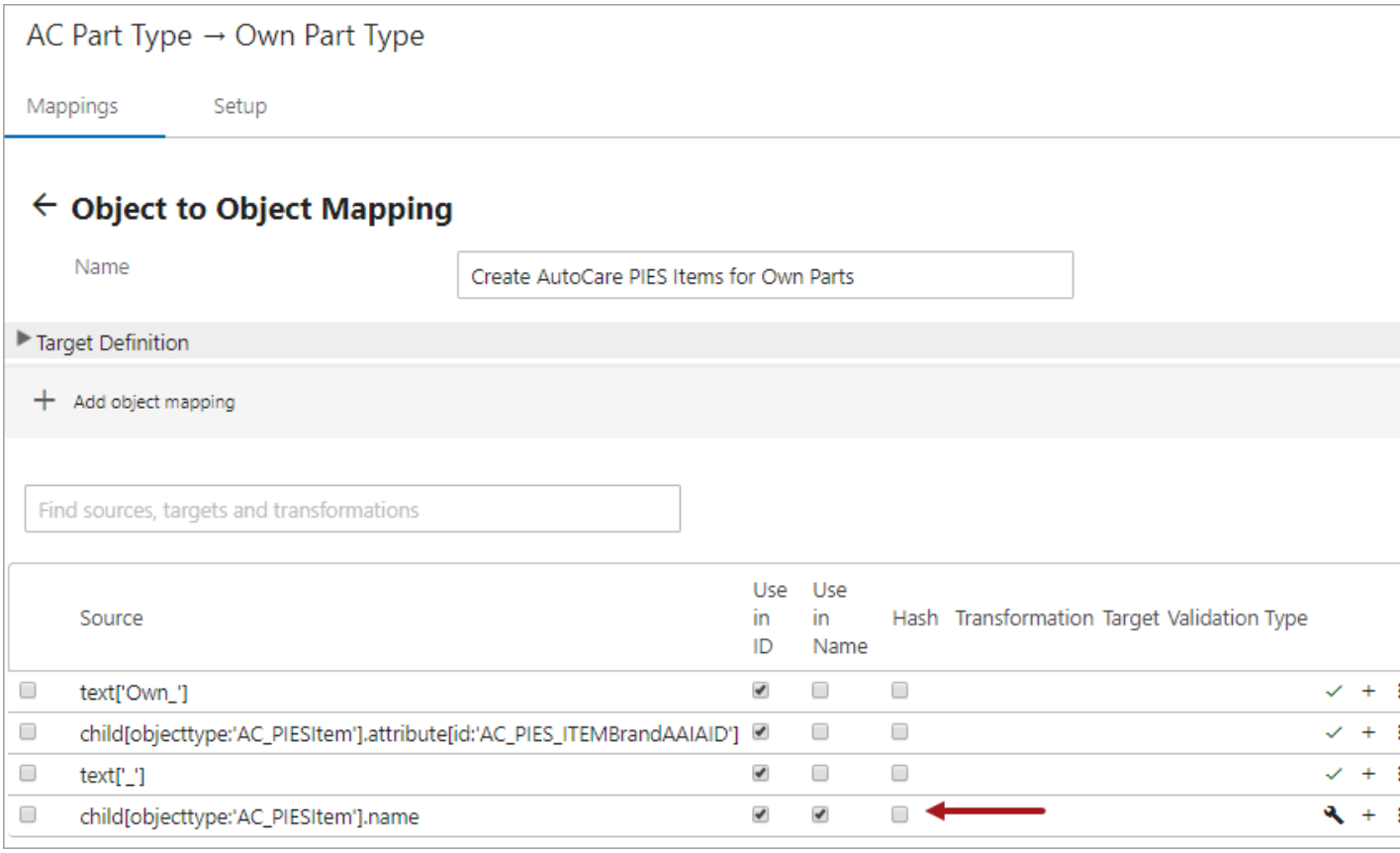

- To get the '_' value in the ID of the new objects that will get created.

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).



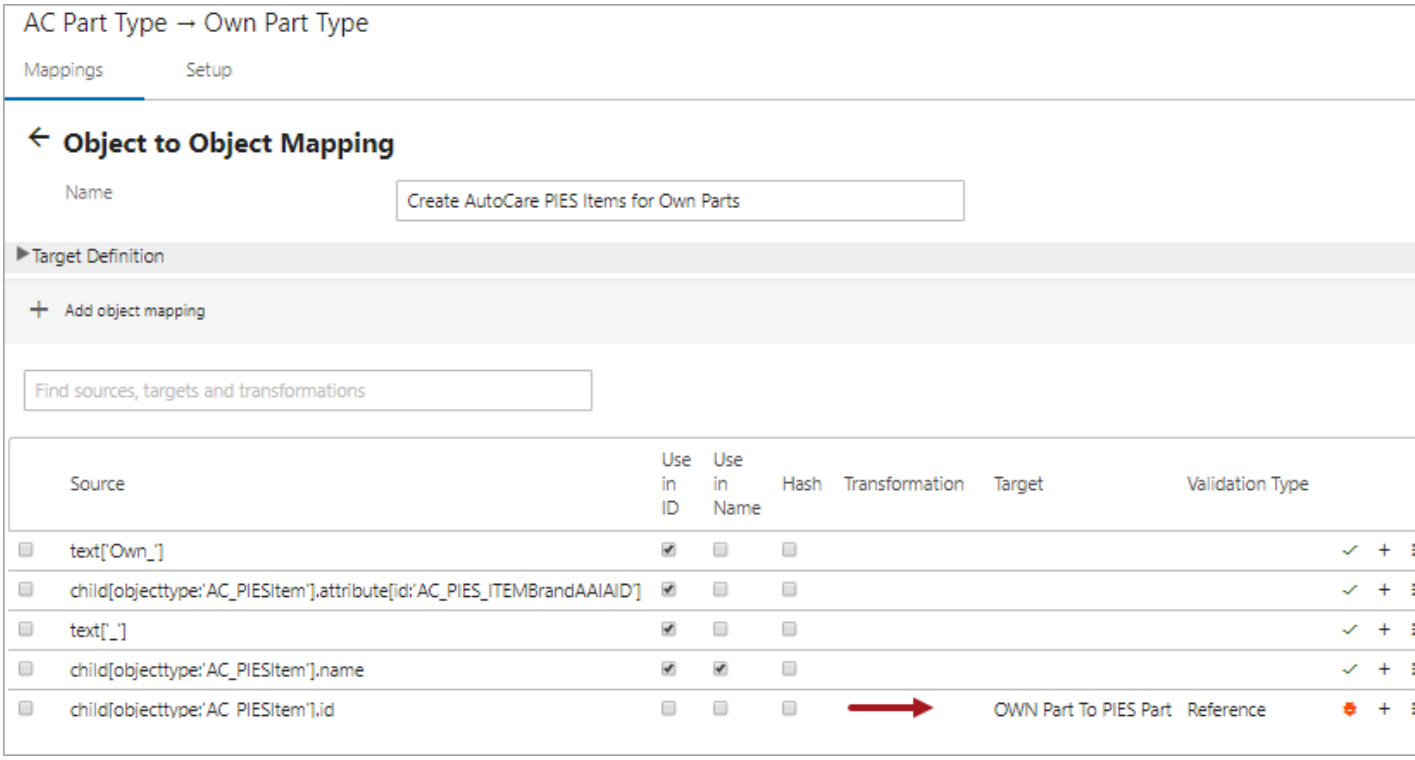

- To define what to use for the Name of the new objects that are getting created, as well to get the PIES Item Name value in the ID of the new objects that will get created.

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).

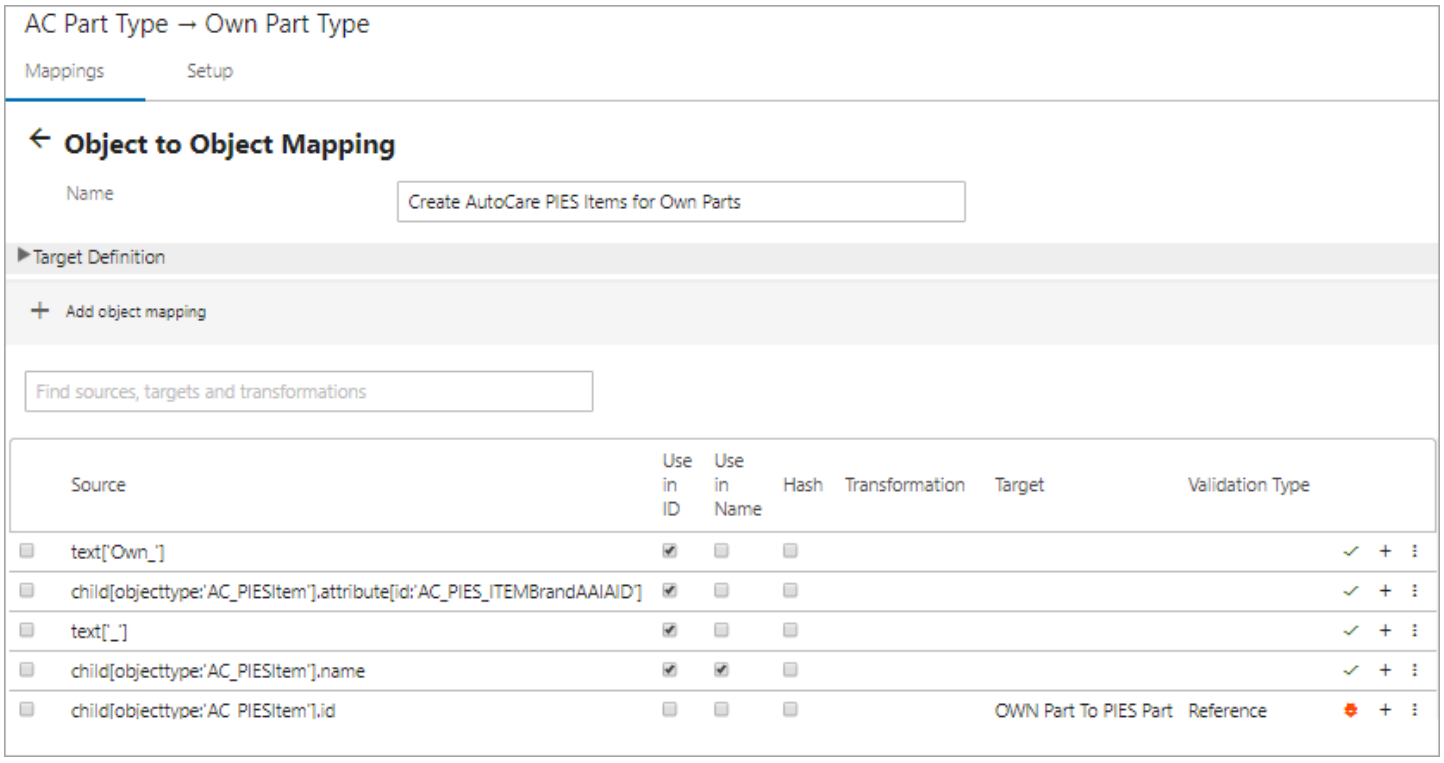



- The use case requires the establishment of the reference from the newly created Own part number to the PIES Item.

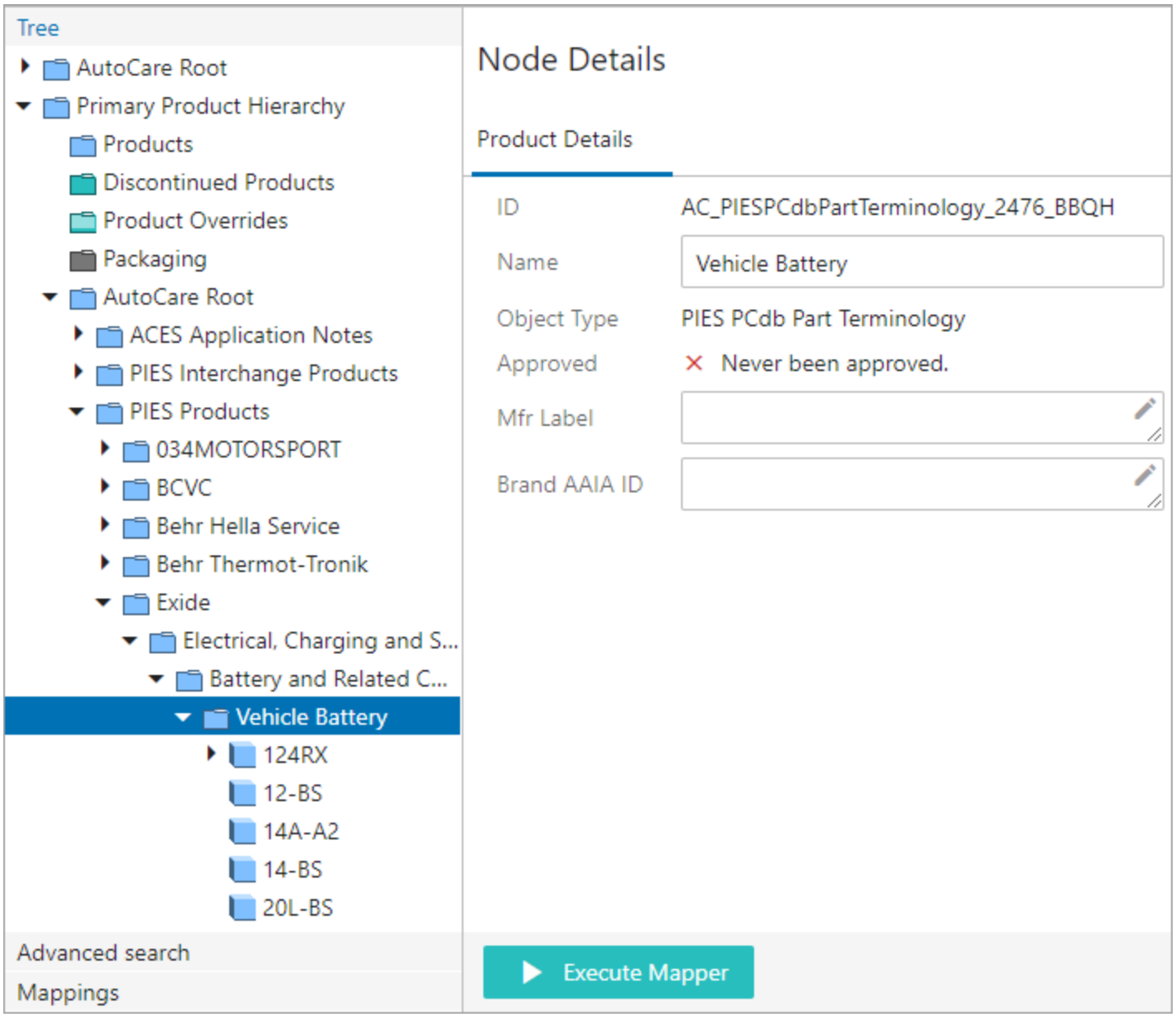
Using the **child[objecttype:'[id]']** and **id** elements in the STEP path to follow the child object of the Source object, the PIES Item is retrieved as the Source Object. Also the reference type 'OwnPartToPIESPart' is defined in the Target field define the Target reference type (as shown below).




8. Click **OK** to save and close the Mapping Guide window. The newly added object mapping row will be listed.



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, refer to topic **Executing Mapper Configuration Setup Entity** within this guide.

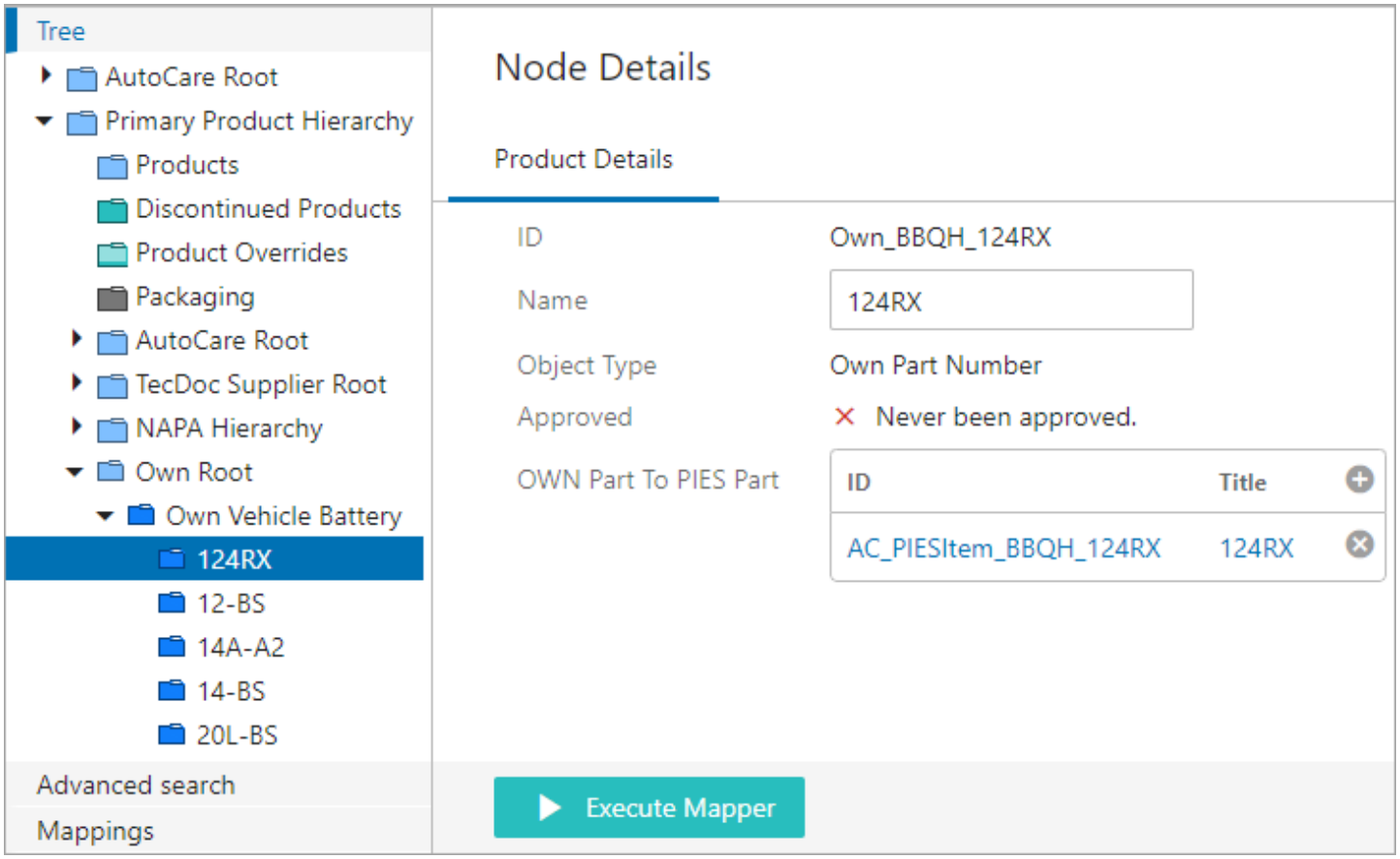



After the user runs the mapping, the following changes are implemented in the system:

- 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.



### 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

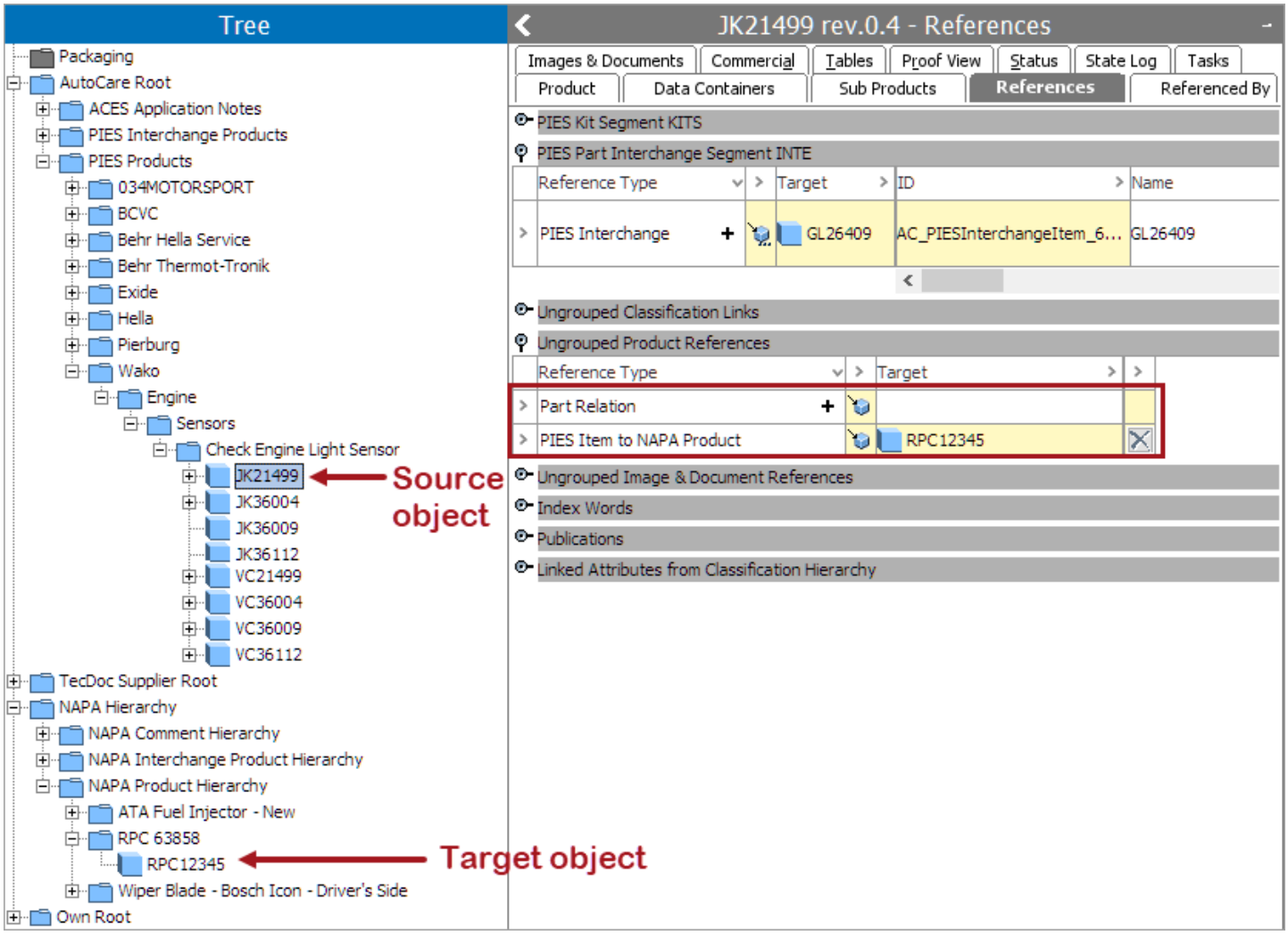
The requirement for this use case entails that the user has maintained part numbers (PIES Items) and Interchange objects (PIES Interchange objects) in the AutoCare Product hierarchy, but also wants to create the PIES Interchange objects as Interchange objects in the NAPA product hierarchy and establish 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.

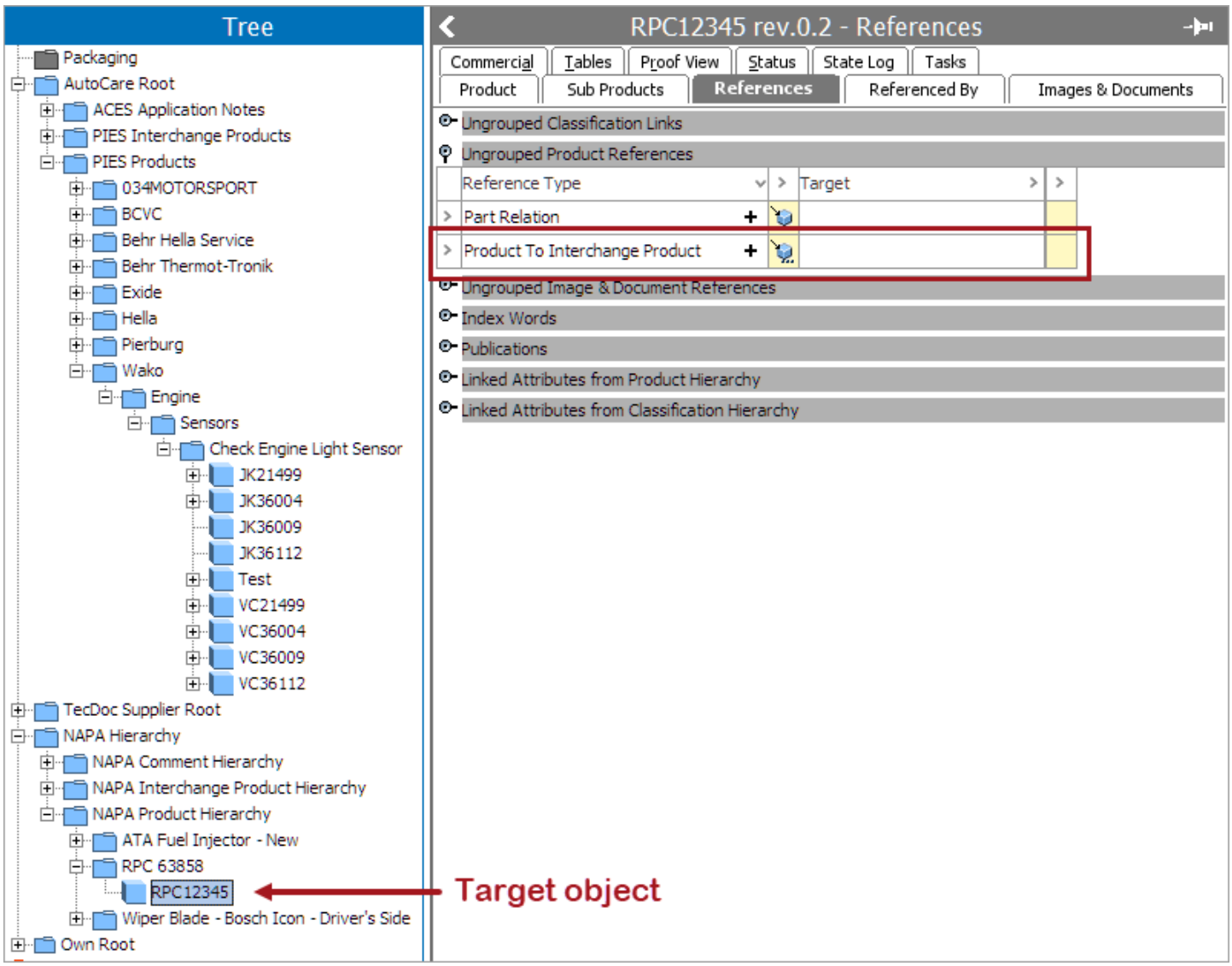
Following are the detailed steps describing how to achieve the same via the mapping solution.

## Prerequisites

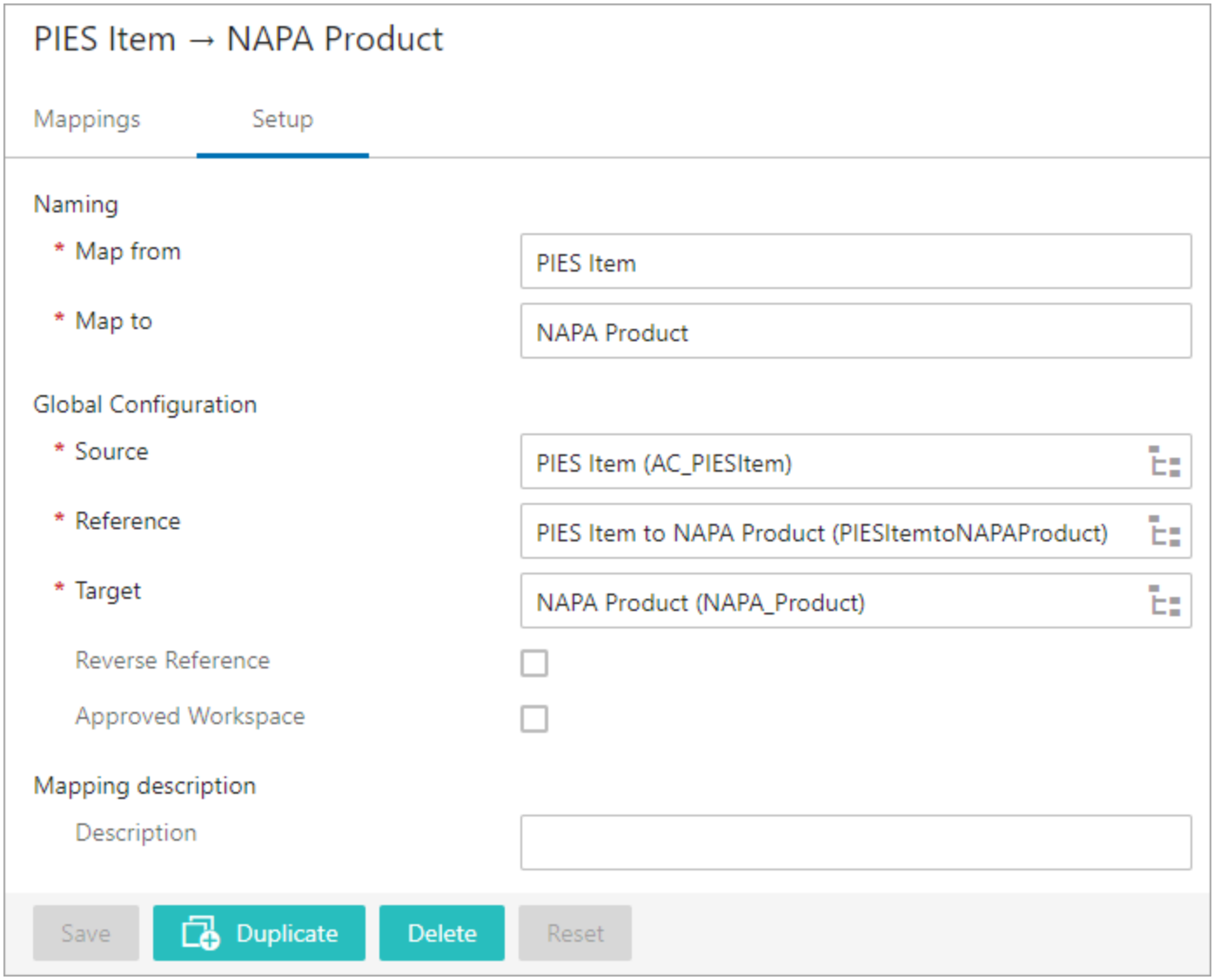
Create a new Product Reference Type to link the Source object PIES Item to the Target object NAPA Product and establish the reference between the PIES Item to the NAPA Product. In this example, the Product Reference Type 'PIES Item to NAPA Product' is defined to create a link between PIES Item and NAPA Product. Below is the image displaying hierarchy of Source object (JK21499), Target object (RPC12345), and the reference linking them.




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, refer to the **Reference Types** topic within the **System Setup** documentation in **STEP Online Help**.

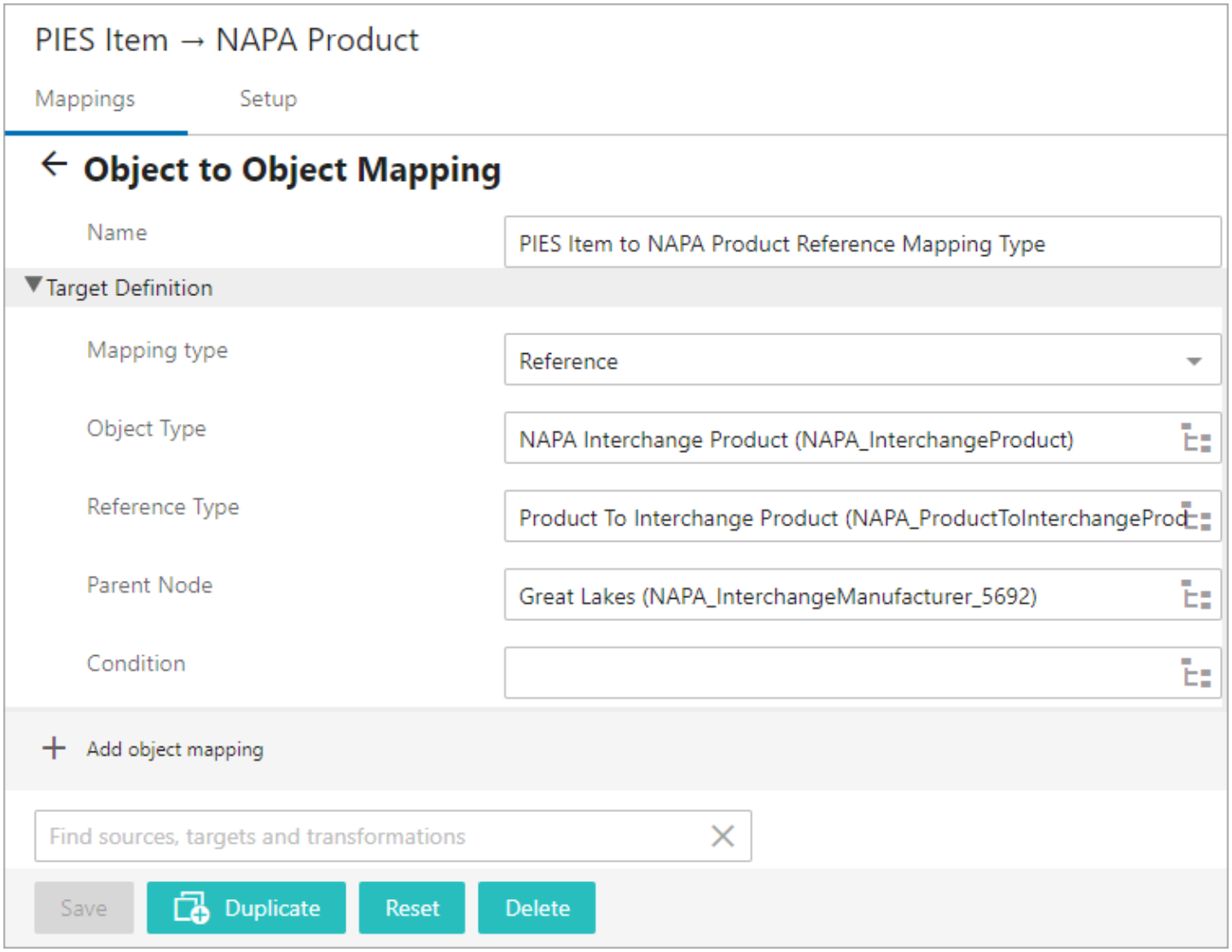



In the Web UI, create and configure the Mapper Configuration and define the Source and Target objects and the Reference type to follow to determine the relationship between the Source and Target object. For more information on configuring and setup procedure for the Mapper Configuration, refer to topic **Configuring Mapper Configuration Setup Entity** within this guide.

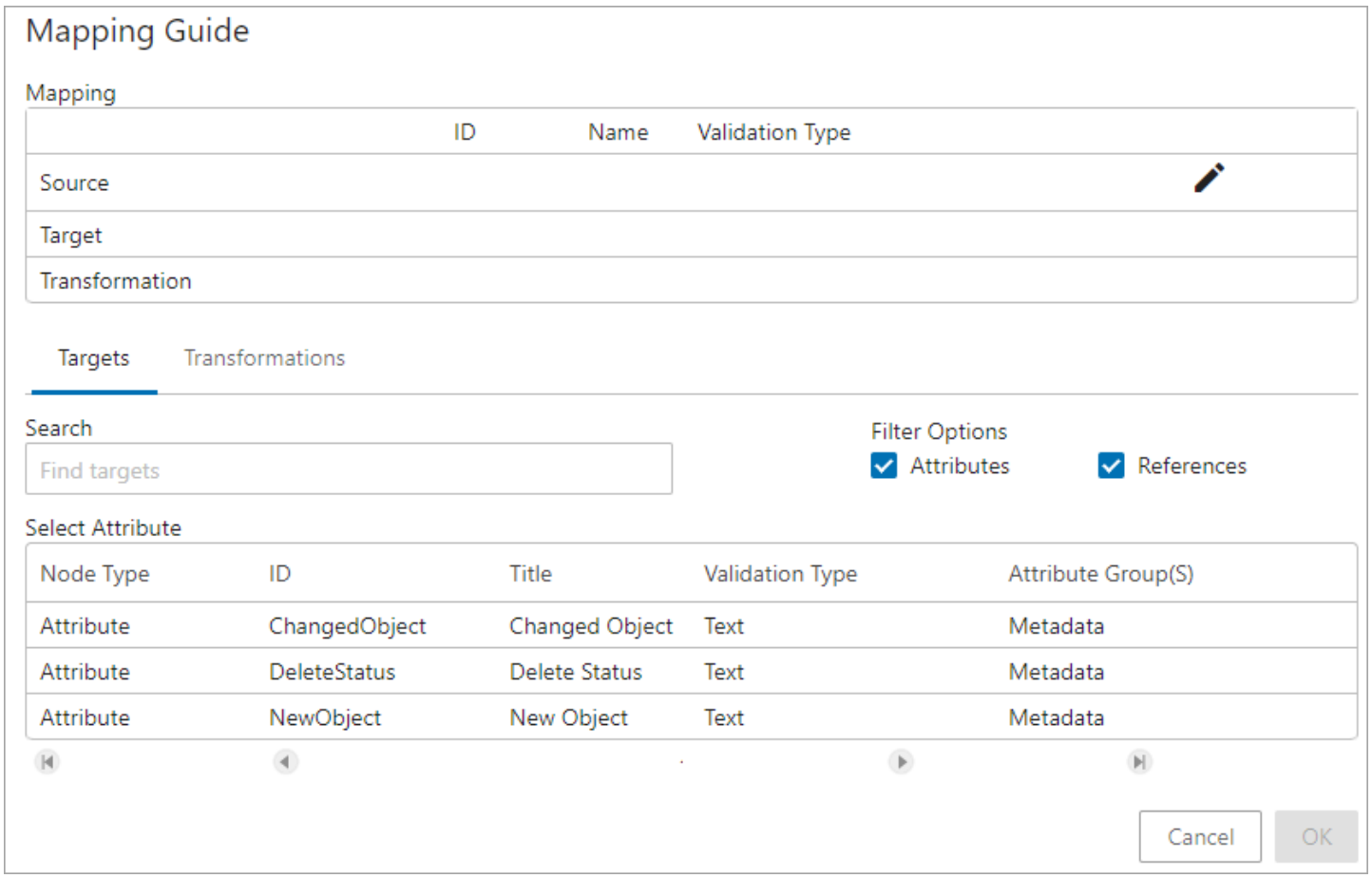



Configure an Initiate Business Action button within the Product Details screen of the Source object which shall execute the Mapper Configuration. For more information on executing the Mapper Configuration, refer to topic **Executing Mapper Configuration Setup Entity** within this guide.

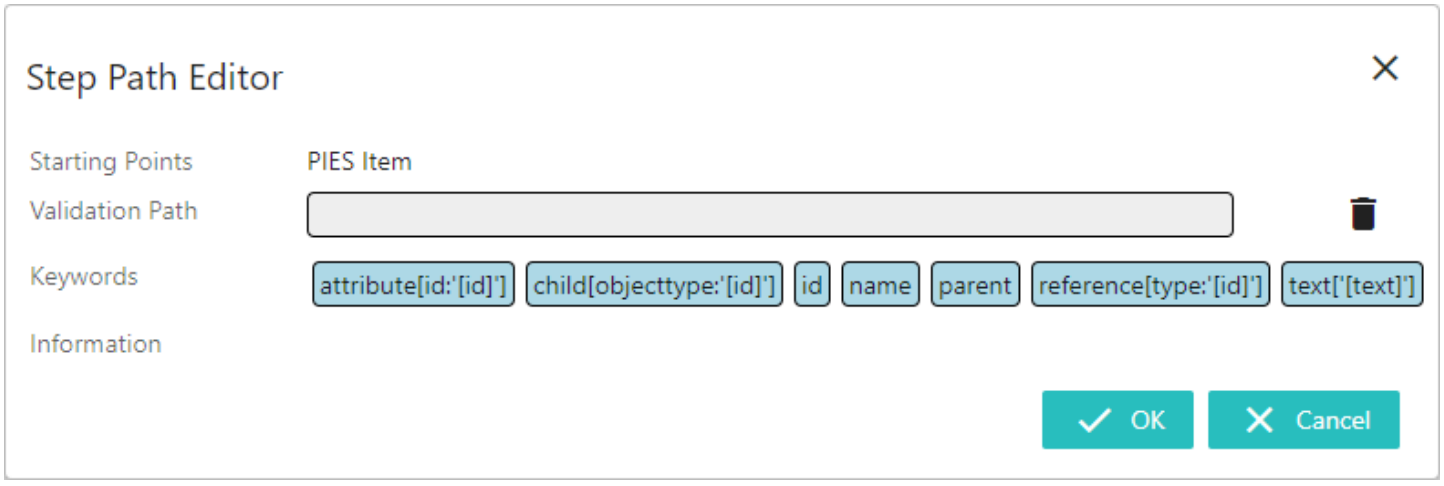
1. Add a new Object to Object Mapping plugin (or edit the one that exists) in the required Mapper Configuration.
2. Type in a suitable name next to Name field. In this example, the Name field is entered with the text 'PIES Item to NAPA Product Reference Mapping Type.'




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.



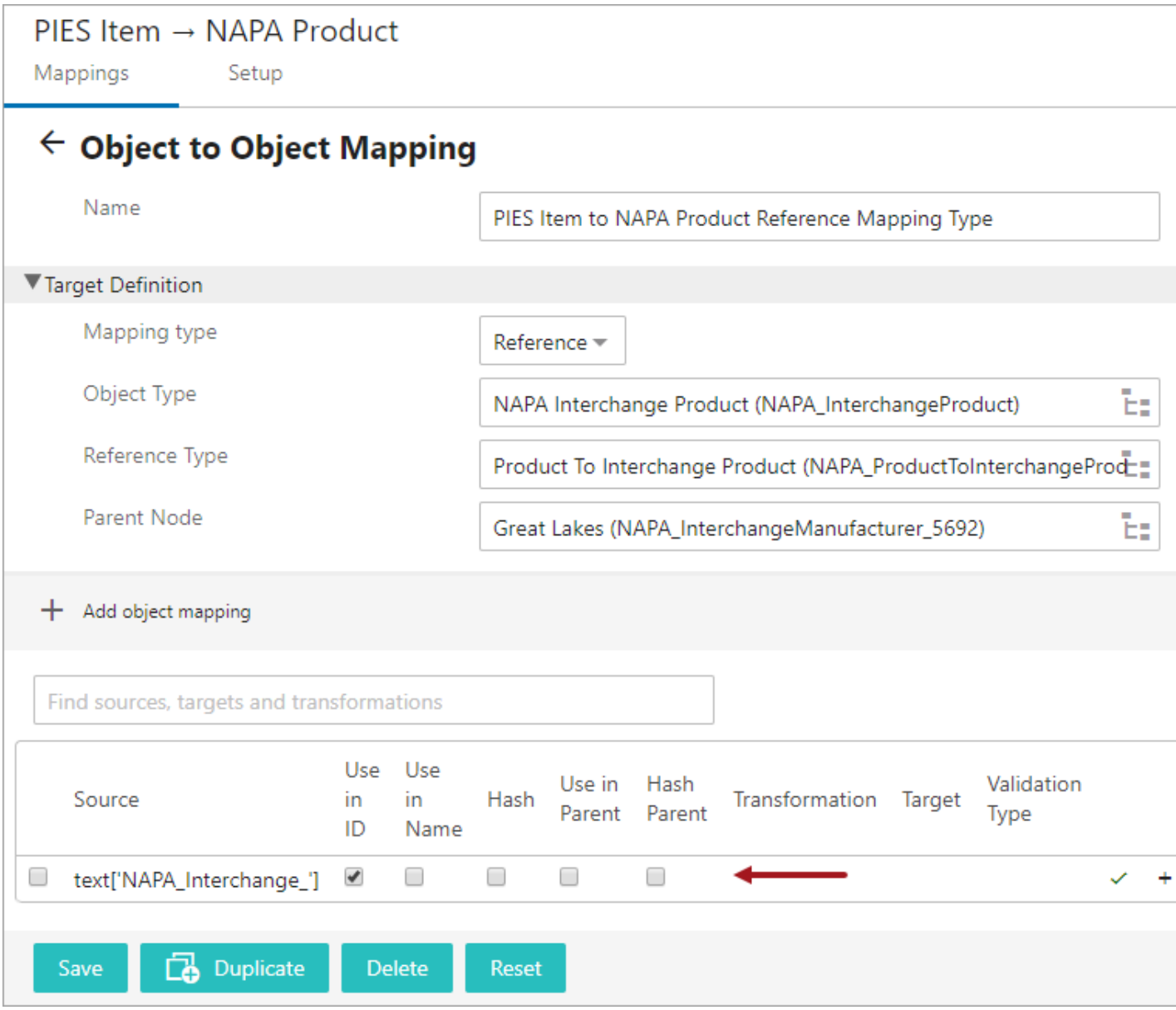

8. Click on the Edit icon ( ) in the Source field, and the Step Path Editor window will display (as shown below). For more information on defining the Source STEP path, refer to topic **Mapping Validation Path Functionality** within this guide.




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.

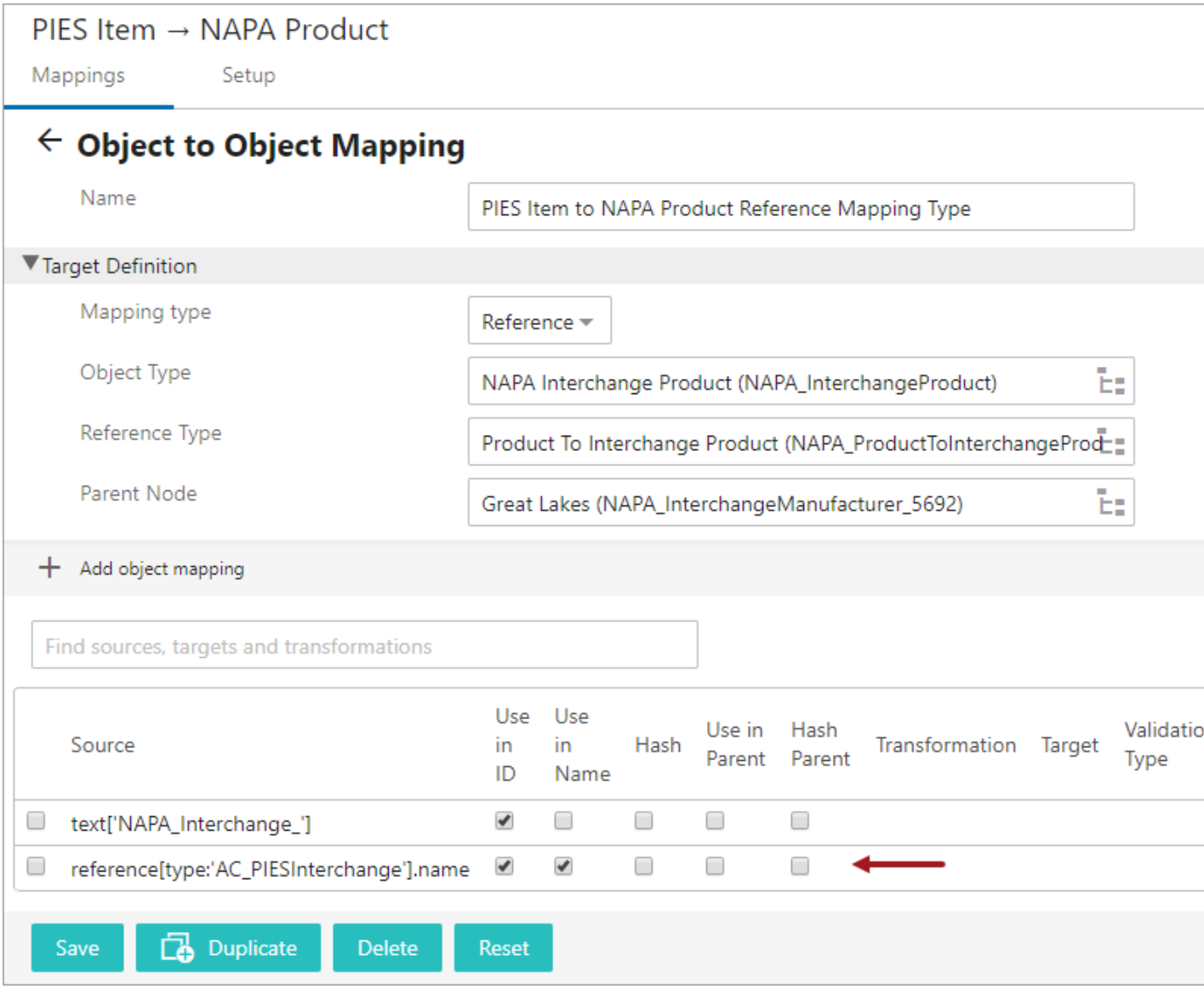
- To define what to use for the ID of the new objects that will get created.

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).

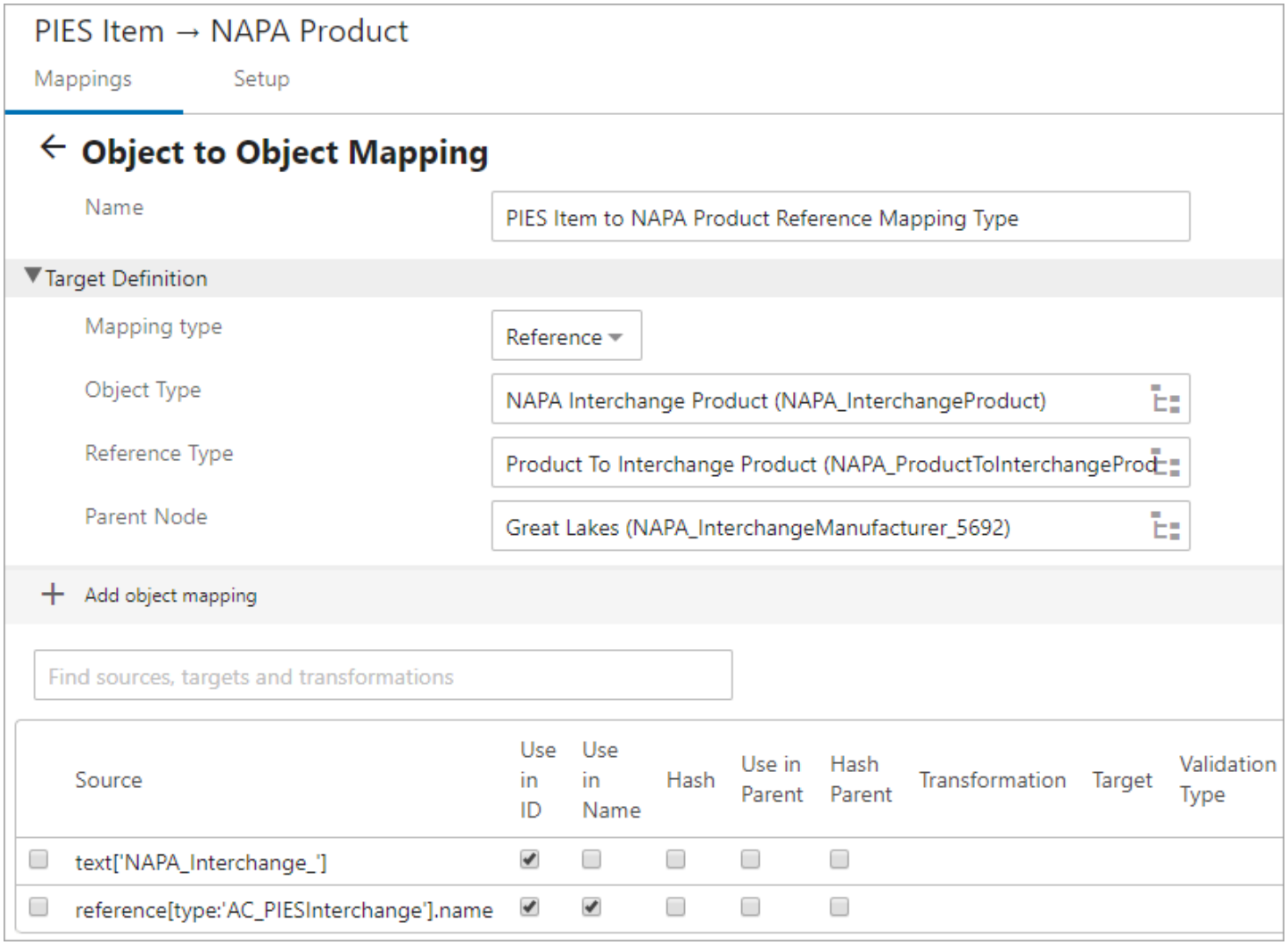



- To define what to use for the Name of the new objects that are getting created, as well to get the PIES Interchange object name value in the ID of the new objects that will get created.

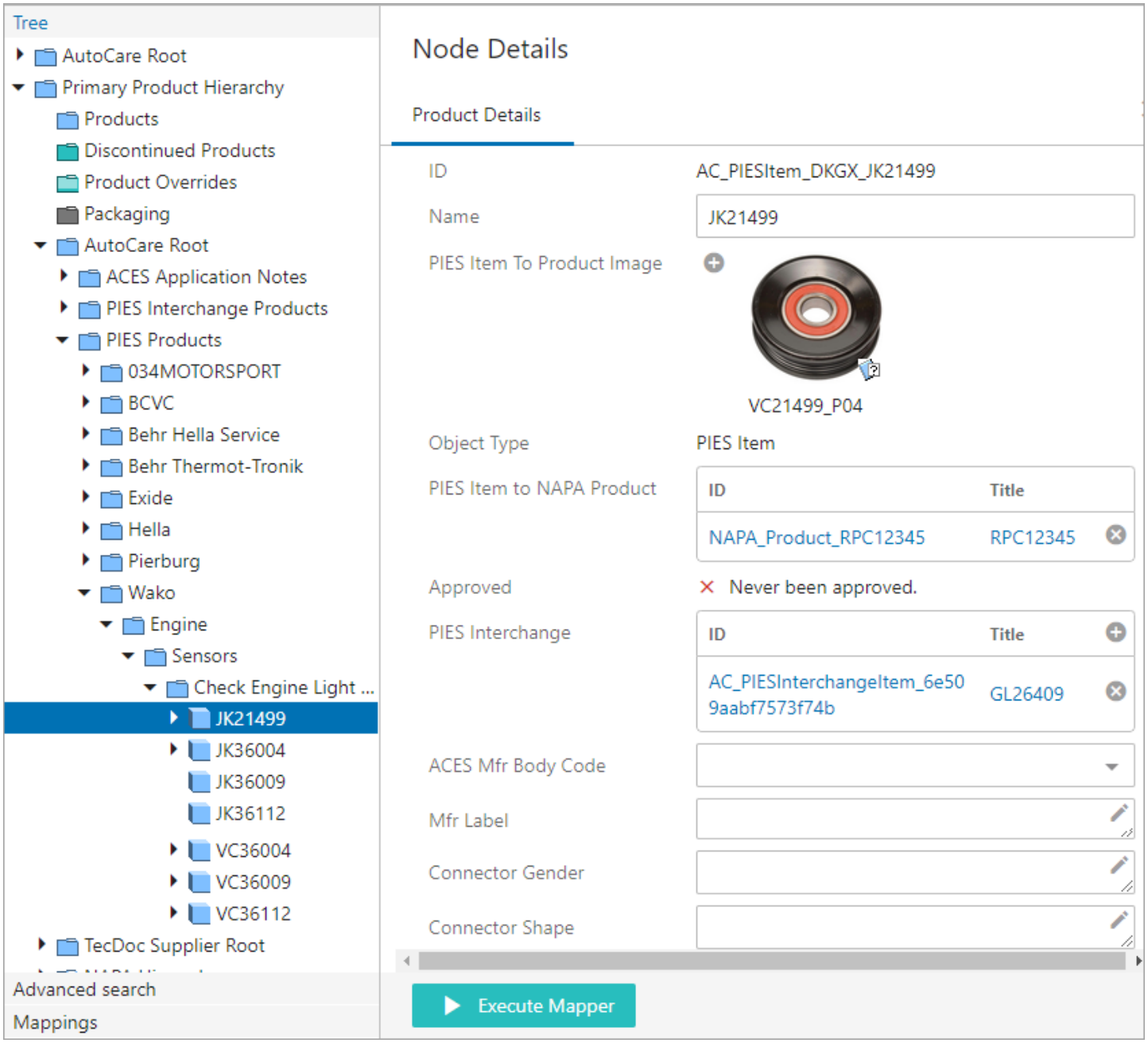
In this example, the PIES Interchange object name will be used in the ID and the Name of the new objects. So, using the **reference[type:'[id]']** and **name** elements in the STEP path will follow defined referenced objects of the Source object and retrieve the PIES Interchange object name. The 'Use in ID' and 'Use in Name' checkboxes are selected (as shown below).




10. Click **OK** to save and close the Mapping Guide window. The newly added object mapping row will be listed.

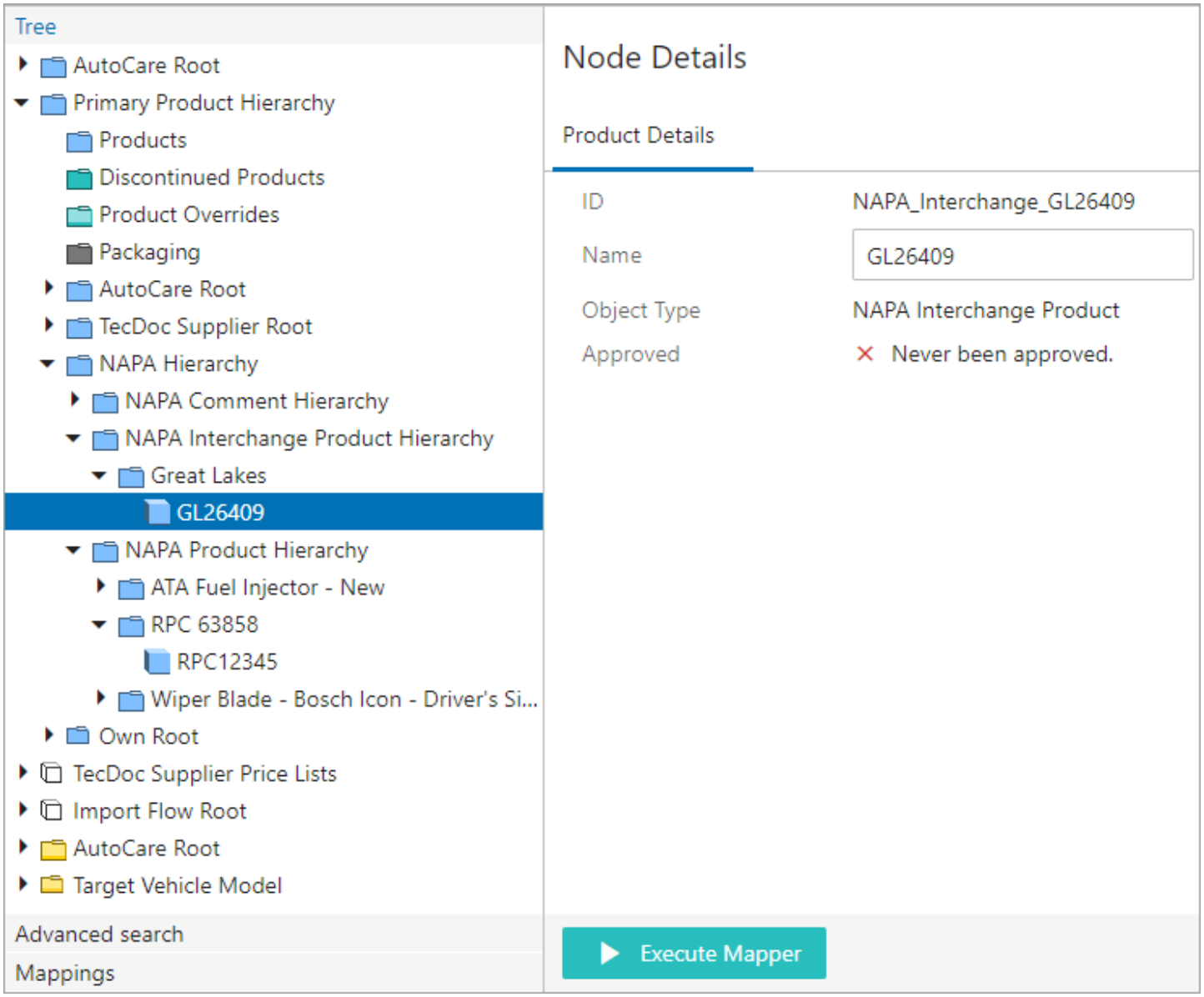



11. In the Web UI, standing on the Source object, execute the Mapper Configuration that holds the mapping plugin. For more information on executing the Mapper Configuration, refer to topic **Executing Mapper Configuration Setup Entity** within this guide.

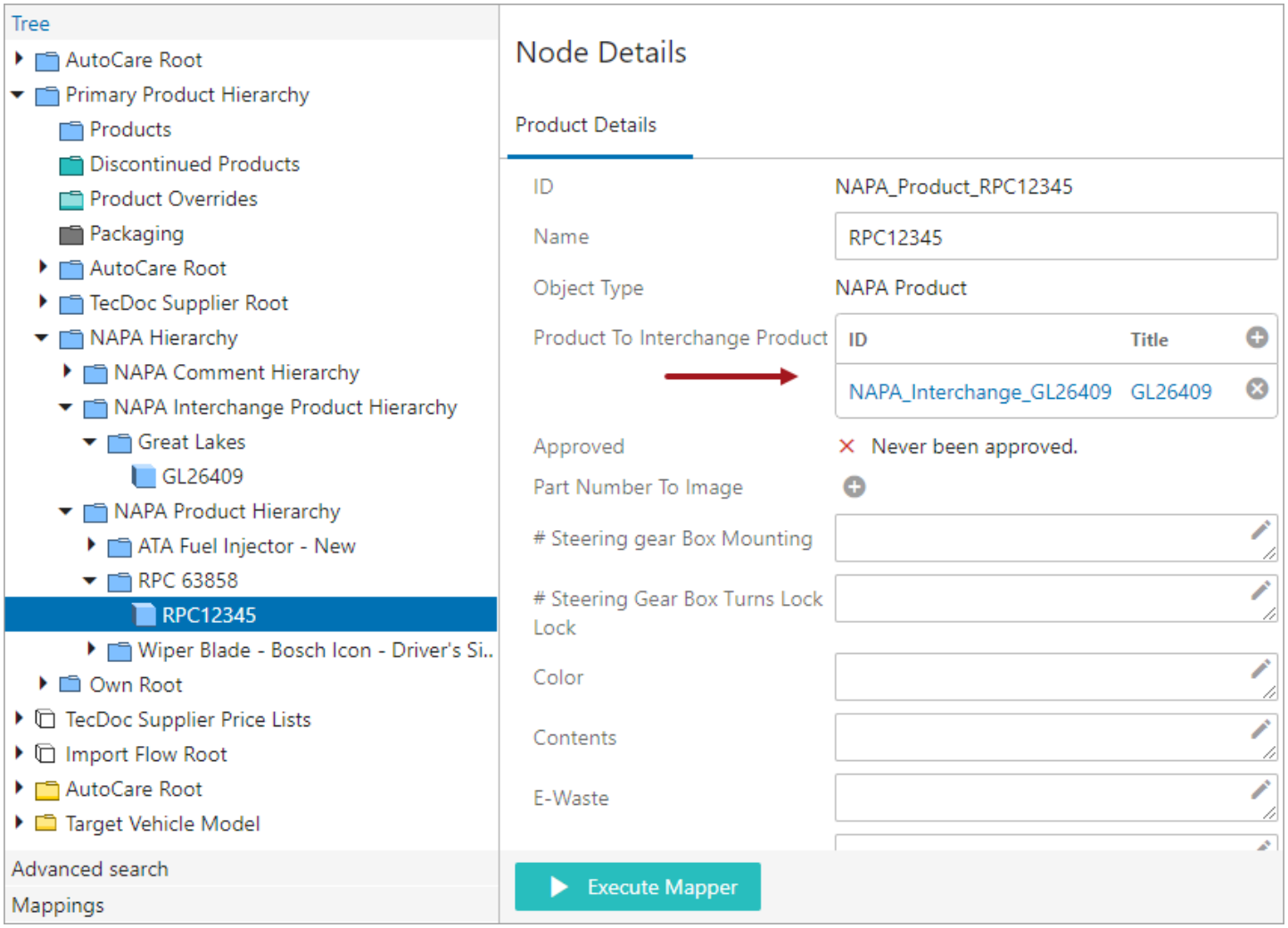



After the user runs the mapping, the following changes are implemented in the system.

- 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.




- 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.

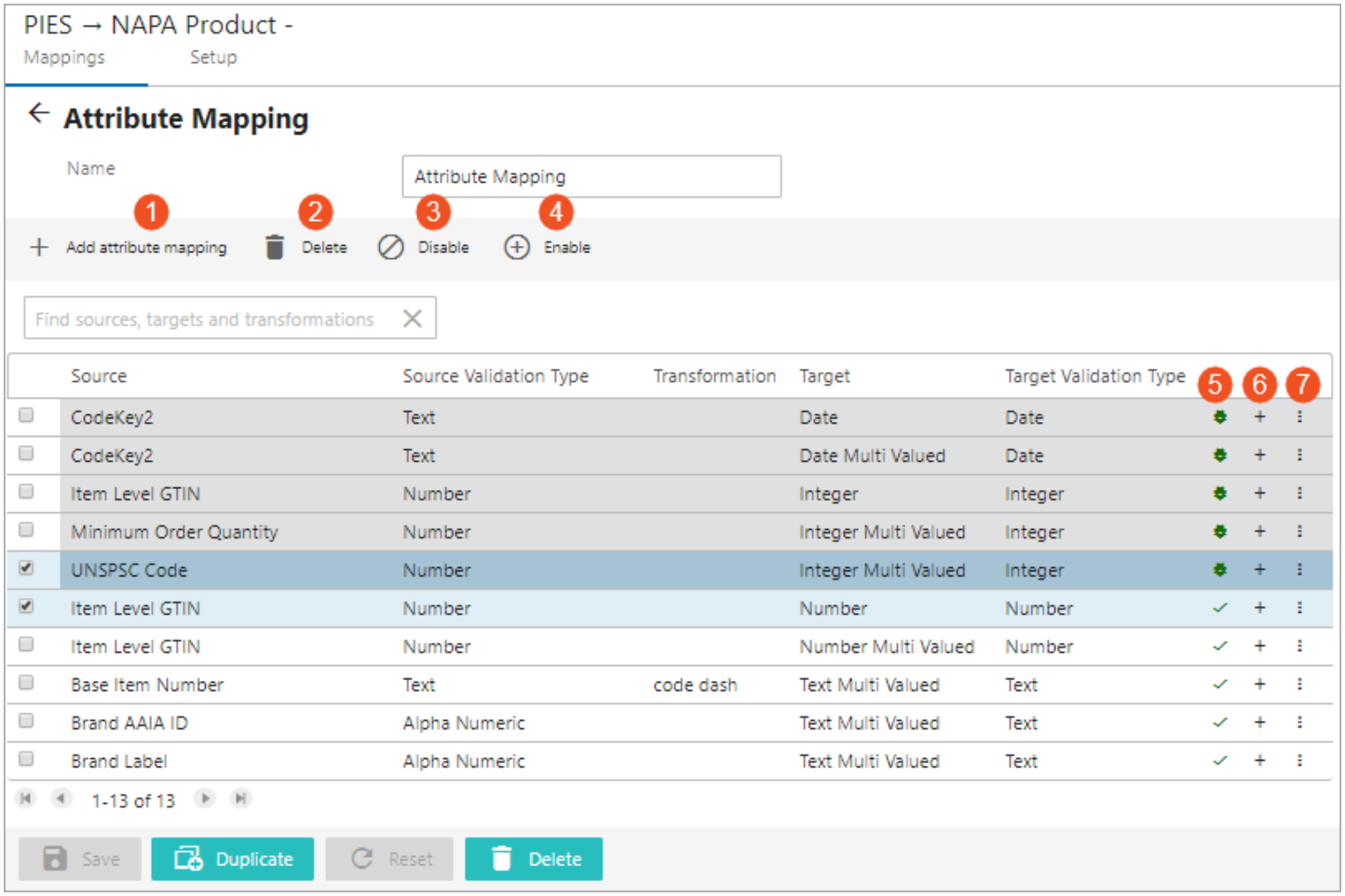



## Modifying Mapper Rows on the Onboarding Mapping Details Screen

As described within each of the mapping plugin topics, the 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 user has the flexibility to disable, delete, or rearrange the listing order of the mapper rows. The health of the mapper row is displayed next to each mapper row. Users can also add some additional information describing each of the mapper rows. In the following image, the various functionalities are numbered.

In the image below, the Attribute Mapping plugin is configured to perform multiple mapping functions. Each of the mapping functions is displayed as rows, and are called mapping rows.




The available modifications defined below include:

1. Add a mapper row to the mapping plugin
2. Delete a mapper row in the mapping plugin
3. Disable a mapper row in the mapping plugin
4. Enable the disabled mapper row in the mapping plugin
5. Check the health of the mapper row
6. Add a description to the mapper row
7. Change the order of the mapper row in the mapping plugin

## Add a mapper row to the mapping plugin

Details on how to add the mapper rows are provided within each of the mapping plugin topics. For more information about how to add these mapper rows, refer to the Mapping Plugins topic.

## Delete a mapper row in the mapping plugin

Deleting a mapper row allows you to no longer display (and execute) the mapper row for the selected plugin in the Onboarding Mappings Details screen when it is no longer required.

1. For the selected Mapper Configuration setup entity, on the Mappings tab, open the mapping plugin for the mapper row that should be deleted.
2. Select the checkbox that is available on the left of the mapper row that is intended to delete. This makes the Delete button visible in the toolbar.
3. Click the Delete button. This removes the mapper row from the Onboarding Mappings Details screen in Web UI and also removes the functionalities defined in that particular mapper row from being executed when the user executes the Mapper Configuration setup entity.
4. Save the configuration after the row has been deleted.

## Disable a mapper row in the mapping plugin

The execution of the mapper rows within the plugin can be controlled. The rows displayed within the plugin can be disabled as needed, and the disabled mapper row lies in a deactivated state on the mapping plugin. Disabling a mapper row allows you to no longer execute (unless enabled) the mapper row for the selected plugin in the Onboarding Mappings Details screen when it is temporarily not required. By default, all the newly created mapper rows remain activated / enabled.

1. For the selected Mapper Configuration setup entity, on the Mappings tab, open the mapping plugin for the mapper row that should be disabled.
2. Select the checkbox that is available on the left of the mapper row that is intended to be disabled. This makes the Disable button visible in the toolbar.
3. Click the Disable button. This disables the particular mapper row from being executed when the user executes the associated Mapper Configuration setup entity.
4. Save the configuration after the row has been disabled.

The mapper row turns gray when disabled.

## Enable the disabled mapper row in the mapping plugin

The disabled mapper row can be enabled (reactivated) as per the user requirement. Enabling a mapper row allows you to execute the mapper row for the selected plugin in the Onboarding Mappings Details screen. By default, all the newly created mapper rows remain activated / enabled.

1. For the selected Mapper Configuration setup entity, on the Mappings tab, open the mapping plugin for the disabled mapper row that should be enabled.
2. Select the checkbox that is available on the left of the disabled mapper row that is intended to be enabled. This makes the Enable button visible in the toolbar.
3. Click the Enable button. This enables the particular mapper row to be executed when the user executes the associated Mapper Configuration setup entity.
4. Save the configuration after the row has been enabled.

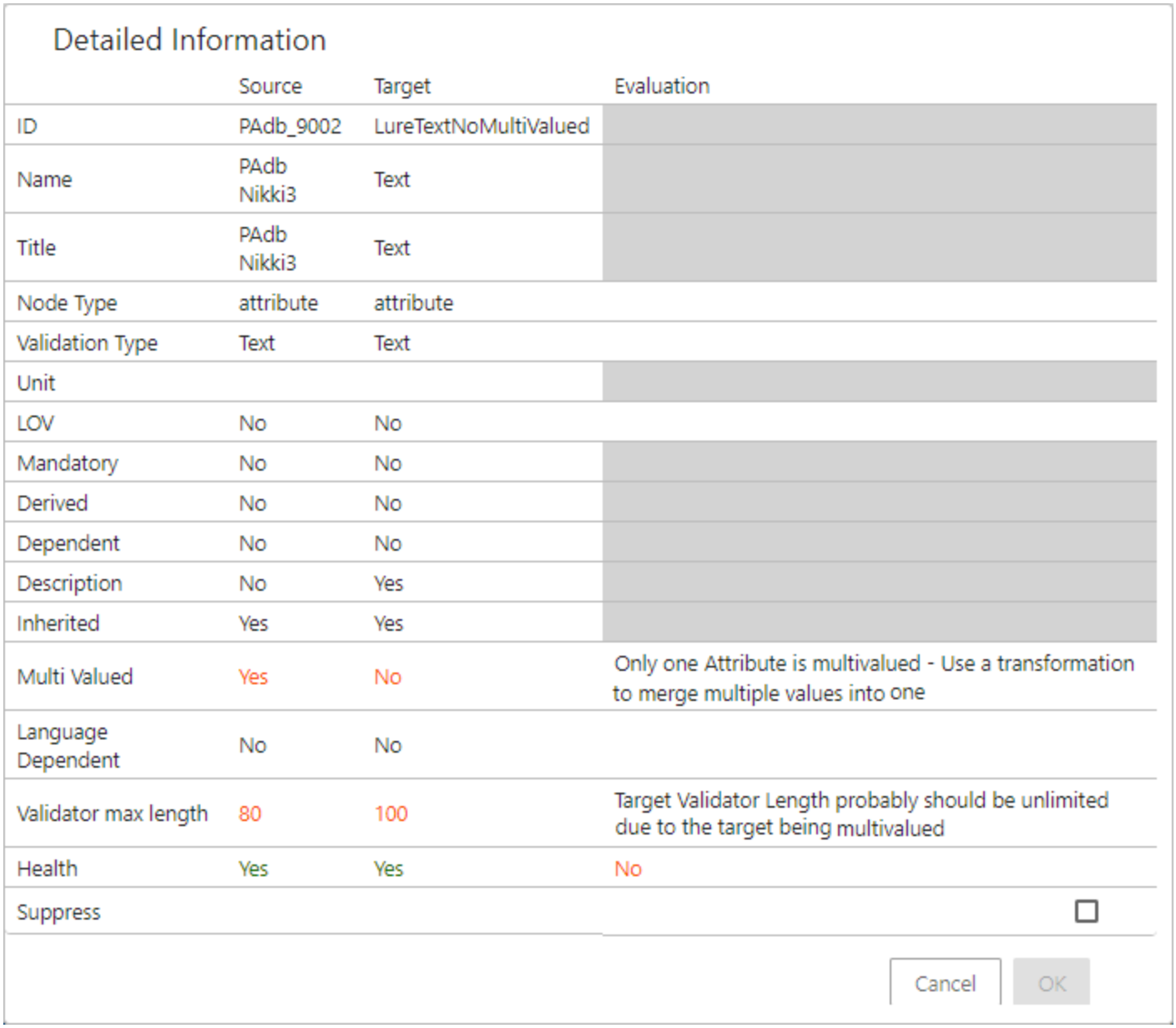
## Check the health of the mapper row

The health of the mapper row is displayed by a unique type of icon that is visible next to each mapper row. The following are three different types of icons that depict the health of the mapper row. Click any of the below-mentioned icons to view the health of the mapper row.

- Red bug icon

  This icon denotes the mapper row as unhealthy and includes some flaws. Clicking on the icon displays a 'Detailed Information' dialog.

  Users can suppress the 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 that is displayed on the Mapping Guide window and **does not** resolve the mismatch irregularities.



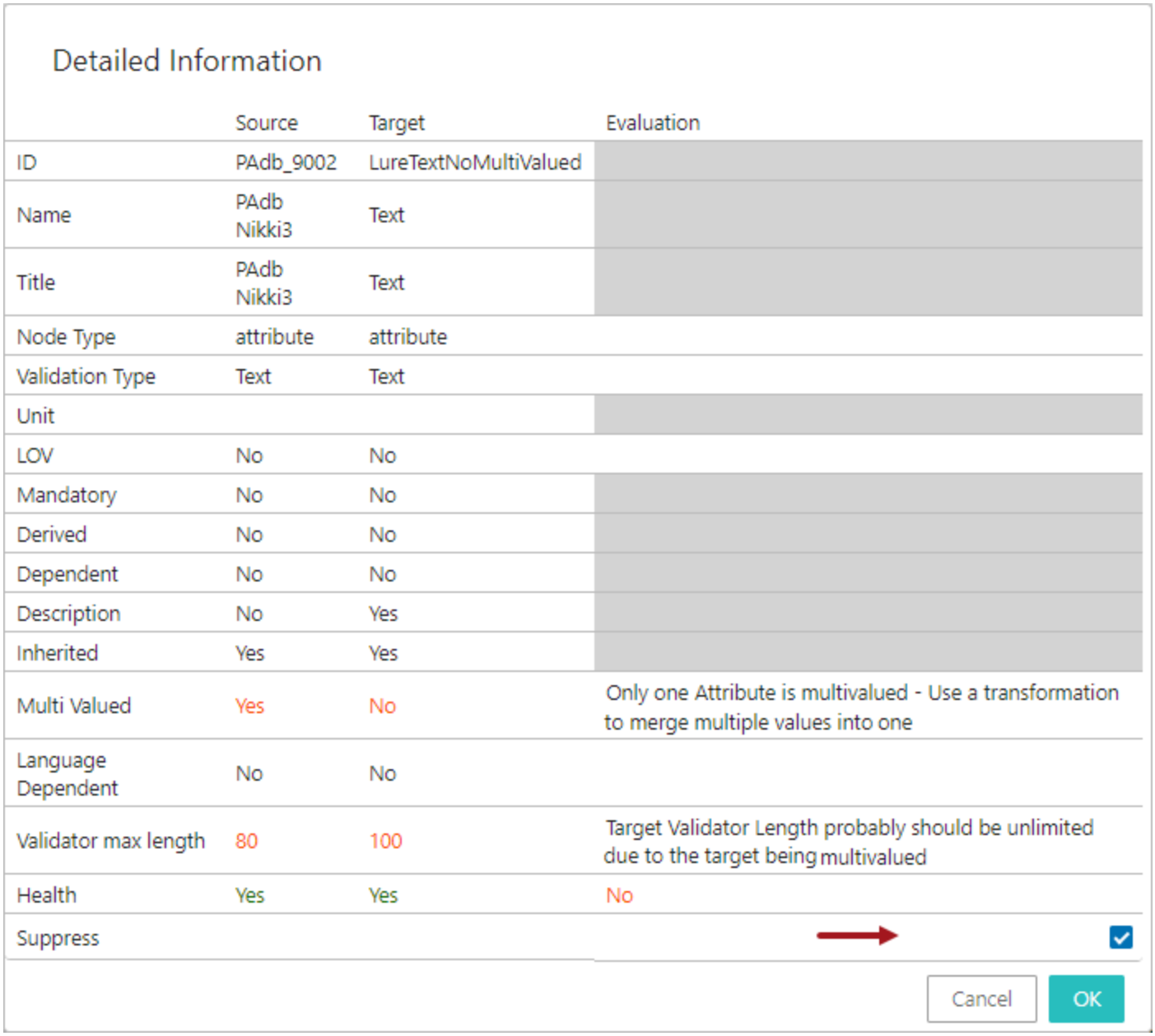
Detailed Information

| | Source | Target | Evaluation |
|---|---|---|---|
| ID | PAdb_9002 | LureTextNoMultiValued | |
| Name | PAdb Nikki3 | Text | |
| Title | PAdb Nikki3 | Text | |
| Node Type | attribute | attribute | |
| Validation Type | Text | Text | |
| Unit | | | |
| LOV | No | No | |
| Mandatory | No | No | |
| Derived | No | No | |
| Dependent | No | No | |
| Description | No | Yes | |
| Inherited | Yes | Yes | |
| Multi Valued | Yes | No | Only one Attribute is multivalued - Use a transformation to merge multiple values into one |
| Language Dependent | No | No | |
| Validator max length | 80 | 100 | Target Validator Length probably should be unlimited due to the target being multivalued |
| Health | Yes | Yes | No |
| Suppress | | | ☐ |

Cancel OK

- Green bug icon

  This icon denotes the mapper row as unhealthy with some flaws but the user has decided to suppress the warning message. Clicking on the icon displays a 'Detailed Information' dialog.



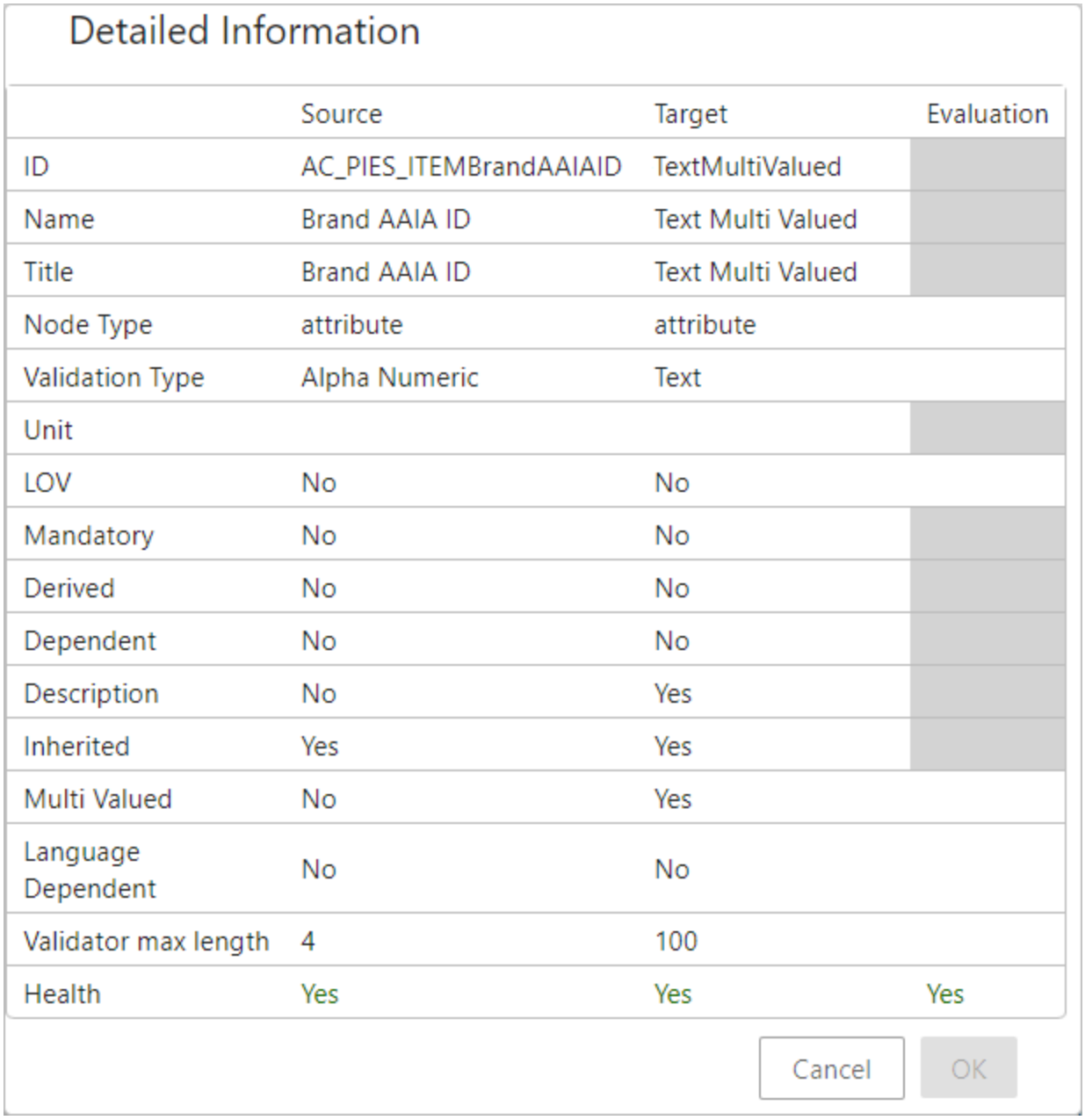
Detailed Information

| | Source | Target | Evaluation |
|---|---|---|---|
| ID | PAdb_9002 | LureTextNoMultiValued | |
| Name | PAdb Nikki3 | Text | |
| Title | PAdb Nikki3 | Text | |
| Node Type | attribute | attribute | |
| Validation Type | Text | Text | |
| Unit | | | |
| LOV | No | No | |
| Mandatory | No | No | |
| Derived | No | No | |
| Dependent | No | No | |
| Description | No | Yes | |
| Inherited | Yes | Yes | |
| Multi Valued | Yes | No | Only one Attribute is multivalued - Use a transformation to merge multiple values into one |
| Language Dependent | No | No | |
| Validator max length | 80 | 100 | Target Validator Length probably should be unlimited due to the target being multivalued |
| Health | Yes | Yes | No |
| Suppress | | | |

Cancel OK

- Green tick

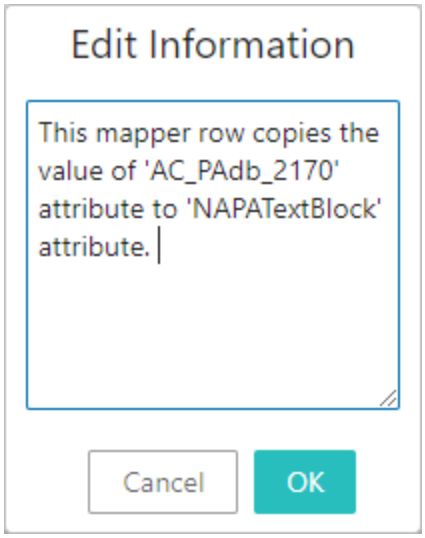
  This icon denotes the mapper row as healthy without any flaws. Clicking on the icon displays a 'Detailed Information' dialog.




## Add a description to the mapper row

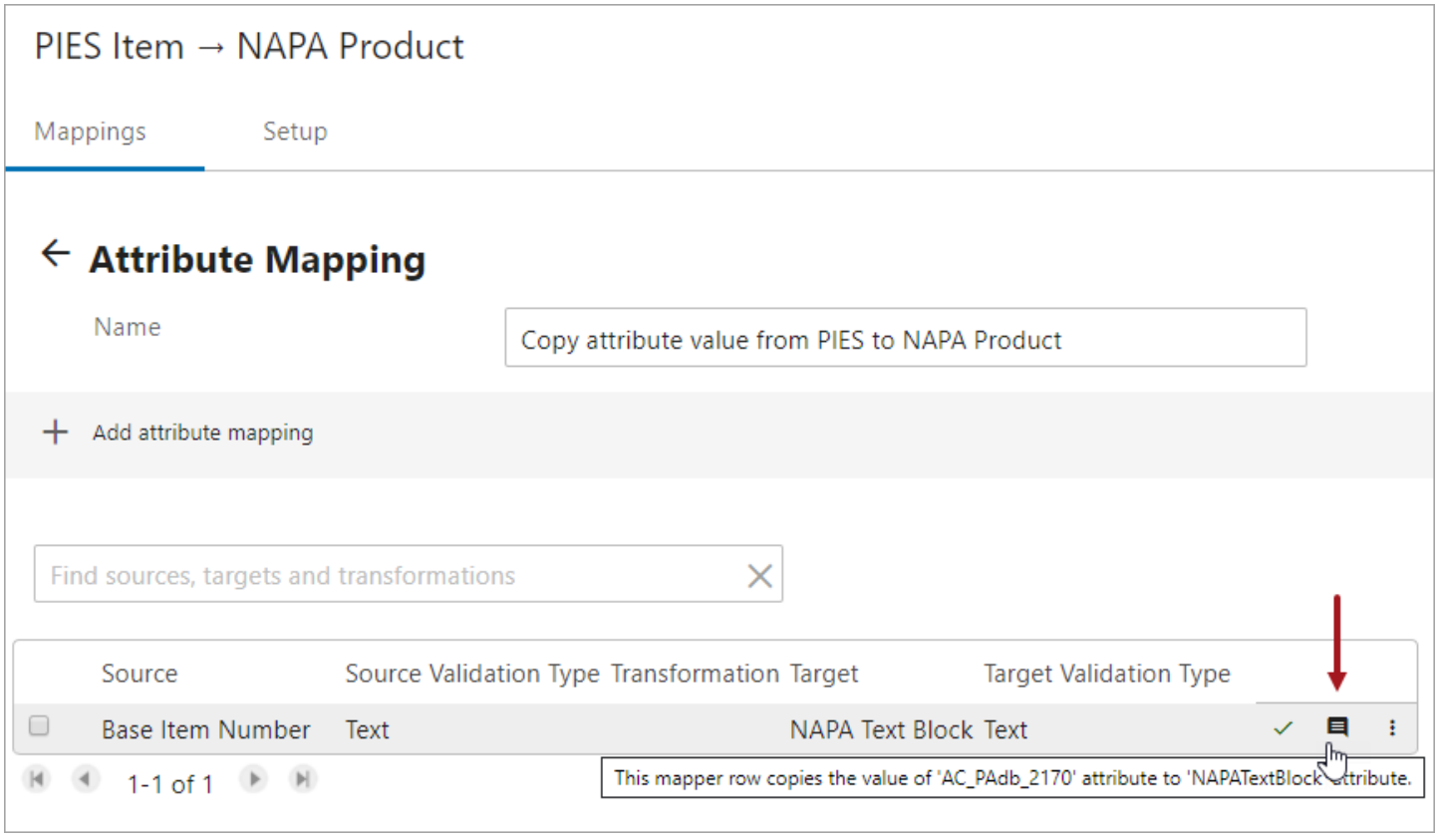
Users can add any information as text that helps explain the mapper row functionality. Below is the detailed procedure to add a description to the mapper row.

1. For the selected Mapper Configuration setup entity, on the Mappings tab, open the mapping plugin for the mapper row that should be described.
2. Click the Add Description icon ( + ) that is available to the right of the mapper row. This will display the Edit Information dialog.




3. Type in the required text and click OK to save and close the dialog.

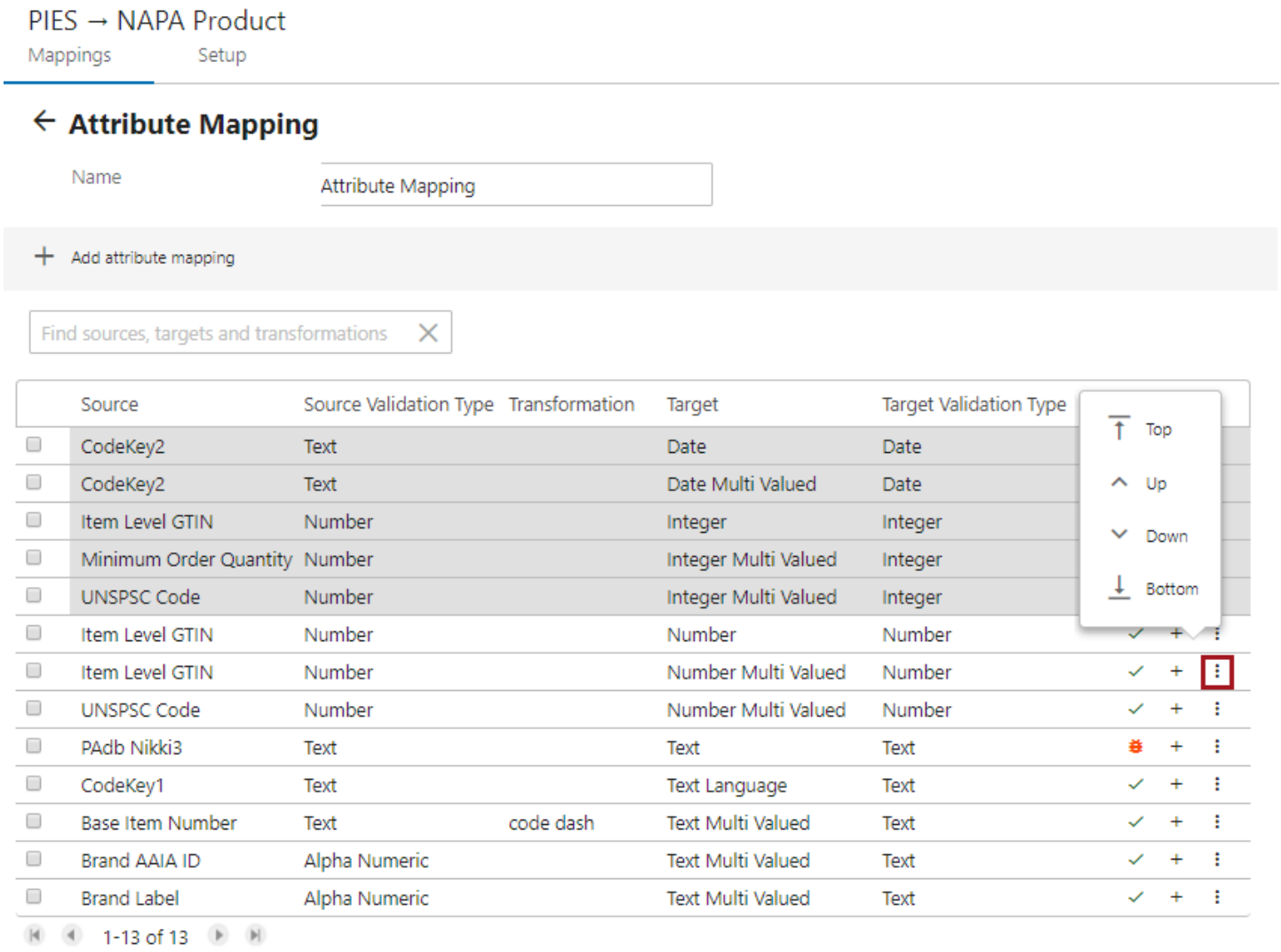
After the description has been entered, the icon style will be changed from ( + ) to ( ) symbolizing the presence of some information within the icon. The description will be visible when the user clicks or hovers over the ( ) icon.



## Change the order of the mapper row 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 user can rearrange the order of the mapper row. Below is the procedure to rearrange the order of the mapper row.

1. For the selected Mapper Configuration setup entity, on the Mappings tab, open the mapping plugin for the mapper row that the order should be rearranged.
2. Click the three-dotted icon ( ) that is available in the right of the mapper row. This will display the rearranging dialog. The user can select the Top, Up, Down, or Bottom button to rearrange the mapper row accordingly.



# Onboarding Comparison Screen

When a large amount of data is onboarded into the system, it is sometimes necessary to preview the Target object changes before the data is onboarded. The Onboarding Comparison Screen allows users to preview the onboarding data, assess each of the changes, and provide users an option to accept or reject the changes that will affect the Target object (or its related object).

When configured with a Mapper Configuration, the Onboarding Comparison Screen compares the current Target object values with the expected Target object value in the configured Mapper Configuration context. With the Source object selected, users can preview the changes affecting the Target object. Additionally, an option is provided to partially / entirely accept the affected changes. If the user chooses to accept some but not all the changes, the Mapper Configuration setup entity will be executed with only the selected changes written on the Target object.

Generally, the Onboarding Comparison Screen is used if user discretion is required regarding the acceptance / rejection of the onboarding changes. If all of the data can be onboarded without the user's discretion, then the user can choose to use any of the methods described in the **Executing Mapper Configuration Setup Entity** to directly onboard the data.

As the Onboarding Comparison Screen requires an object selection, it can be accessed in many different ways. Below are the standard ways of accessing it:

- Through the Search by Card Screen or any other search screens using the Forward Screen Action button. For more information, refer to **Using Search by Cards Screen with Data Onboarding Tasks** topic within this guide.
- Mapping the Onboarding Comparison Screen to be displayed as the primary screen when the Source object is selected in the Web UI. For information on mapping the screen for a particular object type, refer to the **Mappings** topic within the **Web User Interfaces** section of **STEP Online Help**.
- Added as a tab page within different screens configured to view the Source object. To add to an existing screen, use the Sub Screen Tab Page configuration, described in the **Tab Pages** topic in the **Web User Interfaces** section in **STEP Online Help**.

## Prerequisites

It is expected that anyone configuring the Onboarding Comparison Screen is familiar with the Web UI Designer, as basic concepts for working with the designer are not covered in this section. In addition, the user must have appropriate privileges to access the designer. Additional information can be found in the **Designer Access** topic of the **Web User Interfaces** section in **STEP Online Help**.

## Onboarding Comparison Screen Configuration

This topic provides details on how to add and configure the Onboarding Comparison Screen in the Web UI. Methods to access the Onboarding Comparison Screen are also defined in the 'Configuring Access to an Onboarding Comparison Screen' section later in this topic.

### Configuration Prerequisites

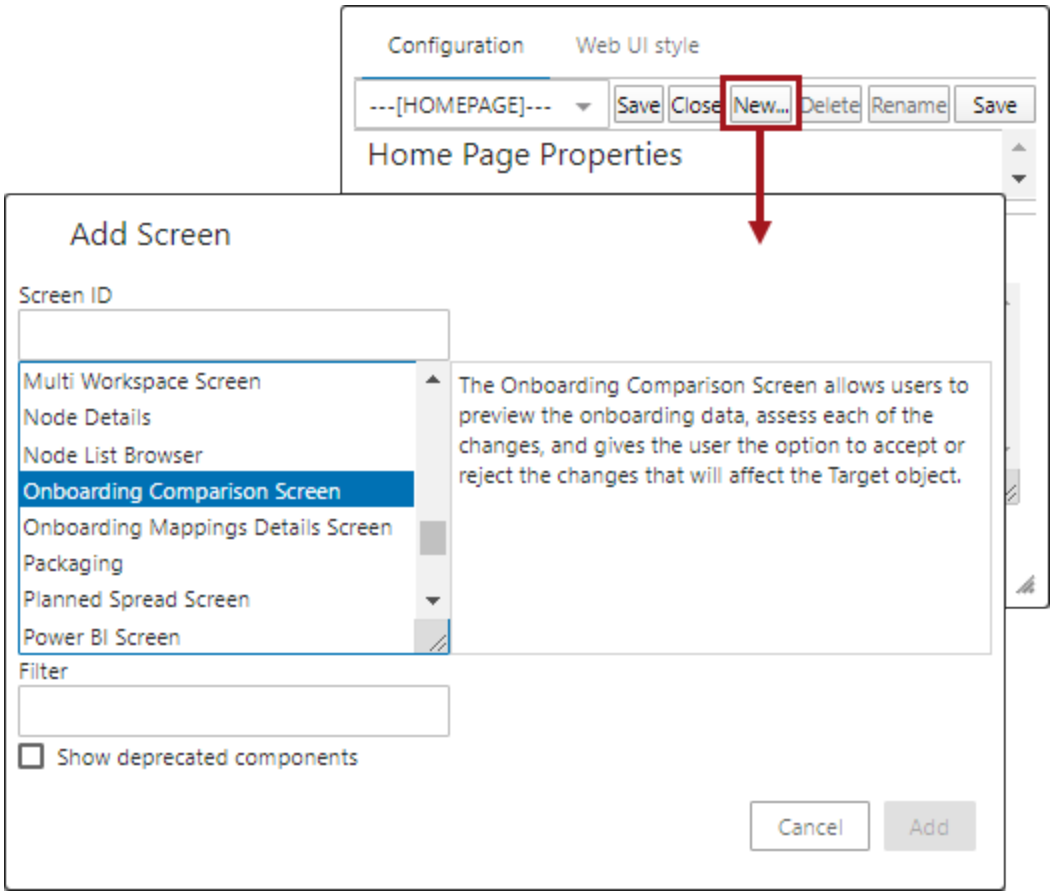
It is expected that anyone configuring the Onboarding Comparison Screen is familiar with the Web UI Designer as basic concepts for working with the designer are not covered in this section. In addition, the user must have appropriate privileges to access the designer. Additional information can be found in the **Designer Access** section of the **Web User Interfaces** documentation.

The required Mapper Configuration setup entity has to be readily available in the system. For creating and configuring Mapper Configurations, refer to **Configuring Mapper Configuration Setup Entity** topic within this guide.

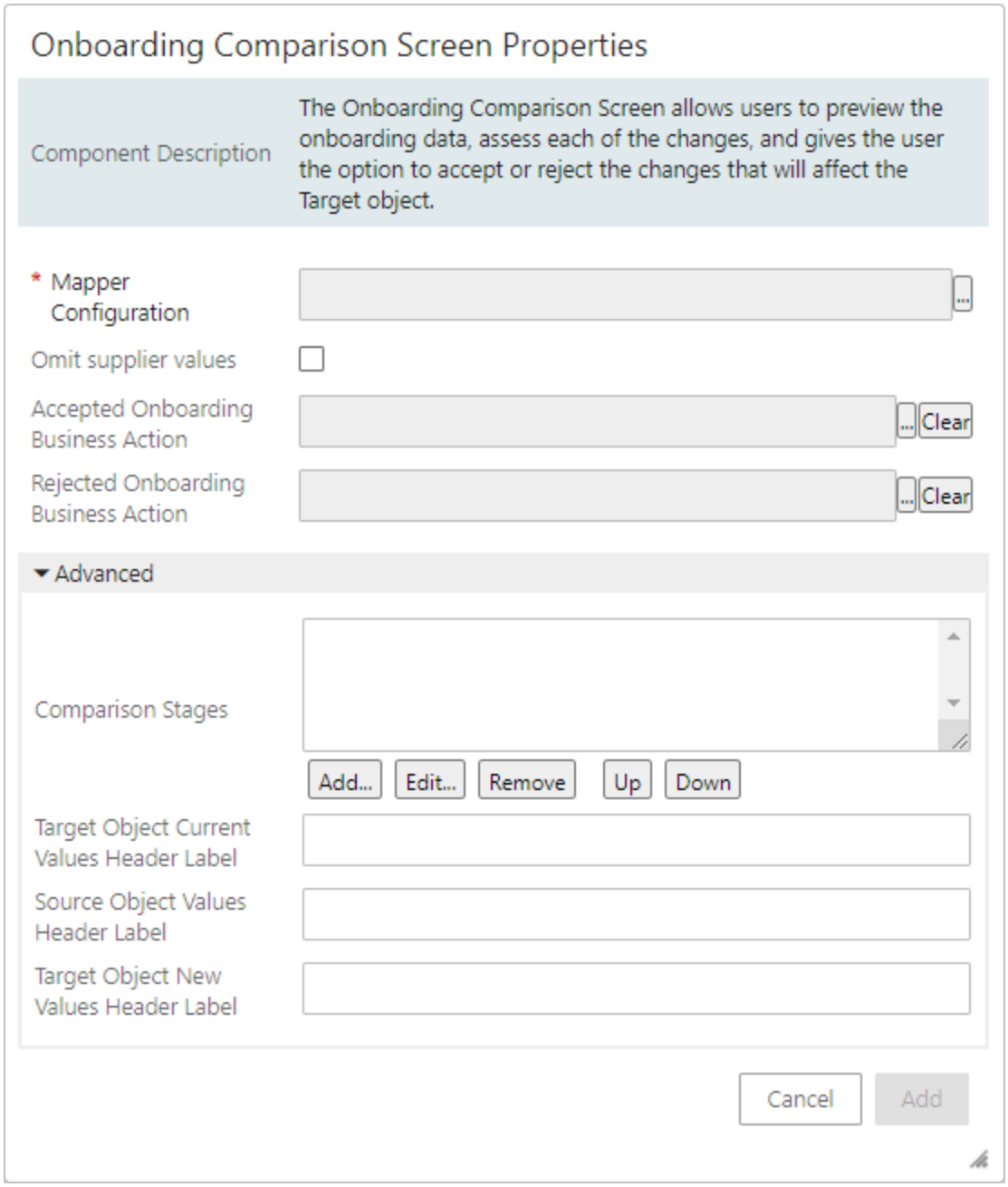
### Configuration Process

In the designer, on the HOMEPAGE create a new screen by selecting 'New.' Select the 'Onboarding Comparison Screen,' and add a name for the screen in the 'Screen ID' field. Click on the 'Add' button.

**Note:** The Screen ID you use should be something descriptive since this will be the name shown on any design component list in the Web UI. Screen IDs cannot be reused and must be unique.




Once this screen is created, the Add Component - Onboarding Comparison Screen Properties designer screen displays, which contains the following parameters for configuration:




**Mapper Configuration:** Select a Mapper Configuration setup entity that will be used as the context for comparison. The Mapper row(s) / Mapping plugin(s) available within the configured Mapper Configuration setup entity will be listed in any of the Onboarding Comparison Screen stages based on their functionality. For example, all Mapper rows of Attribute Mapping plugins will be listed in the Attributes stage, and all the Mapper rows of Object to Object Mapping plugins that onboards data from the Source object attribute to the Target object attribute will also be listed in the Attributes stage. To know more about the logic of how the Mapper rows / Mapping plugins are listed in different stages of the Onboarding Comparison Screen, refer to **Using Onboarding Comparison Screen Interface** topic within this guide.

**Omit supplier values:** This optional parameter allows users to hide the column that displays the Source object's current values in the Onboarding Comparison Screen.

**Accepted Onboarding Business Action:** This optional parameter allows users to configure a business action to be executed when the user clicks the **Onboard** button in the Confirm stage of the Onboarding Comparison Screen.

**Rejected Onboarding Business Action:** This optional parameter allows users to configure a business action to be executed when the user clicks **Reject Update** button in the Onboarding Comparison Screen.

**Comparison Stages:** This parameter allows adding multiple phases of Onboarding that are displayed as individual stages within the Onboarding Comparison Screen. Each stage can only be added once in this parameter. The order that the stages are listed determines the order that they will be displayed within the Onboarding Comparison Screen. Once added, no additional configuration is required. Clicking the Add button next to the Comparison Stages parameter displays an Add Component dialog with the option to add any of the following stages:

- **Application Mapping Stage:** Adding this stage will display the 'Applications' stage in the Onboarding Comparison Screen. Along with displaying the Mapper rows of the Application Mapping plugin, the following Mapper rows from other Mapping plugins are also displayed in this stage:
  - The Object to Object Mapping plugins that are configured to create an application record under the Target object are compared / displayed within the Applications stage.
  - The Business Action Mapping plugin that creates or updates an application record in the Target object is displayed for comparison in the Applications stage.
  - The Children of Type Mapping plugins with Target object's children as an application object are compared / displayed in the Applications stage.

  For the Application Mapping Stage to display within the Onboarding Comparison Screen, the selected Source object must meet all of the following criteria:

  - The Source application must have references to a vehicle and a part type.
  - There must be a reference established between the Source vehicle and the Target vehicle.
  - There must be a reference established between the Source part type and the Target part type.

- **Attribute Mapping Stage:** Adding this stage will display the 'Attributes' stage in the Onboarding Comparison Screen. Along with displaying the Mapper rows of the Attribute Mapping plugin, the following Mapper rows from other Mapping plugins are also displayed in this stage:

  - All Mapper rows from the Automatic Mapping plugins.
  - The Object to Object Mapping plugins that are configured to copy the attribute values from the Source object to the Target object.

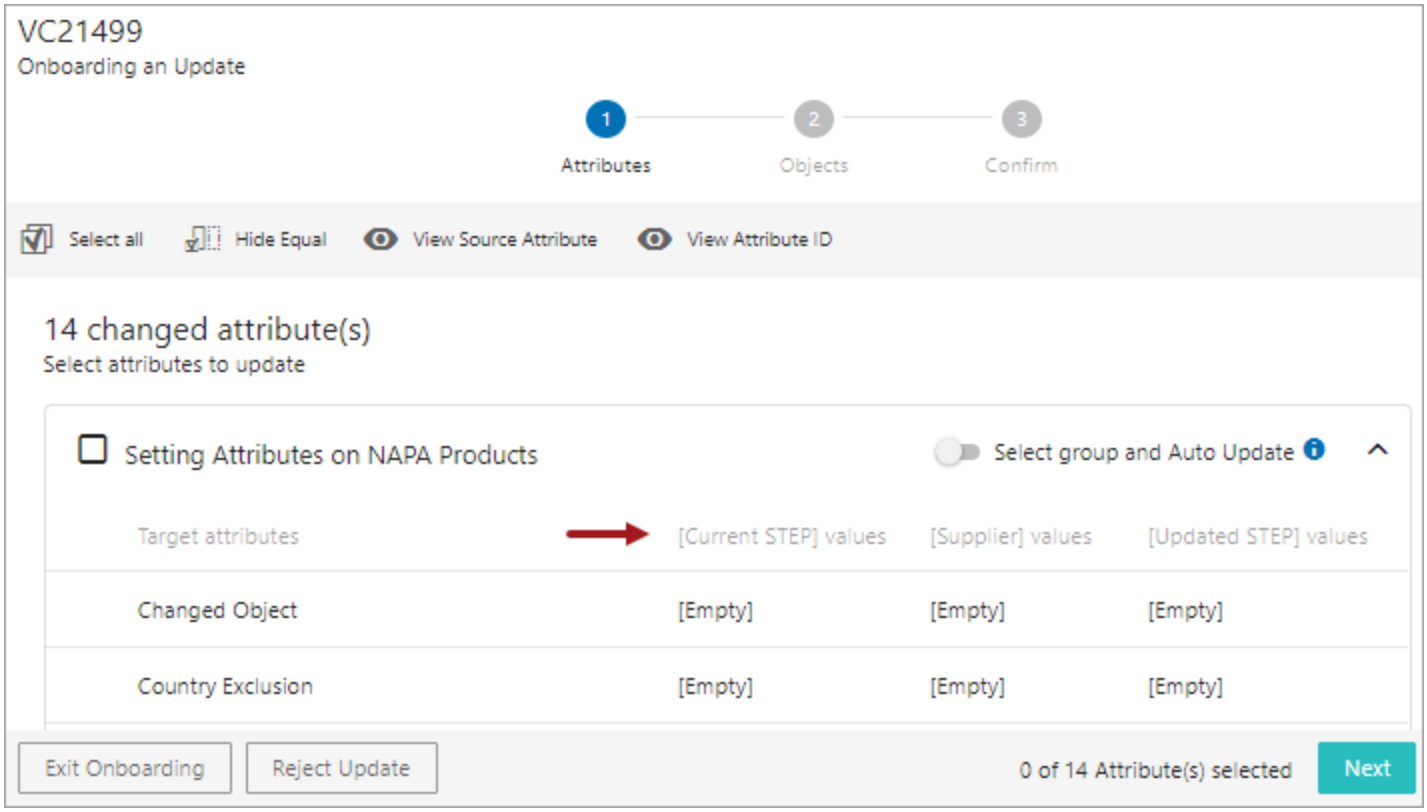
- **Objects Mapping Stage:** Adding this stage will display the 'Objects' stage in the Onboarding Comparison Screen. This stage will display the Object to Object Mapping plugins that are not logical to be displayed for comparison in the other stages of the Onboarding Comparison Screen. All the Children of Type Mapping plugins that do not have the Target object's children as application objects are also displayed for comparison in this stage.

Use the Up / Down buttons to rearrange the order of display for each stage.

**Important:** Only the Mapping Plugins and Mapper rows that are in the enabled (active) state within their Mapper Configuration are displayed for comparison in the Onboarding Comparison Screen. The disabled Mapping plugins and Mapper rows will not be considered for comparison.

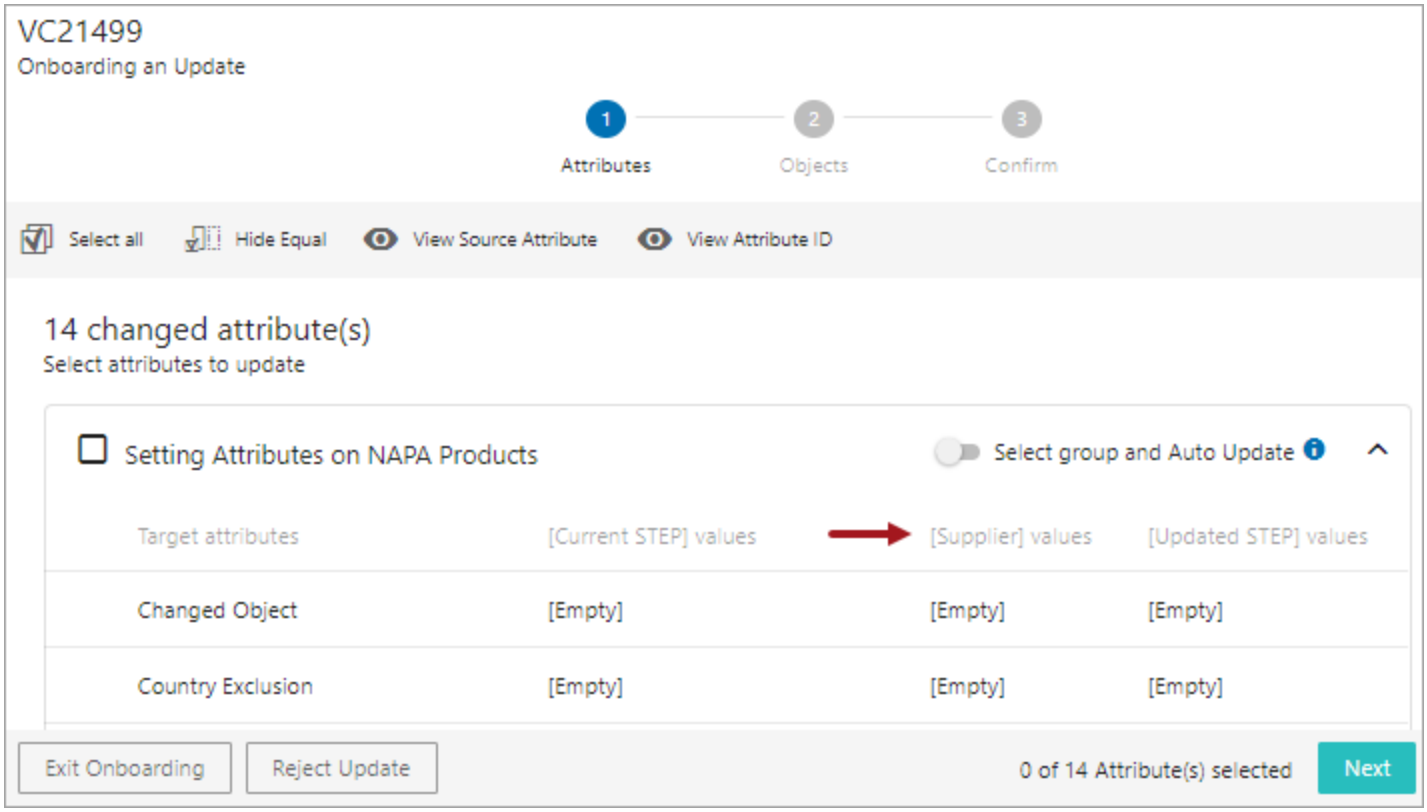
**Target Object Current Values Header Label:** If required, specify a custom label for the column header that displays the current populated value in the Target object. If nothing is added, the label default will display as '[Current STEP] values.'

This is how the column header will display on the Onboarding Comparison Screen.



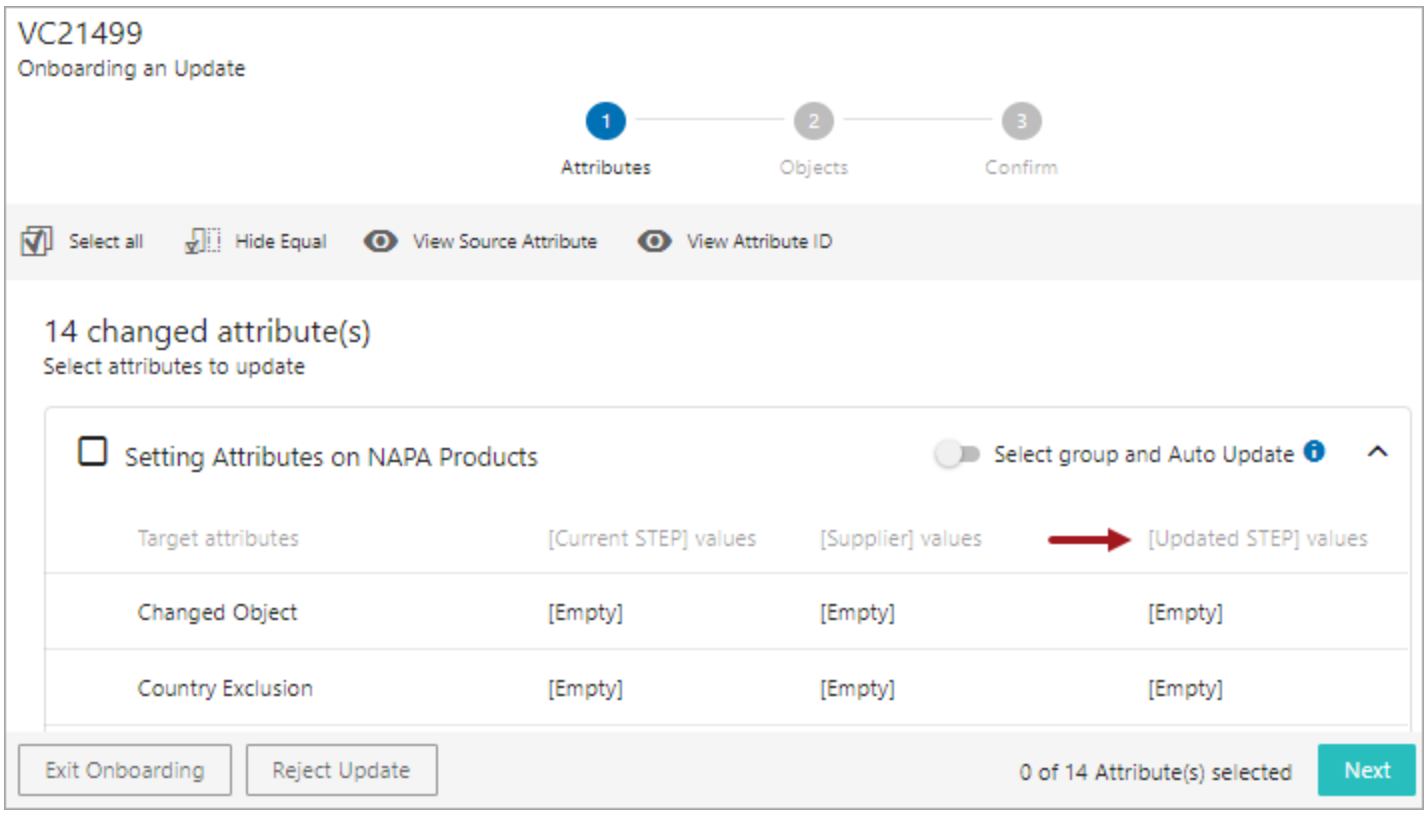

**Source Object Values Header Label:** If required, specify a custom label for the column header that displays the populated value in the Source object. If nothing is added, the label default will display as ' [Supplier] values.'

This is how the column header will display on the Onboarding Comparison Screen.




**Target Object New Values Header Label:** If required, specify a custom label for the column header displaying the value that gets populated in the Target object after onboarding. If nothing is added, the label default will display as '[Updated STEP] values.'

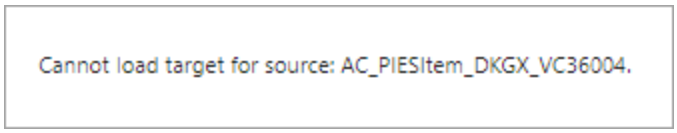
This is how the column header will display on the Onboarding Comparison Screen.




### Configuring Access to an Onboarding Comparison Screen

Any number of Onboarding Comparison Screens can be added to a Web UI. An admin user must configure how users can access the Onboarding Comparison Screens.

**Important:** For the Onboarding Comparison Screen to display, there must be a reference established between the Source object and a single Target object. The Onboarding Comparison Screen does not support the Source object with multiple Target objects. If no Target object exists, or multiple Target objects exist for the selected Source object, then the following error message is displayed on the screen.

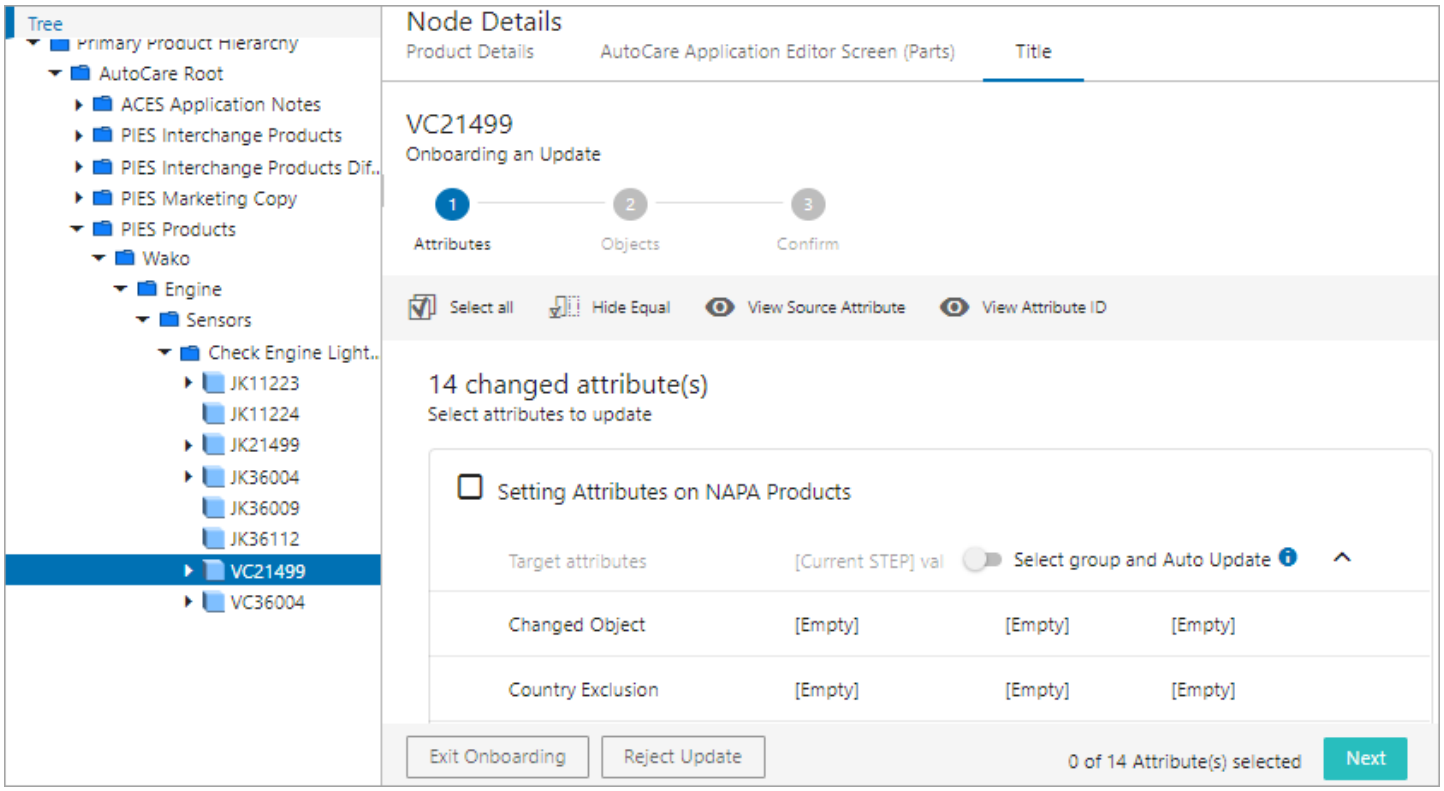



There are multiple ways that the screen can be accessed in the Web UI. These are the standard ways in the Automotive solution that the Onboarding Comparison Screen can be accessed:

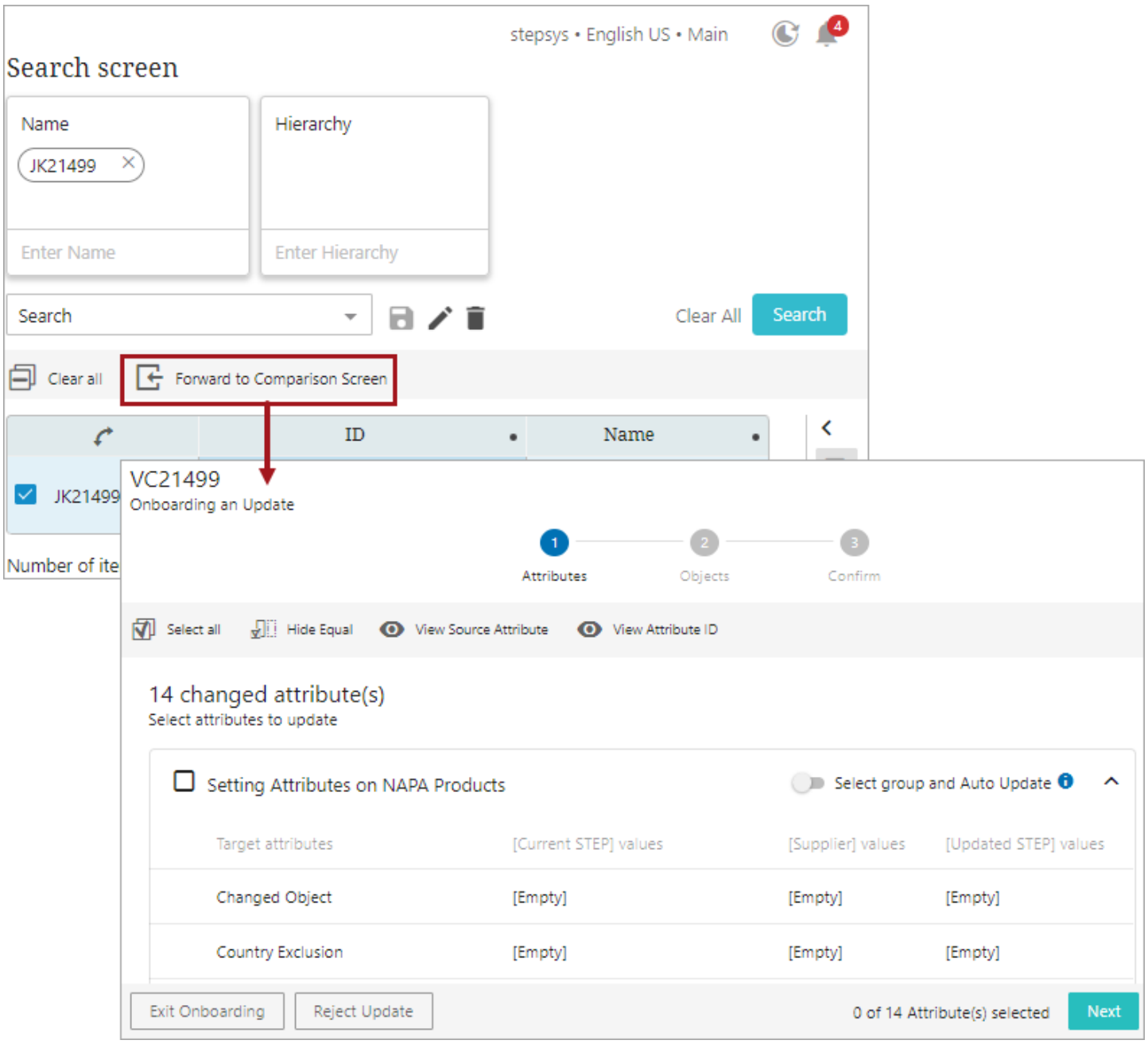
- Mapped to the Source Object using Main Properties. For more information about mapping Web UI screens, refer to **Configuring the Parameters** in the **Main Properties** topic of the **Web User Interfaces** documentation.

- Added as a tab page within different screens that are configured to view the Source object. For more information about sub screen tab pages, refer to the **Tab Pages** topic in the **Node Details Component** section of the **Web User Interfaces** documentation.

  The example below shows the Onboarding Comparison Screen displayed as a sub screen tab page (with the label 'Title') on a Node Details screen and is displayed when a Source object is selected.




- Through the Search by Card Screen or any other search screens using the Forward Screen Action button. This method requires selecting the Source object in the Search by Card result table and navigating the selected Source object using the Forward Screen Action button. For more information, refer to **Using Search by Cards Screen with Data Onboarding Tasks** topic within this guide.




- Accessing the Node Details Screen of the Source object via the Tree navigator and navigating to the Onboarding Comparison Screen via the Forward Screen Action button configured in the Node Details Screen. For more information about Forward Screen Action component, refer to the **Forward Screen Action Component** topic within this guide.

## Using Onboarding Comparison Screen Interface

The Onboarding Comparison Screen interface is designed to allow users to preview the onboarding data, assess each of the changes, and give the user an option to accept or reject each change that will affect the Target object. This screen allows users to distinguish and allow only the required changes to be onboarded.

Each Onboarding Comparison Screen can be configured to display up to three Mapping stages. Because the final step, called the Confirm stage, is set by default, there can be a maximum of four stages displayed in each Onboarding Comparison Screen.

Following are the Mapping stages that can be made available to be configured in the Onboarding Comparison Screen:

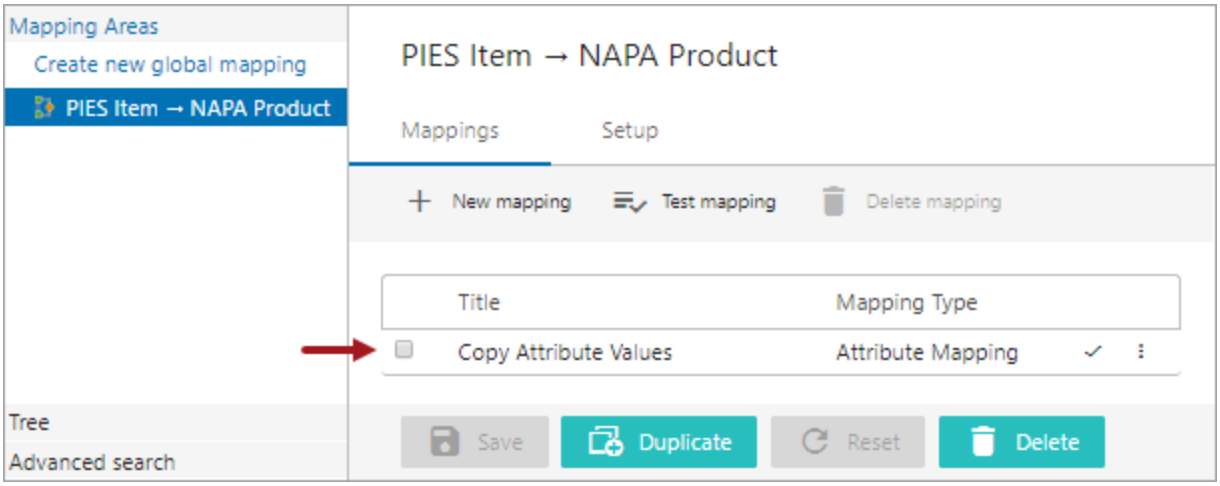
- Applications stage
- Attributes stage
- Objects stage
- Confirm stage

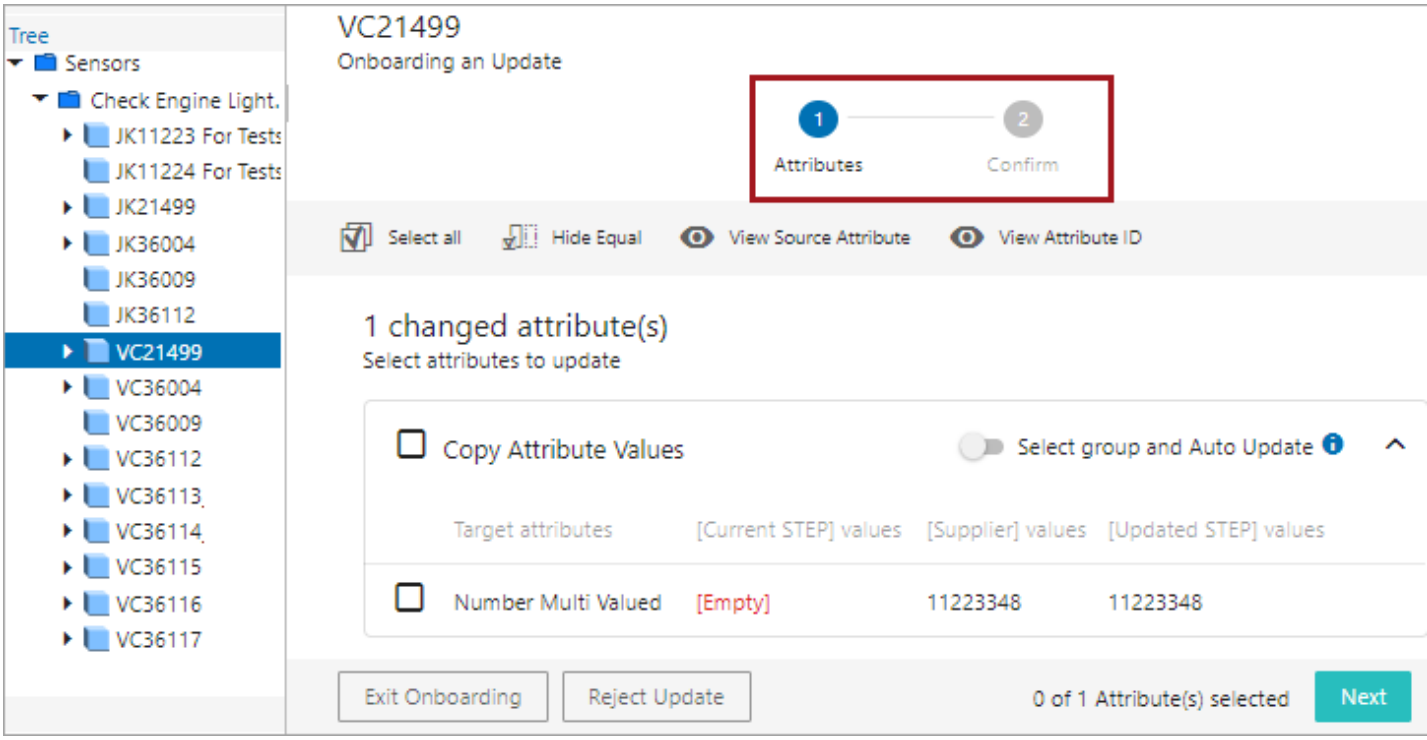
Of all the stages mentioned above, the Confirm stage gets displayed only when any of the other three stages are available. The Applications stage, Attributes stage, and Objects stage are displayed when ALL of the following conditions are met:

- One or more of the associated Mapping plugin is enabled. And at least one of the Mapper rows listed within these Mapping plugins is in the active state.
- The Comparison Stages parameter available within the Onboarding Comparison Screen Properties designer screen is either empty or has the required stage(s) listed in it.
- The Source object has only one Target object. Only single-target references are compatible. Having multiple targets will fail to load the screen.

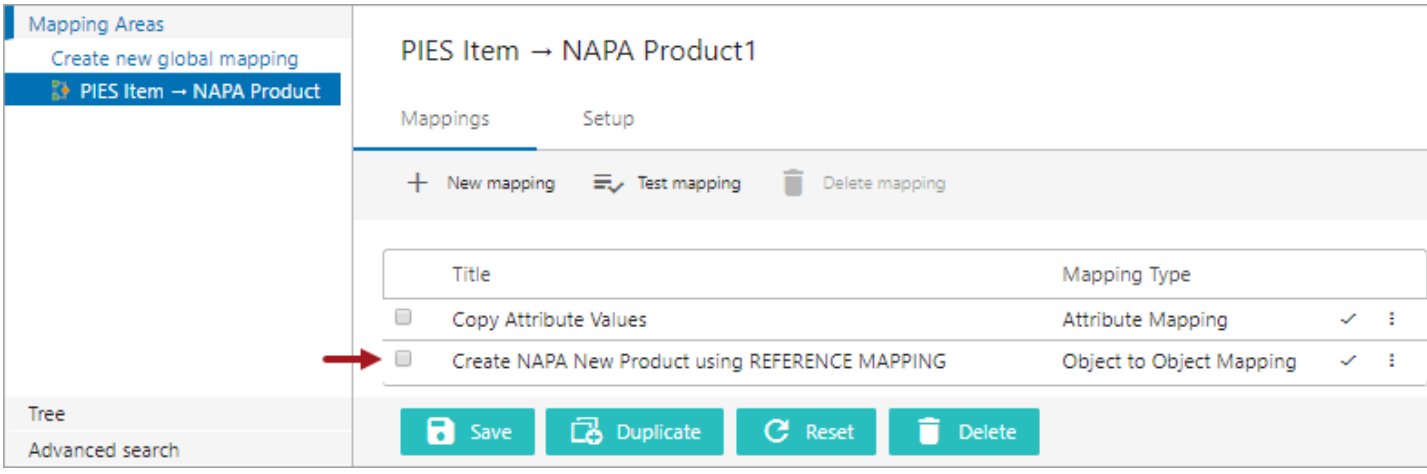
Below is an example of how the Onboarding Comparison Screen will behave in different cases:

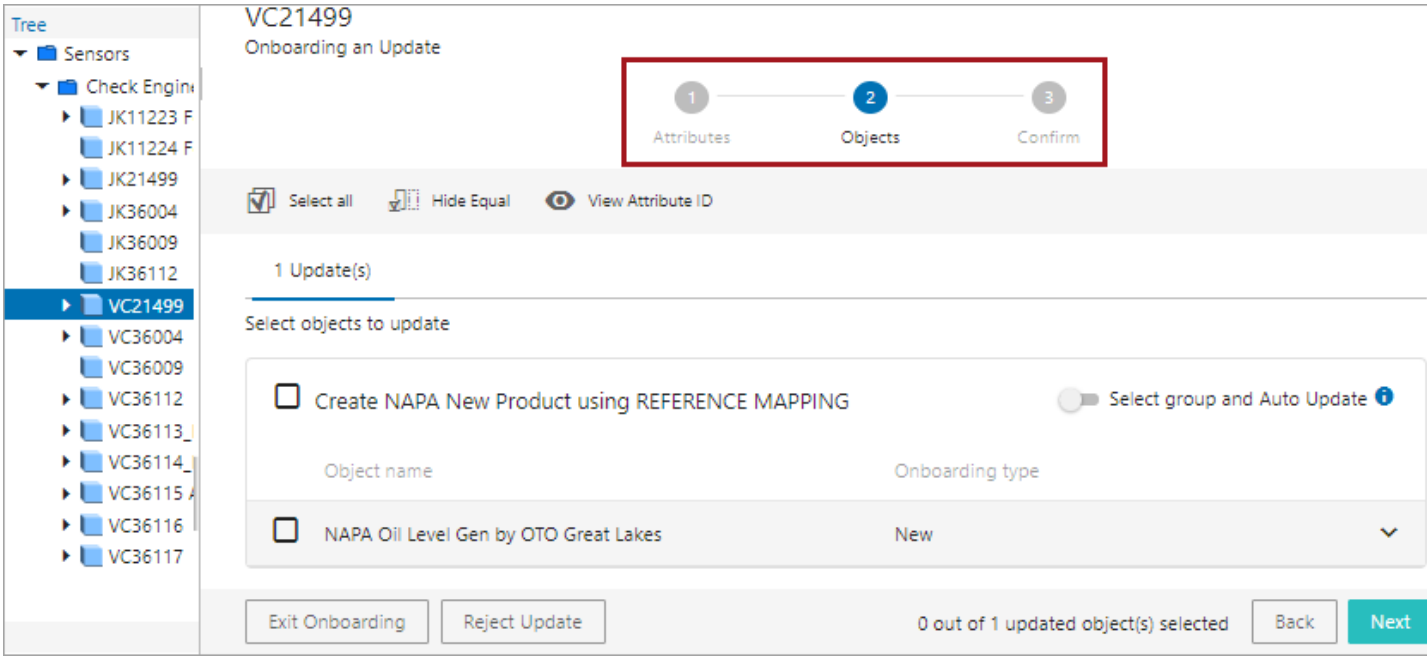
- If the Mapper Configuration constitutes only the Attribute Mapping plugin, then the Onboarding Comparison Screen configured to display the Mapper Configuration will only display the Attributes and the Confirm stage. In the example below, because the 'PIES Item --> NAPA Product' Mapper Configuration has only the Attribute Mapping Plugin, the Attributes stage and the Confirm stage are displayed when the Source object is selected (as shown below).



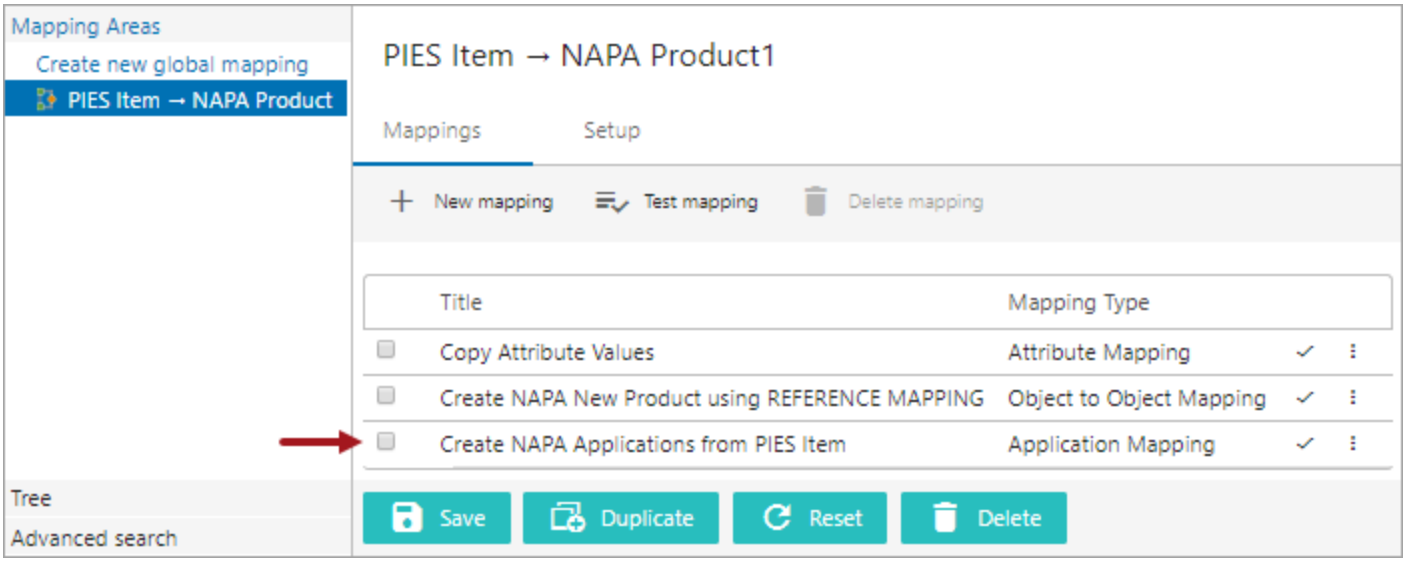


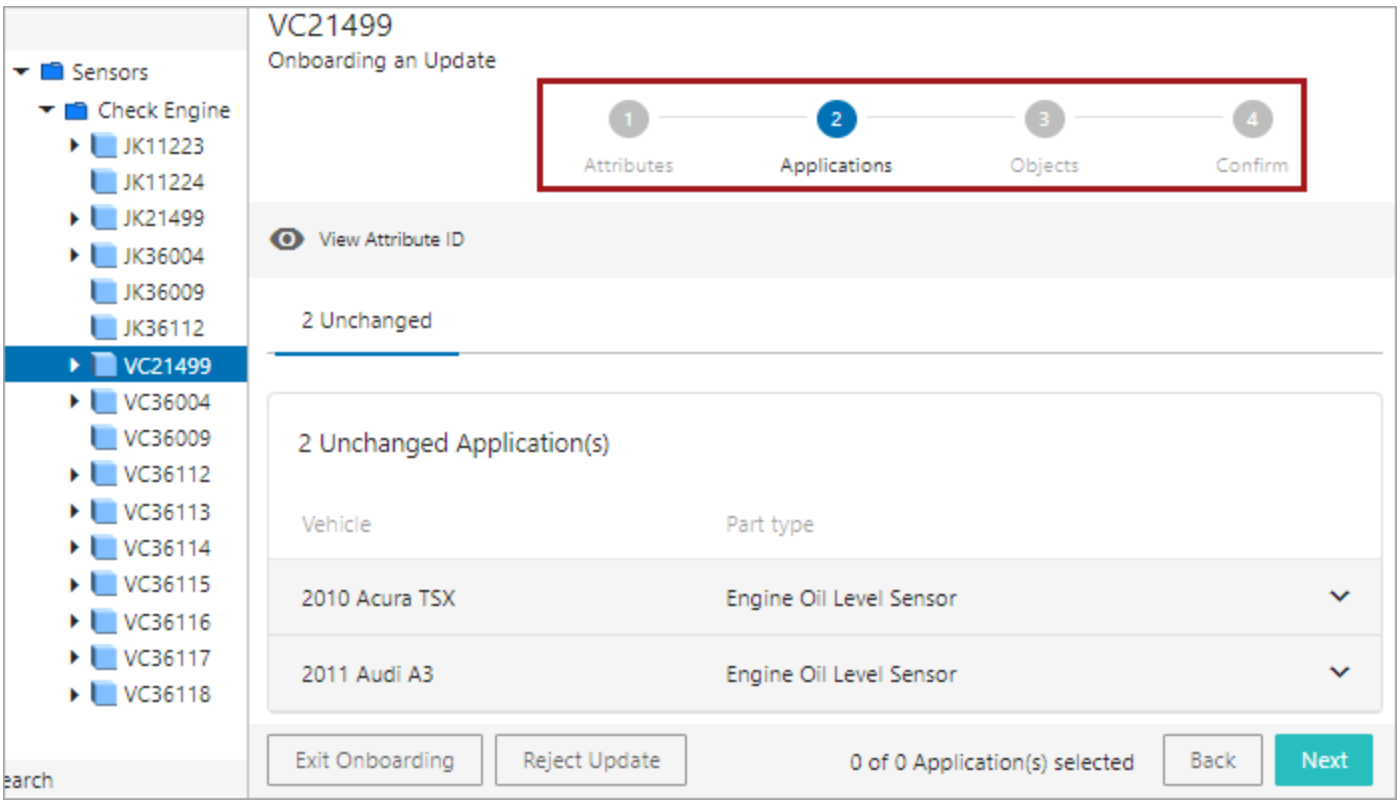

- Additionally, if the 'Reference' type of Object to Object Mapping plugin is added to the 'PIES Item --> NAPA Product' Mapper Configuration setup entity, then the available stages display as Attributes stage, Objects stage and the Confirm stage (as shown below).






- Additionally, consider being able to add the Application Mapping plugin for 'PIES Item --> NAPA Product' Mapper Configuration setup entity. In the screenshot below, the available stages display as Attributes stage, Objects stage, Applications stage and the Confirm stage.

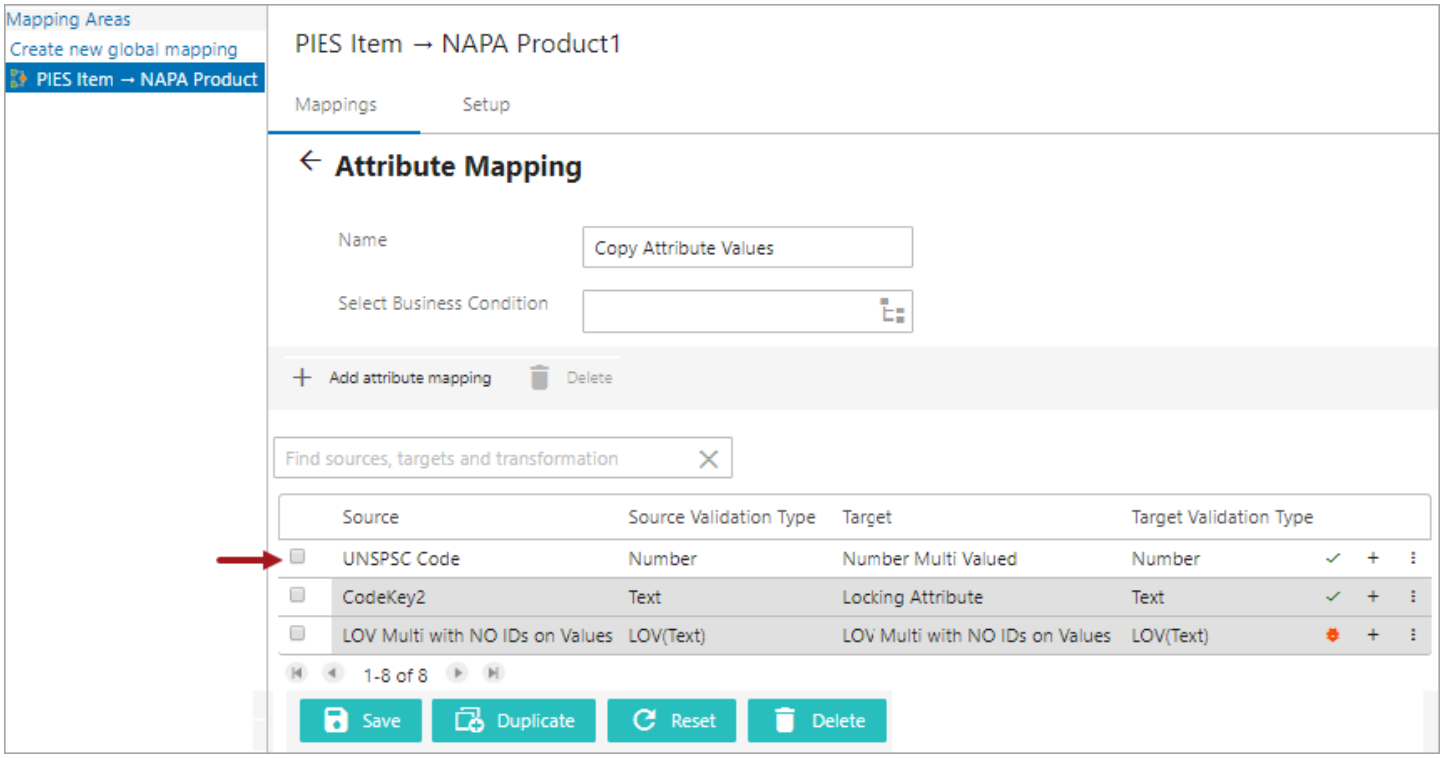


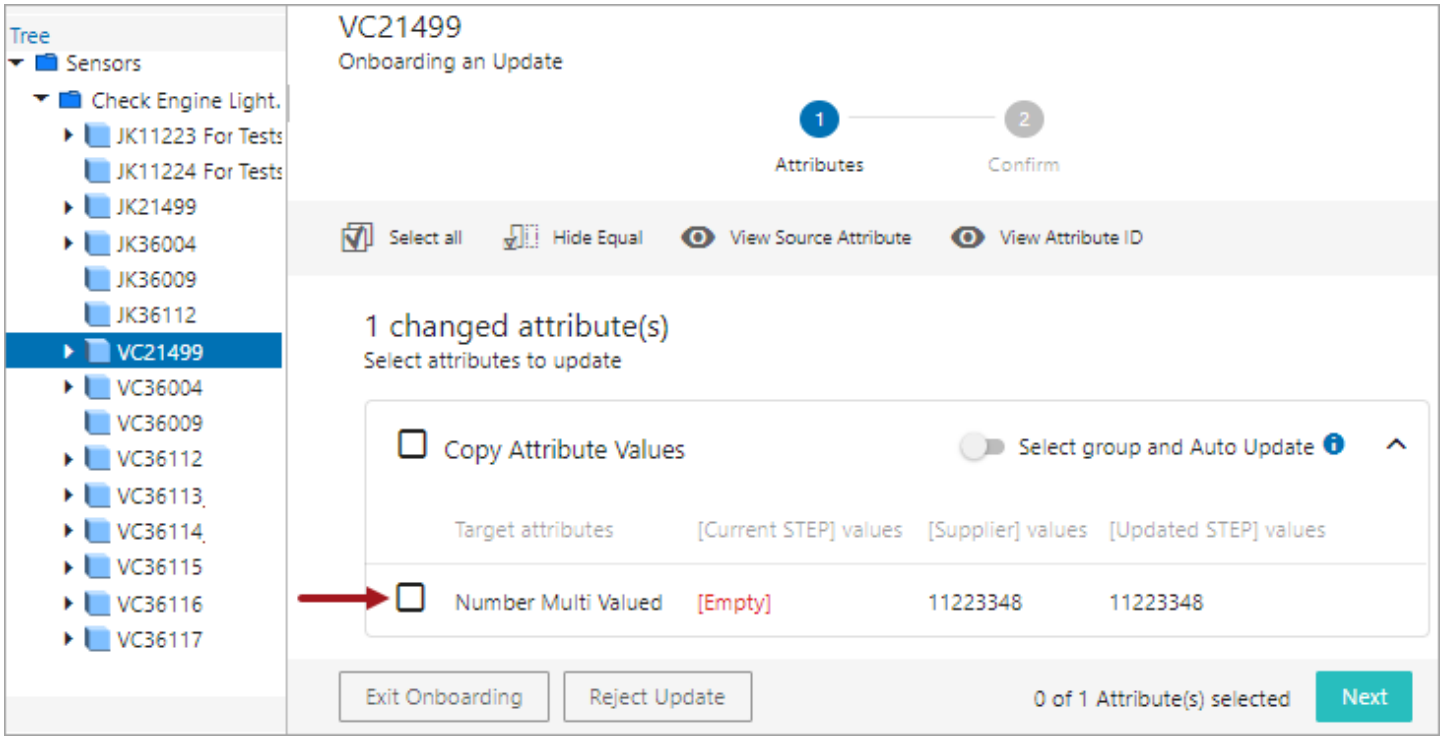



The Applications, Attributes, and the Objects stage gets displayed in the Onboarding Comparison Screen only when there is any data to show for comparison. If there is nothing to compare in a stage, then such stage will be dropped from being displayed within the screen.

Further, only those Mapping plugins / Mapper rows that are active will be displayed within each stage. For example, only those Mapper rows / Attribute Mapping plugins will be displayed that are active in the Mapper Configuration when the Attributes stage is selected.

In the example below, the first screenshot shows that only one Mapper row within an Attribute Mapping plugin is active. Hence, the Onboarding Comparison Screen compares any object only in the context of the rule defined in that active Mapper row (as shown in the second screenshot below). The other disabled Mapper rows are not considered for comparison.


Mapping Areas
Create new global mapping
PIES Item → NAPA Product
PIES Item → NAPA Product1
Mappings
Setup
Attribute Mapping
Name
Copy Attribute Values
Select Business Condition
Add attribute mapping
Delete
Find sources, targets and transformation
Source
Source Validation Type
Target
Target Validation Type
UNSPSC Code
Number
Number Multi Valued
Number
CodeKey2
Text
Locking Attribute
Text
LOV Multi with NO IDs on Values
LOV(Text)
LOV Multi with NO IDs on Values
LOV(Text)
1-8 of 8
Save
Duplicate
Reset
Delete


Tree
Sensors
Check Engine Light.
JK11223 For Tests
JK11224 For Tests
JK21499
JK36004
JK36009
JK36112
VC21499
VC36004
VC36009
VC36112
VC36113
VC36114
VC36115
VC36116
VC36117
VC21499
Onboarding an Update
1
Attributes
2
Confirm
Select all
Hide Equal
View Source Attribute
View Attribute ID
1 changed attribute(s)
Select attributes to update
Copy Attribute Values
Select group and Auto Update
Target attributes
[Current STEP] values
[Supplier] values
[Updated STEP] values
Number Multi Valued
[Empty]
11223348
11223348
Exit Onboarding
Reject Update
0 of 1 Attribute(s) selected
Next

**Note:** The Onboarding Comparison Screen has many configuration options. The Web UI administrator has control over the Stages that are allowed to be displayed within the Onboarding Comparison Screen. If your screen displays differently than the examples within this section, contact your Web UI administrator. For more information on configuring an Onboarding Comparison Screen, refer to the **Onboarding Comparison Screen Configuration** topic within this guide.

This section addresses the following:

- **Using Attributes Stage**
- **Using Applications Stage**
- **Using Objects Stage**

### Using Applications Stage

Onboarding application records from the Source object to the Target object can be as simple as copying over a few attribute values, or it can be more complicated and requires transformations, Lookup tables, and other complex operations. In this process, it might become necessary for the user to view and assess the individual changes affecting the Target's application records before the actual onboarding of information happens. Instead of manually switching between the Source and the Target objects to compare the application records, the Applications stage of the Onboarding Comparison Screen displays the Target object's current application records (if any are present) against the expected changes that could happen after onboarding. The screen is intended to showcase the changes (created / updated / deleted) that could happen in the Target object's application records if onboarded.

For the Application Mapping Stage to display within the Onboarding Comparison Screen, all of the following criteria have to be met.

- The Source application must have references to a vehicle and a part type.
- There must be a reference established between the Source vehicle and the Target vehicle.
- There must be a reference established between the Source part type and the Target part type.

The Applications stage in the Onboarding Comparison Screen is designed to display the following details for comparison:

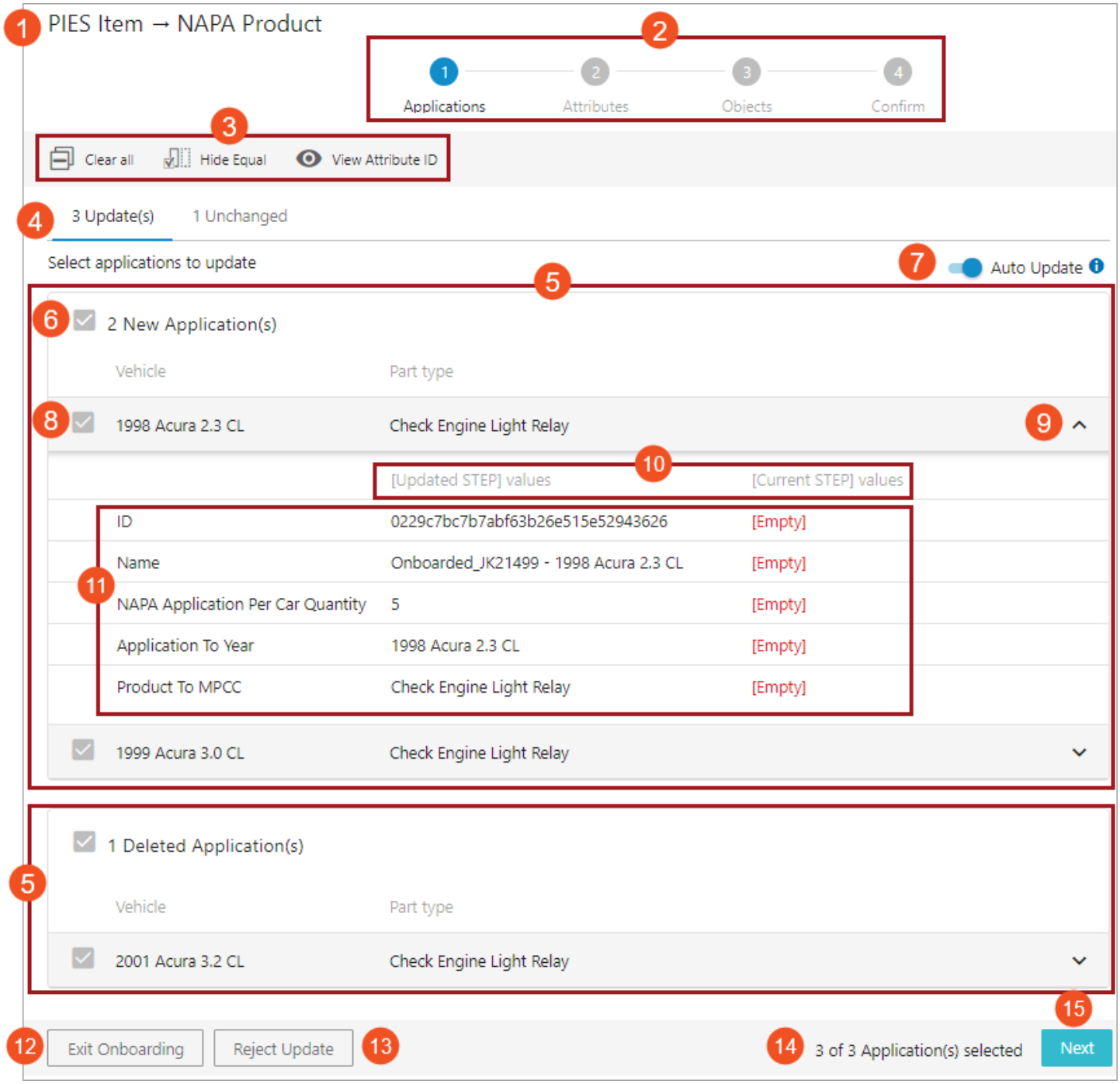
- The current Target object's application records, the expected Target object's application records, and their similarities and/or differences in the context of the **Application Mapping** plugins that are available in the Mapper Configuration.
- The current Target object's application records, the expected Target object's application records, and their similarities and/or differences in the context of the **Business Action Mapping plugin** that creates or updates an application record in the Target object.
- The current Target object's application records, the expected Target object's application records, and their similarities and/or differences in the context of the **Object to Object Mapping** plugin that creates or updates an application record in the Target object.
- The current Target object's application records, the expected Target object's application records, and their similarities and/or differences in the context of the **Children of Type Mapping** plugin that creates or updates an application record in the Target object.

**Note:** If an application record is created / updated / deleted by the Applications Mapping plugin and Children of Type Mapping plugin, then the information about the Part Type and the Vehicle that the Target object's application record is referencing to will be displayed in the Applications stage. The Part Type and Vehicle information will not be displayed if the application record is created / updated by Object to Object Mapping plugin and Business Action Mapping plugins.

**Important:** The Application Mapping plugin, Object to Object Mapping plugin, and the Business Action Mapping plugin never update an existing Target object's application record. Instead, any update made to the application record in the Source object will cause deleting the relevant application record in the Target object and creating a new application record, when onboarded.

This section addresses features and functionalities specific to the Applications stage of the Onboarding Comparison Screen. For information on the Onboarding Comparison Screen's general functionalities, refer to the **Using Onboarding Comparison Screen Interface** topic within this guide.

The Applications stage of the Onboarding Comparison Screen is composed of the following:



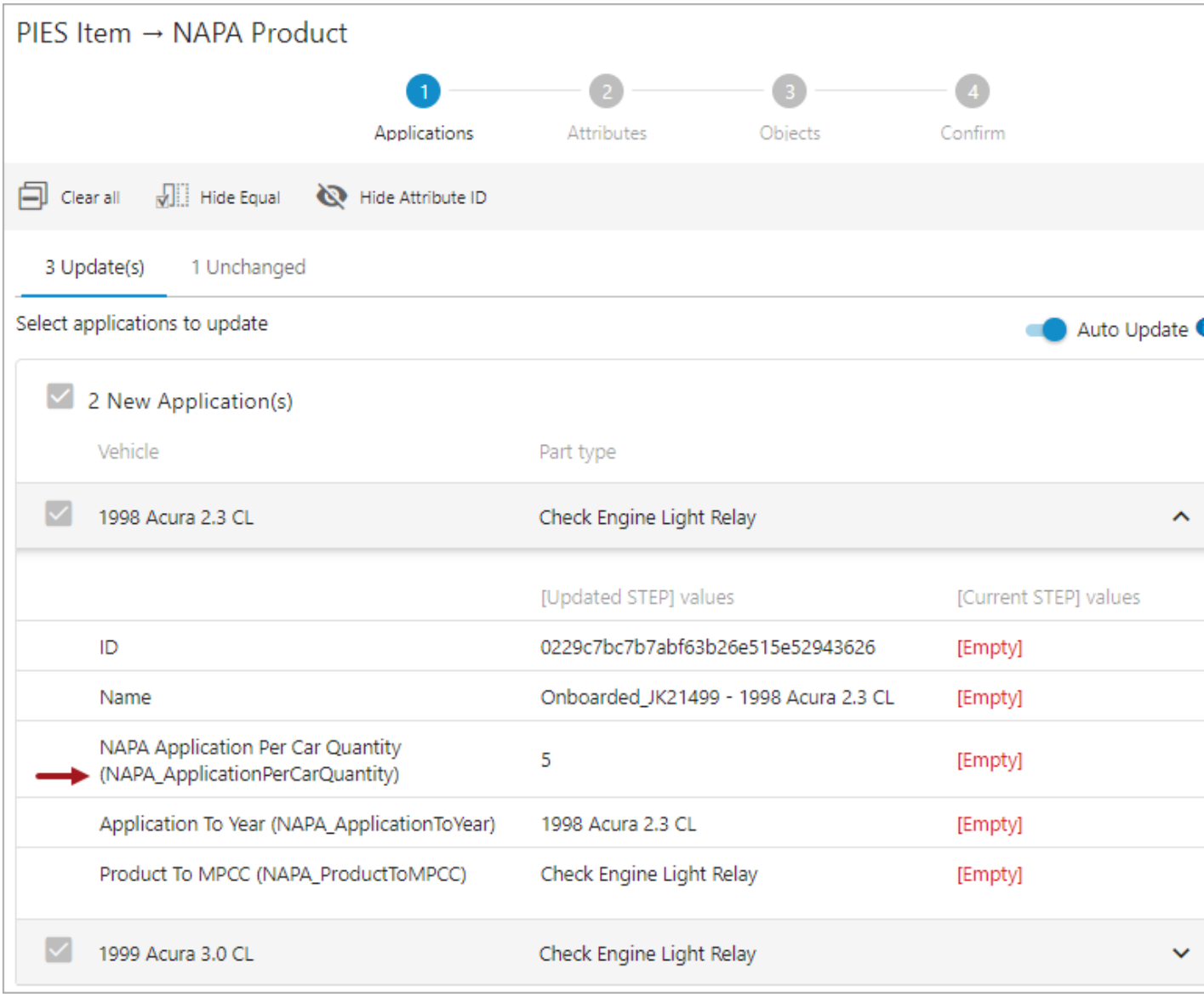

1. **Mapper Configuration name:** Displays the name of the Mapper Configuration being executed for the selected Source object.
2. **Progress Tracker:** An interactive tracker that displays all of the current Onboarding process with the selected stage highlighted in blue. Users may navigate between the stages by clicking on either the stage number or stage label. The highlighted stage shows the current stage of the Onboarding process in the Onboarding Comparison Screen. In the screenshot above, the Onboarding Comparison Screen displays the

Applications stage.

3. **Comparison table toolbar:**

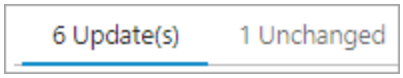
Contains action buttons related to the Applications stage of the Onboarding Comparison Screen. Available action buttons are hard coded to always display in the toolbar and cannot be configured in any way.

- **Select all / Clear all:** The 'Clear all' button displays when all of the Application records that are being created / updated / deleted are selected using the 'Target application selection box.' When any of the application record is unselected, the 'Select all' button will display. Clicking the 'Clear all' button will remove the selections within the 'Target application selection box.' Clicking this button will not clear the comparison table's values but rather disables the selection of 'Target application selection boxes.' Clicking the 'Select all' button enables all the 'Target application selection boxes.'
- **Hide Equal / Show Equal:** This action helps find out the rows that have the current Target object information and the expected Target object information of equal values. The Comparison table is always defaulted to display all rows. However, clicking on the 'Hide Equal' button will hide the rows with the same values. When the table displays with these rows compressed, the 'Show Equal' button will display within the toolbar instead of the 'Hide Equal' button.
- **View Attribute ID / Hide Attribute ID:** By default, the Comparison table does not display the Target application record's attribute ID. Clicking the 'View Attribute ID' button will make the attribute ID surrounded by parentheses to be indicated next to the attribute name. The following screenshot shows an example of the attribute ID displayed for the Target object's application record.



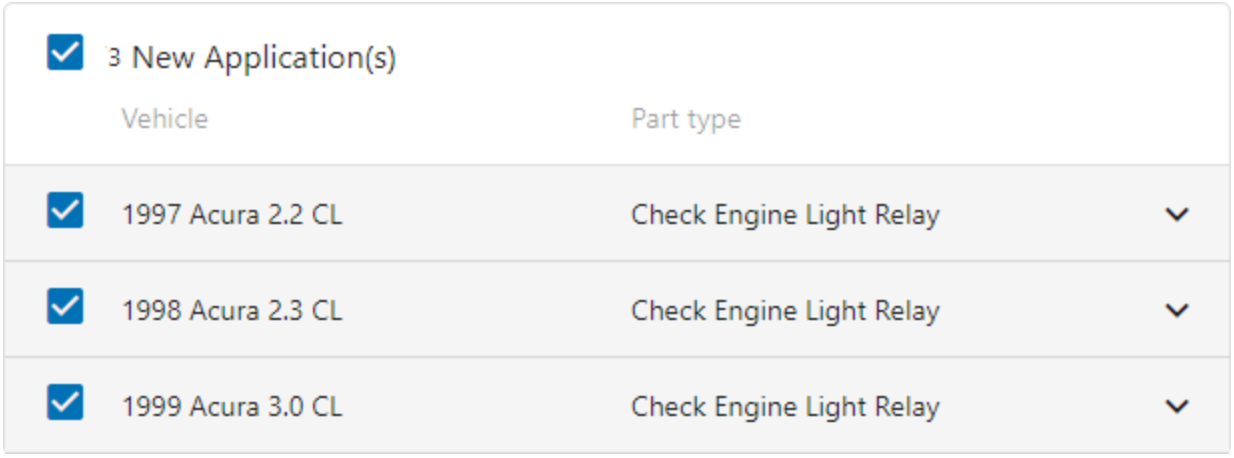

4. **Number of Application changes:**

   Lists the total possible number of Target object's application records that could get updated / created / deleted. Further, if any application records remain unchanged, then such application records are listed in the separate tab called 'Unchanged.' There are six application records in the screenshot below that are updated / created / deleted and one application record that remains unchanged.

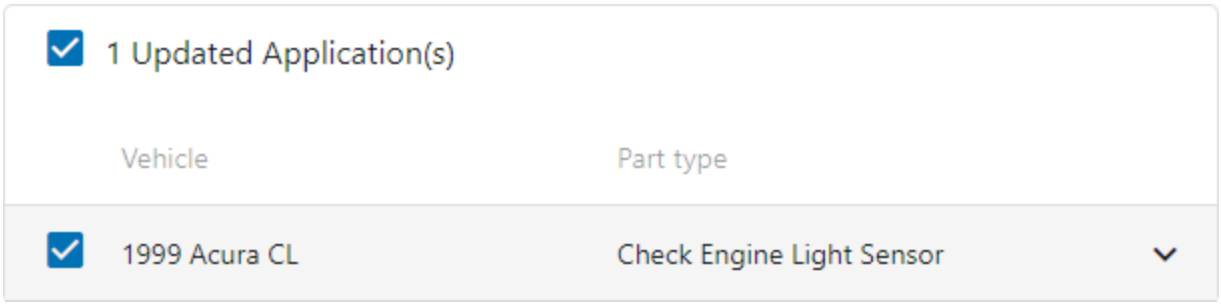



5. **Applications Mapping plugin canvas:** The Applications stage of the Onboarding Comparison Screen can have the following four canvases spread across two different tabs.

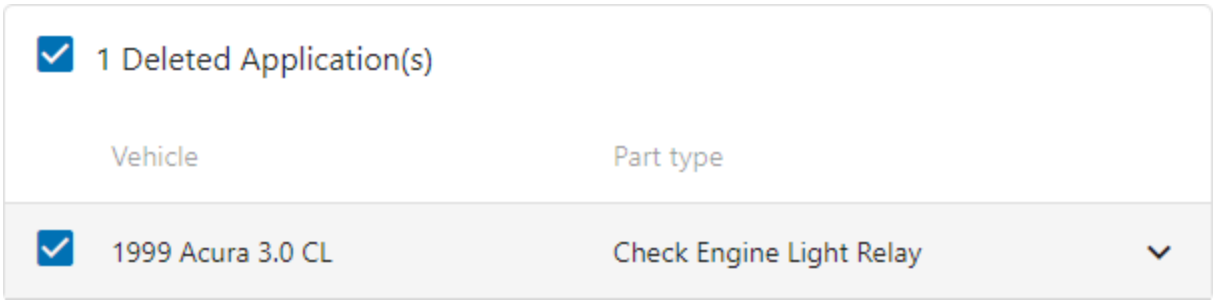
- **New Application(s):** This canvas contains all application records that get created when the Onboard button is clicked. Users can select / deselect individual application records that are to be included / excluded from onboarding. In the example screenshot below, the onboarding creates three new application records in the Target object.



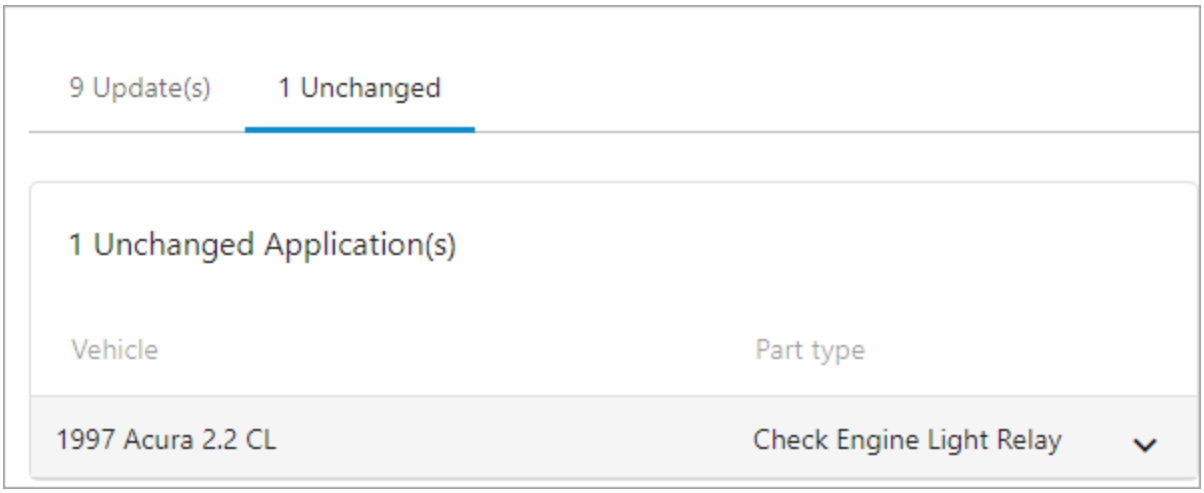

- **Updated Application(s):** This canvas lists all the application records that are present in the Target object and gets updated when the Onboard button is clicked. Users can select individual application records that needs to be updated. In the example screenshot below, the onboarding updates one existing application record in the Target object.




- **Deleted Application(s):** This canvas lists all the application records that are present in the Target object and gets deleted when the Onboard button is clicked. Users can select individual application records that are to be deleted. In the example screenshot below, the onboarding deletes one existing application record in the Target object.



- **Unchanged Application(s):** This canvas lists all the application records present in the Target object that remains unchanged when the Onboard button is clicked. This canvas is present in a separate tab called 'Unchanged' (shown in the screenshot below). Users cannot make any selections in this canvas. In the example screenshot below, one Target object's application record remains unaffected from onboarding.




Each Application Mapping plugin canvas except the Unchanged Application consists of the following:

- New / Updated / Deleted selection box
- Expand / collapse arrow
- Target application selection boxes

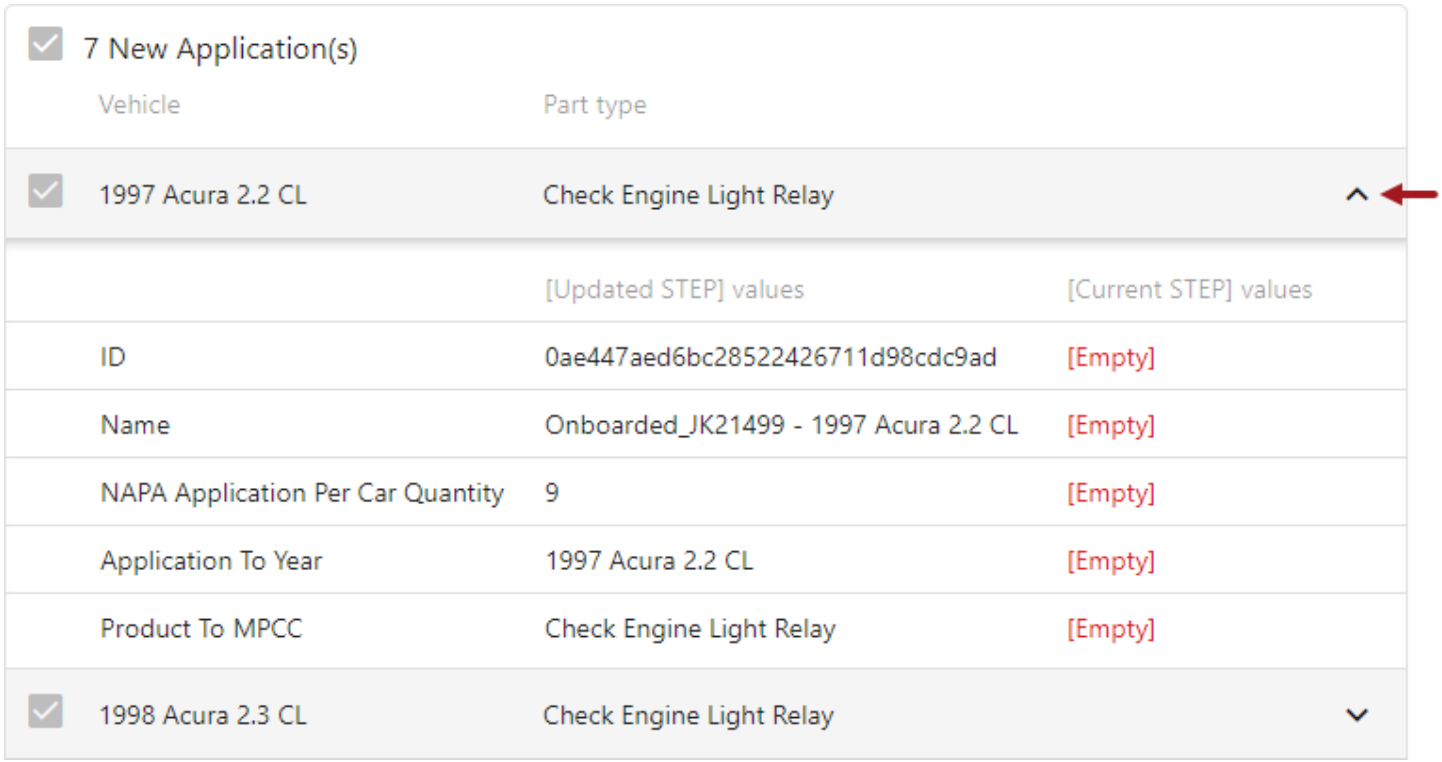
All the above mentioned options are explained in detail in the later section of this topic.

6. **New / Updated / Deleted selection box:** Allows for selection of all of the new / updating / deleting target object's application records that are displayed in the Applications stage of the Onboarding Comparison Screen. Selecting the checkbox also selects all the application records displayed within the respective canvas.
7. **Auto Update switch:** This switch is set to active by default. Leaving this button activated will ensure that all the application records listed in the Applications stage will be included for execution in all future Onboarding of the selected Source object. If the user intends to exclude any application records from being executed for the selected Source object, then the user can turn off the switch and deselect the non-required New / Updated / Deleted selection box or individual application records. The settings applied with this switch will be held as the system's memory to be set as default for all future execution methods for the selected Source object only. When the switch remains turned On, the New / Updated / Deleted selection box (along with their included application record selection boxes) is selected and are grayed out. That means no individual deselection of the application records is allowed, and all application records listed within the Applications stage will be onboarded.

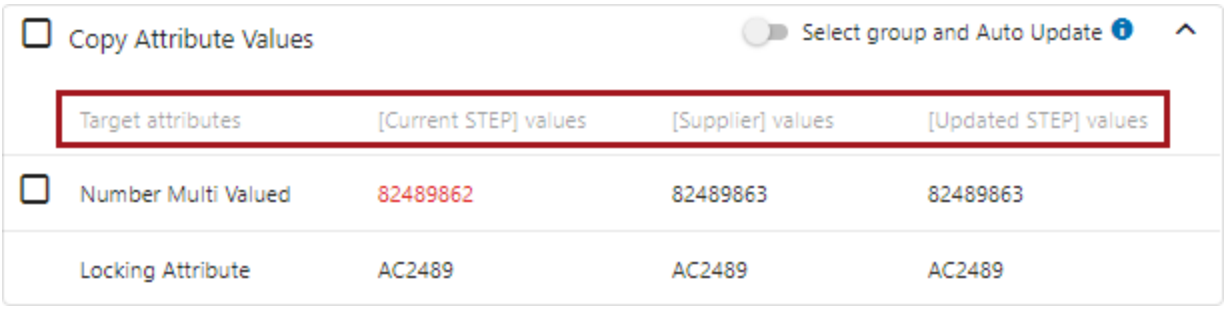
   After turning On / Off the Auto Update switch in the Applications stage, the user should click the Onboard button (available in the lower right corner of the Confirm stage) to save changes made to the Auto Update

switch. Users may safely navigate between the stages in the Onboarding Comparison Screen without losing unsaved selections, but users should not exit the Onboarding Comparison Screen without first navigating to the final screen and clicking the Onboard button.

8. **Target application selection box:** Allows for selecting each application record displayed within each plugin canvas. To select or deselect application records individually, select an application record by clicking the checkbox to the left of the Target application record's Vehicle name. To select multiple application records, either click on additional application records or use the 'Select all' action. When an application record is selected, the 'Select all' icon switches to a 'Clear all' icon. Use this to clear all selections, or click a checkbox again to deselect a particular application record.
9. **Expand / collapse arrow:** When the arrow is clicked, the details of the application records are collapsed (not visible). However, the Vehicle and the Part Type that the application record is referenced to will still be visible (as shown below). Clicking the arrow again will expand and display the collapsed application record's detail.




10. **Column Headers:**

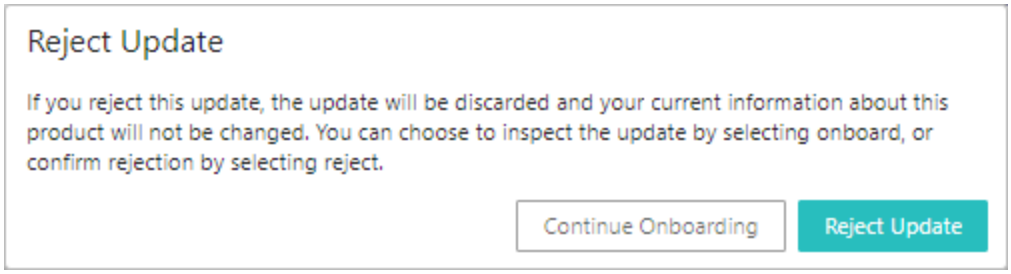


**Note:** Some column header label can be edited as needed by accessing the Onboarding Comparison Screen Properties designer screen. For information on editing column headers in the plugin canvas, refer to the **Configuring Onboarding Comparison Screen** topic within this guide.

By default, the following two column headings are hardcoded to be always visible within the Application Mapping plugin canvas:

- **Vehicle:** This column displays the Vehicle name that the Target object's application record will be referencing if onboarded. This column header label is read-only.
- **Part type**: This column displays the name of the part type that the Target object's application record will be referencing if onboarded. This column header label is read-only.

11. **Comparison table:** This table displays the similarities / differences between the Target object's current application record and the Target object's expected application record. This table allows the user to get a clear picture of which application records are to be Onboarded. For instance, if a user needs to review how the Target object's application records could be changed during the Onboarding process, the expected change in the Target object's application record can be viewed in the table.
12. **Exit Onboarding button:** This button allows the user to exit the Onboarding process without executing / saving any changes.
13. **Reject Update button:** This button requires the configuration of a business action in the Rejected Onboarding Business Action parameter, which is available within the Onboarding Comparison Screen Properties designer screen. The configured business action must be defined with an operation that is executed on the Source object when the user clicks the Reject Update button. Clicking the button displays a message (as shown below) on the screen that provides an option to reject the Onboarding process.




If no business action is configured in the Rejected Onboarding Business Action parameter, then the Reject Update button will not be displayed on the screen.

14. **Total count of application records selected for changes:** Lists the number of application records selected from the Applications stage for execution.
15. **Next button:** On any stage that is not the Confirm stage, a button labeled 'Next' will display. Clicking the 'Next' button will move the user forward one stage in the Onboarding Comparison Screen and will not cause any changes selected to be removed. Users may safely navigate between the steps in the Onboarding Comparison Screen without losing unsaved entries, but users should not exit Onboarding Comparison Screen without first navigating to the Confirm stage and clicking the 'Onboard' button.

### Using Attributes Stage

There are several reasons that Onboarding the attribute values from the Source object to the Target object is a fundamental part of most business models, such as Onboarding attribute values from an object belonging to the Supplier hierarchy to an object belonging to your own data model hierarchy. Data Onboarding can directly copy the attribute values from the Source to a Target object, or apply attribute transformation(s) before updating the Target object attribute.

Comparing the Source and Target attribute values by manually switching between the Source and the Target object is a tedious task. The Attributes stage of the Onboarding Comparison Screen helps the user preview the Source and Target object's attribute values in a single screen. The Attributes stage of the Onboarding Comparison Screen is a single-screen view used to preview the changes affecting the Target object attribute for a selected Source object in the context of the functions defined in the Mapper Configuration. The screen is also designed to showcase the similarities / differences between the Target object's current and potentially updated attribute values.

The Attributes stage in the Onboarding Comparison Screen is designed to display the following details for comparison:

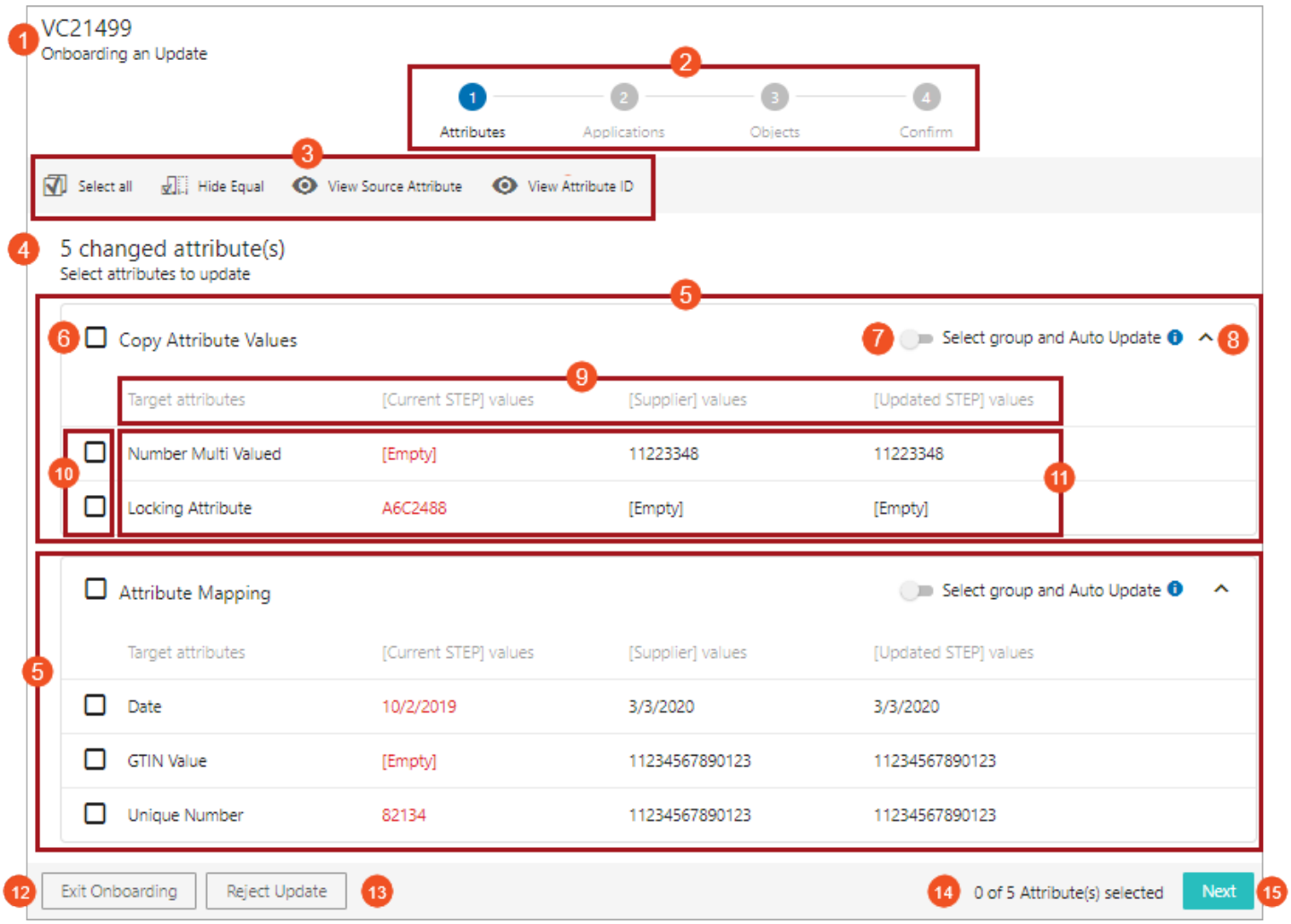
- The current Source and Target object's attribute values, the expected Target object's attribute value, and their similarities and/or differences in the context of the **Attribute Mapping** plugins that are available in the Mapper Configuration.
- The current Source and Target object's attribute values, the expected Target object's attribute value, and their similarities and/or differences in the context of the **Automatic Mapping** plugins that are available in the Mapper Configuration.
- The current Source and Target object's attribute values, the expected Target object's attribute value, and their similarities and/or differences in the context of the 'On Target' type of **Object to Object Mapping** plugins that are available in the Mapper Configuration.

**Important:** The Object to Object Mapping plugin consists of three types, 'Reference' type, 'Child' type, and 'On Target' type. Only the 'On Target' type of Object to Object Mapping plugin is considered for comparison in the Attributes stage. The 'Reference' and 'Child' types of Object to Object Mapping plugins are designed to be compared in the Objects stage / Applications stage.

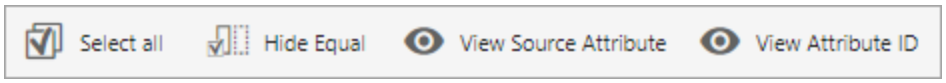
The Attributes stage of the Onboarding Comparison Screen considers only the active Mapping plugins along with their active Mapper rows that are available within the configured Mapper Configuration. Each plugin that is displayed in the Attributes stage has a dedicated canvas called 'Individual plugin canvas.' The options available within each canvas allow users to individually decide what rule of the Mapper Configuration setup entity should be executed for the selected Source object, which affects the related Target object attributes.

This section addresses features and functionalities specific to the Attributes stage of the Onboarding Comparison Screen. For information on the Onboarding Comparison Screen's general functionalities, refer to the **Using Onboarding Comparison Screen Interface** topic within this guide.

The Attributes stage of the Onboarding Comparison Screen is comprised of the following elements:




1. **Source object name:** Displays the name of the Source object.
2. **Progress tracker:** The titles of all the stages that make up the Onboarding are displayed beneath their stage number. The stage's number will be highlighted blue when selected. Users may navigate between the stages by clicking on either the stage number or stage labels. The highlighted stage shows the current stage of the Onboarding process in the Onboarding Comparison Screen. In the screenshot above, the Onboarding Comparison Screen displays the Attributes stage.
3. **Comparison table toolbar:**

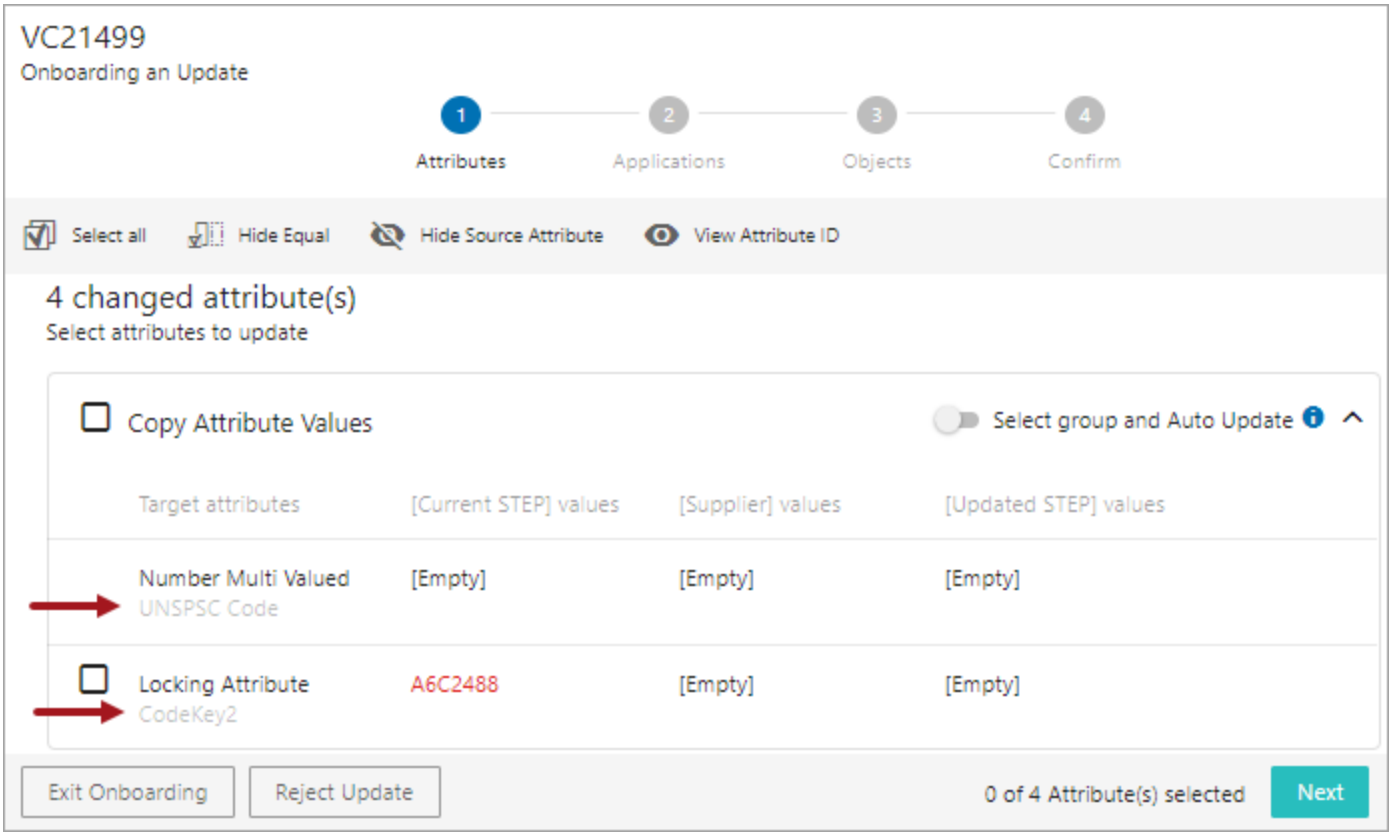



Contains action buttons related to the Attributes stage of the Onboarding Comparison Screen. Available action buttons are hardcoded to always display in the toolbar and cannot be configured in any way.

- **Select all / Clear all:** The 'Clear all' button displays when one or more rows are selected using the 'Mapping plugin / Mapper 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 'Mapping plugin / Mapper

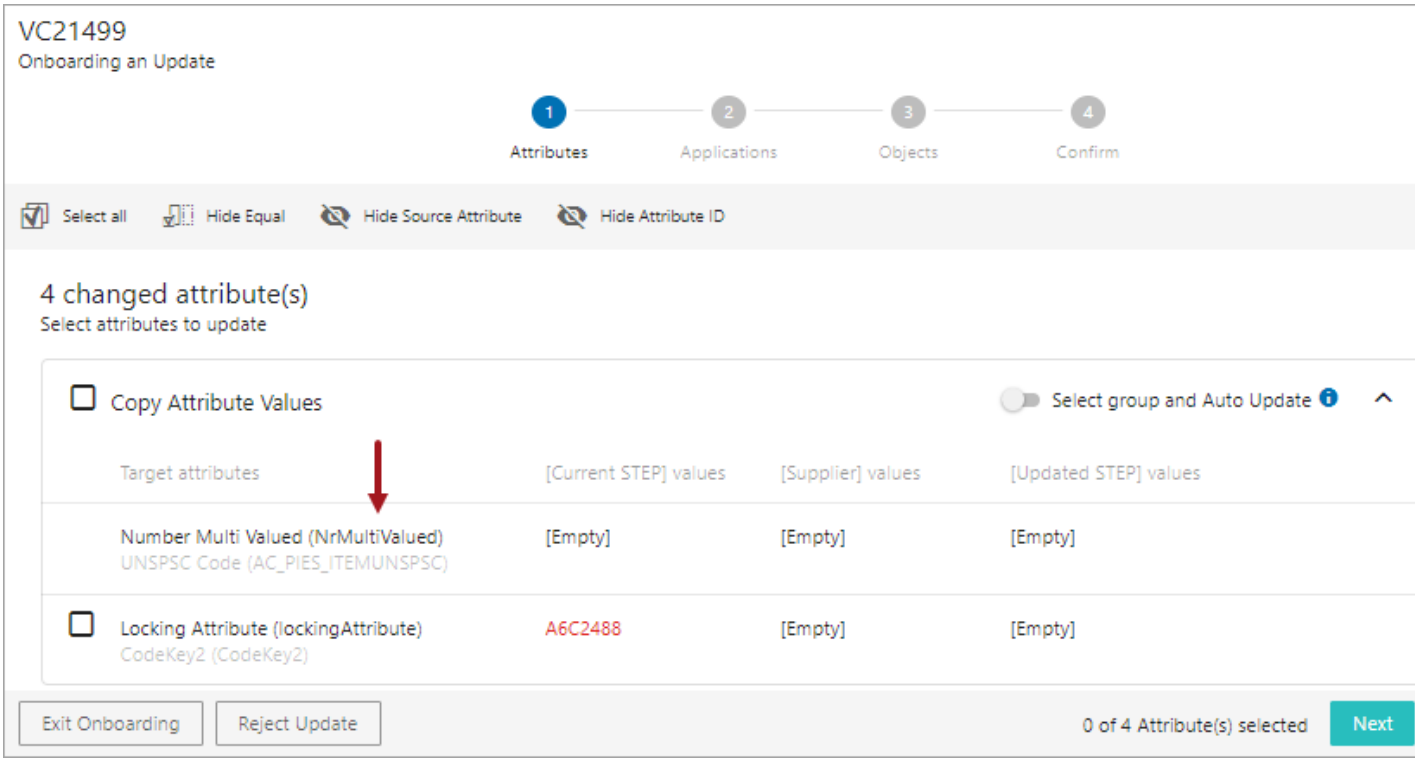
row selection boxes.' Additionally, clicking the 'Clear all' button will not clear the values in the comparison table, but instead disables the selection of 'Mapping plugin / Mapper row selection boxes.' Clicking the 'Select all' button enables all the 'Mapping plugin / Mapper row selection boxes.'

- **Hide Equal / Show Equal:** This action helps determine the rows with identical information between the current Target-related object and the expected Target-related object. The Comparison table is always defaulted to display these rows. However, clicking on the 'Hide Equal' button will hide such rows that have the same values in the current Target object and the expected change in the Target object. When the table displays with these rows compressed, the 'Show Equal' button will display within the toolbar instead of the 'Hide Equal' button.
- **View Source Attribute / Hide Source Attribute:** By default, the Comparison table does not display the Source attribute name. Clicking the 'View Source Attribute' button will make the Source attribute name be displayed in gray color below their respective Target attribute name (as shown below). Meanwhile, the 'Hide Source Attribute' button will compress the Source attribute name displayed below their respective Target attribute name. However, the Source attribute value will always remain displayed in the '[Supplier] values' column irrespective of the Source attribute name's display / suppression. When the table displays the Source attribute name, the 'Hide Source Attribute' button will display within the toolbar instead of the 'View Source Attribute' button.




- **View Attribute ID / Hide Attribute ID:** By default, the Comparison table does not display the Source / Target attribute ID. Clicking the ‘View Attribute ID’ button will make the attribute ID surrounded by parentheses to be indicated next to the Source / Target attribute name. The following

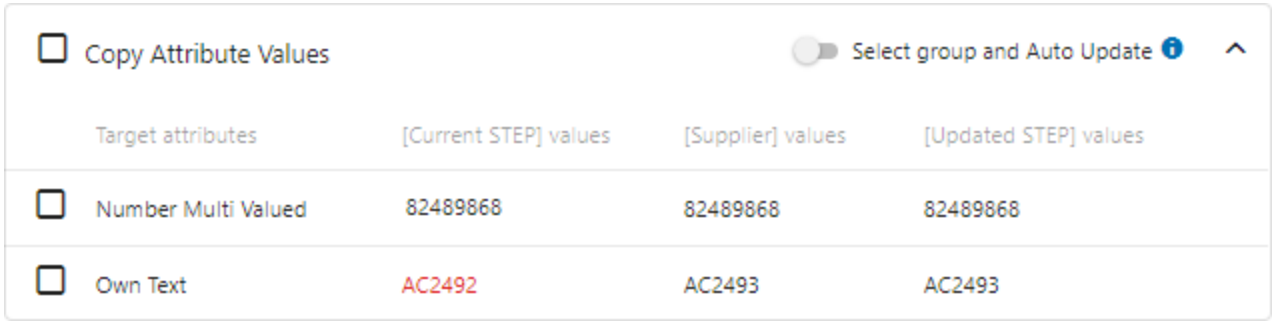
screenshot shows an example (highlighted by the red arrow) of the attribute ID displayed for both the Source object and the Target object.




4. **Number of changed attributes:**

Lists the total number of Target object attributes that are eligible to be updated.

5. **Individual plugin canvas:** Each applicable plugin available in the Mapper Configuration will have a dedicated section in the Attributes stage (the image above which displays the numbered elements contains two individual plugin canvases). Each section is called an 'Individual plugin canvas.' The respective Mapping plugin name identifies each canvas. Each canvas includes active Mapper rows belonging to that Mapping plugin. In the example below, a canvas is identified by the Attribute Mapping plugin name titled 'Copy Attribute Values.'




In the example above, the current value of the Target attribute 'Own Text' will be replaced by the value 'AC2493,' therefore, it is highlighted with red color in the table. Because there are no changes to attribute 'Number Multi Valued,' it is not highlighted in the table.

Each plugin canvas consists of the following:

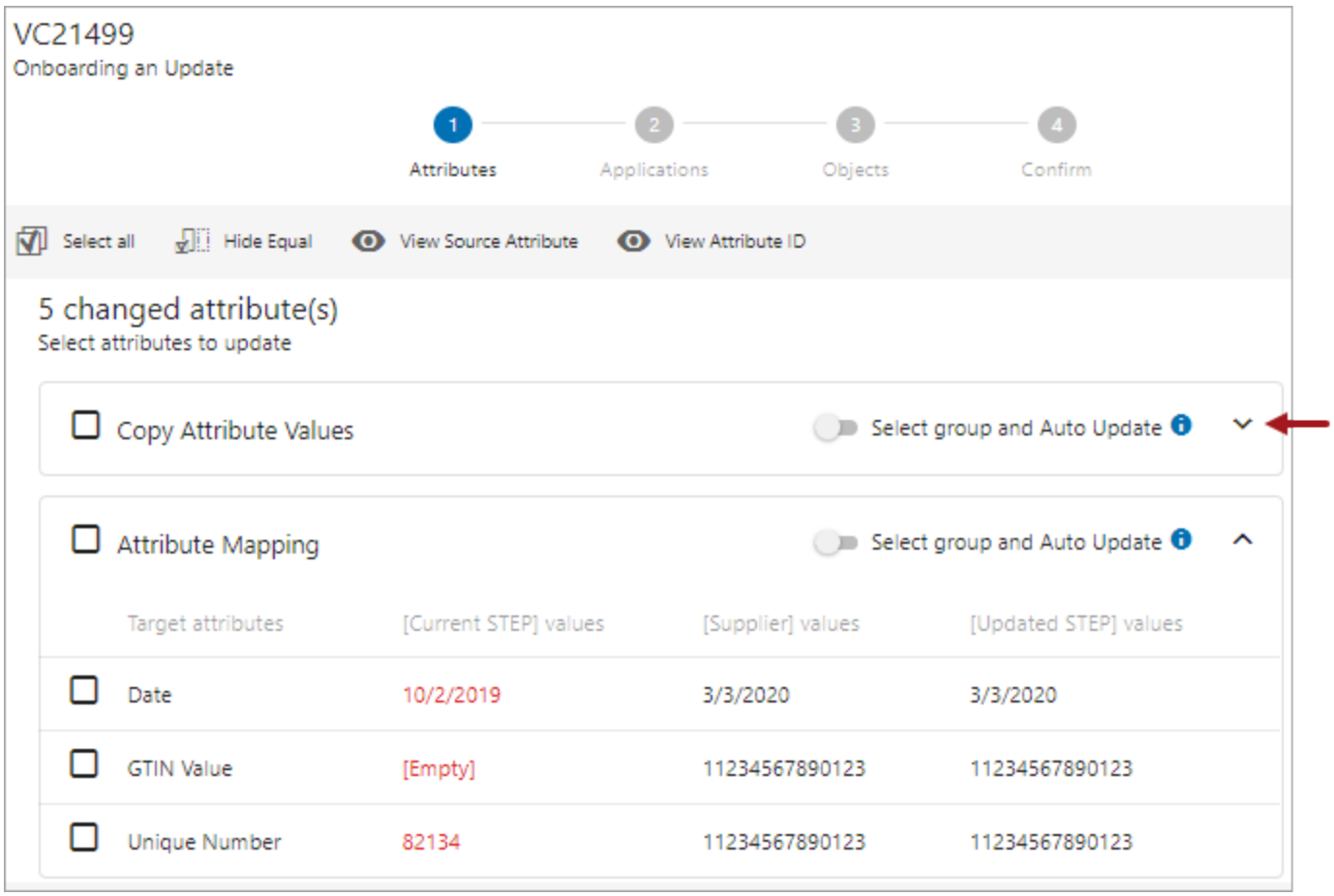
- Plugin selection box
- Auto Update switch
- Expand / collapse arrow
- Mapper row selection boxes

All the options mentioned above are explained in detail below.

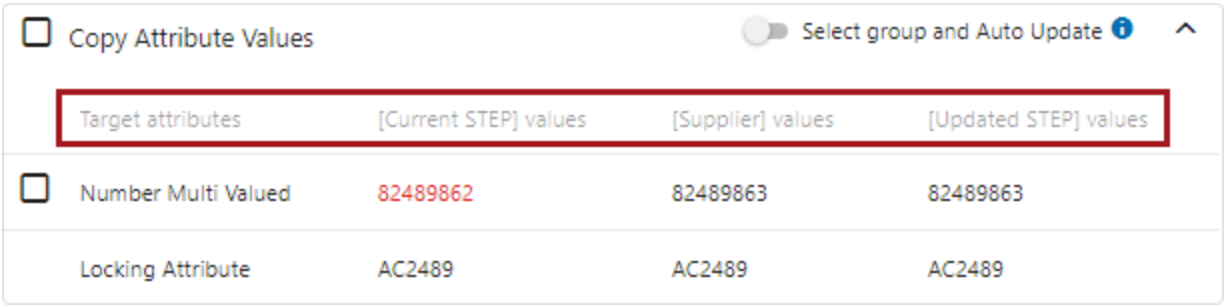
6. **Plugin selection box:** Allows for selection of each applicable plugin displayed in the Attributes stage of the Onboarding Comparison Screen. Selecting the checkbox also selects all the Mapper rows displayed within the respective plugin canvas.
7. **Auto Update switch:** This switch is set to active by default. Leaving this button activated will ensure that all the Mapper rows listed in this canvas will be included for execution in all future Onboarding of the selected Source object. If the user intends to exclude any particular Mapper row / Mapping plugin from being executed for the selected Source object, then the user can turn off the switch and deselect the non-required Mapper row / Mapping plugins. The settings applied with this switch will be held as the system's memory and set as default for all future execution methods for the selected Source object only. When the switch remains turned On, the 'Plugin selection box' (along with their included Mapper rows selection boxes) is selected and are grayed out. That means no individual deselection of the Mapper rows is allowed, and all of the mapper rows listed within the canvas are executed on the selected Source object.

   After turning On / Off the Auto Update switch in any Mapping canvas, the user should click the **Onboard** button (available in the lower right corner of the Confirm stage) to save changes made to the Auto Update switch in the Onboarding Comparison Screen. Users may safely navigate between the stages in the Onboarding Comparison Screen without losing unsaved selections, but users should not exit the Onboarding Comparison Screen without first navigating to the final screen and clicking the Onboard button.

8. **Expand / collapse arrow:** When the arrow is clicked, the plugin canvas where the Mapper rows are listed is collapsed (not visible). However, the name of the Mapping plugin and the Auto Update switch will still be visible in the canvas (as shown below). Clicking the arrow again will expand and display the collapsed Mapper rows.




9. **Column Headers:**



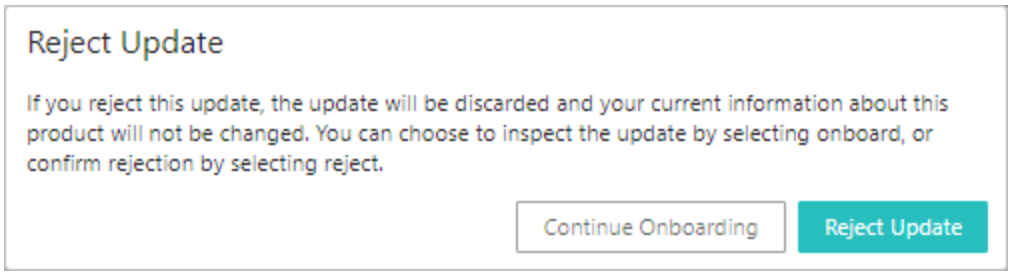

**Note:** Some column header label can be edited as needed by accessing the Onboarding Comparison Screen Properties designer screen. For information on editing column headers in the plugin canvas, refer to the **Configuring Onboarding Comparison Screen** topic within this guide.

By default, the following four column headings are hardcoded to always be visible within the plugin canvas:

- **Target attributes:** This column displays the name of the Target object attributes that are defined as the Target attributes in each Mapper rows within the Mapping plugin. This column header label is read-only.

- **Target object current values**: By default, this column header is labeled '[Current STEP] values.' This column displays the current value(s) populated in the Target object attribute. This column header can be edited as needed.
- **Source object current values:** By default, this column header is labeled '[Supplier] values.' This column displays the current value(s) populated in the Source object attribute. This column header can be edited as needed.
- **Target object new values:** By default, this column header is labeled '[Updated STEP] values.' This column displays the value that gets populated in the Target object attribute after the Mapper Configuration is executed. This column header can be edited as needed.

10. **Mapper row selection boxes:** Allows for selecting each Mapper rows displayed within each plugin canvas. To select or deselect Mapper rows individually, select a Mapper row by clicking the checkbox to the left of the Target attribute's name. To select multiple rows, either click on additional rows or use the 'Select all' action. When a row is selected, the Select all icon switches to a Clear all icon. Use this to clear all selections, or click a checkbox again to deselect a particular row.
11. **Comparison table:** This table displays the similarities / differences between the current Target object attribute values and the expected Target object attribute values. This allows the user to get a clear picture in deciding which attribute values are to be Onboarded. For instance, if a user needs to review how an attribute on the Target object could be changed during the Onboarding process, the expected change in the Target attribute value can be viewed in the table.
12. **Exit Onboarding button:** This button allows the user to exit the Onboarding process without executing / saving any changes.
13. **Reject Update button:** This button requires the configuration of a business action in the Rejected Onboarding Business Action parameter, which is available within the Onboarding Comparison Screen Properties designer screen. The configured business action must be defined with an operation that is executed on the Source object when the user clicks the Reject Update button. Clicking the button displays a message (as shown below) on the screen that provides an option to reject the Onboarding process.




If no business action is configured in the Rejected Onboarding Business Action parameter, then the Reject Update button will not be displayed on the screen.

14. **Total count of attributes selected for changes:** Lists the number of Mapper rows selected from the Attributes stage for execution.
15. **Next button:** On any stage that is not the Confirm stage, a button labeled 'Next' will display. Clicking the 'Next' button will move the user forward one stage in the Onboarding Comparison Screen and will not cause any changes selected to be removed. Users may safely navigate between the steps in the Onboarding Comparison Screen without losing unsaved entries, but users should not exit Onboarding Comparison Screen without first navigating to the Confirm stage and clicking the 'Onboard' button.

### Using Objects Stage

The Objects stage in the Onboarding Comparison Screen is used as a one-screen view to preview the changes affecting the Target related object for a selected Source object in the context of the functions defined in the Object to Object Mapping plugins.

**Important:** The Object to Object Mapping plugin can be of three types: 'Reference' type, 'Child' type, and 'On Target' type. Only the 'Reference' and 'Child' types of Object to Object Mapping plugins are considered for comparison in the Objects stage. The 'On Target' type of Object to Object Mapping plugin is designed to be compared in the Attributes stage. Therefore, the effecting changes displayed in the Objects stage will be either for the child object or referenced object of the actual Target object, and they are not the Target object itself.

**Note:** This topic uses two different words, 'Target object' and 'Target related object' as and when required. While the Target object means the object referenced to the Source object, the Target related object is considered either the child of the Target object or as a reference object of the Target object. The Objects stage displays only the comparison chart for Target related objects and not the Target object.

When retrieving information (ID, Name, attribute values, or references) present on the Source object or from its related object (Source STEP Path) to the Target related object, there could be minimal data related to the Source copied on the Target related object. Or some advanced process demands the use of transformations, Lookup tables, and other complex operations. In this process, it might become necessary for the user to view and assess the individual changes affecting the Target related object before the actual onboarding of information happens.

**Note:** Though the Object to Object Mapping plugin supports onboarding information from a single Source object to multiple Target objects, the Onboarding Comparison Screen only supports viewing and comparing the Source object with a single Target object. Trying to view a Source object with multiple Target objects in the Onboarding Comparison Screen will not load the screen and displays an error.

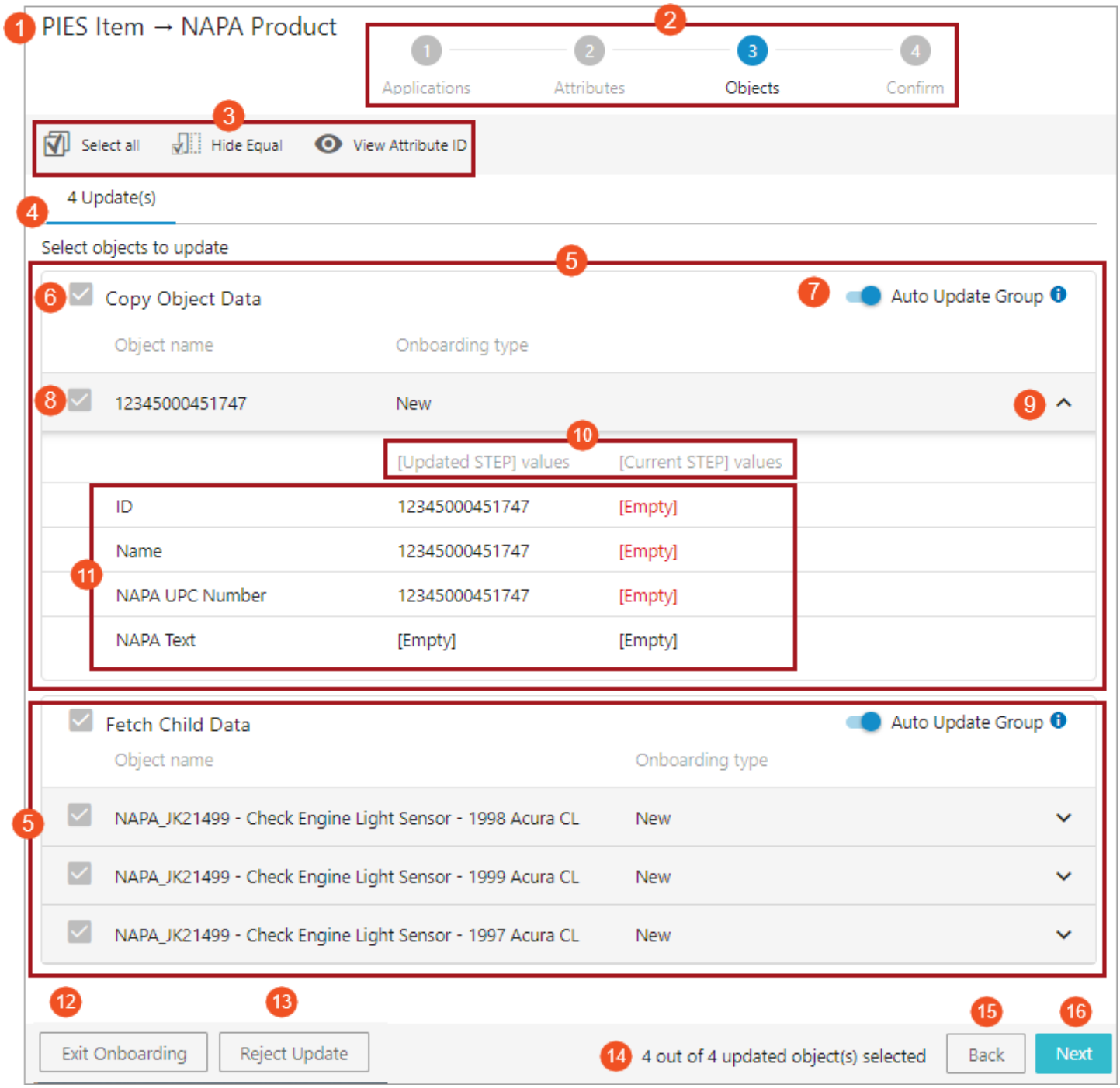
The Objects stage in the Onboarding Comparison Screen displays a comparison chart of current and expected values on the Target related object's ID, Name, and attributes. The changes affecting the Target-related object are displayed and allow / disallow progress for Onboarding as desired. The comparison chart is produced based on the logic defined on the Mapper rows of the Object to Object Mapping plugin for the configured Mapper Configuration setup entity. Only the active Object to Object Mapping plugin(s) and their active Mapper rows are considered to display the comparison chart.

The 'Child' and 'Reference' type of Object to Object Mapping plugin always checks for any existing Target-related object present in the system. If there is a Target-related object with the same ID, the Object to Object Mapping plugin will consider updating that existing Target related object (marked as Update). If no similar Target-related object is present in the system, then a new object will be created based on the logic defined in the Mapper rows of the Object to Object Mapping plugin. Whether it is for creating or updating the Target related object, the Onboarding Comparison Screen evaluates the scenarios and marks correctly as New / Update for comparison.

Each Object to Object Mapping plugin will have a dedicated canvas called 'Objects Mapping plugin canvas.' The checkbox available within each canvas allows users to individually decide which Mapping plugin of the Mapper Configuration setup entity should be executed for the selected Source object that affects the Target related object's ID, Name, and attribute values.

This section addresses features and functionalities that are only specific to the Objects stage of the Onboarding Comparison Screen. For information on the general functionalities of the Onboarding Comparison Screen, refer to the **Using Onboarding Comparison Screen Interface** topic within this guide.

The Objects stage of the Onboarding Comparison Screen is composed of the following:

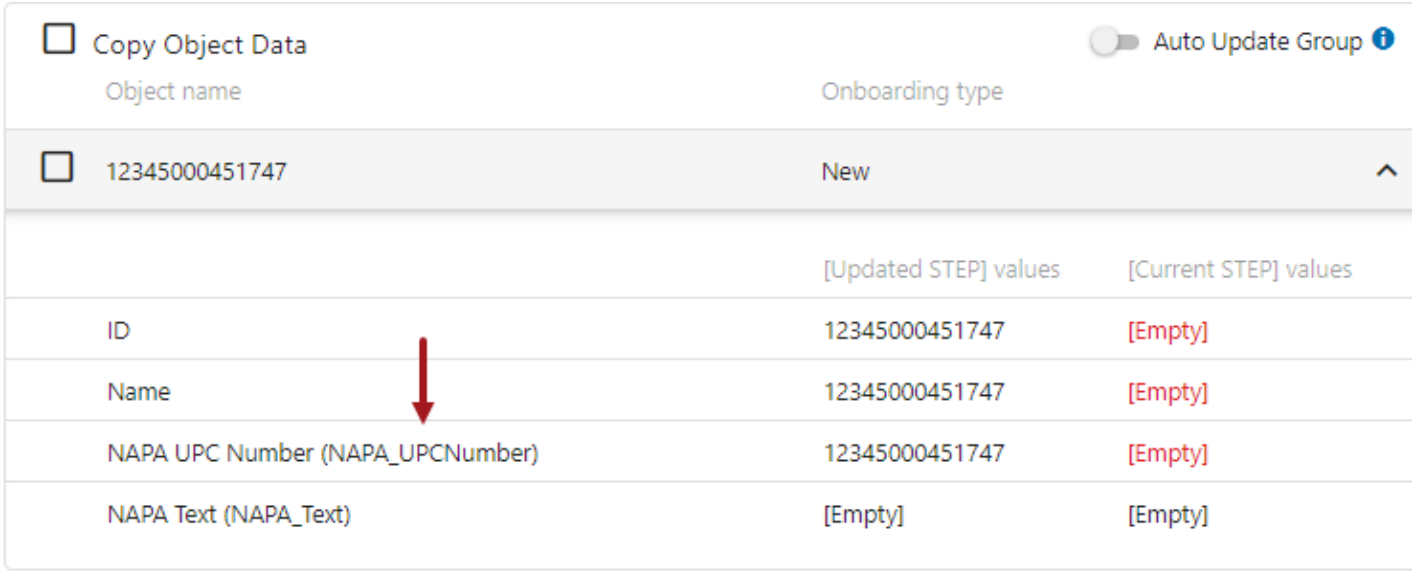



1. **Mapper Configuration name:** Displays the name of the Mapper Configuration executed for the selected Source object.
2. **Progress Tracker:** An interactive tracker that displays all of the current Onboarding process with the selected stage highlighted in blue. Users may navigate between the stages by clicking on either the stage number or stage label. The highlighted stage shows the current stage of the Onboarding process in the Onboarding Comparison Screen. In the screenshot above, the Onboarding Comparison Screen displays the Objects stage.

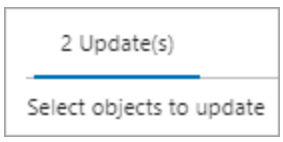
3. **Comparison table toolbar:**

   Contains action buttons related to the Objects stage of the Onboarding Comparison Screen. Available action buttons are hardcoded to always display in the toolbar and cannot be configured in any way.

   - **Select all / Clear all:** The 'Clear all' button displays when one or more Mappings are selected using the 'Mapping plugin selection box.' When no Mappings are selected, the 'Select all' button will display. Clicking the 'Clear all' button will remove the selections within the 'Mapping plugin selection box.' Clicking this button will not clear the comparison table's values but rather disables the selection of 'Mapping plugin selection boxes.' Clicking the 'Select all' button enables all the 'Mapping plugin selection boxes.'
   - **Hide Equal / Show Equal:** This action helps find out the rows that have the current Target object information and the change expected Target object information of equal values. The Comparison table is always defaulted to display all rows. However, clicking on the 'Hide Equal' button will hide the rows with the same values. When the table displays with these rows compressed, the 'Show Equal' button will display within the toolbar instead of the 'Hide Equal' button.
   - **View Attribute ID / Hide Attribute ID:** By default, the Comparison table does not display the Target related object's attribute ID. Clicking the 'View Attribute ID' button will make the attribute ID surrounded by parentheses to be indicated next to the attribute name. The following screenshot shows an example of the attribute ID displayed for the Target object.

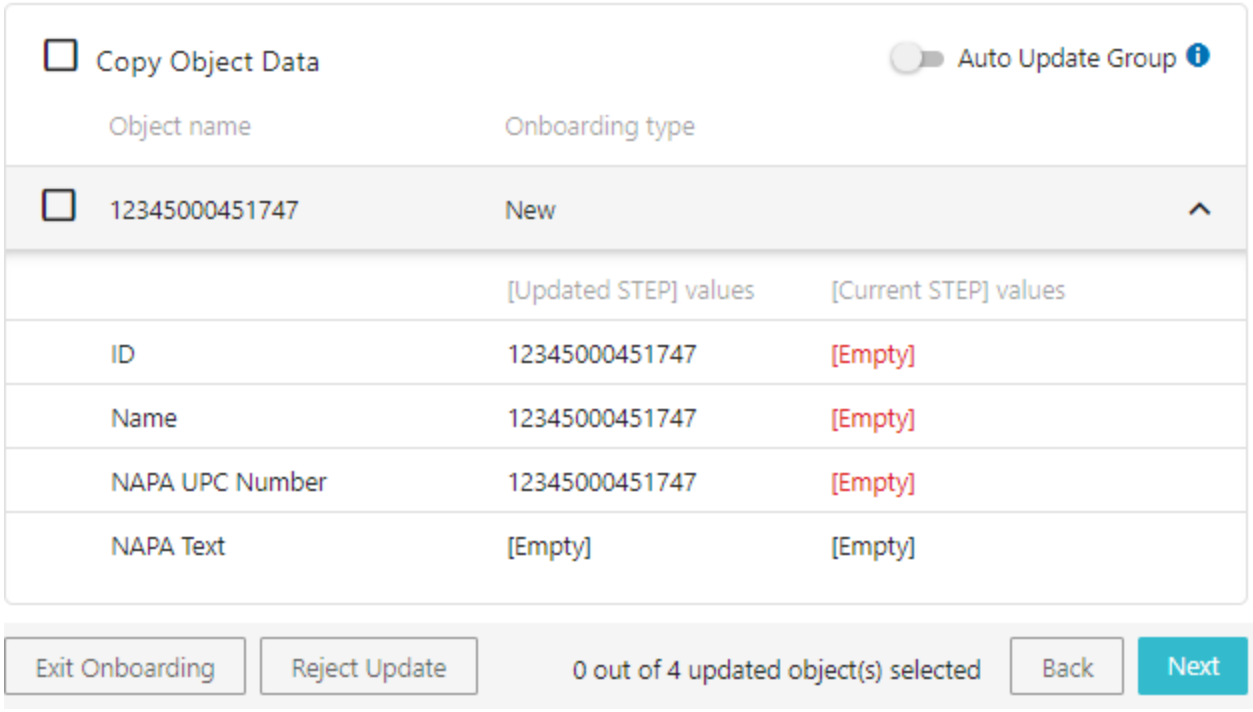



4. **Number of Object changes:**




   Lists the total possible number of Target related objects that could get updated / created.

5. **Objects Mapping plugin canvas:** Each active Object to Object Mapping plugins (of 'Reference' and 'Child' Mapping types) available in the Mapper Configuration setup entity will have a dedicated section in the Objects stage, and each section is called an 'Objects Mapping plugin canvas.' Each Objects Mapping plugin canvas is identified by its respective Object to Object Mapping plugin name. Each canvas displays ID, Name, and attribute values of the Target related object as defined in the active Mapper rows of the Object to Object Mapping plugin. In the example below, a canvas is identified by the Object to Object Mapping plugin name called 'Copy Object Data.'




In the example above, the current value of the Target related object's attribute 'NAPA UPC Number' will be replaced by the value '12345000451747,' consequently, it is highlighted with red color in the table. The value of the Target attribute 'NAPA Text' will not be changed, and thus it is not highlighted in the table.

Each Objects Mapping plugin canvas consists of the following:

- Plugin selection box
- Auto Update switch
- Expand / collapse arrow

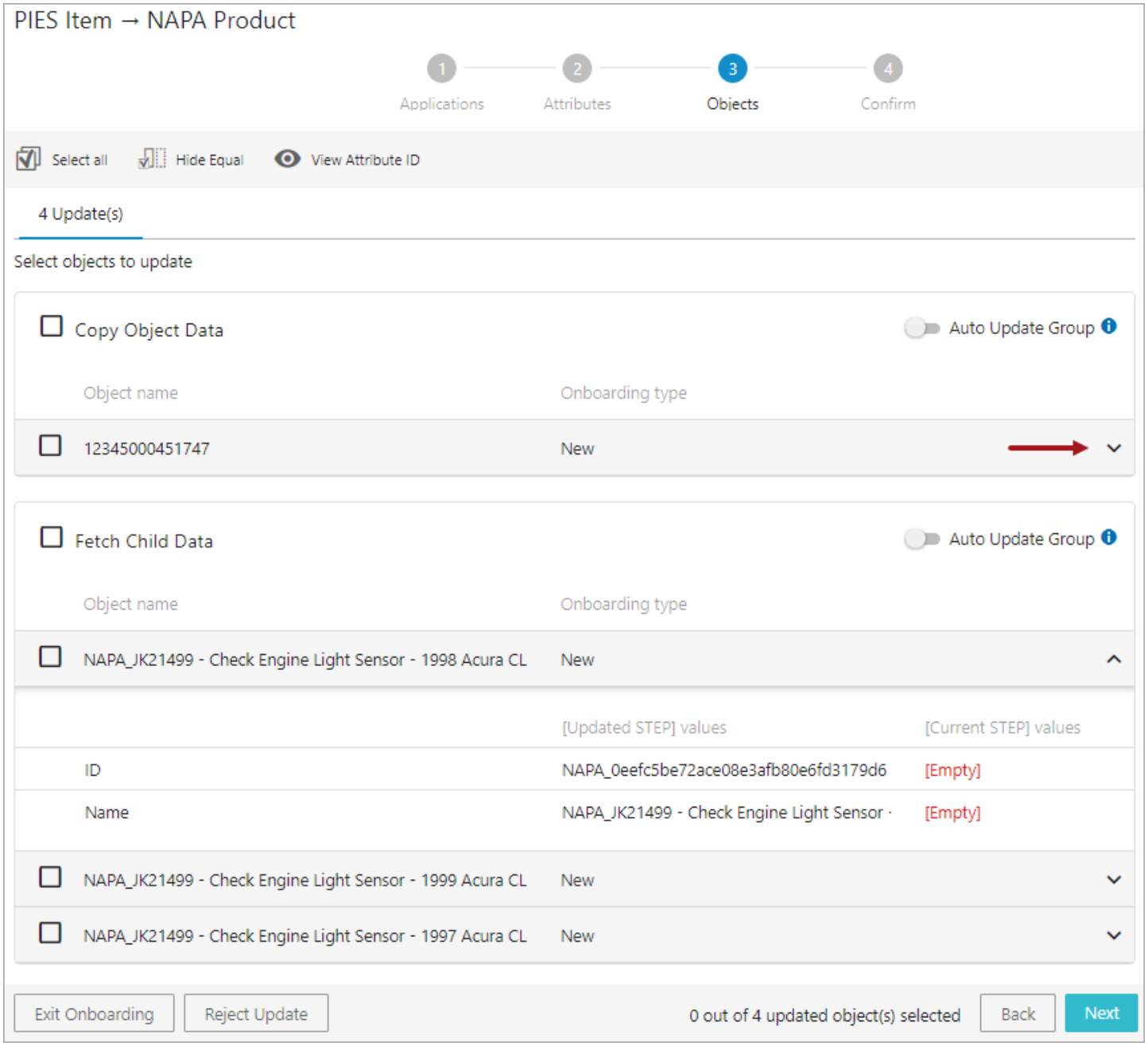
All the options mentioned above are explained in detail in the later section of this topic.

6. **Plugin selection box:** Allows for selecting each Object to Object Mapping plugin displayed in the Objects stage of the Onboarding Comparison Screen. Selecting the checkbox determines that all the functionalities defined (as Mapper rows) within the respective Object to Object Mapping plugin canvas are selected for execution.
7. **Auto Update switch:** This button is set to active by default. Leaving this button activated will ensure that the Mapping plugin listed in the canvas will be included for execution in all future Onboarding for the selected Source object. If the user intends to exclude any particular Mapping plugin from being executed for the

selected Source object, then the user can turn off the switch and deselect the non-required Mapping plugins / Target related objects. The settings applied with this switch will be held as the system's memory and set as default for all future execution methods for the selected Source object only. When the button remains turned On, the 'Plugin selection box' (along with their included Target related object's selection boxes) is selected and are grayed out. That means no individual deselection of the Target object is allowed, and the whole of the Target objects listed within the canvas is executed on the selected Source object.

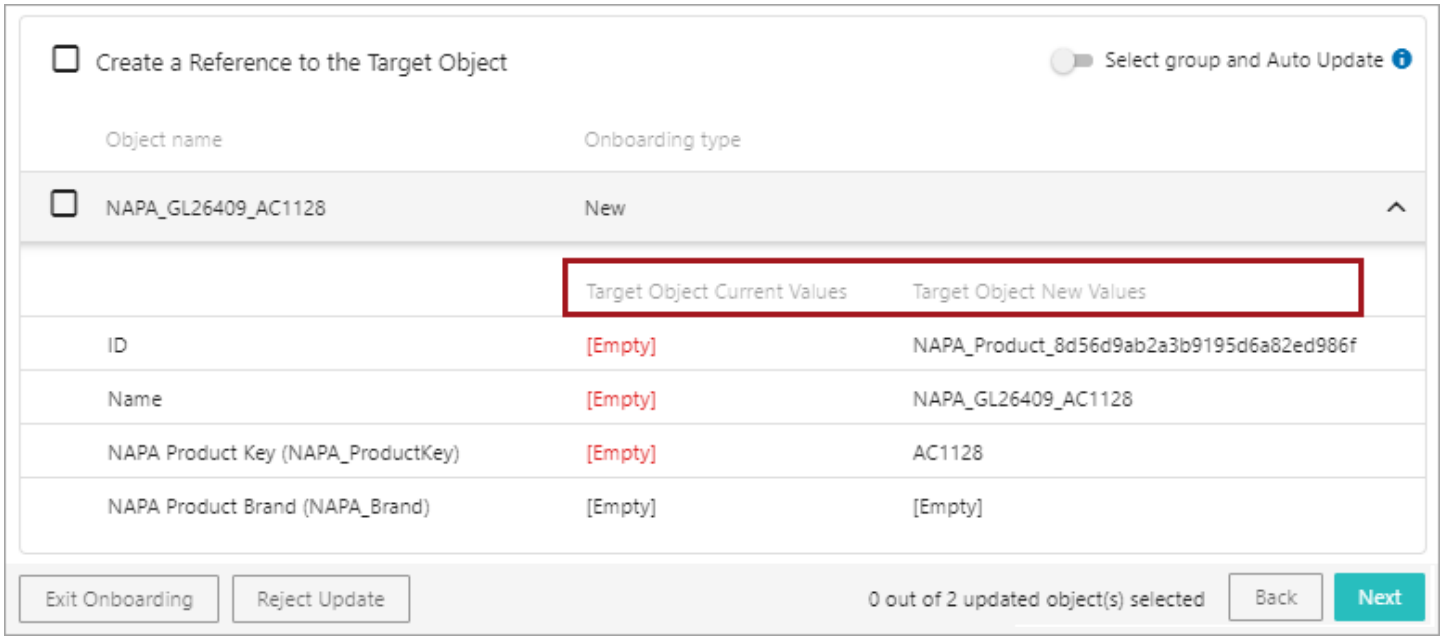
After turning On / Off the Auto Update switch in any Mapping canvas, the user should click the **Onboard** button (available in the lower right corner of the Confirm stage) to save changes made to the Auto Update switch in the Onboarding Comparison Screen. Users may safely navigate between the stages in the Onboarding Comparison Screen without losing unsaved selections, but users should not exit the Onboarding Comparison Screen without first navigating to the final screen and clicking the Onboard button.

8. **Target related object name:** The name of the Target related object (either child or reference object) that gets created / updated is displayed along with a checkbox in this row. If a new Target related object is getting created, then the term 'New' is displayed next to the name. If an existing Target related object is being updated, then the term 'Update' is displayed next to the name of the object.
9. **Expand / collapse arrow:** When the arrow is clicked, the Mapping plugin canvas where the Target related object changes are listed is collapsed (not showing). However, the name of the Mapping plugin and the Auto Update switch will still be visible in the canvas (as shown below). Clicking the arrow again will expand the display of the collapsed Target related object changes.




10. **Column Headers:**

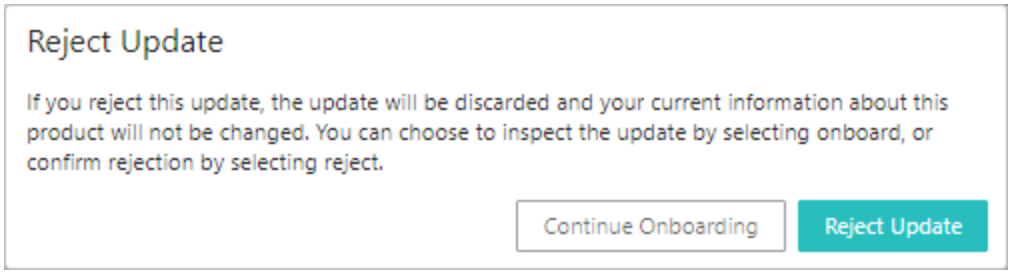
**Note:** Each column header label can be edited as needed by accessing the Onboarding Comparison Screen Properties designer screen. For information on editing column headers in the Application Mapping plugin canvas, refer to the **Configuring Onboarding Comparison Screen** topic within this guide.




By default, the following two column headings are hardcoded to be always visible within the Objects Mapping plugin canvas:

- **Target Object Current Values:** This column displays the current value(s) populated in the Target related object's ID, name, and attributes. This column header can be edited as needed.
- **Target Object New Values:** This column displays the value that gets populated in the Target related object's ID, Name, or attribute after the corresponding Mapper Configuration is executed. This column header can be edited as needed.

11. **Comparison table:** This table displays the similarities / differences between the current Target related object's ID, Name, and attribute values against the expected Target related object's ID, Name, and attribute values. This way, the user gets a clear picture of which Target-related object is to be Onboarded.
12. **Exit Onboarding button:** This button allows the user to exit the Onboarding process without executing / saving any changes.
13. **Reject Update button:** This button requires the configuration of business action in the Rejected Onboarding Business Action parameter that is available within the Onboarding Comparison Screen Properties designer screen. The configured business action must be defined with an operation that is to be executed on the Source object when the user clicks the Reject Update button. Clicking the button displays a message (as shown below) on the screen that asks for the confirmation to reject the Onboarding process.



If no business action is configured in the Rejected Onboarding Business Action parameter, then the Reject Update button will not be displayed on the screen.

14. **Total count of objects selected for changes:** Lists the number of Target related objects selected from the Objects stage for execution.
15. **Back button:** On any stage that is not the first stage, a button labeled 'Back' will display. Clicking the 'Back' button will move the user to the previous stage in the Onboarding Comparison Screen and will not cause any changes selected to be removed.
16. **Next button:** On any stage that is not the Confirm stage, a button labeled 'Next' will display. Clicking the 'Next' button will move the user forward one stage in the Onboarding Comparison Screen, but will not cause any changes selected to be removed. Users may safely navigate between the steps in the Onboarding Comparison Screen without losing unsaved entries, but users should not exit Onboarding Comparison Screen without first navigating to the Confirm stage and clicking the 'Onboard' button.

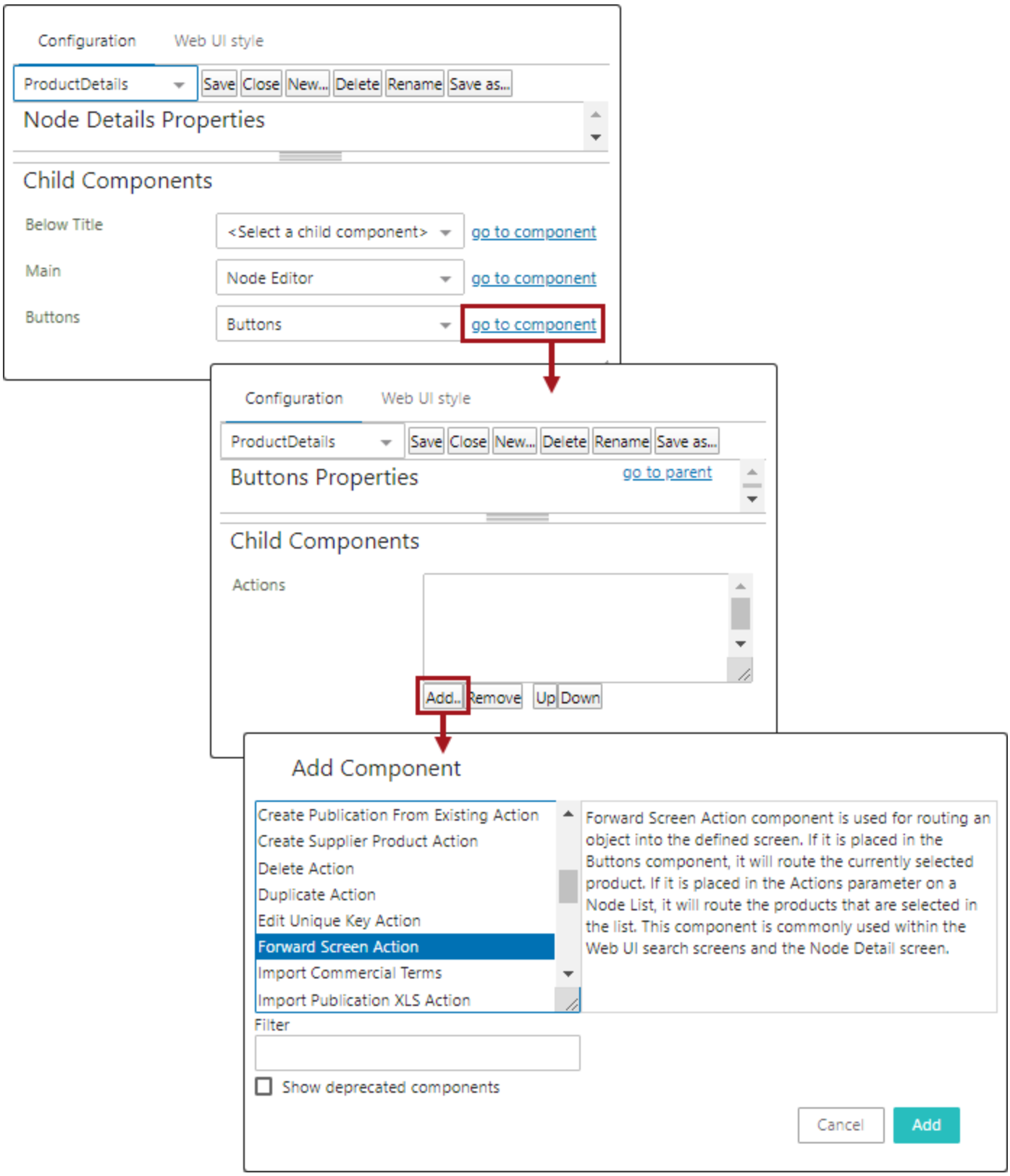
## Forward Screen Action Component

Forward Screen Action component is used for routing an object into a defined screen. If it is placed in the Buttons component, it will route the currently selected product. If it is placed in the Actions parameter on a Node List, it will route the object that is selected in the list. This component is commonly used within Web UI search screens and the Node Detail screen.

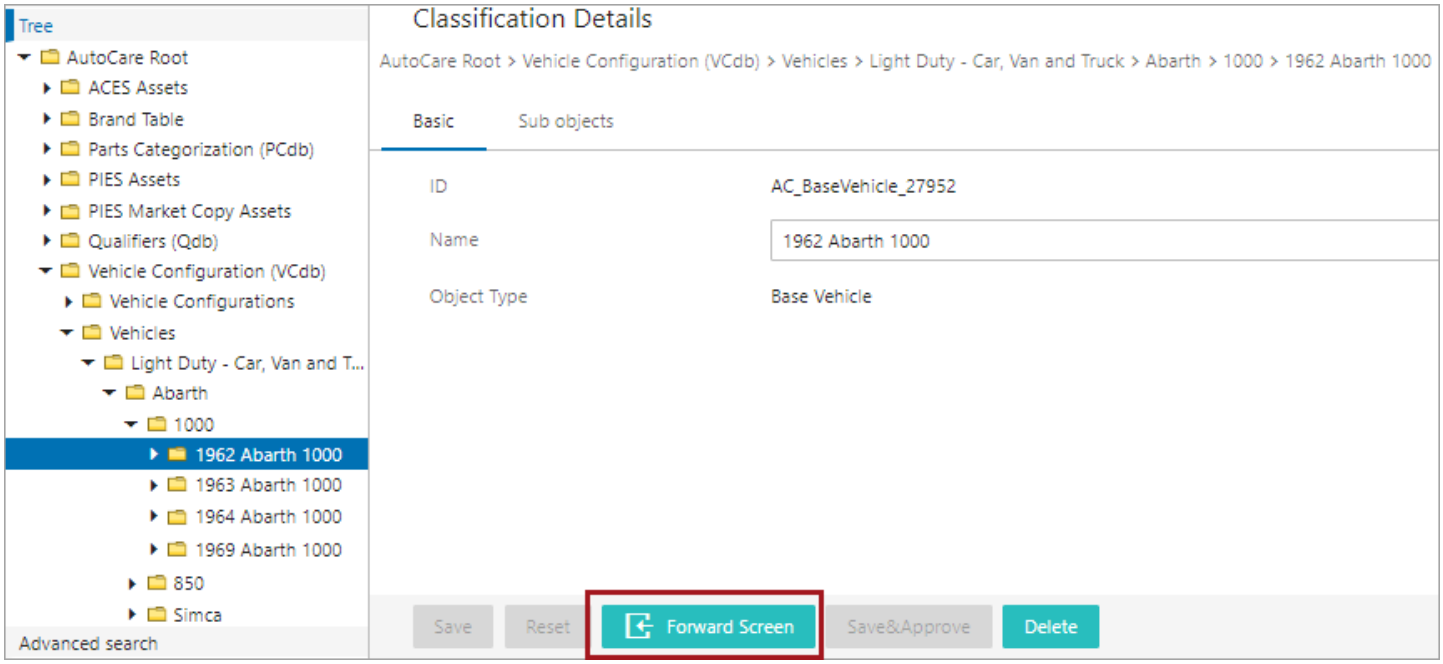
One use case of the Forward Screen Action component within the Automotive solution is to route the selected object to the Onboarding Comparison Screen. There are no explicit limits on how the component can be used, but it was created specifically to route the selected object towards the Onboarding Comparison Screen so that the user can compare the selected Source object with the Target object in the context of the configured Mapper Configuration setup entity.

As defined below, the **Forward Screen Action** button can be added to any Node Details screen as a standard action button (a button accessible below the table) and to any Node List as a toolbar action (a button accessible above the table).

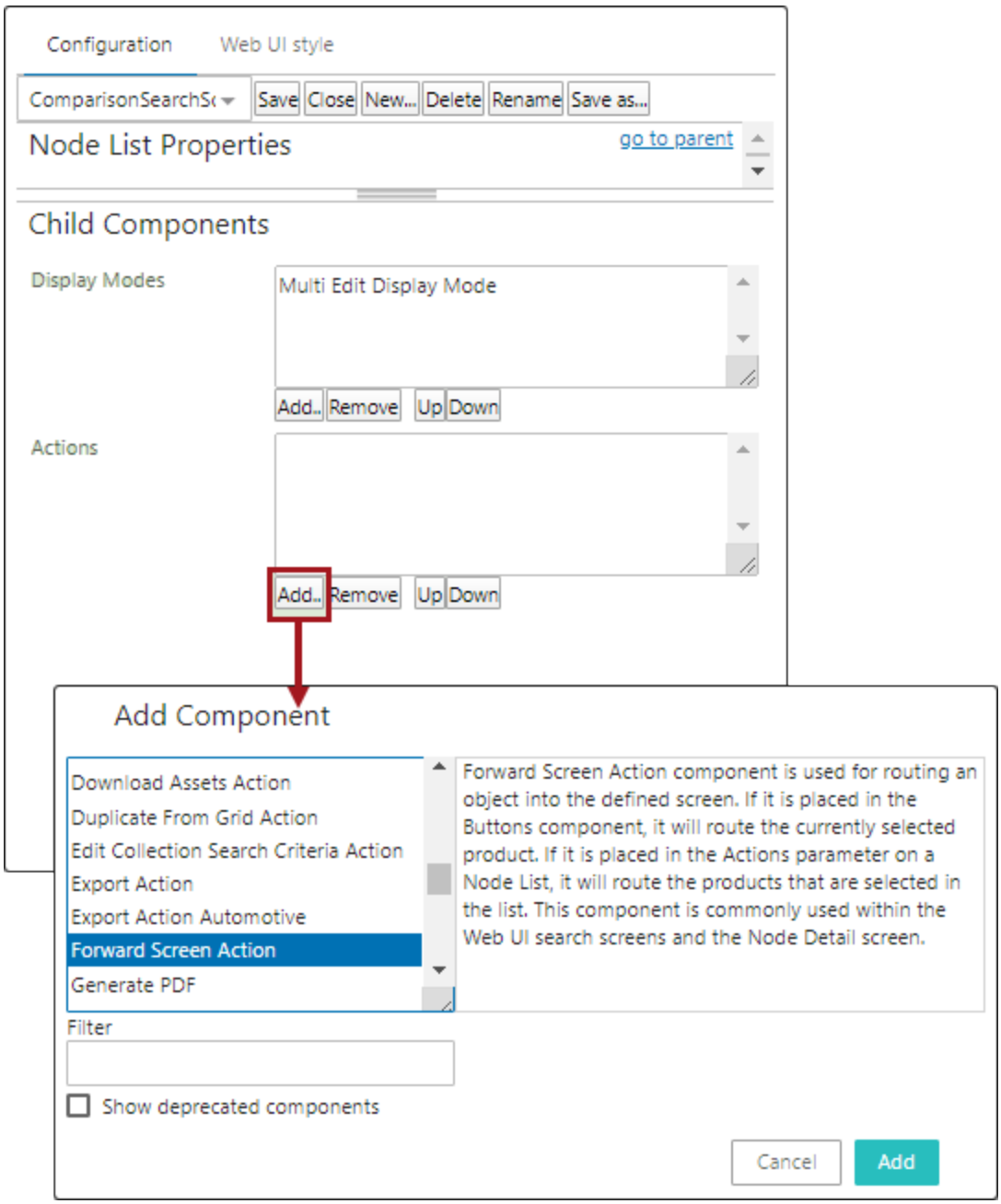
To add the **Forward Screen Action** button to a Node Details screen as a standard action button, navigate to the relevant 'Node Details Properties' screen in the designer. Under 'Child Components,' click the 'go to component' link next to the 'Buttons' parameter. Click 'Add' and then find, select, and add the 'Forward Screen Action' component.


Configuration
Web UI style
ProductDetails
Save
Close
New...
Delete
Rename
Save as...
Node Details Properties
Child Components
Below Title
<Select a child component>
go to component
Main
Node Editor
go to component
Buttons
Buttons
go to component
Configuration
Web UI style
ProductDetails
Save
Close
New...
Delete
Rename
Save as...
Buttons Properties
go to parent
Child Components
Actions
Add...
Remove
Up
Down
Add Component
Create Publication From Existing Action
Create Supplier Product Action
Delete Action
Duplicate Action
Edit Unique Key Action
Forward Screen Action
Import Commercial Terms
Import Publication XLS Action
Forward Screen Action component is used for routing an object into the defined screen. If it is placed in the Buttons component, it will route the currently selected product. If it is placed in the Actions parameter on a Node List, it will route the products that are selected in the list. This component is commonly used within the Web UI search screens and the Node Detail screen.
Filter
Show deprecated components
Cancel
Add

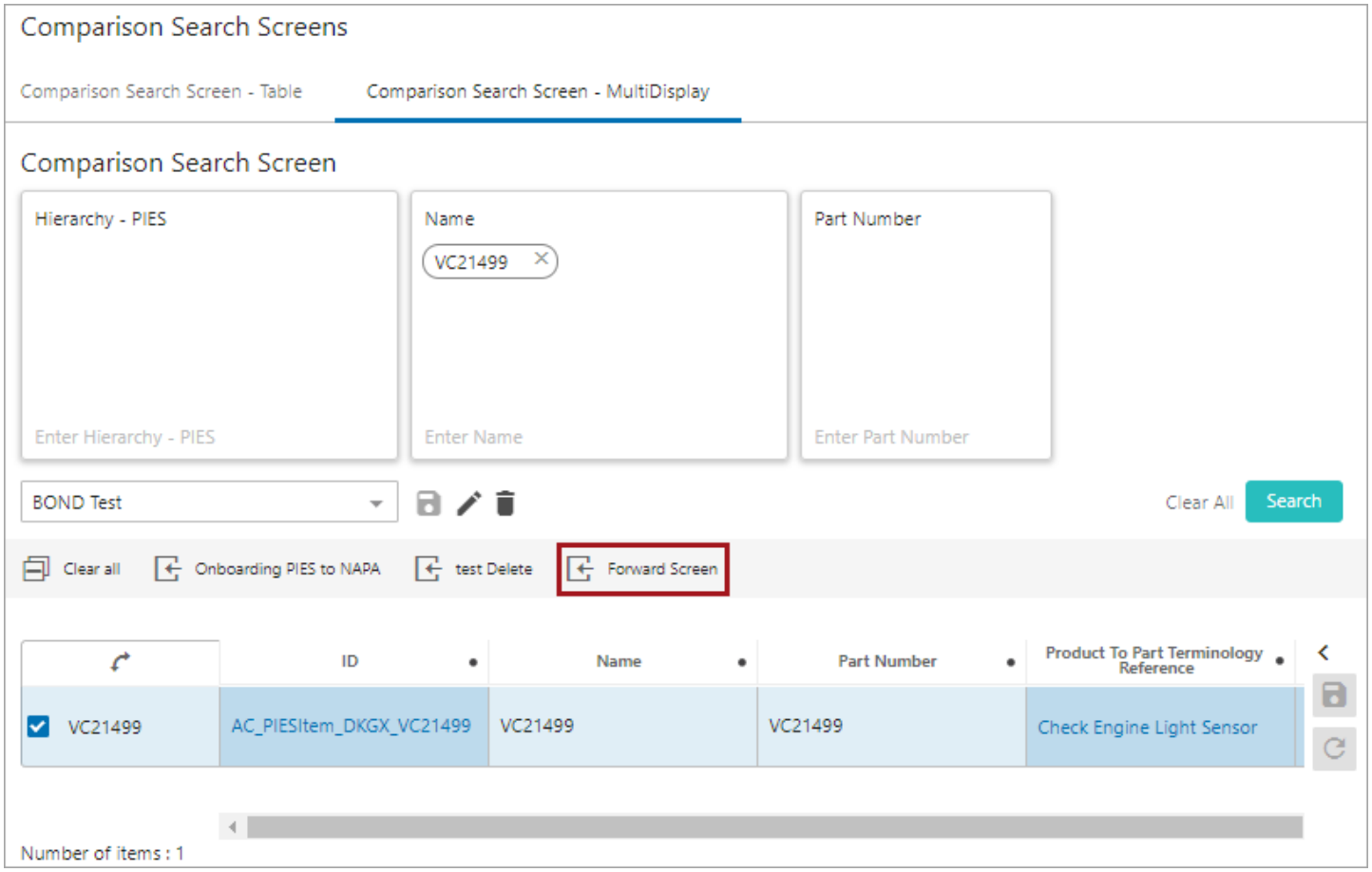
Displayed in the screenshot below is the Forward Screen action as it displays when configured within the Node Details screen.




To add the **Forward Screen Action** button to a Node List screen as a toolbar action button, navigate to the relevant Node List component in the designer. In the 'Child Components' section, click the 'Add' button below the 'Actions' field. In the 'Add Component' screen that displays, find, select, and add the 'Forward Screen Action' component.

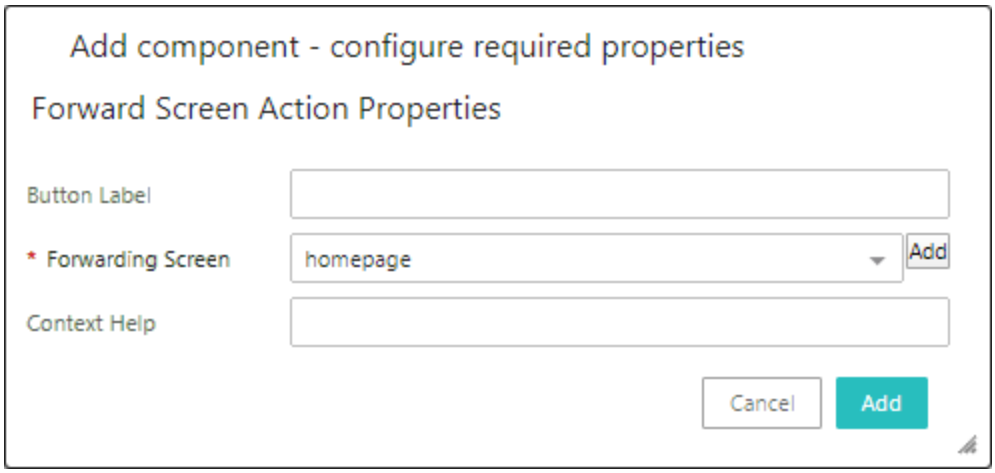



Displayed in the screenshot below is the Forward Screen action as it displays when configured within the Node List of the Search by Card Screen.




Whether configuring the Forward Screen Action component as an action button or a toolbar action button, the options displayed in the designer are the same.

Displayed in the screenshot below is the Forward Screen Action properties screen as it displays when being configured for a Node List screen. The properties screen for this component when being configured for a Node Details screen will have slight differences, which will be outlined in the parameter descriptions located below the screenshot.



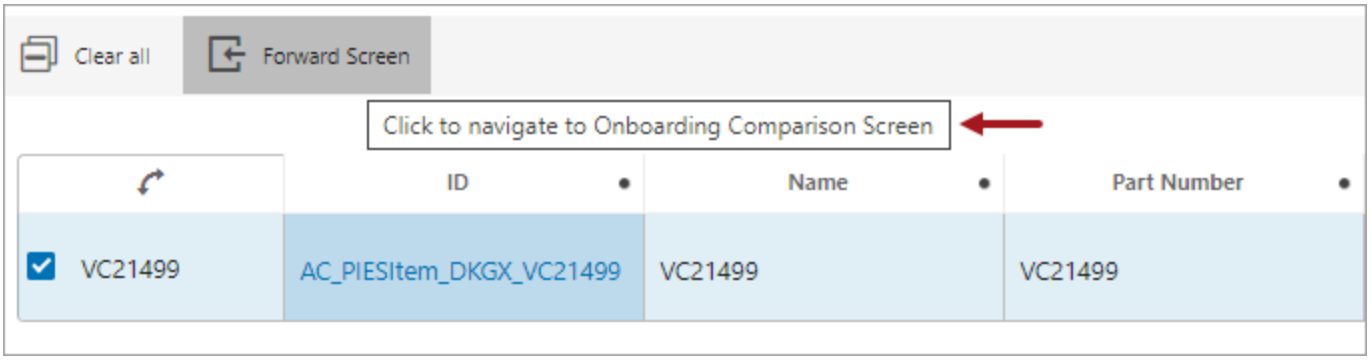
- **Button Label**: When configuring a standard action button, the text entered in this field will display on the button. When configuring a toolbar action button, the text entered in this field will display beside the button. If nothing is added, the Label default will display as 'Forward Screen.'

  This is how the button will display at the Node List toolbar:

- **Forwarding Screen:** Select a specific screen using the dropdown so that users are forwarded to that screen when they click on the action button. It is important that only the right type of screen is configured within this parameter. Configuring the wrong type of screen will not allow the action button to perform as intended and an error will be displayed.
- **Context Help**: Enter the text that will display in a small informational popup message when the cursor is placed over the button.

  For example, if the field is populated with the value 'Click to navigate to Onboarding Comparison Screen,' this is how the help text will display when the cursor is placed over the action button in the Node List toolbar:

### Using Forward Screen Action button in Search Screens

The **Forward Screen Action** button (

) is enabled when a single object is selected in the search results table. If no object or multiple objects are selected, the Forward Screen Action button remains disabled.

1. Perform a search for objects.
2. Select the object that needs to be routed to the specific screen.
3. Click the **Forward Screen Action** button (

) that is available in the toolbar of the screen.

# Exporting Automotive Data

The Automotive solution provides extensive export capabilities. For information on the supported versions, refer to the **Supported Versions and Formats** topic within this guide. For information on the way a user initiates and manages exports, refer to 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, 4.0, 4.1, and 4.2 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**.

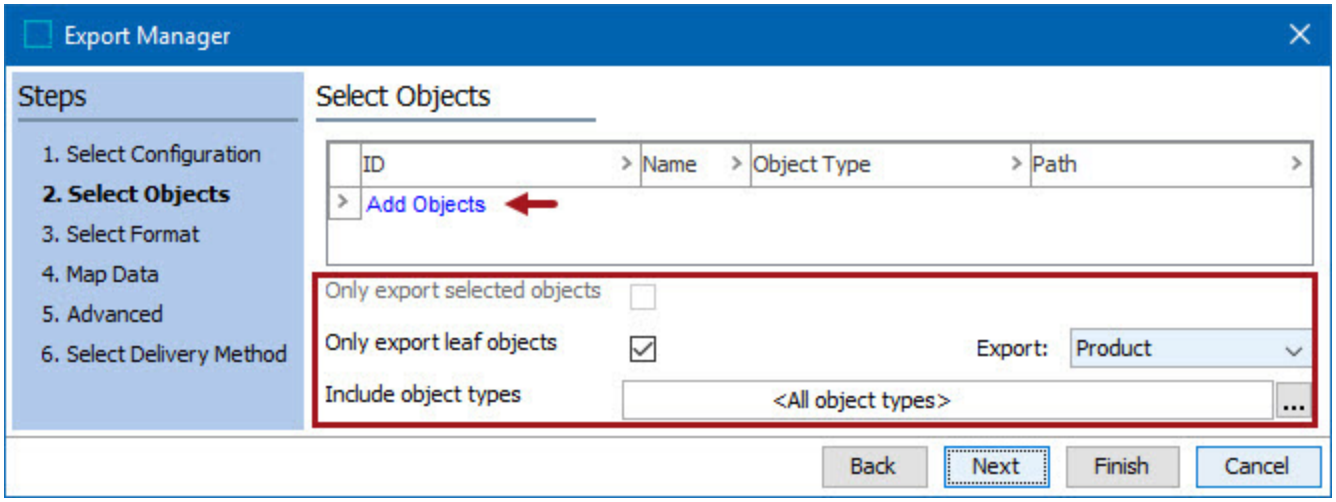
## Using ACES Application Exporter

### 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.

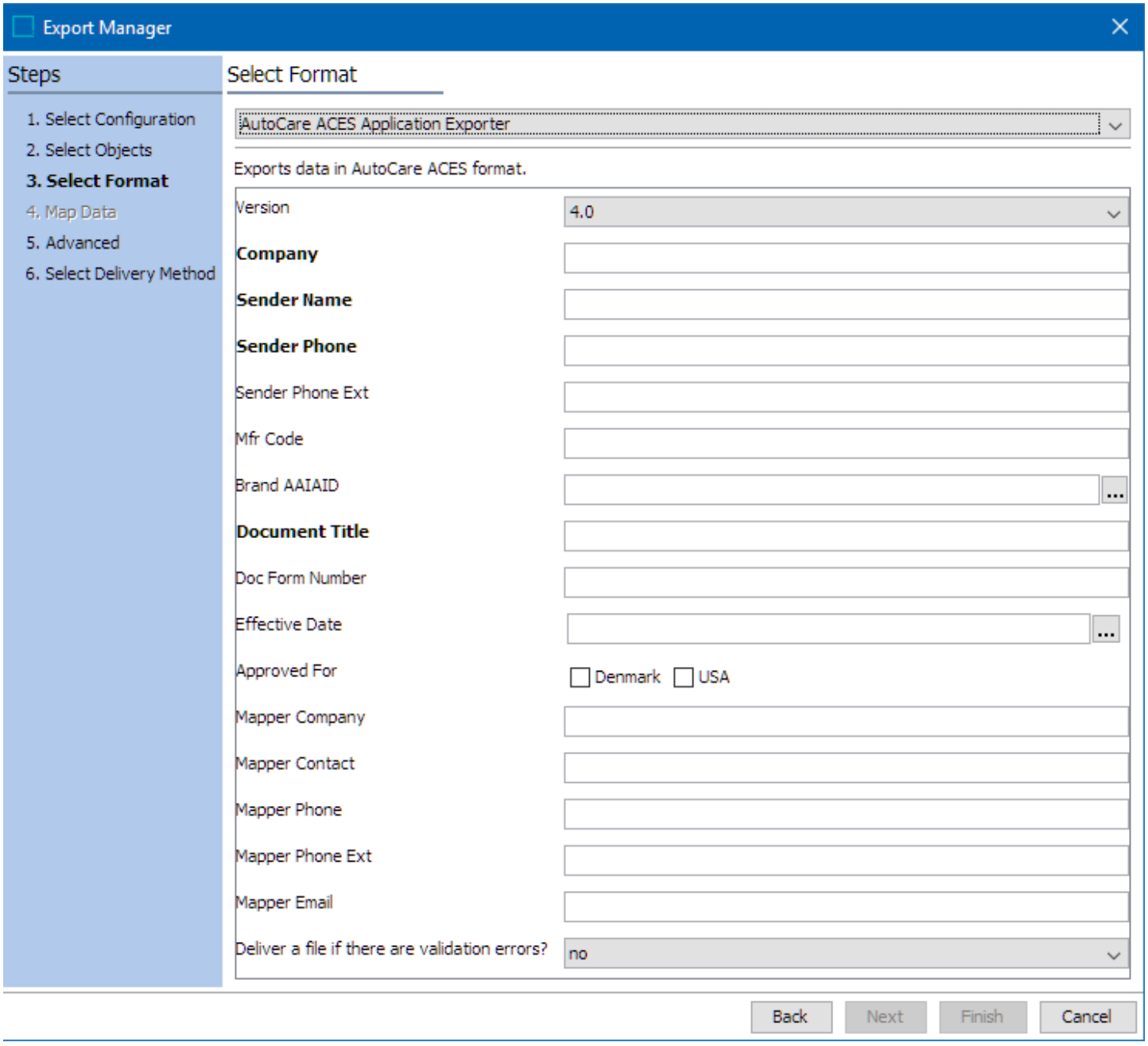



Click the **Add Objects** link to add one or more blue folders in the AutoCare PIES Products hierarchy, or one or more classification folders.

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 4.2, 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.

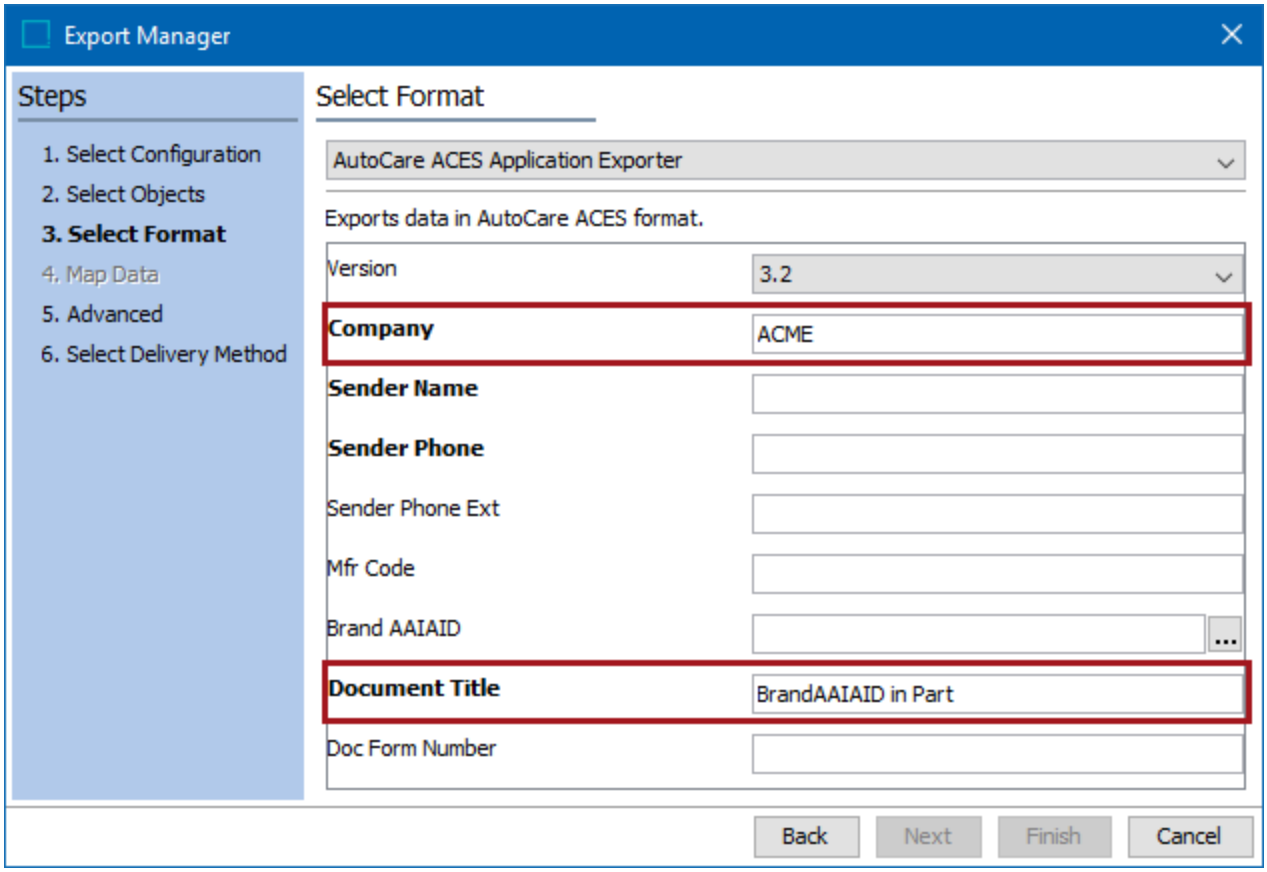
```
<SenderPhone>7701231234</SenderPhone>
```

- **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'




## 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">
```

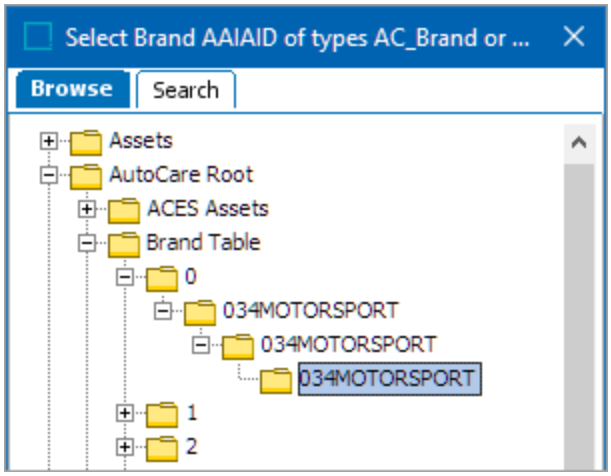
- **Sender Phone Ext**: Allows a sender phone extension to be specified within the free text field. The value will display within the ACES export between the <SenderPhoneExt> tags. For example, if the value ‘6789’ is provided, then the exported ACES file will display as shown below.

```
<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 AAIAID**: Allows a Brand or Sub Brand AAIAID (Automotive Aftermarket Industry Association ID) object to be selected. Click the ellipsis button (...), and the 'Select Brand AAIAID' dialog will display, as shown in the image below.



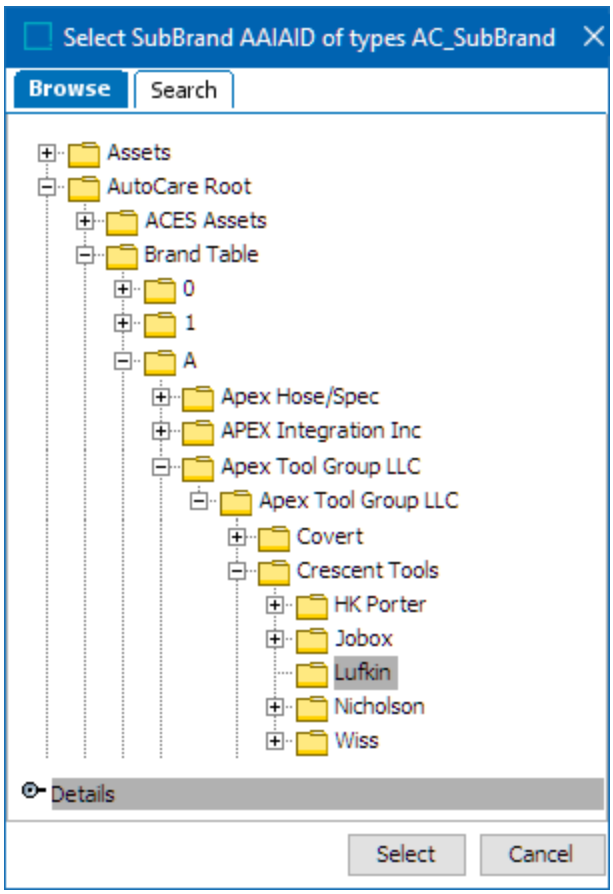

Select the desired Brand, and save and close the dialog. The STEP Name of the selected Brand will display followed by the Brand AAIAID in parentheses. However, only the four character Brand AAIAID will display between the <BrandAAIAID> tag in the export.

For Example, if the Brand object '034MOTORSPORT (ID=GWWQ)' is selected, then the exported ACES file will display as shown below.

```
<BrandAAIAID>GWWQ</BrandAAIAID>
```

**Note:** The <BrandAAIAID> tag embedded within the <Part> tag is not populated from this value. The embedded <BrandAAIAID> 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).

- **SubBrand AAIAID**: Allows a Sub Brand AAIAID (Automotive Aftermarket Industry Association ID) object to be selected. Click the ellipsis button (...), and the 'Select SubBrand AAIAID' dialog will display, as shown in the image below.




Select the desired SubBrand, and save and close the dialog. The STEP Name of the selected SubBrand will display followed by the Brand AAIAID in parentheses. However, only the four character SubBrand AAIAID will display between the <SubBrandAAIAID> tag in the export.

For Example, if the SubBrand object 'Lufkin' (ID=BBLK)' is selected, then the exported ACES file will display as shown below.

```
<SubBrandAAIAID>BBLK</SubBrandAAIAID>
```

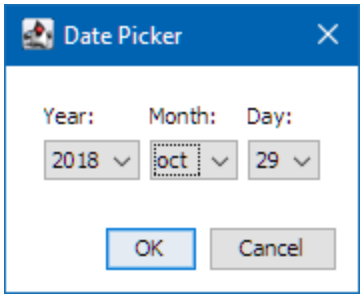
**Note:** The <SubBrandAAIAID> tag embedded within the <Part> tag is not populated from this value. The embedded <SubBrandAAIAID> tag is populated directly from the application record itself (i.e., if the application record had the parameter populated on import, it is populated with the same value on 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**: This parameter will be applicable only for the ACES 3.0, 3.2, 4.0, and 4.1 versions. The parameter gets grayed out when option 4.2 is selected in the Version parameter. This parameter 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, refer to 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>
```

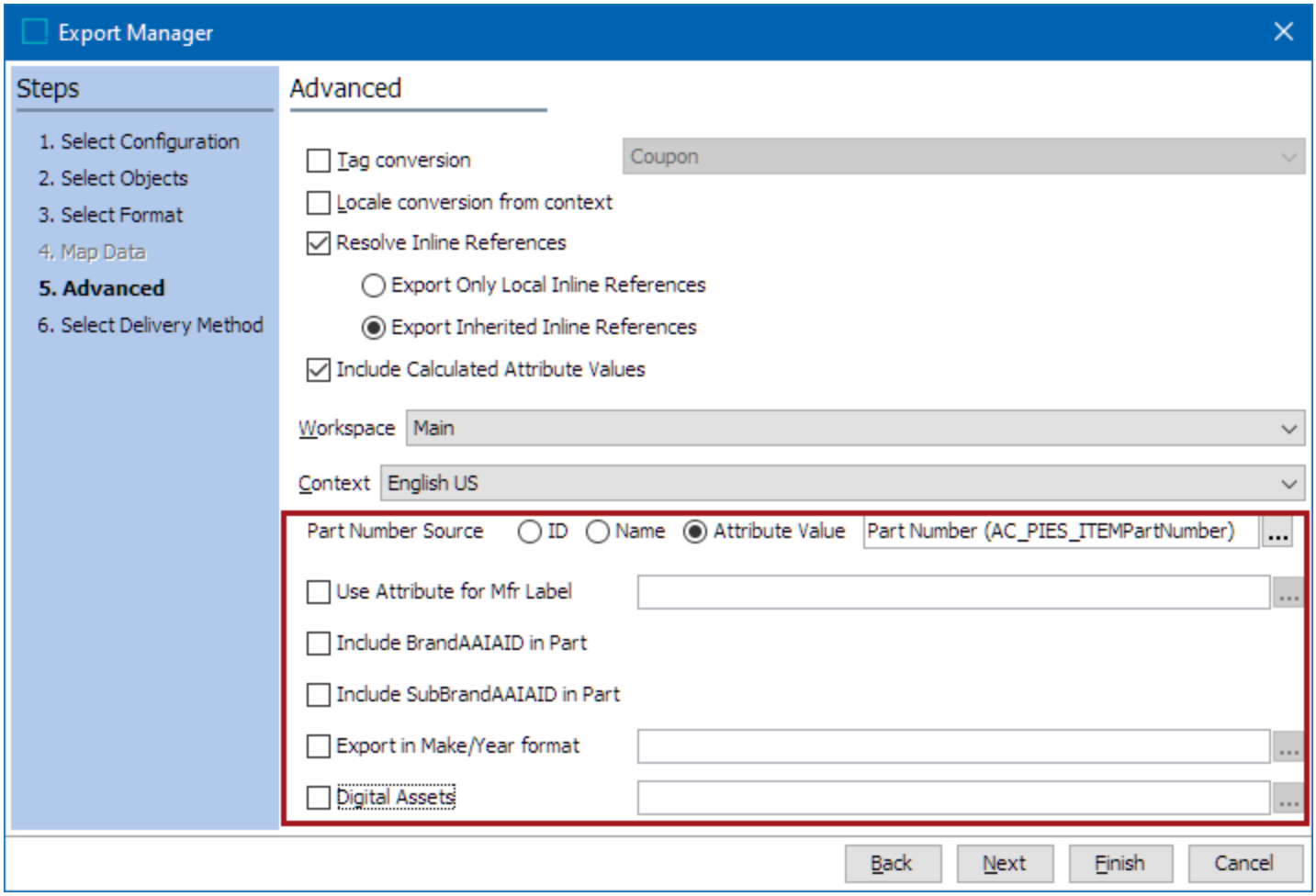
- **Mapper Email**: Allows for the mapper contact's email information to be specified within the free text field. The value will display within the ACES export between the <MapperEmail> tags. For example, if the value 'abcd@acme.com' is provided, then the exported ACES file will display as shown below.

```
<MapperEmail>abcd@acme.com</MapperEmail>
```

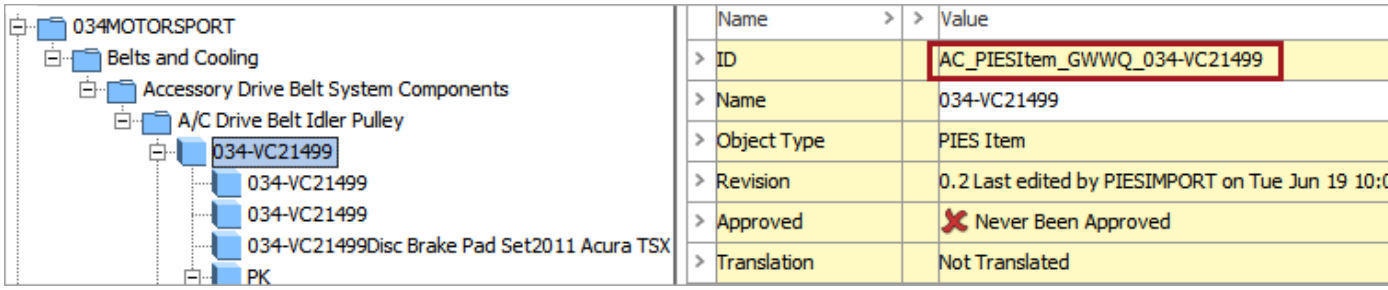
- **Deliver a file if there are validation errors?**: By default, 'no' is selected, and the export file will not be delivered when validation errors occur. When 'yes' is selected, the export file will be delivered even though validation errors occur.

### 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, refer to the **Export Manager - Advanced** topic within **STEP Online Help**.




- **Part Number Source:** The options available within this parameter determine the value that is to be populated within the <Part> tag in the exported ACES file. The Following options are available for the selection:
  - **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.



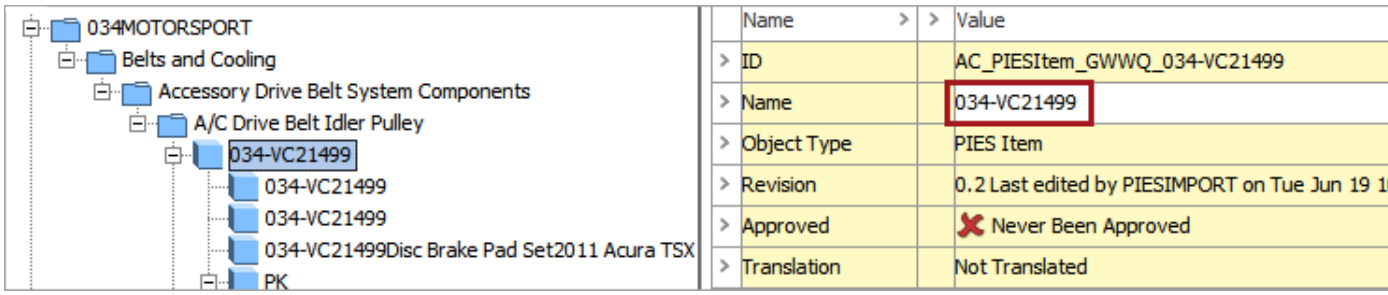
When the 'Include BrandAAIAID in Part' parameter is disabled, then the ACES export will display as shown below.

```
<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.



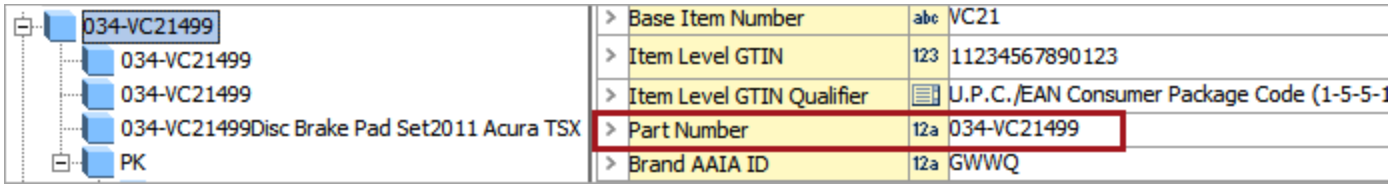

When the 'Include BrandAAIAID in Part' parameter is disabled, then the ACES export will display as shown below.

```
<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>
```

- **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.



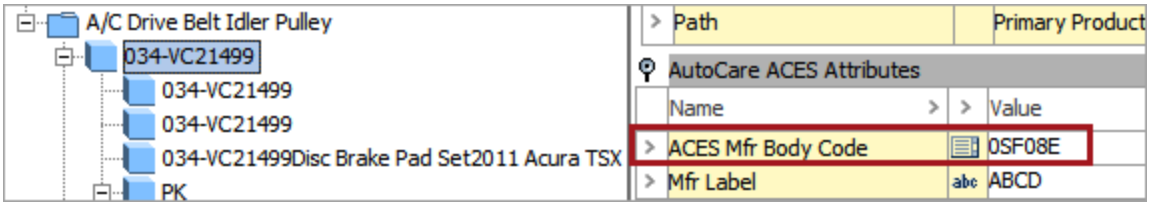
When the 'Include BrandAAIAID in Part' parameter is disabled, then the ACES export will display as shown below.

```
<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.

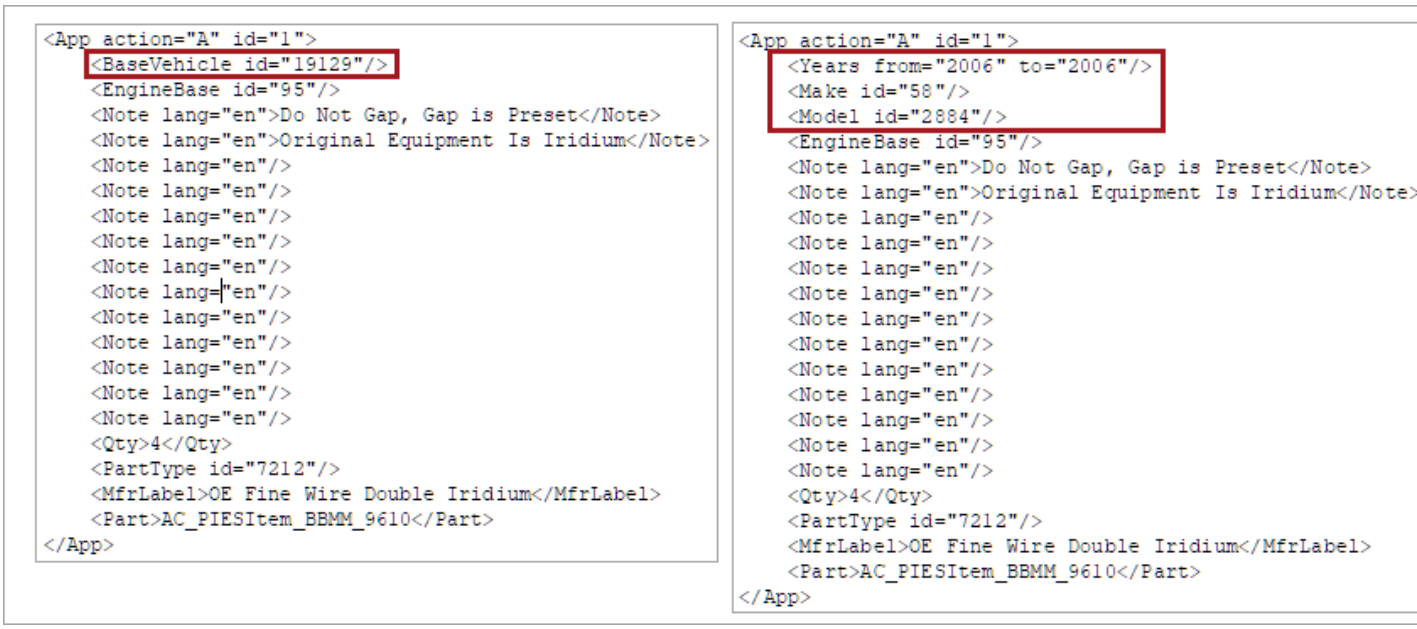



**Note:** The automotive solution does not support the MfrLabel value coming from both the part and the application. The MfrLabel value can only be retrieved from the part or the application.

  - 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.
- **Include SubBrandAAIAID in Part:** By default this parameter is deselected so that the SubBrandAAIAID is excluded from the part number tag. When selected, the Sub Brand AAIAID is inserted between the <Part> tag of the export. Examples are included within the above Part Number Source parameter details.

- **Export in Make/Year format:** By default this parameter is deselected so that the exported file will include the application record with the linked Base Vehicle / Equipment Base object information (as shown in the image on the left below). If the parameter is selected, the exported file will include the application record with the information about related Make, Model and Year (or Manufacturer, Equipment Model and Vehicle Type for Equipment application records) in the exported file (as shown in the image on the right below).



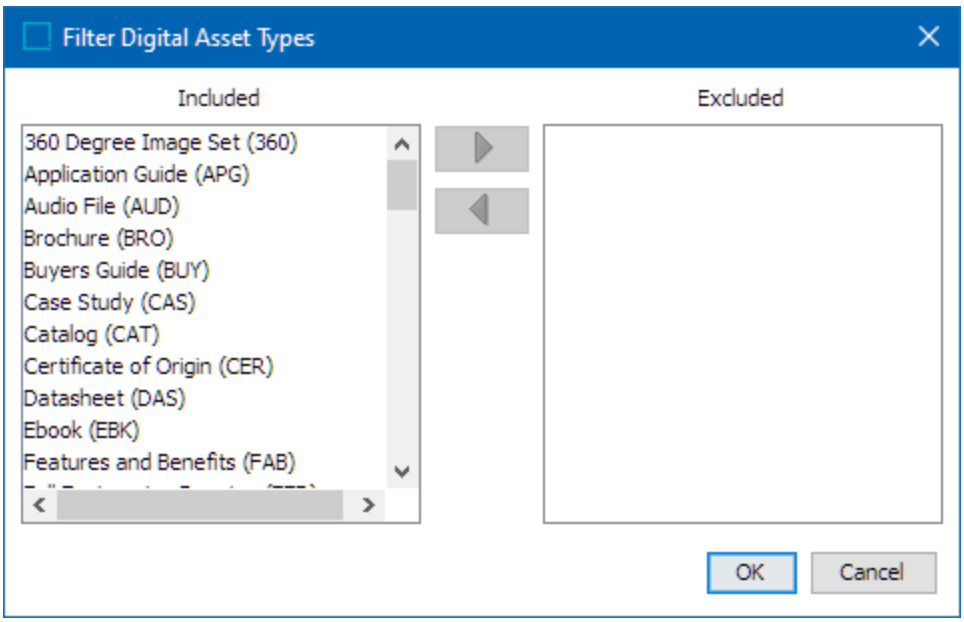
```
<App action="A" id="1">
    <BaseVehicle id="19129"/>
    <EngineBase id="95"/>
    <Note lang="en">Do Not Gap, Gap is Preset</Note>
    <Note lang="en">Original Equipment Is Iridium</Note>
    <Note lang="en"/>
    <Note lang="en"/>
    <Note lang="en"/>
    <Note lang="en"/>
    <Note lang="en"/>
    <Note lang="en"/>
    <Note lang="en"/>
    <Note lang="en"/>
    <Note lang="en"/>
    <Note lang="en"/>
    <Qty>4</Qty>
    <PartType id="7212"/>
    <MfrLabel>OE Fine Wire Double Iridium</MfrLabel>
    <Part>AC_PIESItem_BBMM_9610</Part>
</App>
```

```
<App action="A" id="1">
    <Years from="2006" to="2006"/>
    <Make id="58"/>
    <Model id="2884"/>
    <EngineBase id="95"/>
    <Note lang="en">Do Not Gap, Gap is Preset</Note>
    <Note lang="en">Original Equipment Is Iridium</Note>
    <Note lang="en"/>
    <Note lang="en"/>
    <Note lang="en"/>
    <Note lang="en"/>
    <Note lang="en"/>
    <Note lang="en"/>
    <Note lang="en"/>
    <Note lang="en"/>
    <Note lang="en"/>
    <Note lang="en"/>
    <Note lang="en"/>
    <Qty>4</Qty>
    <PartType id="7212"/>
    <MfrLabel>OE Fine Wire Double Iridium</MfrLabel>
    <Part>AC_PIESItem_BBMM_9610</Part>
</App>
```

**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**.

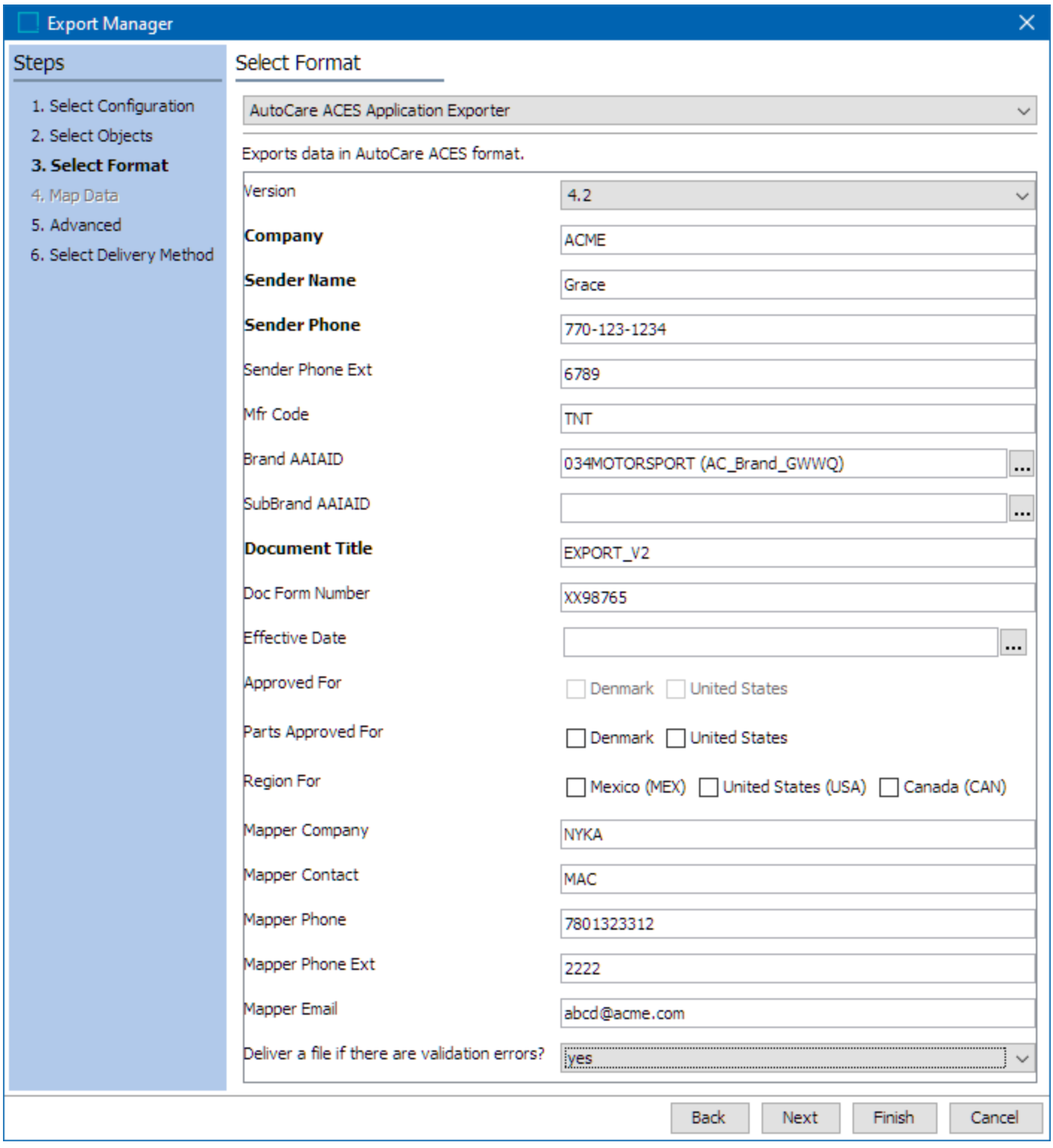
- **Digital Assets:** Selecting this parameter will include the digital asset information within the <DigitalAsset> tag in the exported ACES file. In addition, users can also filter Digital assets based on their Asset Type. The ellipsis button (...) that is next to the parameter will be activated only when the parameter checkbox is selected. Clicking the ellipsis button will display the Filter Digital Asset Types dialog (as shown below).

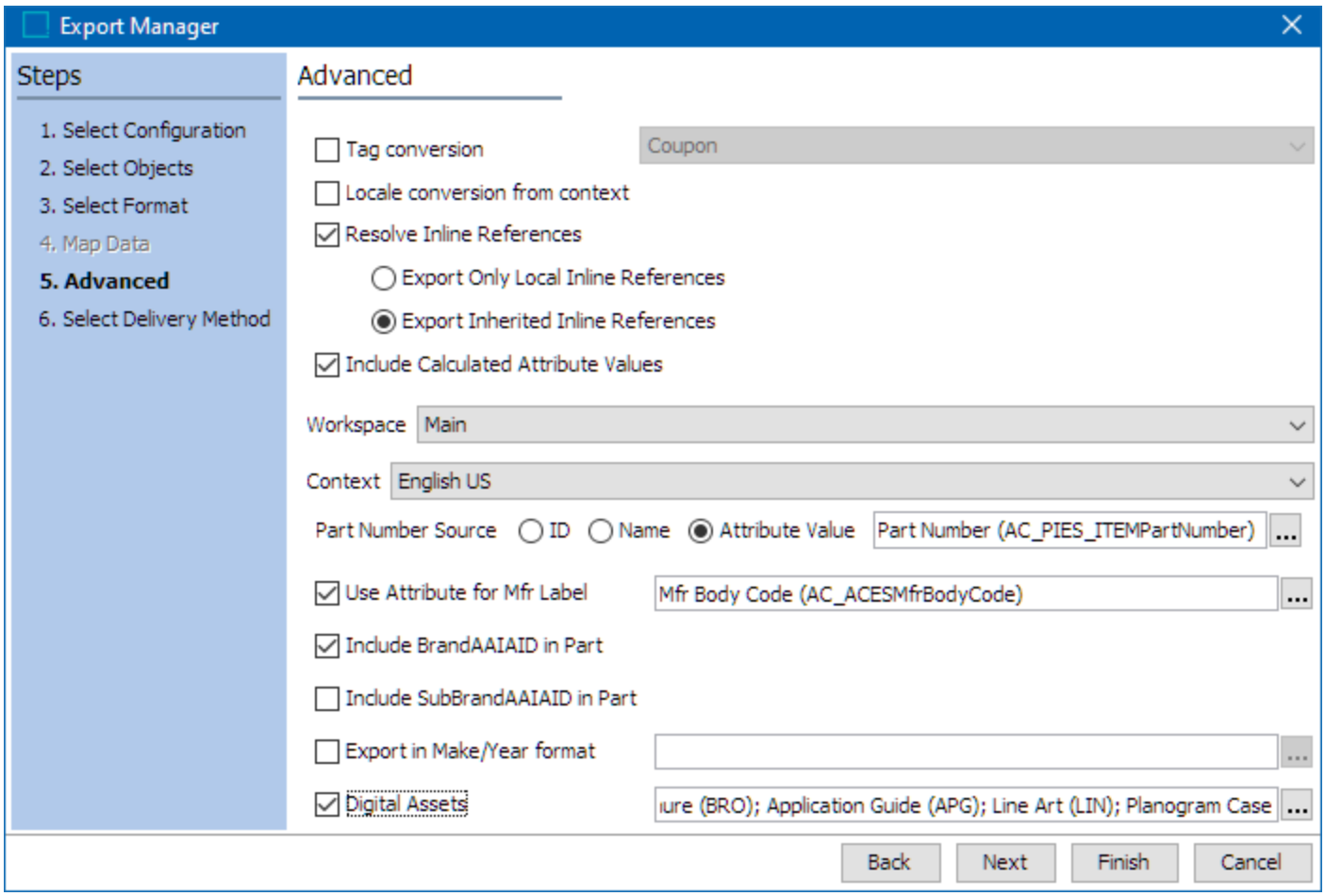
  To select the Asset Type to be excluded, users can click on the options individually from the Included column and use the (>) icon to move it to the Excluded column. Users can use features like Ctrl + click, Shift + click, or Ctrl + A (Select All) to multi-select. If no Asset Type is moved from the Included to the Excluded column, the export continues to function as before, and no digital assets will be excluded from the export.




**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
4.2
Company
ACME
Sender Name
Grace
Sender Phone
770-123-1234
Sender Phone Ext
6789
Mfr Code
TNT
Brand AAIAID
034MOTORSPORT (AC_Brand_GWWQ)
SubBrand AAIAID
Document Title
EXPORT_V2
Doc Form Number
XX98765
Effective Date
Approved For
Denmark
United States
Parts Approved For
Denmark
United States
Region For
Mexico (MEX)
United States (USA)
Canada (CAN)
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




```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ACES version="4.2">
    <Header>
        <Company>ACME</Company>
        <SenderName>Grace</SenderName>
        <SenderPhone>770-123-1234</SenderPhone>
        <SenderPhoneExt>6789</SenderPhoneExt>
        <TransferDate>2022-06-16</TransferDate>
        <MfrCode>TNT</MfrCode>
        <BrandAAIAID>GWWQ</BrandAAIAID>
        <DocumentTitle>EXPORT_V2</DocumentTitle>
        <DocFormNumber>XX98765</DocFormNumber>
        <EffectiveDate>2022-06-16</EffectiveDate>
        <SubmissionType>FULL</SubmissionType>
        <MapperCompany>NYKA</MapperCompany>
        <MapperContact>MAC</MapperContact>
        <MapperPhone>7801323312</MapperPhone>
        <MapperPhoneExt>2222</MapperPhoneExt>
        <MapperEmail>abcd@acme.com</MapperEmail>
```

```
        <VcdbVersionDate>2021-02-26</VcdbVersionDate>
        <QdbVersionDate>2021-03-26</QdbVersionDate>
        <PcdbVersionDate>2021-10-29</PcdbVersionDate>
    </Header>
    <App action="A" id="1">
        <BaseVehicle id="30489"/>
        <BodyNumDoors id="11"/>
        <RearBrakeType id="5"/>
        <BrakeABS id="8"/>
        <Qty>1</Qty>
        <PartType id="5696"/>
        <Part BrandAAIAID="DKGX">VC21499</Part>
        <AssetName>123</AssetName>
        <AssetItemOrder>1</AssetItemOrder>
        <AssetItemRef>1234</AssetItemRef>
    </App>
    <DigitalAsset>
        <DigitalFileInformation AssetName="123" action="A" LanguageCode="EN">
            <FileName>abc1.jpg</FileName>
            <AssetDetailType>360</AssetDetailType>
            <FileType>JPG</FileType>
            <Representation>A</Representation>
            <FileSize>123456</FileSize>
            <ColorMode>RGB</ColorMode>
            <OrientationView>ANG</OrientationView>
            <AssetDimensions UOM="PX">
                <AssetHeight>250</AssetHeight>
                <AssetWidth>250</AssetWidth>
            </AssetDimensions>
            <AssetDescription>High resolution image from brochure</AssetDescription>
            <FilePath>\Mfg\xyz.jpg</FilePath>
            <URI>http://www.mfg.com/Images/abc.jpg</URI>
            <FileDateModified>2013-01-31</FileDateModified>
            <EffectiveDate>2013-01-31</EffectiveDate>
            <ExpirationDate>2013-12-31</ExpirationDate>
            <Country>US</Country>
        </DigitalFileInformation>
    </DigitalAsset>
    <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 value for AssetType comes from the metadata attribute ID=AC_PIES_ASSTAssetType on the asset reference.
  - 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
    - AC_ACESToAsset

**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 or 4.1 or 4.2 format.

The version will default to 4.1. 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>

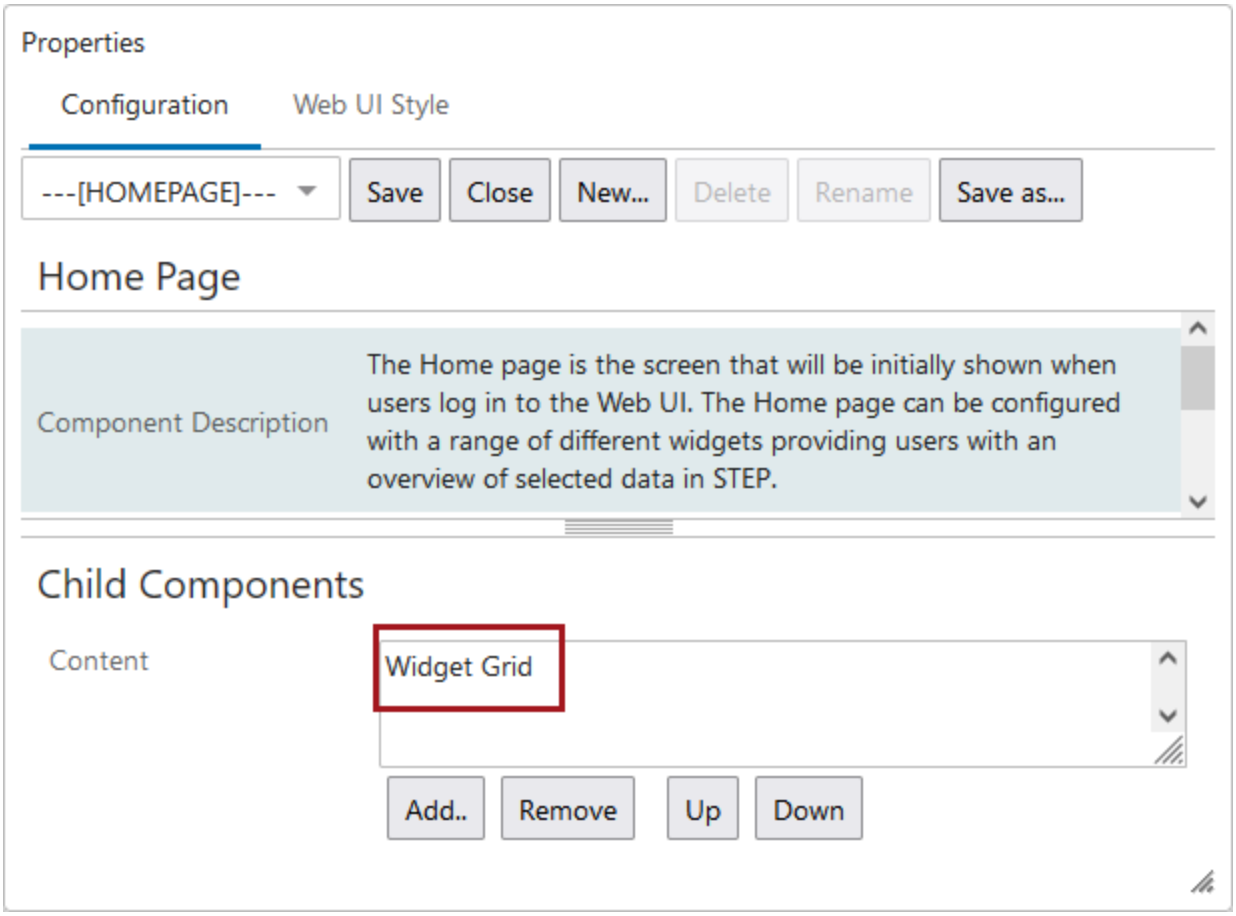
**Important:** This tag must be removed from the a 3.0 file before importing through the STEP ACES Importer, otherwise it will fail validation against 3.2 and 4.0 XSD.

## Start Export from Configuration Plugin

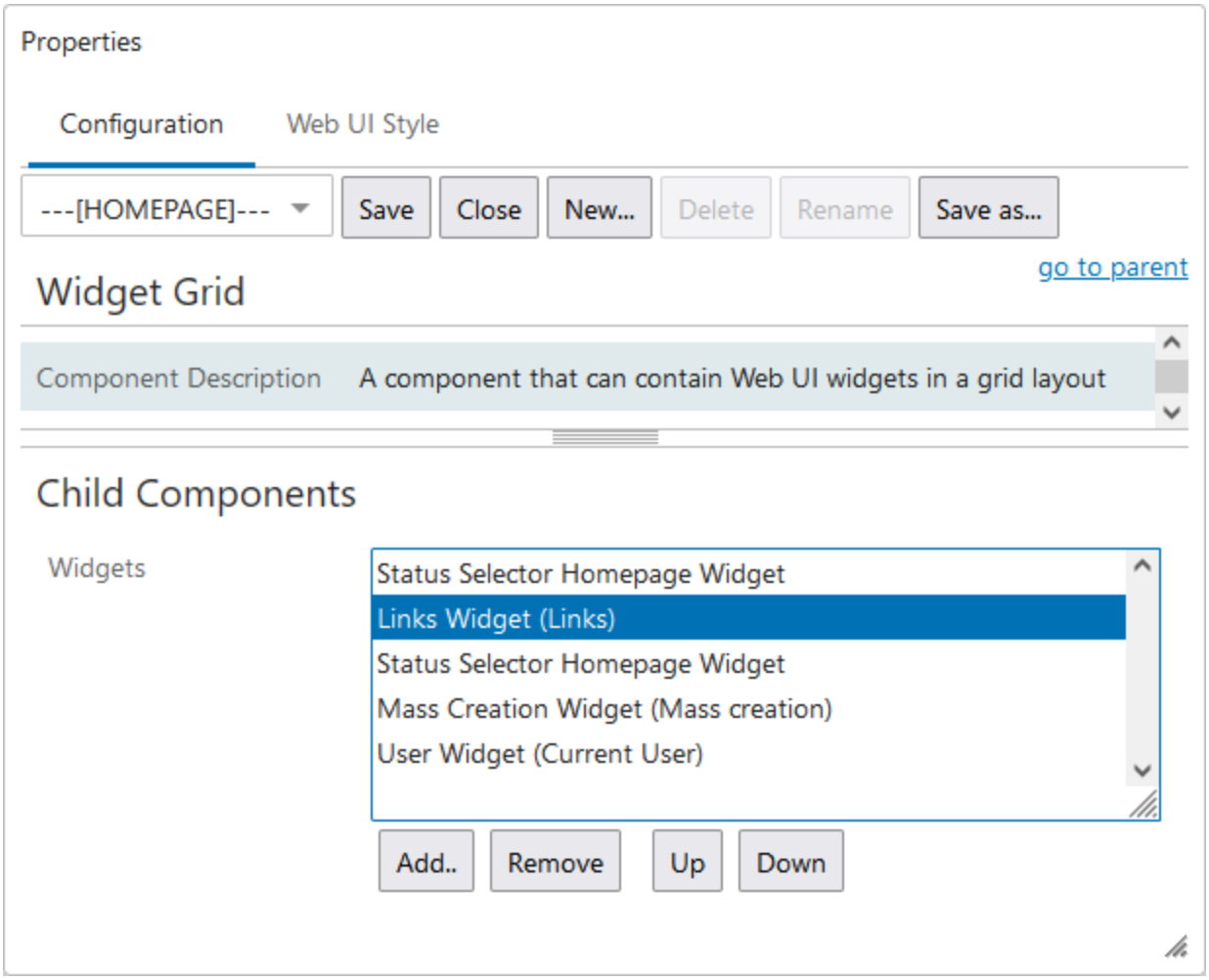
The Start Export from Configuration plugin, when configured within the Dialog Popup Widget, allows users to initiate a previously saved export configuration from the Homepage. While a user can export any product data, this plugin is mainly developed to export ACES applications from the Web UI. The Dialog Popup Widget with the Start Export from Configuration plugin configured within it can be accessed on the Homepage by adding it to the Links Widget.

To configure the Start Export from Configuration plugin, follow these steps:

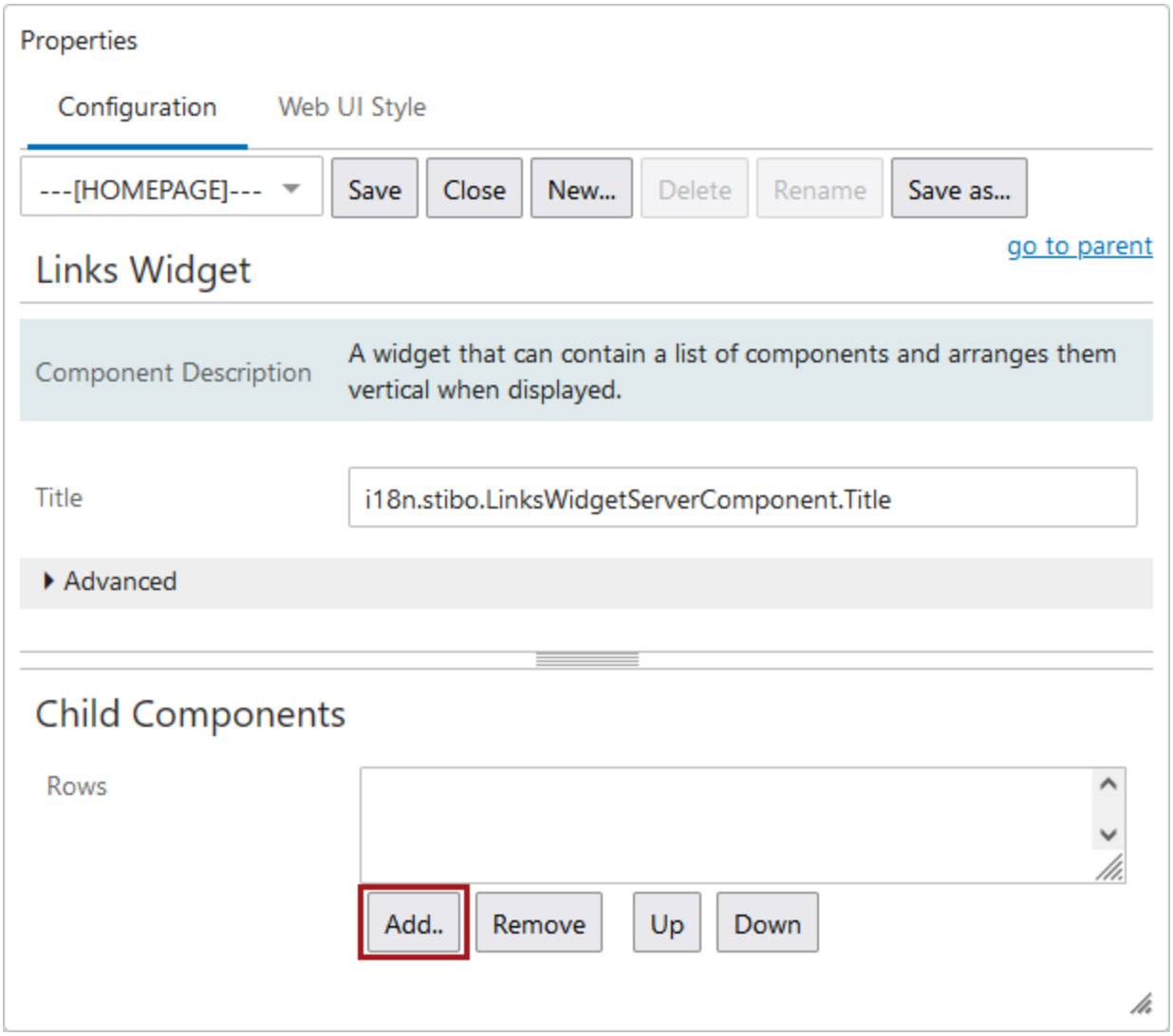
1. From the Homepage, go to Child Components > Content > click on the **Widget Grid**.



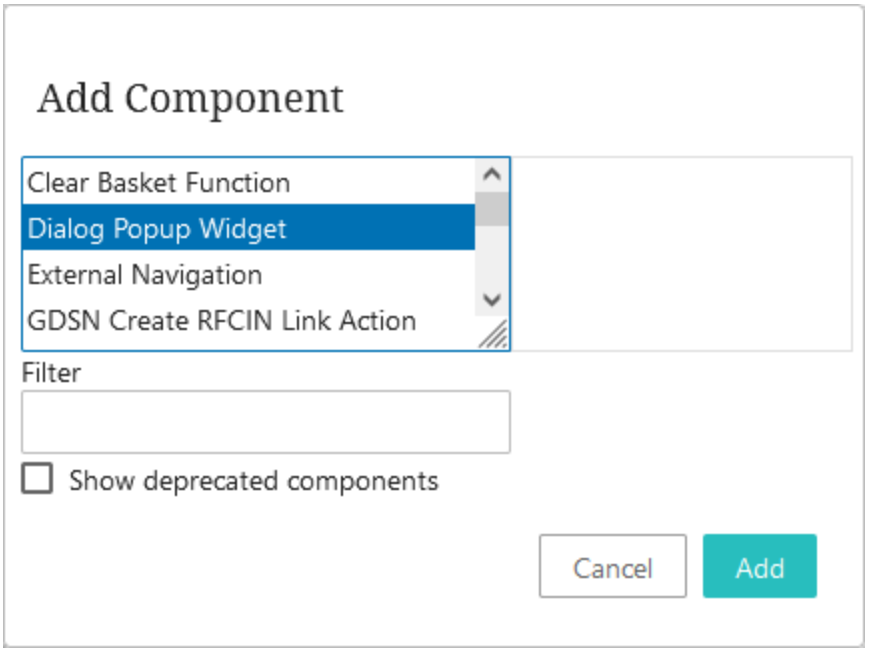

2. From the Widget Grid, got to Child Components > Widgets > double click on the **Links Widget**.



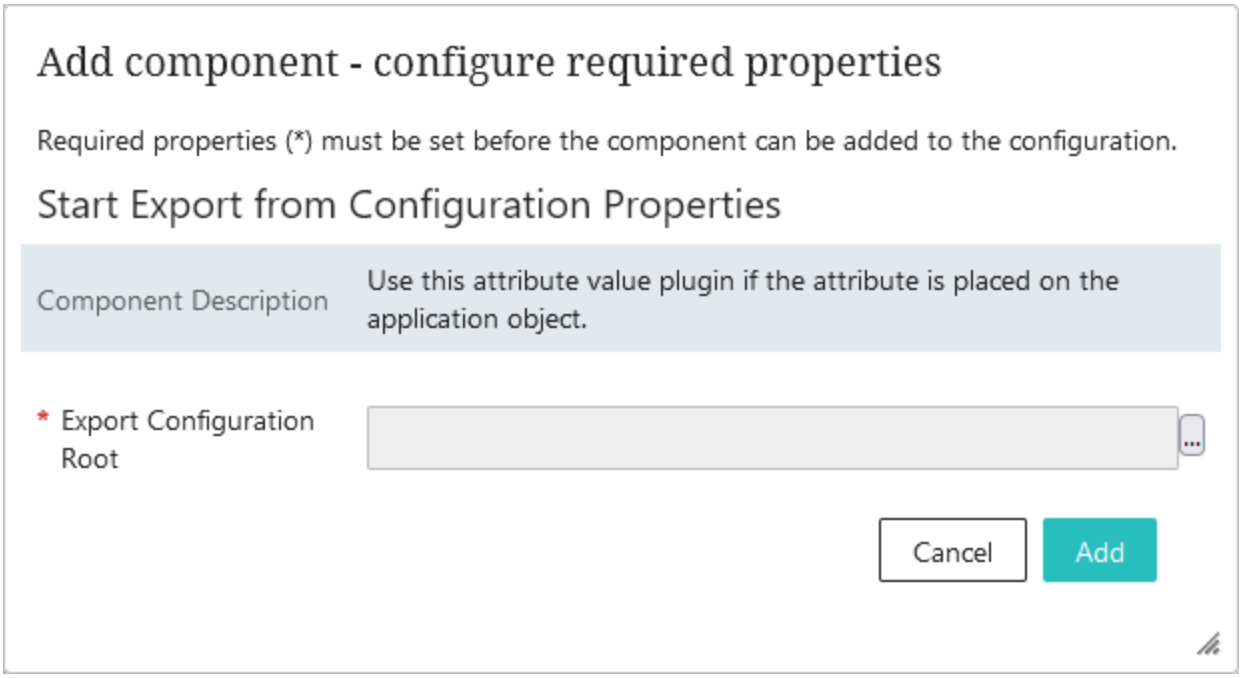

3. From the Links Widget, go to Child Components> Rows> and click on **Add**.




4. Select the **Dialog Popup Widget** from the Add Component pop-up window, and click **Add**.

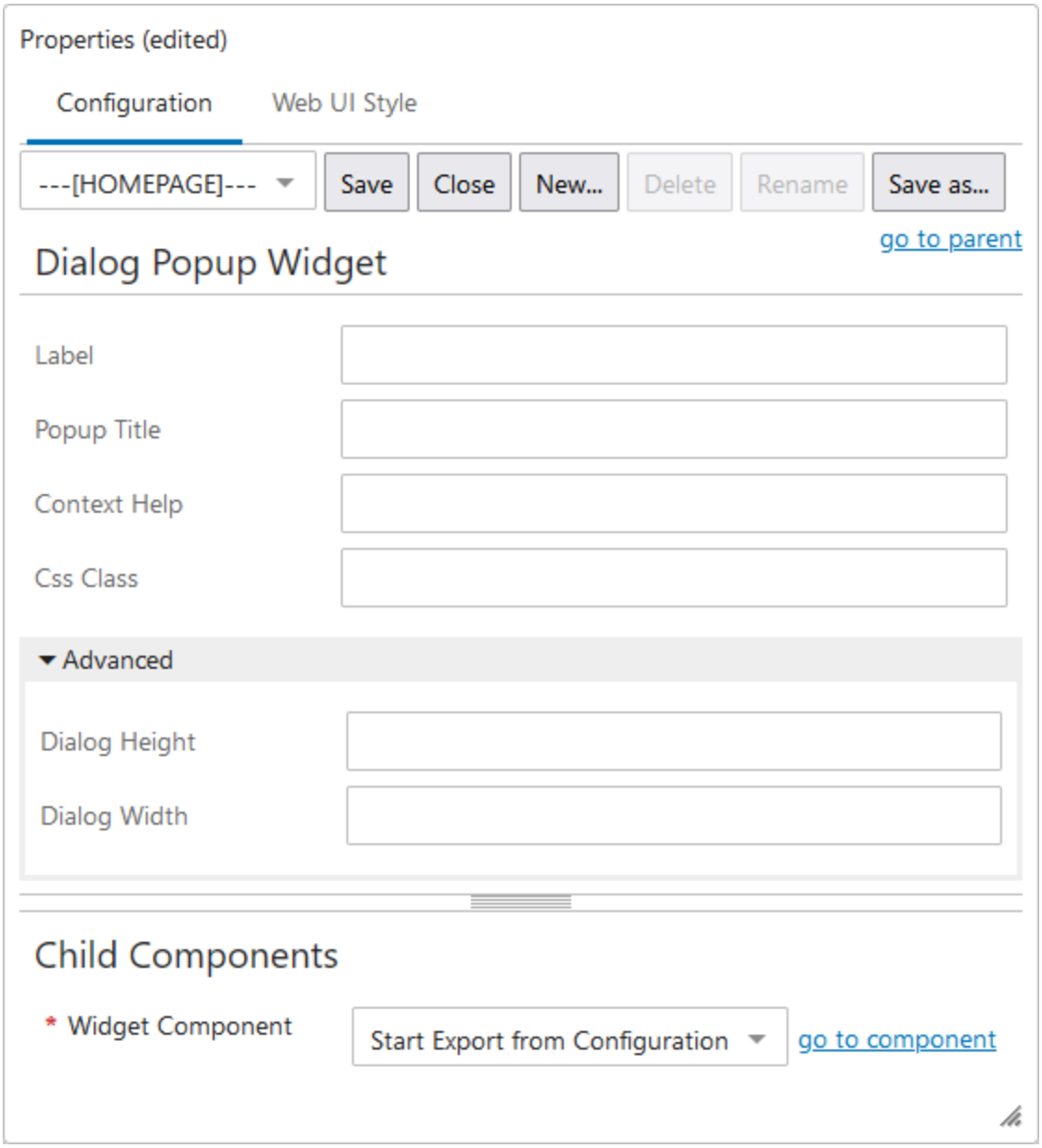



5. Double click on the Dialog Popup Widget, and select an **Export Configuration Root**.

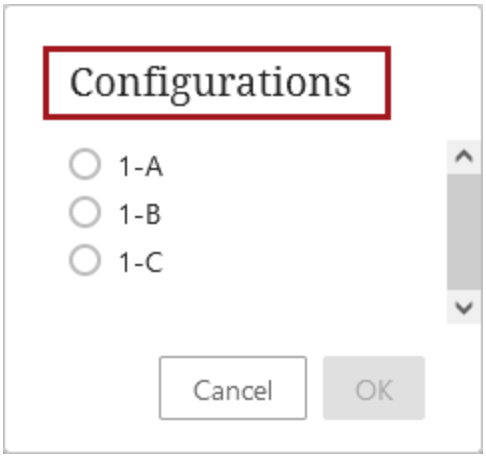



If the root needs to be changed, go to Child Components > Widget Component > and click **go to component** next to the "Start Export from Configuration" dropdown.

6. Fill in any other necessary parameters on the Dialog Popup Widget. Parameters are described below:

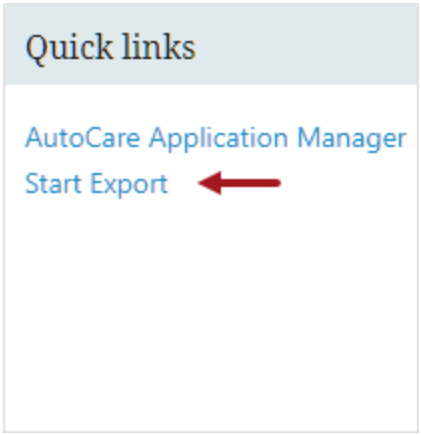



- **Label**: Name of the quick link on the Links Widget
- **Popup Title**: Title given to the popup dialog that displays when the Dialog Popup Widget link is clicked.




- **Context Help**: Text that displays when the Dialog Popup Widget is hovered over
- **Css Class**: Optional name of a ccs-style to be set as the primary style for the component.
- **Dialog Height**: Optional Dialog popup height in pixels.
- **Dialog Width**: Optional dialog popup width in pixels.

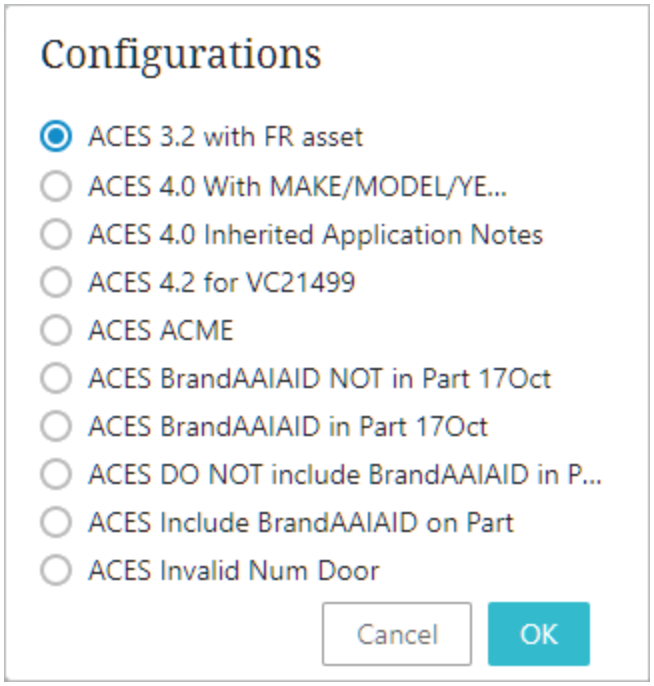
7. When finished configuring the link, click Save and close the designer. The link will appear in the Link Widget on the Web UI homepage.




## Exporting Data Using Start Export from Configuration plugin

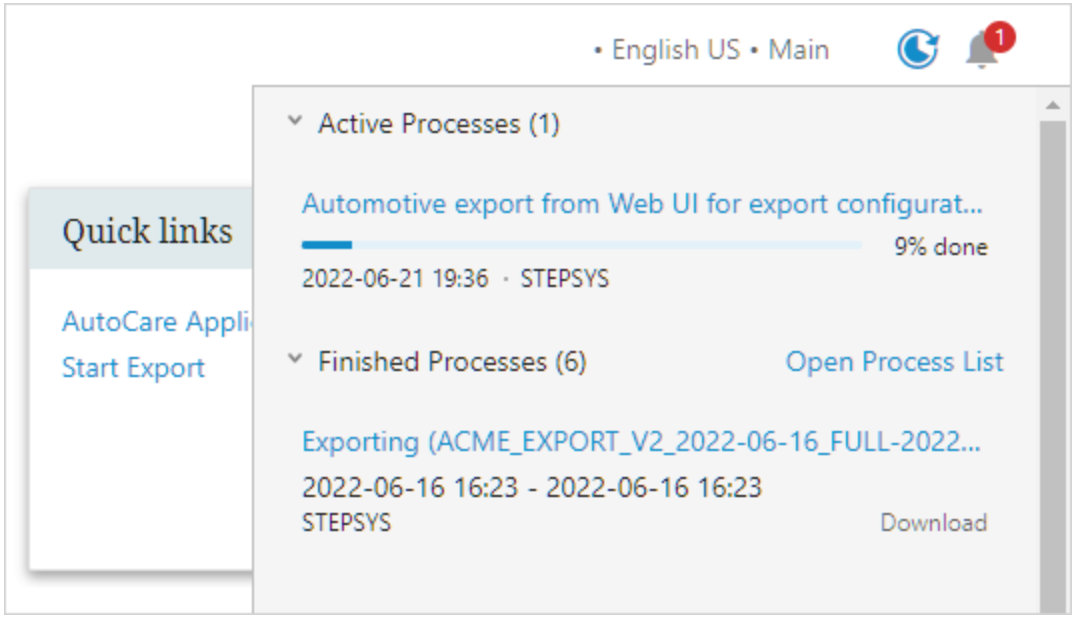
The central feature of the Dialog Popup Widget is to display a list of pre-saved export configurations that, when run, would export the desired data from the Home screen. An export configuration should already be saved in the system and configured in the widget in order to export any data. For more information on saving an export configuration in the system, refer to the **Running a Data Export** topic within **STEP Online Help**.

Clicking the link will display a list of export configurations that are available in the configured root node. Selecting an export configuration and clicking **OK** will create a background process and start exporting the file (or put it in the export queue).

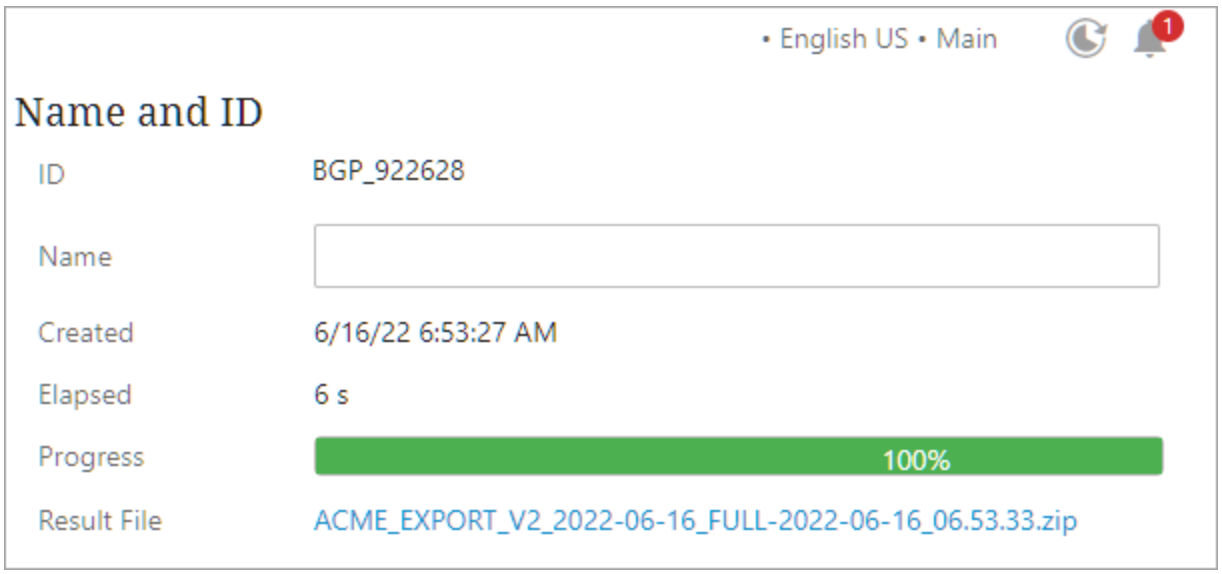



Users can view the progress of the export using the Background Process Notification Component.

In the example below, the BGP Notification Panel is expanded and the export is 9% processed.




Further, clicking on the background process displayed in the BGP Notification Panel will navigate you to view the details of the export process within the Background Process Details screen. Users can download the result file in the Background Process Details screen.


• English US • Main
1
Name and ID
ID
BGP_922628
Name
Created
6/16/22 6:53:27 AM
Elapsed
6 s
Progress
100%
Result File
ACME_EXPORT_V2_2022-06-16_FULL-2022-06-16_06.53.33.zip

# 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**.

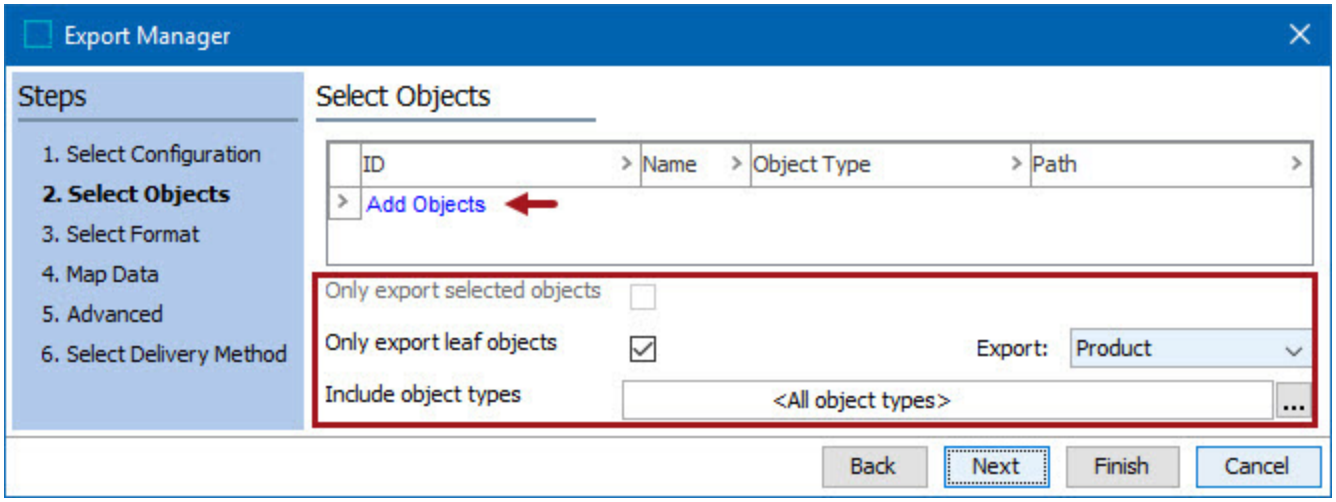
## Using PIES Exporter

### 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.



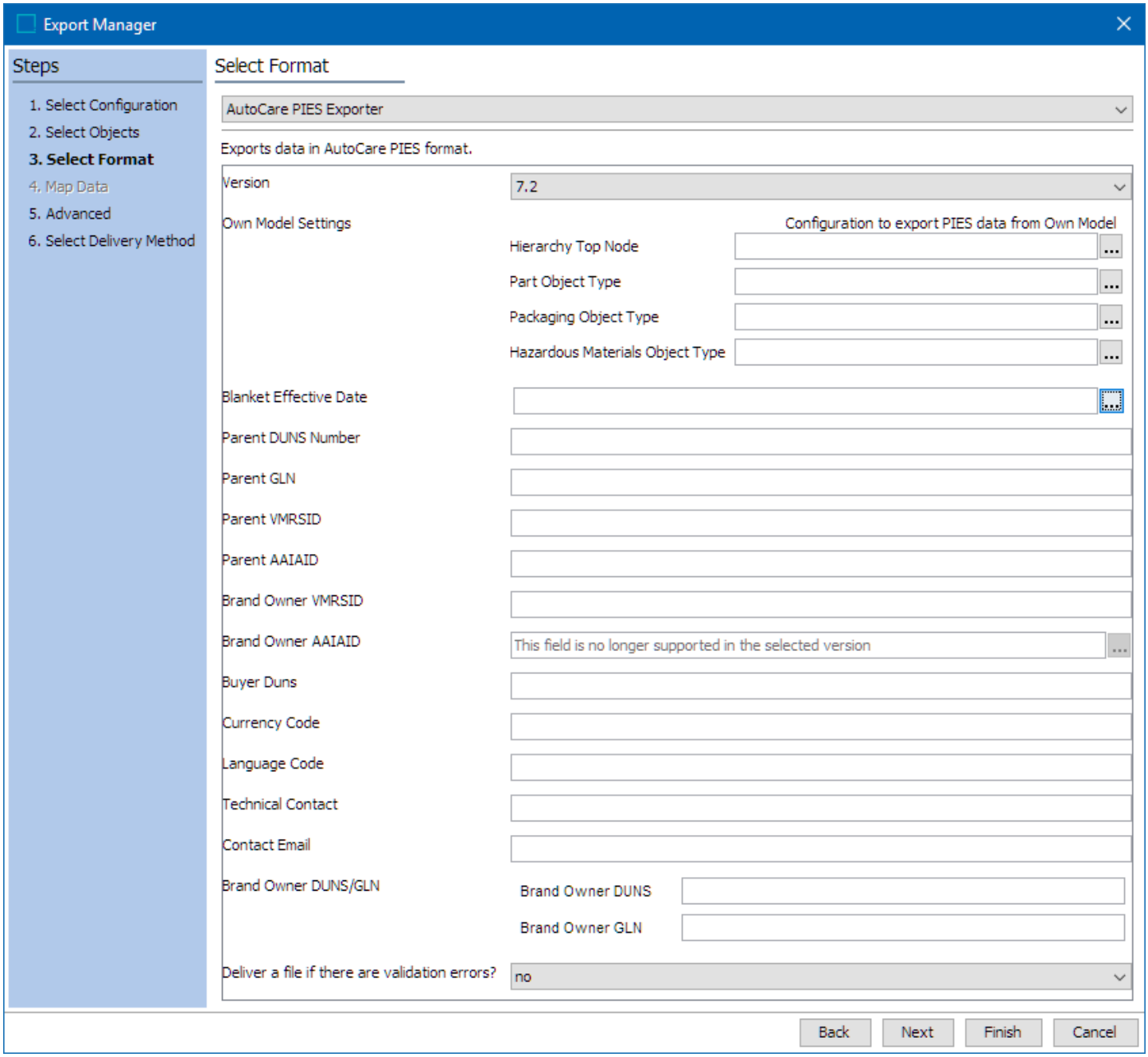
Click the **Add Objects** link to add one or more blue folders in the AutoCare PIES Products hierarchy, or one or more blue folders from an Own hierarchy, or one or more classification folders.

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 7.2, but any one of the available PIES export versions can be selected from the dropdown list.




All the parameters that are available within this tab are described below. By default, all the parameter values are blank unless an existing configuration is selected.

## Own Model Settings

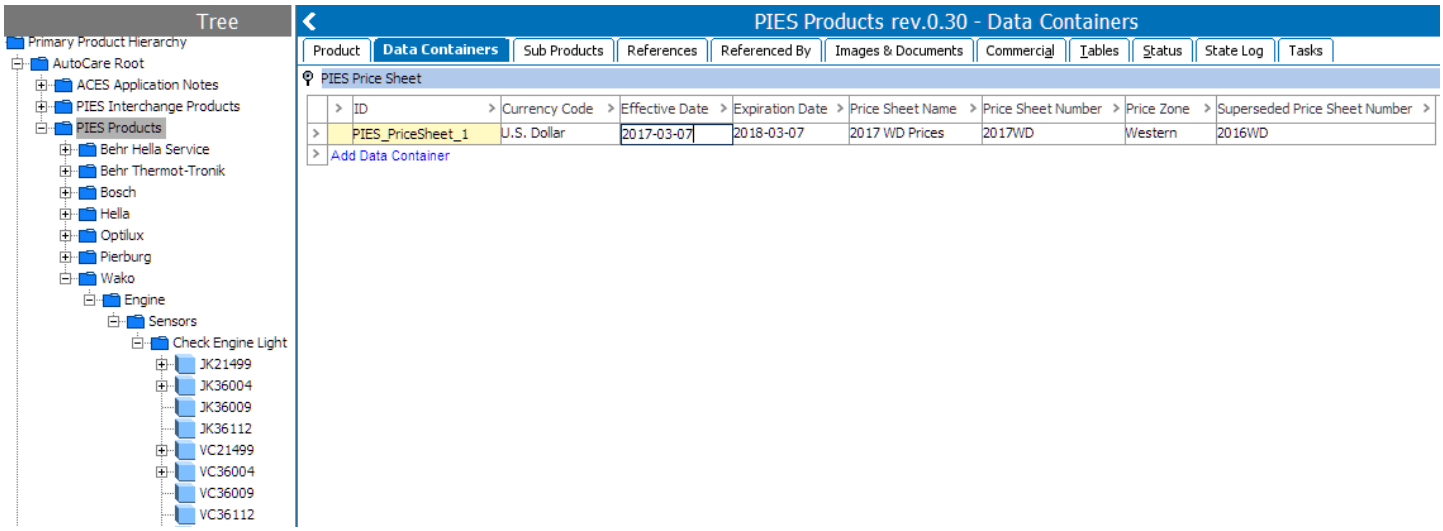
If the standard AutoCare model is used, no selections should be made in the Own Model Settings section. If an Own model is used, the Own Model Settings must be populated as described below.

- **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_PRCS_**. '

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:



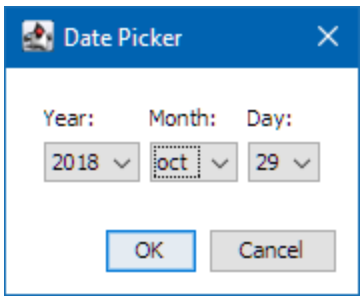

- **Part Object Type**: This is a required parameter for selection of the Own model object type representing the PIES Item.
  - The selected object must have all required PIES attributes and references (including the AC_ ProductToPartTerminology reference) from the standard model made valid on the object 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.
- **Version**: By default, the '7.2' 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, 7.1 and 7.2 are supported. Within the PIES export, the version displays between the <PIESVersion> tags within the Header column (as shown below).

```
<PIESVersion>7.2</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 Blanket Effective Date field prior to selection. Any past or future date can be selected, and changes made are displayed in the Blanket Effective Date field and populated in the export as YYYY-MM-DD. Within the PIES export, the date value will display between the <BlanketEffectiveDate> 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:** Allows a unique 13 digit number that includes the 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 athirst 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 'GIANT' is provided, then the exported PIES file will display as shown below.

```
<ParentVMRSID>GIANT</ParentVMRSID>
```

- **Parent AAIAID:** Allows a Brand Parent 4 character code to be entered. The 4 characters that are entered in this field will be used to validate against the Brand table once the export has started.

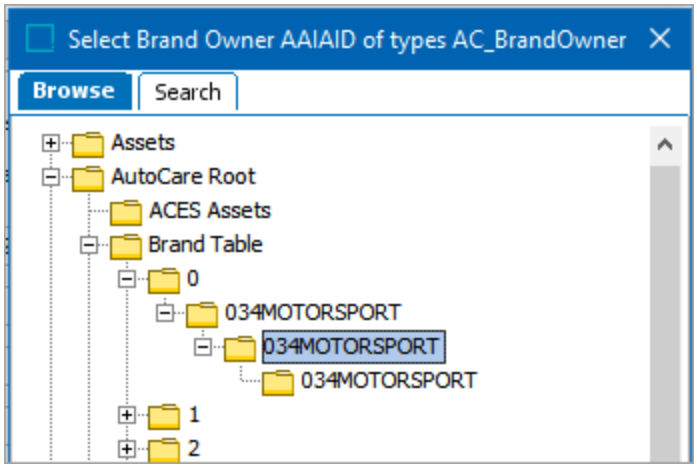
  For example, if the Brand Parent code 'GWWN' is entered, then the exported PIES file will display as shown below.

```
<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 'WONDR' is provided, then the exported PIES file will display as shown below.

```
<BrandOwnerVMRSID>WONDR</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. This field can be defined only in PIES version 7.1 and below.




Select the desired Brand Owner object. 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 the selected Brand Owner object will display between the <BrandOwnerAAIAID> tag in the export. Optionally, instead of browsing to select the Brand Owner object, users can type in the STEP Name of the object in the field and select the Brand Owner from the options in the drop-down list.

For example, if the Brand Owner 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 three character currency code (from the AC_PIESCurrencyCode LOV Value ID) to be specified within the free text field. If the characters do not match the LOV Value ID, then the field will be displayed as red. 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>
```

> **Note:** The Currency Code LOV is populated by the coded values from the PCdb file. If there is a Currency Code that does not exist in the LOV, then users can manually add the missing Currency Code LOV Value ID and Value to the AC_PIESCurrencyCode LOV.

- **Language Code:** Allows a two character language code (from the AC_PIESLanguageCode LOV Value ID) to be specified within the free text field. If the characters do not match the LOV Value ID, then the field will be displayed as red. 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>
```

**Note:** The Language Code LOV is populated by the coded values from the PCdb file. If there is a Language Code that does not exist in the LOV, then users can manually add the missing Language Code with the proper LOV Value ID and Value to the AC_PIESLanguageCode LOV.

- **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>
```

- **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 are entered.
  - **Brand Owner DUNS:** Allows a 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 either 9 or 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.

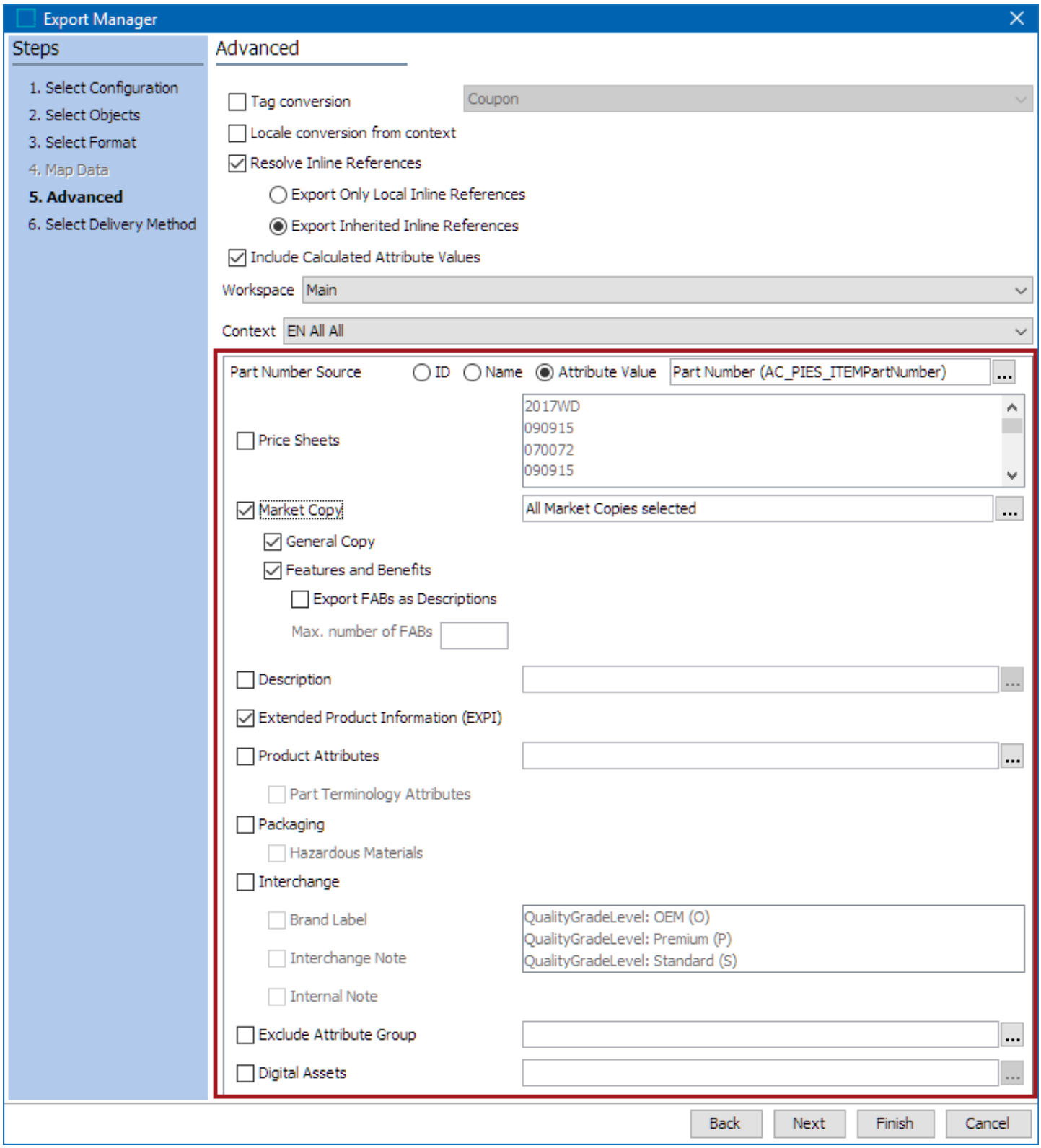
```
<BrandOwnerGLN>1100001009824</BrandOwnerGLN>
```

- **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

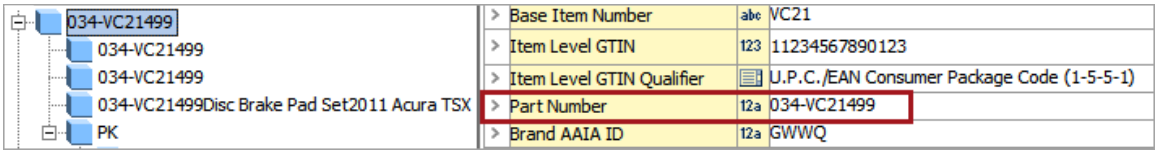
When a PIES format is selected in the previous dialog, the Advanced step within the Export Manager includes several parameters unique to the PIES exporter (outlined in red below). Only the parameters specific to the PIES format are described within this topic. For more information on the general parameters within the Advanced Export Manager dialog, refer to 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.


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
090915
Market Copy
All Market Copies selected
General Copy
Features and Benefits
Export FABs as Descriptions
Max. number of FABs
Description
Extended Product Information (EXPI)
Product Attributes
Part Terminology Attributes
Packaging
Hazardous Materials
Interchange
Brand Label
Interchange Note
Internal Note
QualityGradeLevel: OEM (O)
QualityGradeLevel: Premium (P)
QualityGradeLevel: Standard (S)
Exclude Attribute Group
Digital Assets
Back
Next
Finish
Cancel

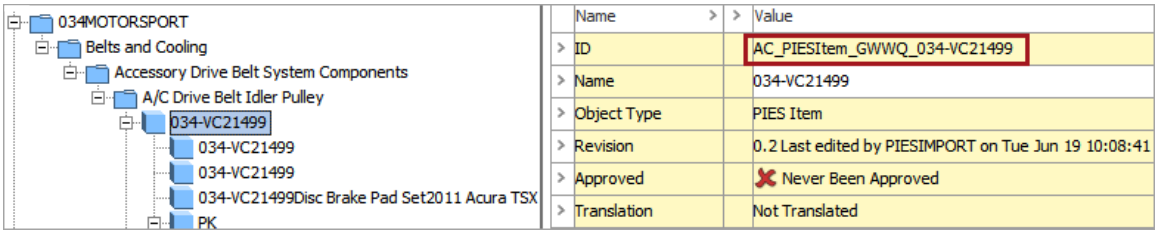
- **Part Number Source:** The following options are available for this parameter: ID, Name, Attribute Value.
  - **Attribute Value:** By default, the 'Attribute Value' radio button is selected and the 'Part Number (AC_PIES_ITEMPartNumber)' attribute value from the PIES Item is used to be inserted between the <PartNumber> tags in the exported file. For example, if the 034-VC21499 PIES Item is included in the PIES export and the Attribute Value Part Number Source option is selected, then the "034-VC21499" value from the Part Number attribute will be used to be inserted between the <PartNumber> tags as shown below.

    **Note:** If the part number should be retrieved from a different attribute, then users can select another attribute by clicking on the ellipsis button (...).



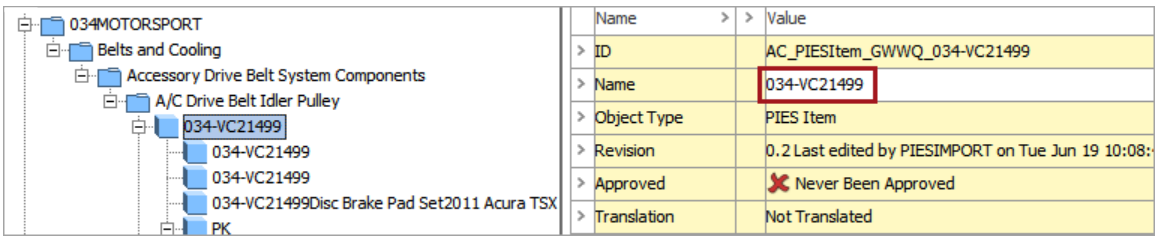

```
<PartNumber>034-VC21499</PartNumber>
```

  - **ID:** When enabled, the STEP ID of the PIES Item is used to be inserted between the <PartNumber> tags in the exported file. For example, if the 034-VC21499 PIES Item is included in the PIES export and the ID Part Number Source option is selected, then the STEP object ID will be used to be inserted between the <PartNumber> tags as shown below.




```
<PartNumber>AC_PIESItem_GWWQ_034-VC21499</PartNumber>
```

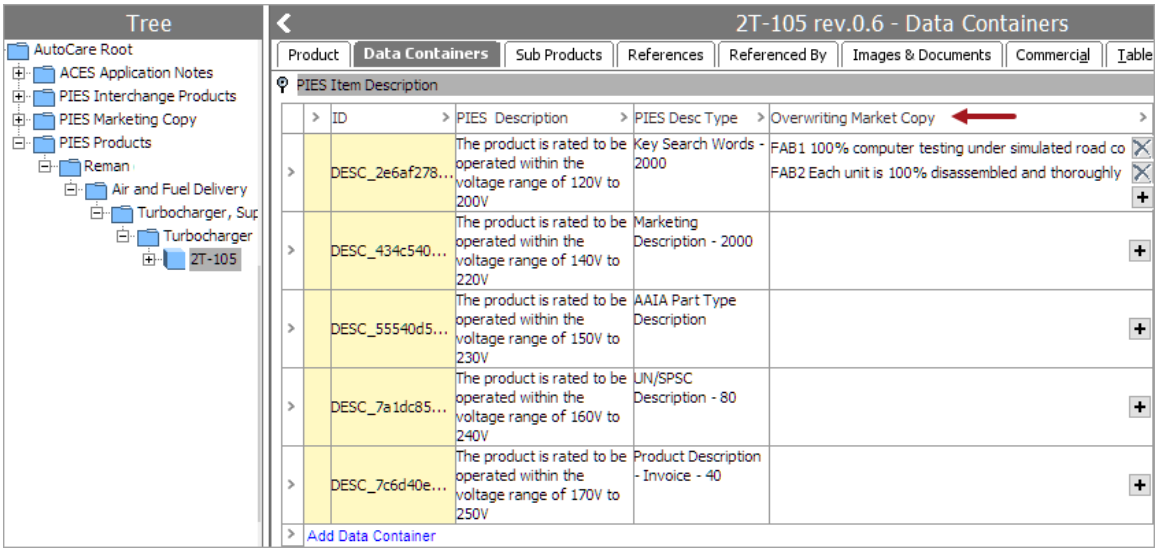
  - **Name:** When enabled, the STEP Name of the PIES Item is used to be inserted between the <PartNumber> tags in the exported file. For example, if the 034-VC21499 PIES Item is included in the PIES export and the Name Part Number Source option is selected, then the STEP object Name will be used to be inserted between the <PartNumber> tags as shown below.



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

- **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.
- **Market Copy:**If selected, all marketing copy that is linked to the Brand or Part Type that the PIES Item(s) belongs to will be included in the export. By default, both General Copy (GCC) and Features and Benefits (FAB) marketing copy options are selected and will get exported in the <MarketingCopy> segment. If these options are not selected, then no marketing copy will get exported.
    - The **Export FABs as Descriptions** option is active only if the Features and Benefits option is selected. Selecting this option will take the market copy content and include it within the Item <Descriptions> segment instead of the <MarketingCopy> segment.

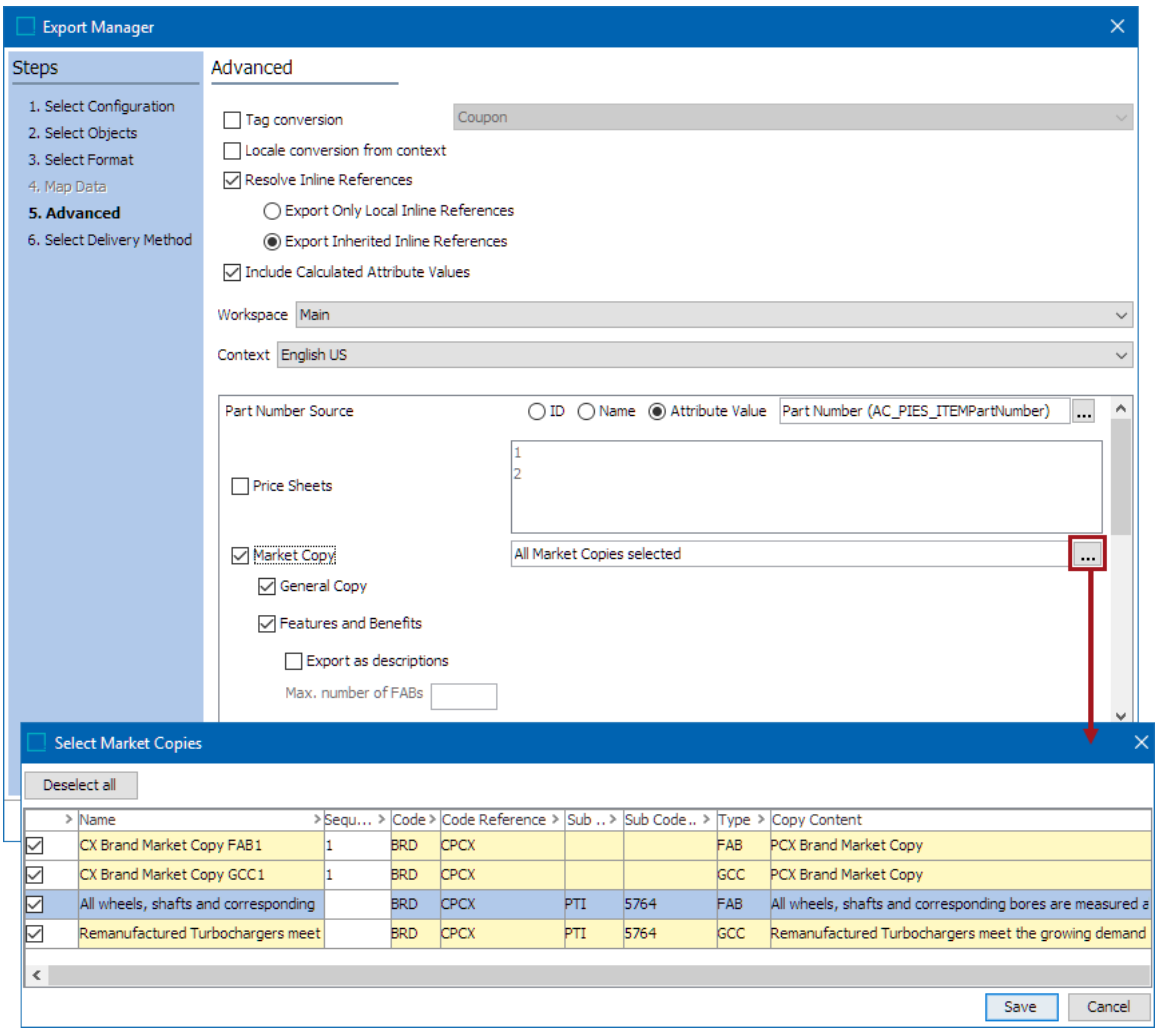
      For this functionality to work, there should be a reference established from the PIES Item Descriptions (AC_PIESItem_Description) data container to the Market Copy object (AC_ PIESMarketCopy) using the reference type Overwriting Market Copy (ID = AC_PIES_ DescriptionToMarketCopy)



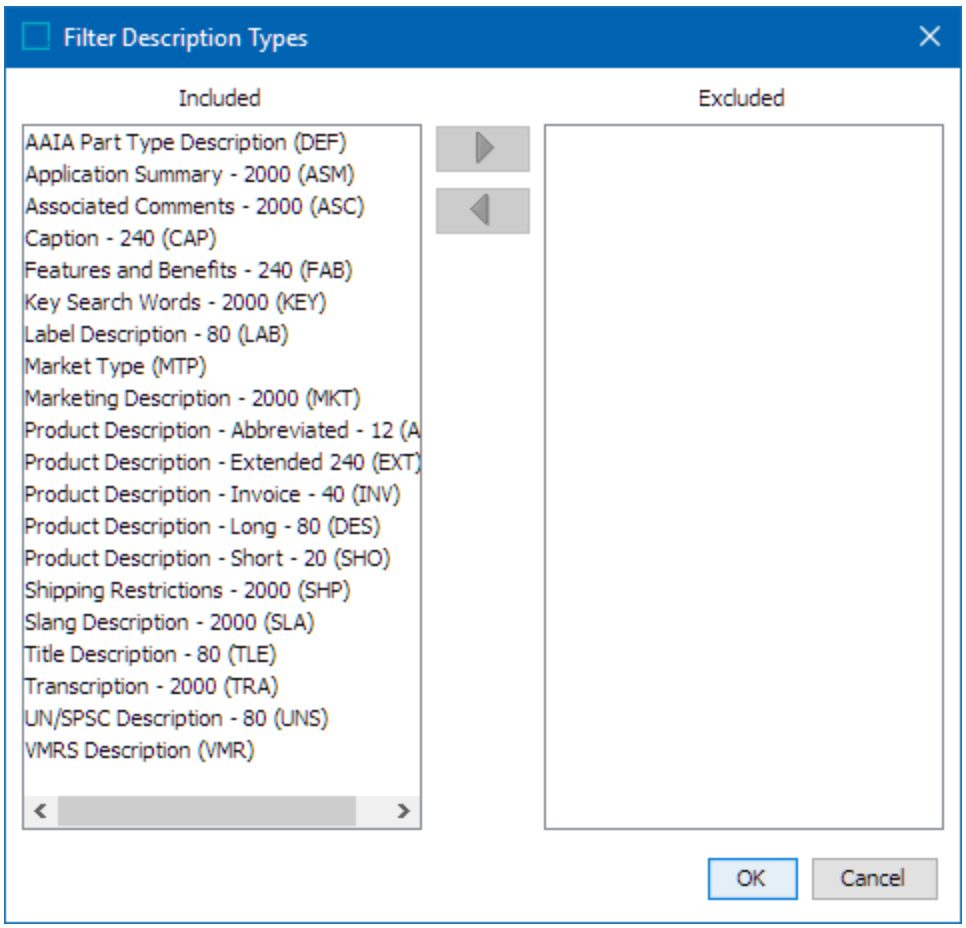

**Note:** The Descriptions parameter that is available within the Advanced step of the PIES Export Manager has to be selected to include <Descriptions> segment in the exported file. Selecting 'Export as descriptions' parameter without selecting the Descriptions parameter will not include the FAB type Market Copies in the exported file.

- The **Max. number of FABs** field gets active only when the 'Export as descriptions' parameter is selected. This parameter allows users to define the maximum number of FAB type of Market Copies that can be written within the <Descriptions> segment.

Click on the ellipsis button (...) to open the Select Market Copies window where users can filter market copy content that should get exported by selecting or deselecting the available copy content.




- **Description:** If selected, all Description Types within the PIES Item Description data container for the PIES object will be exported in the <Descriptions> segment. By default, Descriptions of all Description Types will be included in the exported file. However, to exclude any Descriptions based on the Description Type, click the ellipsis button (...) that is available next to the parameter to display the File Description Types dialog (as shown below). Select a Description Type from the left 'Included' panel and click the right arrow button ( ) to move the selected Description Type(s) to the right 'Excluded' panel. Based on the selected Description Type (s), the Descriptions belonging to the excluded Description Types will not be included in the exported file.

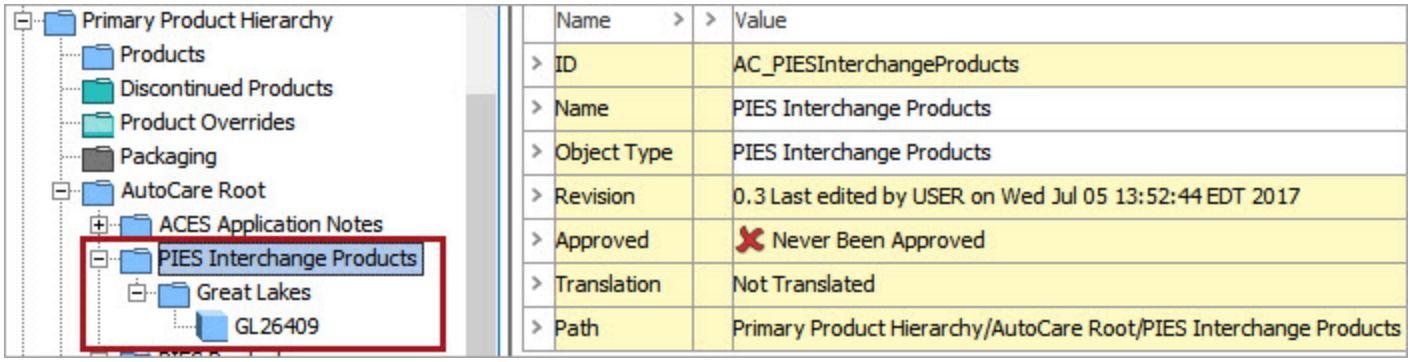



- **Extended Product Information (EXPI):** If selected, all standard PIES EXPI attributes that are valid for the PIES object and have a value will be exported in the <ExtendedInformation> segment.
- **Product Attributes / Part Terminology Attributes:** Allows the selection of attribute groups that contain 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). Selecting the Part Terminology Attributes option will export all PAdb attributes that are linked to the Part Type if there are values for those attributes. If a user only wants their own custom attributes to get exported but do not want to include PAdb attributes, then an attribute group must be selected first before the user can deselect the Part Terminology Attributes option. 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. By default, if no Quality Grade Level / Type Code is selected after selecting the Interchange parameter, then the exported PIES file *will not* include any PIES Interchange items. It is important to always select the Quality Grade Level / Type Code after the 'Interchange' parameter is checked. The selection of the 'Brand Label', 'Interchange Note', and the 'Internal Note' parameters will include the values from the Brand Label, Interchange Note, and the Internal Note attribute (related to the PIES Interchange item) in the exported file.

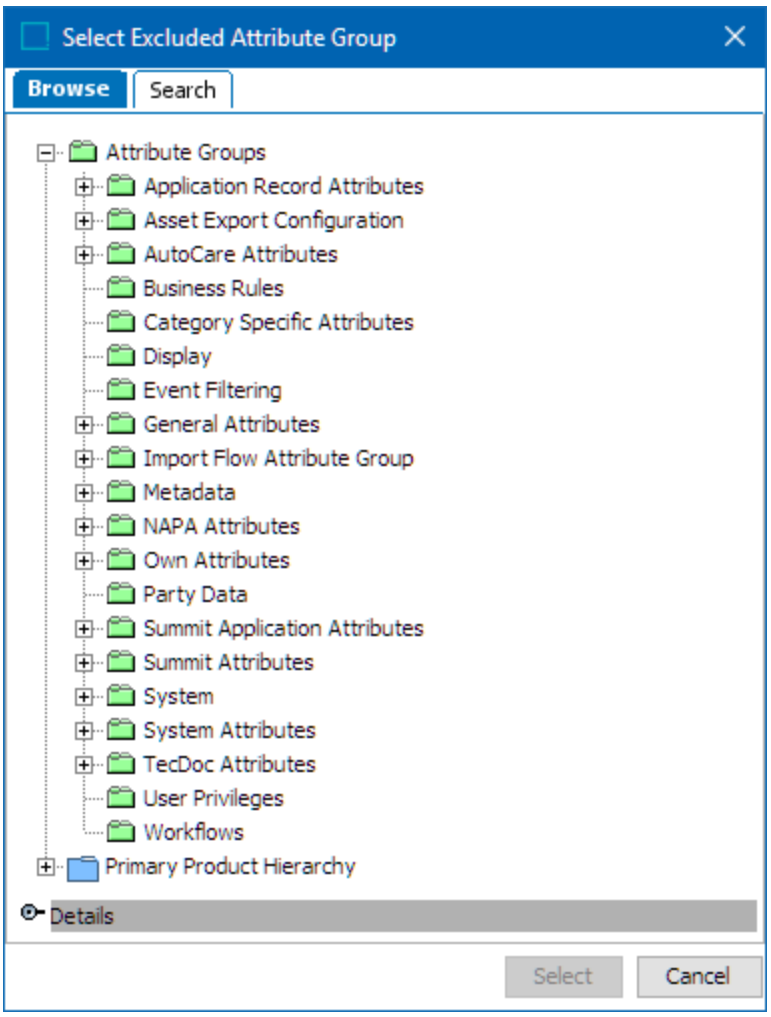
  **Note:** Users can filter the PIES Interchange items based on their Type Code only in the PIES export file version 6.5, and can filter the PIES Interchange items based on their Quality Grade Level only in the PIES export file versions of 6.7 and above.

  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).




  **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.

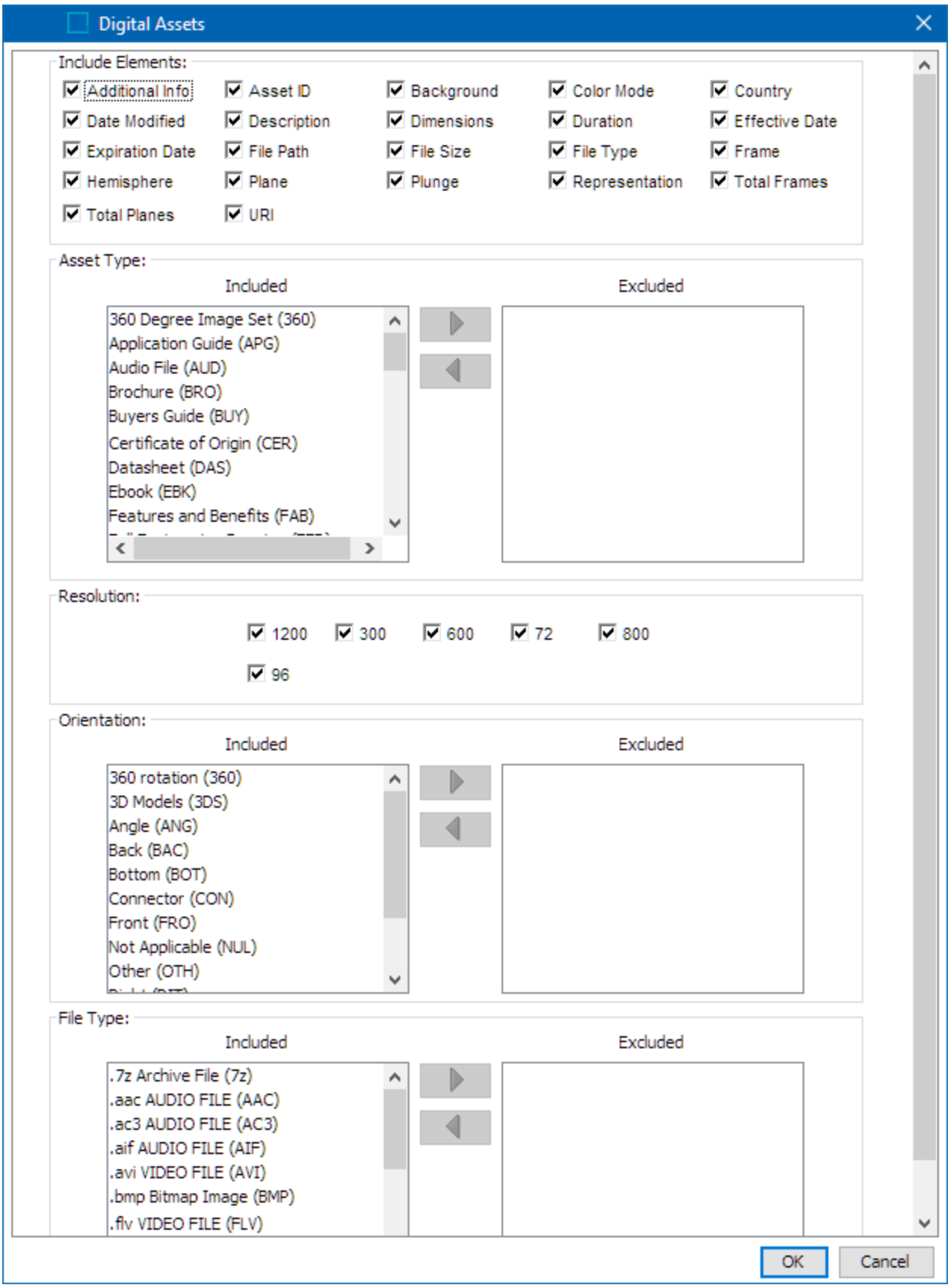
- **Exclude Attribute Groups:** This parameter allows users to select attribute groups that can be excluded from being exported. By default, if no attribute group(s) is defined after selecting the Exclude Attribute Group parameter, the export continues to function as before and no attribute groups will be excluded from the export. After selecting the parameter, click the ellipsis button (...) that is available next to the parameter to display the 'Select Excluded Attribute Group' dialog (as shown below) and select an attribute group that is to be excluded from being exported.




- **Digital Assets:** This parameter allows users to filter Digital assets based on their Asset Type, Resolution, Orientation, and File Type. In addition, users can also select which Digital Asset attributes should be excluded from being exported. Clicking the ellipsis button (...) that is available next to the parameter will display the Digital Assets dialog (as shown below). If no asset attribute is deselected in the Include Elements section, then the export continues to function as before and no attribute will be excluded from export. Similarly, if no Asset Type, Resolution, Orientation, and File Type is moved from the Included to the Excluded column, the export continues to function as before and no Digital assets will be excluded from export.

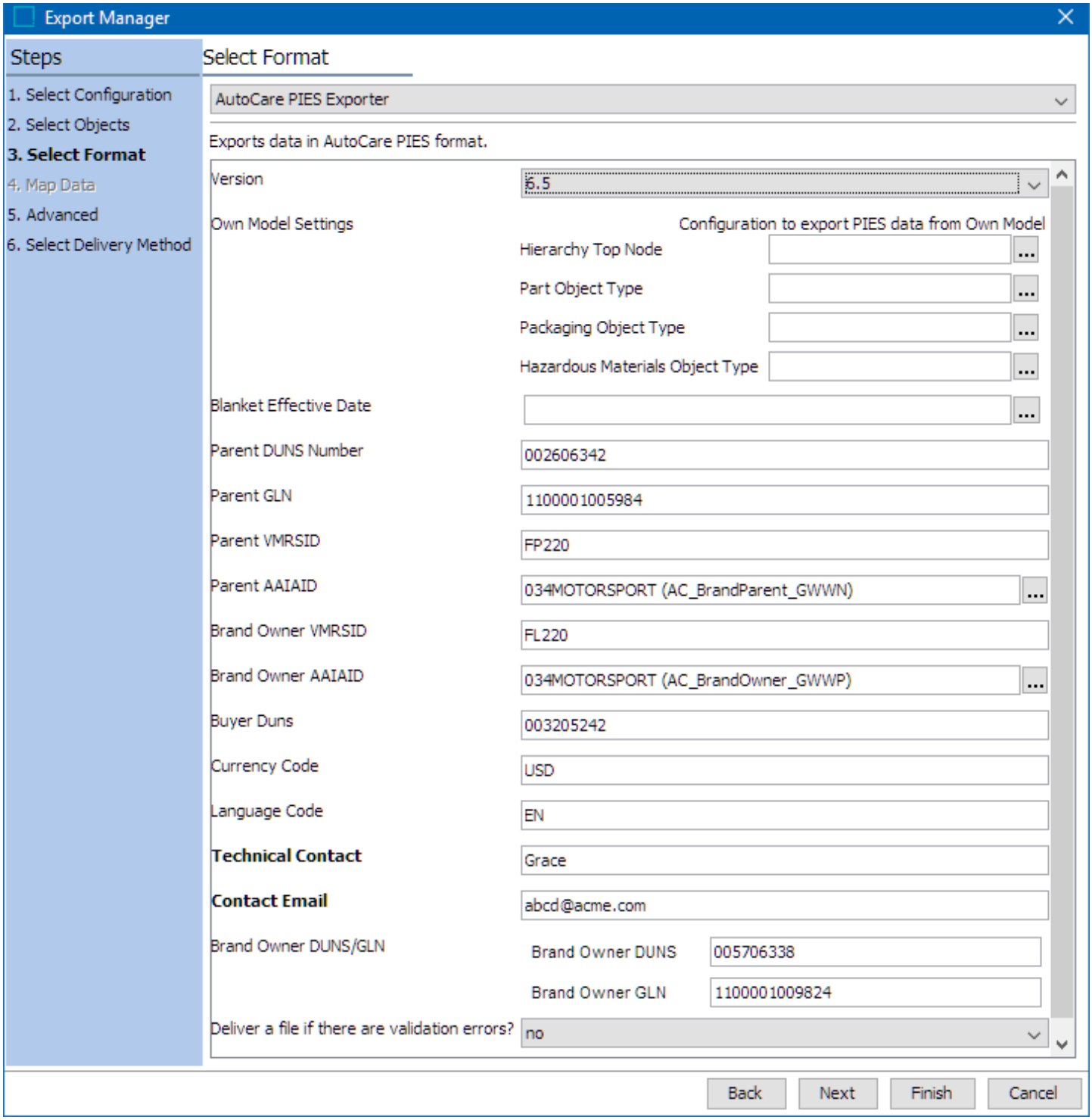
  To select the Asset Type, Orientation, or File Type to be excluded, users can click on the options individually from the Included column and use the (>) icon to move it to the Excluded column. Or use Ctrl + click, Shift + click, or Ctrl + A (Select All) to multi-select.

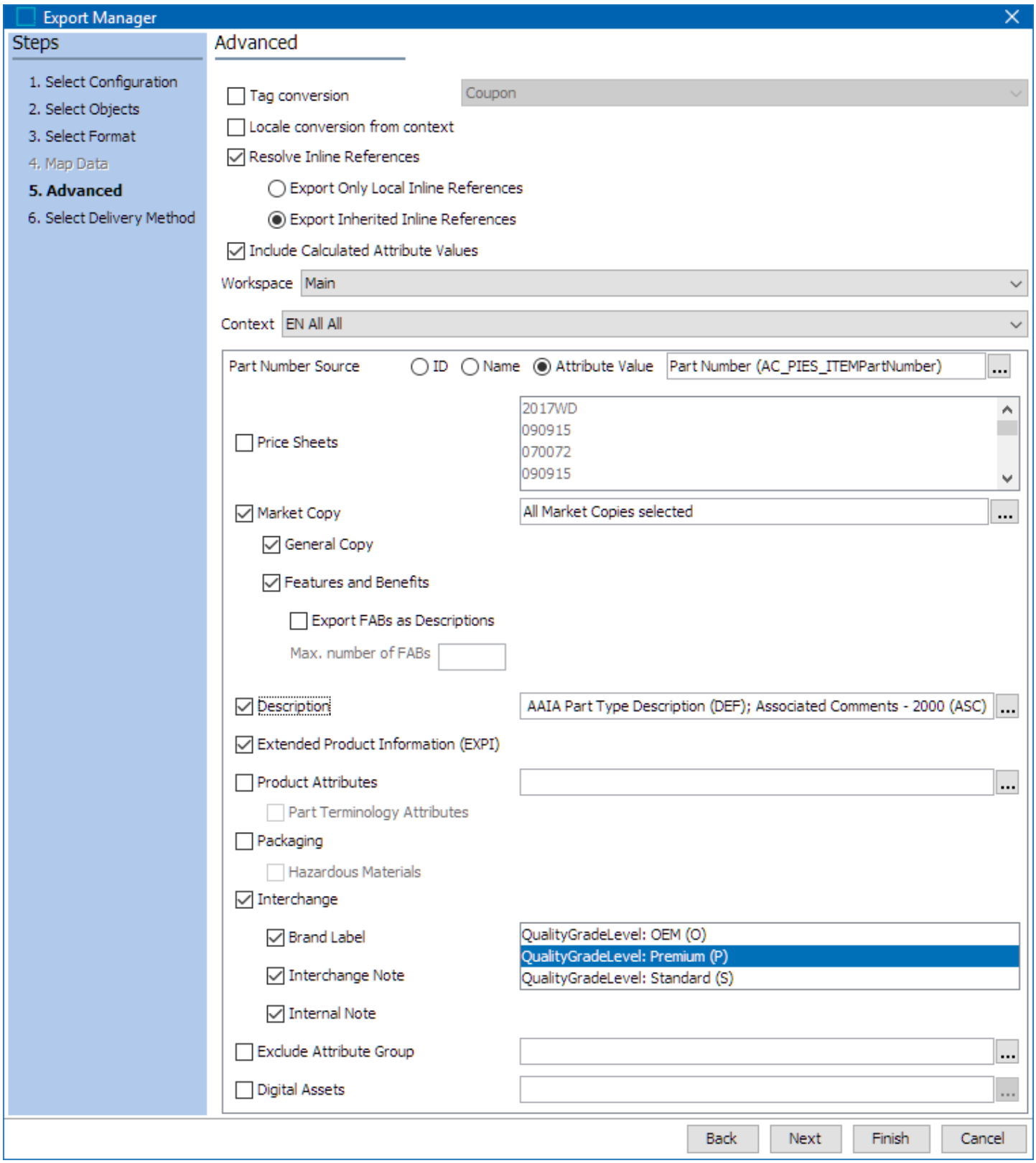
  For example, if the user wants to specify that certain Asset Types such as Primary Photo (P04) and User Defined (ZZ1-ZZ9) should be exported, but exclude all other Asset Types, then the user will be able to do that by selecting all other Asset Types that are not Primary Photo (P04) and User Defined (ZZ1-ZZ9), and moving the other Asset Types to the Excluded column.


Digital Assets
Include Elements:
Additional Info
Asset ID
Background
Color Mode
Country
Date Modified
Description
Dimensions
Duration
Effective Date
Expiration Date
File Path
File Size
File Type
Frame
Hemisphere
Plane
Plunge
Representation
Total Frames
Total Planes
URI
Asset Type:
Included
Excluded
360 Degree Image Set (360)
Application Guide (APG)
Audio File (AUD)
Brochure (BRO)
Buyers Guide (BUY)
Certificate of Origin (CER)
Datasheet (DAS)
Ebook (EBK)
Features and Benefits (FAB)
Resolution:
1200
300
600
72
800
96
Orientation:
Included
Excluded
360 rotation (360)
3D Models (3DS)
Angle (ANG)
Back (BAC)
Bottom (BOT)
Connector (CON)
Front (FRO)
Not Applicable (NUL)
Other (OTH)
File Type:
Included
Excluded
.7z Archive File (7z)
.aac AUDIO FILE (AAC)
.ac3 AUDIO FILE (AC3)
.aif AUDIO FILE (AIF)
.avi VIDEO FILE (AVI)
.bmp Bitmap Image (BMP)
.flv VIDEO FILE (FLV)
OK
Cancel

### Example

Below is an example of how the exporter could be configured and the expected PIES xml file results for one part.






```
<?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>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>

-<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

**Note:** NAPA: The commercial license has been discontinued for this functionality.

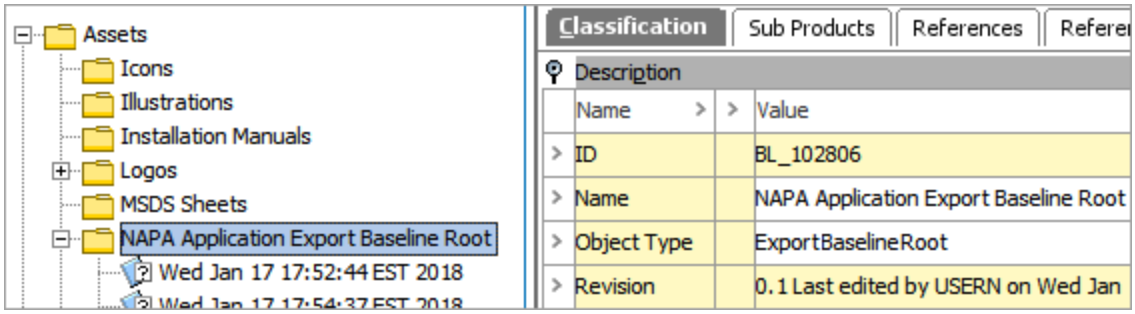
The NAPA Application Exporter is used to export application data that resides in a NAPA product hierarchy. Users can select an object from the NAPA product hierarchy, and the NAPA Application objects beneath the selected object (both child objects and linked objects) will be exported. The export file 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 the 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.

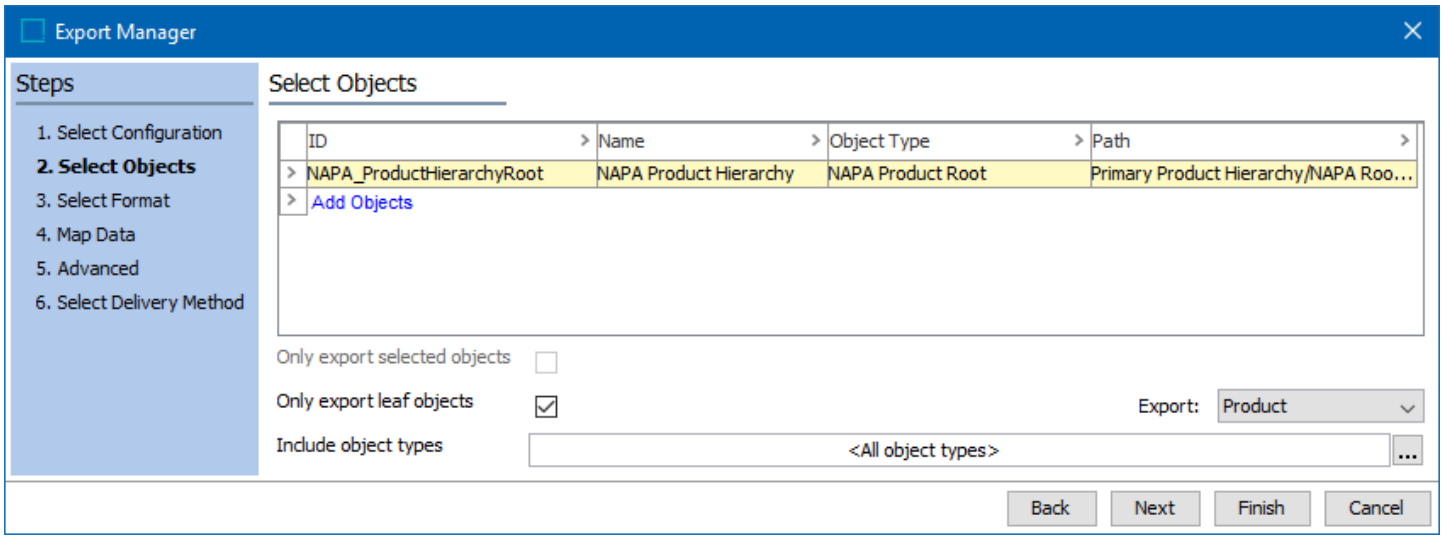
- **Object Type Requirements:** The following object types must exist within STEP: ExportBaselineRoot (Alternate Classification) and ExportBaselineAsset (Assets). As of the Automotive 8.3 release, these object types are created automatically when Easy Setup is run via System Setup > Component Models > 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.



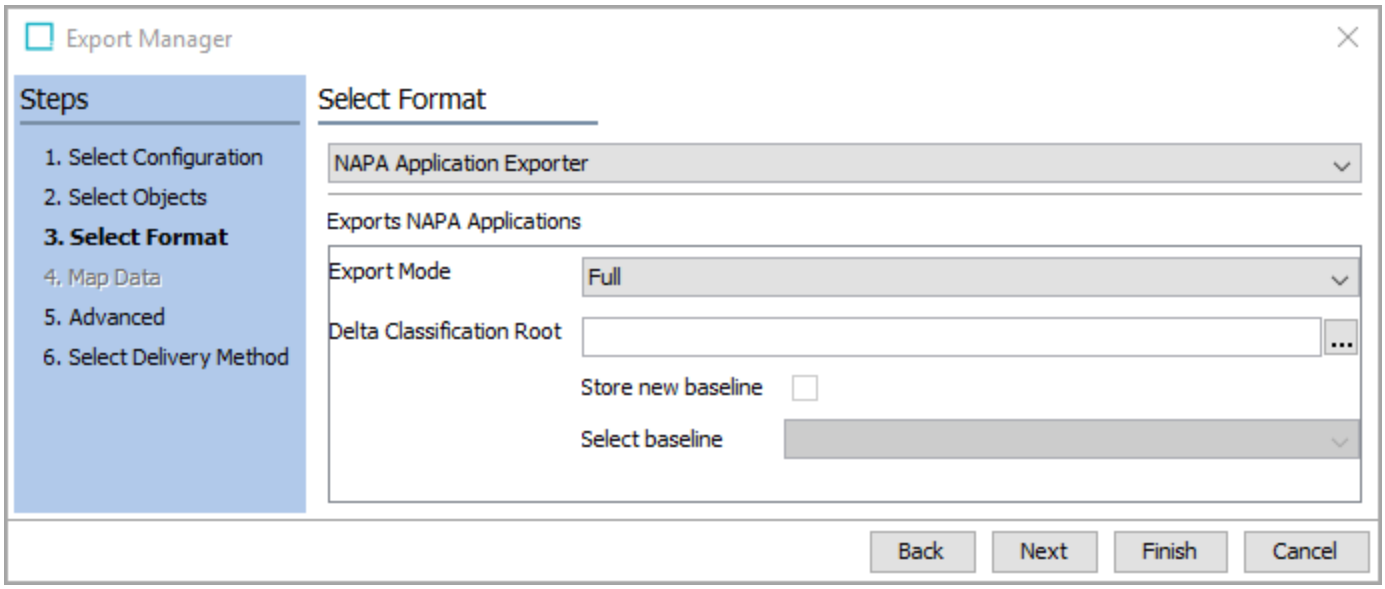

## Using the NAPA Application Exporter

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.




2. 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.

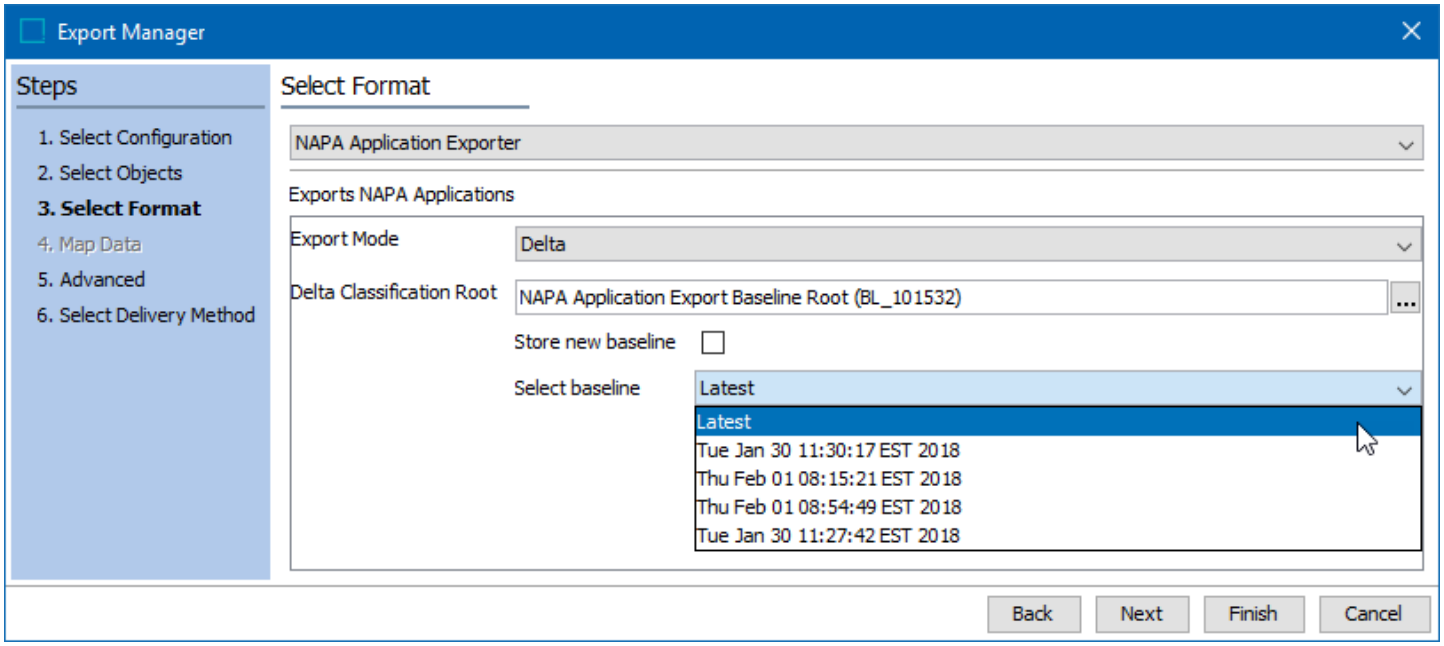



3. Select the one of the following Export Modes from the dropdown:
   - **Full:** This export mode will export applications linked to and/or beneath the objects selected in the previous step.
   - **Delta:** This export mode will export only applications changed in comparison to the baseline selected 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.

4. 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.
5. 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).




6. 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, refer to the **Export Manager** topic within the **Data Exchange** section of STEP Online Help.

# 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, refer to the **Supported Versions and Formats** topic.

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

# Import Framework

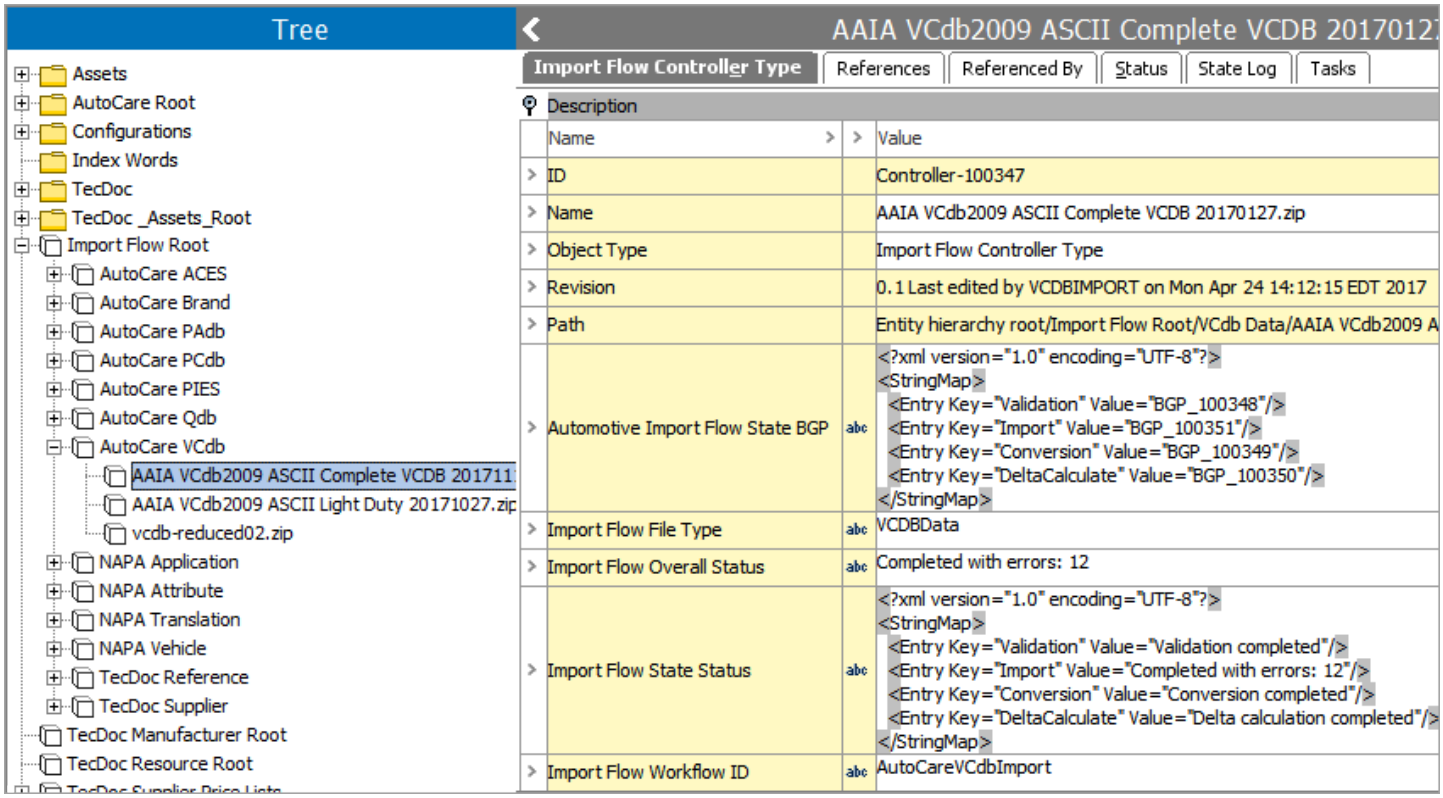
The intention of the import aspect of the Automotive core solution is to provide out-of-the-box importers for the various automotive standards, for which each customer can then apply their own validations, business processes, and data management procedures. To do this successfully, it is crucial to understand 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, refer to 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 Homepage 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:



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).

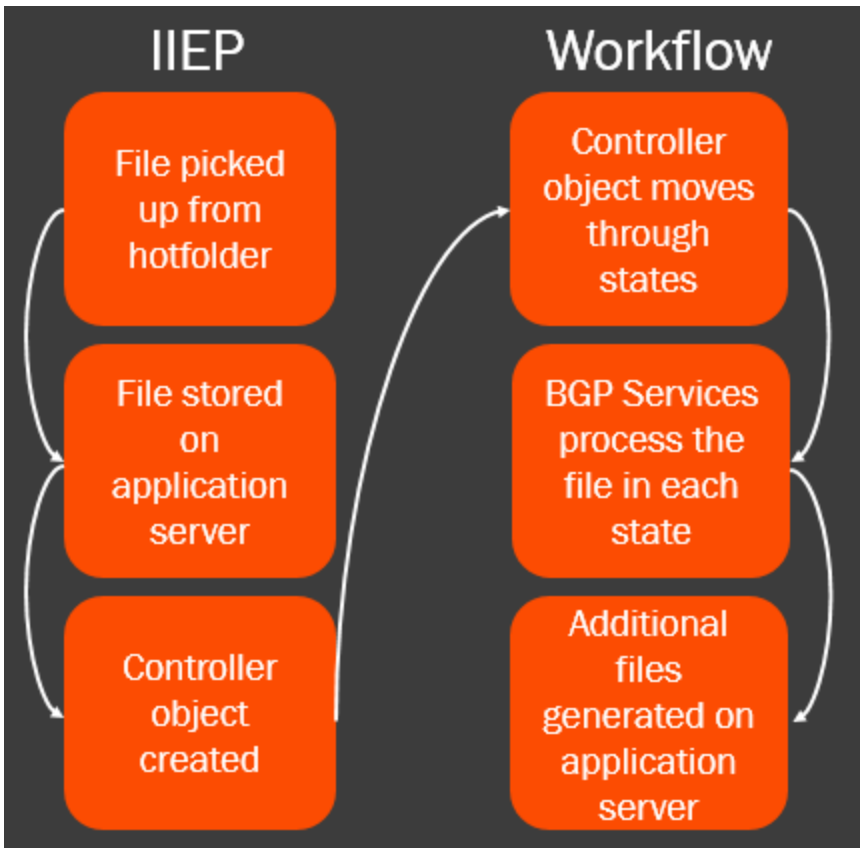
**Note:** All of the information displayed on the controller Entity is also displayed within the Web UI Import Controller Screen, which are discussed in the **Quick Start for Users** section of the **Automotive Quick Start Guide**.

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.

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:

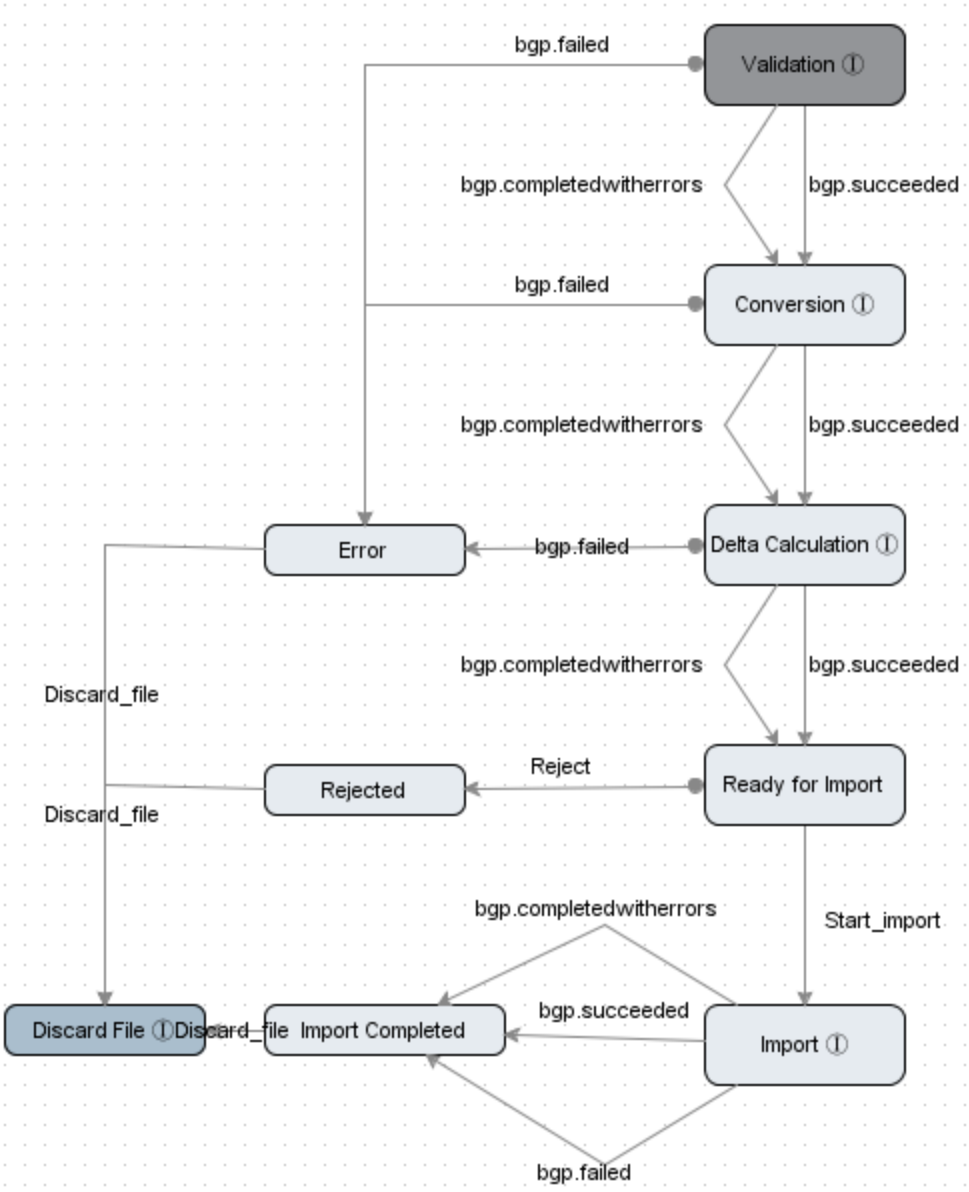



Each import has an associated workflow and all proceed through the same states by default. However, it is intended that customers will expand on the existing states and actions to add their own validations, reporting, and additional processing as needed.

**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

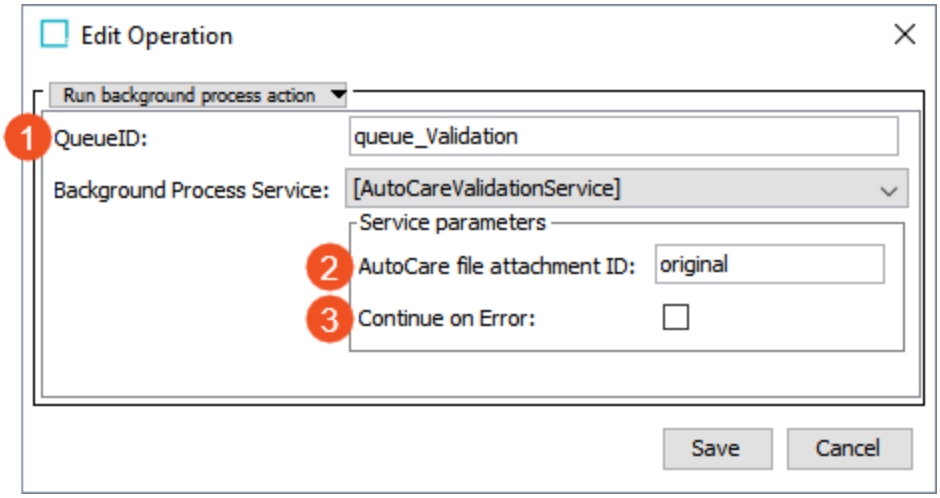
The validation state includes one business action (**Run background process action**) by default, which runs the validation service as a background process. Each standard has its own validation service.

**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.




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
```

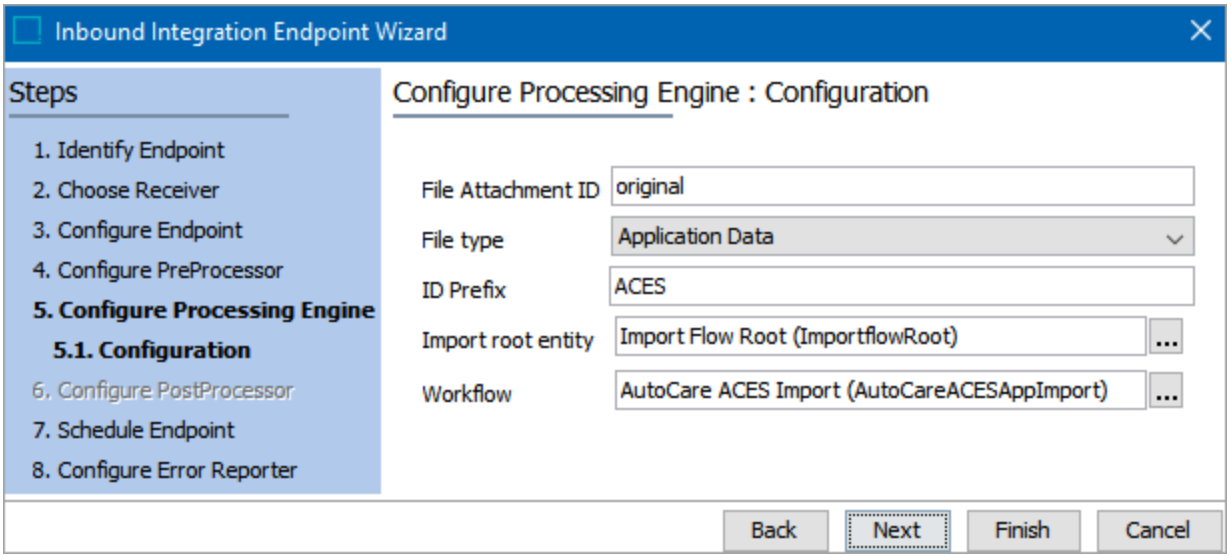
Typically, reference files do not need to have parallel processes because they are only updated once a month, and it is important to have certain reference files imported before others. Therefore, running them in parallel is not ideal.

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).



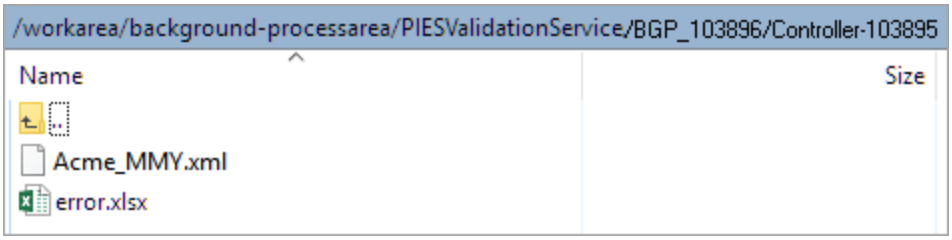

3. **Continue on Error:** Specifies whether or not the file can continue to process after encountering some 'allowable' data errors. For additional information and considerations relevant for this setting, refer to the **7. Determine Validation Error Handling for Each Import** topic within the **Automotive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

**Validation State AutoCare PIES Specific Parameter**

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

4. **Error file attachment ID:** Optional parameter available only for the PIES validation service. This additional parameter enables the creation of an error report if the imported PIES file encounters validation errors. When configured within the AutoCare PIES import workflow, the new validation service performs its validation tasks in the same way as 'AutoCareValidationService,' but when the new parameter 'Error file attachment ID' is configured, users can define a value to identify the error file. The value entered in the 'Error file attachment ID' parameter will be part of the Excel file error report's file name.

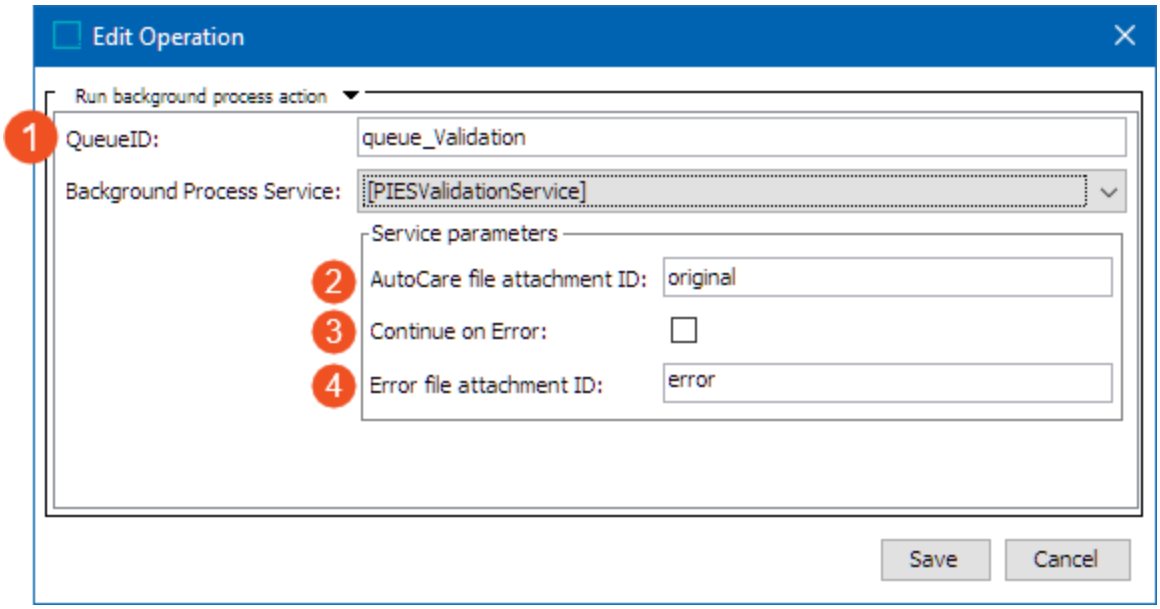
   When a PIES file completes validation with errors, the validation service also creates an additional Excel error file in the BGP folder. In the screenshot below, **error.xlsx** is the error report file placed in the Controller-103895 folder. Users can retrieve the Excel error file by accessing the folder on the application server.




   This error file can also be accessed in the Web UI's AutoCare PIES Import Controller screen, which is used to display the details of the PIES file import. To access the file in the Web UI, users need to add the new

**Importflow Attachment** table header component to the AutoCare PIES Import Controller screen. For more information, refer to the **Adding Additional Headers in Controller Screen** topic within the **Automotive Quick Start Guide**.

The PIES validation service (PIESValidationService) is shown as an example below.



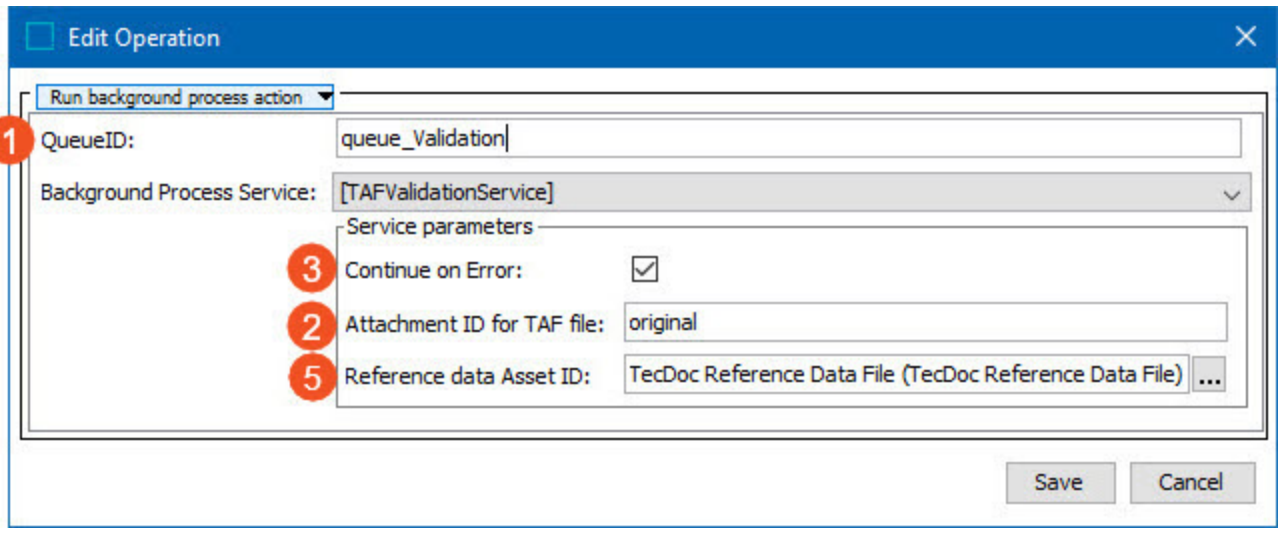

**Validation State TecDoc Specific Parameter**

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

5. **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.

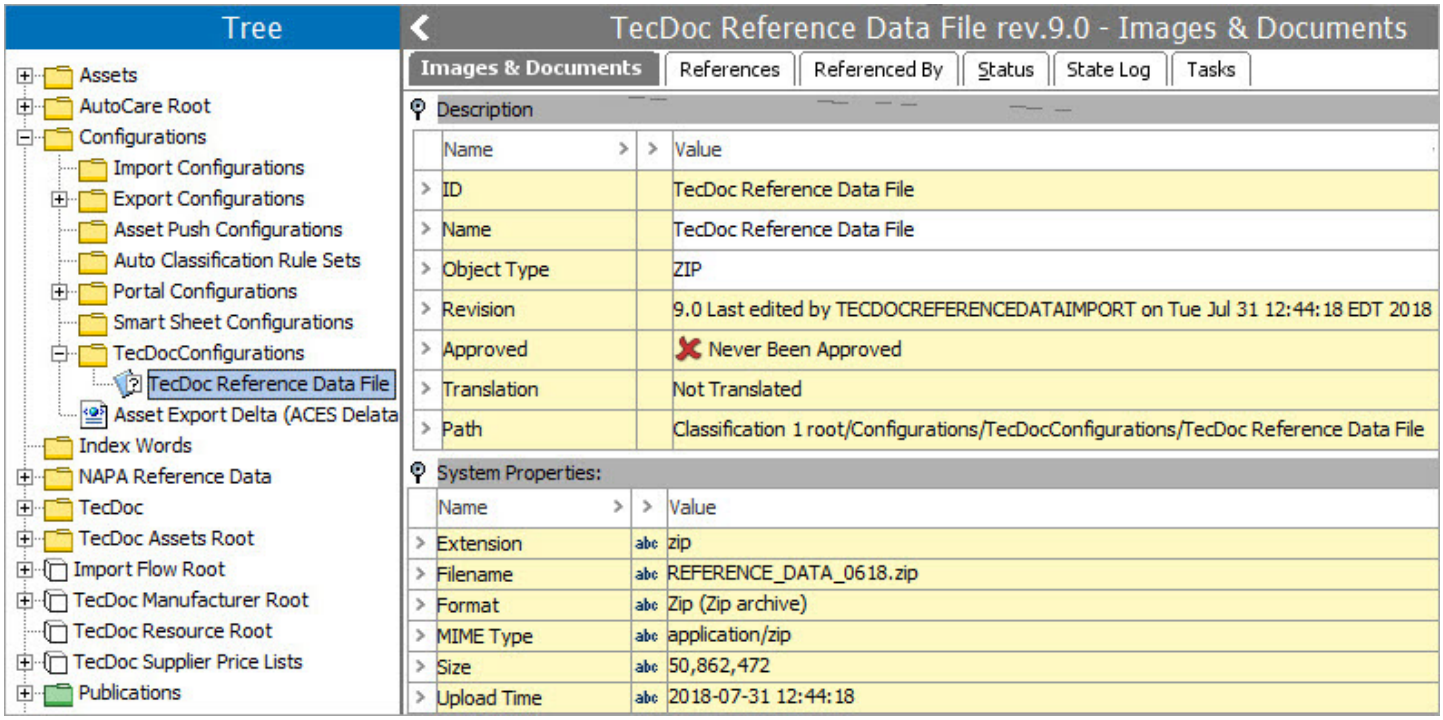
**Important:** When importing a Supplier Data file, the Supplier Data file is validated against the Reference Data file stored in the selected 'Reference data Asset ID.'

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




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

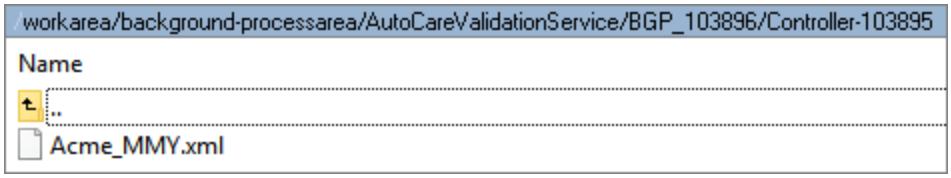
In the example below, the TecDoc Reference Data File is displayed below the TecDocConfigurations classification folder.



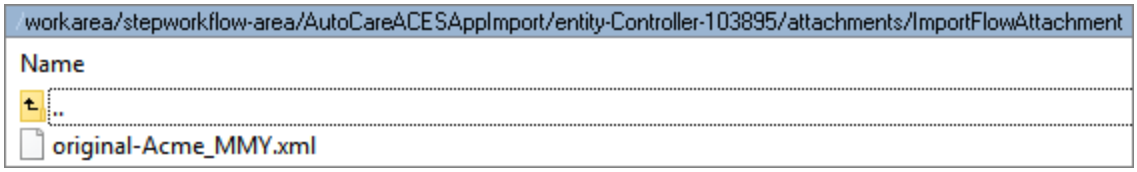

**Validation Service Function Details**

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.




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.




**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.

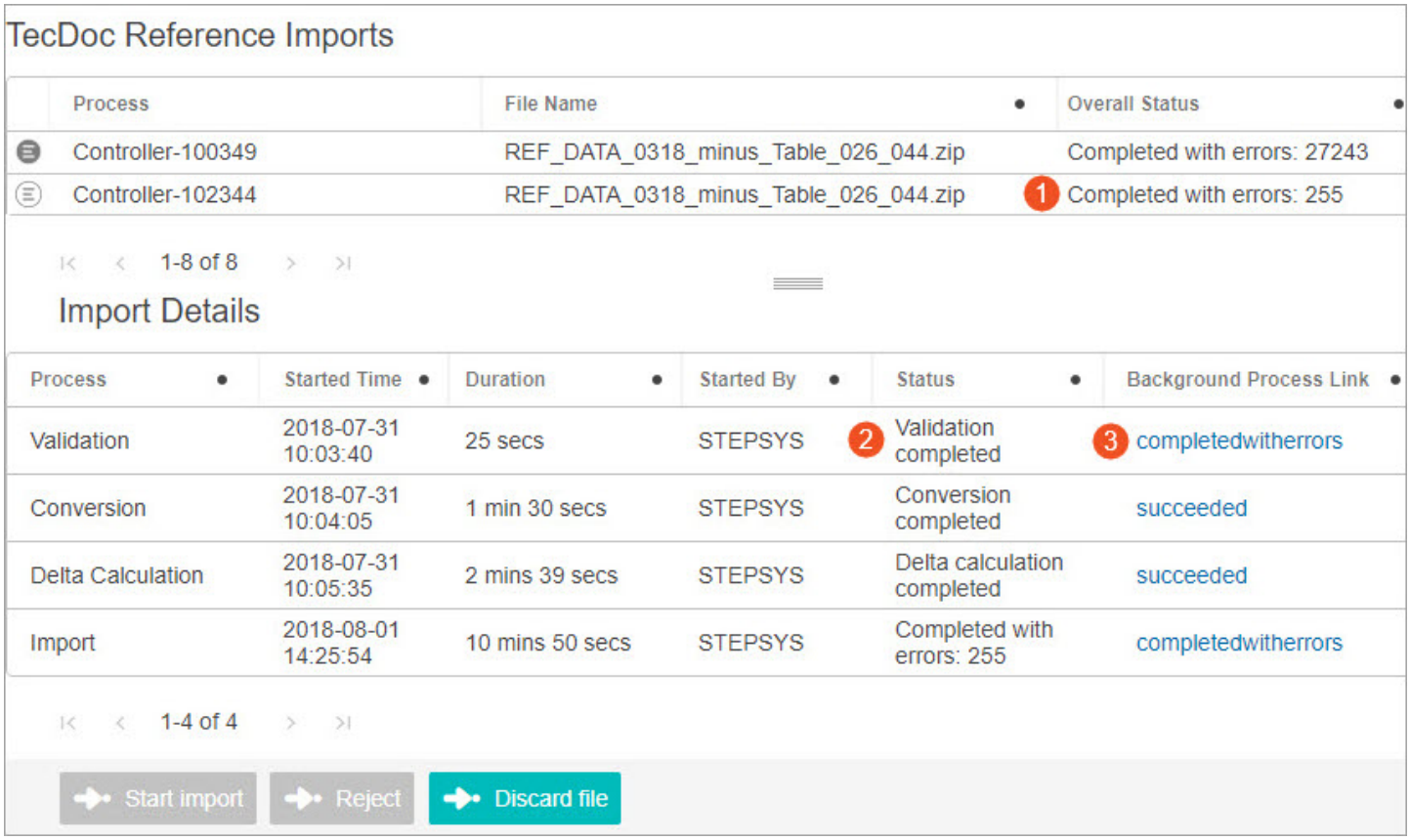
**Important:** Though an import can have a status of 'Validation failed,' the controller object in the workflow can still be sent to the next step of the workflow if the 'Continue on Error' parameter is enabled for the workflow. When this occurs, the validation issues are written to the execution report, but all valid data is converted and made available for the import process. For more information, refer to 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.




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.

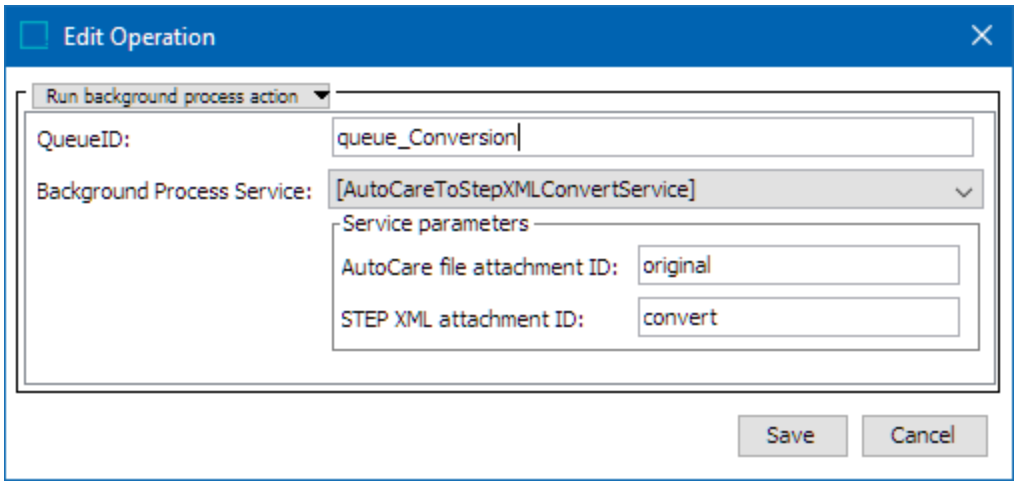
By default, the conversion state includes one business action (**Run background process action**), which runs the conversion service as a background process. Each standard has its own conversion service and some parameters are specific to the standards. Each standard's Conversion state parameters are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

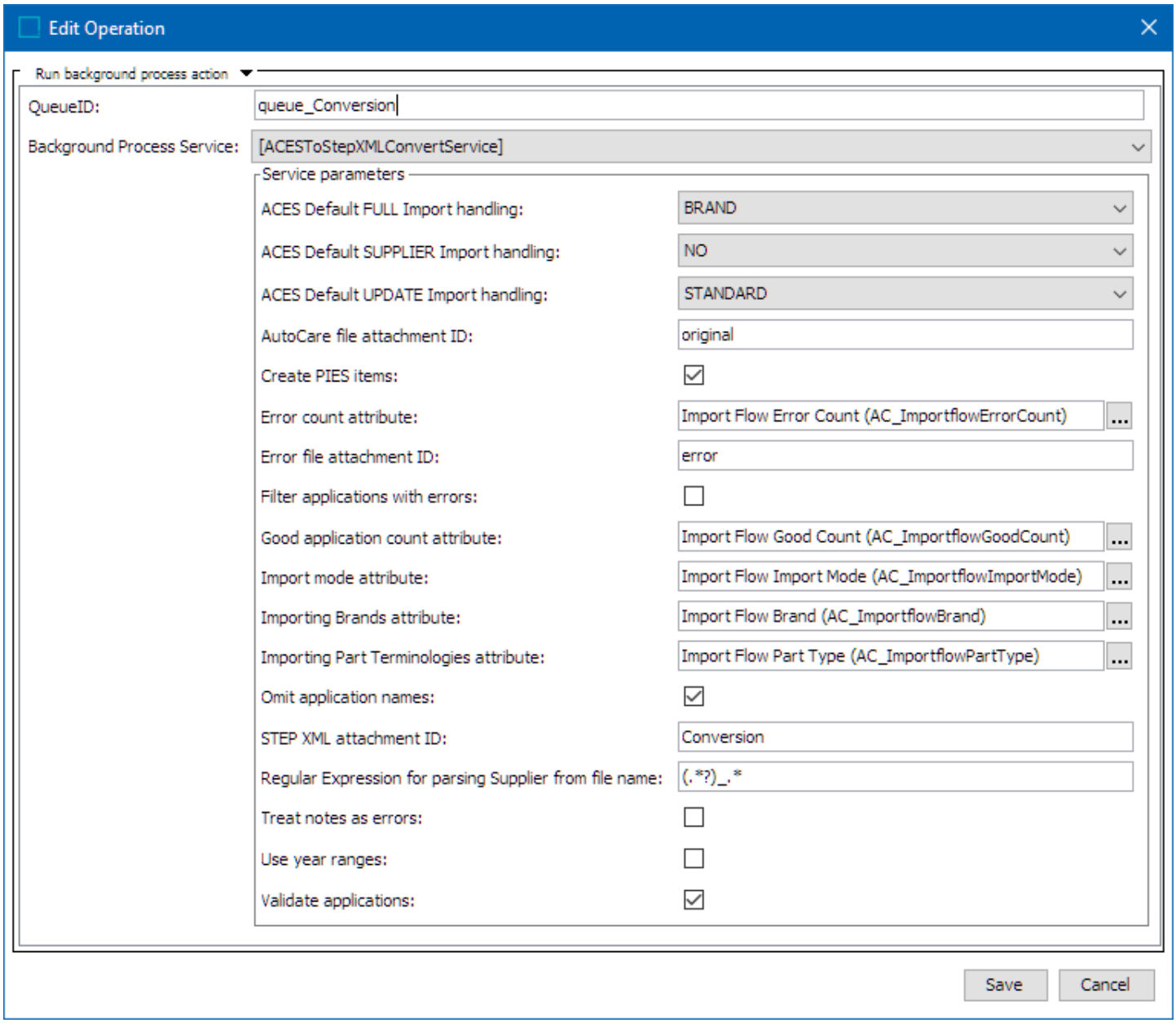



- **QueueID:** Required parameter for all background process actions that specifies the queue in which the background process should run, which defaults to 'queue_Conversion'. The BGP queue can be configured to control which processes run on which server and how many can run at the same time on each server.
- **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, refer to 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.




- **QueueID:** Required parameter for all background process actions that specifies the queue in which the background process should run, which defaults to 'queue_Conversion'. The BGP queue can be configured to control which processes run on which server and how many can run at the same time on each server.

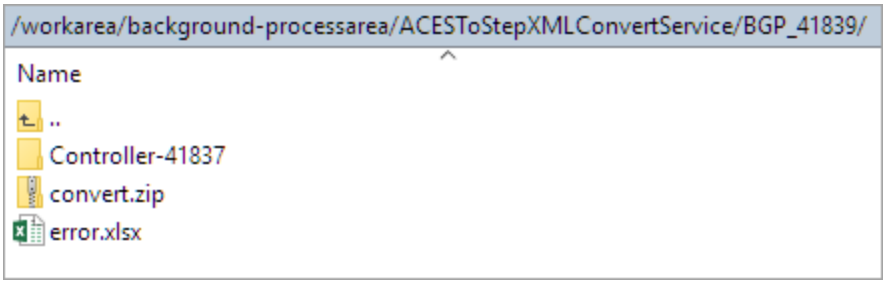
- **Background Process Service:** Required parameter for all background process actions that specifies the background process, which defaults to '[ACESToStepXMLConvertService]'.
- **ACES Default FULL Import handling:** By default, BRAND is selected. Defines how files with a SubmissionType=FULL are handled. For more information, refer to 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, refer to 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, refer to 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, refer to the **Validation State** topic.
- **Create PIES items:** Optional parameter used to import applications when the associated PIES Item does not exist in the system. If this parameter is selected, importing the ACES file will create the Part Terminology product hierarchy (PIES Brand / PIES PCdb Category / PIES PCdb Sub Category / PIES PCdb Part Terminology) and missing PIES Item, and applications will be created below the PIES Item that is provided as the <Part> in the ACES file. If this parameter is deselected and the PIES Item does not exist in the system, then the ACES import will not be able to create any applications because of the missing PIES Item.
- **Error count attribute:** Optional parameter that keeps track of the count for bad application records in the imported ACES file. Running Easy Setup will create and populate this parameter with the Import Flow Error Count (AC_ImportflowErrorCount) attribute.

  If users would like to create this attribute manually, then the attribute should be defined with:

  - Type = Description
  - Externally Maintained = No
  - Validation Base Type = Number
  - Valid for Entity Types = Import Flow Controller Type

- **Error file attachment ID:** An optional parameter that allows users to define an ID for the filename of the ACES Excel error report. If the 'Validate applications' option is selected, then any errors that are encountered are written to the report (e.g. vehicle configuration errors or wrong part type). This same ID will be used as the Attachment ID in the Importflow Attachment table header component for users to download the error report. For more information, refer to the **Adding Additional Headers in Controller Screen** topic within the **Automotive Quick Start Guide**.

  The error report will be added to the background process. The screenshot below shows an example report named error.xlsx that is placed in the BGP_41839 folder.




- **Filter applications with errors:** This optional parameter determines if application records need to be filtered out of the import file or not when validation errors are encountered. If the 'Filter applications with errors' and the 'Validate applications' parameters are selected, then any application record that encounters a validation error within the Conversion state will be filtered out and will not be created in the system on import. The applications that encounter validation errors will be written to the error report.

  If the 'Filter applications with errors' parameter is not selected and the 'Validate applications' parameter is selected, then any application that encounters a validation error within the Conversion state will be written to the error report, and will also be created in the system on import.

- **Good application count attribute:** Optional parameter that allows users to keep track of the count of application records that did not encounter errors in the ACES file. Running Easy Setup will create and populate this parameter with the Import Flow Good Count (AC_ImportflowGoodCount) attribute.

  If users would like to create this attribute manually, then the attribute should be defined with:

  - Type = Description
  - Externally Maintained = No
  - Validation Base Type = Number
  - Valid for Entity Types = Import Flow Controller Type

- **Import mode attribute:** This optional parameter allows users to define the attribute that will store the value for the Submission Type (FULL or UPDATE).

  If users would like to create this attribute manually, then the attribute should be defined with:

  - Type = Description
  - Externally Maintained = No
  - Validation Base Type = Text
  - Valid for Entity Types = Import Flow Controller Type

- **Importing Brands attribute:** This optional parameter allows users to define the attribute that will store the Name of the BrandAAIAID that is used in the ACES import file. The Name is retrieved by looking up the BrandAAIAID in the Brand table.

  Running Easy Setup will create and populate this parameter with the Import Flow Brand (AC_ ImportflowBrand) attribute.

  If users would like to create this attribute manually, then the attribute should be defined with:

- Type = Description
- Externally Maintained = No
- Validation Base Type = Text
- Multi Valued = Yes
- Valid for Entity Types = Import Flow Controller Type

- **Importing Part Terminologies attribute:** This optional parameter allows users to define the attribute that will store the Name of the Part Type that is used in the ACES import file. The Name is retrieved by looking up the PartType ID in the PCdb table.

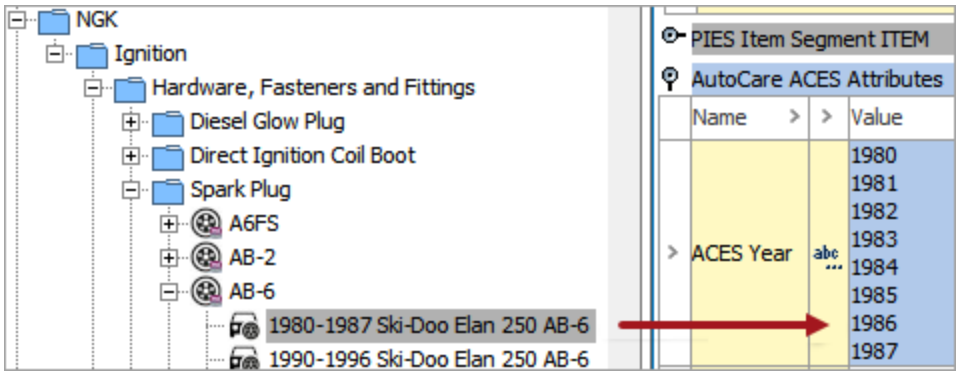
  Running Easy Setup will create and populate this parameter with the Import Flow Part Type (AC_ ImportflowPartType) attribute.

  If users would like to create this attribute manually, then the attribute should be defined with:

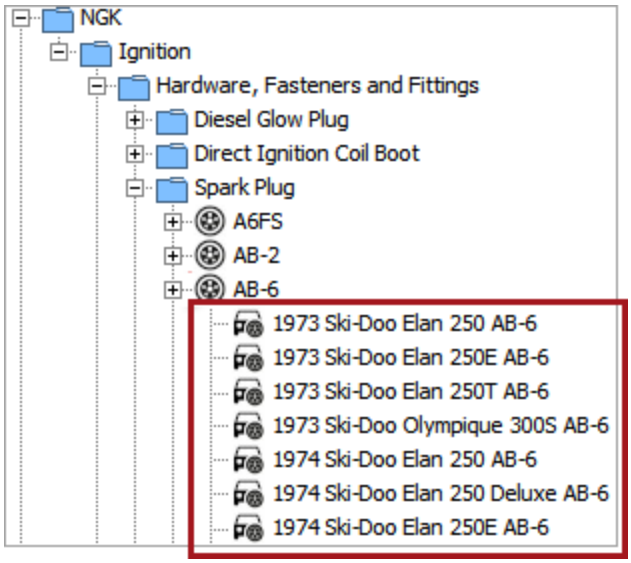
  - Type = Description
  - Externally Maintained = No
  - Validation Base Type = Text
  - Multi Valued = Yes
  - Valid for Entity Types = Import Flow Controller Type

- **Omit application names:** This optional parameter is specific for the Auto Care standard and is deselected by default. When the ACES Importer creates application objects, it sets the application STEP Name to be the same value as the Part Number. Alternatively, users have the option to change the application STEP Name to something other than the Part Number (e.g. 2010 Acura TSX) by creating a business rule and adding it to the Import action in the ACES Import workflow, For more information, refer to the **Specify an Application Naming Convention** topic within the **Quick Start Setup for Admins** guide. However, an issue is encountered the next time that an ACES file containing those same applications are 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 is not altered and should always match the part number, then this parameter should remain deselected.
  - If the application name should be changed by a business rule to something other than just the part number, then this parameter must be selected so that the application names will be ignored during the conversion process so that the record will not be included in the delta file as a changed record.
- **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:** This optional parameter uses 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.

- **Treat notes as errors:** This parameter allows users to treat application records with application notes as errors. With this parameter checked, when an application record with application notes is found, an error stating '*Application has note(s), that is not allowed on application id=<ID>, partNumber=<Part Number>*' is written to the execution report of the conversion process. If the 'Treat notes as errors' parameter is not checked, importing ACES applications with application notes will pass the conversion process but without writing an error in the execution report.
- **Use year ranges:** If this parameter is selected, then the applications will get created in the system with a year range that is stored in the 'AC_ACESYears' attribute. The created applications will be linked to the vehicle Model instead of the Base Vehicle. If there is a gap within the year range, then more than one application will get created. This can be used to compress the number of applications that gets created in the system to reduce the application object count like in the example below.




- If this parameter is not selected, then applications will be imported as before with individual applications created for each year and linked to the Base Vehicle like in the example below.



**Important:** In order to validate vehicle configurations during ACES imports, the system only allows one validation path for Auto Care attributes to retrieve configuration data deriving either from the Model or the Base Vehicle objects. Because of this reason, the system supports only one of these data models (applications linked to Base Vehicles (individual), or applications linked to Model (year range)), not both. So, It is recommended to use one method or the other, not combined.

**Note:** The import of applications with or without selecting 'Use year ranges' parameter does not impact the ACES exporter. The 'Export in Make/Year format' parameter that is available in the ACES Export Manager will function separately.

- **Validate applications:** This optional parameter will validate application records with invalid vehicle configurations and invalid part type conditions. With the 'Validate application' parameter checked, when an application record is found that has invalid condition(s), an error is written to the execution report of the conversion process. The error contains the application ID from the ACES file, as well as name of the objects that it failed on. If the 'Validate application' parameter is not checked, importing ACES applications that have invalid configurations will pass the conversion process without writing an error in the execution report. The benefit of this is that users will be able to detect invalid applications prior to having to import the file.

**Important:** If applications with invalid conditions should not get created in STEP, then this parameter along with the 'Filter applications with errors' parameter must be selected. The benefit of this is that users will be able to detect invalid applications prior to having to import the file.

Some examples of applications that will throw an error are: Missing Part Type; missing Vehicle; or vehicle configurations such as Sub Model, Engine Base, Drive Type, etc. that are not valid for the vehicle.

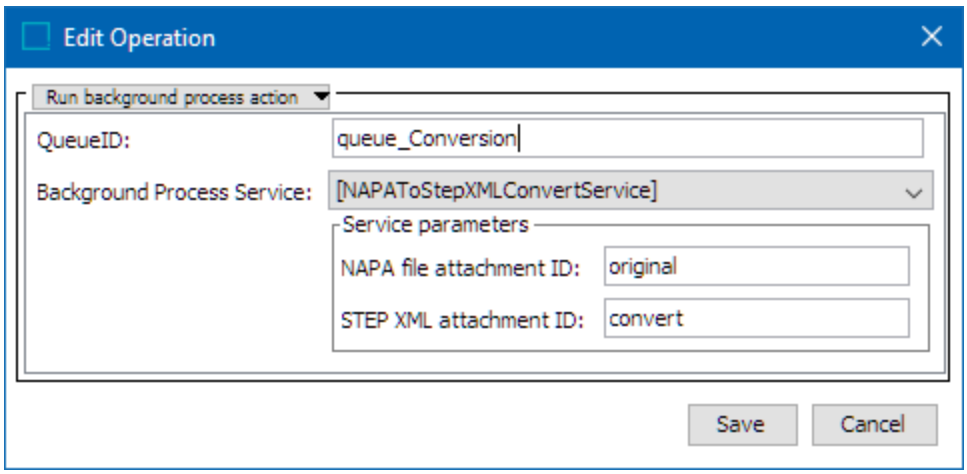
**Note:** Since the validation is done in the ACES converter, the Conversion process will be slower if the 'Validate application' parameter is selected.

#### 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]

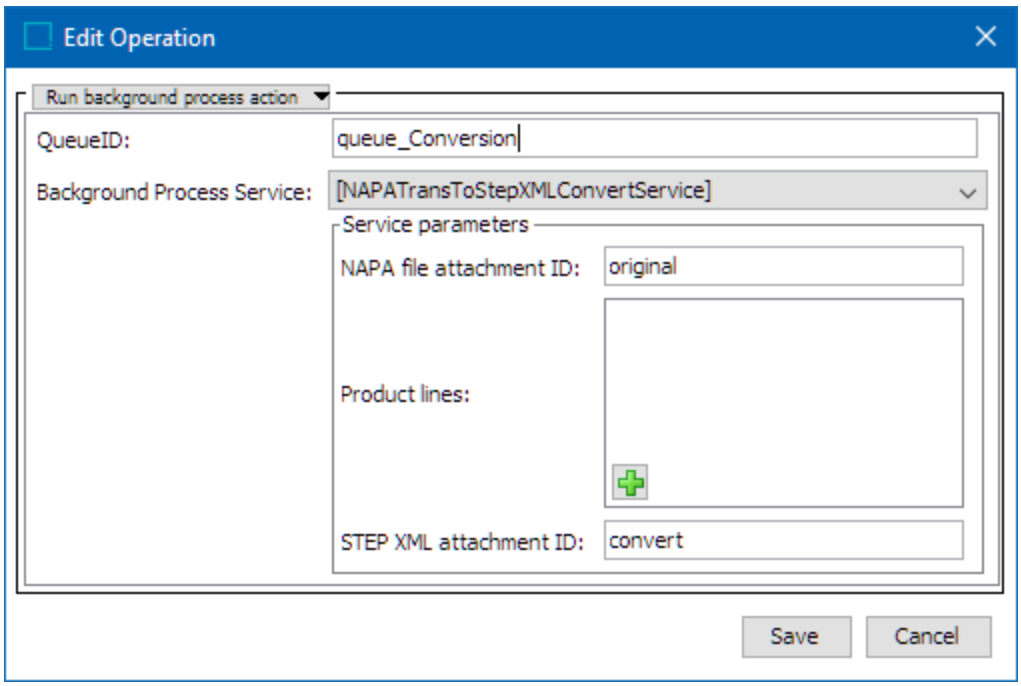
Each of the parameters are explained in detail below.




- **QueueID:** Required parameter for all background process actions that specifies the queue in which the background process should run, which defaults to 'queue_Conversion'. The BGP queue can be configured to control which processes run on which server and how many can run at the same time on each server.
- **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, refer to 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.



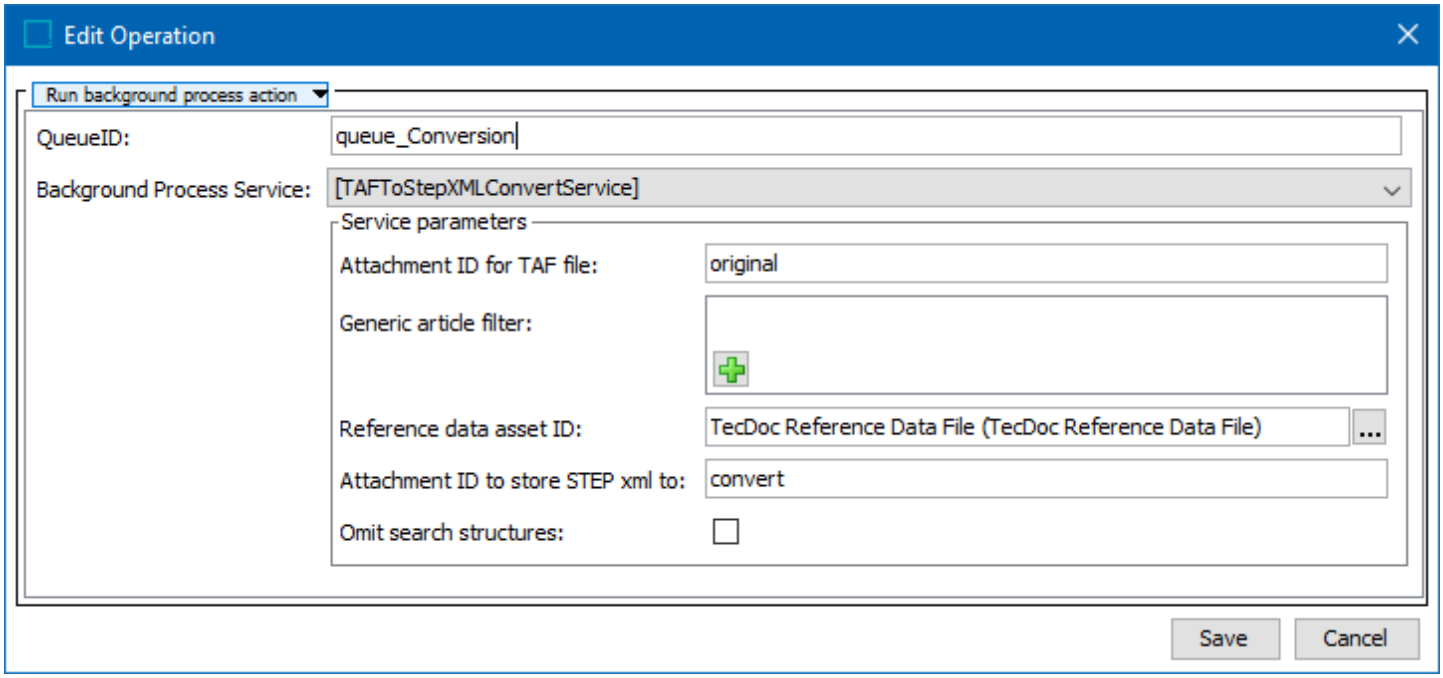

- **QueueID:** Required parameter for all background process actions that specifies the queue in which the background process should run, which defaults to 'queue_Conversion'. The BGP queue can be configured to control which processes run on which server and how many can run at the same time on each server.
- **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, refer to 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 [TAFToStepXMLConvertService] conversion services is offered.

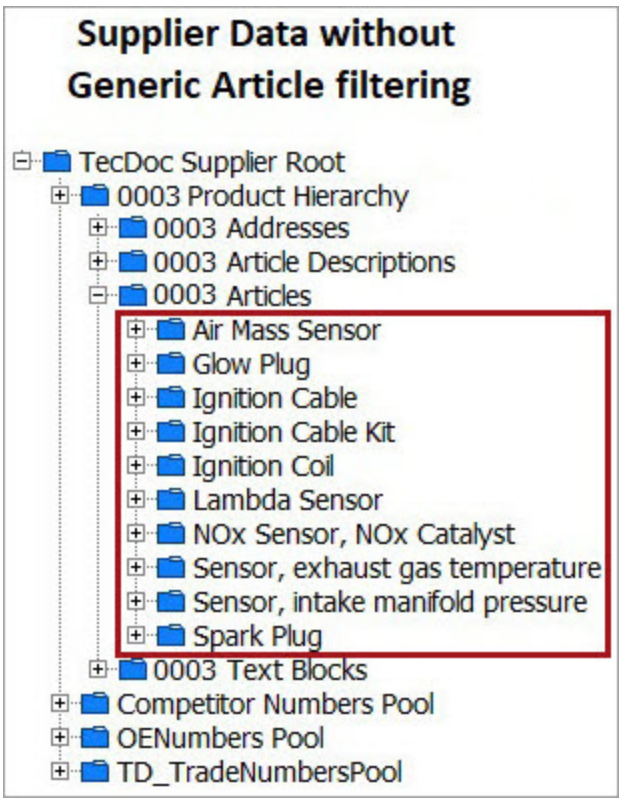
## [TAFToStepXMLConvertService]

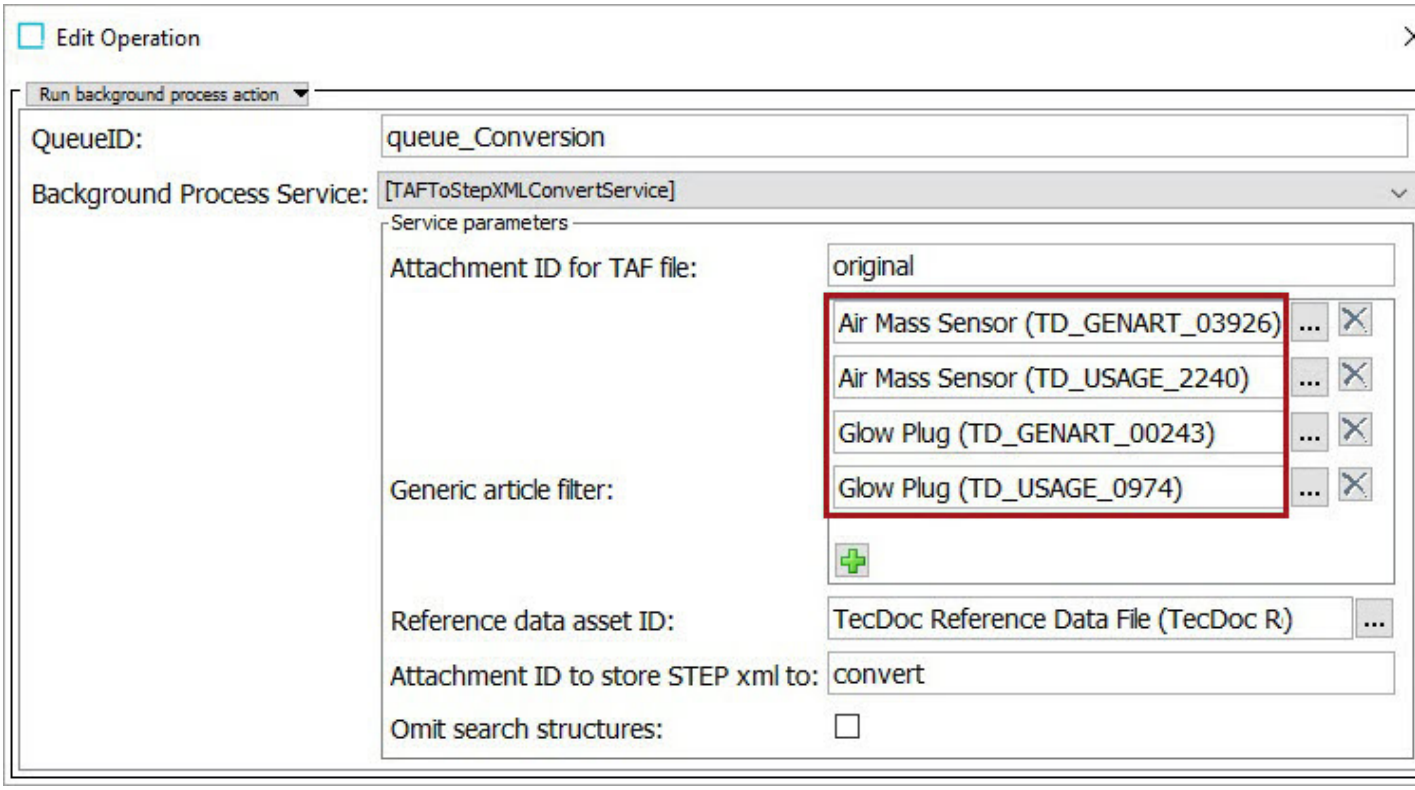
Each of the parameters are explained in detail below.

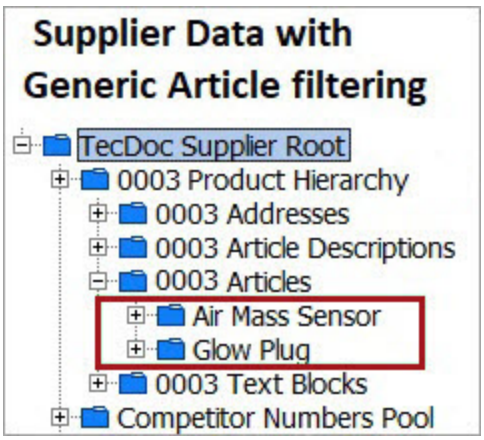



- **QueueID:** Required parameter for all background process actions that specifies the queue in which the background process should run, which defaults to 'queue_Conversion'. The BGP queue can be configured to control which processes run on which server and how many can run at the same time on each server.
- **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, refer to 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:
queue_Conversion
Background Process Service:
[TAFToStepXMLConvertService]
Service parameters
Attachment ID for TAF file:
original
Air Mass Sensor (TD_GENART_03926)
Air Mass Sensor (TD_USAGE_2240)
Glow Plug (TD_GENART_00243)
Generic article filter:
Glow Plug (TD_USAGE_0974)
Reference data asset ID:
TecDoc Reference Data File (TecDoc R)
Attachment ID to store STEP xml to:
convert
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.

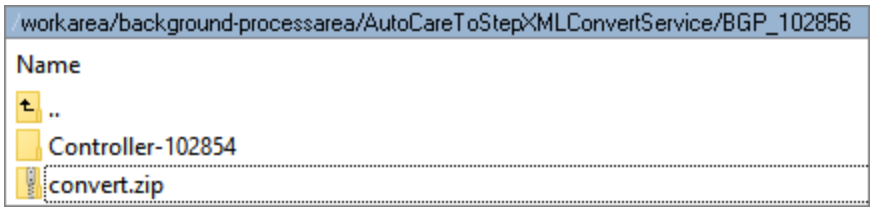
- **Reference data asset ID:** Required parameter used in all TecDoc conversion services that functions the same as described for the validation service (providing a link between the file being imported and the reference data asset it should be stored in or validated against). For more information, refer to 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.
- **Use Criteria Blocks:**Optional parameter specific to the TecDoc standard. This option handles application records with varying criteria. Instead of creating multiple application records for each criterion, this parameter allows the system to create one application record linked to multiple criteria blocks. When enabled, the TecDoc Supplier Data importer creates a criteria block for each criterion as a child object under the application record instead of creating multiple application records. When disabled, the TecDoc Supplier Data importer continues to import as before, where the importer considers each criterion a discrete application record.

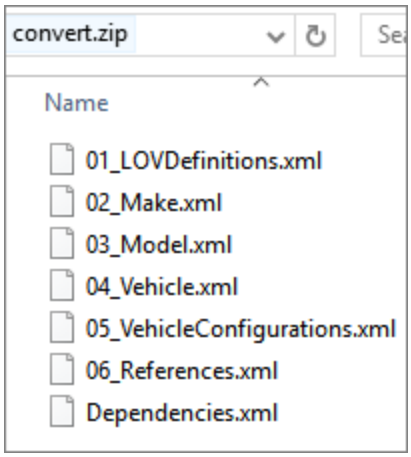
#### Conversion State Function Details

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 found in the validation service. The conversion

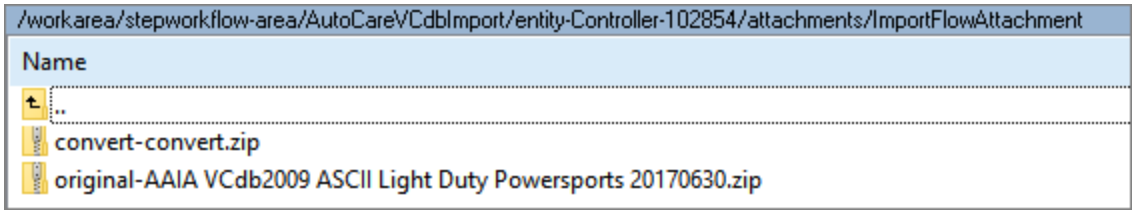
service also writes an additional file, which is the output of the conversion process, (e.g., a zipped file with the name specified in the STEPXML attachment ID parameter). This zipped file contains sub-files in STEPXML format that are submitted to the delta calculation service for further processing. 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.

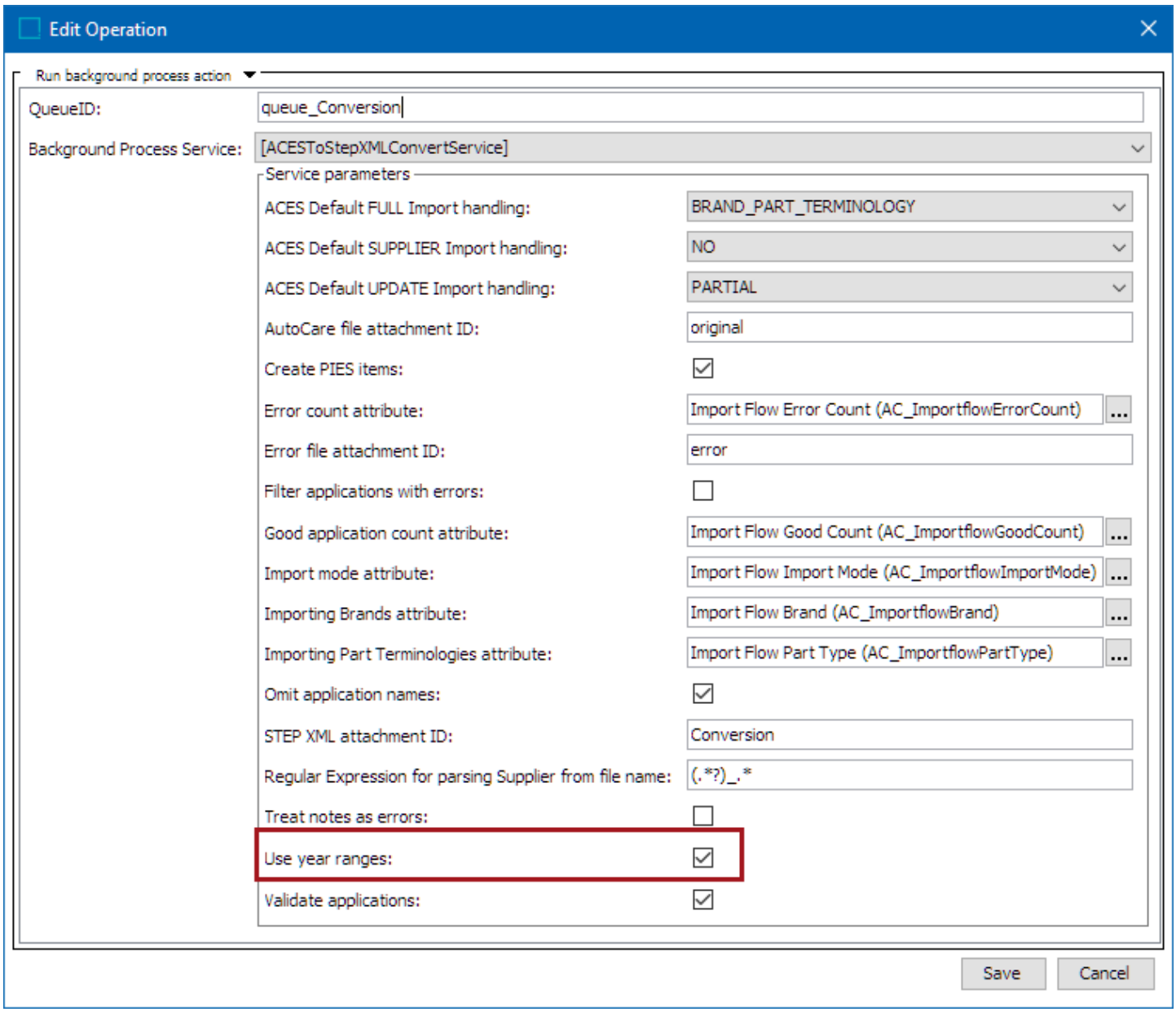
**Conversion State Results**

The following are possible conversion state results:

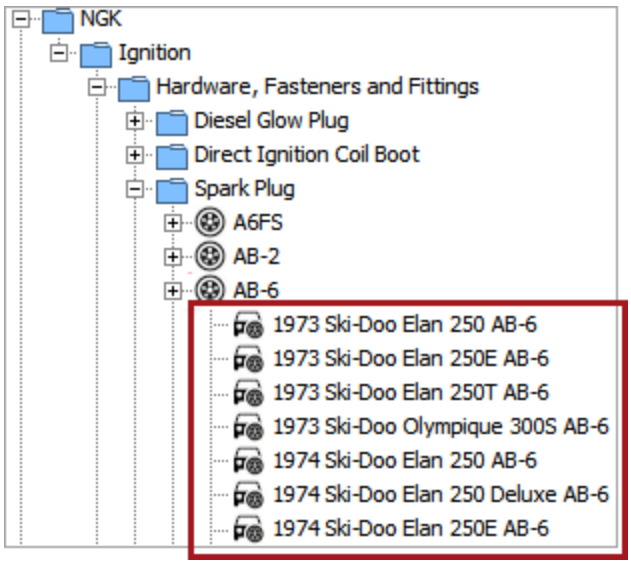
- **Failed:** If the import file fails conversion for some reason, the controller object in the workflow is sent to the Error state via the bgp.failed transition. This transition includes one default business rule which is used to populate the overall status of the controller entity (the overall status 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.

#### Configuring Application Year Range Functionality in Workbench

The ACES importer Conversion state contains the ‘Use year ranges’ parameter.




If the 'Use year ranges' parameter is not selected, then applications will get created individually, one application per Base Vehicle year like in the example below.




If the 'Use year ranges' parameter is selected:

- Applications will get created with a year range that is stored in the ACES Year attribute.
- The application is linked to the Model instead of the Base Vehicle.
- If there is a gap within the year range, then more than one application will get created.

This option will reduce the amount of application objects that are created in STEP.

**Important:** In order to validate vehicle configurations during ACES imports, the system only allows one validation path for Auto Care attributes to retrieve configuration data deriving either from the Model or the Base Vehicle objects. Because of this reason, the system supports only one of these data models (applications linked to Base Vehicles (individual), or applications linked to Model (year range)), not both. So, It is recommended to use one method or the other, not combined.

In order to use the 'Use year ranges' feature, the following manual setup needs to be done in the workbench before importing any ACES files:

1. Select the 'Use year ranges' parameter in the ACES Conversion state
2. Create a new attribute below attribute group AutoCare ACES Attributes (ID = AC_ACESAttributes) with the following values:
   - ID = AC_ACESYears
   - Name = ACES Years
   - Validation Base Type = Number
   - Multi Valued = Yes
   - Type = Specification
   - Valid for Product Types = ACES Application

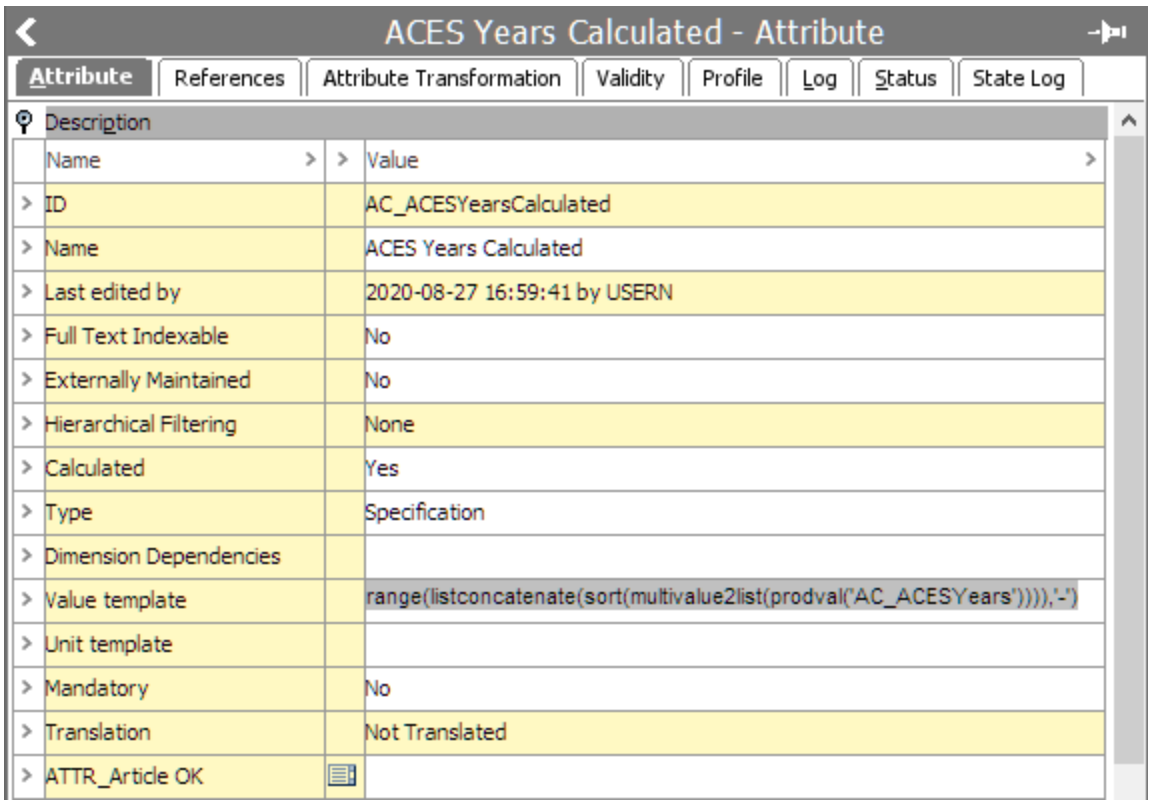
- Link to AC_PIESProducts Hierarchy
- Automotive Validation Path = child.attribute[id:'AC_VCdbYear']

3. Create a new Classification Reference with the following values:
   - ID = AC_ACESApplicationToModel
   - Name = ACES Application To Model
   - Valid Source Type = AC_ACESApplication
   - Valid Target Type = AC_Model
   - Allow multiple references = No
4. In the Automotive - Application Model component do the following
   - Add AC_Model object type to Assembly / Vehicle aspect
   - Remove AC_BaseVehicle object type from Assembly / Vehicle
   - Add AC_ACESApplicationToModel classification reference to Application to assembly aspect
   - Remove AC_ACESApplicationToBaseVehicle classification reference from Application to assembly
5. Add 'child.' as a prefix to all attributes that have a validation path within attribute group AutoCare ACES Attributes (ID = AC_ACESAttributes). For example, child.child.reference[type:'AC_VehicleToBodyStyleConfig'].attribute[id:'AC_VCdbBodyType']
6. In order for the applications to be searchable within the Application Manager Screen, the Vehicle Type Search Panel need to have the Vehicle Object Types changed from AC_BaseVehicle to AC_Model. Furthermore, all Options Group Search Box configurations need to have the validation paths updated to add the "child." prefix.

### Configuring Application Name to Include Year Range

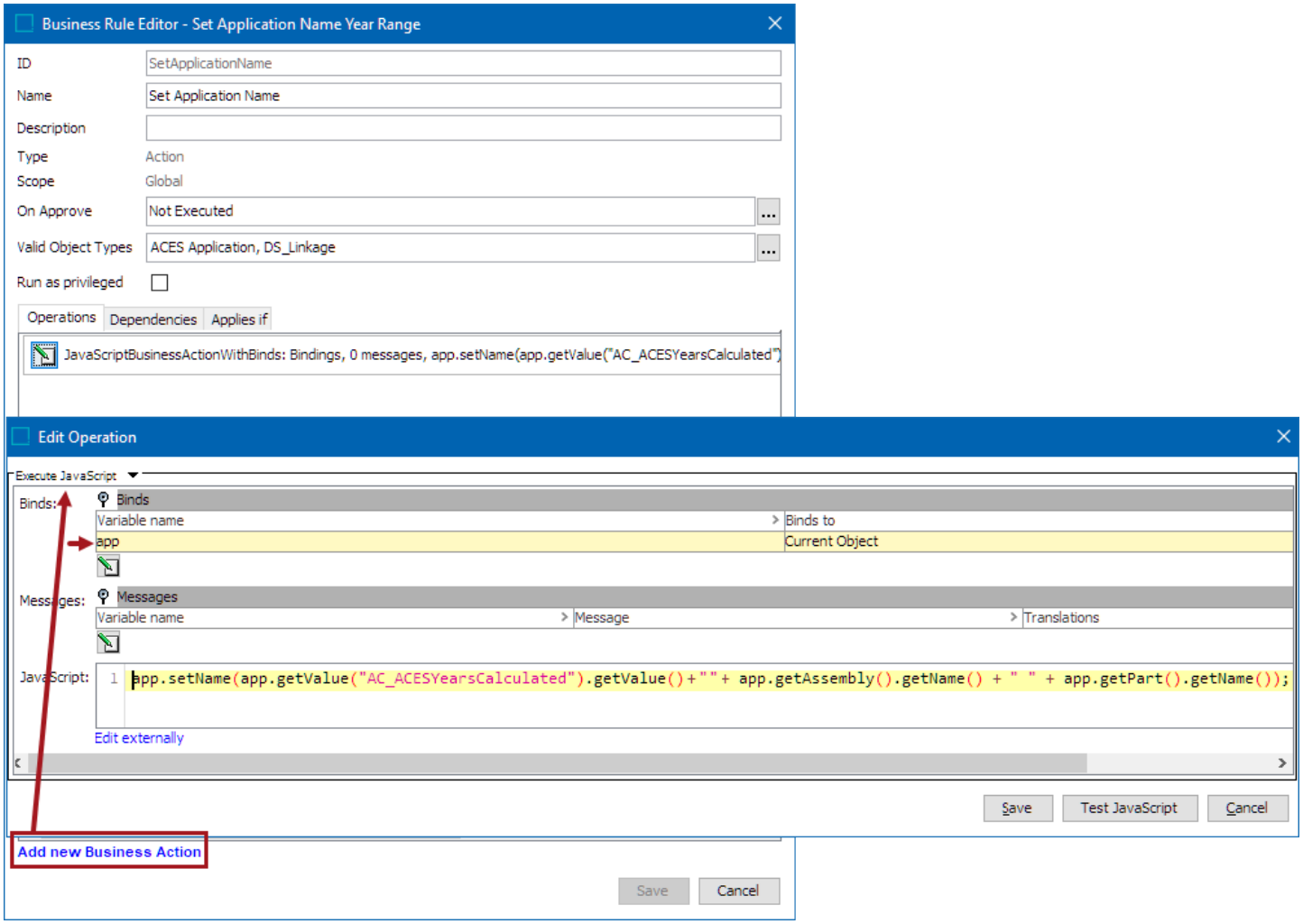
To create applications with the year range in the Name, the following manual setup must be done in workbench.

1. Create a new calculated attribute below attribute group AutoCare ACES Attributes (ID = AC_ACESAttributes) with the following values:
   - ID = AC_ACESYearsCalculated
   - Name = ACES Years Calculated
   - Validation Base Type = Number
   - Multi Valued = No
   - Calculated = Yes
   - Type = Specification
   - Value template = range(listconcatenate(sort(multivalue2list(prodval('AC_ACESYears')))),'-')
   - Valid for Product Types = ACES Application
   - Link to AC_PIESProducts Hierarchy

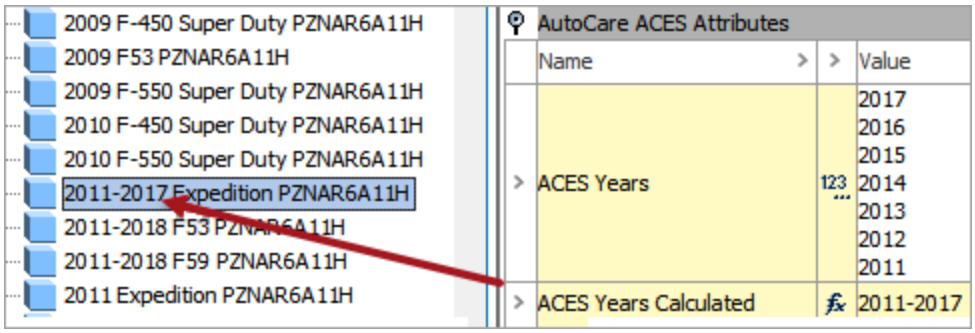



2. Create a Business Action using the following JavaScript:

```
app.setName(app.getValue("AC_ACESYearsCalculated").getValue() + " " +
app.getAssembly().getName() + " " + app.getPart().getName());
```




3. Add the Business Action to the Import Action parameter of the ACES workflow Import state.
4. After the ACES file is imported, the application Name will be in “[YearRange] [Vehicle Model] [Part Number]” format like in the screenshot below:




For more information on setting application names, refer to the **8. Specify an Application Naming Convention** topic within the **Automotive Quick Start Guide**.

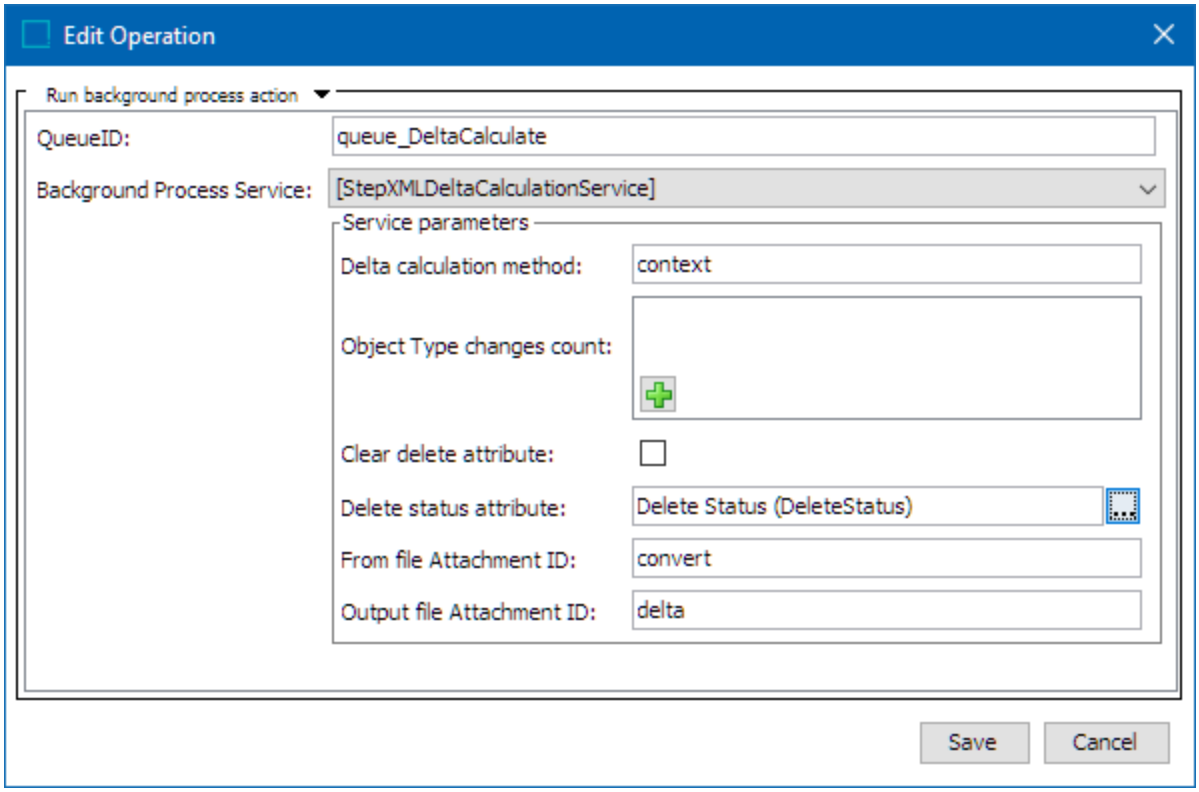
### Delta Calculation State

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.




- **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:** The BGP queue can be configured to control which processes run on which server and how many can run at the same time on each server.

- **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, refer to the **5. Update Delete Status Attribute and Delta Calculation Method in Import Workflows** section of the **Quick Start Setup for Admins**.

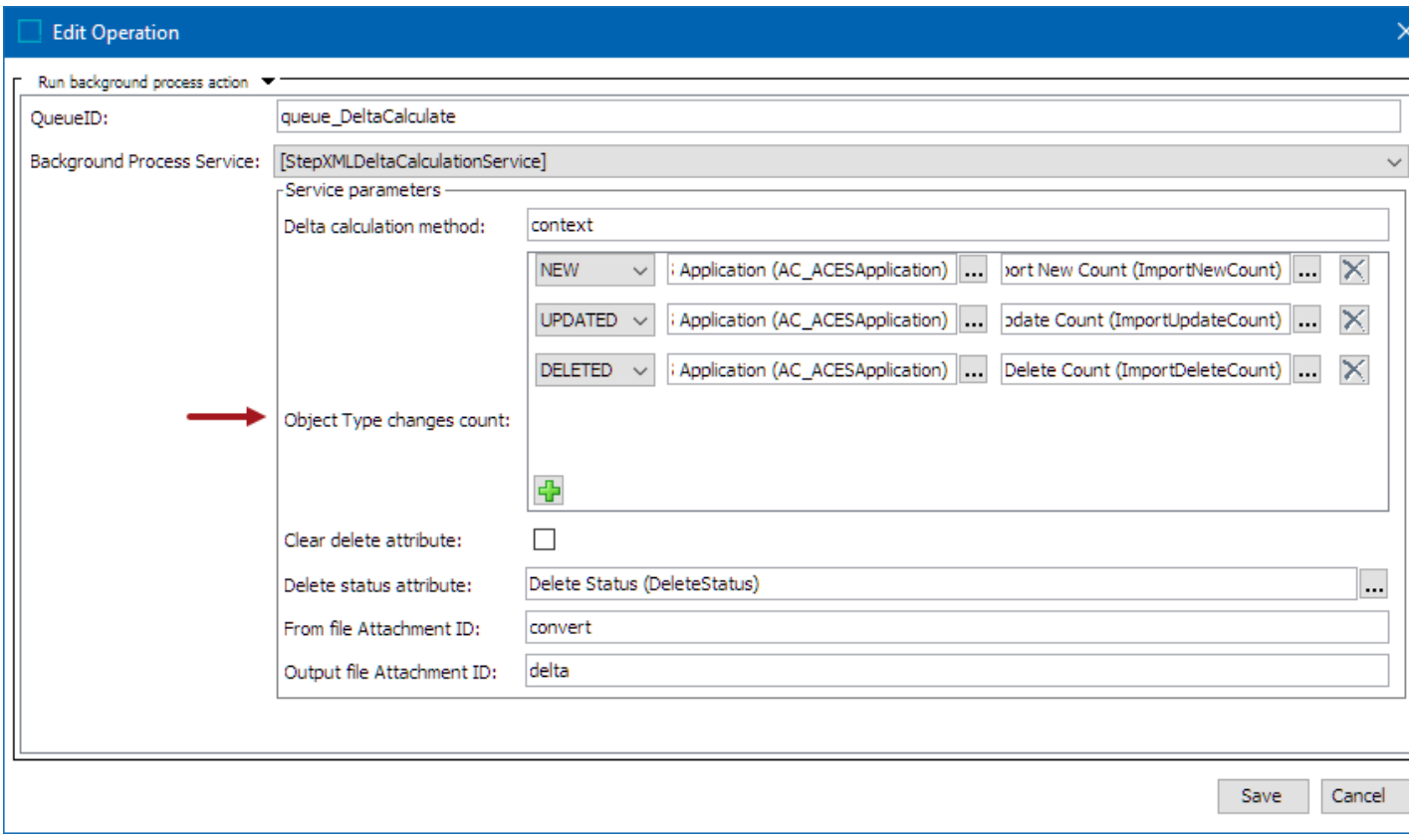
- **Object Type changes count:** This optional parameter is used to retrieve information on the total count of new, updated, and deleted objects for an Automotive data import. This functionality works only for the 'context' method of delta calculation. Users have to define the object type and attributes that will store the New, Updated, or Deleted count.

  In order to use these new features, the following manual setup needs to be done in workbench:

  1. Create Text Validation Base Type, Description attributes to hold the counts for the New, Deleted, and Updated objects. Make the attribute Valid for Entity Types = Import Flow Controller Type.
  2. For any import workflow that uses 'context' Delta calculation method (e.g., ACES Delta Calculate), go to the Delta Calculate business action within the workflow and edit the business rule. In the 'Object Type changes count' parameter do the following:

  - Click on the Add icon ( ), select 'NEW' from the drop-down list, select the object type that should be included in the new count (e.g., ACES Application), and add the 'New' attribute
  - Click on the Add icon ( ), select 'DELETED' from the drop-down list, select the object type that should be included in the delete count, and add the 'Deleted' attribute
  - Click on the Add icon ( ), select 'UPDATED' from the drop-down list, select the object type that should be included in the update count, and add the 'Updated' attribute

  Below is an example of the 'Object Type changes count' parameter that is configured to retrieve information on the total count of new, updated, and deleted objects for an ACES import.




- **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 object that is flagged for deletion, and the object is reintroduced in the later imports.

  For example, consider part number 'PIES Item A' which already has delete attribute value set as 'true' in the system. With the 'Clear delete attribute' parameter selected, importing 'PIES Item A' again via a new PIES file will clear the attribute value of 'true' from the delete flag for 'PIES Item A.'

- **Delete status attribute:** Optional parameter to select an attribute to store the value of 'true' for objects that should be flagged 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 is not included in the imported file.

  > **Note:** AutoCare ACES files using complete replacement concept and require a delete status attribute to be set.

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

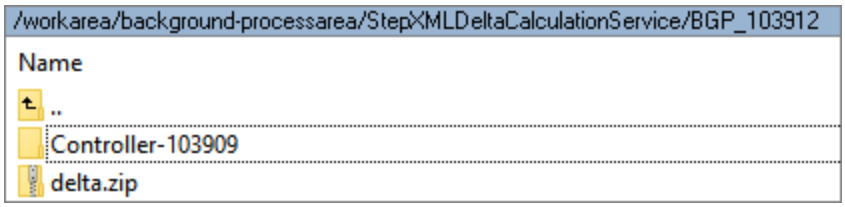
- **From file Attachment ID:** Required parameter that must be populated with the attachment ID from the prior state that should be used for the delta calculation service to act on. This defaults to 'convert' as that would be

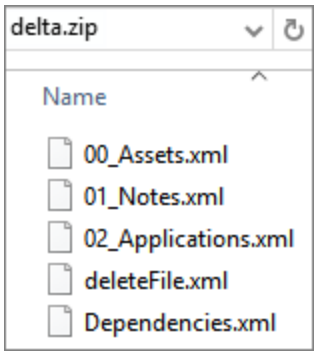
the correct attachment ID if no additional states and/or processing were added to the default workflow. If an additional workflow state or service had been added that generated some other output file that the delta calculation should act on, the parameter should be updated accordingly.

**Delta Calculation State Function Details**

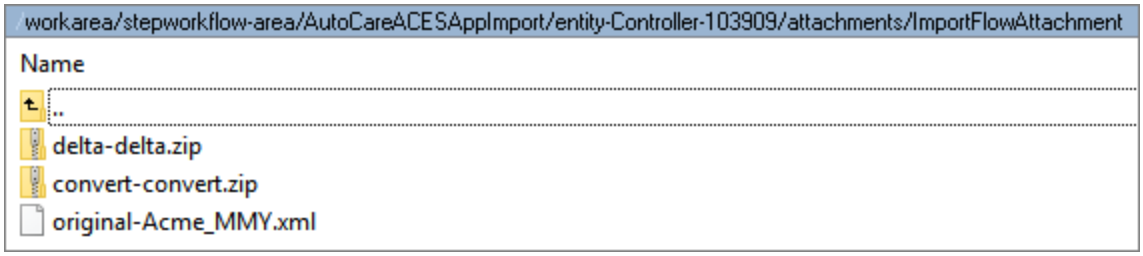
As the delta calculation service runs a background process (BGP), a corresponding BGP folder is created on the application server at /workarea/background-processarea/StepXMLDeltaCalculationService. The BGP folder contains a sub-folder whose name matches the STEP ID of the controller entity that is moving 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.

## Replacement Rules in PIES Importer

The PIES Importer uses the ReplacementRules tag to calculate and replace attribute values, data container values, and references populated for the PIES Items that are already existing in the system. The attribute groups, data containers, and references that are listed below are currently handled for replacement using replacement rules. This information can be found within the 09_Items.xml file of the Delta Calculation state.

```
<ReplacementRules>
    <Products>
      <ReplaceValues AttributeGroupID="AC_PIESItemSegmentITEM" CrossContext="N"/>
      <ReplaceValues AttributeGroupID="AC_PIESDescriptionSegmentDESC"
CrossContext="N"/>
      <ReplaceValues AttributeGroupID="AC_PIESExtendedProductInfoSegmentEXPI"
CrossContext="N"/>
      <ReplaceValues AttributeGroupID="AC_PAdbAttributes" CrossContext="N"/>
      <ReplaceValues AttributeGroupID="AC_PIESPackagingSegmentPACK"
CrossContext="N"/>
      <ReplaceDataContainers DataContainerTypeID="AC_PIESItem_Description"/>
      <ReplaceDataContainers DataContainerTypeID="AC_PIES_PRCE"/>
      <ReplaceCrossReferences ReferenceTypeID="AC_PIESInterchange"
CrossContext="Y"/>
      <ReplaceCrossReferences ReferenceTypeID="AC_PIESItemToKitComponentItem"
CrossContext="Y"/>
      <ReplaceClassificationReferences TypeID="AC_ProductToPartTerminology"
CrossContext="Y"/>
    </Products>
  </ReplacementRules>
```

For information about the ReplacementRules tag in STEPXML files, refer to the **ReplacementRules Tag in STEPXML** topic within the **STEPXML Format** section of **STEP Online Help**.

#### 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.

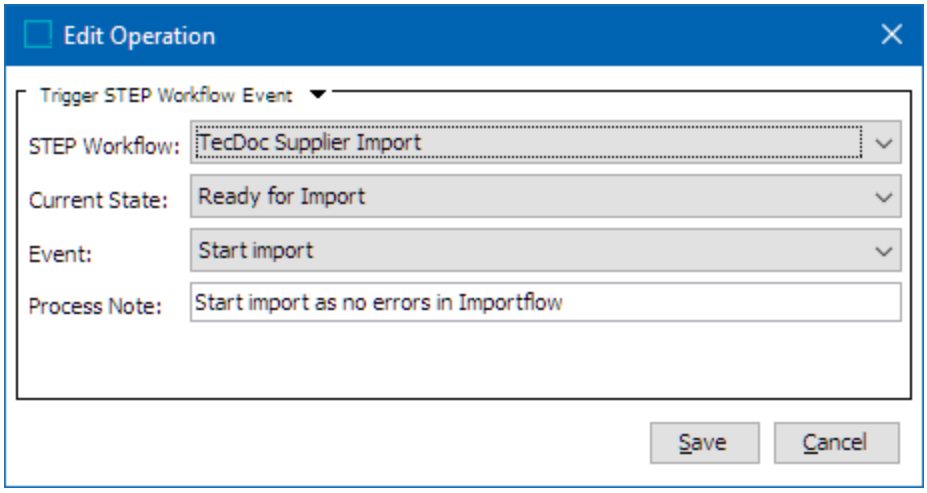
### Error State

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 files with warnings and/or errors. If the file has no warnings and/or errors in the validation, conversion, and delta calculation, then no user interaction will be required in this state, and the file will be moved to the Import state. 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 that has warnings and/or errors. 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.

The Ready for Import state includes one business action (**Trigger STEP Workflow Event)** and one business condition (**Check for Errors in Import Flow**) by default. The business condition is configured in the 'Applies If' tab of the On Entry tab. This serves as a precondition for applying the business action (Trigger STEP Workflow Event) configured in the 'Operations' tab of the On Entry tab. While the business condition (Check for Errors in Import Flow) does not hold any parameters to be configured, the business action (Trigger STEP Workflow Event) has some pre-configured parameters that are described below:



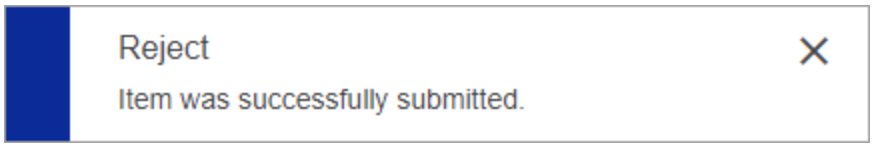

- **STEP Workflow:** From the dropdown select the relevant workflow. The workflow will be the current workflow.
- **Current State:**From the dropdown select the state that the controller object must be in when the action is applied. Users should select the Ready for Import state.
- **Event:**Within this parameter from the dropdown select the event that the action triggers. User should select the Start Import event that automatically start the import for the file that has no warnings and/or errors in the previous states.
- **Process Note:**Within this parameter, enter a comment to be written in the state log when the event is triggered.

The controller objects with files that have entirely failed a previous process will end up in the Error state. The files that have warnings and/or errors will continue successfully through the process and enter into the Ready for Import state, so it is important for users to view the data provided in the Web UI background processes 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.

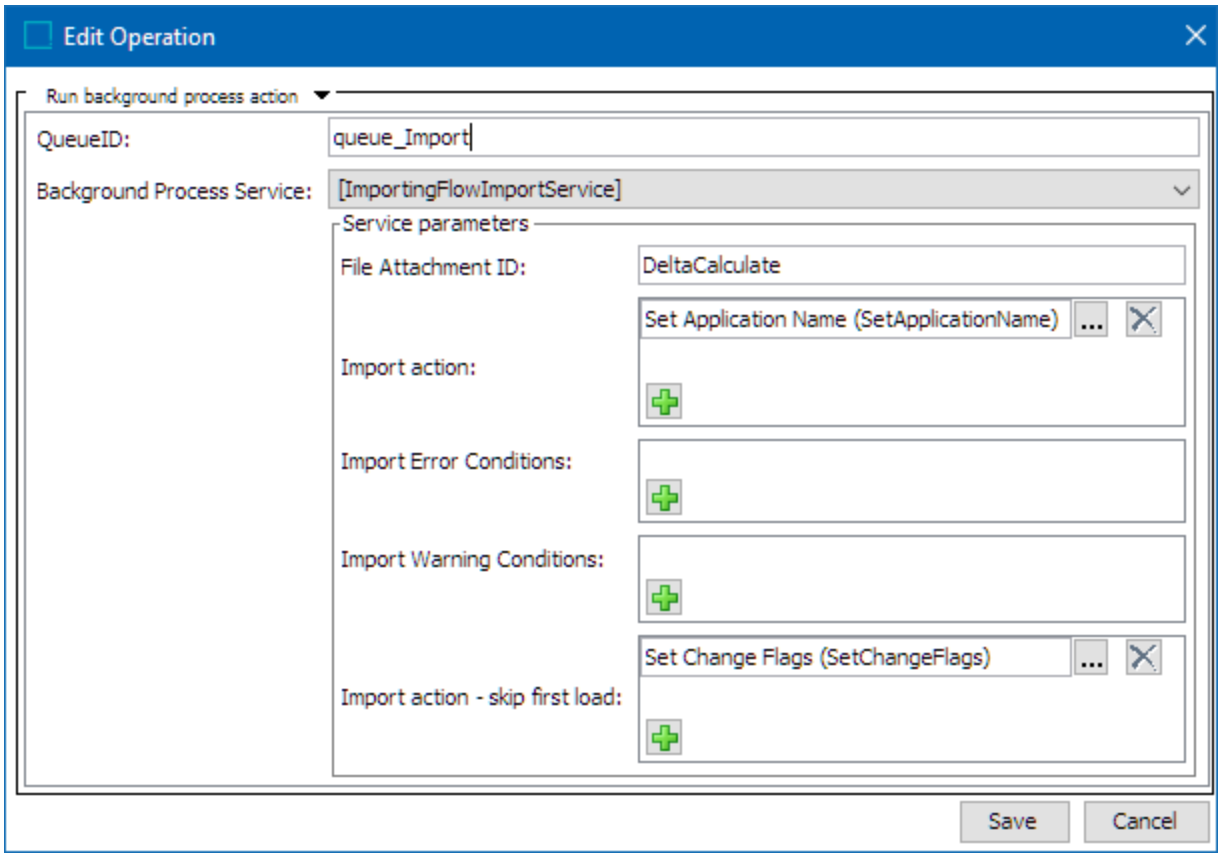


### Import State

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.




- **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 as needed.
- **Import Warning Conditions**: This parameter allows users to add multiple business conditions. When users add a validation condition in this parameter, the warning messages about the data being excluded from the import will be written to a log.
- **Import Error conditions:** 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.
- **Import action - skip first load:** Optional parameter allowing administrators to add a business action. This parameter omits running the added business action on the first load of a file-type. Configuring the Set Change Flags business action in this parameter will ignore objects that are being created for the first time so that it does not get flagged as 'New.' The next time that another file is imported and there is an existing controller entity, the change flags will set for New, Deleted, or Updated as expected. Users should add the Set Change Flags business action either in the 'Import action' parameter or in the 'Import action - skip first load' parameter, but not in both the parameter.

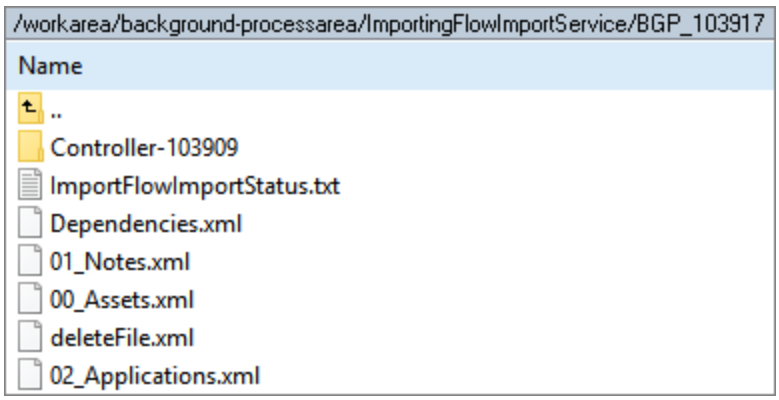
**Note:** Any business rules selected within the import service function the same as business rules applied in a standard STEP import, meaning that they act on the objects being imported (rather than the controller entity). If conditions are used, they must render true or the data will be excluded from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

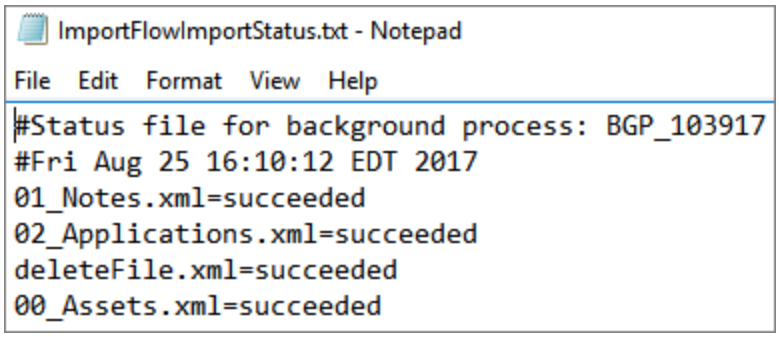
**Important:** Automotive importers do not automatically approve objects that are created by importer. So if Objects need to be approved, then a Business Action using the following JavaScript function must be created: `'node.approve();'.` Additionally, that Business Action must be added to the 'Import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

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

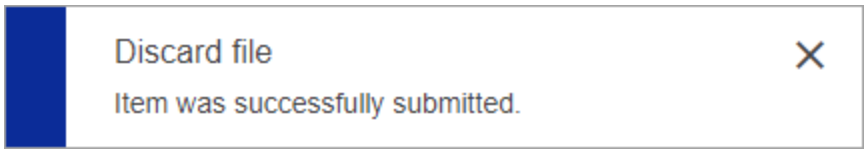
### Import Completed State

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, files that have been discarded cannot be used for delta calculations. Therefore, the last loaded file (controller entity) must be retained in order for it to be used in delta calculation.

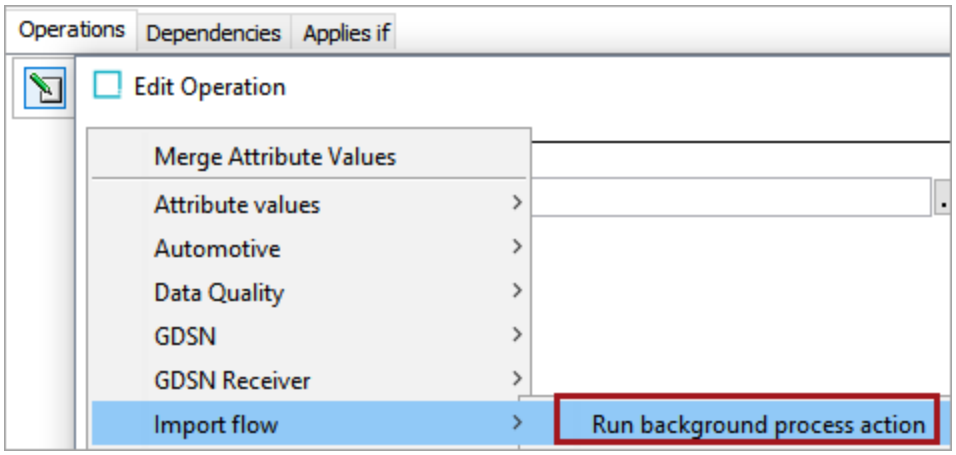
## Modifying Import Framework

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.

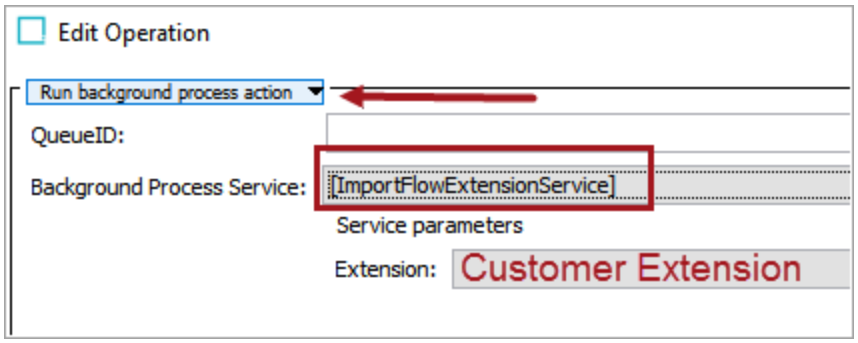
There are many out of the box business rules that can be added for advanced functionality in the importers. For more information, refer to the **Automotive Business Rule Plugins** section.

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]

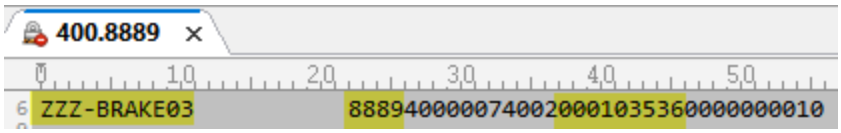
| | |
|---|---|
| AA4 | AA000001 |

| | | | |
|---|---|---|---|
| > | 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]

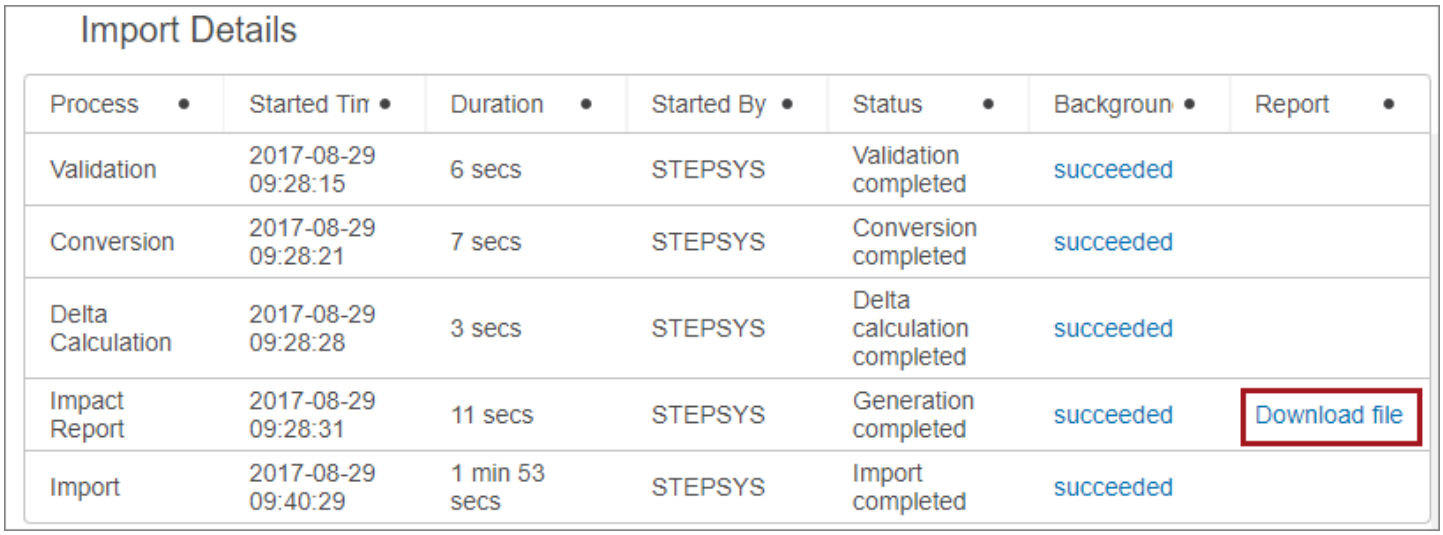
| | | |
|---|---|---|
| ID | | TD_L_a2f9742628a61dcf5537cbcba4efb547 |
| Name | | TD_L_000103536 |
| Object Type | | DS_Linkage |

**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 theses 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 Tin | Duration | Started By | Status | Backgroun | 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, refer to 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. Refer to 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.

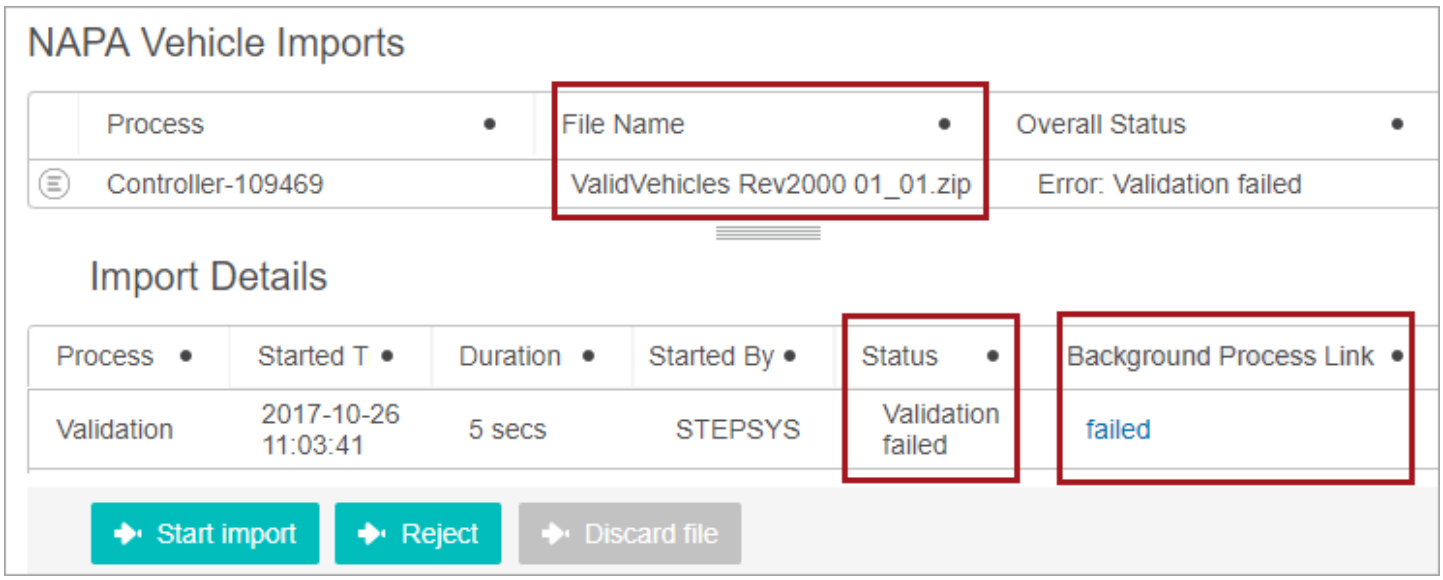
For example, when reference data includes multiple files and a data point included in one is not present in a corresponding related file (e.g., a PCdb Codemaster file includes a position that is omitted from the Position file in the same PCdb), an error will be written to the execution report of the 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.

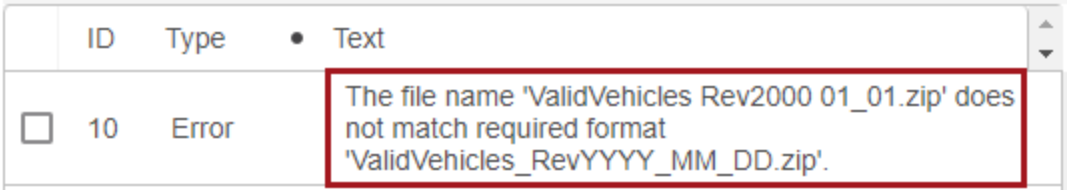
For the examples in the sections below, consider when an importer has a strict file name validation rule (e.g., NAPA Vehicle and NAPA Translation Importers), should a bad file name stop the data from being imported? If so, then the parameter should remain disabled. If not, then the parameter should be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

## Continue on Error Disabled

In the example below, the NAPA Vehicle Import Controller Screen displays the import file name, notice 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.'




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).

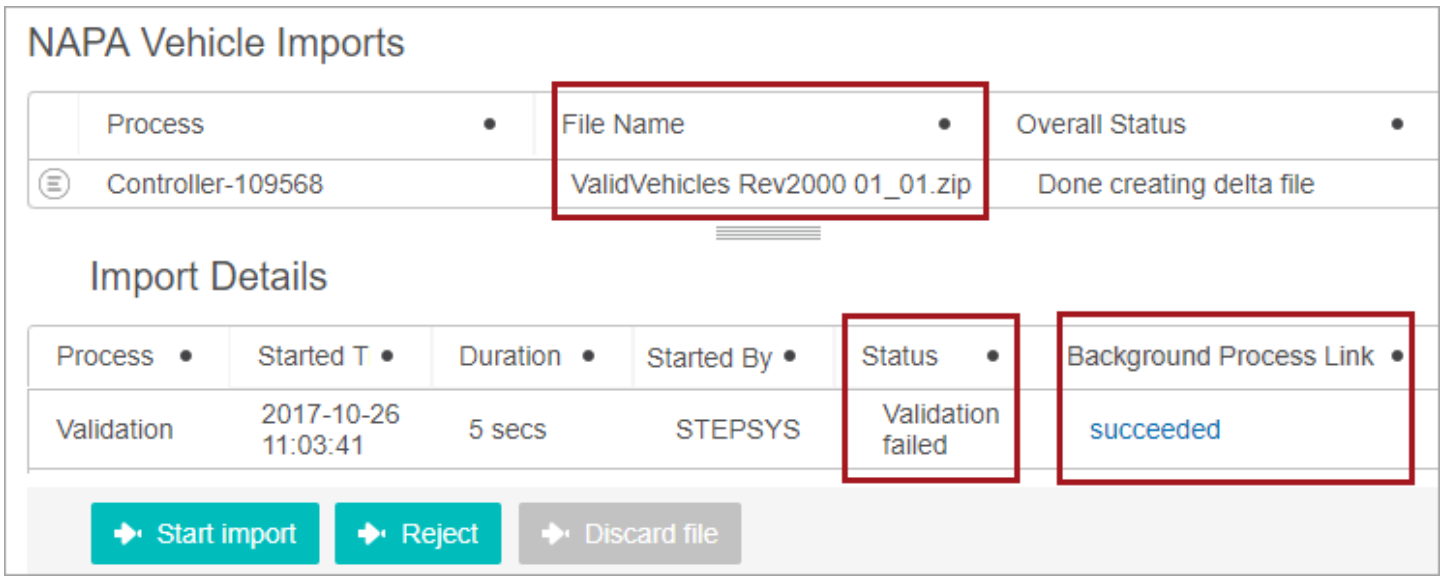



Once the error is addressed the user will need to begin the file import process, as before.

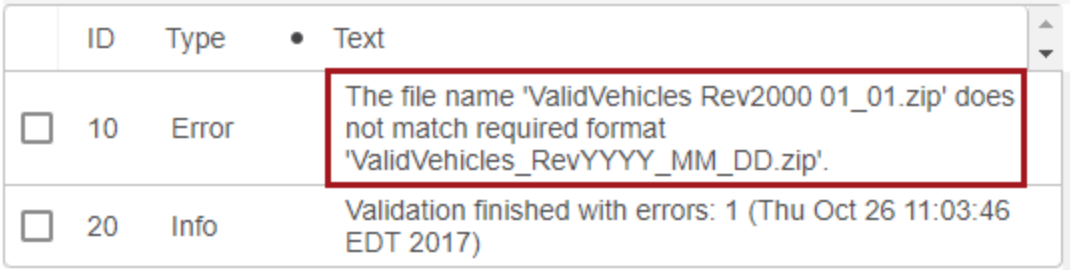
## Continue on Error Enabled

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.




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



# 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, refer to 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 Homepage Widget, including additional information about working with these widgets, refer to File Loading Widget topic, and Status Selector Homepage Widget topic, both within the **Web User Interfaces** section of the **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, refer to 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, refer to the **Workflows** section of **STEP Online Help**.

For general information about the File Loading Widget and Status Selector Homepage Widget, including additional information about working with these widgets, refer to File Loading Widget topic, and Status Selector Homepage Widget topic, both within the **Web User Interfaces** section of the **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, a Node Details screen must be created. For information about configuring the Node Details Screen, refer to the **Node Details Screen** topic within the **Web User Interfaces** section of **STEP Online Help**.

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, refer to 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, refer to 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 Homepage 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.

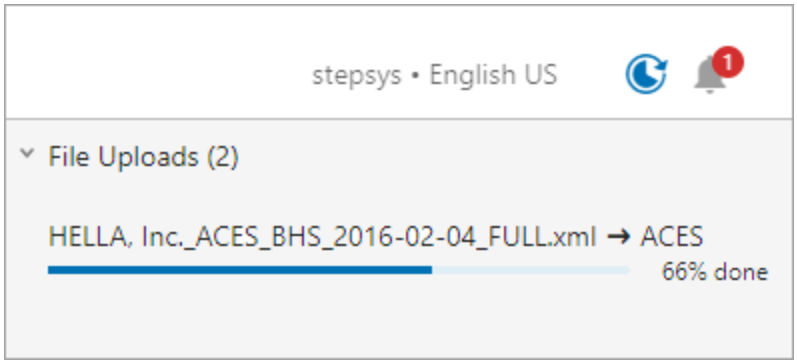
### Using the AutoCare ACES Importer in Web UI

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, refer to 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 'Recent background processes' side panel.

   In the example below, the 'Recent background processes' side panel is expanded and the file, 'HELLA, Inc._ACES_BHS_2016-02-04_FULL.xml' is 66% processed.




   For more information on background process notifications, refer to the **Recent Background Processes Side Panel** 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,

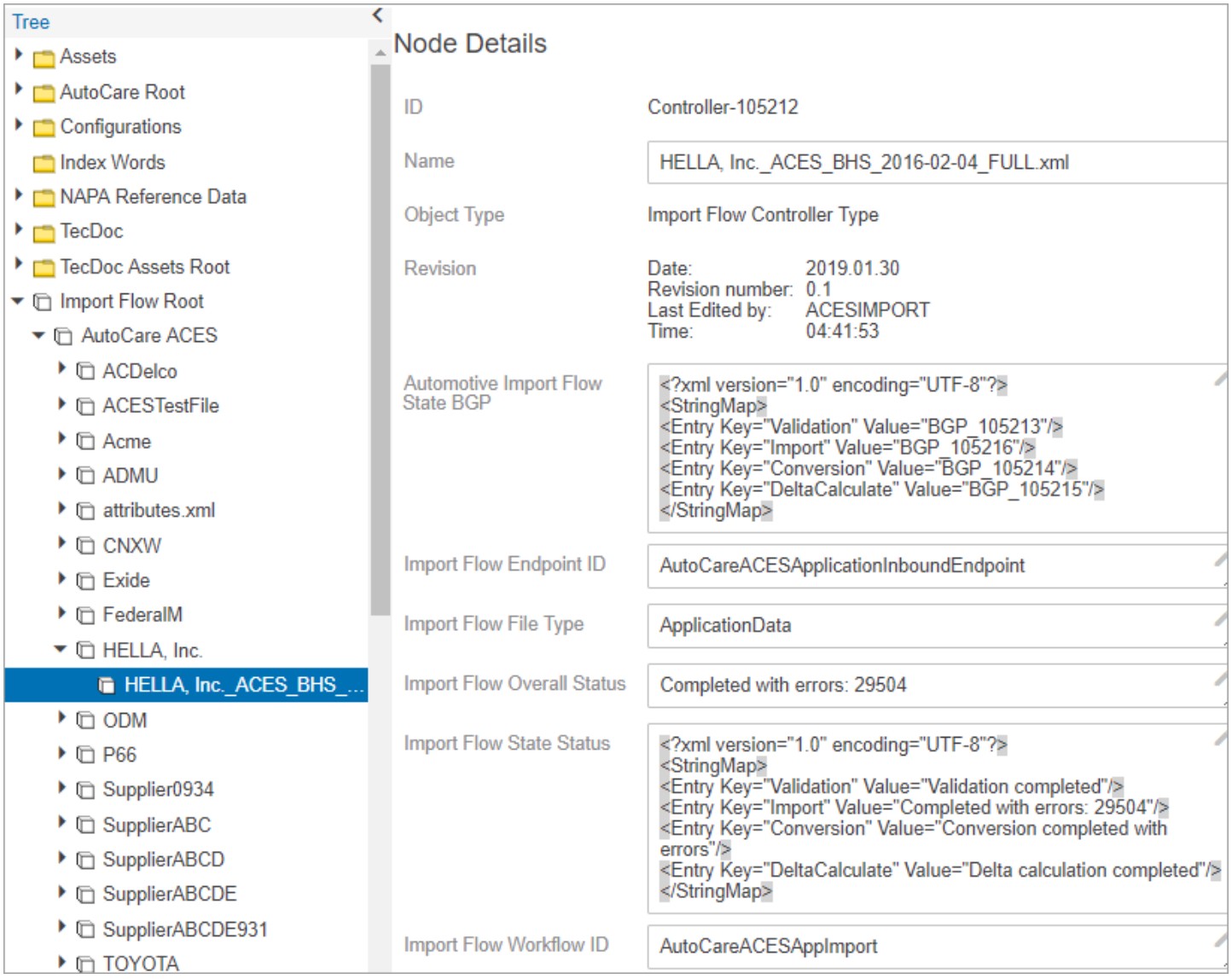
not the objects in the input file. In order for the Business Action to act on the objects being imported, the Business Action must be added to the 'Import action' parameter on the 'Import' state. For more information, refer to the **Import State** topic within the **Default Workflow States and Functions** section.

4. If a Node 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 Node Details screen has not been configured. For information about configuring the Node Details Screen, refer to the **Node Details Screen** topic within the **Web User Interfaces** section of **STEP Online Help**.

   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.

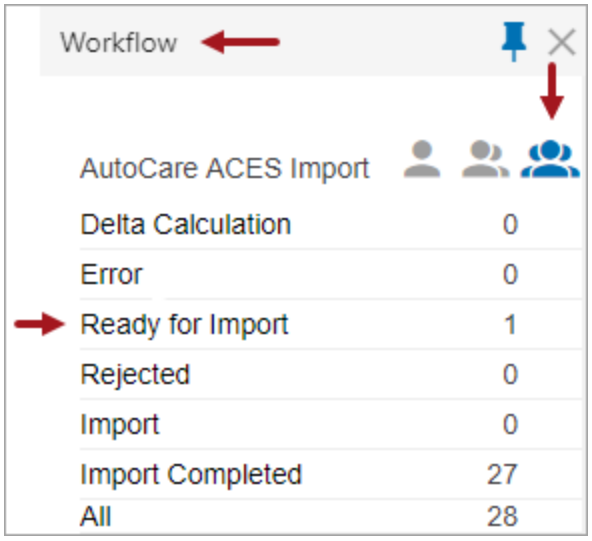



- **Name:** STEP Name of the Controller Entity. This is generated from the original name of the uploaded file.
- **Automotive Import Flow State BGP:** Attribute used to store the IDs of the background processes.
- **Import Flow State Status:** Attribute used to store and display the status of each process (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 Homepage Widget on the Web UI homepage and/or the Status Selector component on the left side panel.

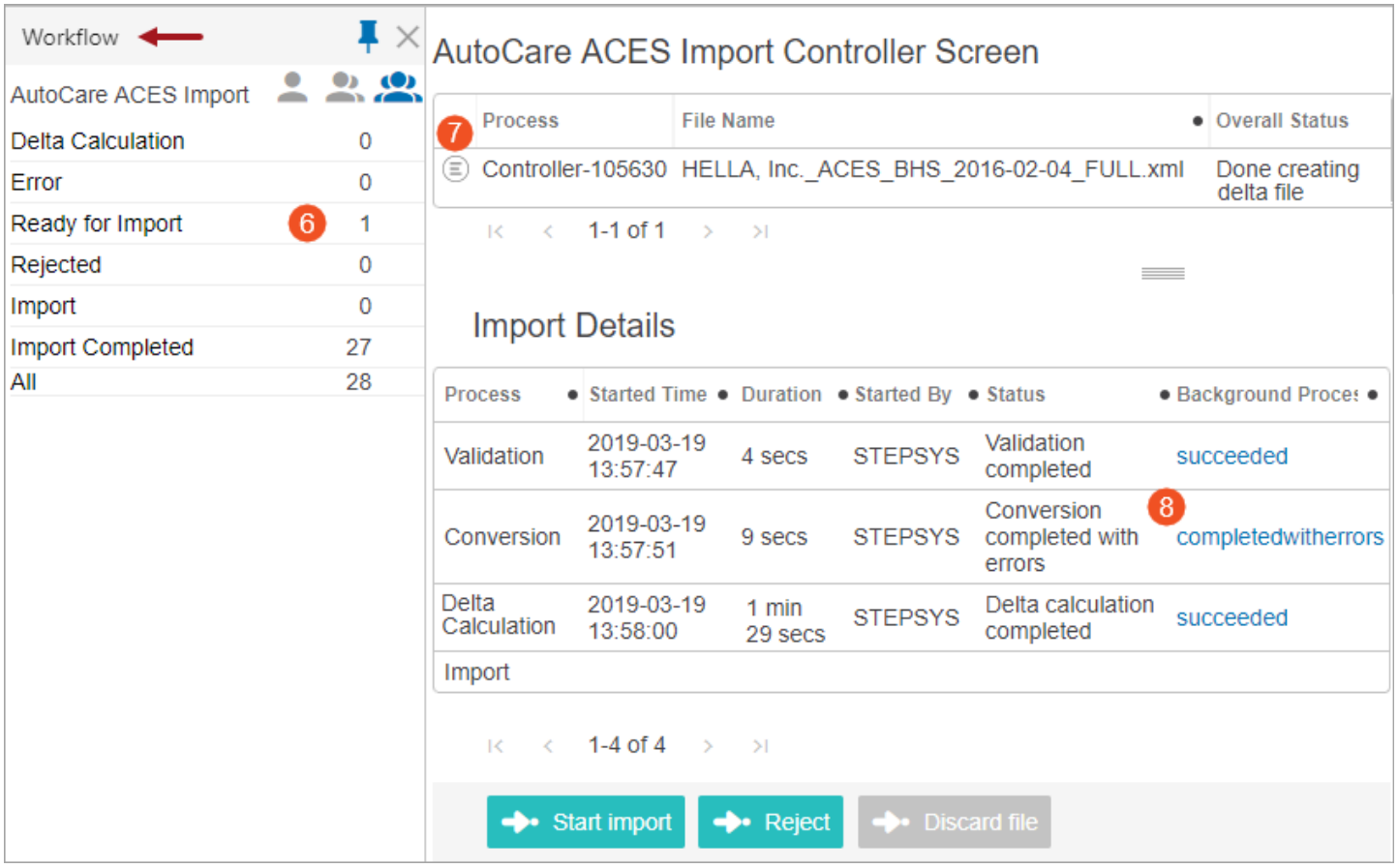
   In the example below, the Workflow panel is expanded, and the **Triple user button** is selected so that all items assigned to any user are displayed. Notice that one file is in the 'Ready for Import' state, and twenty seven files in the 'Import Completed' state, for a total of twenty eight files.

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

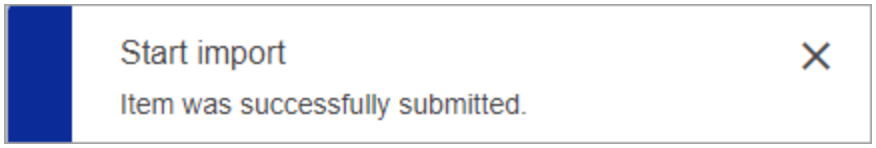



6. Clicking on the **Ready for Import** state within the Status Selector homepage and/or the left side panel widget will navigate the user to the AutoCare ACES Import Controller Screen with only those Controllers in the 'Ready for Import' state displayed.
7. Clicking the Controller icon displays the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

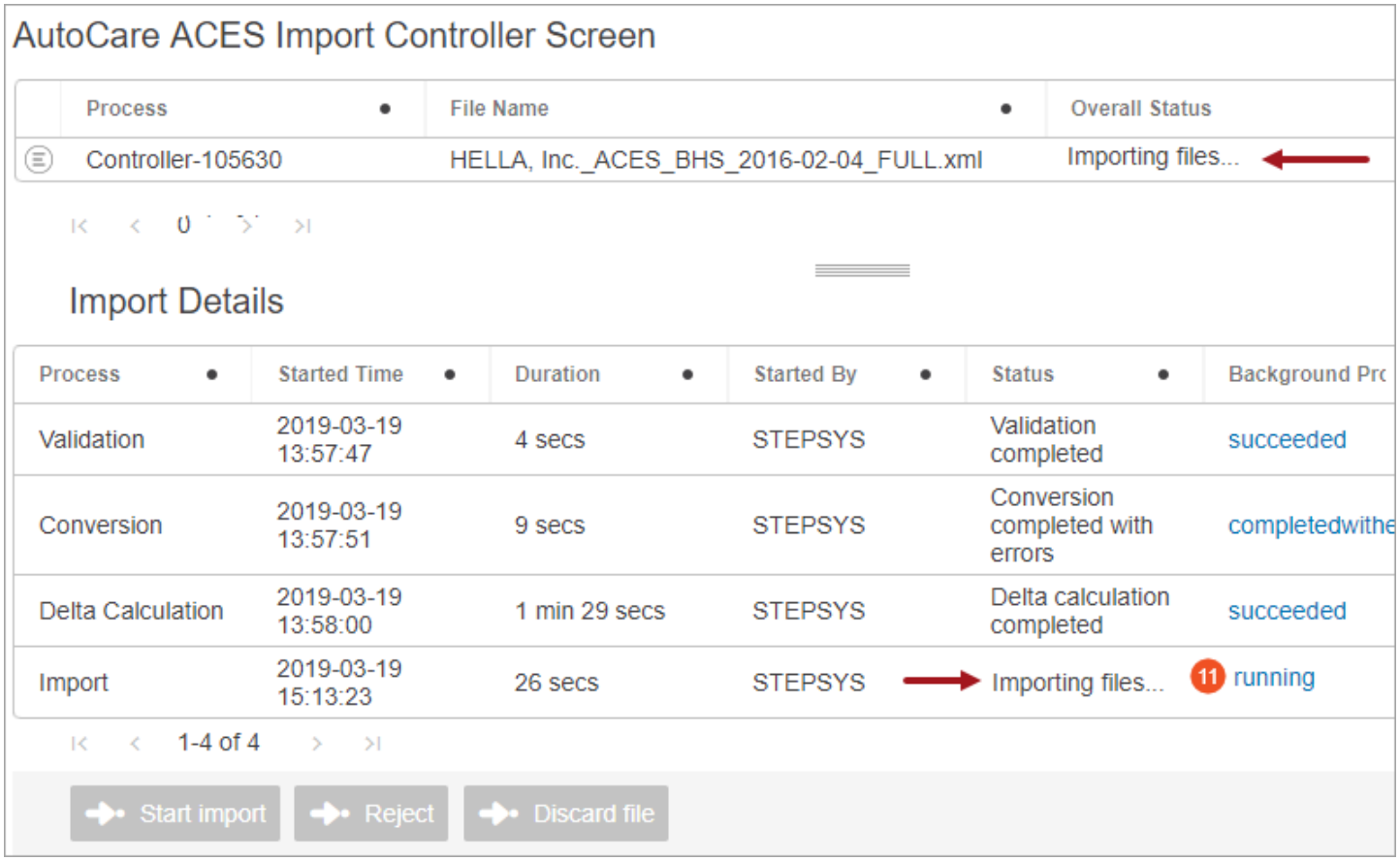



8. Optionally, before starting the import, clicking on the 'completedwitherrors' Background Process Link will display the Background Process Details screen where users can view or download the details of the validation process.
9. Click the **Reject** button to move the Controller to the Rejected state. For more information, refer to 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, refer to 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 Homepage Widget on the Web UI homepage and/or the Status Selector component in the left side panel. More detailed progress can be monitored in the import process Status column that is available within the Import Controller Screen.

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.'




11. 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, refer to the **Import Completed State** topic within the **Import Framework** section.

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

    For more information on automotive workflow states, refer to 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, ACES 4.0, ACES 4.1 and ACES 4.2. 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>ACME</Company>
       <SenderName>Katherine Jenkins</SenderName>
       <SenderPhone>6777978488</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 cannot 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>
```

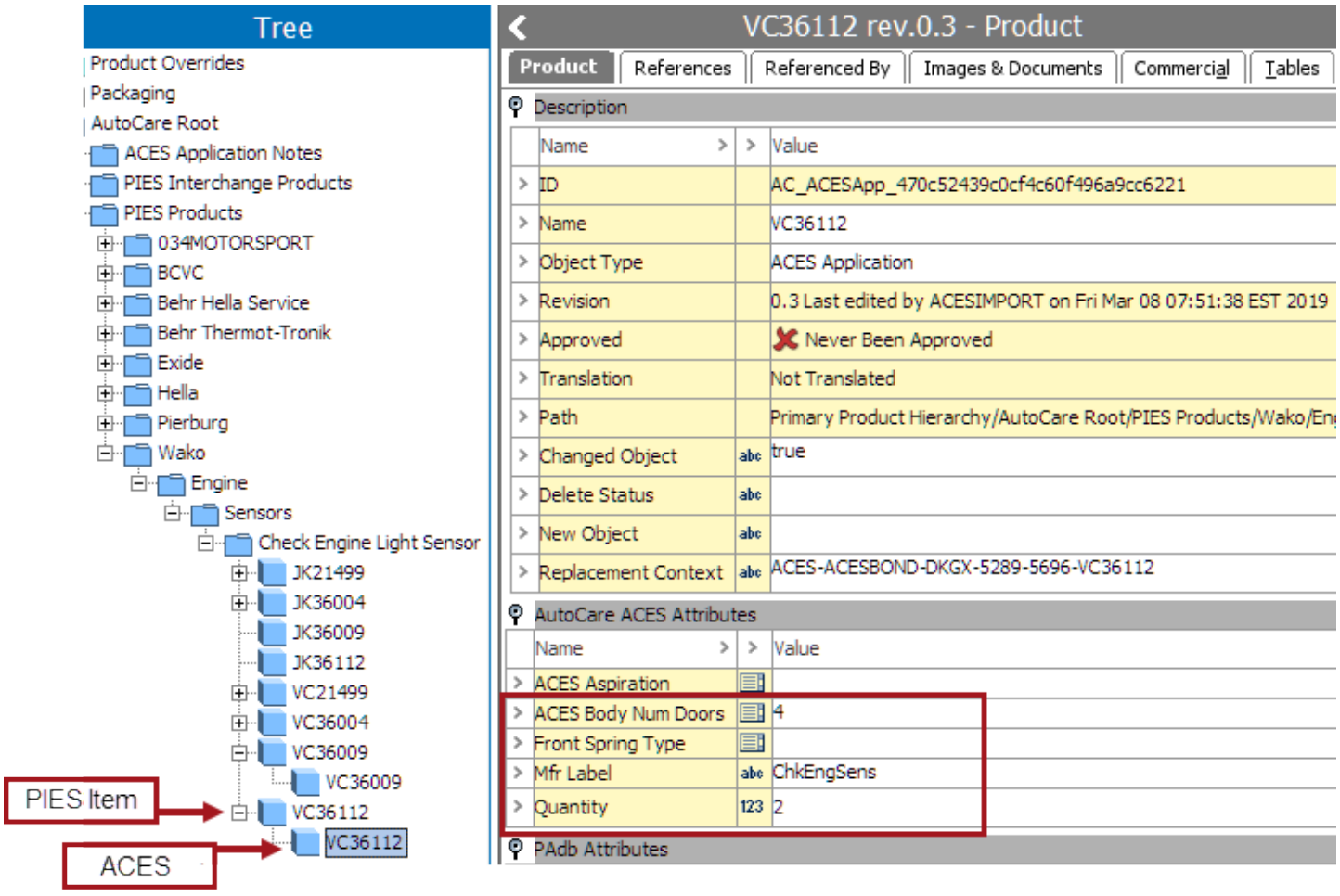
```
        <Part BrandAAIAID="DKGX">VC36112</Part>
    </App>
```

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).

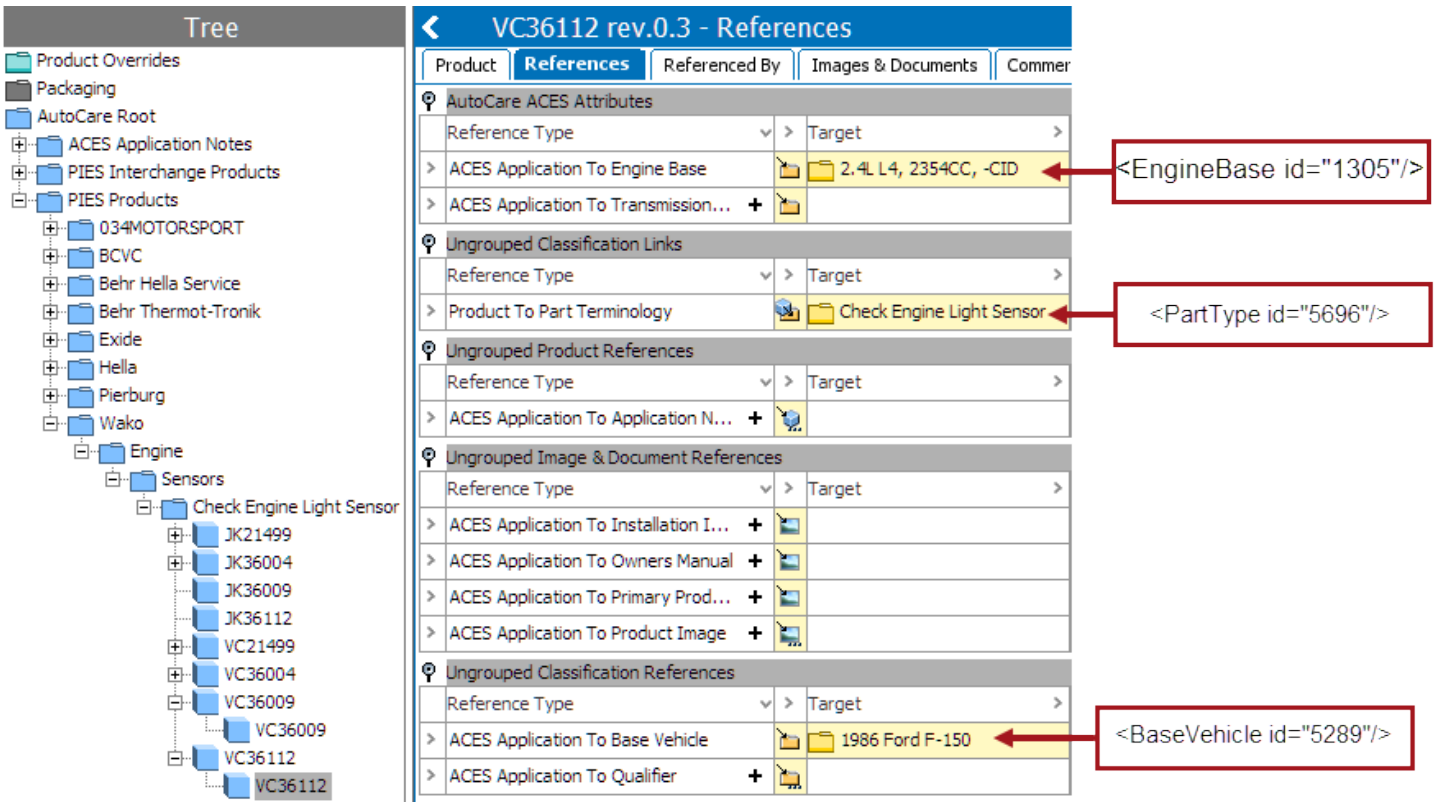
**Note:** The automotive importers do not delete any objects, it just marks objects as new, delete, or change through change flags.

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.




<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.




## 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, refer to 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 en 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_129461111111111111' 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, refer to the **Designer Access** topic within the **Web User Interfaces** 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, a Node Details screen must be created. For information about configuring the Node Details Screen, refer to the **Node Details Screen** topic within the **Web User Interfaces** section of **STEP Online Help**.

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 Homepage Widget, including additional information about working with these widgets, refer to File Loading Widget topic, and Status Selector Homepage Widget topic, both within the **Web User Interfaces** section of the **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, refer to 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.

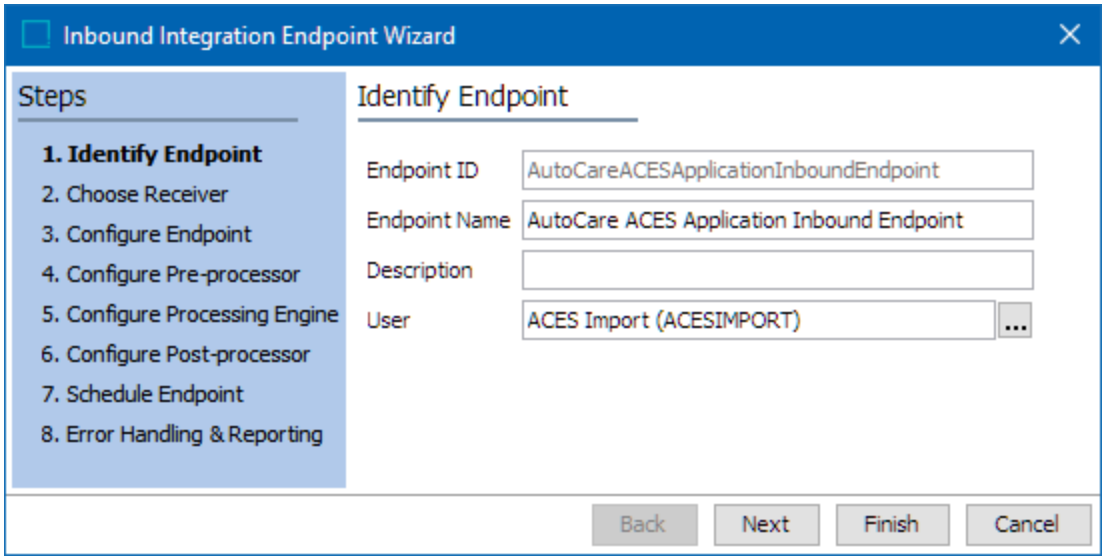
## Prerequisites

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, refer to 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.

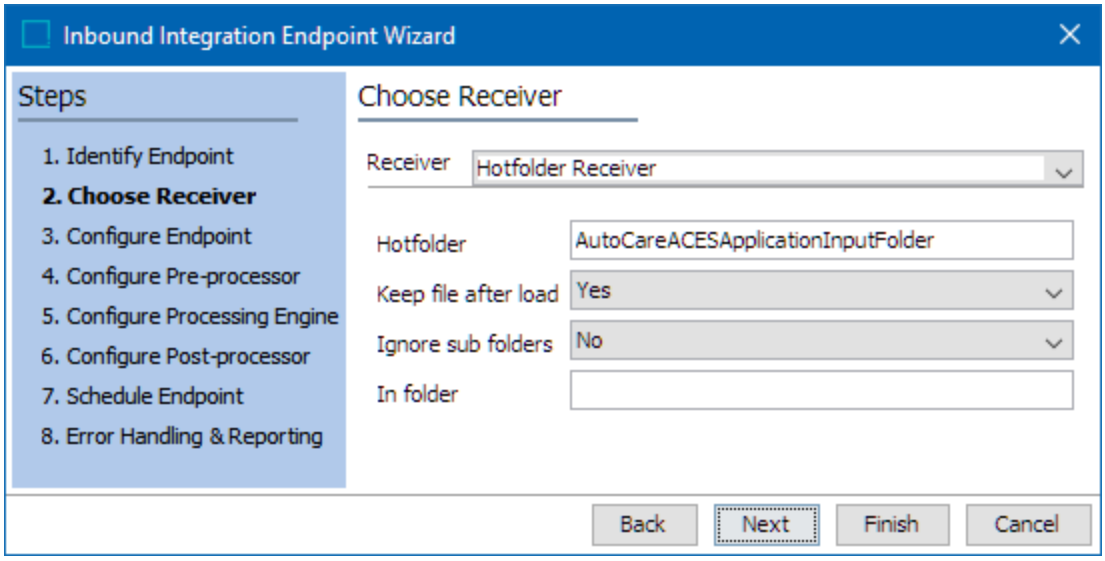
   In the example below, an 'ACES Import' user was created prior to configuring this IIEP. This is recommended to more easily track when this IIEP is responsible for changes to data.



For more information about the parameters, refer to the **IIEP - Identify Endpoint** topic within the **Data Exchange** section of **STEP Online Help**.

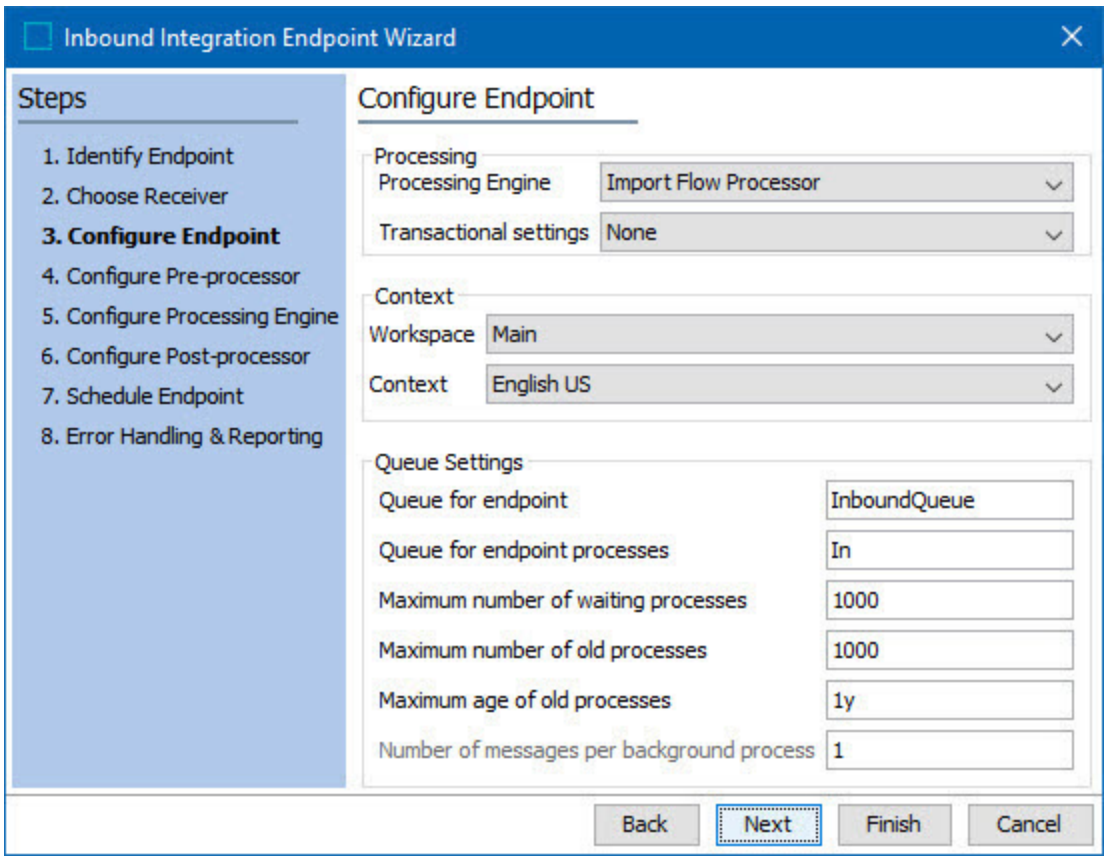
3. Click the **Next** button, and the Choose Receiver parameters will display. By default, the parameters are populated as recommended and shown below, except the Hotfolder parameter. This mandatory parameter must be populated with a hotfolder name before the Next button will enable.

> **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, refer to the **IIEP - Choose Receiver** topic within the **Data Exchange** section of **STEP Online Help**.

4. 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

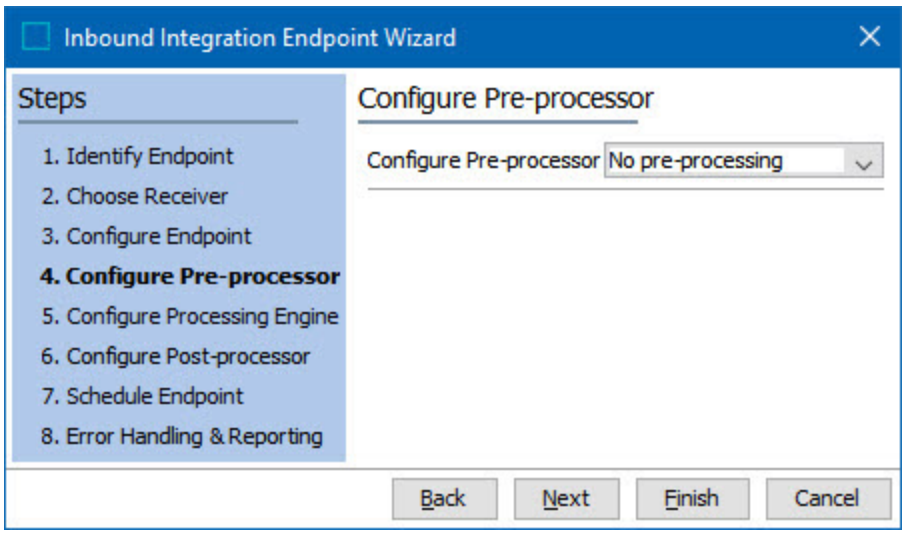



**Important:** The Import Flow Processor is only responsible for picking up files from the configured hotfolder, creating a Controller Entity object, and updating the Entity with the Import Flow State BGP and Status. The Import Flow Processor works with the Background Process Service for each workflow 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.

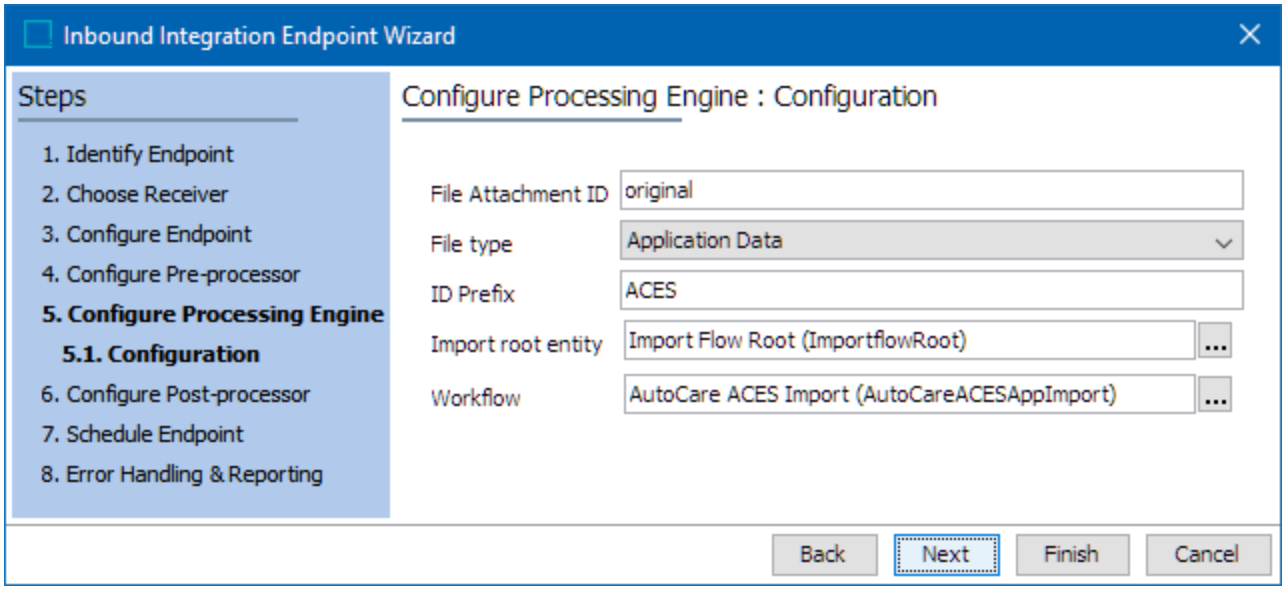
For more information about the parameters, refer to the **IIEP - Configure Endpoint** topic within the **Data Exchange** section of **STEP Online Help**.

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




For more information about the parameter, refer to the **IIEP - Configure Pre-processor** topic within the **Data Exchange** section of **STEP Online Help**.

6. 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.



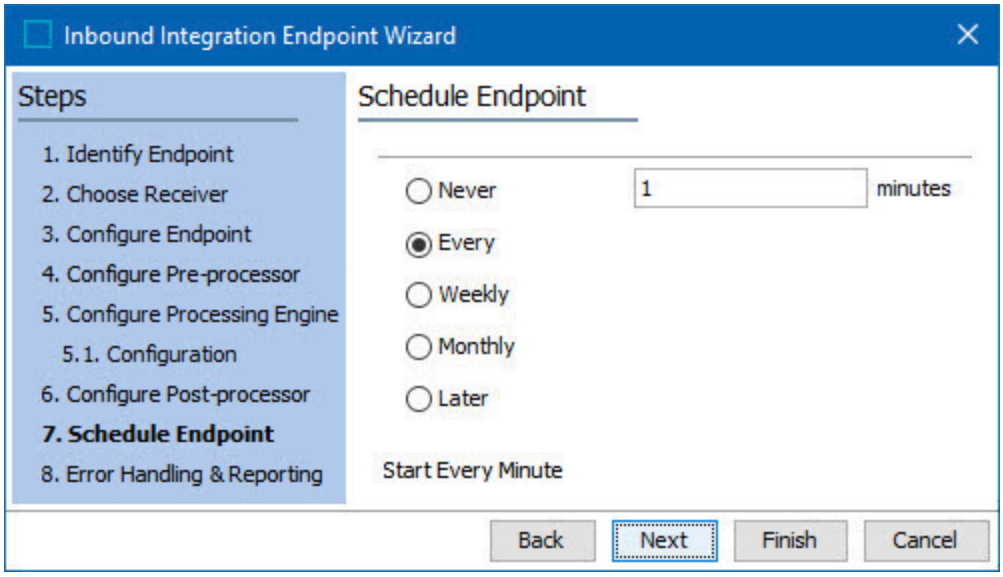

The prior Configure Endpoint step determines the options available for the Configure Processing Engine step.

- 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.

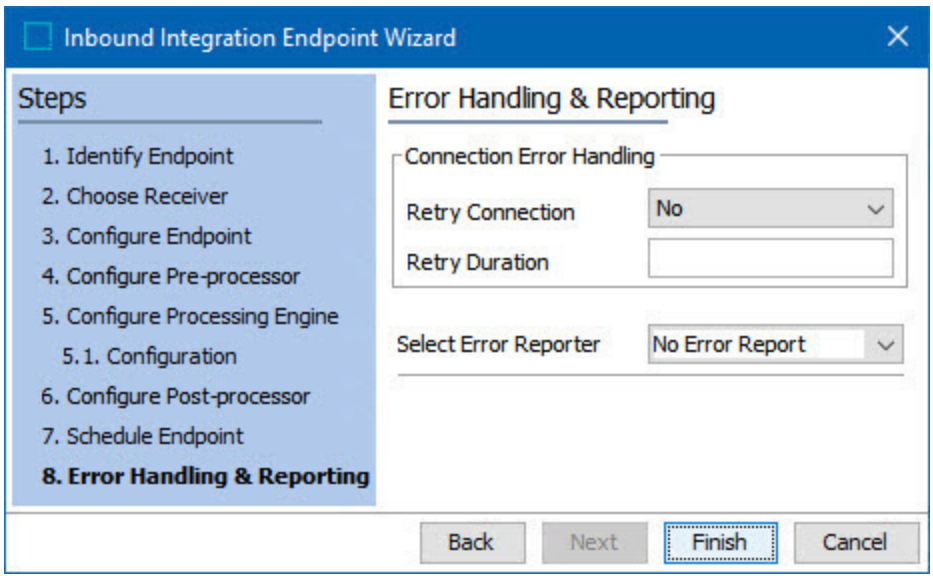
7. Click the **Next** button, and the Schedule Endpoint parameters will display. By default, 'Never' is selected. Optionally, update the values to those shown below.




**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, refer to the **IIEP - Schedule Endpoint** topic within the **Data Exchange** section of **STEP Online Help**.

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




For more information about the parameter, refer to the **IIEP - Error Handling & Reporting** topic within the **Data Exchange** 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, refer to the **Running an Inbound Integration Endpoint** topic within the **Data Exchange** 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, refer to 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, refer to 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.

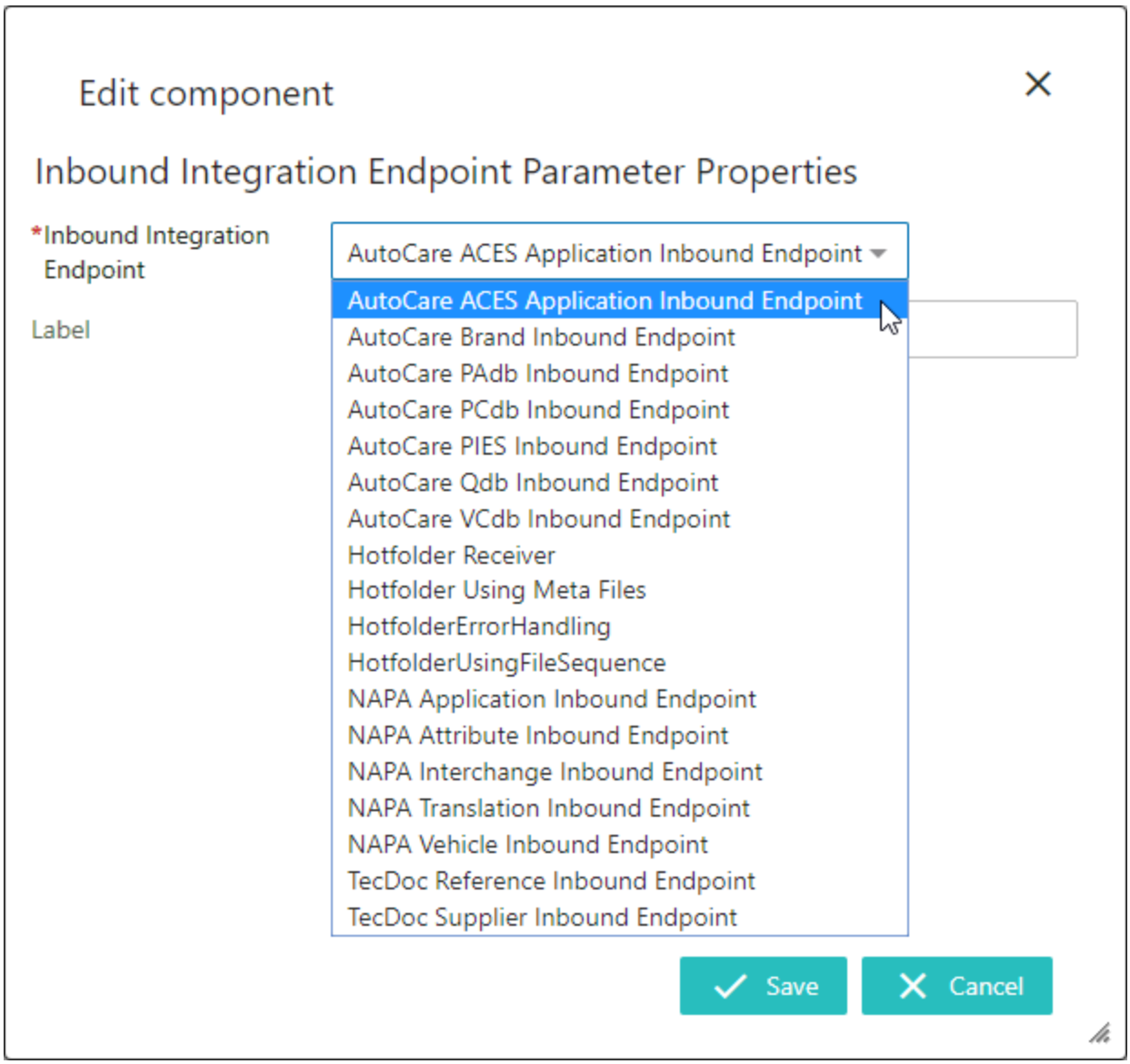
## Configuration

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, refer to 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).

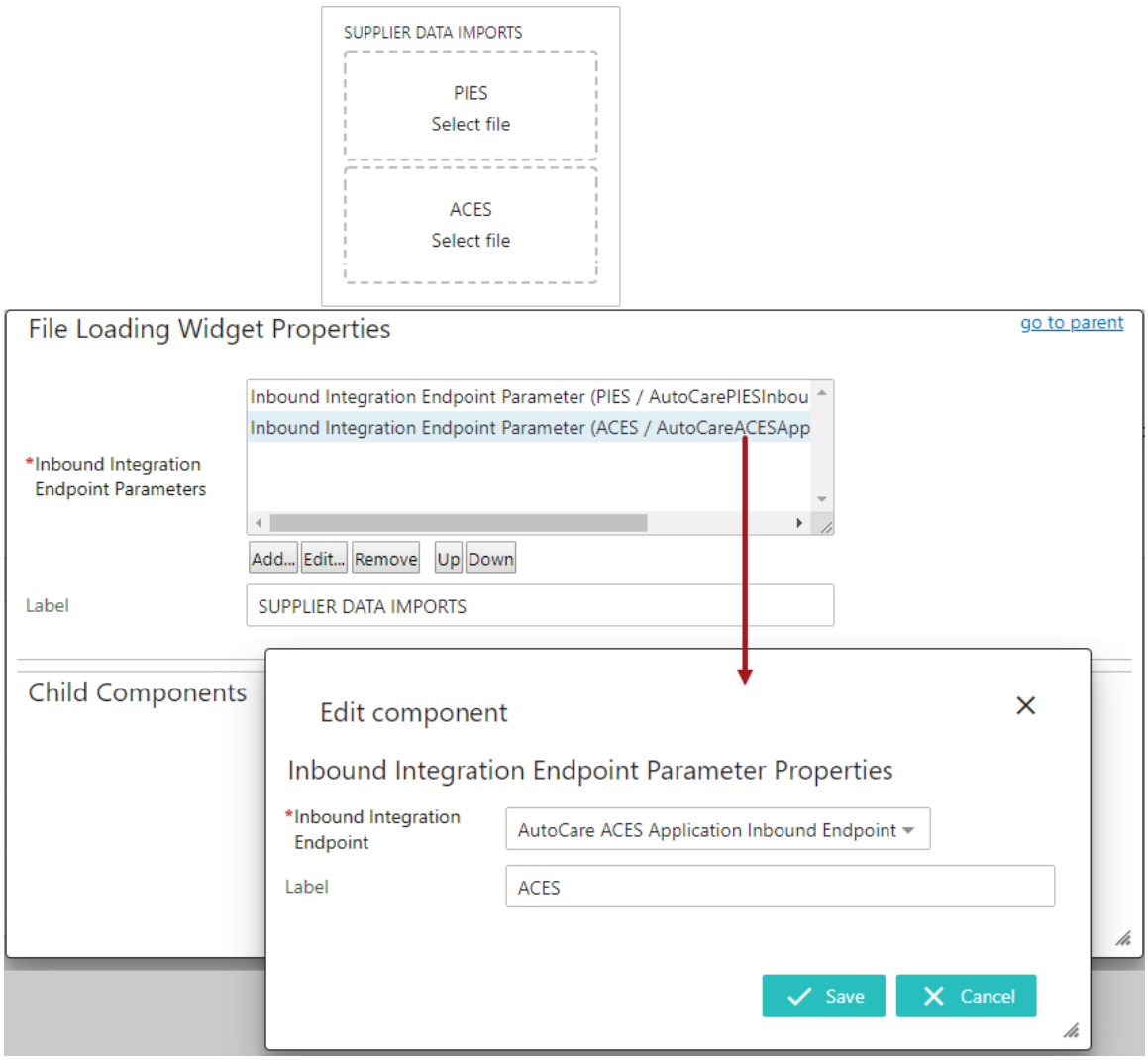



**Note:** If the desired IIEP does not display in the dropdown, then it can be created using the steps described in the **Configuring an IIEP for ACES Imports** topic.

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

For general information about the File Loading Widget and Status Selector Homepage Widget, including additional information about working with these widgets, refer to File Loading Widget topic, and Status Selector Homepage Widget topic, both within the **Web User Interfaces** section of the **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, a Node Details screen must be created. For information about configuring the Node Details Screen, refer to the **Node Details Screen** topic within the **Web User Interfaces** section of **STEP Online Help**.

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, refer to 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 Homepage 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.

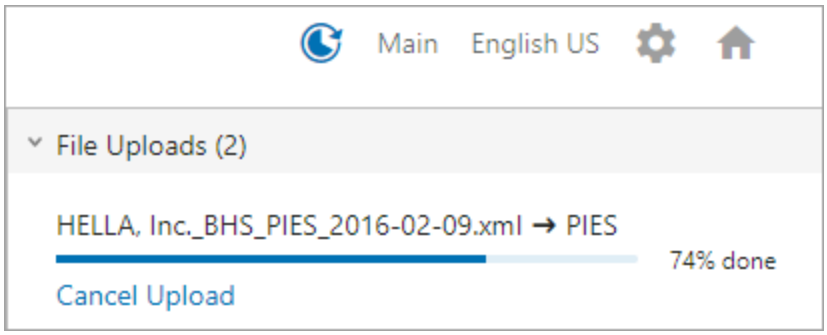
### Using the AutoCare PIES Importer in Web UI

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, refer to 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 'Recent background processes' side panel.

   In the example below, the 'Recent background processes' side panel is expanded and the file, 'HELLA, Inc._BHS_PIES_2016-02-09.xml' is 74% processed.




   For more information on using the 'Recent background processes' side panel, refer to the **Recent Background Processes Side Panel** 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.

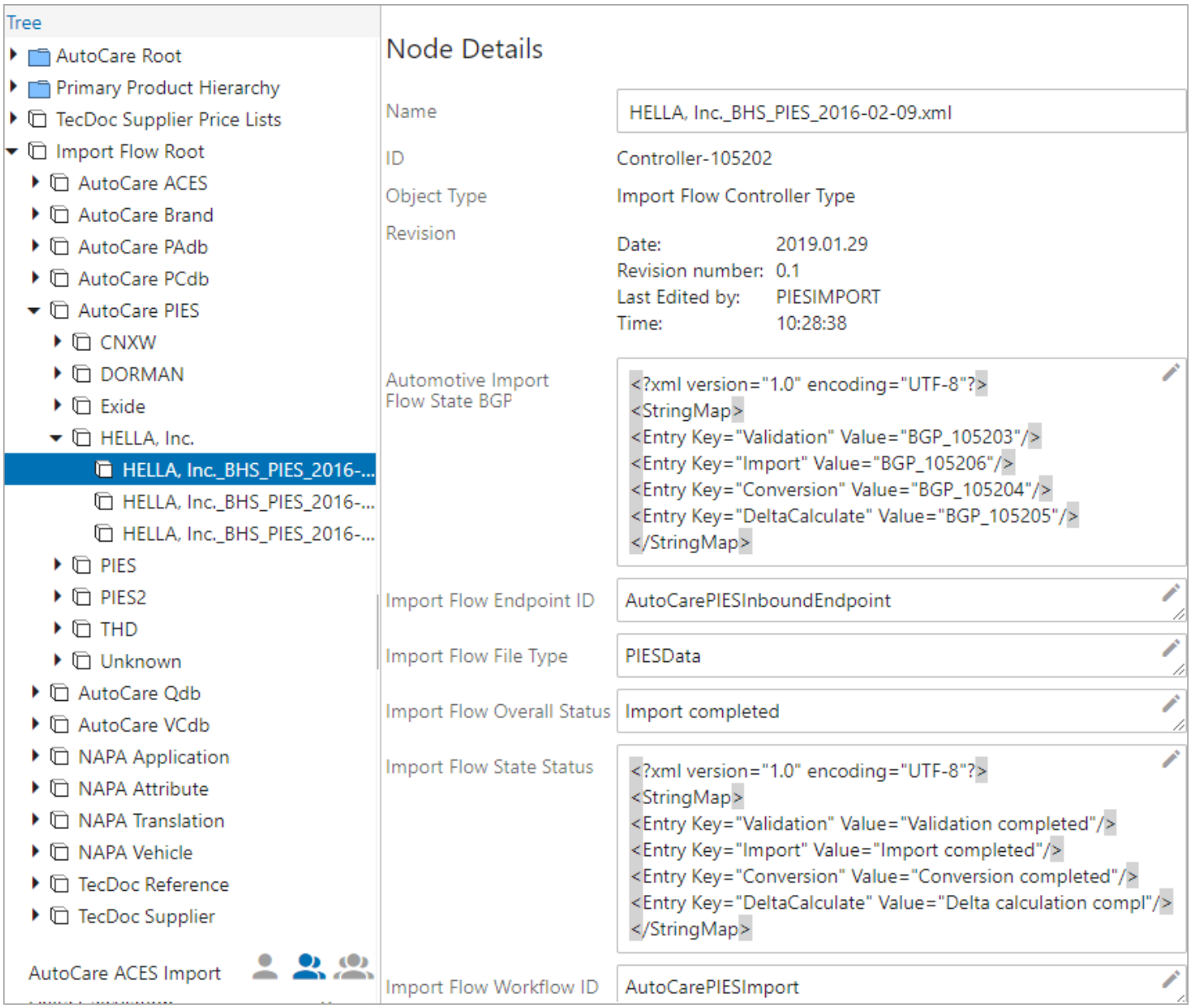
**Important:** It is critical to understand that it is only the Controller Entity that is in the workflow - the objects being acted on (created / updated / deleted) via information supplied in the import file are not in the workflow. Therefore, running standard business actions acting on current object 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, refer to the **Import State** topic within the **Default Workflow States and Functions** section.

4. If a Node 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 Node Details screen has not been configured. For information about configuring the Node Details Screen, refer to the **Node Details Screen** topic within the **Web User Interfaces** section of **STEP Online Help**.

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.



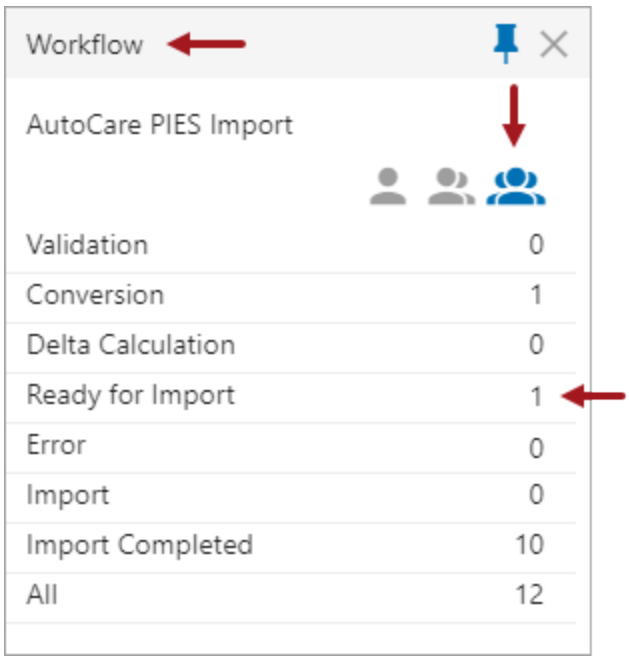

- **Name:** STEP Name of the Controller Entity. This is generated from the original name of the uploaded file.
- **Automotive Import Flow State BGP:** Attribute used to store the IDs of the background processes.

- **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 Homepage Widget on the Web UI homepage and/or the Status Selector component on the left side panel.

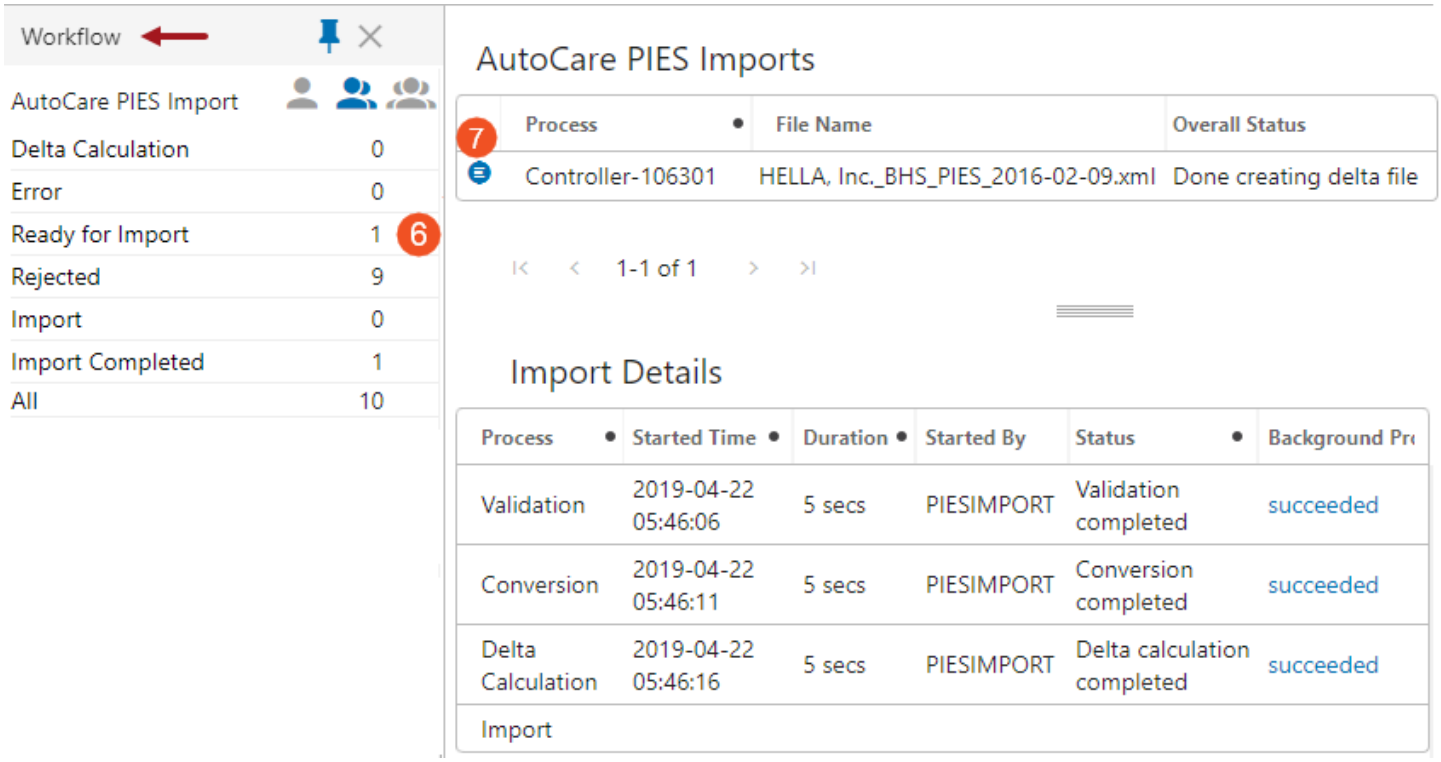
   In the example below, the Workflow panel is expanded, and the **Triple user button** is selected so that all items assigned to any user are displayed. Notice that one file is in the 'Ready for Import' state, and ten files in the 'Import Completed' state, for a total of twelve files.

   For more information, refer to 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.

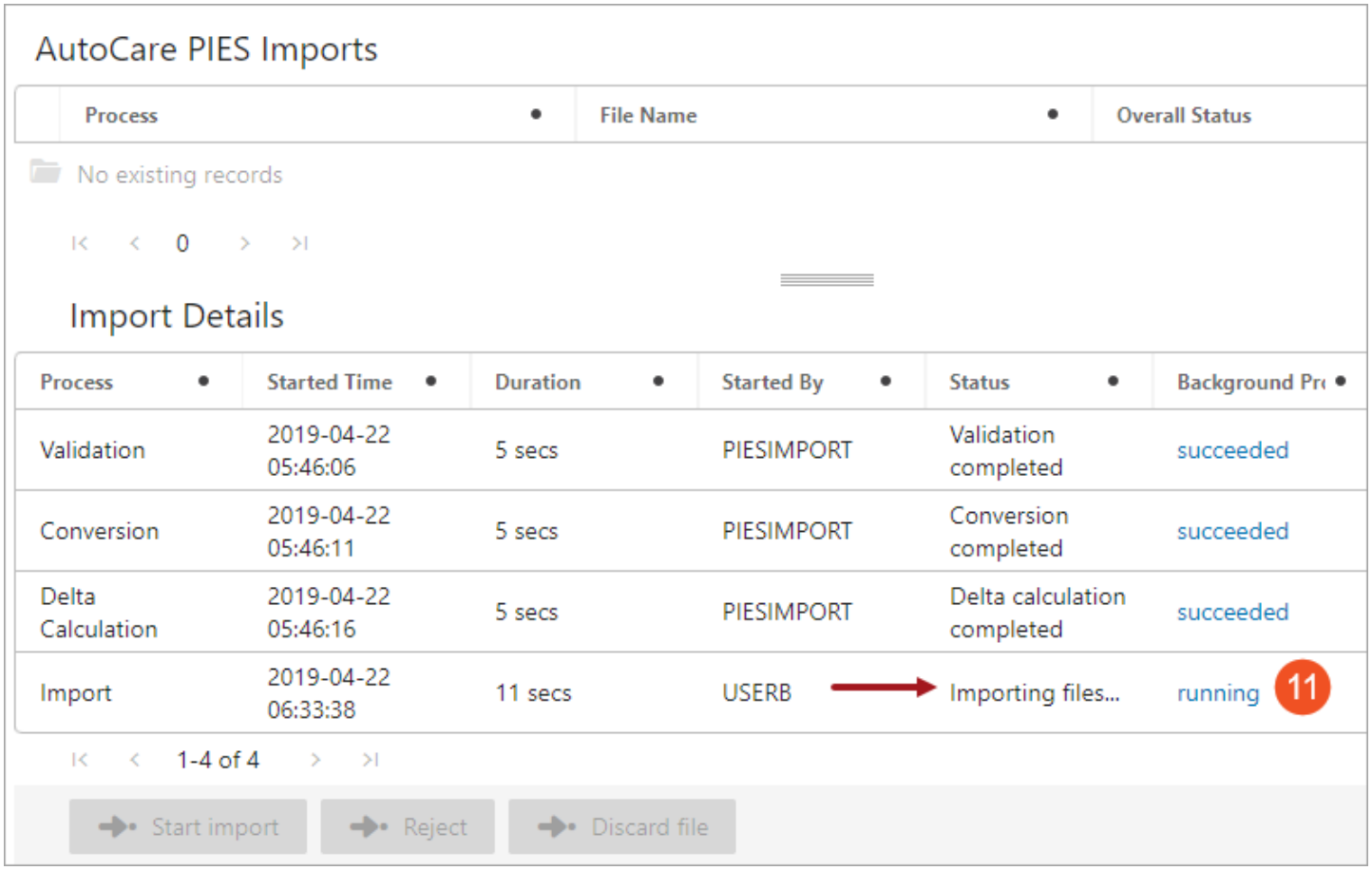
   In the example below, Controller 'HELLA, Inc._BHS_PIES_2016-02-09.xml' (ID 'Controller-106301') with an overall status of 'Done creating delta file' is selected and the Import Details are displayed below the Controller list.




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, refer to 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, refer to 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 Homepage Widget on the Web UI homepage and/or the Status Selector component in the left side panel. More detailed progress can be monitored in the import process Status column that is available within the Import Controller Screen.

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.'




11. 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, refer to the **Import Completed State** topic within the **Import Framework** section.

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

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

## Replacement Rules in PIES Importer

The PIES Importer uses the ReplacementRules tag to calculate and replace attribute values, data container values, and references populated for the PIES Items that are already existing in the system. The attribute groups, data containers, and references that are listed below are currently handled for replacement using replacement rules. This information can be found within the 09_Items.xml file of the Delta Calculation state.

```
<ReplacementRules>
    <Products>
```

```
      <ReplaceValues AttributeGroupID="AC_PIESItemSegmentITEM" CrossContext="N"/>
      <ReplaceValues AttributeGroupID="AC_PIESDescriptionSegmentDESC"
CrossContext="N"/>
      <ReplaceValues AttributeGroupID="AC_PIESExtendedProductInfoSegmentEXPI"
CrossContext="N"/>
      <ReplaceValues AttributeGroupID="AC_PAdbAttributes" CrossContext="N"/>
      <ReplaceValues AttributeGroupID="AC_PIESPackagingSegmentPACK"
CrossContext="N"/>
      <ReplaceDataContainers DataContainerTypeID="AC_PIESItem_Description"/>
      <ReplaceDataContainers DataContainerTypeID="AC_PIES_PRCE"/>
      <ReplaceCrossReferences ReferenceTypeID="AC_PIESInterchange"
CrossContext="Y"/>
      <ReplaceCrossReferences ReferenceTypeID="AC_PIESItemToKitComponentItem"
CrossContext="Y"/>
      <ReplaceClassificationReferences TypeID="AC_ProductToPartTerminology"
CrossContext="Y"/>
    </Products>
  </ReplacementRules>
```

For information about the ReplacementRules tag in STEPXML files, refer to the **ReplacementRules Tag in STEPXML** topic within the **STEPXML Format** section of **STEP Online Help**.

**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, PIES 7.0, PIES 7.1 and PIES 7.2. Each version shares a common structure with additional 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>Katherine</TechnicalContact>
        <ContactEmail>Katherine</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>
```

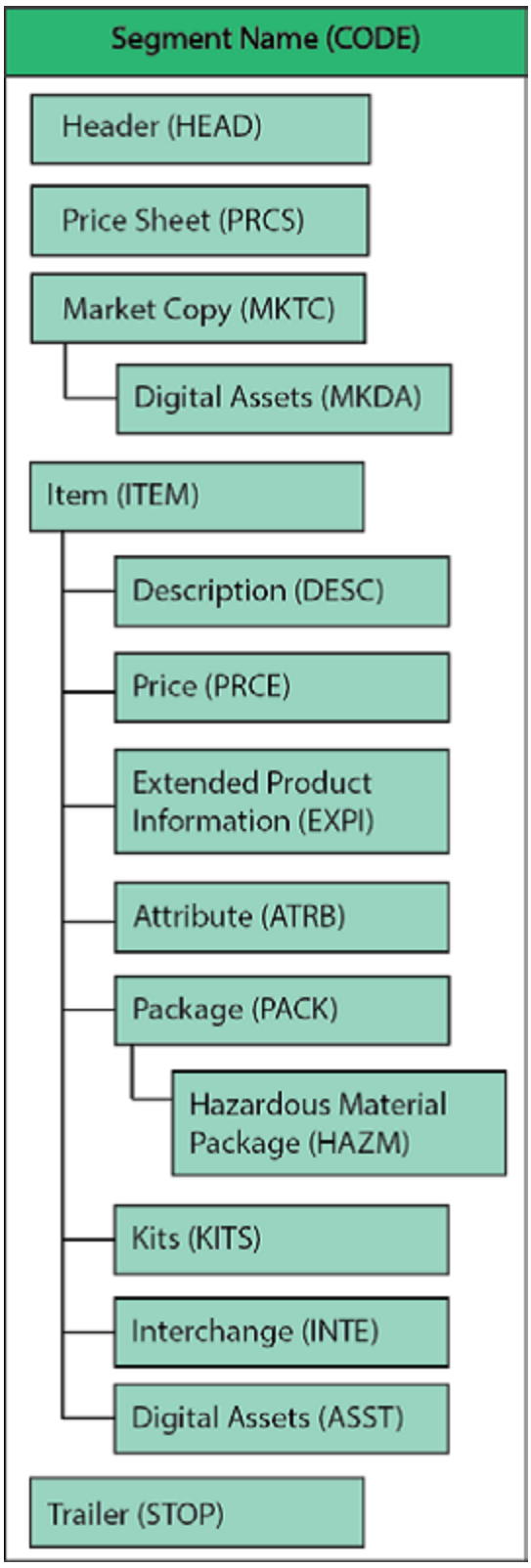
```
    <Items>
        <Item MaintenanceType="A">
            <HazardousMaterialCode>Y</HazardousMaterialCode>
            <BaseItemID>VC21</BaseItemID>
            <ItemLevelGTIN GTINQualifier="UP">11234567890123</ItemLevelGTIN>
            <PartNumber>VC21499</PartNumber>
            <BrandAAIAID>DKGX</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>
            <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>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>
            <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.




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 section 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 does not 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, refer to 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>7.2</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 date the last | <ChangesSinceDate>….</ChangesSinceDate> |

| XML Element | Requirement | Description | Example XML Code |
|---|---|---|---|
| | | 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>9999999990001</ParentDUNSNumber> |
| ParentGLN | Optional | GS1 Company ID + Location ID + Check Digit. The GLN is a standard means of identifying global trading partner | <ParentGLN>7777777555551</ParentGLN> |

| XML Element | Requirement | Description | Example XML Code |
|---|---|---|---|
| | | 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. | |
| 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 | 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 | <BrandOwnerDUNS>8888888880001</BrandOwnerDUNS> |
| BrandOwnerGLN | Optional | 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 | <BrandOwnerGLN>7777777123459</BrandOwnerGLN> |

| XML Element | Requirement | Description | Example XML Code |
|---|---|---|---|
| | | 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. | |
| 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 | <BuyerDuns>8888888880001</BuyerDuns> |

| XML Element | Requirement | Description | Example XML Code |
|---|---|---|---|
| | | identification 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/Sende | <ContactEmail>john@smith.com</ContactEmail> |

| XML Element | Requirement | Description | Example XML Code |
|---|---|---|---|
| | | r company for resolving issues with 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)

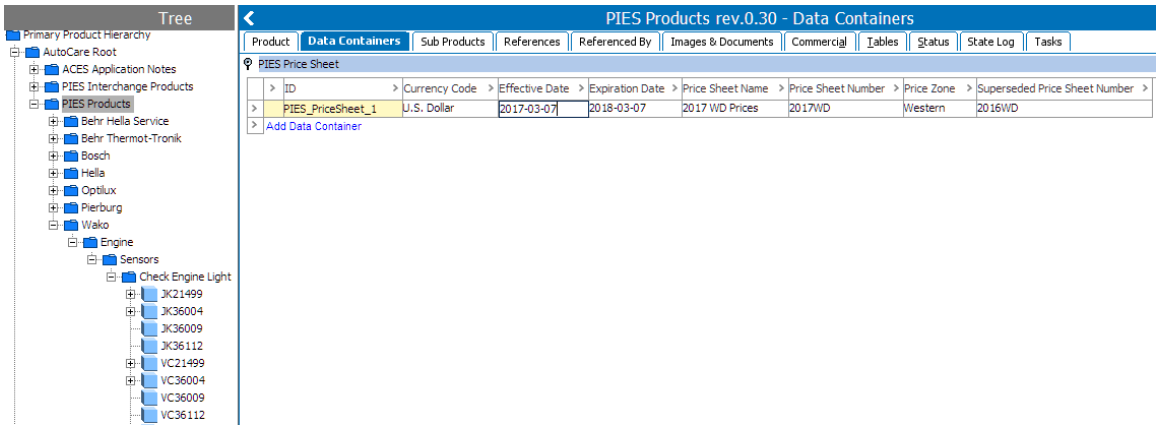
The Price Sheet Header Segment is used to define any number of price sheets. When exchanging pricing data with a trading partner, it is recommended to define at least one Price Sheet Header. The <PriceSheetNumber> defined in the Price Sheet Header segment will be referenced within the Price Segment for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




## 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).

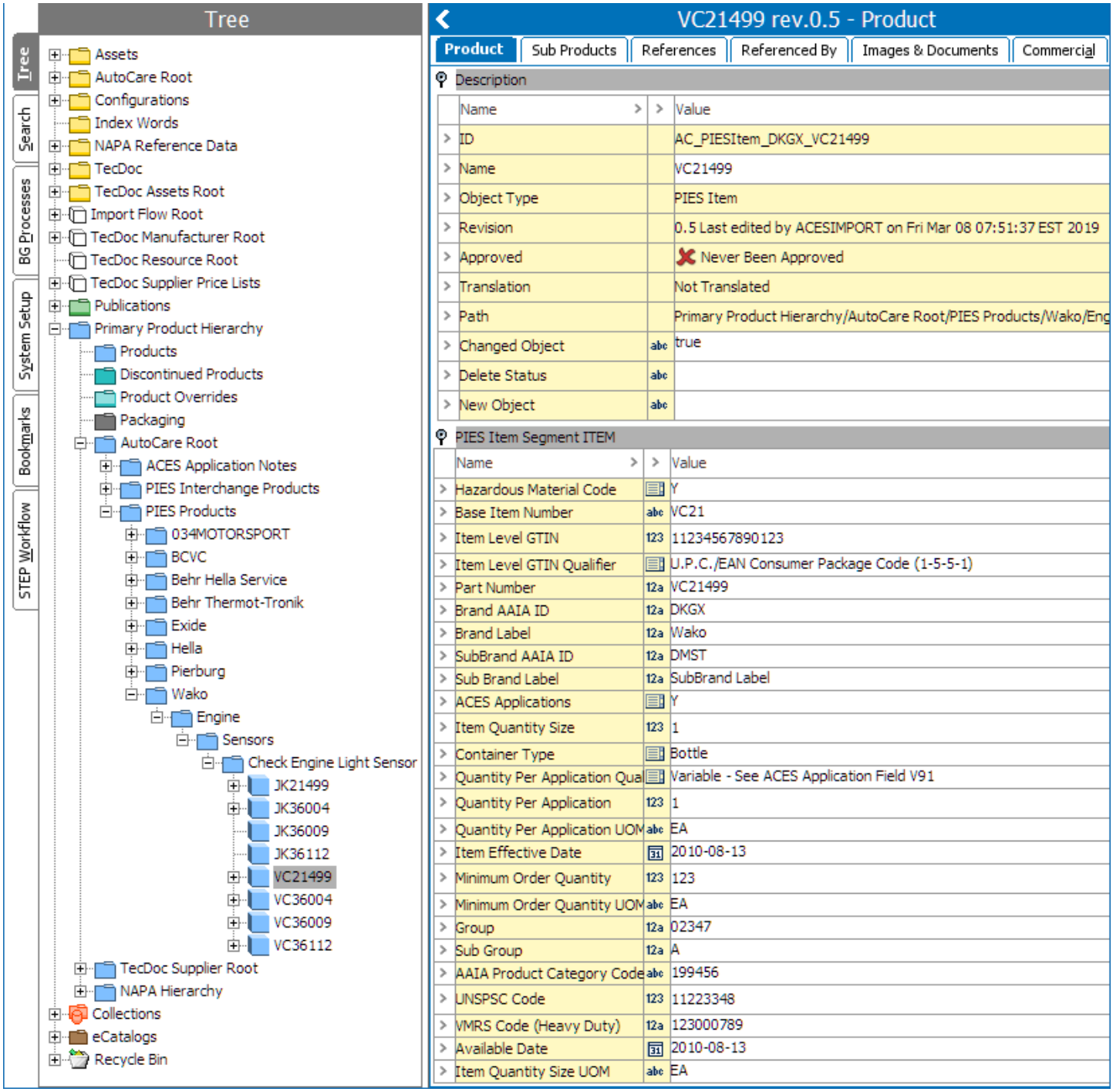
All <Item> instances are contained within the parent <Items> element. For each PIES Item to be defined, a new instance of <Item> and its child segments are initiated. Many of the child segments under <Item> are themselves segments of PIES which are defined later in this topic.

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




### 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

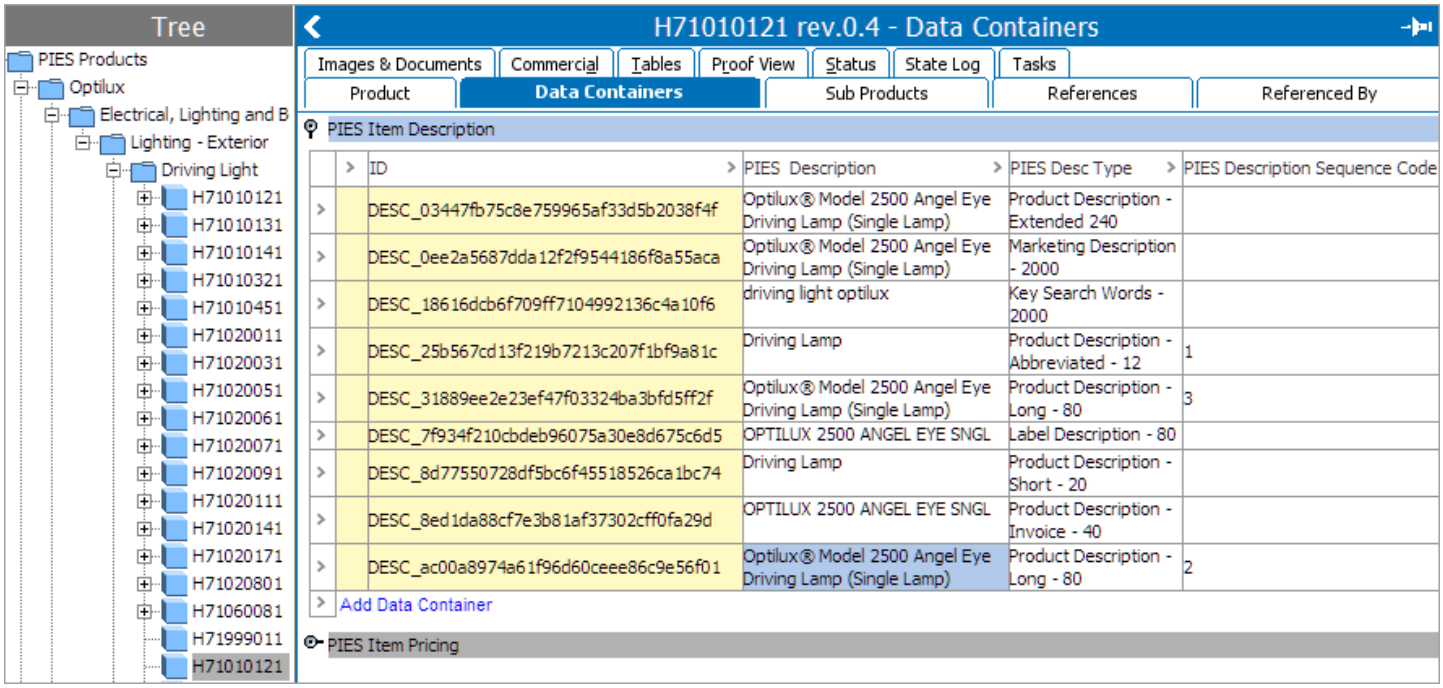
group available within the <Item> tag. To send multiple descriptions about an individual Item part number (PIES Item), a user can loop (repeat) the <Description> element, changing the 'DescriptionCode' attribute as well as the string value for the Description.

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>
```




### PIES Price Segment PRCE

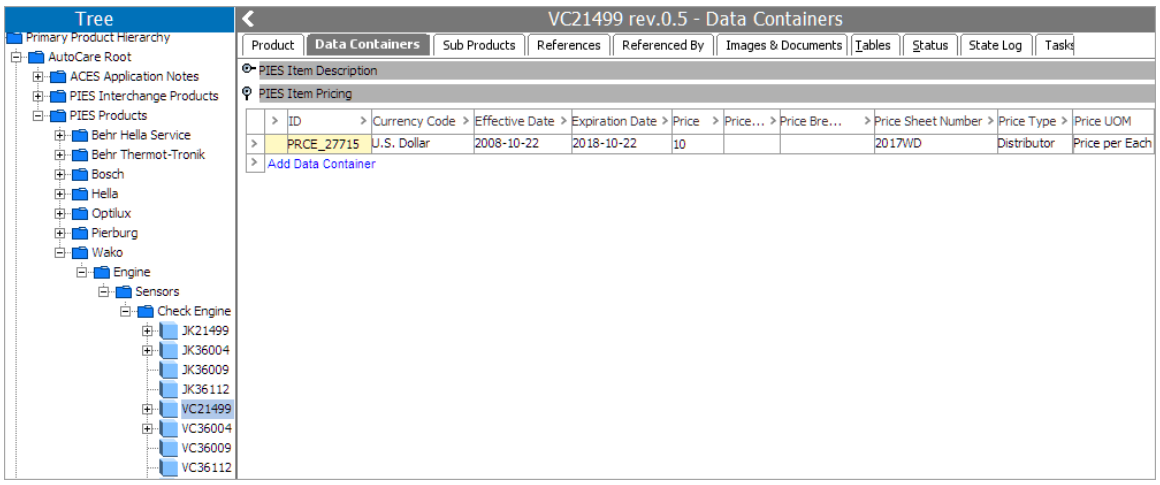
The Pricing Segment is defined by the PIES XML parent element <Prices>. Within this element group, there may be multiple instances of <Pricing> that define a single pricing instance. This element takes a 'Price Type' as an attribute indicating which type of price the following child segments needs to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

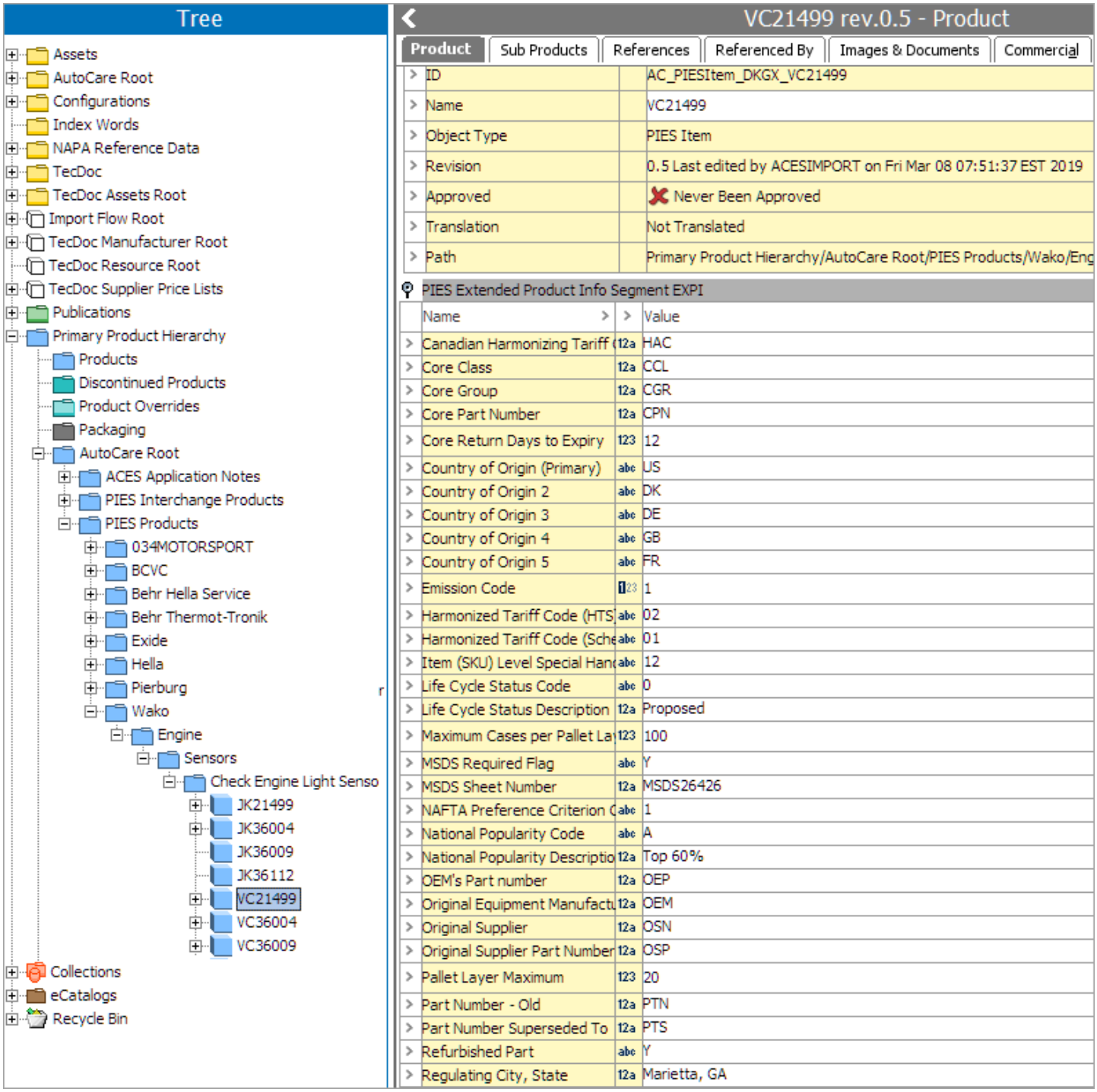
To express information covering multiple (or one) 'EXPI Codes', the <ExtendedProductInformation> element is looped once for each code. Any and all instances of <ExtendedProductInformation> are grouped under the <ExtendedInformation> element.

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

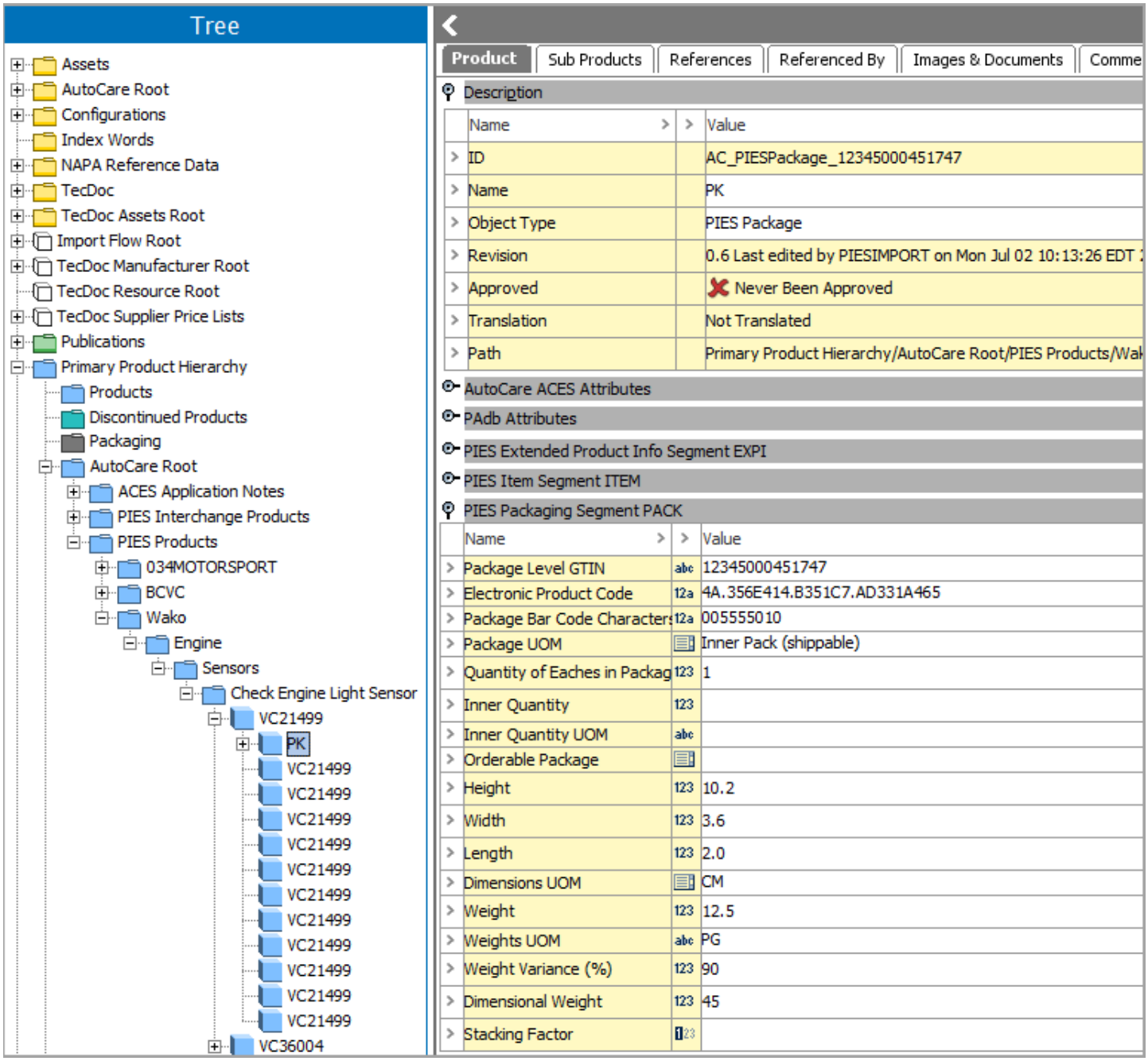
## Packaging Segment (PACK)

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>
```



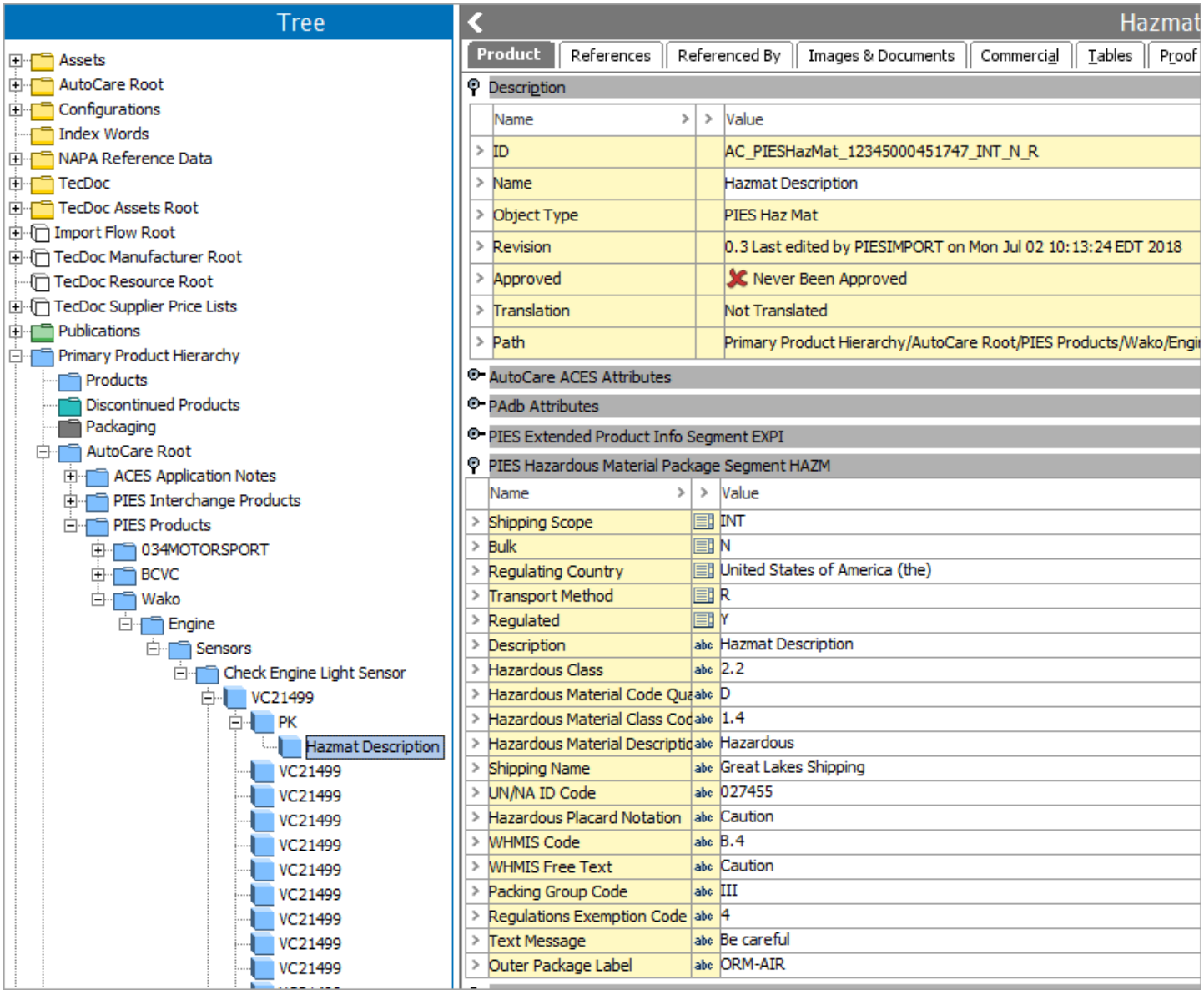

## Hazardous Material Package Segment (HAZM)

Hazmat information is relayed due to different Hazmat rules and regulations, each of which may depend on the type of package a product is using. For instance, a product may have one set of regulations (or none) at the Consumer Pack level, but a completely different set of government regulations may apply if the package is a Case or Pallet of product. Regulations may change as well depending on the transportation method used when shipping a product.

The Hazardous Material Package Segment does not repeat as a loop of <Item>. Rather, this segment is a looping structure within the <Packaging> segment.

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>
```




### 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.

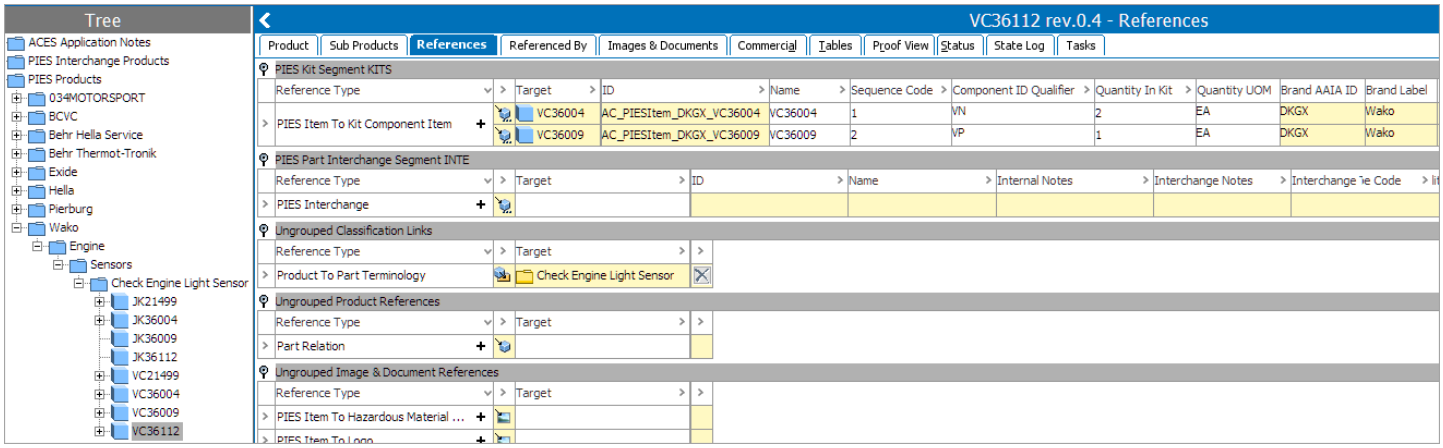
A Set is a group or collection of things that belong together, resemble one another, or are usually found together. A range of tools (such as a drill bit set, or a tool assortment), or a battery booster cable set are examples of sets.

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 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.

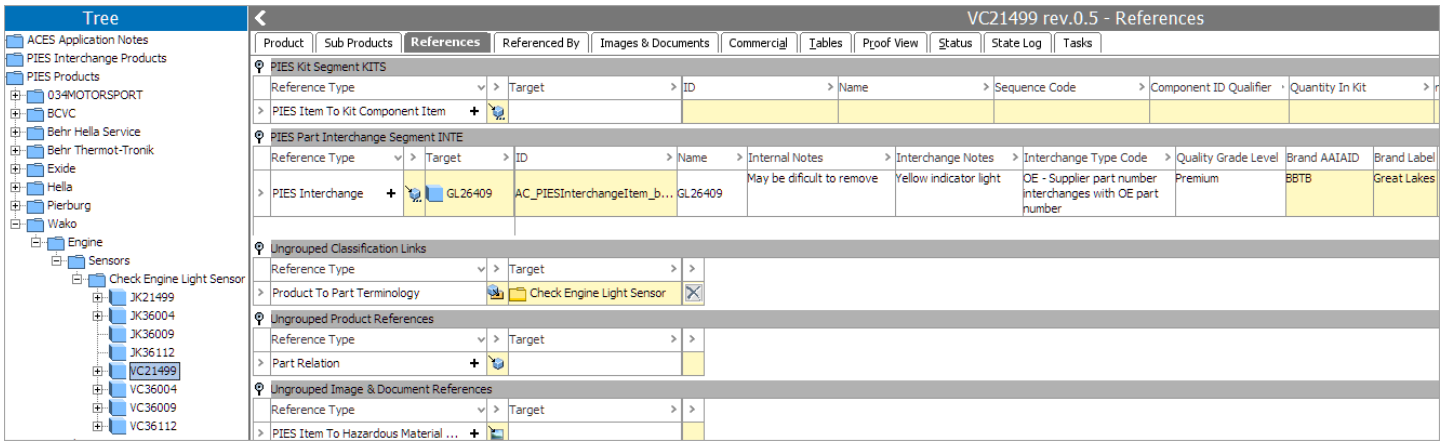
All interchange part numbers are defined by both the interchange part number and a 'Brand AAIAID'. The 'Brand AAIAID' value for the interchange supplier can be found in the Brand table.

As with some of the other segments described above, to express multiple interchanges for a PIES Item, 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

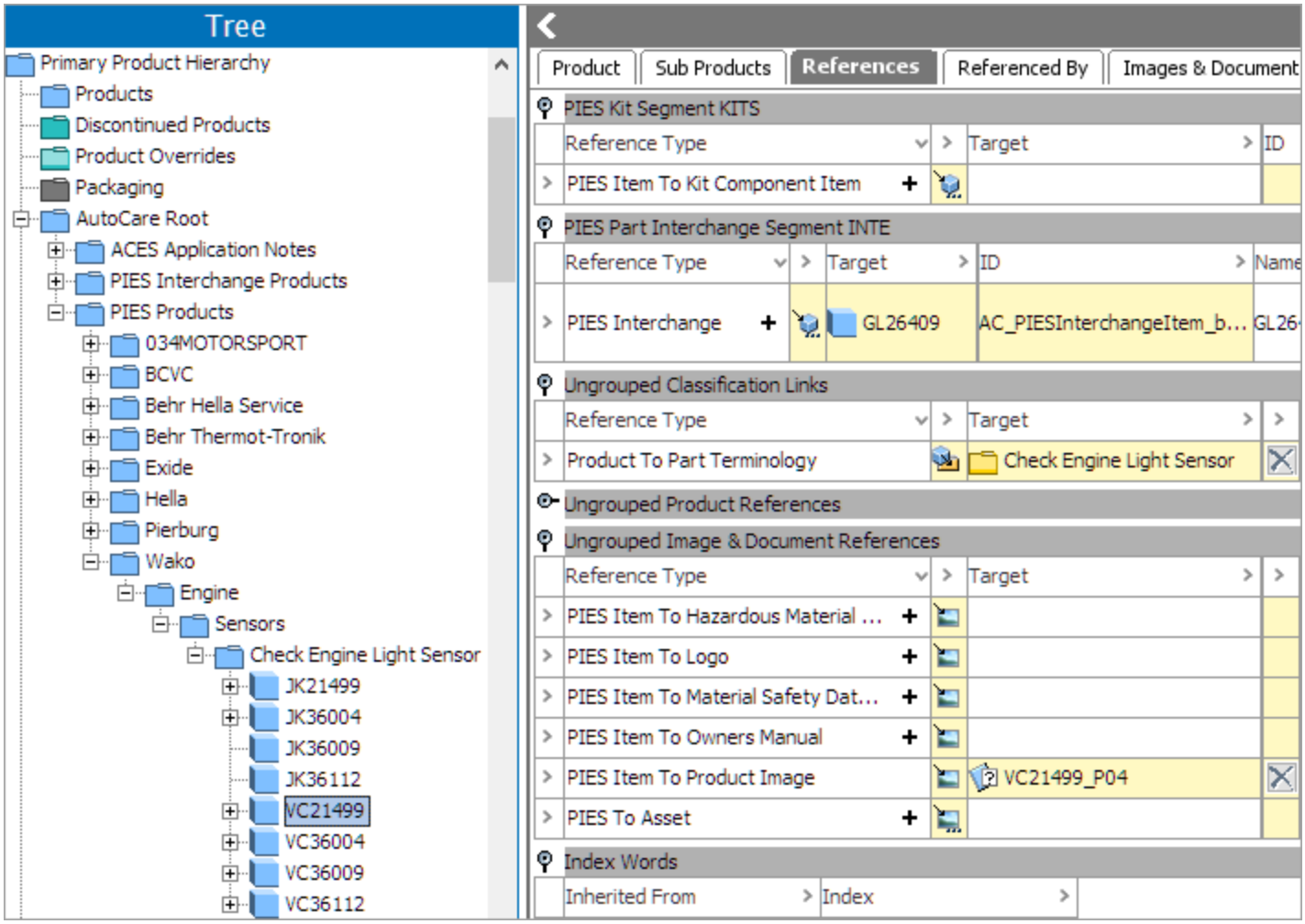
Five asset types (HMS, LGO, MSD, OWN, and P04) will get its own object type and reference type, and all other asset types will be created as PIES Asset (General) object type and referenced using 'PIES To Asset' reference type.

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

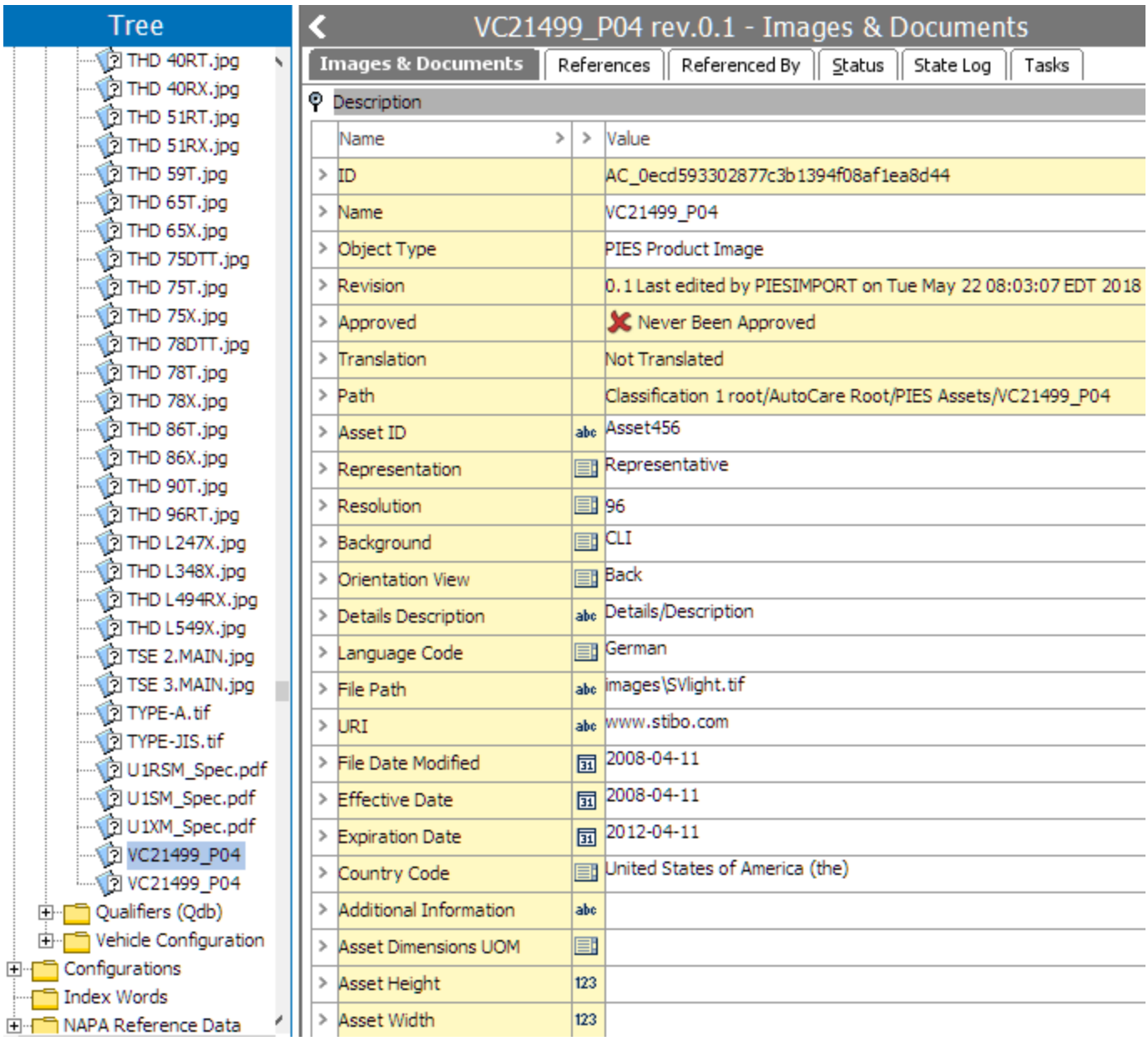
```
<DigitalAssets>
                <DigitalFileInformation MaintenanceType="A" AssetID="Asset456"
LanguageCode="EN">
                    <FileName>VC21499_P04</FileName>
                    <AssetType>P04</AssetType>
                    <Representation>R</Representation>
                    <Resolution>96</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.'




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.

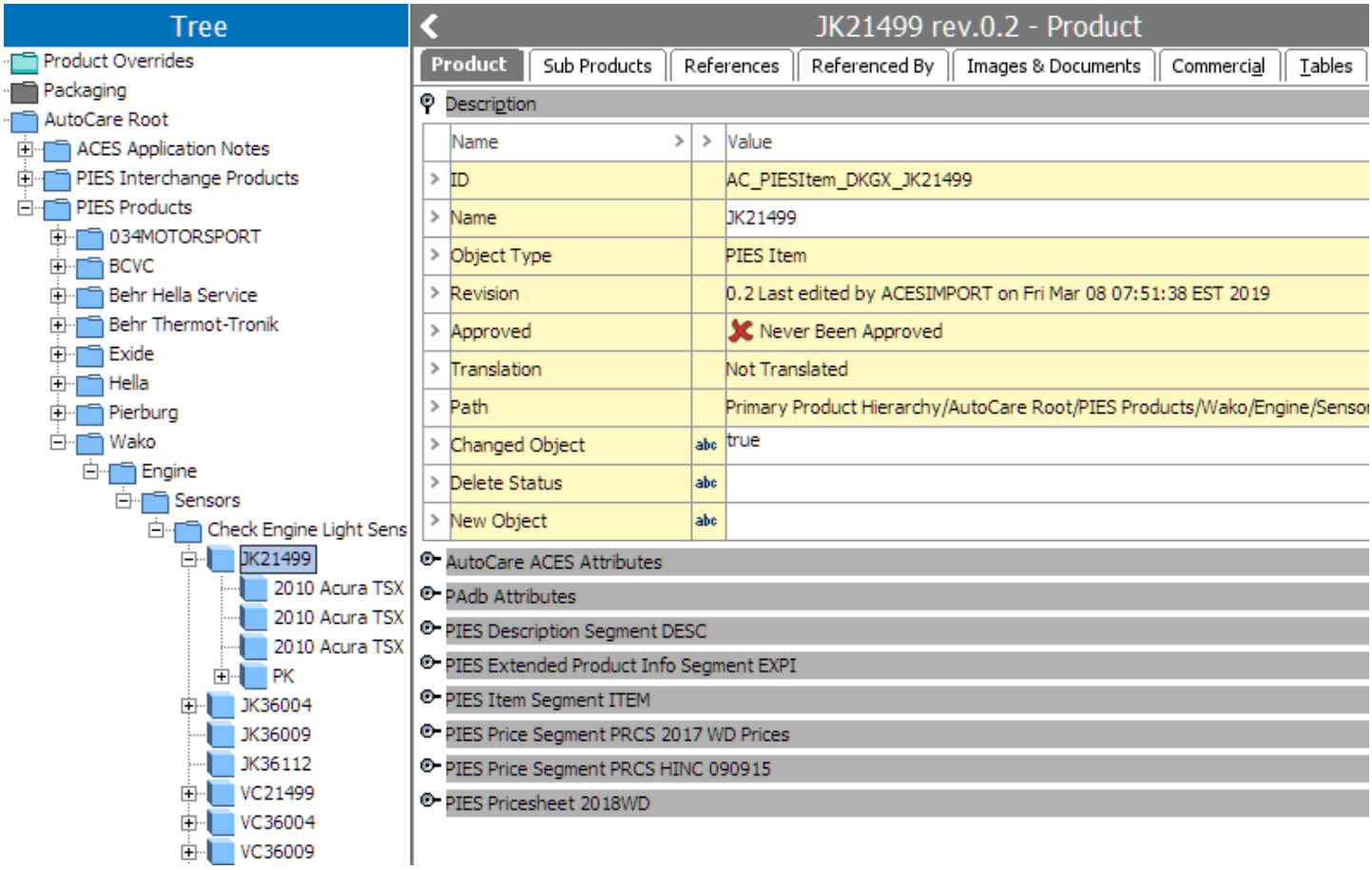



## Output

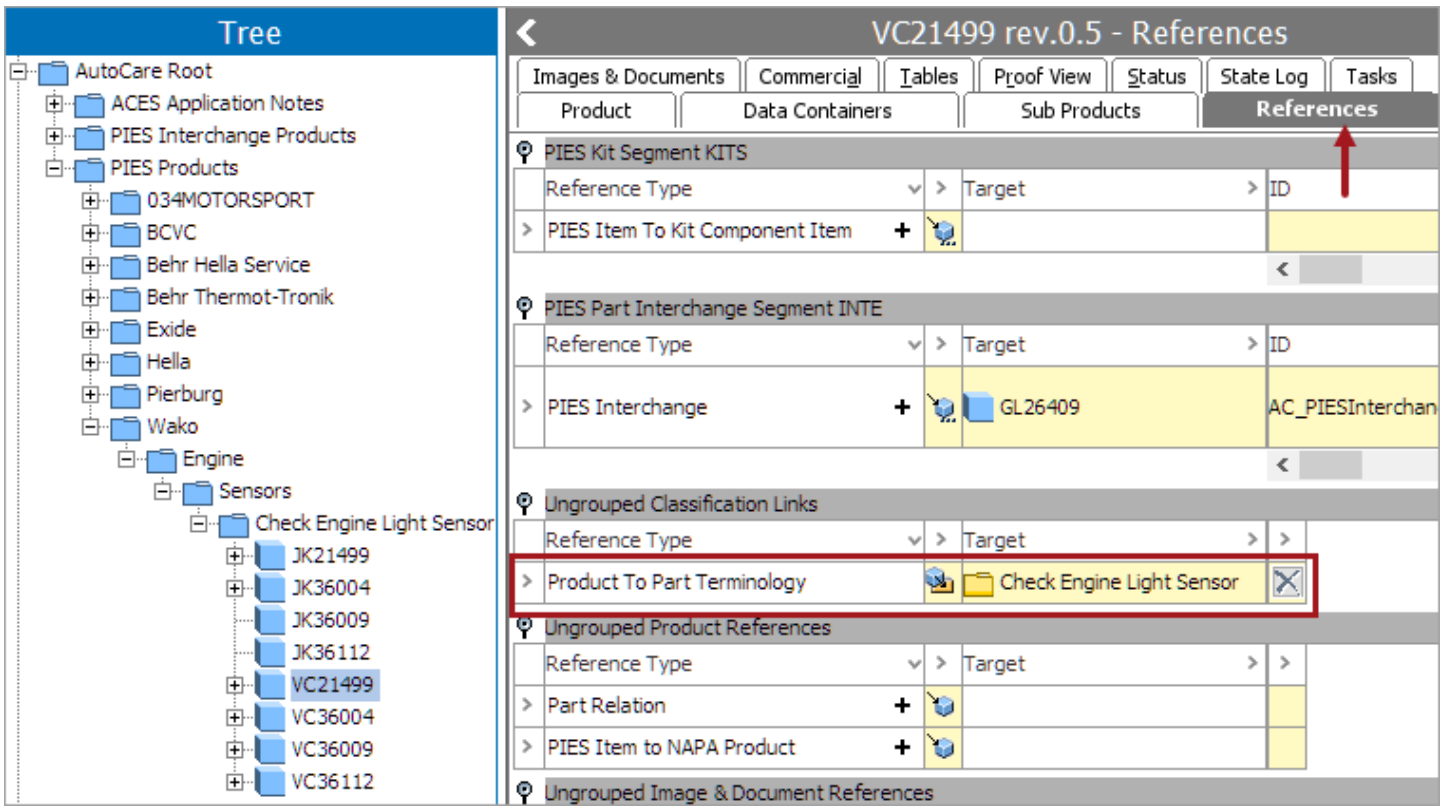
Upon successful import, a PIES Item will be created / modified. The PIES Importer do not delete any PIES Item. The Importer marks the PIES Item as new, delete, or change through change flags.

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.




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.


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, refer to 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 (containing .xml files)

| 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 'Headeasdfr.' 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. |

### 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, refer to the **Designer Access** topic within the **Web User Interfaces** 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, a Node Details screen must be created. For information about configuring the Node Details Screen, refer to the **Node Details Screen** topic within the **Web User Interfaces** section of **STEP Online Help**.

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 Homepage Widget, including additional information about working with these widgets, refer to File Loading Widget topic, and Status Selector Homepage Widget topic, both within the **Web User Interfaces** section of the **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, refer to 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.

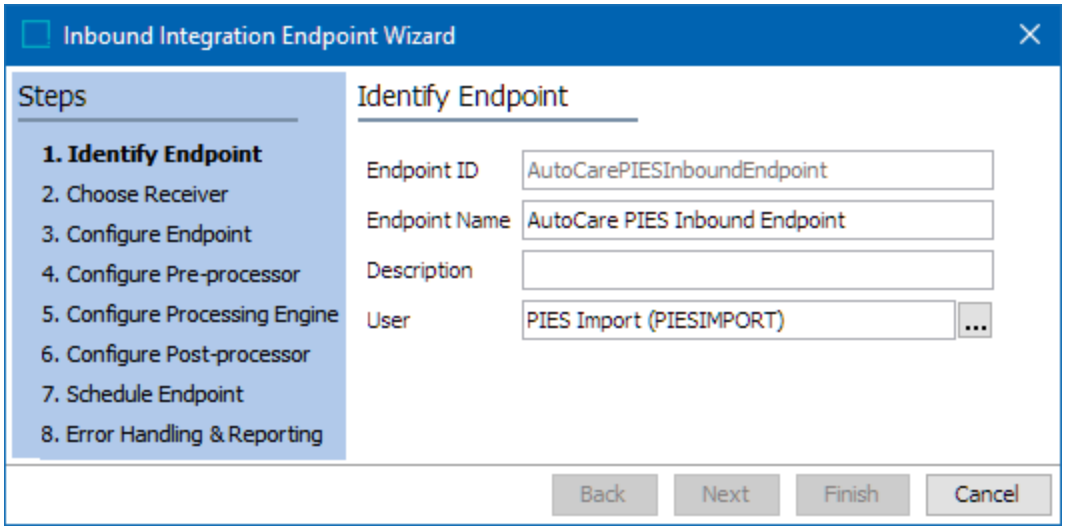
## Prerequisites

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, refer to 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.

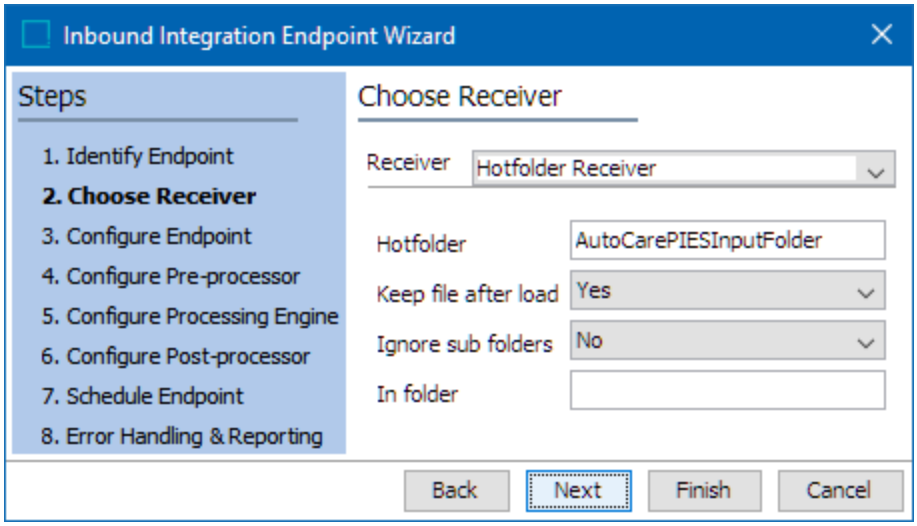
   In the example below, a 'PIES Import' user was created prior to configuring this IIEP. This is recommended to more easily track when this IIEP is responsible for changes to data.



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

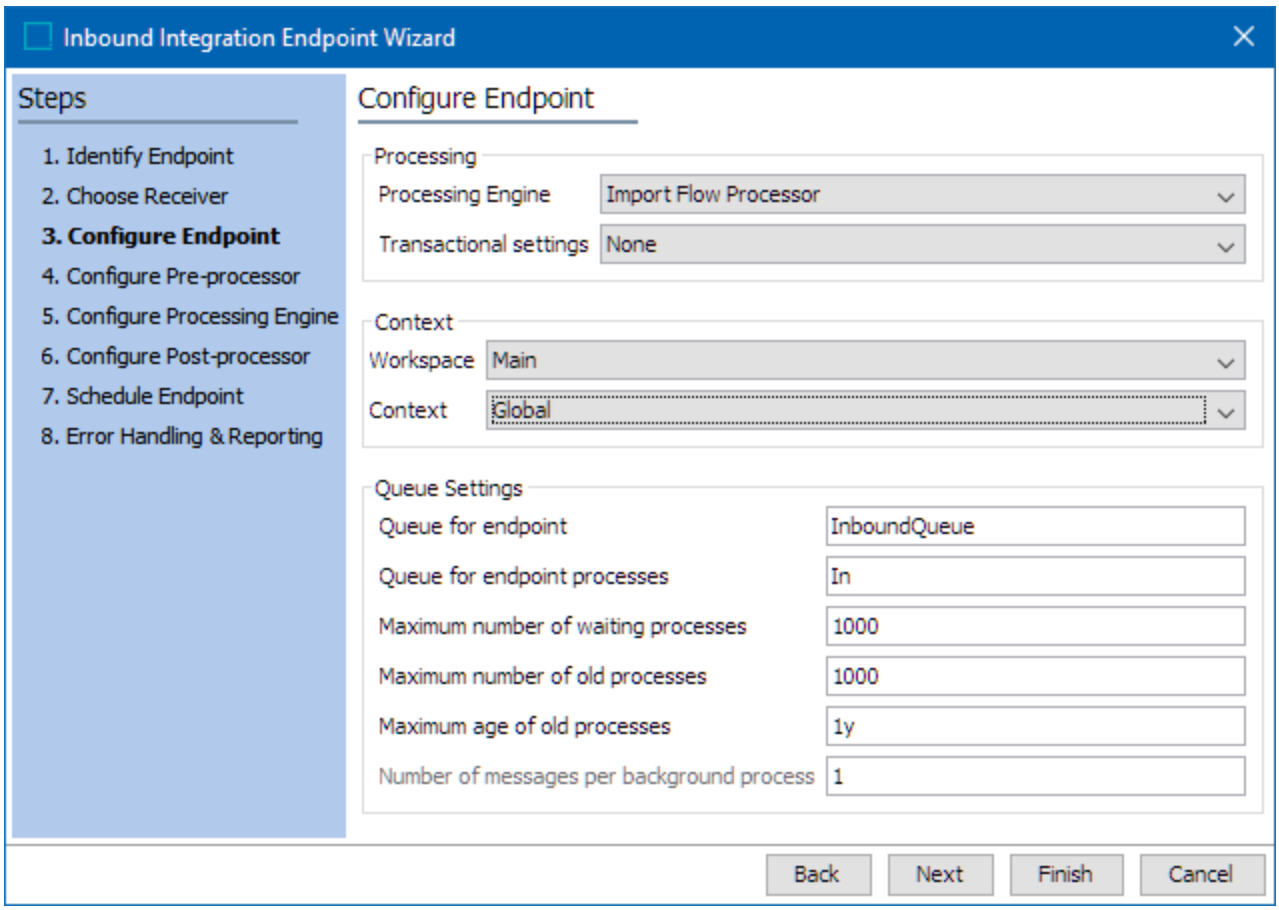
3. Click the **Next** button, and the Choose Receiver parameters will display. By default, the parameters are populated as recommended and shown below, except the Hotfolder parameter. This mandatory parameter must be populated with a hotfolder name before the Next button will enable.

> **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, refer to the **IIEP - Choose Receiver** topic within the **Inbound Integration Endpoint** section of **STEP Online Help**.

4. 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

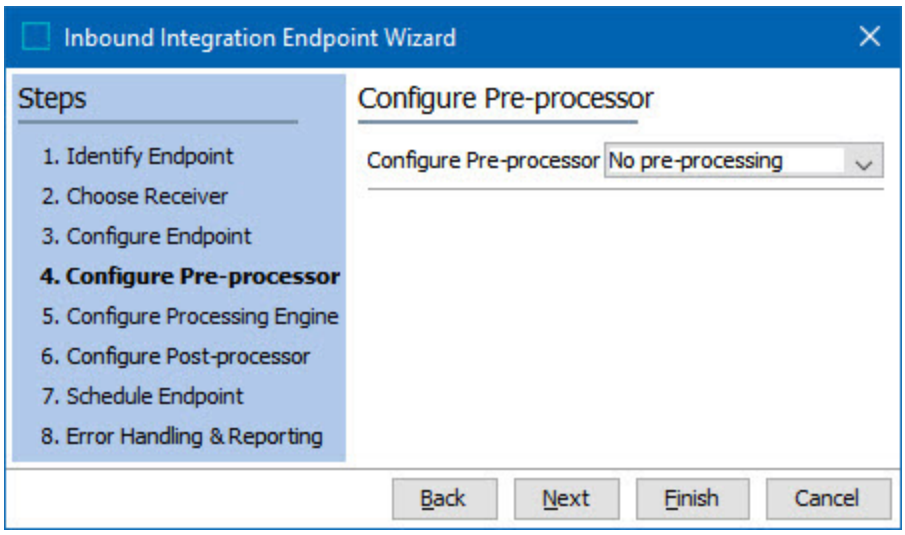



**Important:** The Import Flow Processor is only responsible for picking up files from the configured hotfolder, creating a Controller Entity object, and updating the Entity with the Import Flow State BGP and Status. The Import Flow Processor works with the Background Process Service for each workflow 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.

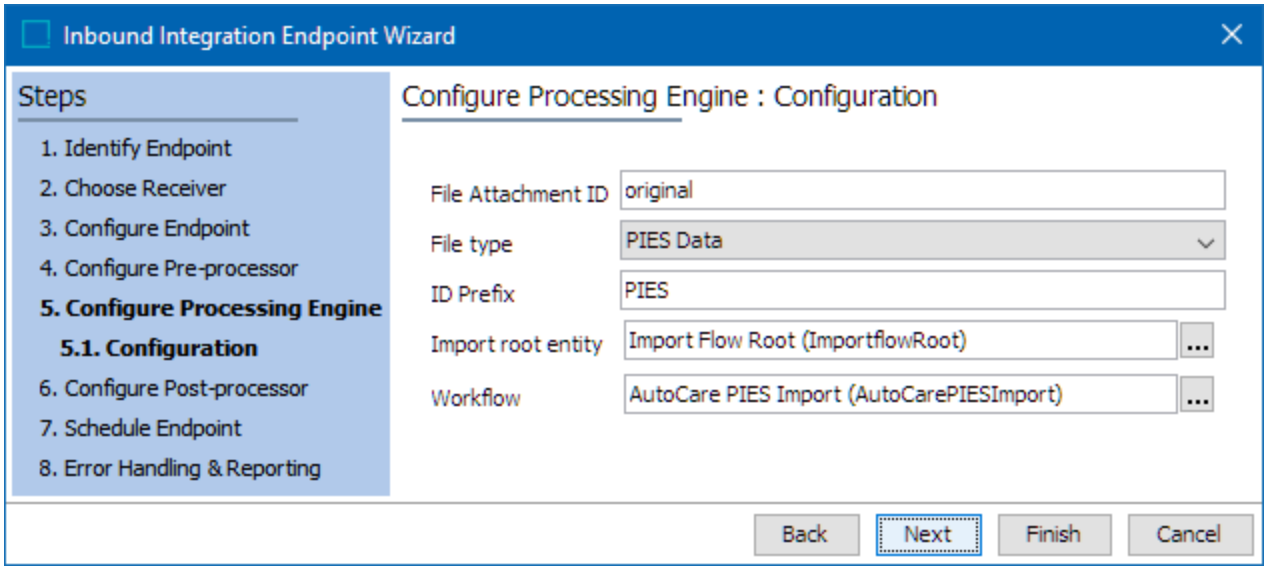
For more information about the parameters, refer to the **IIEP - Configure Endpoint** topic within the **Inbound Integration Endpoint** section of **STEP Online Help**.

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




For more information about the parameter, refer to the **IIEP - Configure Pre-processor** topic within the **Inbound Integration Endpoint** section of **STEP Online Help**.

6. 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.



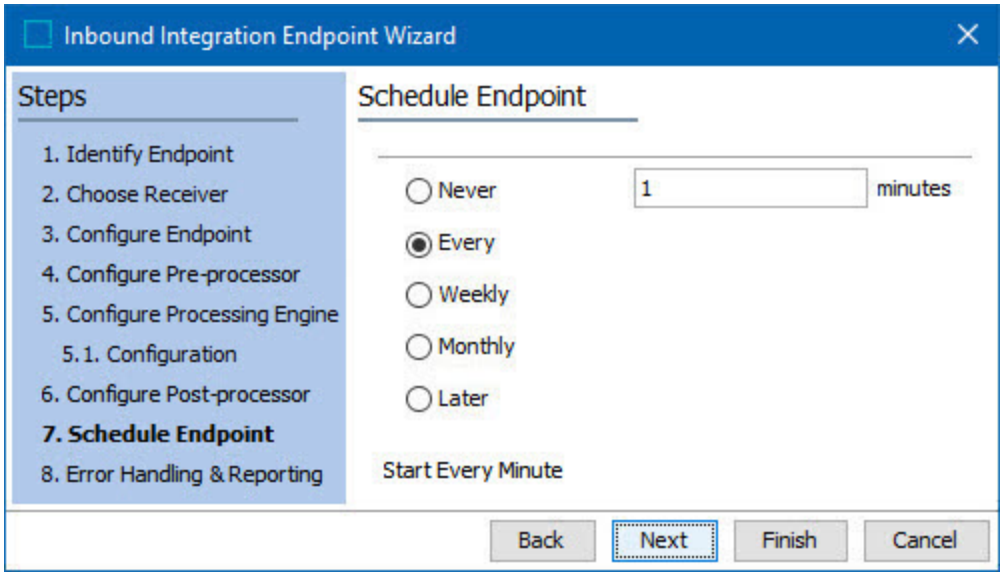

The prior Configure Endpoint step determines the options available for the Configure Processing Engine step.

- 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.

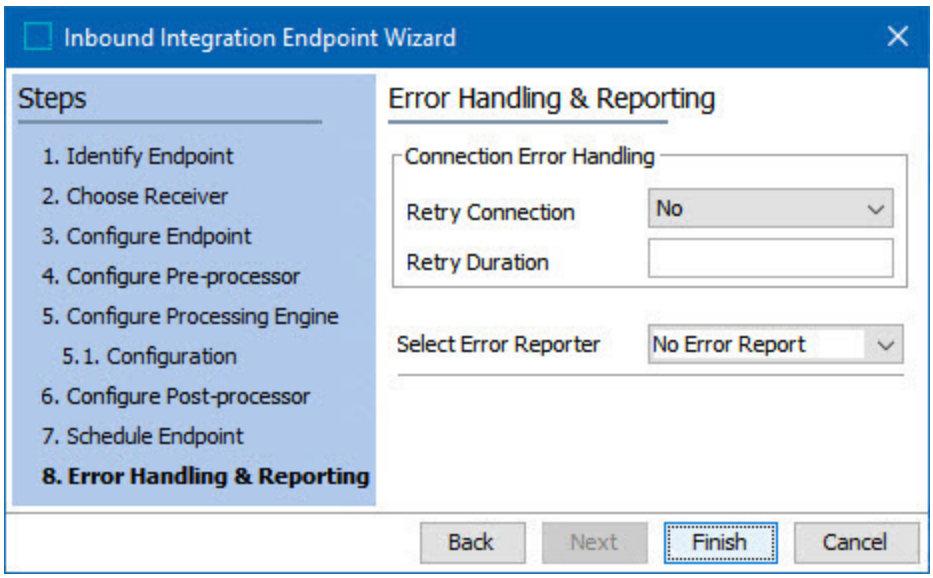
7. Click the **Next** button, and the Schedule Endpoint parameters will display. By default, 'Never' is selected. Optionally, update the values to those shown below.




> **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, refer to the **IIEP - Schedule Endpoint** topic within the **Inbound Integration Endpoint** section of **STEP Online Help**.

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




For more information about the parameters, refer to the **IIEP - Error Handling & Reporting** 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, refer to 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, refer to 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, refer to 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.

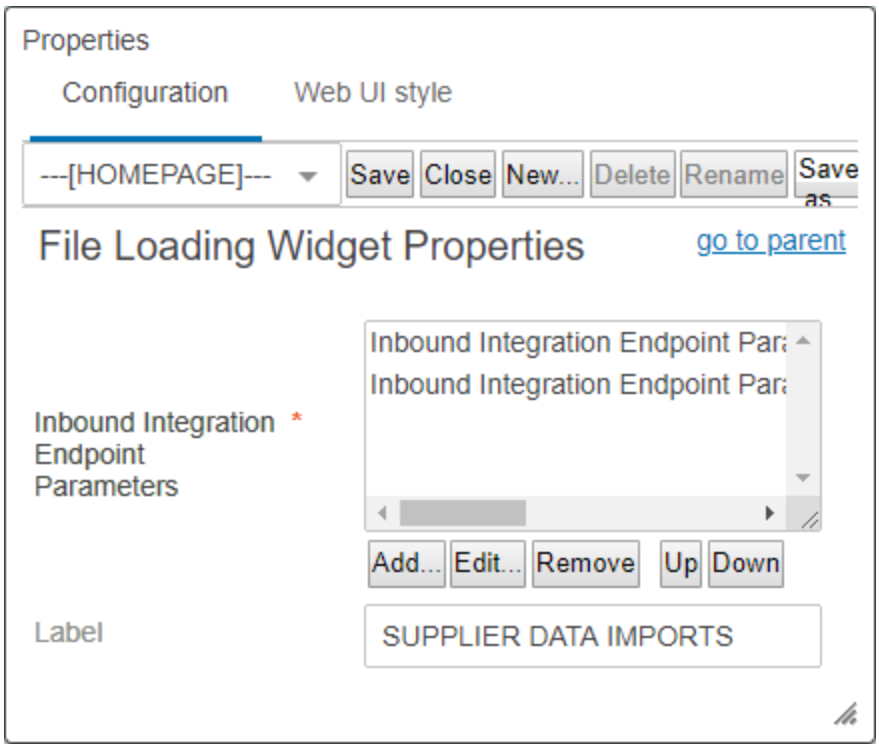
## Configuration

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

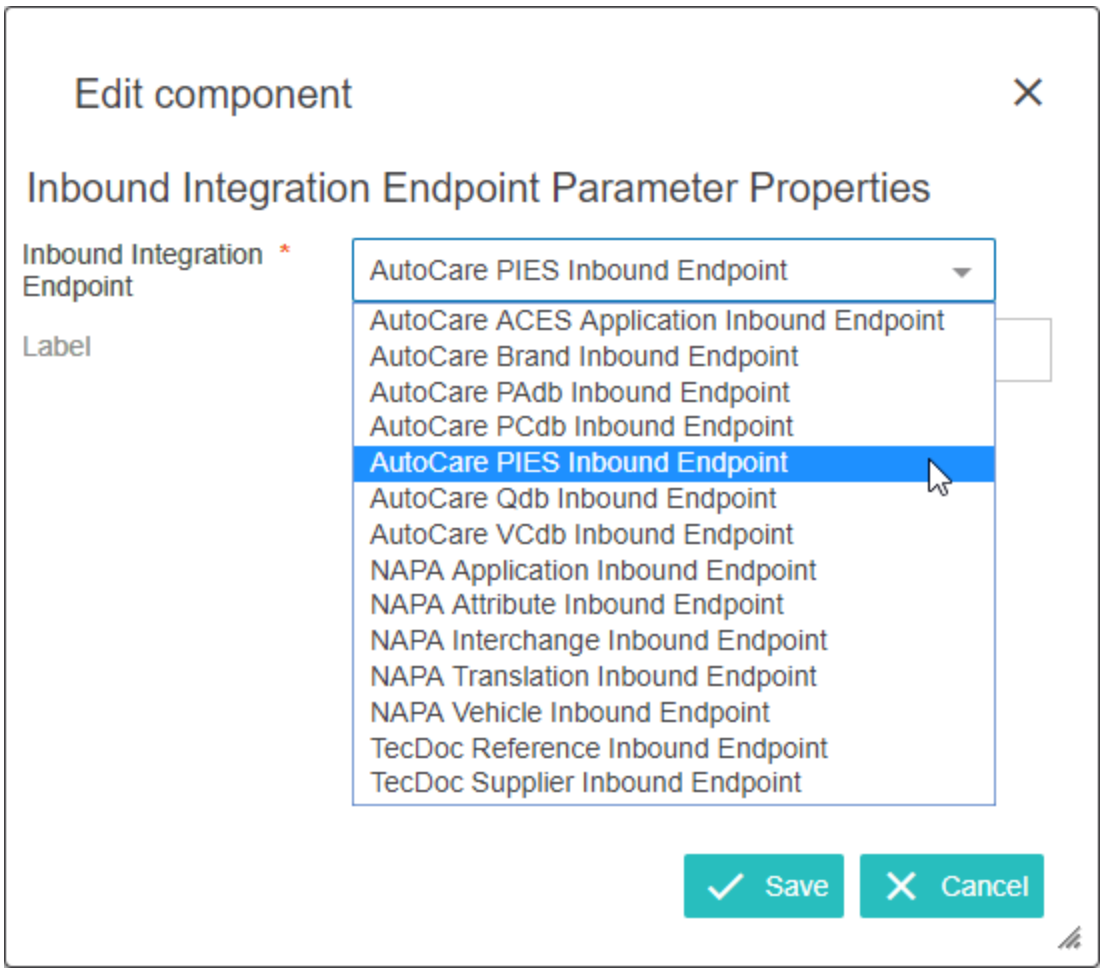
This topic describes how to configure a File Loading Widget so that users can drag and drop AutoCare PIES files onto a File Loading Widget on a Web UI Homepage.

**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, refer to 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).

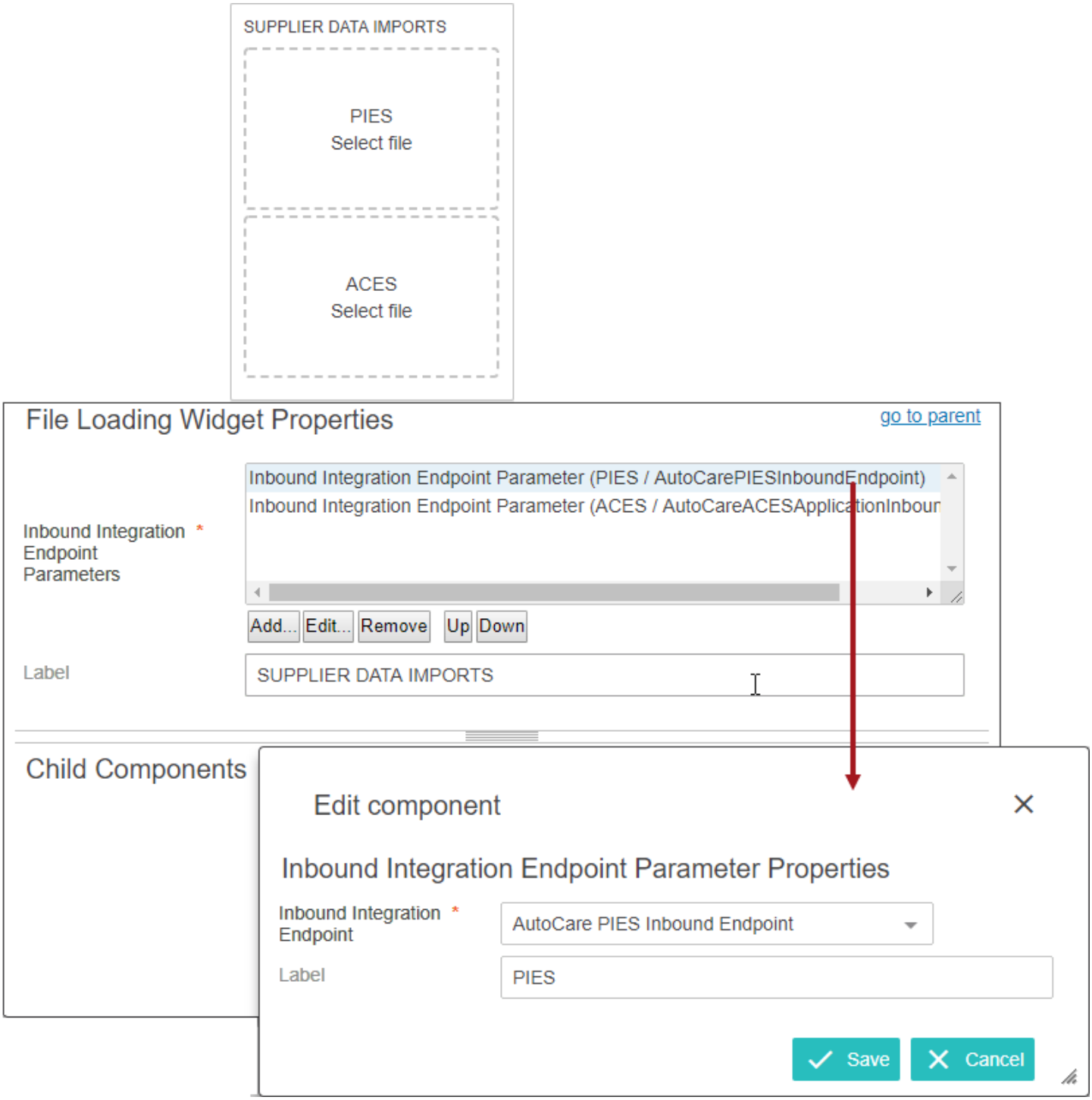



**Note:** If the desired IIEP does not display in the dropdown, then it can be created using the steps described in the **Configuring an IIEP for PIES Imports** topic.

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.




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

For general information about the File Loading Widget and Status Selector Homepage Widget, including additional information about working with these widgets, refer to File Loading Widget topic, and Status Selector Homepage Widget topic, both within the **Web User Interfaces** section of the **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, a Node Details screen must be created. For information about configuring the Node Details Screen, refer to the **Node Details Screen** topic within the **Web User Interfaces** section of **STEP Online Help**.

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, refer to 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 Homepage 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.

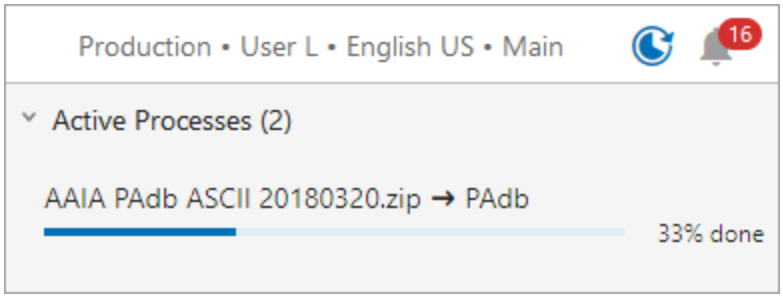
### Using the AutoCare PAdb Importer in Web UI

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, refer to 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 'Recent background processes' side panel.

   In the example below, the 'Recent background processes' side panel is expanded and the file, 'AAIA PAdb ASCII 20180320.zip' is 33% processed.




   For more information on using the 'Recent background processes' side panel, refer to the **Recent Background Processes Side Panel** 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.

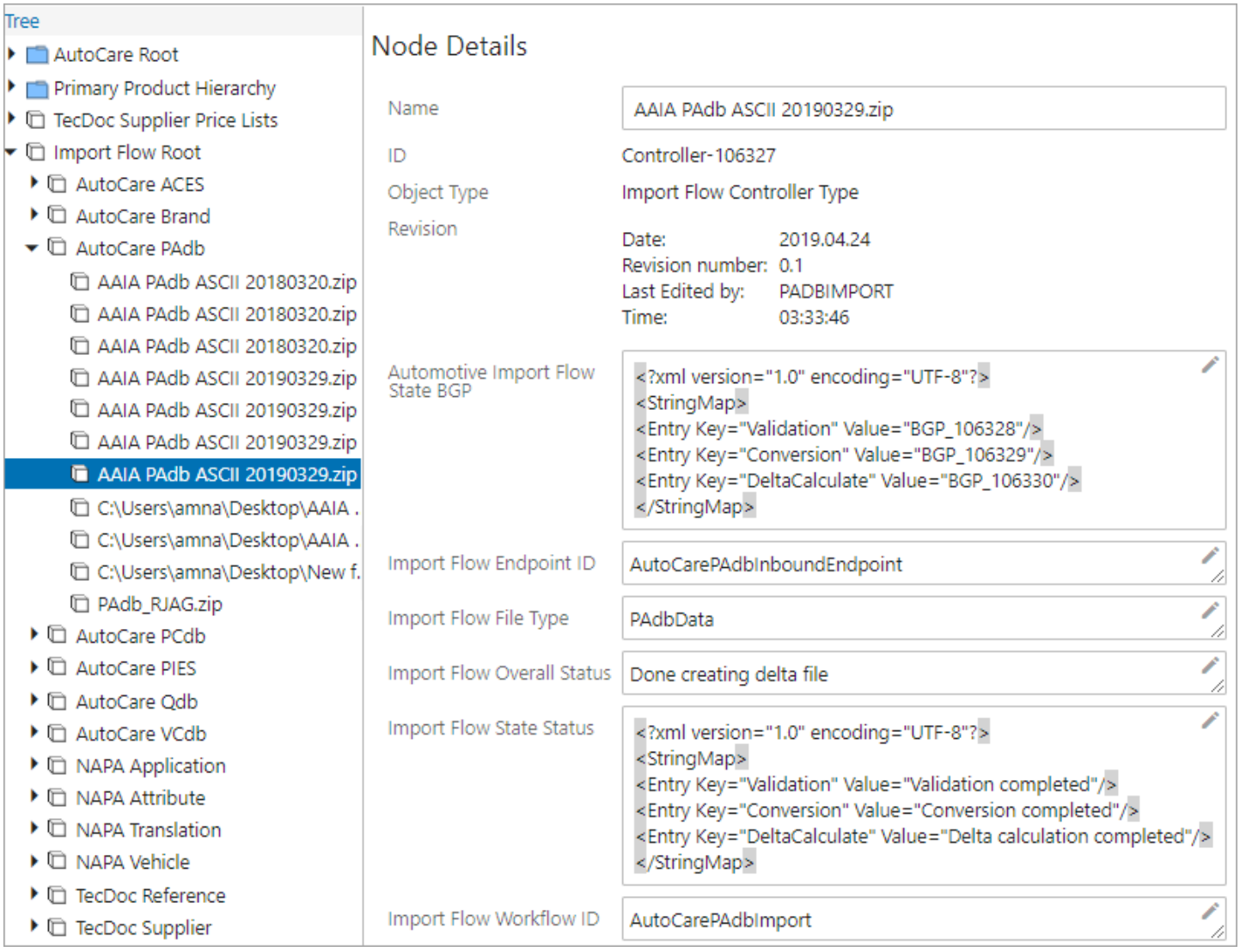
**Important:** It is critical to understand that it is only the Controller Entity that is in the workflow - the objects being acted on (created / updated / deleted) via information supplied in the import file are not in the workflow. Therefore, running standard business actions acting on current object 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, refer to the **Import State** topic within the **Default Workflow States and Functions** section.

4. If a Node 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 Node Details screen has not been configured. For information about configuring the Node Details Screen, refer to the **Node Details Screen** topic within the **Web User Interfaces** section of **STEP Online Help**.

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.



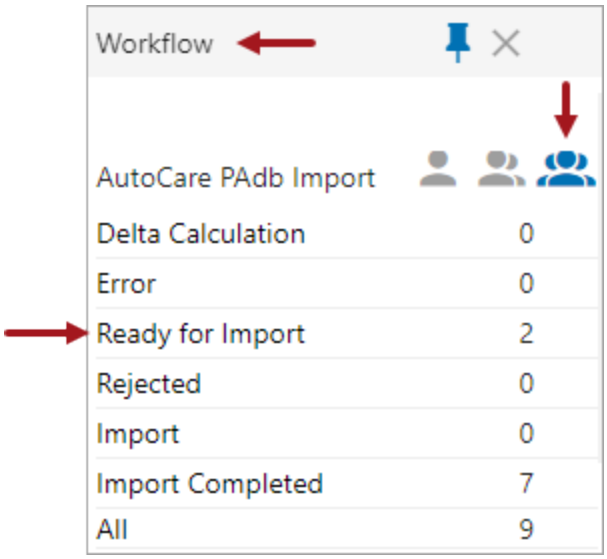

- **Name:** STEP Name of the Controller Entity. This is generated from the original name of the uploaded file.

- **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 Homepage Widget on the Web UI homepage and/or the Status Selector component on the left side panel.

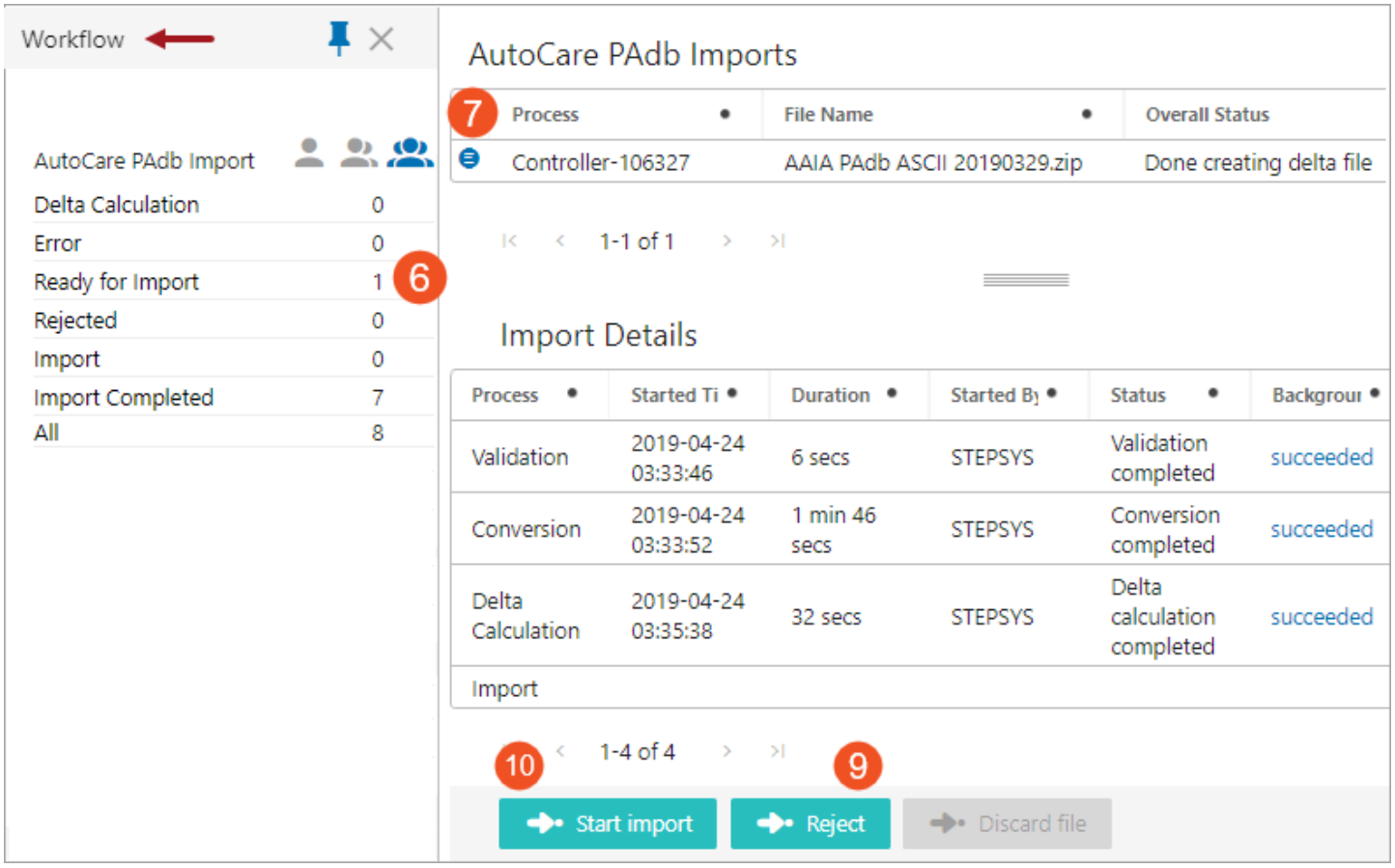
   In the example below, the Workflow panel is expanded, and the **Triple user button** is selected so that all items assigned to any user are displayed. Notice that six file is in the 'Ready for Import' state, and nineteen files in the 'Import Completed' state, for a total of twenty five files.

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

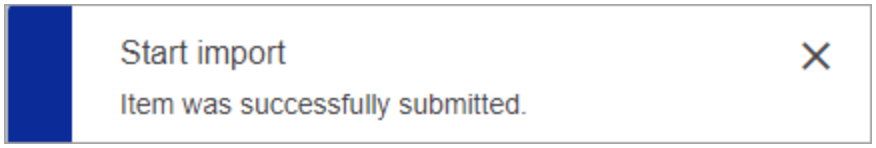



6. Clicking on the **Ready for Import** state within the Status Selector homepage and/or the left side panel widget will navigate the user to the AutoCare PAdb Import Controller Screen with only those Controllers in the 'Ready for Import' state displayed.
7. Clicking the Controller icon displays the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

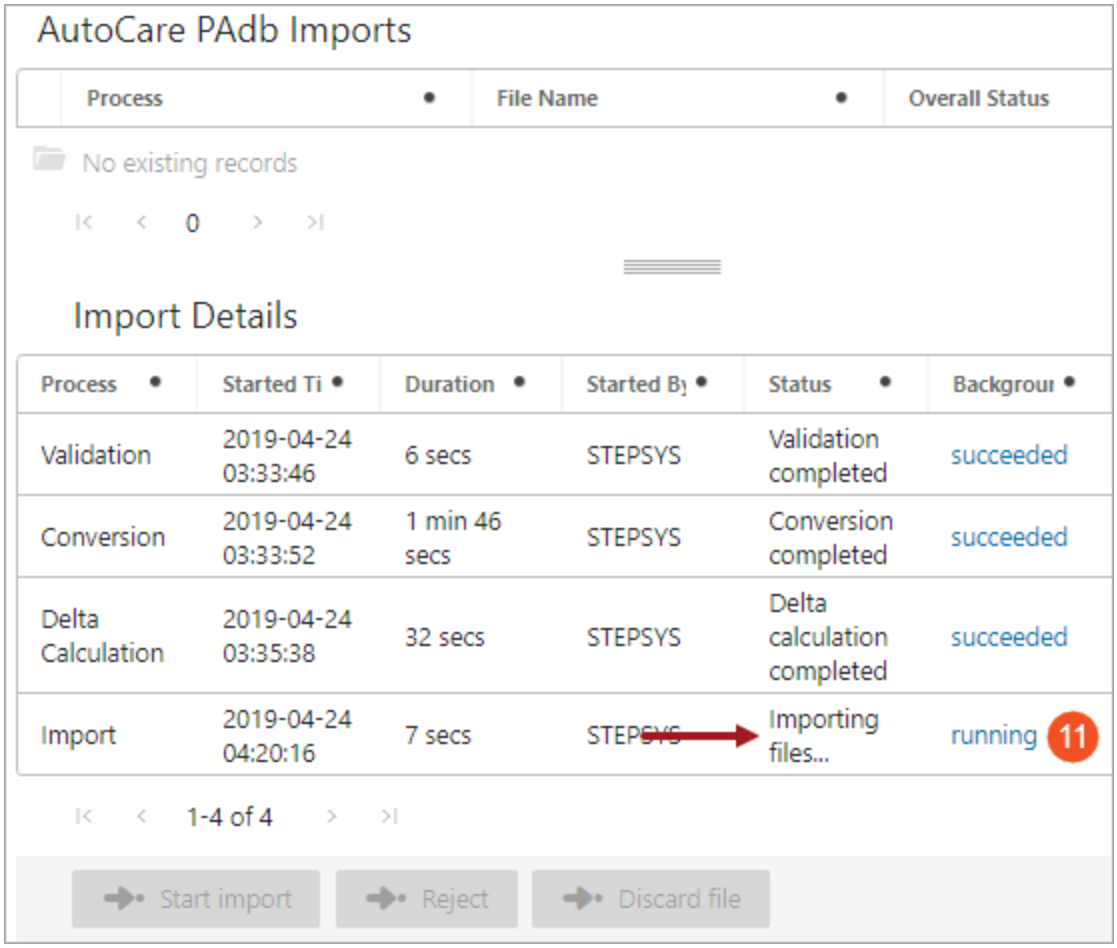



8. Optionally, before starting the import, clicking on the 'completedwitherrors' Background Process Link (only displayed when there is any error) will display the Background Process Details screen where users can view or download the details of the validation process.
9. Click the **Reject** button to move the Controller to the Rejected state. For more information, refer to 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, refer to 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 Homepage Widget on the Web UI homepage and/or the Status Selector component in the left side panel. More detailed progress can be monitored in the import process Status column that is available within the Import Controller Screen.

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.'




11. 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, refer to the **Import Completed State** topic within the **Import Framework** section.

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

    For more information on automotive workflow states, refer to 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, refer to 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 that 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_ |

| Validation Rule | Occurs | When Validation Fails | Failed Validation Message Example |
|---|---|---|---|
| | | | 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, refer to the **Designer Access** topic within the **Web User Interfaces** 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, a Node Details screen must be created. For information about configuring the Node Details Screen, refer to the **Node Details Screen** topic within the **Web User Interfaces** section of **STEP Online Help**.

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 Homepage Widget, including additional information about working with these widgets, refer to File Loading Widget topic, and Status Selector Homepage Widget topic, both within the **Web User Interfaces** section of the **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, refer to 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.

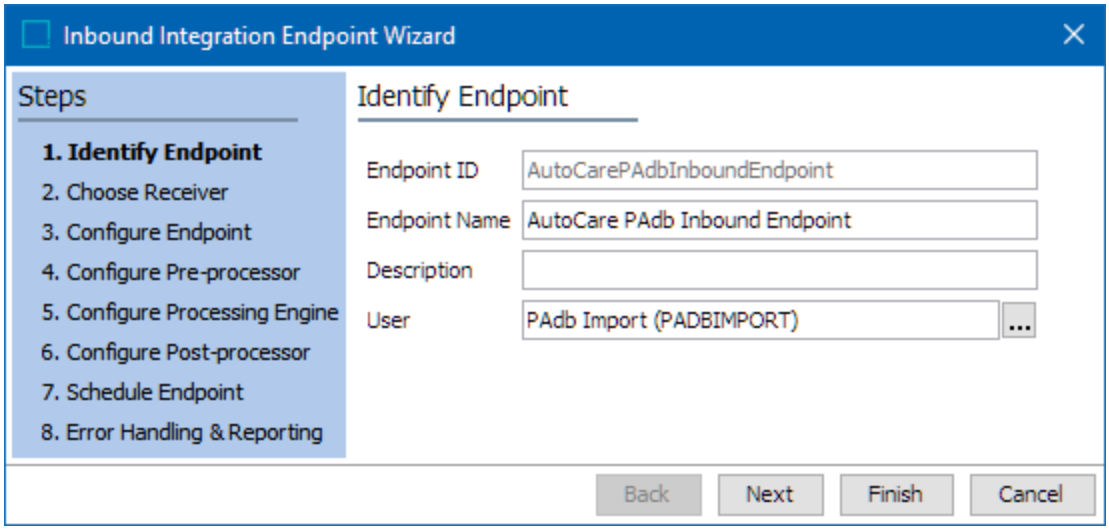
## Prerequisites

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, refer to 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.

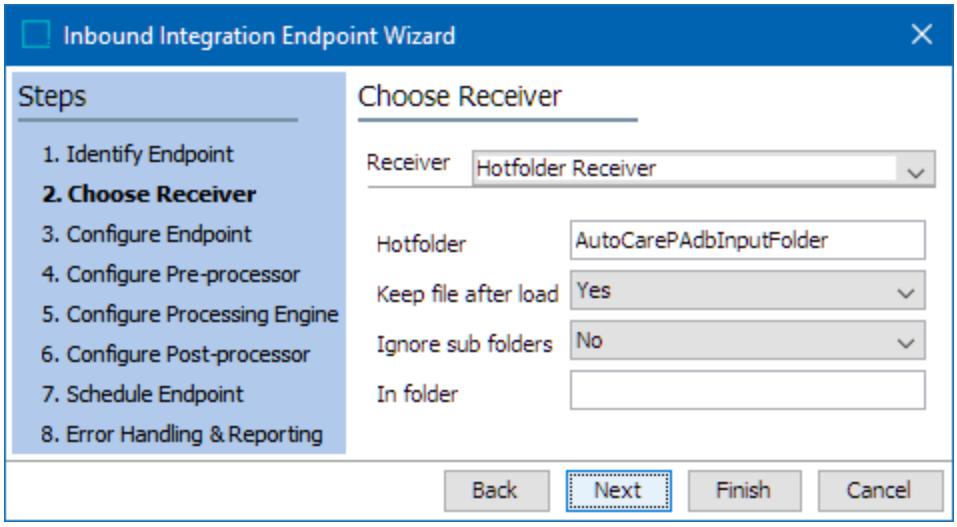
   In the example below, a 'PAdb Import' user was created prior to configuring this IIEP. This is recommended to more easily track when this IIEP is responsible for changes to data.



For more information about the parameters, refer to the **IIEP - Identify Endpoint** topic within the **Data Exchange** section of **STEP Online Help**.

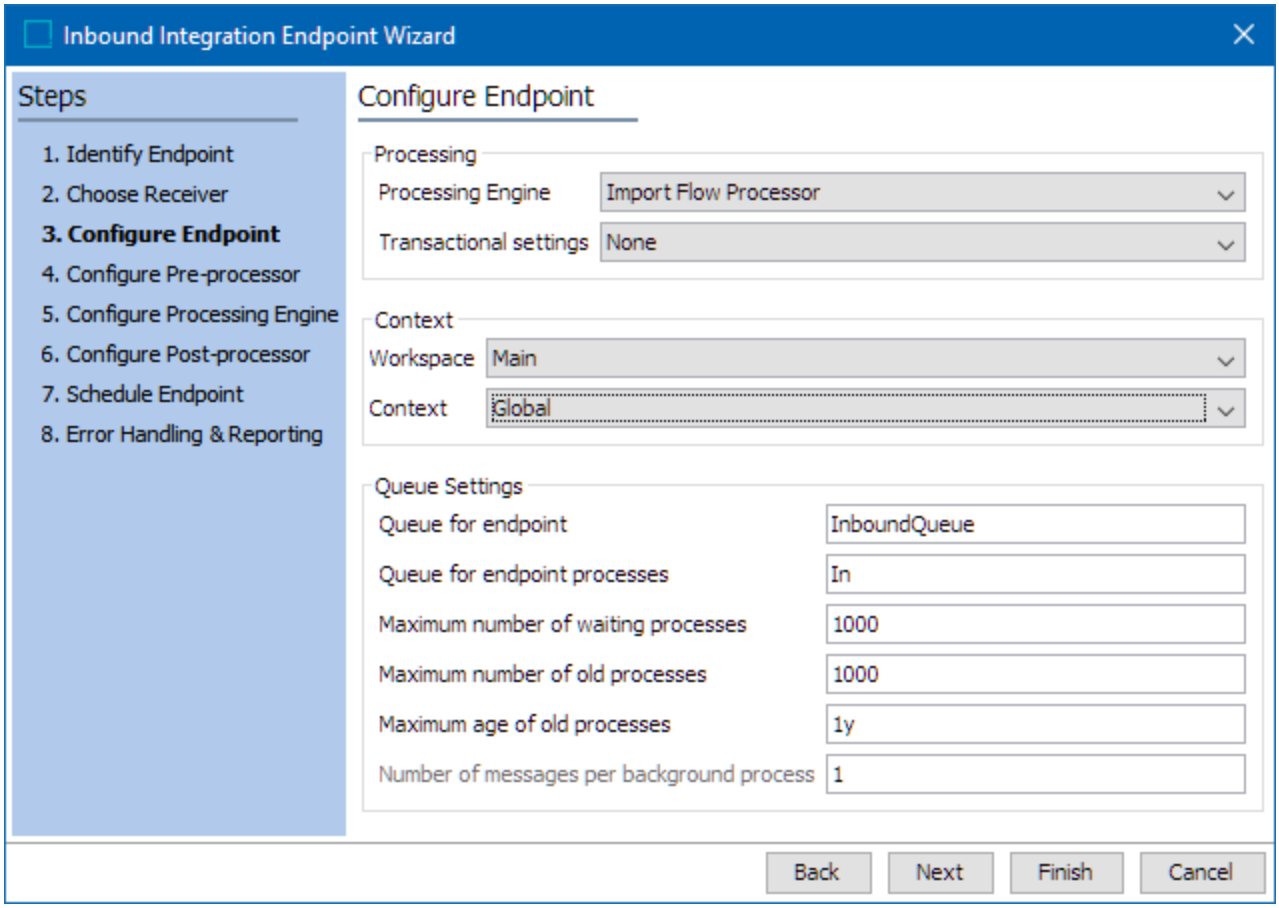
3. Click the **Next** button, and the Choose Receiver parameters will display. By default, the parameters are populated as recommended and shown below, except the Hotfolder parameter. This mandatory parameter must be populated with a hotfolder name before the Next button will enable.

> **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, refer to the **IIEP - Choose Receiver** topic within the **Data Exchange** section of **STEP Online Help**.

4. 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

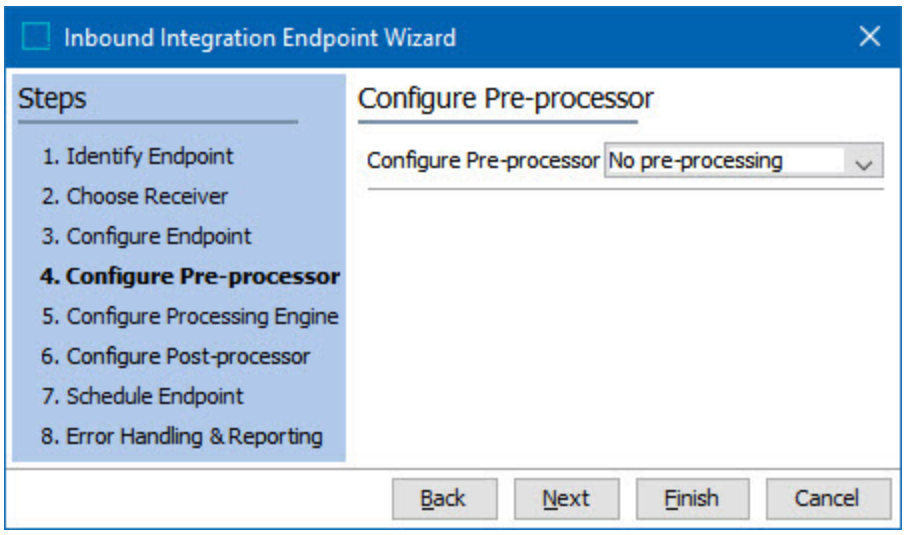



**Important:** The Import Flow Processor is only responsible for picking up files from the configured hotfolder, creating a Controller Entity object, and updating the Entity with the Import Flow State BGP and Status. The Import Flow Processor works with the Background Process Service for each workflow 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.

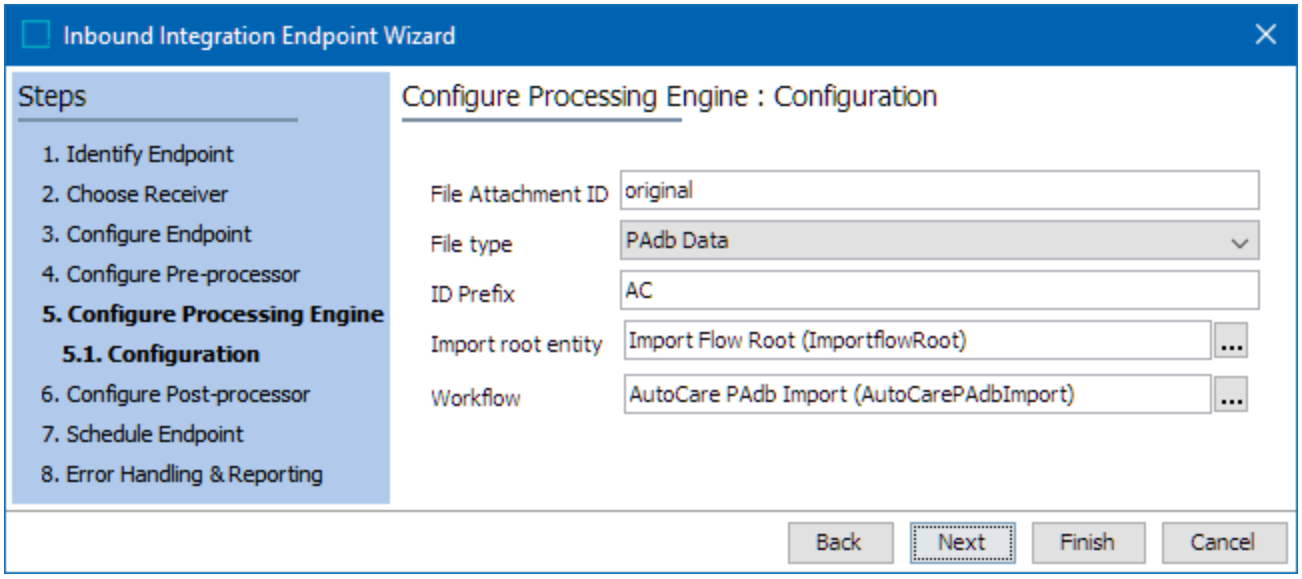
For more information about the parameters, refer to the **IIEP - Configure Endpoint** topic within the **Data Exchange** section of **STEP Online Help**.

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




For more information about the parameter, refer to the **IIEP - Configure Pre-processor** topic within the **Data Exchange** section of **STEP Online Help**.

6. 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.



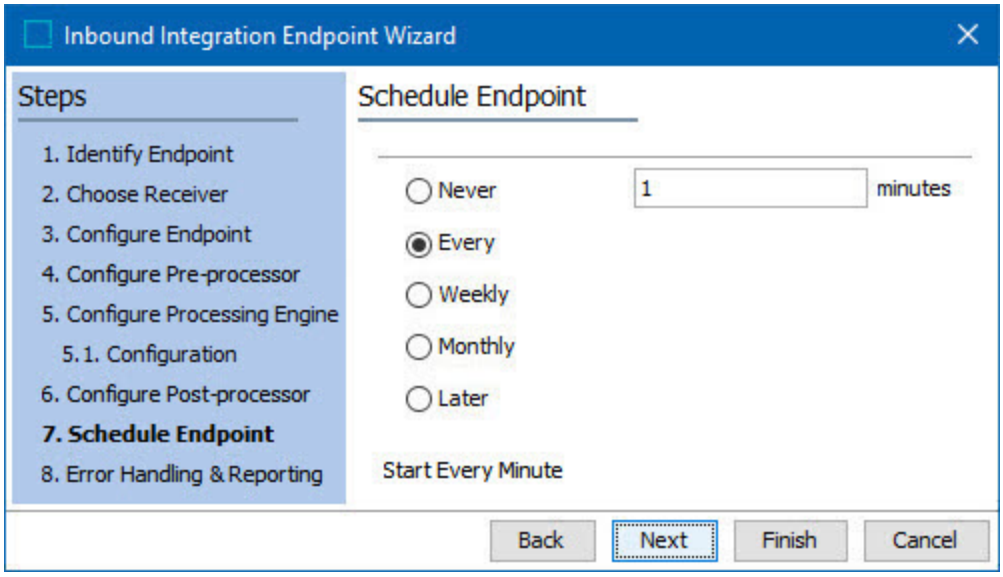

The prior Configure Endpoint step determines the options available for the Configure Processing Engine step.

- 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.

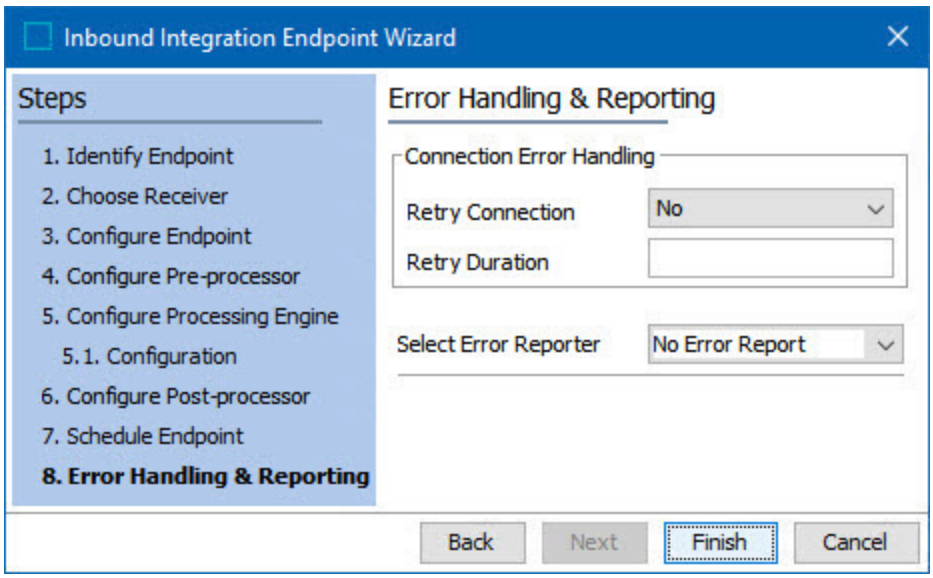
7. Click the **Next** button, and the Schedule Endpoint parameters will display. By default, 'Never' is selected. Optionally, update the values to those shown below.




**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, refer to the **IIEP - Schedule Endpoint** topic within the **Data Exchange** section of **STEP Online Help**.

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




For more information about the parameters, refer to the **IIEP - Error Handling & Reporting** topic within the **Data Exchange** 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, refer to the **Running an Inbound Integration Endpoint** topic within the **Data Exchange** 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, refer to 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, refer to 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.

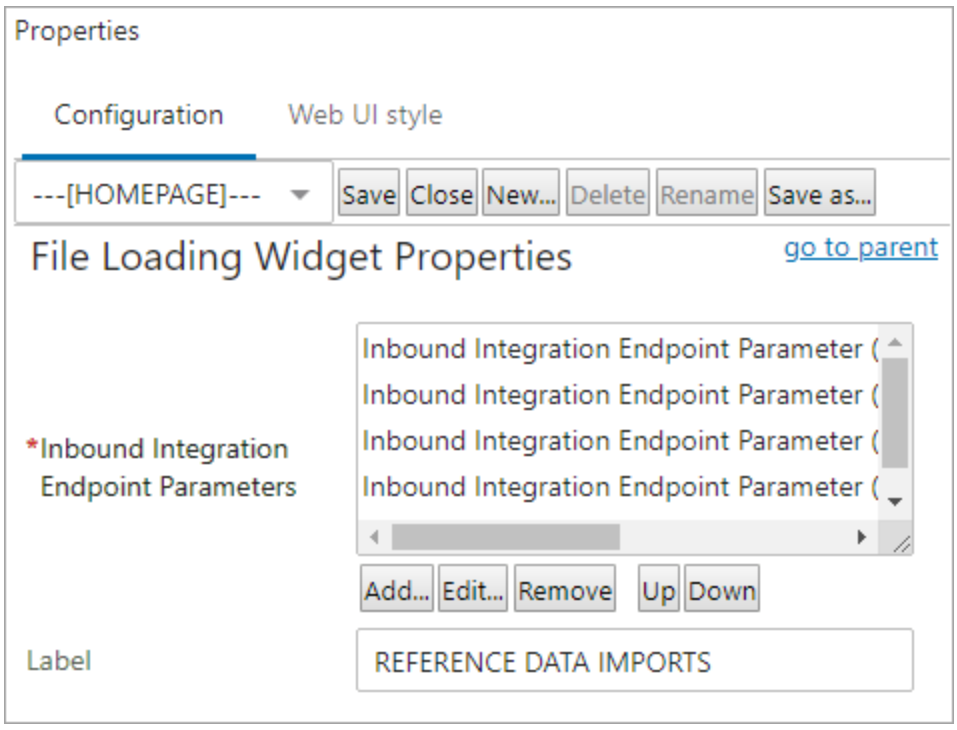
## Configuration

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

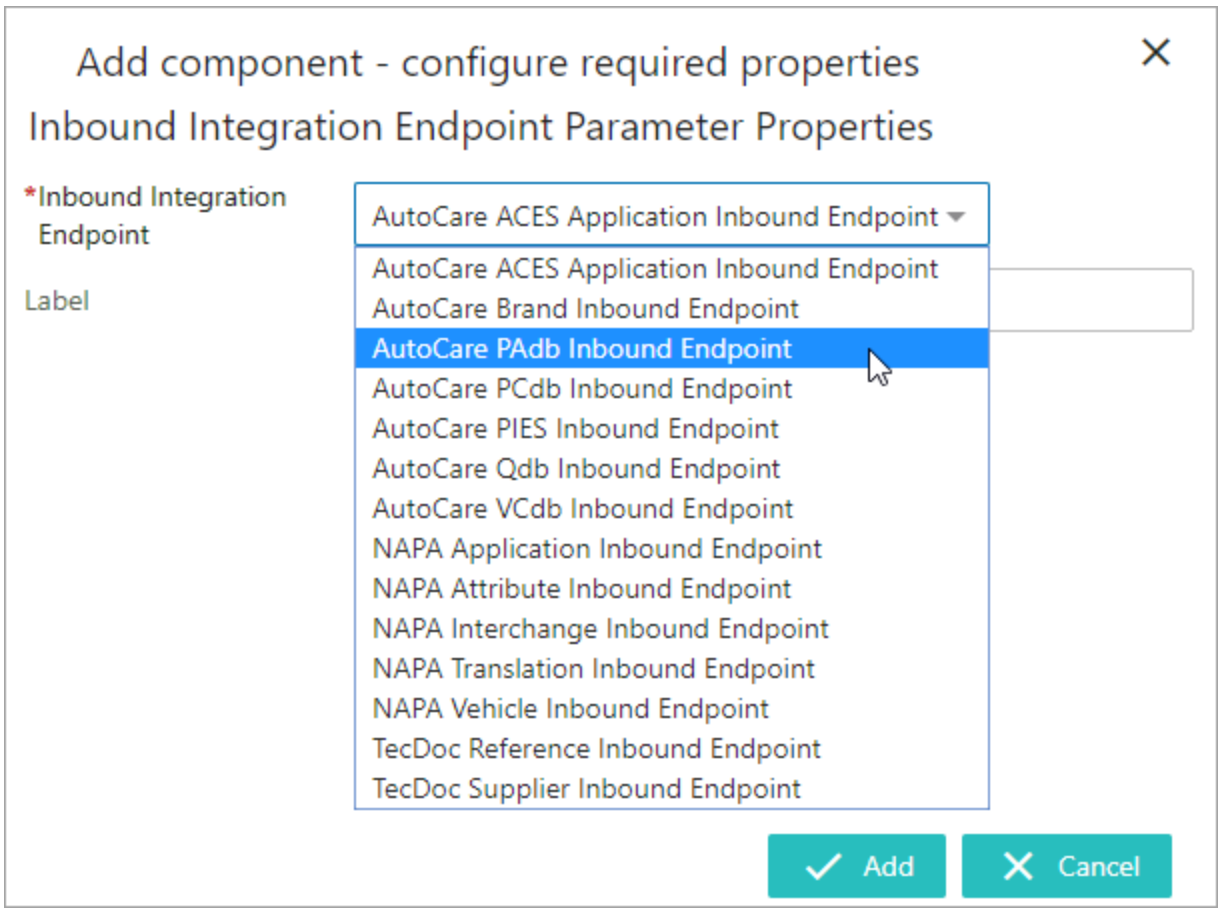
This topic describes how to configure a File Loading Widget so that users can drag and drop AutoCare PAdb files onto a File Loading Widget on a Web UI Homepage.

**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, refer to 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).

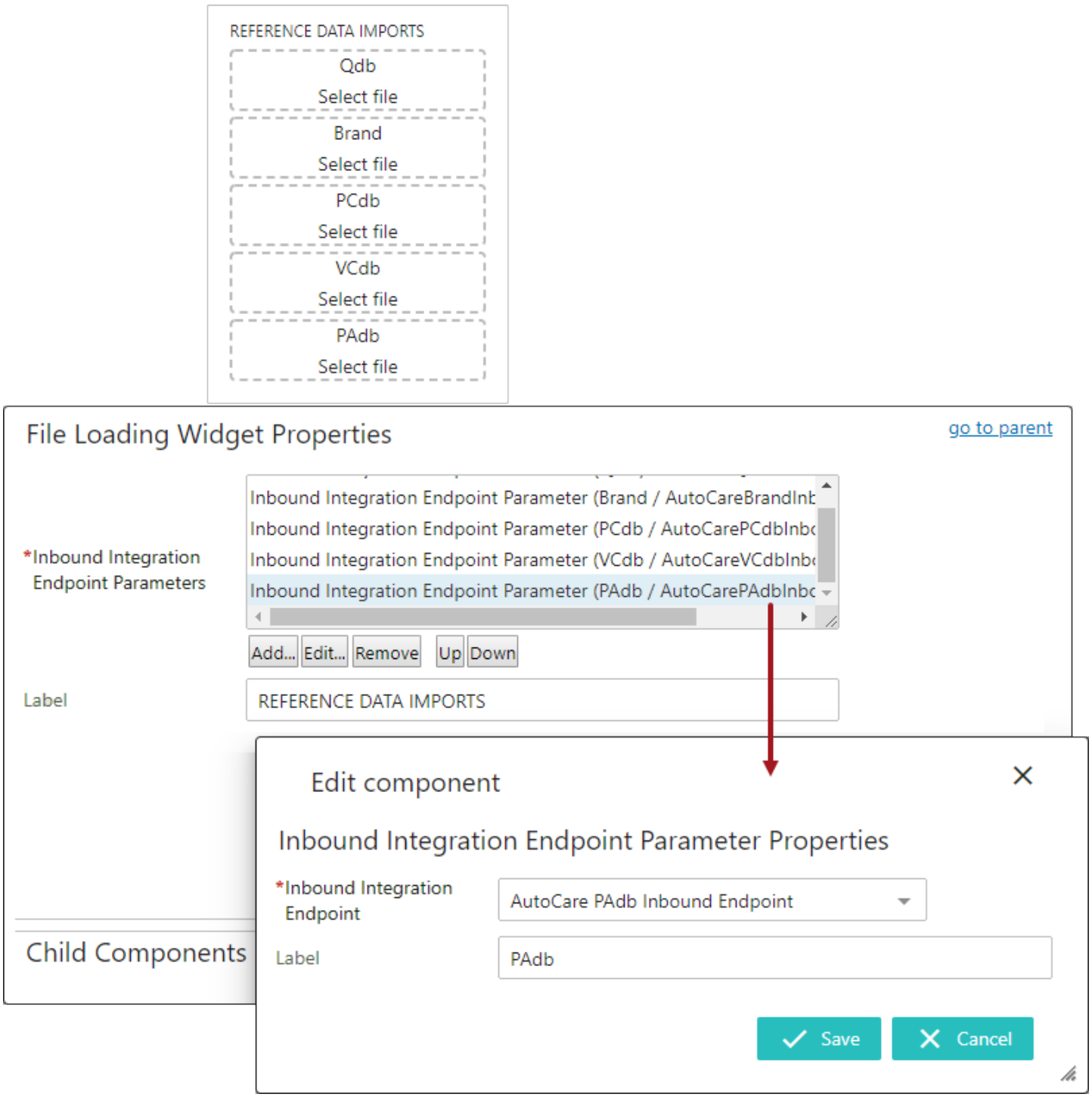



**Note:** If the desired IIEP does not display in the dropdown, then it can be created using the steps described in the **Configuring an IIEP for PAdb Imports** topic.

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.




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

## AutoCare VCdb Importer

The intention of the Automotive AutoCare VCdb Importer is to provide an out-of-the-box solution for importing data included within a supported VCdb file format. An ASCII encoded .zip file is the only file type that is supported to be used for upload.

This section includes information on:

- Using AutoCare VCdb Importer
- Configuring AutoCare VCdb Importer

## Prerequisites

It is expected that anyone using and/or configuring the AutoCare VCdb 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 Homepage Widget, including additional information about working with these widgets, refer to File Loading Widget topic, and Status Selector Homepage Widget topic, both within the **Web User Interfaces** section of the **STEP Online Help**.

### Using AutoCare VCdb Importer

The VCdb 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 VCdb Importer.

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

## Prerequisites

It is expected that anyone using and/or configuring the VCdb 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, refer to 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, refer to the **Workflows** section of **STEP Online Help**.

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

### Importing VCdb Files via Web UI

The intention of the AutoCare VCdb 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 VCdb file format. An ASCII encoded .zip file is the only file type that is supported to be used for upload.

A VCdb file defines the way trading partners exchange information for products that require vehicle, off-highway, and/or equipment fitment data. The VCdb file contains light, medium, and heavy duty vehicles, powersports, off-highway, and equipment configurations and attributes.

**Important:** It is always required that the latest VCdb file is imported into the system. Importing the VCdb file validates the version date against the version date stored on the 'AC_VCdbVersionDate' attribute which is valid on the VCdb 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, a Node Details screen must be created. For information about configuring the Node Details Screen, refer to the **Node Details Screen** topic within the **Web User Interfaces** section of **STEP Online Help**.

To easily start and monitor a VCdb import, the recommended practice is to use a Web UI Import Controller screen specific to the AutoCare VCdb 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 VCdb Importer** topic.

It is expected that anyone using and/or configuring the AutoCare VCdb 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, refer to the **AutoCare VCdb 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 Homepage 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.

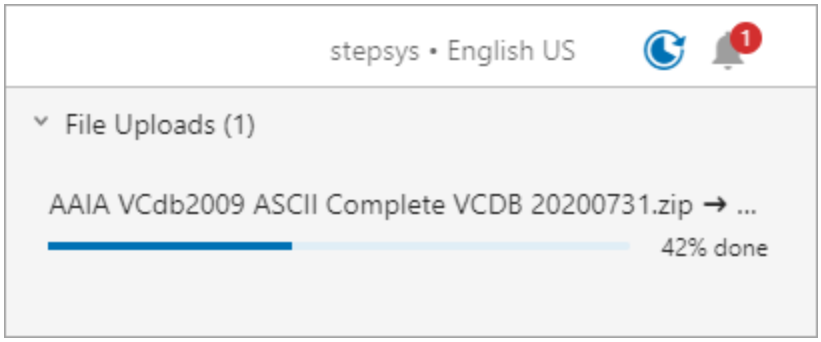
### Using the AutoCare VCdb Importer in Web UI

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

   For more information, refer to 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 'Recent background processes' side panel.

   In the example below, the 'Recent background processes' side panel is expanded and the file, 'AAIA VCdb2009 ASCII Complete VCDB 20200731.zip' is 42% processed.




   For more information on using the 'Recent background processes' side panel, refer to the **Recent Background Processes Side Panel** 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 VCdb Import). From there, the AutoCare VCdb 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.

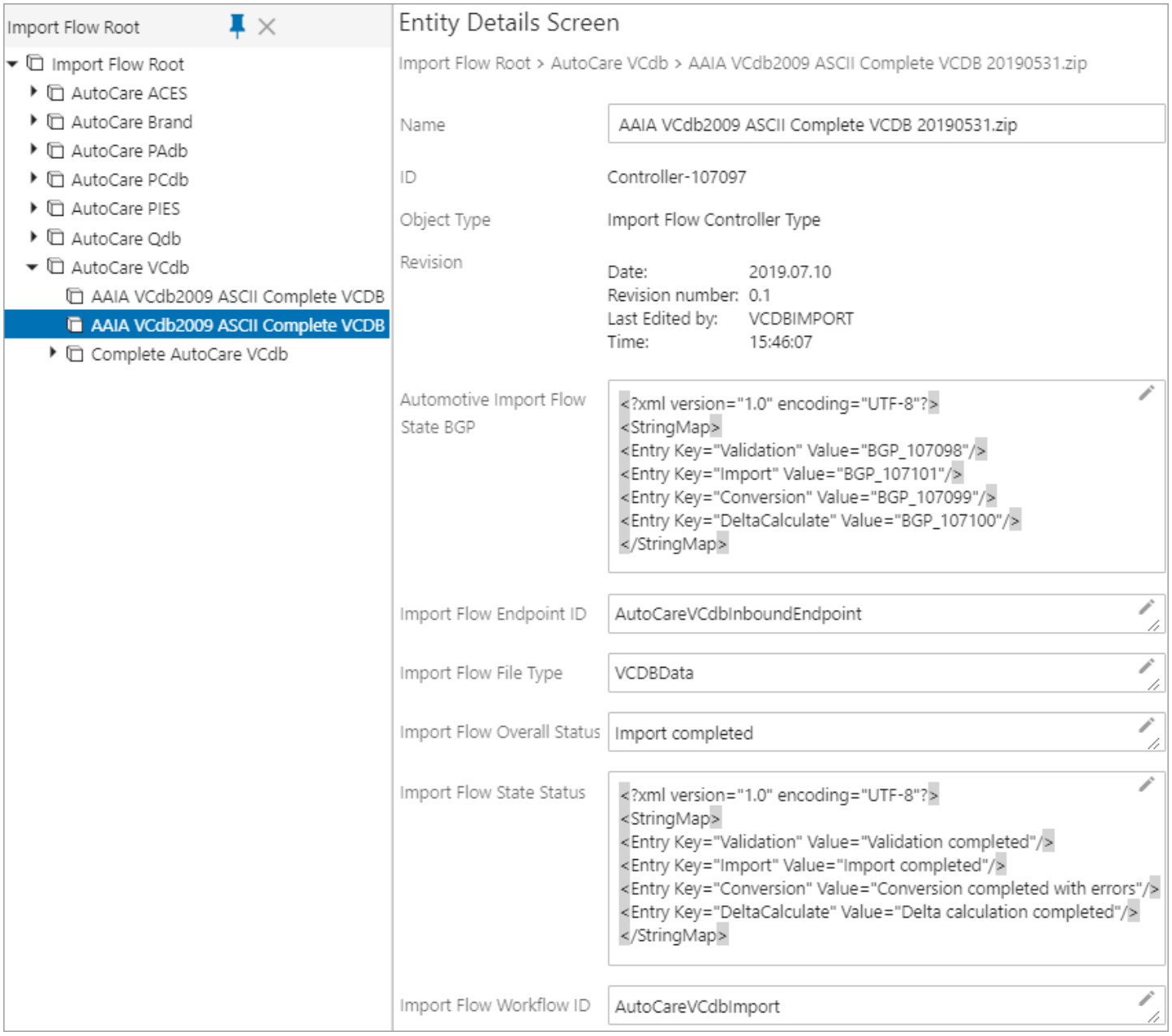
**Important:** It is critical to understand that it is only the Controller Entity that is in the workflow - the objects being acted on (created / updated / deleted) via information supplied in the import file are not in the workflow. Therefore, running standard business actions acting on current object 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, refer to the **Import State** topic within the **Default Workflow States and Functions** section.

4. If a Node 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 VCdb. 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 Node Details screen has not been configured. For information about configuring the Node Details Screen, refer to the **Node Details Screen** topic within the **Web User Interfaces** section of **STEP Online Help**.

In the example below, the Tree side panel is expanded, and the 'AAIA VCdb2009 ASCII Complete VCDB 20200731.zip' Controller with ID 'Controller- 107097' is displayed below the 'AutoCareVCdb' 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.

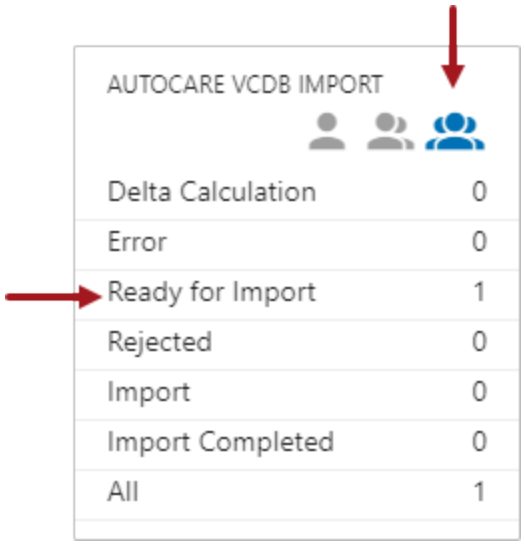


- **Name:** STEP Name of the Controller Entity. This is generated from the original name of the uploaded file.
- **Automotive Import Flow State BGP:** Attribute used to store the IDs of the background processes.
- **Import Flow Overall Status:** Attribute used to store and display the final status of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

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

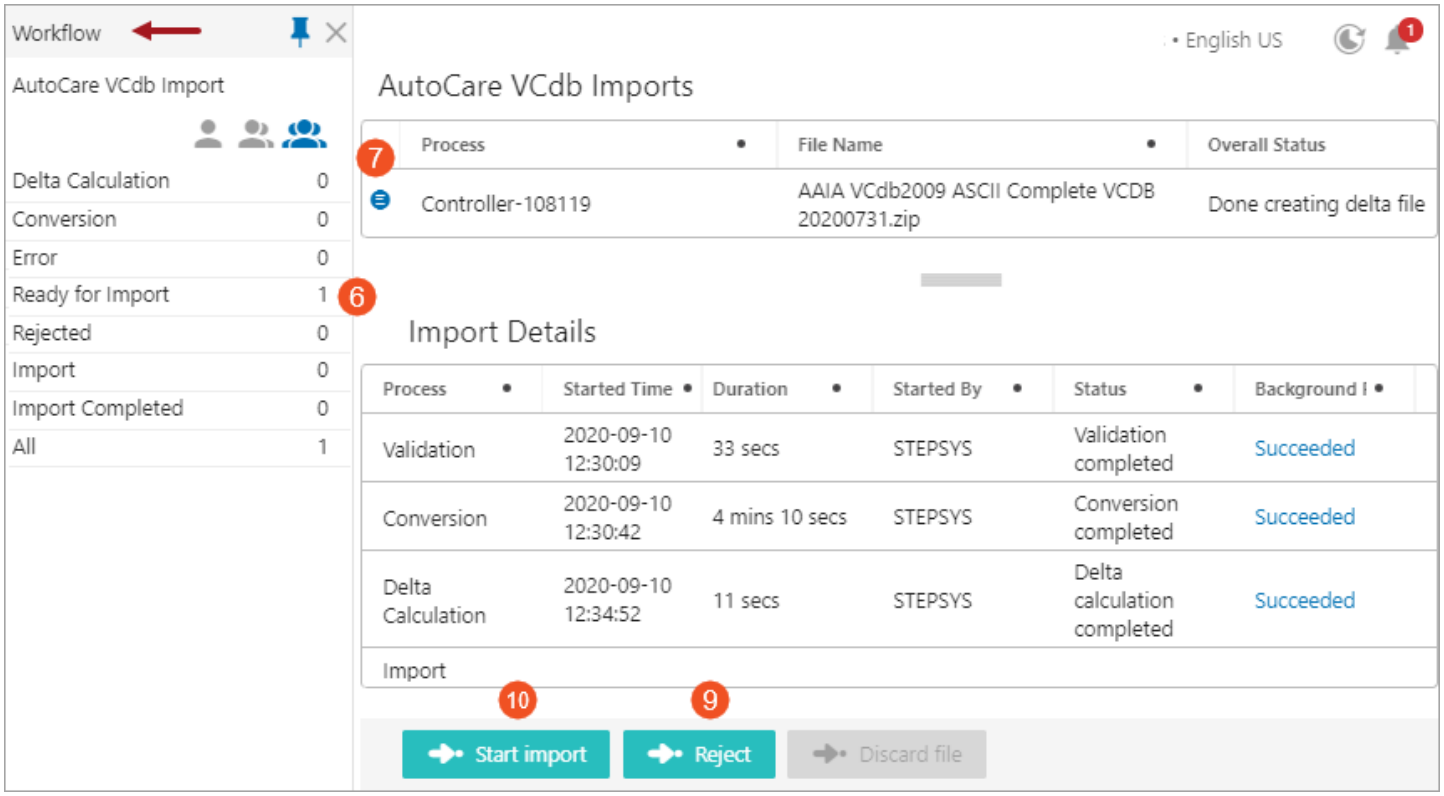
   In the example below, the Workflow panel is expanded, and the **Triple user button** is selected so that all items assigned to any user are displayed. Notice that one file is in the 'Ready for Import' state, and zero files in the 'Import Completed' state, for a total of one file.

   For more information, refer to 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 VCdb 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.

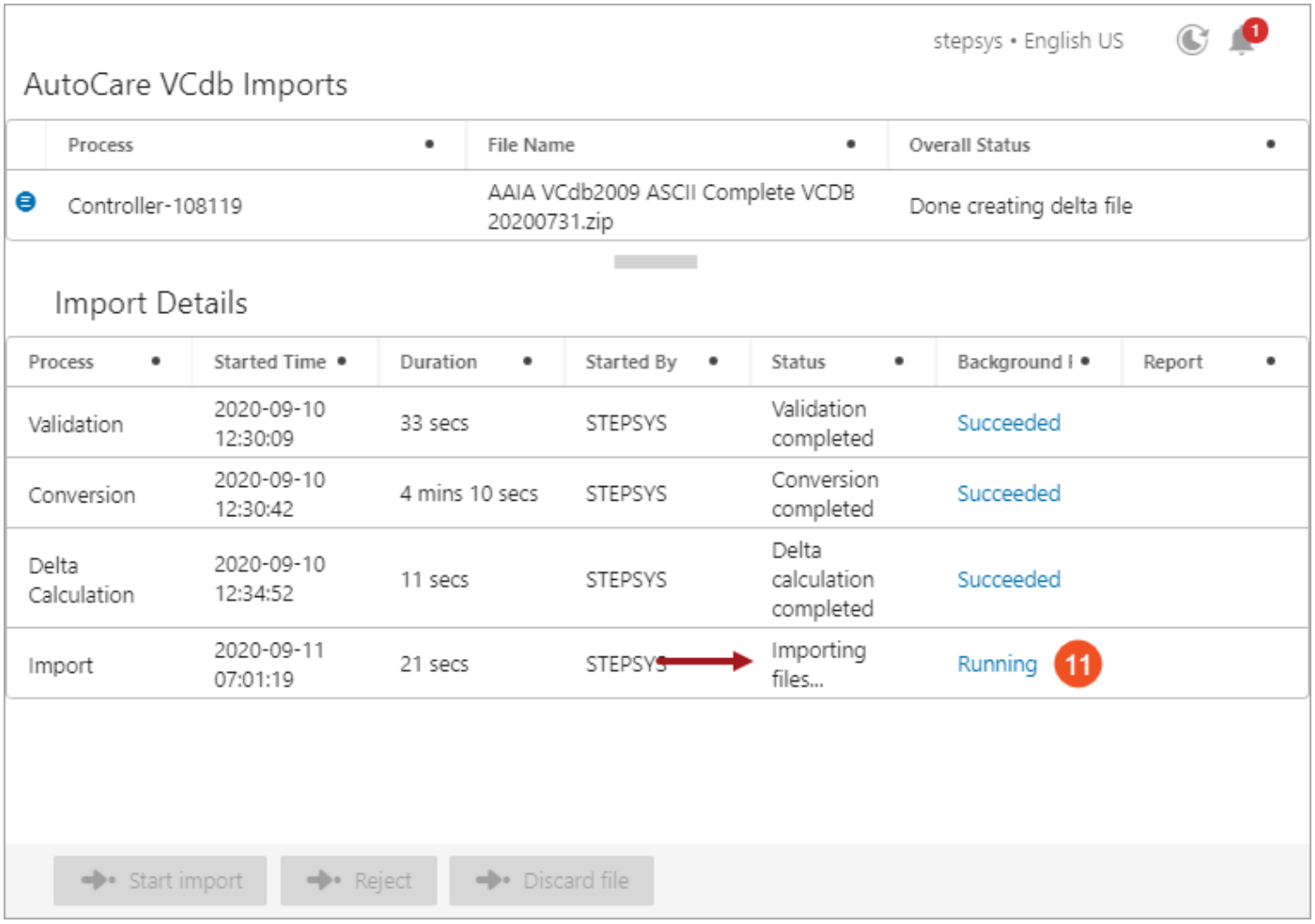
   In the example below, Controller 'AAIA VCdb2009 ASCII Complete VCDB 20200731.zip' (ID 'Controller-108119') with an overall status of 'Done creating delta file' is selected and the Import Details are displayed below the Controller list.



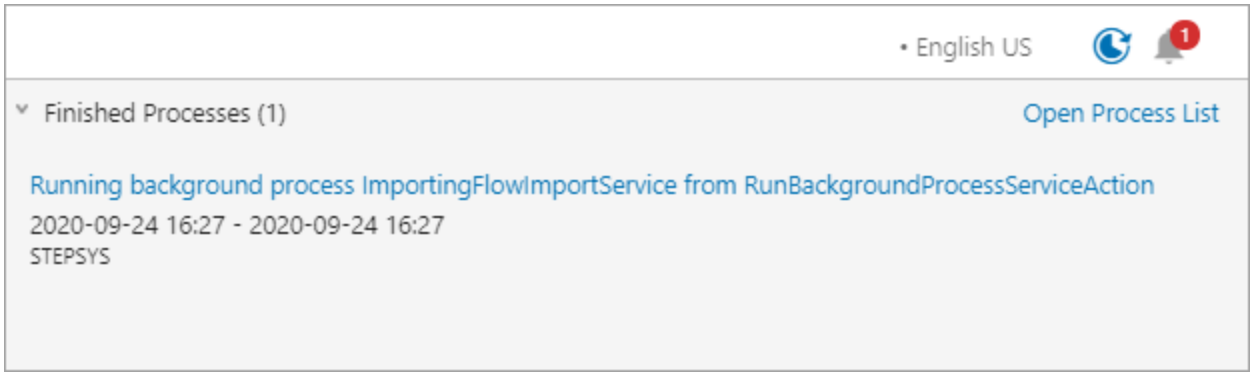

8. Optionally, before starting the import, clicking on the 'completedwitherrors' Background Process Link (only displayed when there is any error) will display the Background Process Details screen where users can view or download the details of the validation process.
9. Click the **Reject** button to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

While importing a VCdb file the import progress can be monitored using the 'AUTOCARE VCDB IMPORT' Status Selector Homepage Widget on the Web UI homepage and/or the Status Selector component in the left side panel. More detailed progress can be monitored in the import process Status column that is available within the Import Controller Screen.

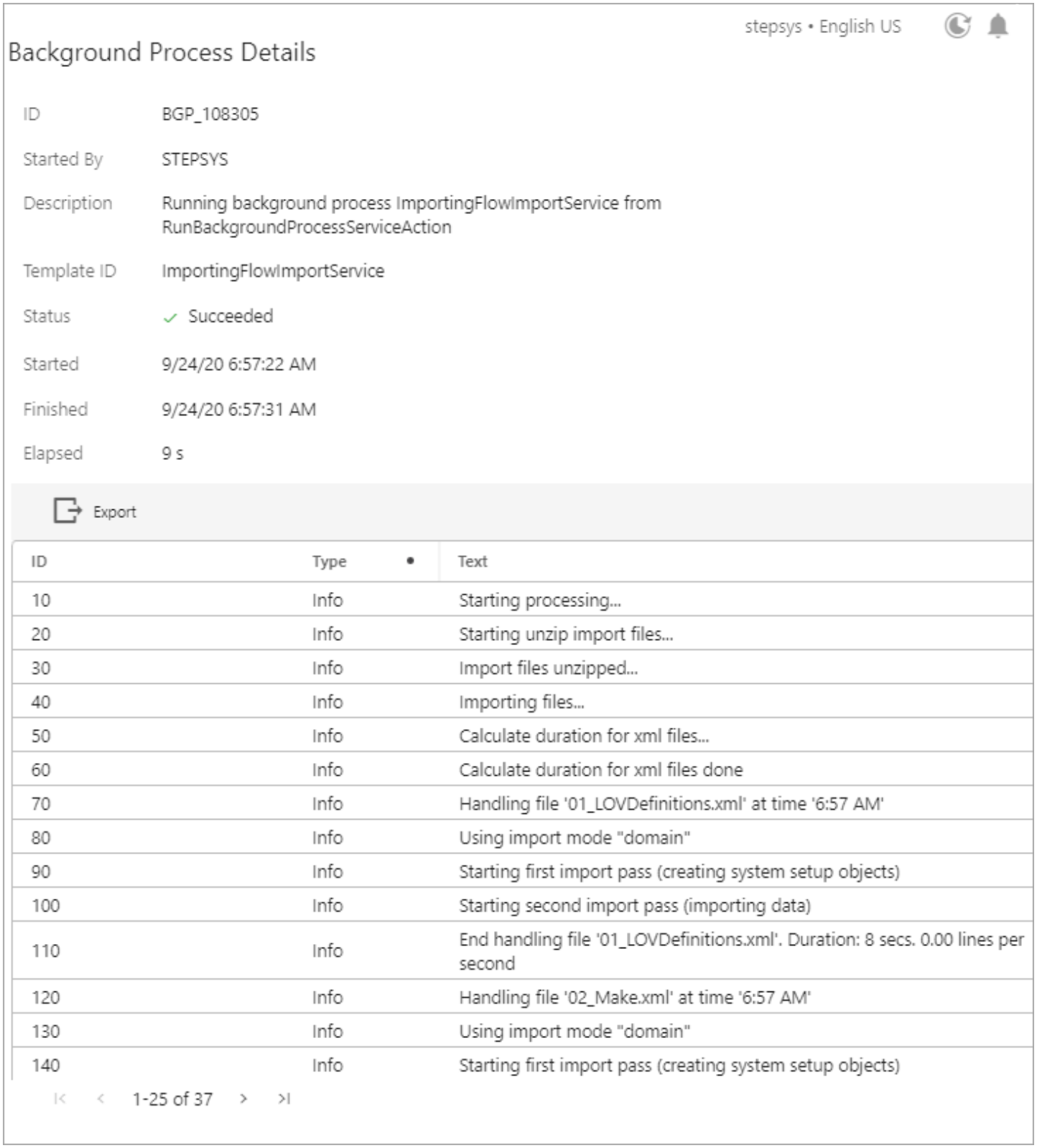
In the example below, the AutoCare VCdb Import Controller Screen displays the Overall Status and the Import process status as 'Importing files...' Additionally, the Background Process Link column displays a link to the Background Process Details screen as 'Running.'




11. Optionally, to view the details of the import process within the Background Process Details screen click the 'Running' Background Process Link (shown above) or the background process displayed in the 'Recent background processes' side panel.




In the example below, the background process with ID BGP_103805 has finished the import process.




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

For more information about VCdb import validation rules and example error messages, refer to the **VCdb Import Validation Rules** topic.

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

#### Sample AutoCare VCdb File Structure

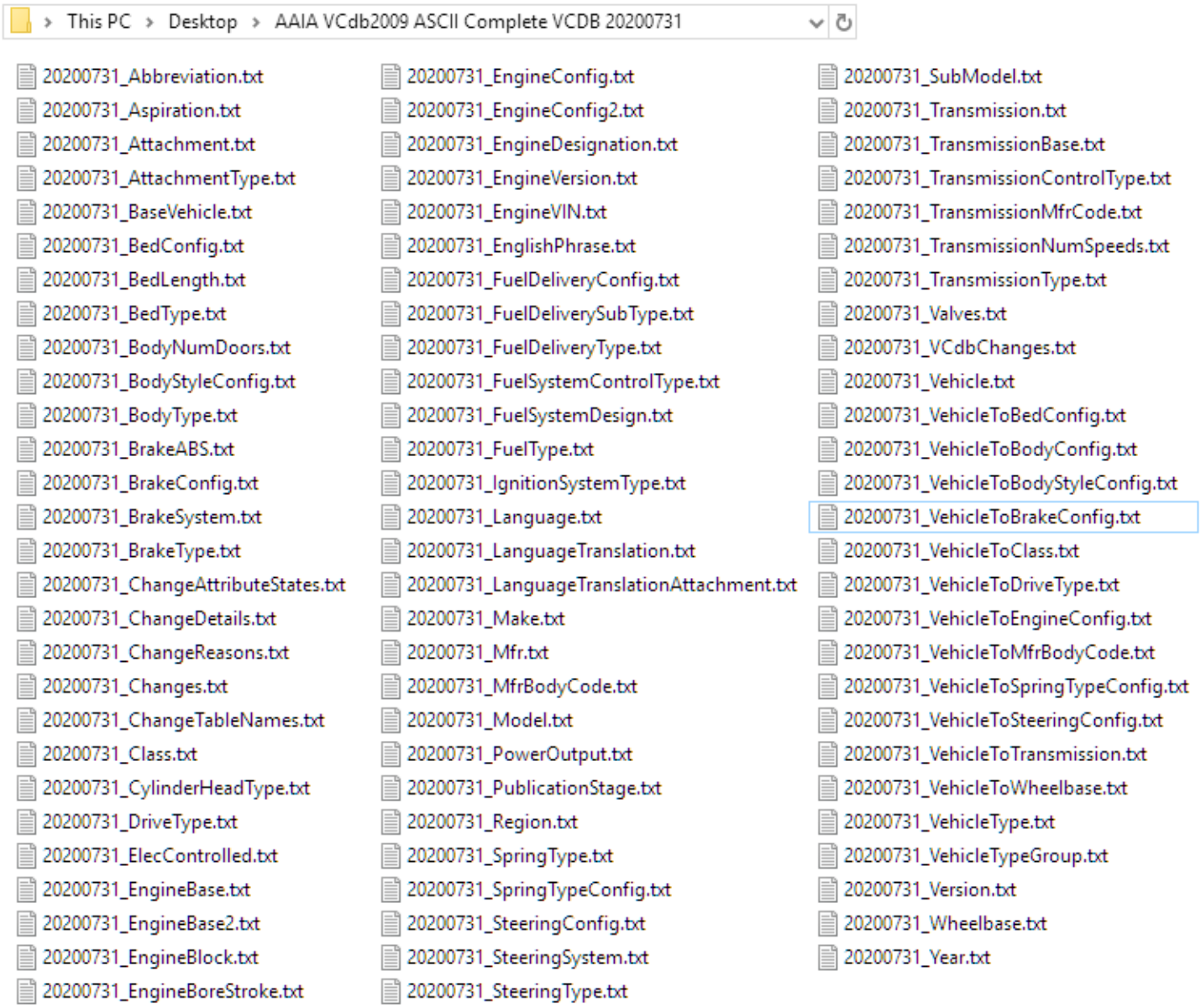
The intention of this topic is to introduce the basic AutoCare VCdb file structure and the way the objects are built in the system when a VCdb file is imported into the system.

## Delivering VCdb Files

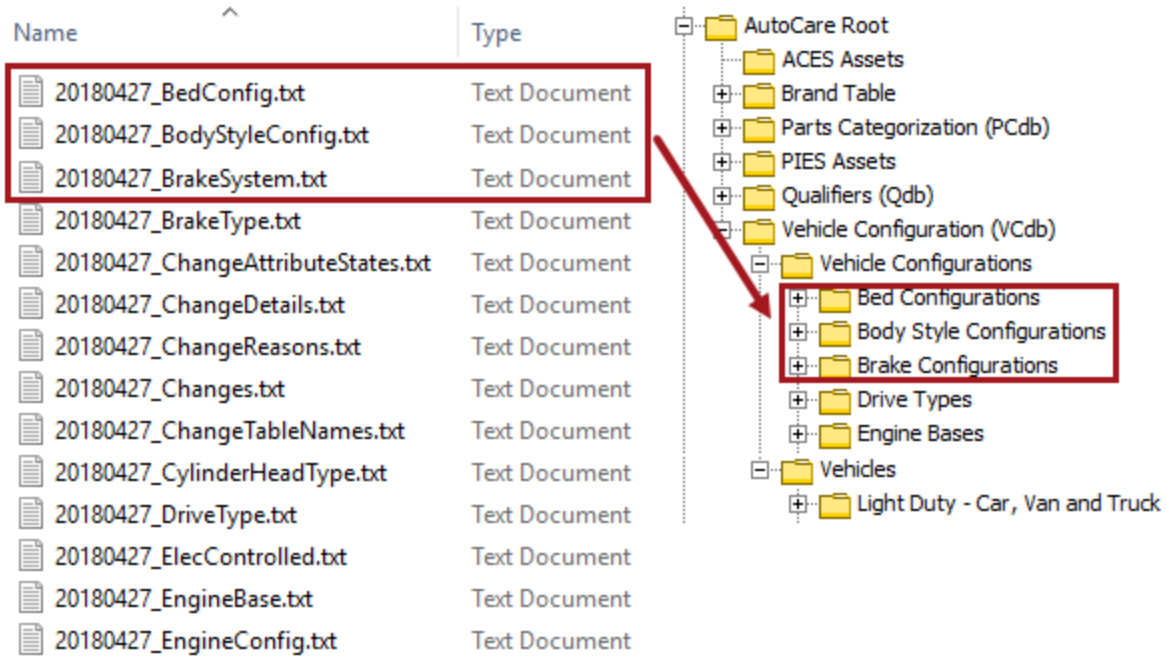
The VCdb is made up of 83 tables (83 pipe-delimited text files ) that contains light, medium, and heavy duty vehicles, powersports, and off-highway configurations and attributes. The Equipment VCdb contains 43 tables.

Auto Care will provide new VCdb files at the end of each month to be available for Auto Care subscribers to download. In the Auto Care VIP portal, when the selecting ASCII file type to download, a .zip file will be generated with the file naming such as 'AAIA VCdb2009 ASCII Complete VCDB 20200828.zip' or 'Auto Care VCDB ASCII Equipment 20200828.zip.'

The following 83 pipe-delimited text files are available within the VCdb Zip file and the way it is structured is demonstrated in the screenshot below.



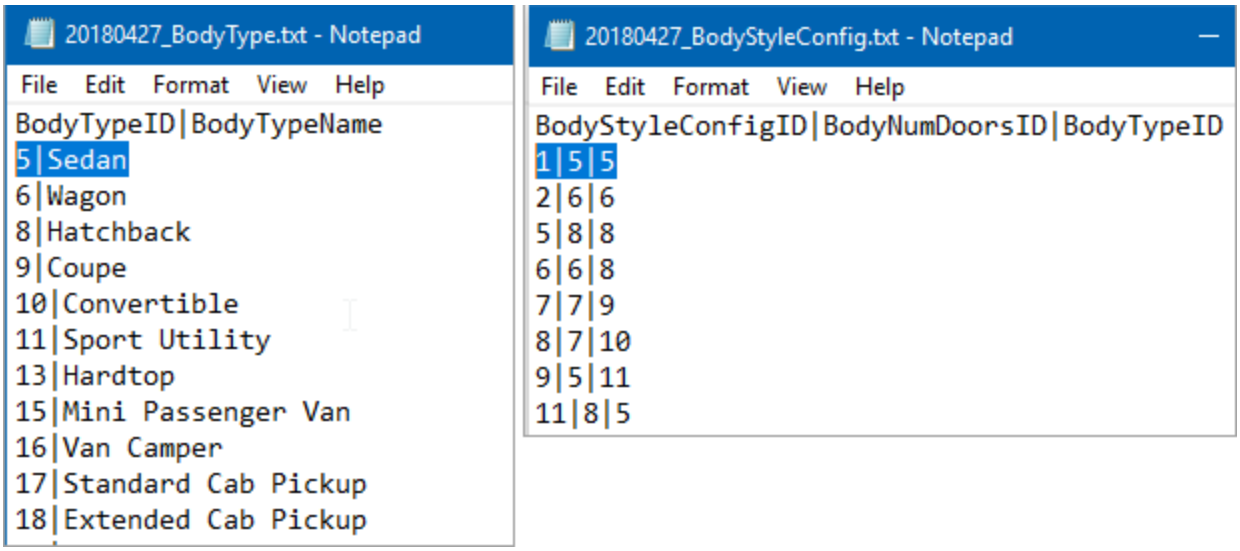

All of the information present in the VCdb file is modeled to be listed under the 'Vehicle Configuration (VCdb)' classification folder. The files 'BedConfig.txt,' 'BodyStyleConfig,' and 'BodyStyleConfiguration.txt,' and a screenshot of showing how it is populated in STEP is displayed below.



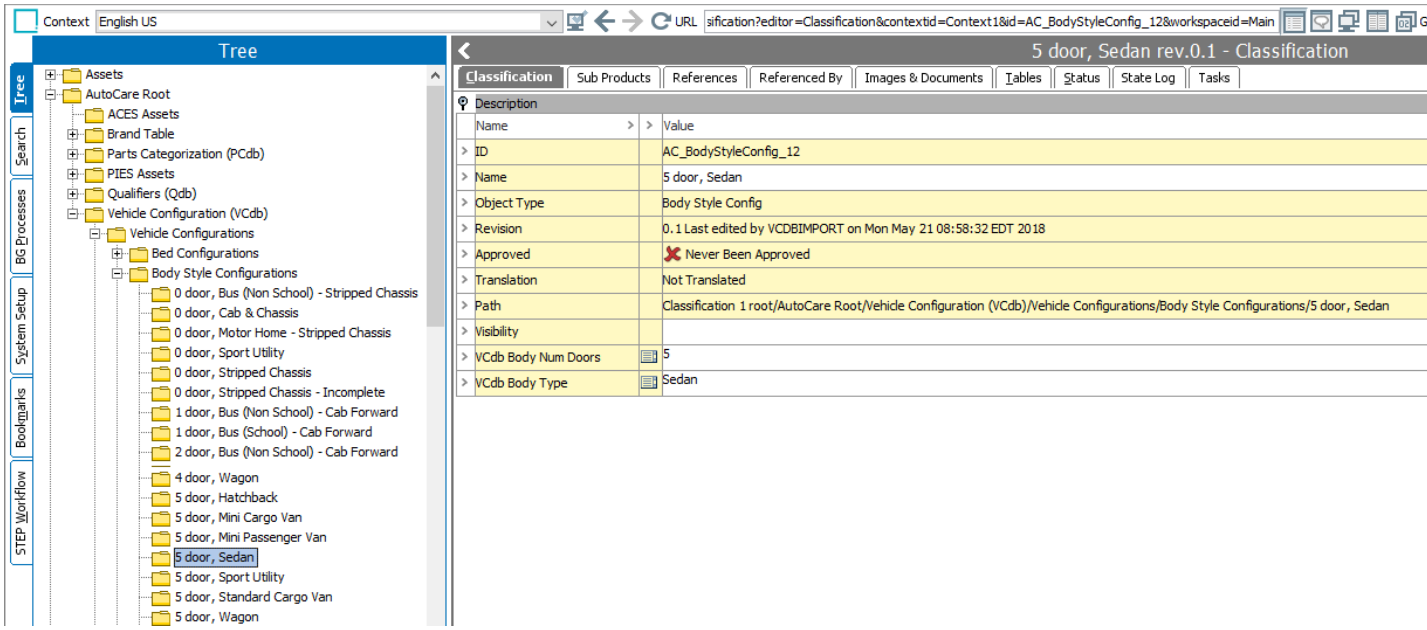

Below are some examples of the information within the txt files and a screenshot of showing how it is populated in STEP is displayed below.

- Sample files 'BodyType.txt' and 'BodyStyleConfiguration.txt,' and a screenshot of showing how the information within it is populated in STEP is displayed below.

  Input:

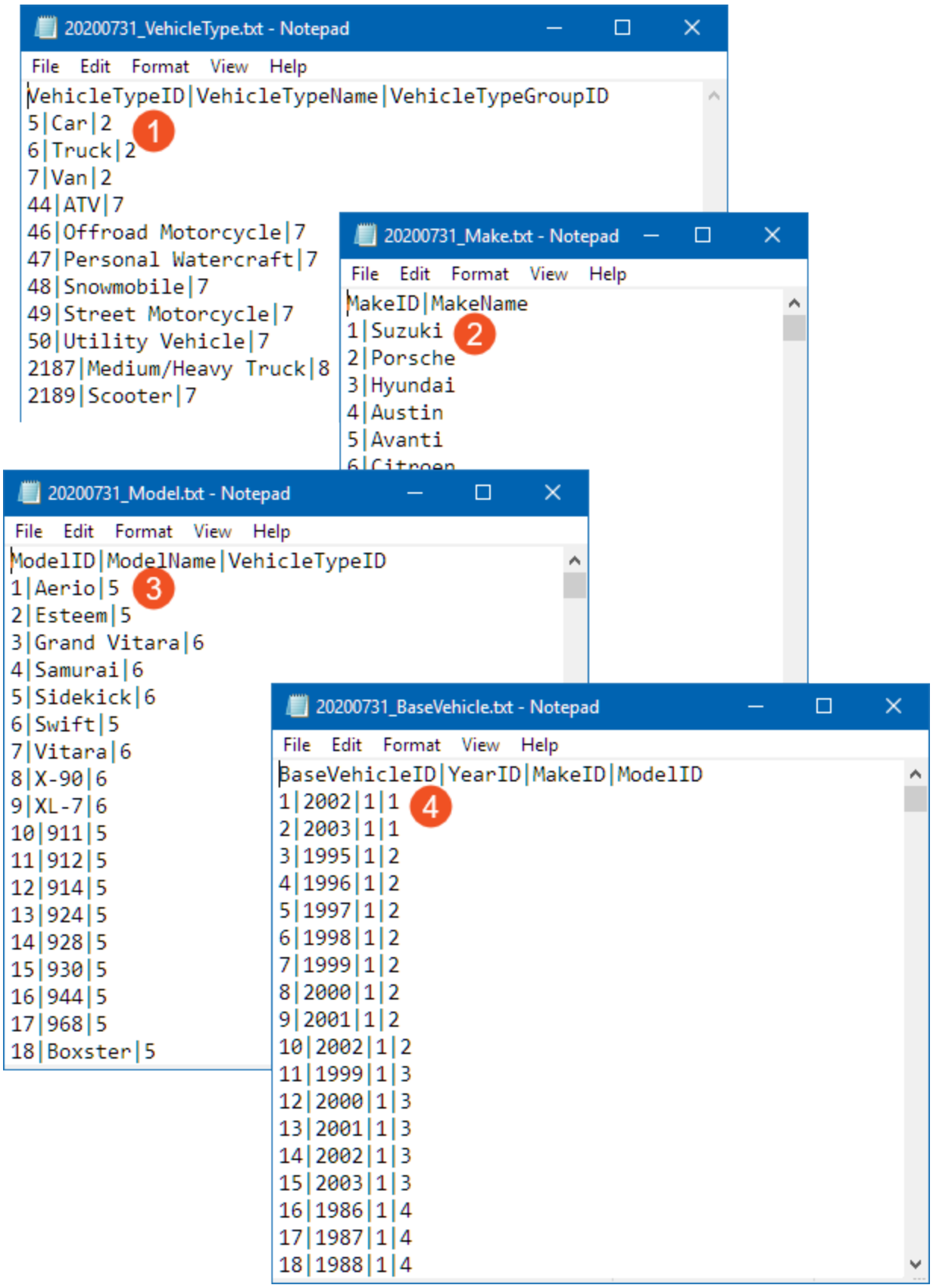



  **Output:** Based on the files 'BodyType.txt' and 'BodyStyleConfiguration.txt,' the classification shown in the screenshot below will be created.

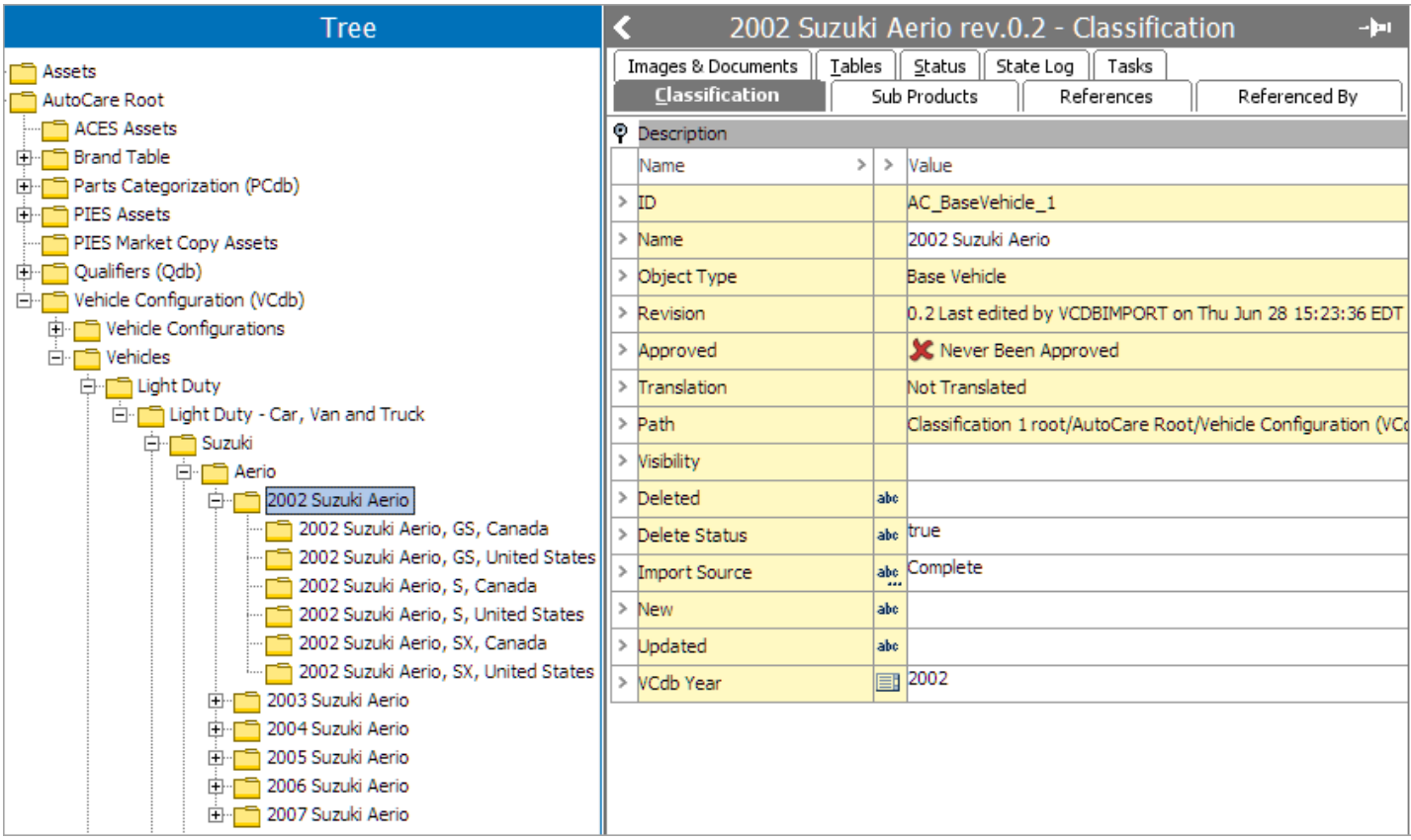



- The sample text files below shows how the 'Make.txt,' 'Model.txt,' 'BaseVehicle.txt,' and 'VehicleType.txt' files are present in the VCdb file folder, and the screenshot below shows how the Vehicle Type (1), Make (2), Model (3) , and the Base Vehicle (4) objects are populated in the system.

  **Input files**

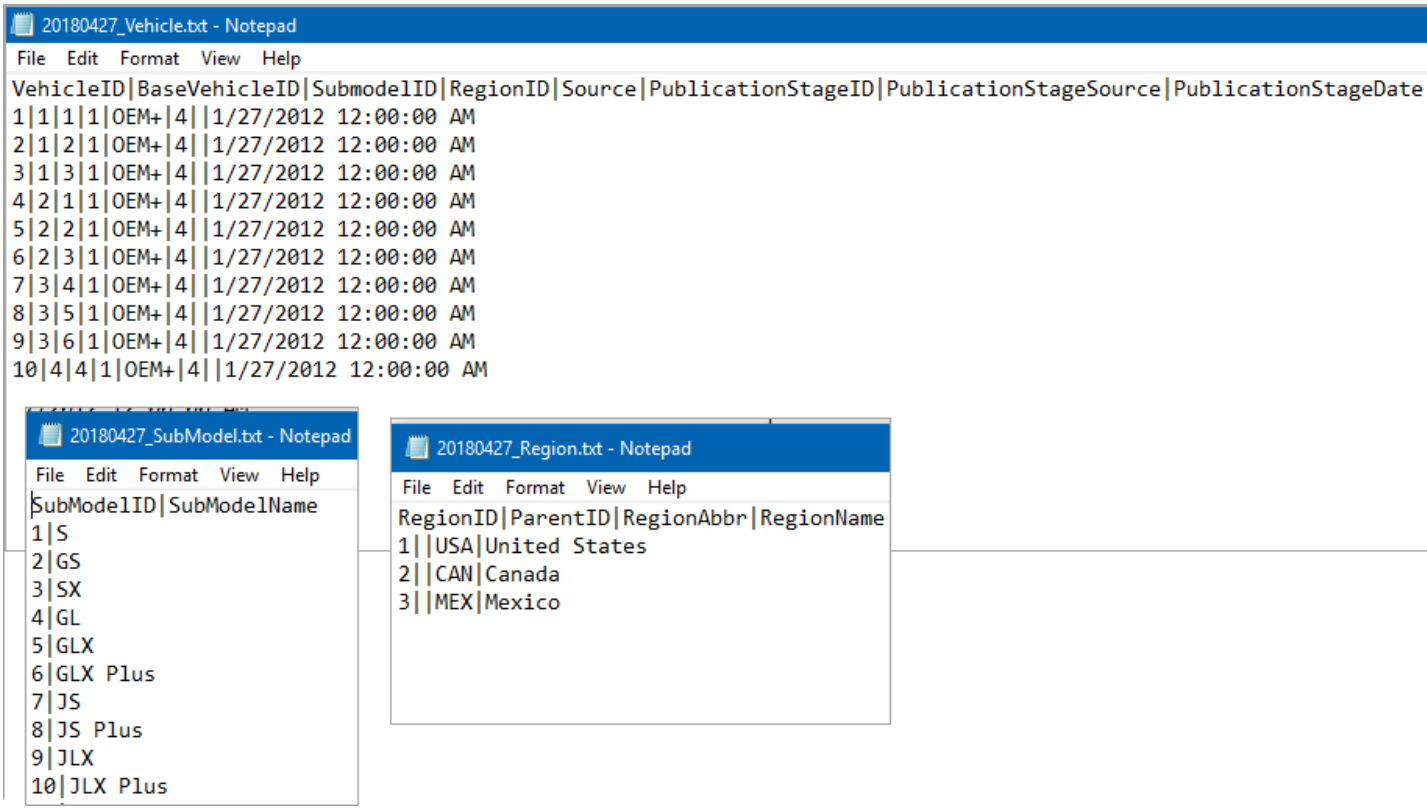

20200731_VehicleType.txt - Notepad
File Edit Format View Help
VehicleTypeID|VehicleTypeName|VehicleTypeGroupID
5|Car|2
6|Truck|2
7|Van|2
44|ATV|7
46|Offroad Motorcycle|7
47|Personal Watercraft|7
48|Snowmobile|7
49|Street Motorcycle|7
50|Utility Vehicle|7
2187|Medium/Heavy Truck|8
2189|Scooter|7
1
20200731_Make.txt - Notepad
File Edit Format View Help
MakeID|MakeName
1|Suzuki
2|Porsche
3|Hyundai
4|Austin
5|Avanti
2
20200731_Model.txt - Notepad
File Edit Format View Help
ModelID|ModelName|VehicleTypeID
1|Aerio|5
2|Esteem|5
3|Grand Vitara|6
4|Samurai|6
5|Sidekick|6
6|Swift|5
7|Vitara|6
8|X-90|6
9|XL-7|6
10|911|5
11|912|5
12|914|5
13|924|5
14|928|5
15|930|5
16|944|5
17|968|5
18|Boxster|5
3
20200731_BaseVehicle.txt - Notepad
File Edit Format View Help
BaseVehicleID|YearID|MakeID|ModelID
1|2002|1|1
2|2003|1|1
3|1995|1|2
4|1996|1|2
5|1997|1|2
6|1998|1|2
7|1999|1|2
8|2000|1|2
9|2001|1|2
10|2002|1|2
11|1999|1|3
12|2000|1|3
13|2001|1|3
14|2002|1|3
15|2003|1|3
16|1986|1|4
17|1987|1|4
18|1988|1|4
4

**Output**

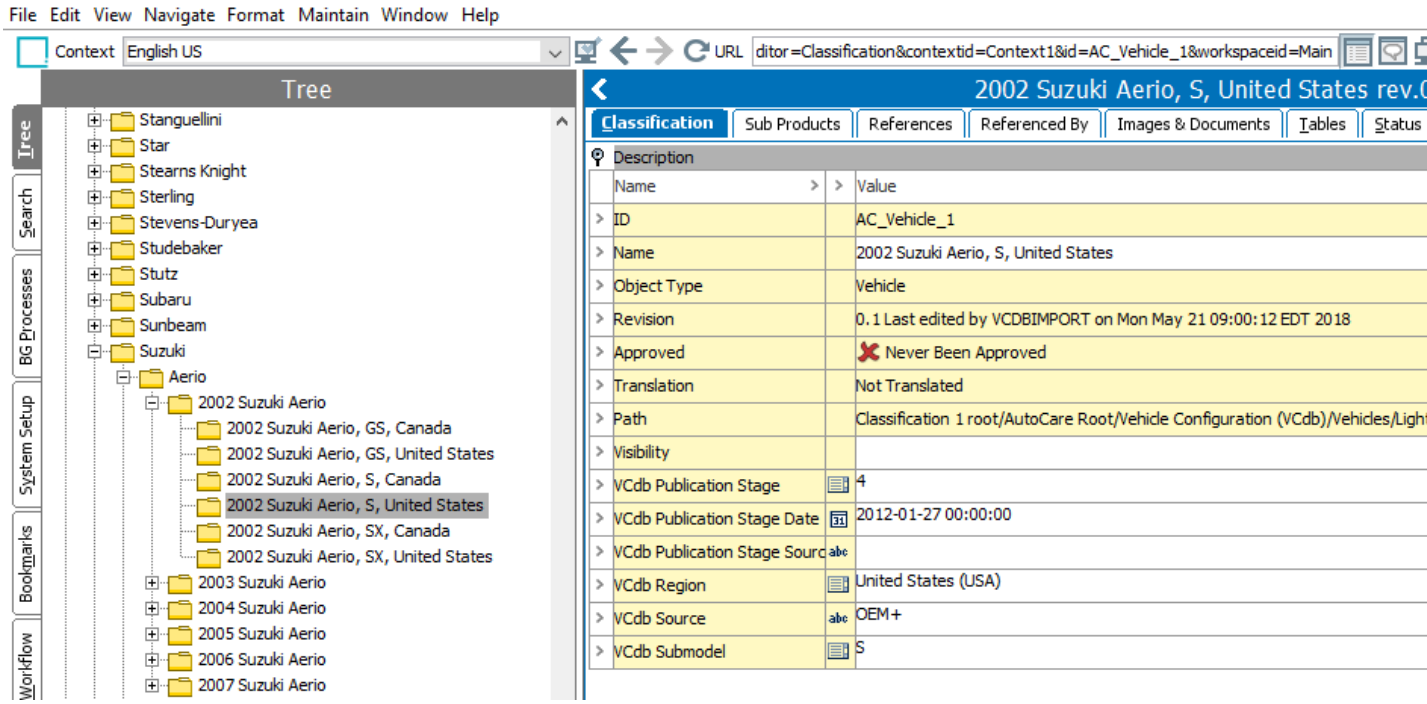



- Sample files 'Vehicle.txt,' 'SubModel.txt,' and 'Region.txt,' and a screenshot of showing how the information present in it is populated in STEP is displayed below.

  **Input file**

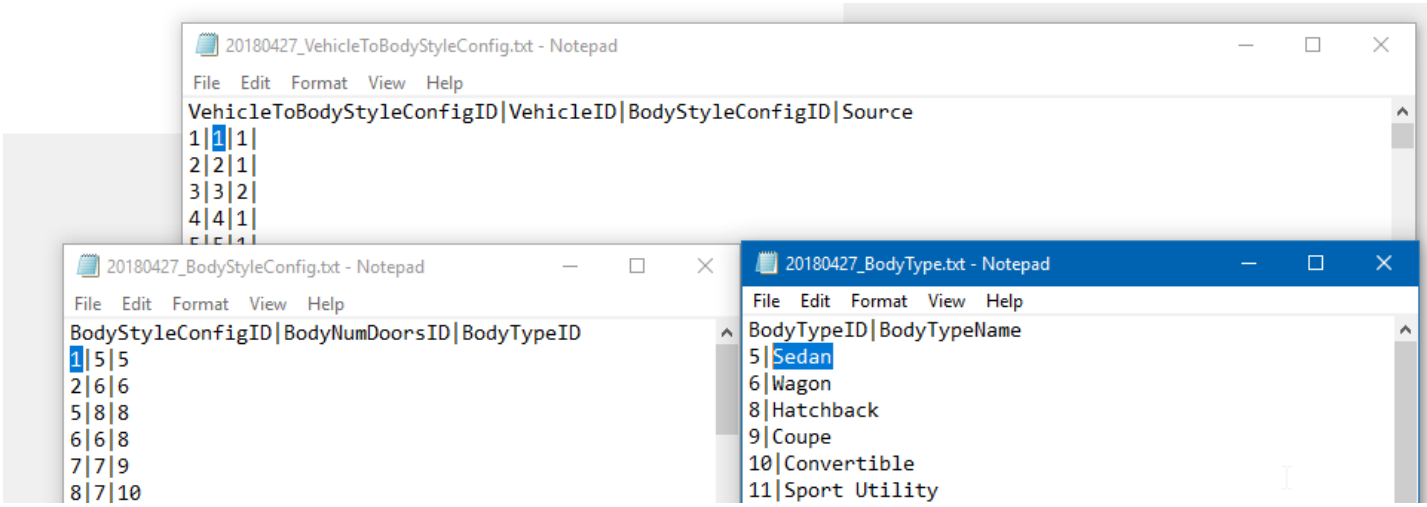

20180427_Vehicle.txt - Notepad
File Edit Format View Help
VehicleID|BaseVehicleID|SubmodelID|RegionID|Source|PublicationStageID|PublicationStageSource|PublicationStageDate
1|1|1|1|OEM+|4||1/27/2012 12:00:00 AM
2|1|2|1|OEM+|4||1/27/2012 12:00:00 AM
3|1|3|1|OEM+|4||1/27/2012 12:00:00 AM
4|2|1|1|OEM+|4||1/27/2012 12:00:00 AM
5|2|2|1|OEM+|4||1/27/2012 12:00:00 AM
6|2|3|1|OEM+|4||1/27/2012 12:00:00 AM
7|3|4|1|OEM+|4||1/27/2012 12:00:00 AM
8|3|5|1|OEM+|4||1/27/2012 12:00:00 AM
9|3|6|1|OEM+|4||1/27/2012 12:00:00 AM
10|4|4|1|OEM+|4||1/27/2012 12:00:00 AM
20180427_SubModel.txt - Notepad
File Edit Format View Help
SubModelID|SubModelName
1|S
2|GS
3|SX
4|GL
5|GLX
6|GLX Plus
7|JS
8|JS Plus
9|JLX
10|JLX Plus
20180427_Region.txt - Notepad
File Edit Format View Help
RegionID|ParentID|RegionAbbr|RegionName
1||USA|United States
2||CAN|Canada
3||MEX|Mexico

## Output

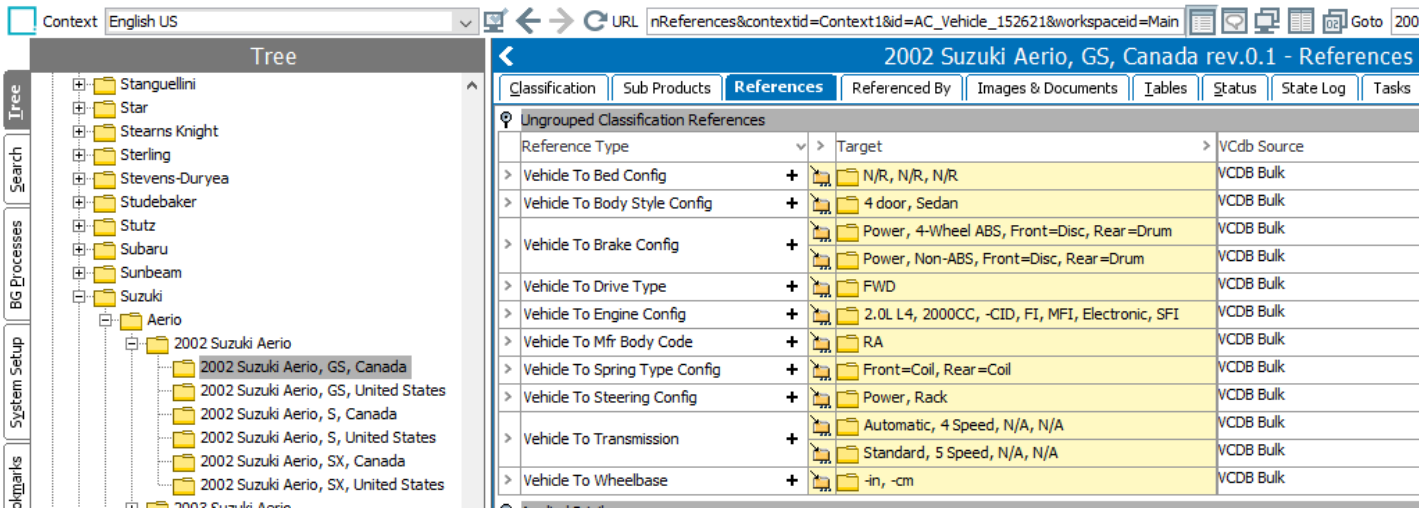

File Edit View Navigate Format Maintain Window Help
Context English US
URL ditor=Classification&contextid=Context1&id=AC_Vehicle_1&workspaceid=Main
Tree
Tree
Search
BG Processes
System Setup
Bookmarks
Workflow
Stanguellini
Star
Stearns Knight
Sterling
Stevens-Duryea
Studebaker
Stutz
Subaru
Sunbeam
Suzuki
Aerio
2002 Suzuki Aerio
2002 Suzuki Aerio, GS, Canada
2002 Suzuki Aerio, GS, United States
2002 Suzuki Aerio, S, Canada
2002 Suzuki Aerio, S, United States
2002 Suzuki Aerio, SX, Canada
2002 Suzuki Aerio, SX, United States
2003 Suzuki Aerio
2004 Suzuki Aerio
2005 Suzuki Aerio
2006 Suzuki Aerio
2007 Suzuki Aerio
2002 Suzuki Aerio, S, United States rev.0
Classification
Sub Products
References
Referenced By
Images & Documents
Tables
Status
Description
Name
Value
ID
AC_Vehicle_1
Name
2002 Suzuki Aerio, S, United States
Object Type
Vehicle
Revision
0.1 Last edited by VCDBIMPORT on Mon May 21 09:00:12 EDT 2018
Approved
Never Been Approved
Translation
Not Translated
Path
Classification 1 root/AutoCare Root/Vehicle Configuration (VCdb)/Vehicles/Ligh
Visibility
VCdb Publication Stage
4
VCdb Publication Stage Date
2012-01-27 00:00:00
VCdb Publication Stage Sourc
VCdb Region
United States (USA)
VCdb Source
OEM+
VCdb Submodel
S

- Sample files 'VehicleToBodyStyleConfig.txt,' 'BodyStyleConfig.txt,'and 'BodyType.txt,' and a screenshot of showing how the information present in it is populated in STEP is displayed below.

  **Input files:**




  **Output:** Based on the files 'VehicleToBodyStyleConfig.txt,' 'BodyStyleConfig.txt,'and 'BodyType.txt,' the vehicle will be linked to the body type via the 'VehicleToBodyStyleConfig' reference type.




### 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.

**Important:** For best results, AutoCare imports should be executed in the order specified within the **AutoCare Importers** 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 that provided '2014-11-28.' |
| Empty line | Validation state | Warning is displayed. | Table 'VehicleToBedConfig' contains an empty row at line 3. |

#### Using VCdb Change Handling Service

Whenever Auto Care releases a new version of the VCdb file, vehicles and/or vehicle configurations could be added, updated, or deleted. It is required for the system to monitor these updates and take action whenever necessary.

In STEP, a Base Vehicle is composed of multiple vehicle configuration objects. If one of these objects changes, it means that the Base Vehicle(s) referring to this configuration object also changes. Currently, STEP can set change flags on these objects while importing the VCdb file, but it will only set the change flag on the configuration objects, not on the Base Vehicle that uses the configuration. That means that it could be hard to get an overview of the Base Vehicles that contain changes somewhere in the configuration. That has been improved by the **VCdbChangeHandlingService** as it can move updates in the configuration objects to the Base Vehicles that use the changed configuration object.

The '**VCdbChangeHandlingService**' background service performs the following functionalities:

- Identifies the Base Vehicles that have changes in their Vehicle Configuration

  The system finds all VCdb Vehicle Configurations that are flagged as New, Updated, or Deleted, and then flags the corresponding VCdb Base Vehicle(s) accordingly.

- Handles the Vehicles that are marked for deletion and then handles the application records referring to these deleted Vehicles

  In this activity, the background process can purge all VCdb Base Vehicles and vehicle configurations that are flagged for deletion (Delete = true), and then it can purge their application records. If it is not required to be directly deleted, then the Vehicles that are flagged for deletion (Delete = true) can be initiated into a different workflow to be handled separately.

- Merges the custom vehicles with the new VCdb Vehicles

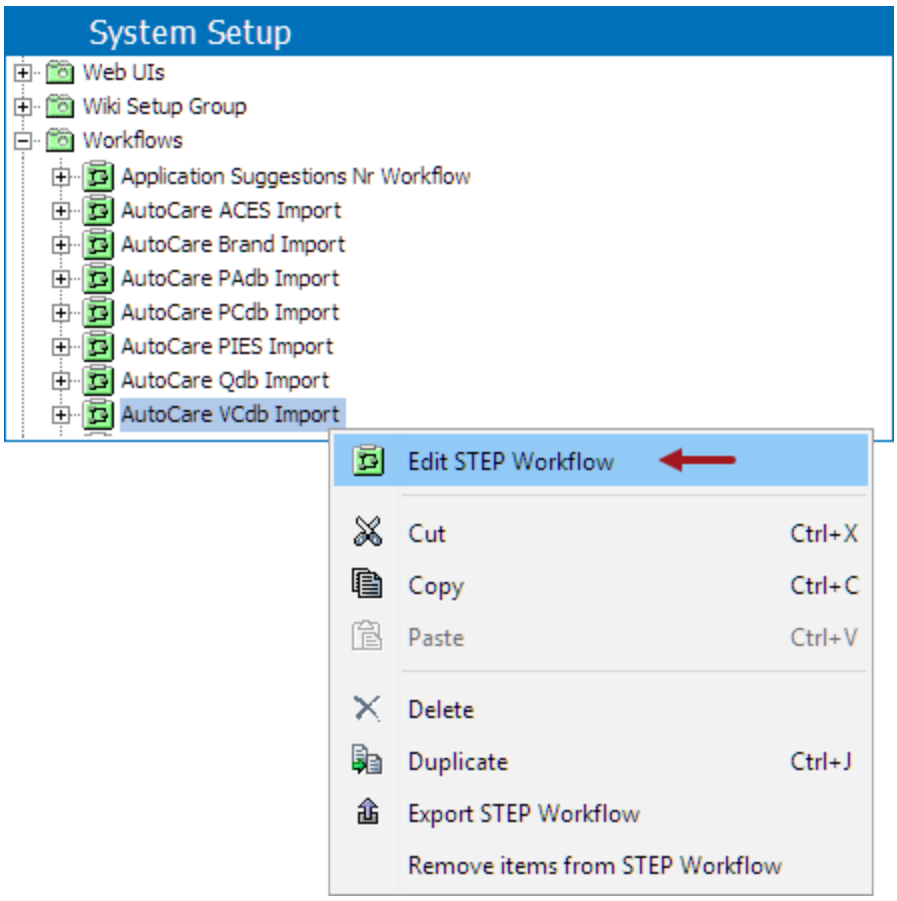
  Whenever a new Vehicle / Vehicle Configuration is introduced in the market, the information is not always immediately added to be available in the VCdb file. Often, users end up creating a custom database for such Vehicles / Vehicle Configurations as an interim solution and would also create application records referencing them in the system. To handle such scenario, the 'VCdbChangeHandlingService' background service will look for the VCdb Base Vehicles that are flagged as either New or Updated and then check if a custom vehicle exists in the system that matches the new VCdb Base Vehicle based on the Vehicle name, Model name, and Make name. For each custom vehicle that matches a new or updated VCdb Base Vehicle, the background process will determine if the application records that are created for the custom vehicles can be moved to the VCdb Base Vehicles.

## Configuring AutoCare VCdb Import Workflow to Run VCdb Change Handling

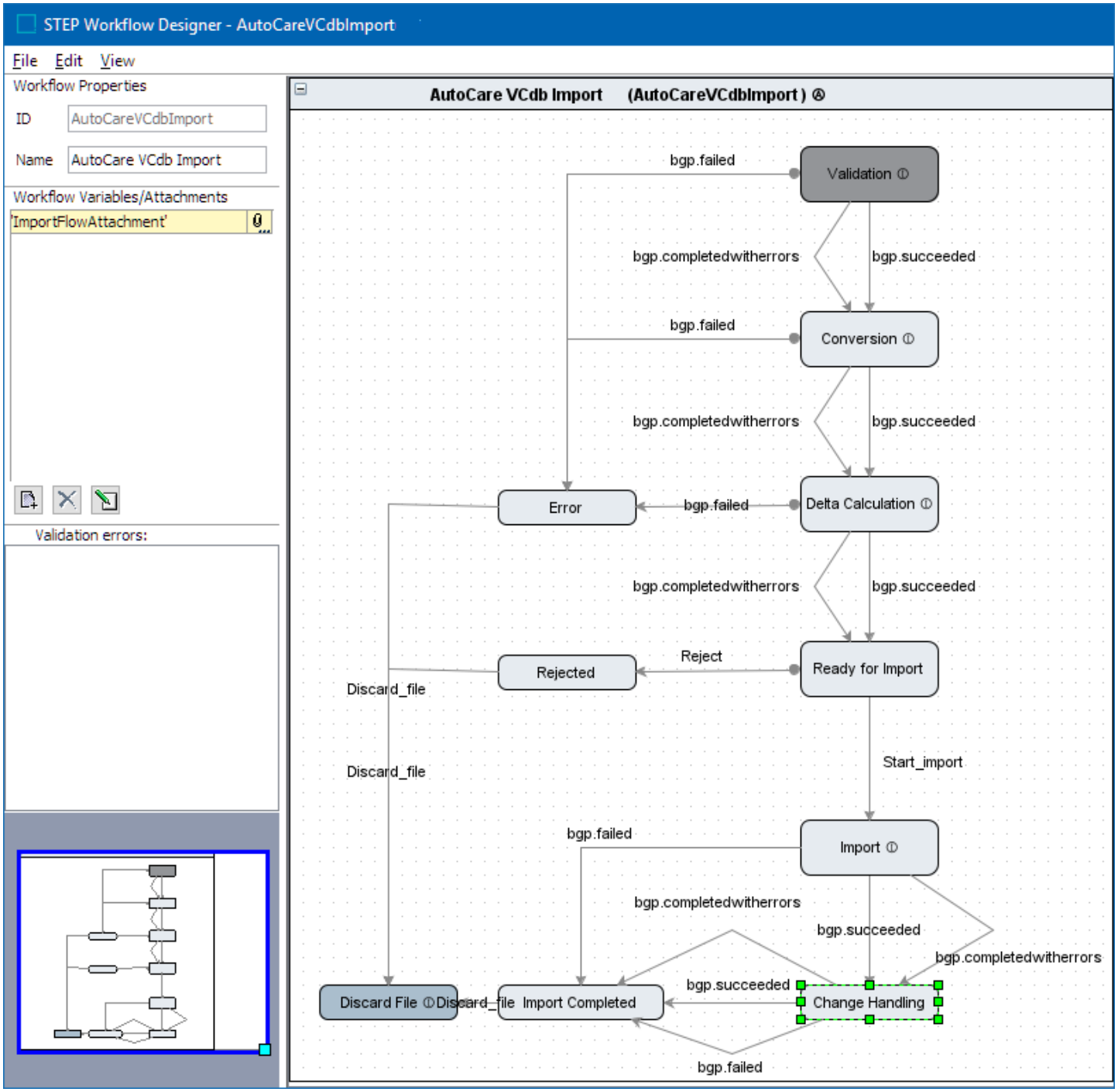
- Change flag attributes and a business action must be configured and added to the Import state for the **AutoCare VCdb Import** workflow in order to set the change flag status. For more information on how to configure the 'Set import status attributes' operation, refer to the **Business Action: Set Import Status Attributes** topic in the **Automotive Reference Guide**.
- If vehicles that are marked for deletion should be moved into a separate workflow, then the workflow that handles the deleted vehicle objects must be created first. For more information on creating / editing workflows, refer to the **Workflows** section of the **STEP Online Help**.
- If custom vehicles are used, then the custom vehicle data model must be configured.

The 'VCdbChangeHandlingService' *cannot* be executed on any existing state in the AutoCare VCdb Import workflow. To achieve this, a new state after the Import state needs to be created, and a business action that triggers a background process to execute the 'VCdbChangeHandlingService' background service is added to the newly created Change Handling state in the import workflow. The business action gets automatically executed whenever there is a new import. Below are the steps to create and update the AutoCare VCdb Importer workflow to execute the 'VCdbChangeHandlingService':

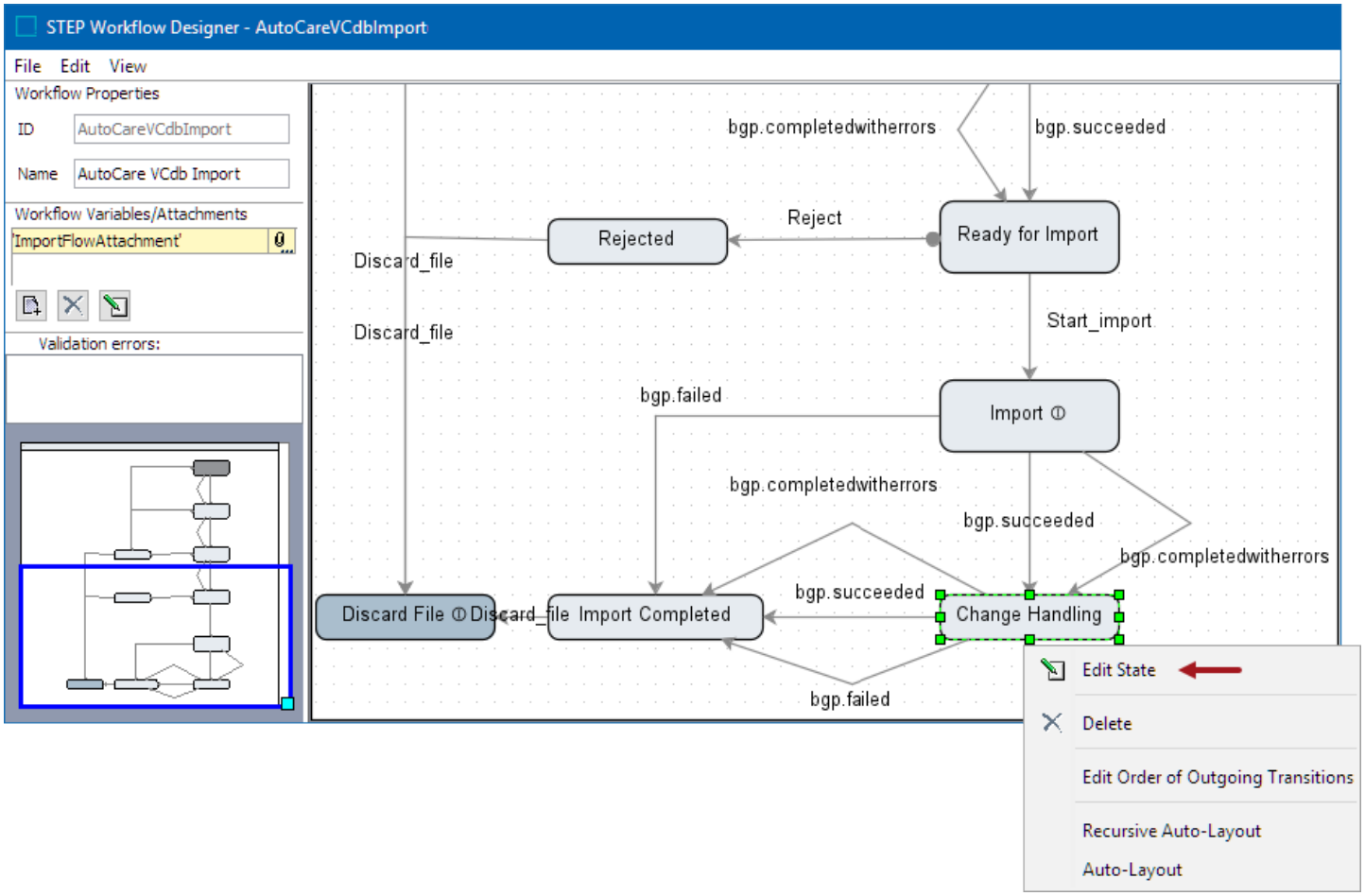
1. Go to System Setup > Workflows > Select the AutoCare VCdb Importer workflow.
2. Right-click the workflow, select **Edit STEP Workflow**, and the STEP Workflow Designer will display.



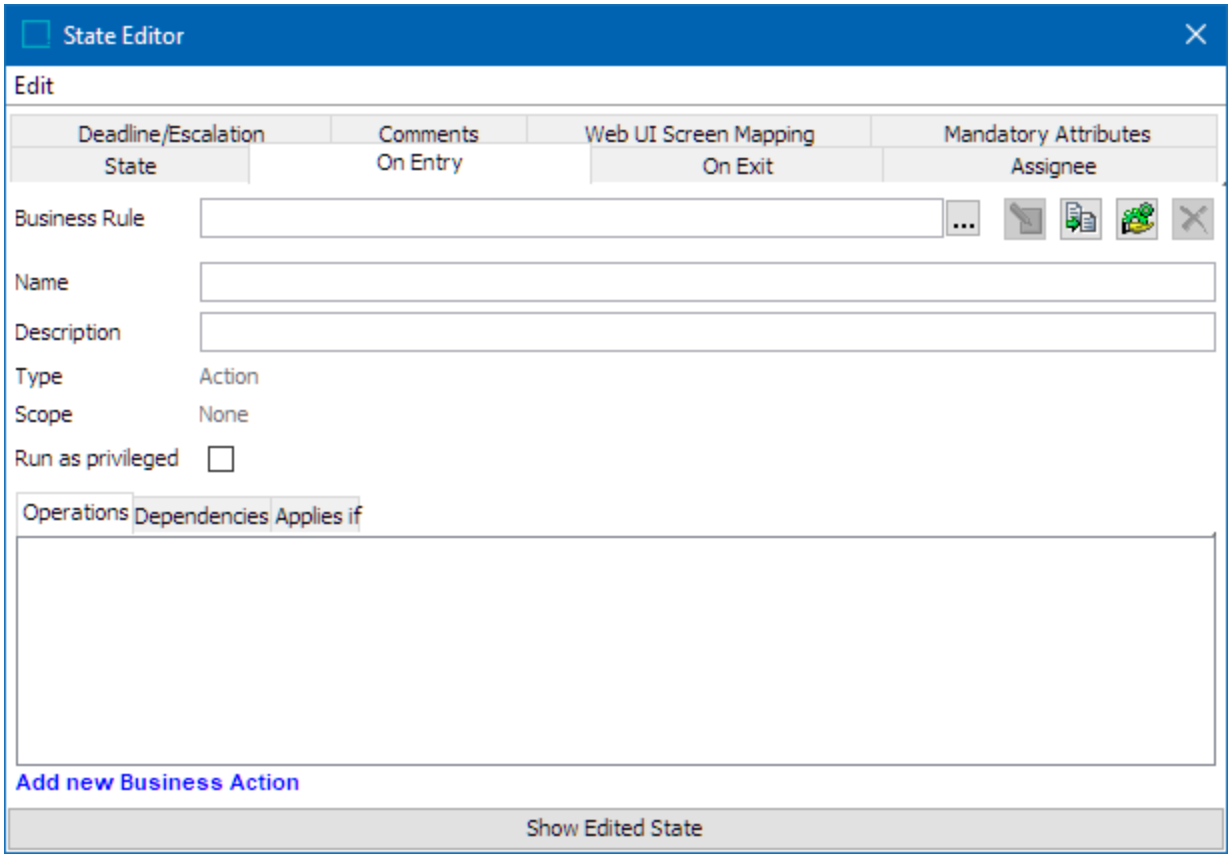

3. Within the STEP Workflow Designer, create a new state between the Import state and Import Completed state. In this example, a new state called 'Change Handling' is created (as shown below). For detailed instructions on creating States and Transitions, refer to **Creating a Workflow** topic of the **Workflows** section of the **STEP Online Help**.



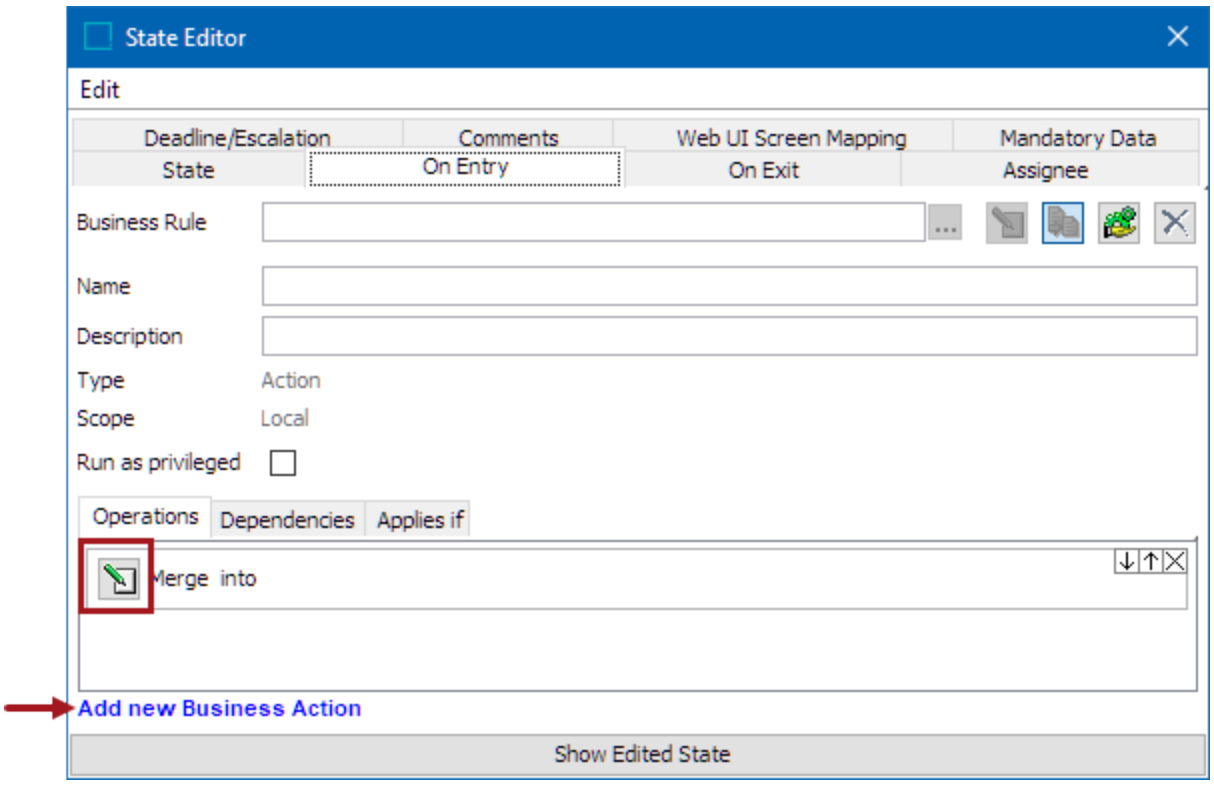

4. Right-click the newly created state (in this case it is 'Change Handling' state), select **Edit State**, and the State Editor will display.



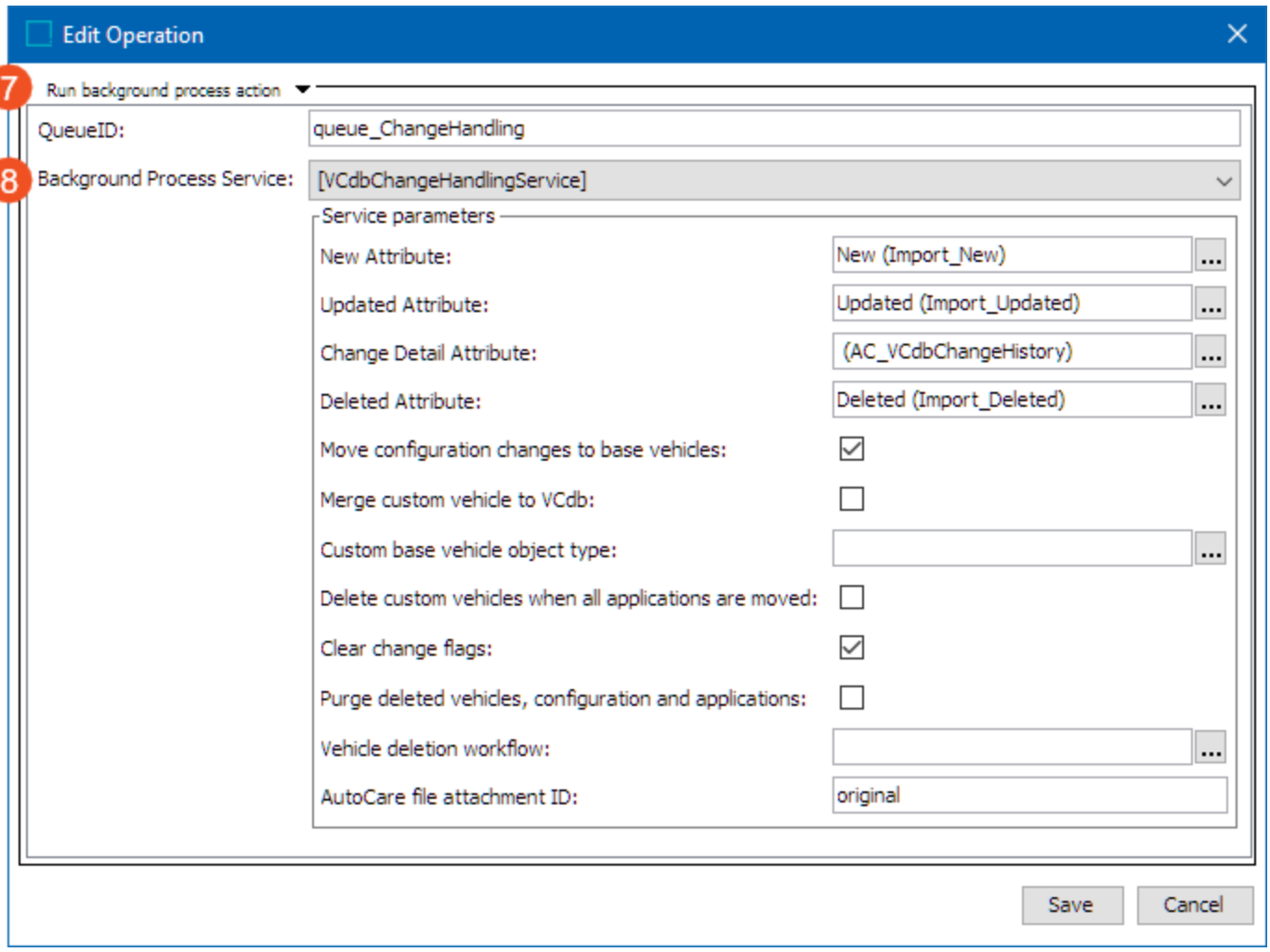

5. Within the State Editor, click the **On Entry** tab (as shown below).




6. Click on the **Add new Business Action** link to create a local action and click the edit button ( ) to open the Edit Operation dialog. For more information on creating a business rule, refer to the **Creating a Business Rule, Function, or Library** topic of the **Business Rules** section of the **STEP Online Help**.




7. Use the dropdown menu within the Edit Operation dialog to select **Import flow** > **Run background process action** operation.




8. Select **[VCdbChangeHandlingService]** 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_VCdbChangeHandling' 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 [VCdbChangeHandlingService].
   - **New attribute:** This is the required parameter that must be populated with the same attribute that is configured within the 'New object attribute' parameter in the 'Set import status attributes' operation. Click the ellipsis button (...) to find and select an attribute that is used to identify when an object is created while importing. When the attribute is configured, the system will search for all the VCdb objects where the configured attribute holds the value 'true' and executes the VCdb Change Handling business action on all the objects with New attribute value = true. For more information on how to configure this attribute in 'Set import status attributes' operation, refer to the **Business Action: Set Import Status Attributes** topic of the **Automotive Reference Guide** found within the **Solution Enablement** section of **STEP Online Help**.

- **Updated attribute:** This is the required parameter that must be populated with the same attribute that is configured within the 'Updated values attribute' parameter in the 'Set import status attributes' operation. Click the ellipsis button (...) to find and select an attribute that is used to identify when an object is changed while importing. When the attribute is configured, the system will search for all the VCdb objects where the configured attribute holds the value 'true' and executes the VCdb Change Handling business action on all the objects with Changed attribute value = true.
- **Change Detail Attribute:** The attribute configured in this parameter will store information about the changes that have been introduced in the latest VCdb file for Base Vehicle, Vehicle, and Vehicle Configuration objects. The information that is stored in the attribute includes the VCdb version date along with an explanation about the change. This feature helps users track how the Base Vehicle has changed / evolved over time.
- **Deleted attribute:** This is the required parameter that must be populated with the same attribute that is configured within the 'Deleted values attribute' parameter in the 'Set import status attributes' operation. Click the ellipsis button (...) to find and select an attribute that is used to identify when an object is deleted while importing. When the attribute is configured, the system will search for all the VCdb objects where the configured attribute holds the value 'true' and executes the VCdb Change Handling business action on all the objects with Deleted attribute value = true.
- **Move configuration changes to base vehicles:** Selecting this parameter will allow the background service to find all VCdb Vehicle Configurations that are flagged as new (New = true) or updated (Updated = true), and then flag the corresponding VCdb Base Vehicle(s) as updated (Updated = true). If the Base Vehicle is already flagged new (New = true), then such Base Vehicles will not be flagged separately.
- **Merge custom vehicle to VCdb:** Selecting this parameter will look for the VCdb Base Vehicles that are flagged as either (New = true) or updated (Updated = true) and then check if a custom vehicle exists in the system that matches the new VCdb Base Vehicle based on the custom Vehicle name,. For each custom vehicle that matches a new or updated VCdb Base Vehicle, the background process will determine if the application records that are created for the custom vehicles can be moved to the VCdb Base Vehicles.
- **Custom base vehicle object type:** Optional parameter to specify the object type of the custom vehicle object.
- **Delete custom vehicles when all applications are moved:** Selecting this parameter will delete the custom vehicles after the application records that are created for the custom vehicles are moved to the VCdb Base Vehicles. If unselected, the custom vehicles will not be deleted.
- **Clear change flags:** Selecting this parameter will clear the flags that are defined in the attributes that are configured within the 'New Attribute', 'Updated Attribute', and 'Deleted Attribute' parameters after the background service has been executed.

  **Note:** The 'Clear change flags' parameter should only be selected when it has been decided that Vehicle and Vehicle Configuration objects that are flagged for New, Updated, or Deleted will be of no more use by any other functionalities and should have the value of ‘true’ cleared on those objects. The flags on Base Vehicle objects will need to be cleared manually after users have determined what to do with New, Updated, or Deleted Base Vehicles.

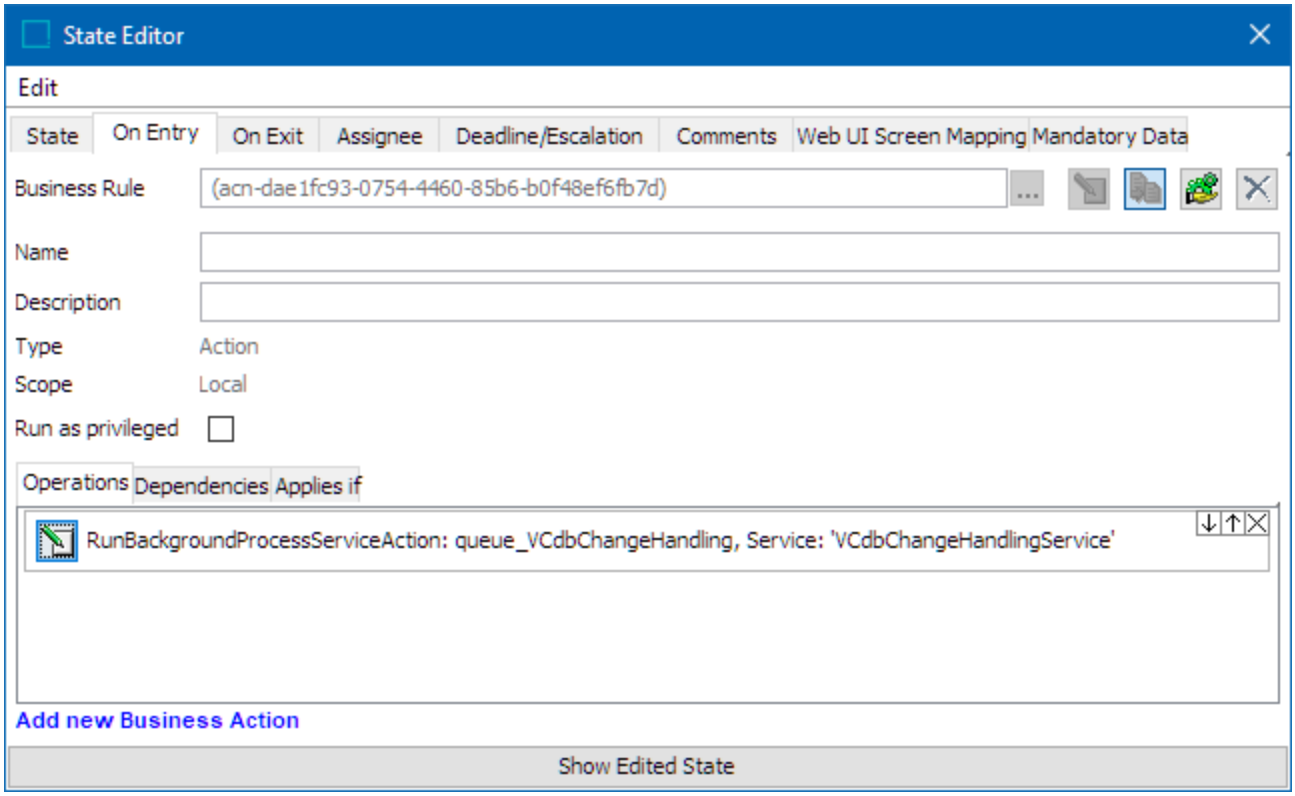
- **Purge deleted vehicles, configuration and applications:** Selecting this parameter will purge all VCdb configuration and Base Vehicle objects that are flagged for deletion (Delete = true), along with the application records that are referenced to the Base Vehicle.

- **Vehicle deletion workflow:** Select this parameter if the VCdb configuration and Base Vehicle objects that are marked for deletion should **not** get purged, but instead get routed to a workflow. (It is required that the workflow to handle deleted vehicles must be manually created first in order to be available to select within this parameter.)

> **Note:** Either leave 'Purge deleted vehicles, configuration and applications' and 'Vehicle deletion workflow' parameters unchecked, or select only one or the other, but do not select both because it is not possible to purge vehicle objects AND initiate it into a workflow. How the vehicle deletion workflow should be created is beyond the scope of this topic. It is a process that the customer needs to determine along with a Solution Consultant to help implement on to their system.

9. Click the **Save** button to save and close the configuration.

   Below is an example of how the State Editor should look prior to saving:




10. Close the State Editor, and save and close the STEP Workflow Designer.

### Configuring AutoCare VCdb Importer

The following topics provide the configuration steps necessary to allow users to be able to drag and drop VCdb 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 VCdb Imports
- Configuring a File Loading Widget for AutoCare VCdb Imports

## Prerequisites

It is expected that anyone configuring the AutoCare VCdb Import solution within a Web UI be familiar with the Web UI Designer, as basic concepts for working with the designer are not covered in this section. In addition, the user must have appropriate privileges to access the designer. For more information, refer to the **Designer Access** topic within the **Web User Interfaces** 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, a Node Details screen must be created. For information about configuring the Node Details Screen, refer to the **Node Details Screen** topic within the **Web User Interfaces** section of **STEP Online Help**.

Anyone configuring the AutoCare VCdb 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 Homepage Widget, including additional information about working with these widgets, refer to File Loading Widget topic, and Status Selector Homepage Widget topic, both within the **Web User Interfaces** section of the **STEP Online Help**.

#### Configuring an IIEP for AutoCare VCdb 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 VCdb data into the system. Once an IIEP is configured for AutoCare VCdb imports, VCdb 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, refer to the **AutoCare VCdb Importer** topic.

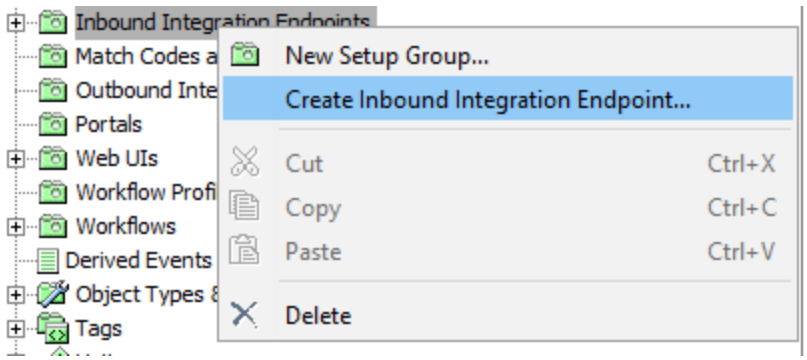
This section describes how to configure an IIEP that can allow for the automated processing of VCdb files. Each screenshot example within this section provides recommended values for the parameters and AutoCare VCdb Importer.

## Prerequisites

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

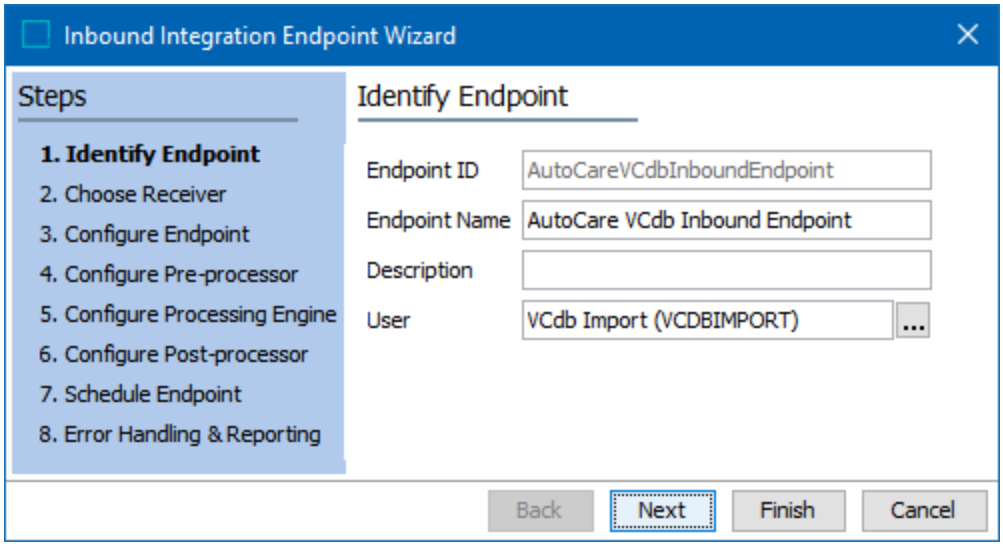
## Configuration Steps

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.

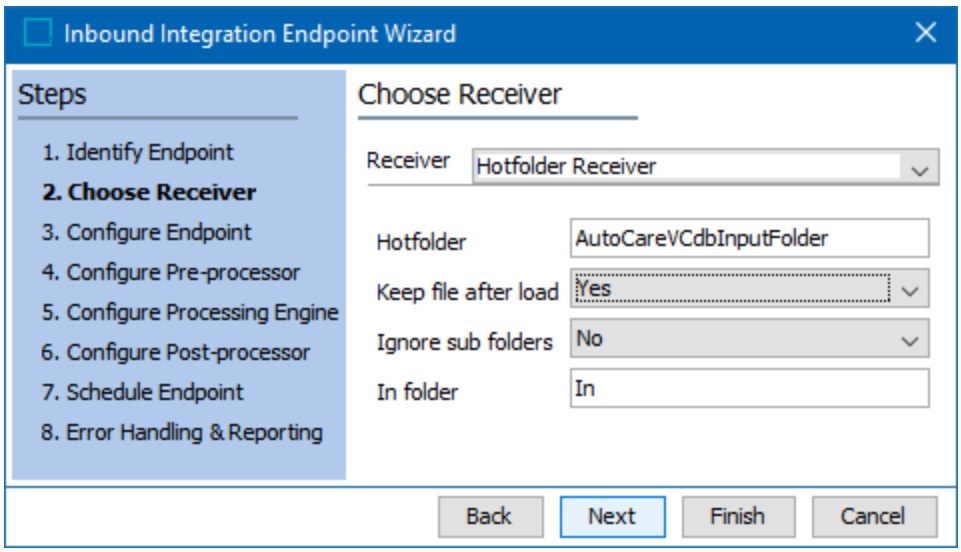
   In the example below, a 'VCdb Import' user was created prior to configuring this IIEP. This is recommended to more easily track when this IIEP is responsible for changes to data.




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

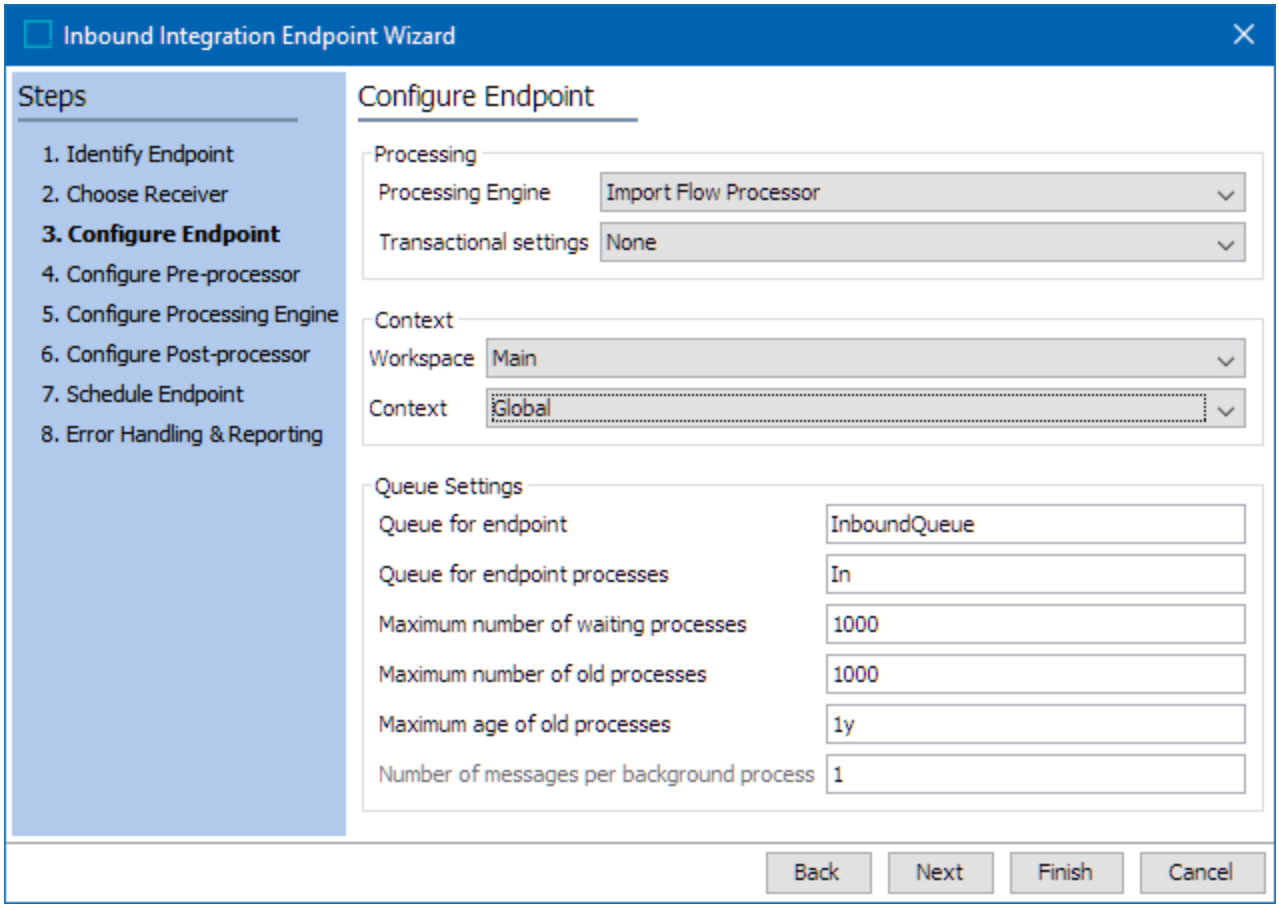
3. Click the **Next** button, and the Choose Receiver parameters will display. By default, the parameters are populated as recommended and shown below, except the Hotfolder parameter. This mandatory parameter must be populated with a hotfolder name before the Next button will enable.

   **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, refer to the **IIEP - Choose Receiver** topic within the **Inbound Integration Endpoint** section of **STEP Online Help**.

4. 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

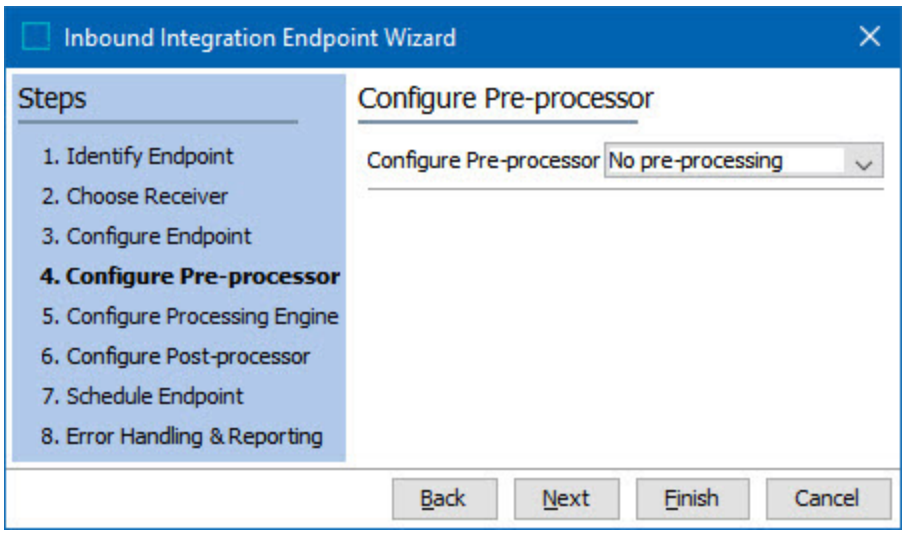



**Important:** The Import Flow Processor is only responsible for picking up files from the configured hotfolder, creating a Controller Entity object, and updating the Entity with the Import Flow State BGP and Status. The Import Flow Processor works with the Background Process Service for each workflow 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.

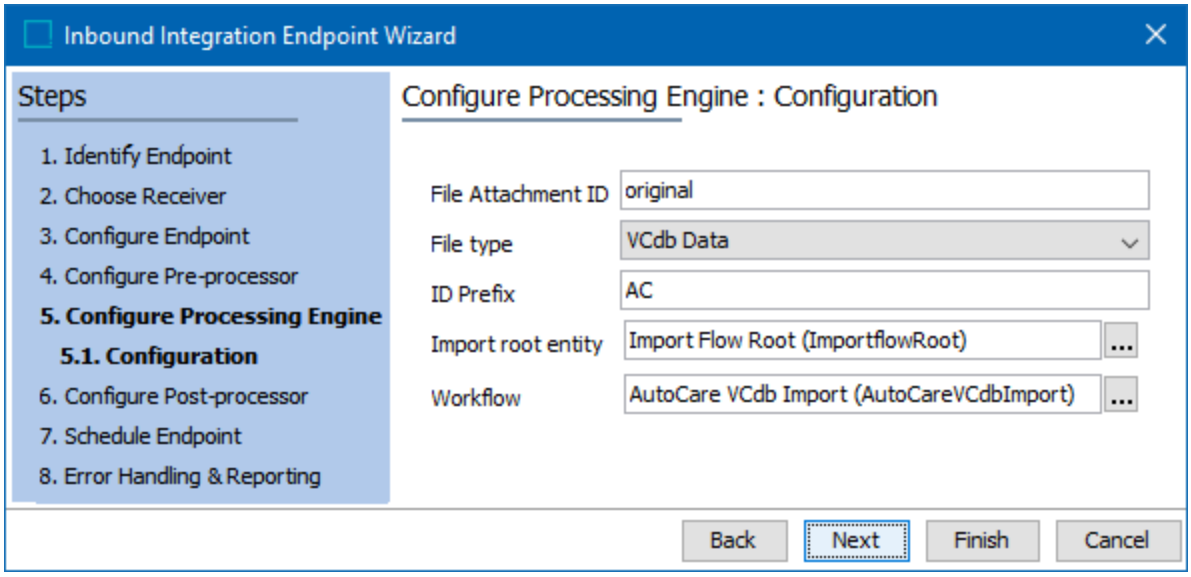
For more information about the parameters, refer to the **IIEP - Configure Endpoint** topic within the **Inbound Integration Endpoint** section of **STEP Online Help**.

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




For more information about the parameter, refer to the **IIEP - Configure Pre-processor** topic within the **Inbound Integration Endpoint** section of **STEP Online Help**.

6. 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.



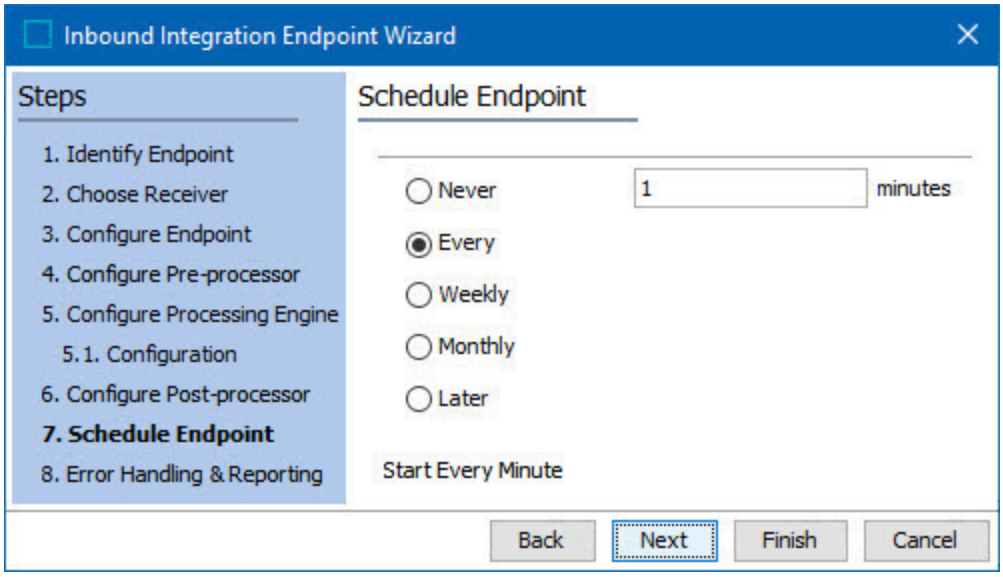

The prior Configure Endpoint step determines the options available for the Configure Processing Engine step.

- Click the 'File type' parameter, and select the **VCdb 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 VCdb 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.

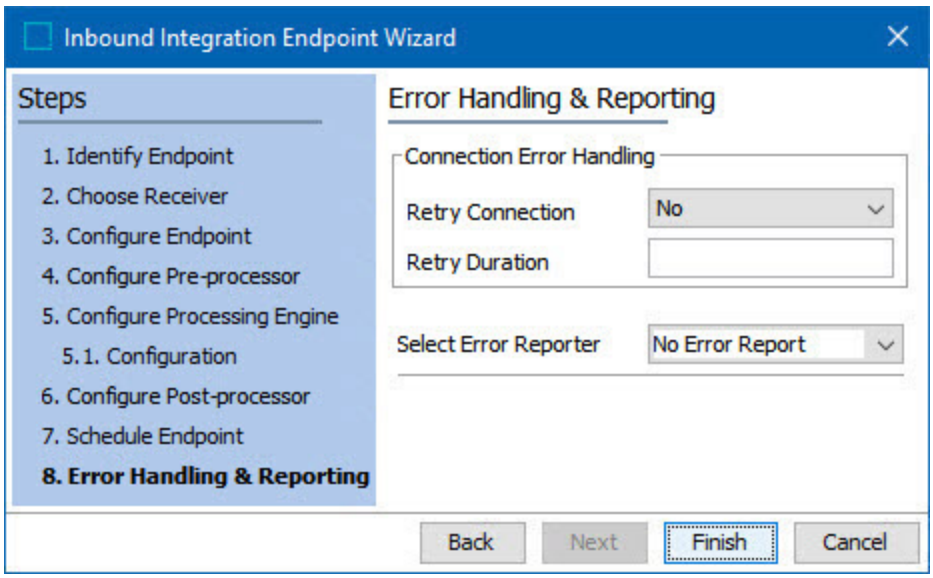
7. Click the **Next** button, and the Schedule Endpoint parameters will display. By default, 'Never' is selected. Optionally, update the values to those shown below.




**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, refer to the **IIEP - Schedule Endpoint** topic within the **Inbound Integration Endpoint** section of **STEP Online Help**.

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




For more information about the parameter, refer to the **IIEP - Error Handling & Reporting** 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, refer to 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, refer to the **Configuring a File Loading Widget for AutoCare VCdb Imports** topic.

#### Configuring a File Loading Widget for AutoCare VCdb Imports

When configured, Web UI users can import VCdb 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 VCdb Importer must be configured within workbench. For more information, refer to the **Configuring an IIEP for VCdb 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.

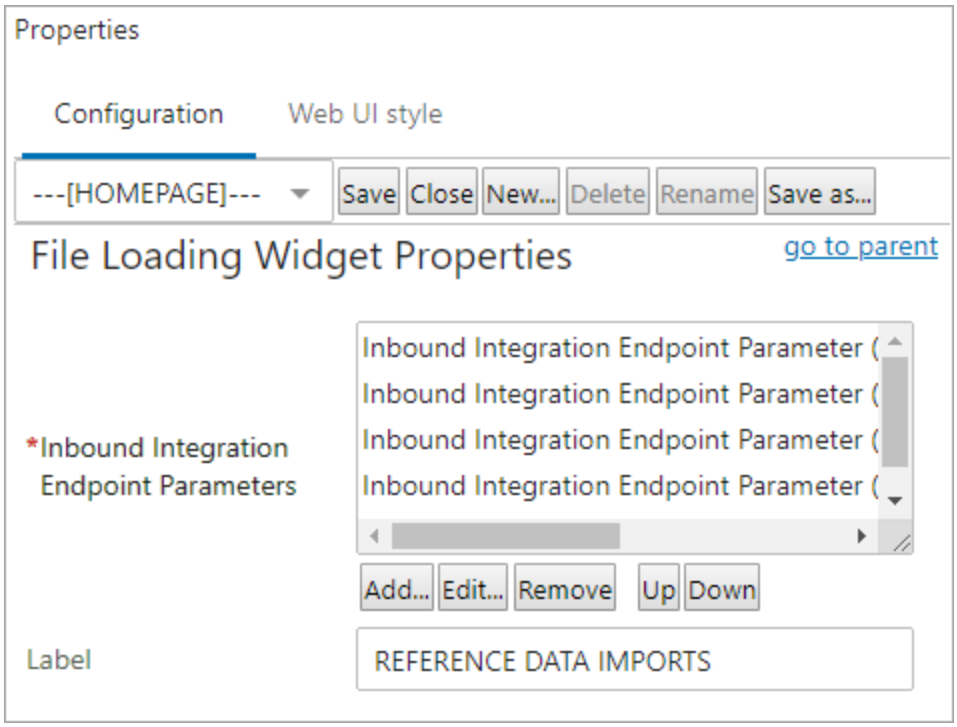
## Configuration

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

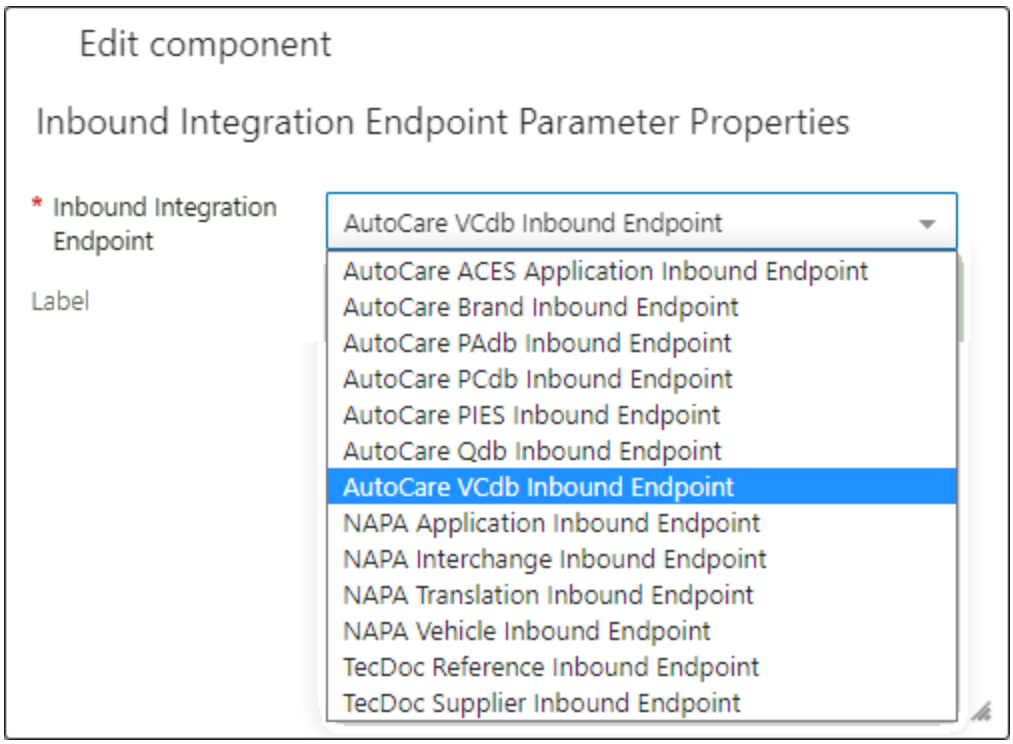
This topic describes how to configure a File Loading Widget so that users can drag and drop AutoCare VCdb files onto a File Loading Widget on a Web UI Homepage.

**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, refer to 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 VCdb Inbound Endpoint** (the IIEP created for AutoCare VCdb imports).

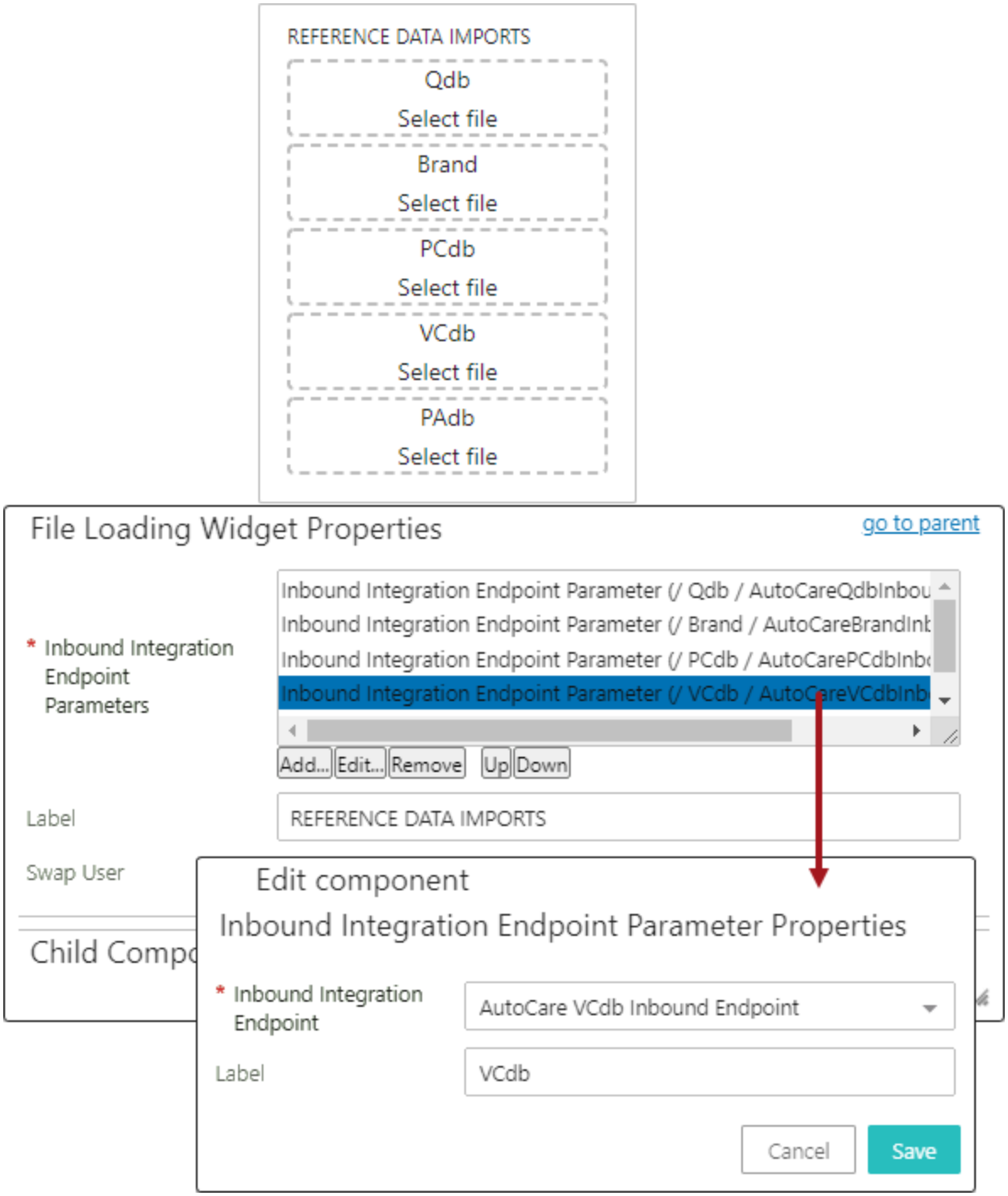



**Note:** If the desired IIEP does not display in the dropdown, then it can be created using the steps described in the **Configuring an IIEP for VCdb Imports** topic.

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 PAdb imports are added within the same File Loading Widget as the IIEP for the AutoCare VCdb imports.




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, refer to 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, refer to 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 that 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, refer to 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 that provided '2014-11-28.' |

# 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 Homepage Widget, including additional information about working with these widgets, refer to File Loading Widget topic, and Status Selector Homepage Widget topic, both within the **Web User Interfaces** section of the **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, refer to 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, refer to the **Workflows** section of **STEP Online Help**.

For general information about the File Loading Widget and Status Selector Homepage Widget, including additional information about working with these widgets, refer to File Loading Widget topic, and Status Selector Homepage Widget topic, both within the **Web User Interfaces** section of the **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, a Node Details screen must be created. For information about configuring the Node Details Screen, refer to the **Node Details Screen** topic within the **Web User Interfaces** section of **STEP Online Help**.

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, refer to 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 Homepage 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.

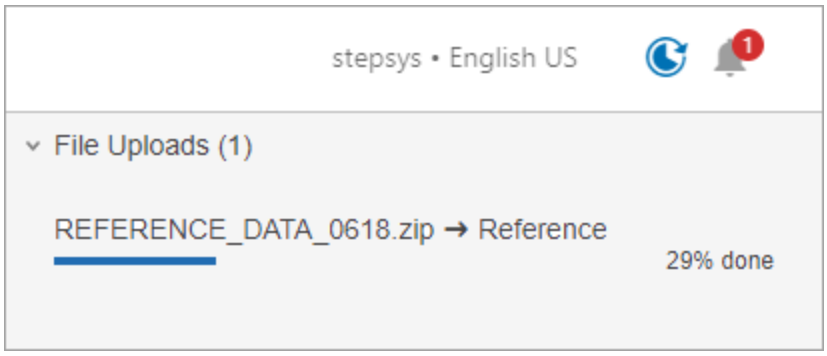
#### Using the TecDoc Reference Data Importer in Web UI

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, refer to 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 'Recent background processes' side panel.

   In the example below, the 'Recent background processes' side panel is expanded and the file, 'REFERENCE_DATA_0618.zip' is 29% processed.




   For more information on using the 'Recent background processes' side panel, refer to the **Recent Background Processes Side Panel** 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.

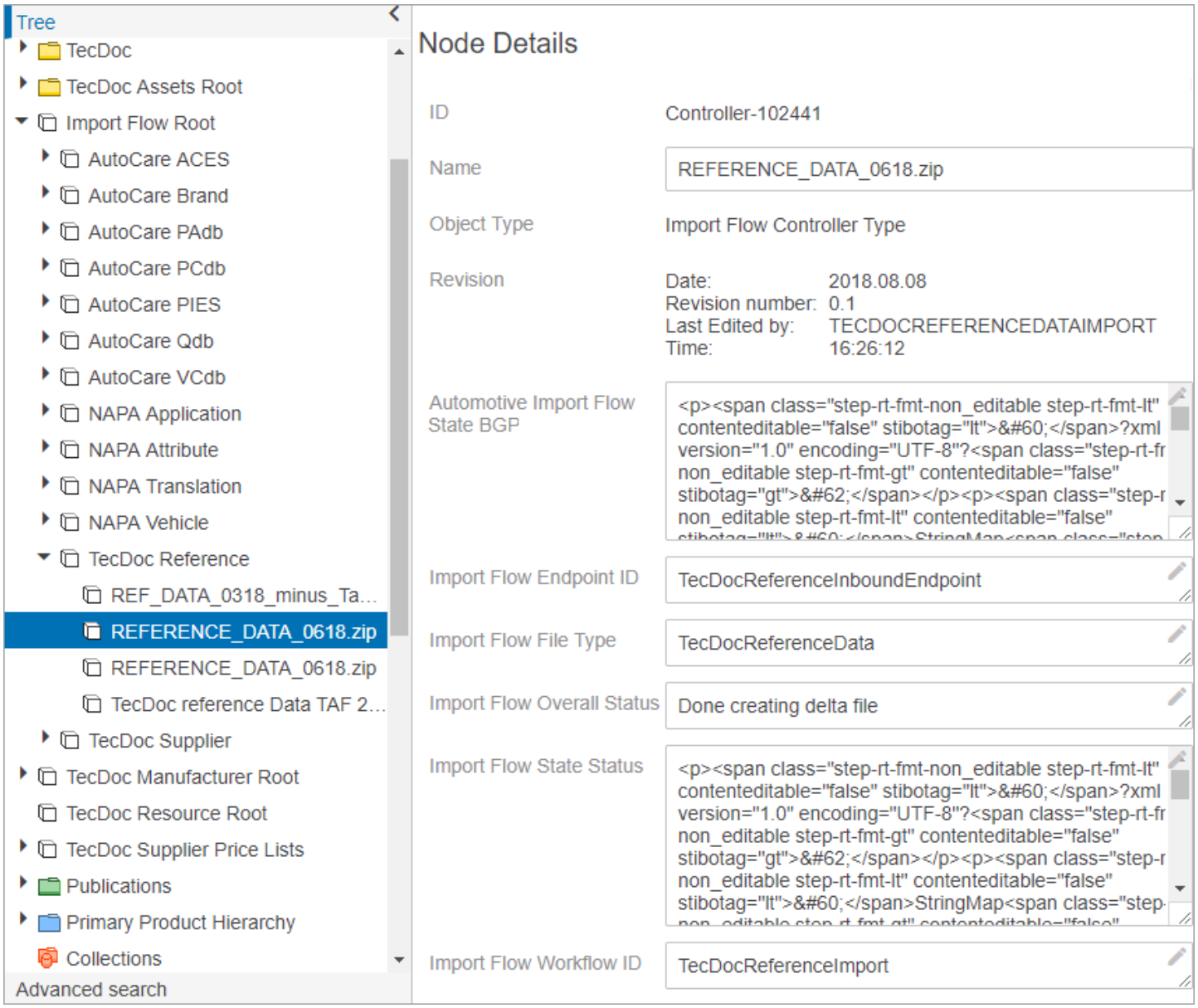
**Important:** It is critical to understand that it is only the Controller Entity that is in the workflow - the objects being acted on (created / updated / deleted) via information supplied in the import file are not in the workflow. Therefore, running standard business actions acting on current object 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, refer to the **Import State** topic within the **Default Workflow States and Functions** section.

4. If a Node 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 Node Details screen has not been configured. For information about configuring the Node Details Screen, refer to the **Node Details Screen** topic within the **Web User Interfaces** section of **STEP Online Help**.

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.



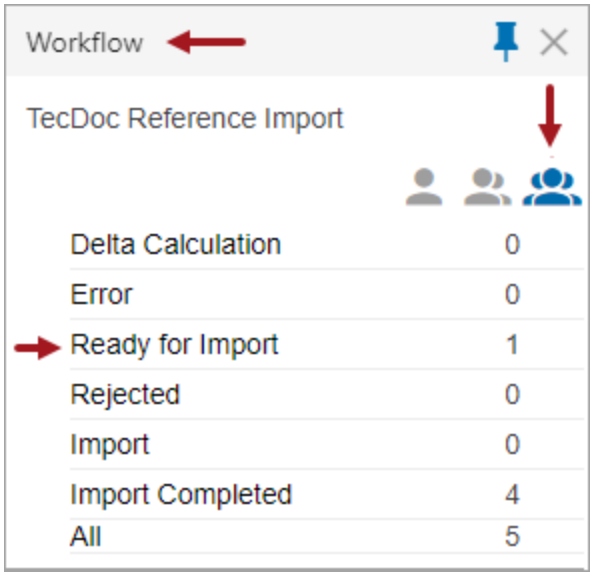

- **Name:** STEP Name of the Controller entity. This is generated from the uploaded file original name.
- **Automotive Import Flow State BGP:** Attribute used to store the IDs of the background processes.
- **Import Flow State Status:** Attribute used to store and display the status of each process (rather 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 Homepage Widget on the Web UI homepage and/or the Status Selector component on the left side panel.

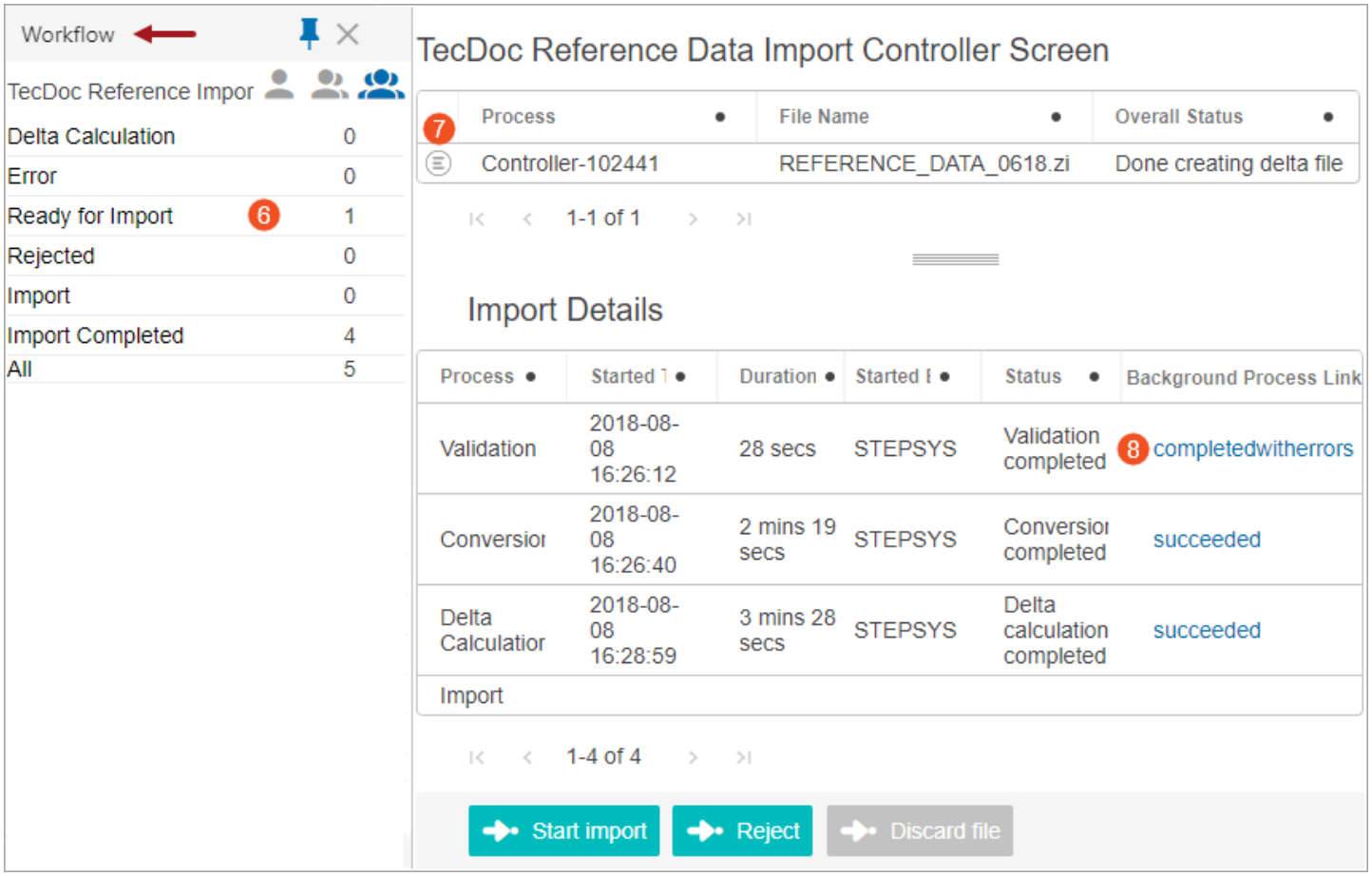
   In the example below, the Workflow panel is expanded, and the **Triple user button** is selected so that all items assigned to any user are displayed. Notice that one file is in the 'Ready for Import' state, and four files in the 'Import Completed' state, for a total of five files.

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

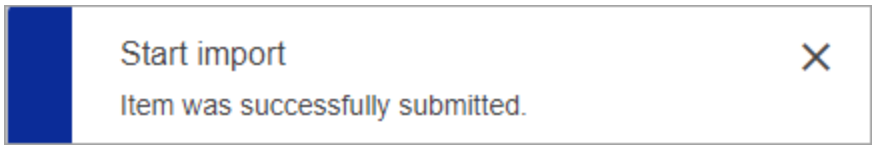



6. Clicking on the **Ready for Import** state within the Status Selector homepage and/or the left side panel widget will navigate the user to the TecDoc Reference Data Import Controller Screen with only those Controllers in the 'Ready for Import' state displayed.
7. Clicking the Controller icon displays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

   In the example below, Controller '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.

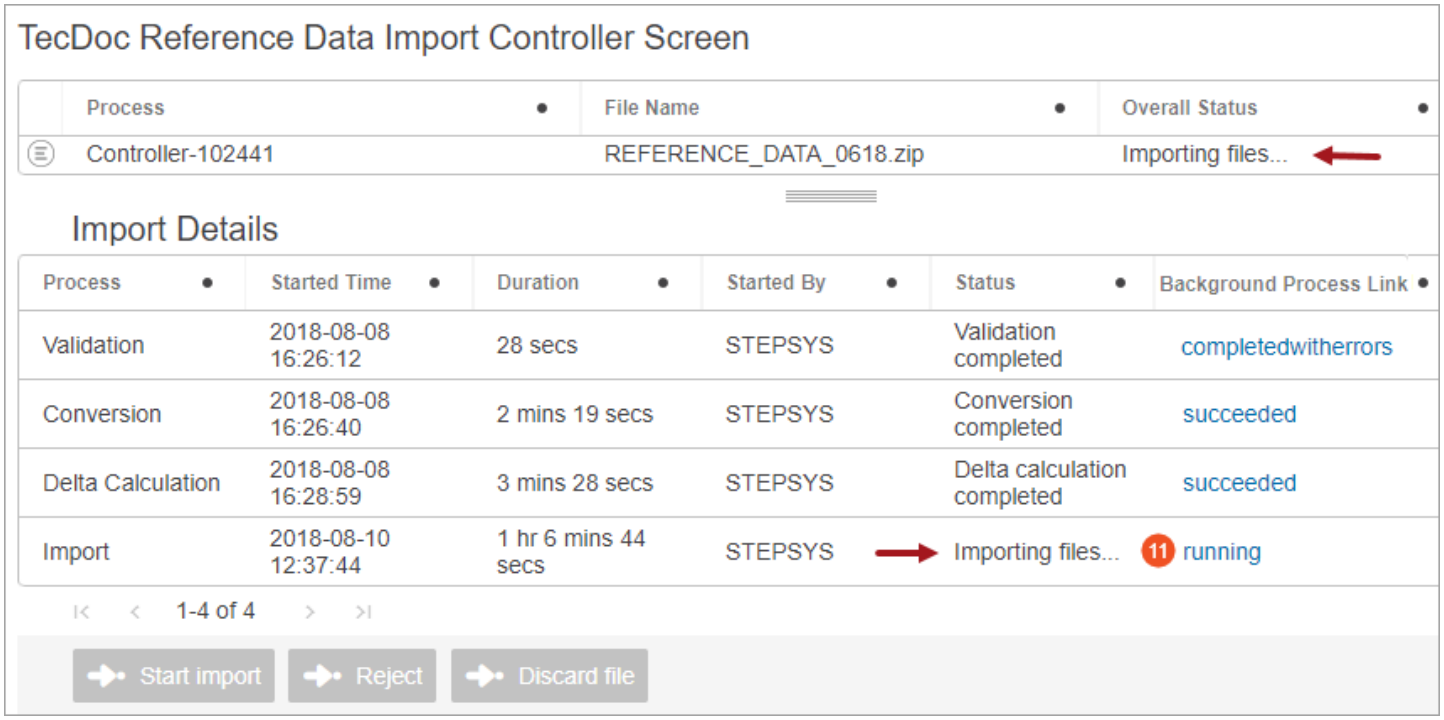



8. Optionally, before starting the import, clicking on the 'completedwitherrors' Background Process Link will display the Background Process Details screen where users can view or download the details of the validation process.
9. Click the **Reject** button to move the Controller to the Rejected state. For more information, refer to 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, refer to 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 Homepage Widget on the Web UI homepage and/or the Status Selector component in the left side panel. More detailed progress can be monitored in the import process Status column that is available within the Import Controller Screen.

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.'




11. 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, refer to the **Import Completed State** topic within the **Import Framework** section.

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

    For more information on automotive workflow states, refer to 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, refer to 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.'<br><br>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.<br><br>For more information, refer to 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, refer to 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, refer to 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, refer to 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

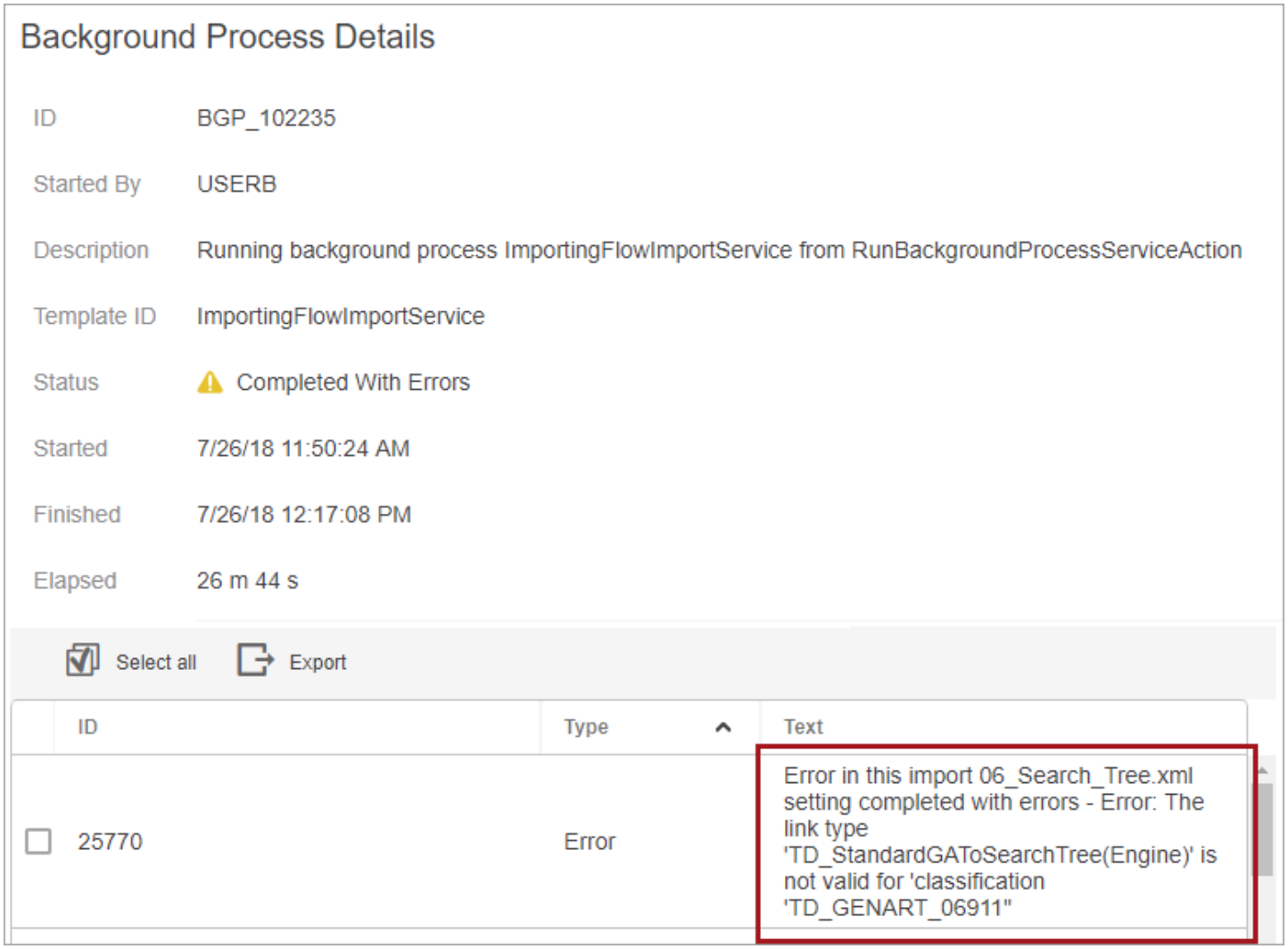
Error in this import 06_Search_Tree.xml setting completed with errors - Error: The link type <'Reference ID'> is not valid for 'classification 'TD_GENART_<GenArtNr>."

## 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

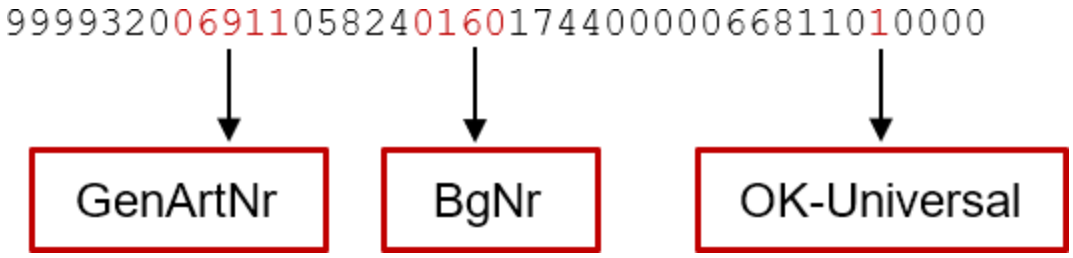
The screenshot below displays this error message example within a Background Process Details Web UI screen.




## 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.

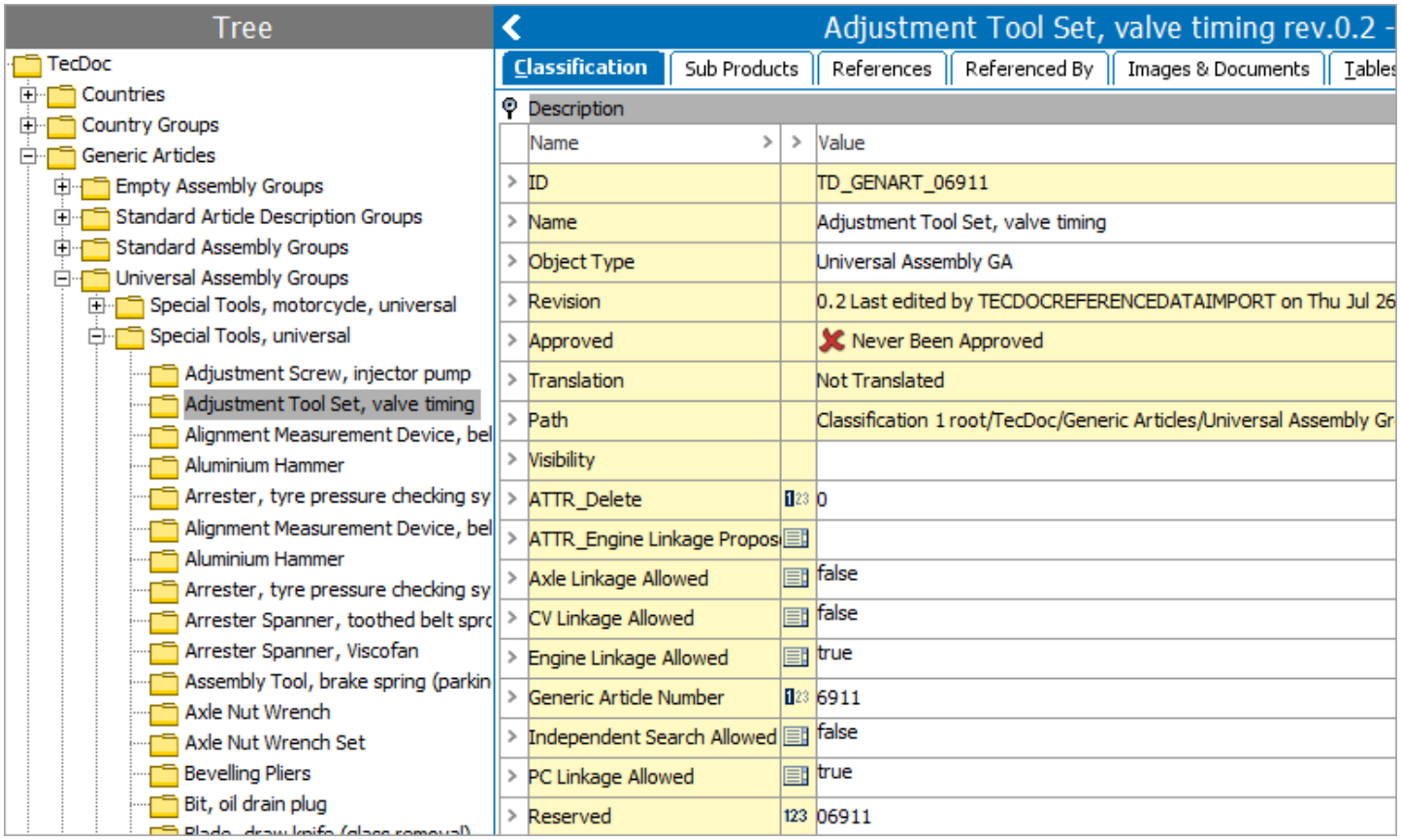


## Understanding Data Table 320

The following important Delta Keys are within this table:

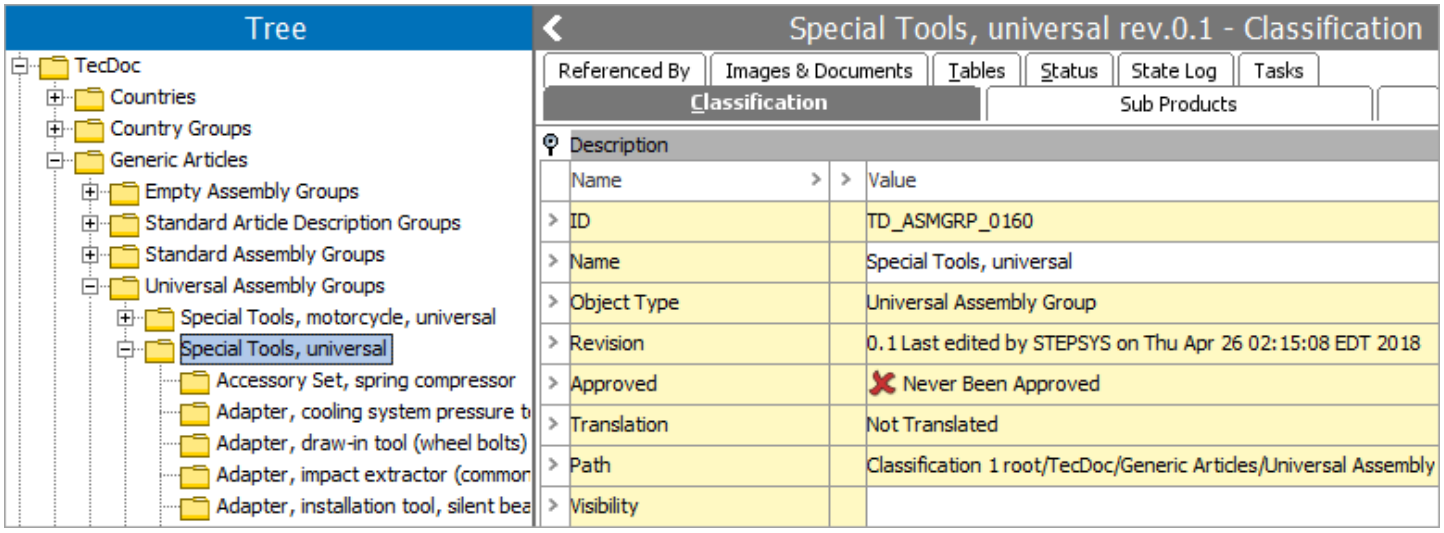
**GenArtNr:** Five digit number representing a Generic Article (GA) number. Stored in STEP as a 'Universal Assembly GA' Object Type using the following prefix: 'TD_GENART_.'

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).




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

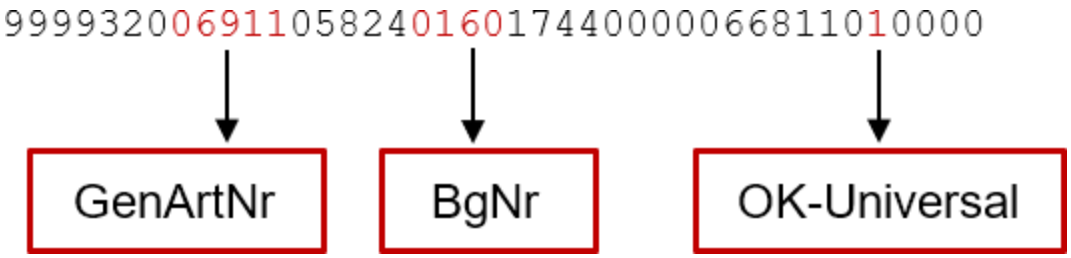
In the example below, BgNr 0160 is displayed in Workbench as the 'Universal Assembly Group' Object Type, Special Tools, universal (TD_ASMGRP_0160).




**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, refer to 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 |

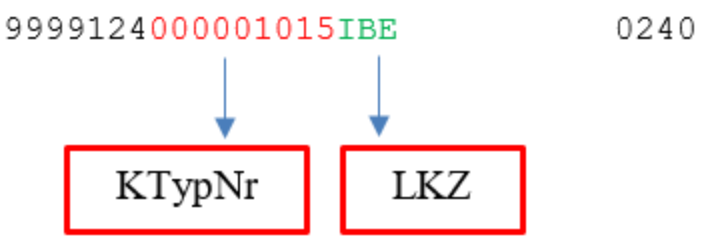
Select all Export

| ID | Type | Text |
|---|---|---|
| 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

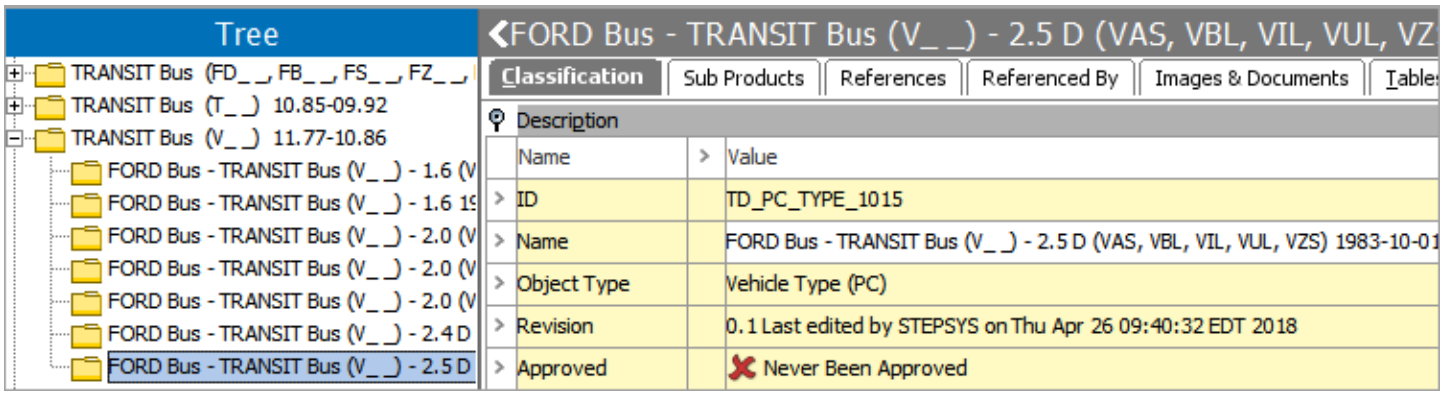
Data Table 010 lists Country Codes featured in Delta Key **LKZ**.

## Understanding Data Table 124

The following important Delta Keys are within this table:

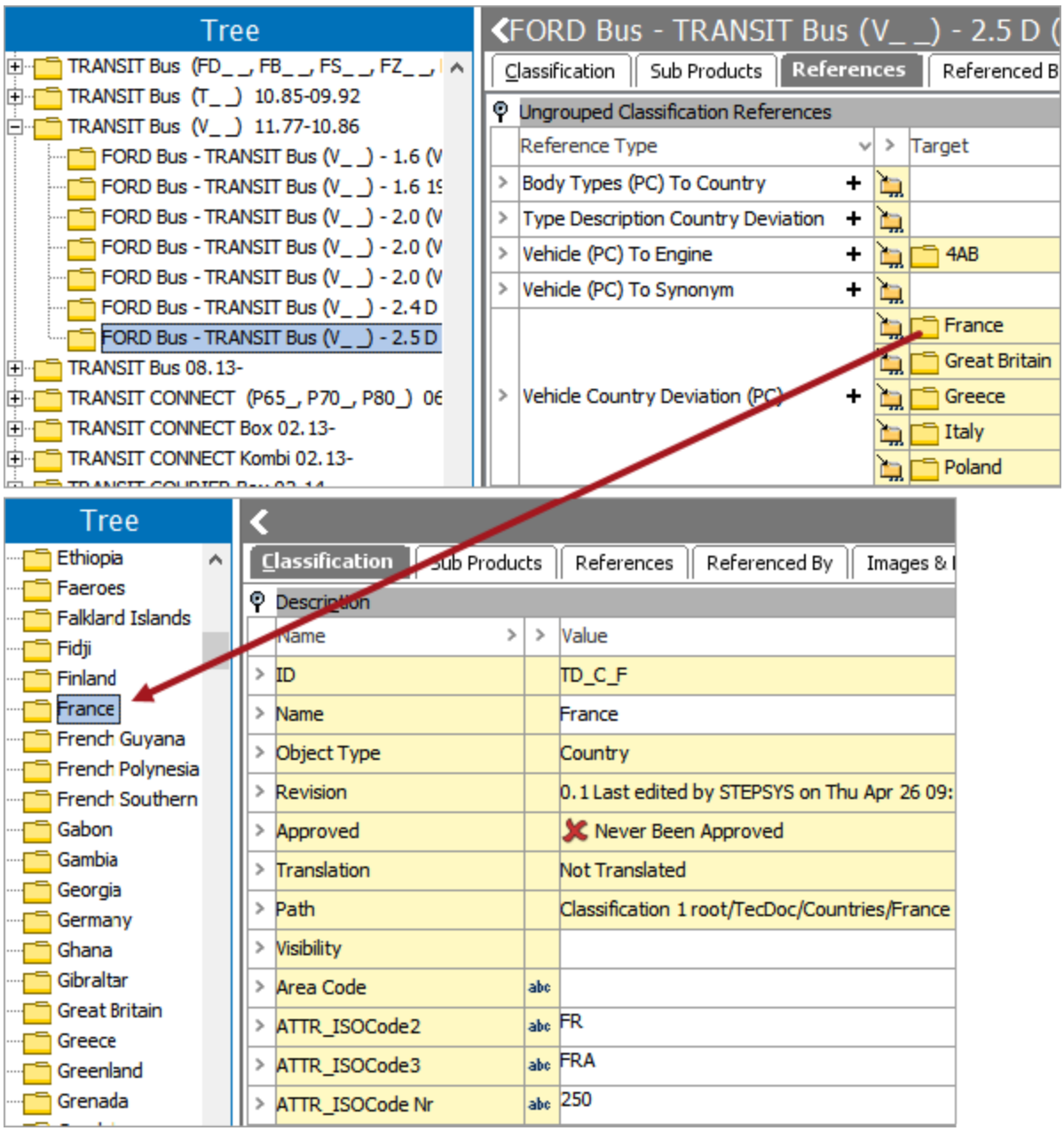
**KTypNr:** Nine digit number representing the Vehicle Types.

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).




**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.




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, refer to 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

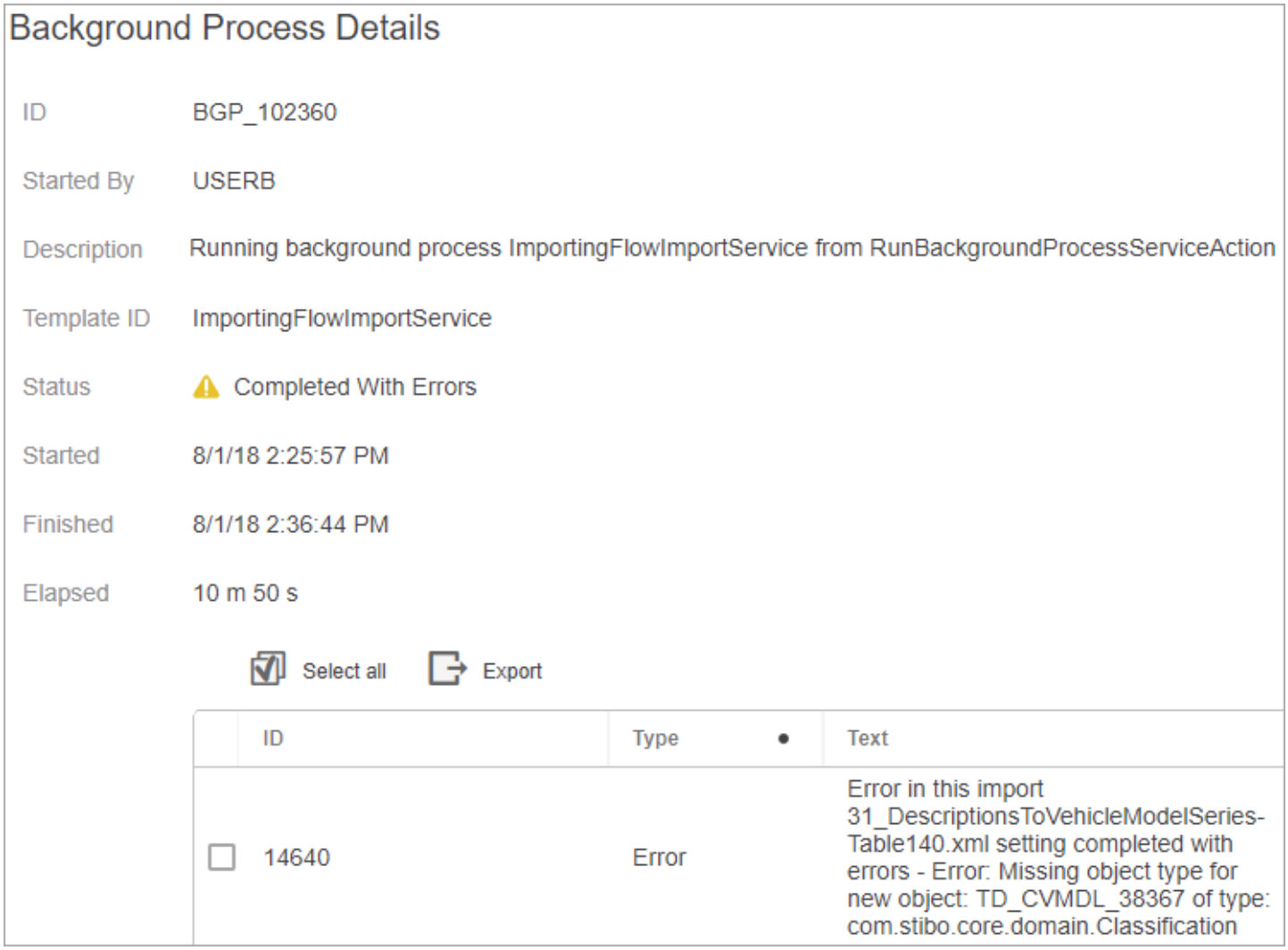
Error in this import [import file name] setting completed with errors - Error: Missing object type for new Object: [Object ID] of type: com.stibo.core.domain.Classification.

## 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.




## Explanation

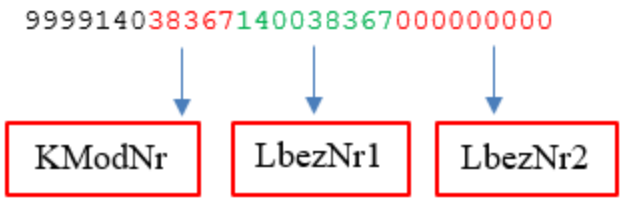
Because this error pertains to the 'Vehicle Model Series (CV)' Object (featured in Data Table 110) and adding Additional Descriptions to Vehicle Model Series (featured in Data Table 140), it is important to understand Data Tables 110 and 140.

## 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.



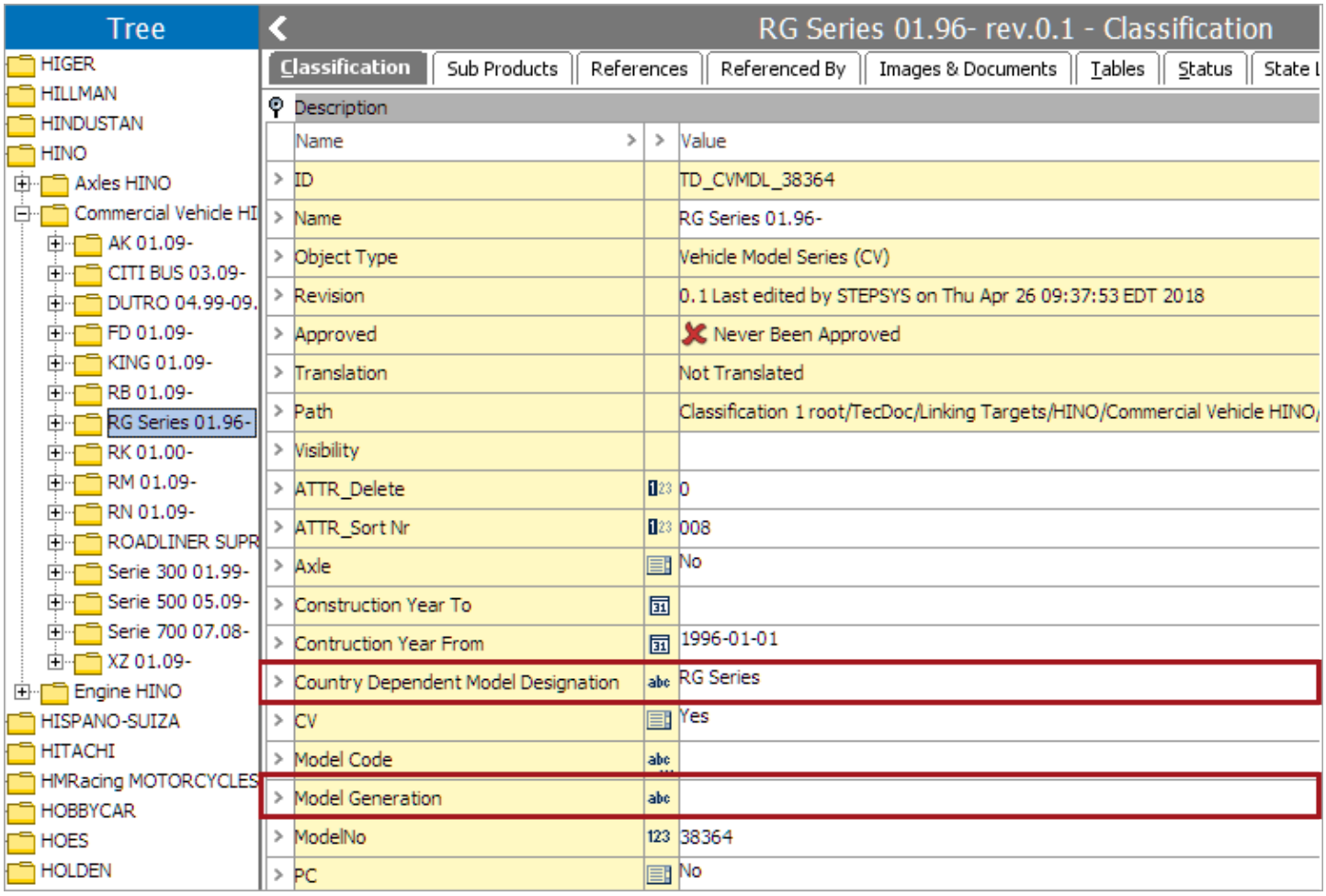

The following important Delta Keys are within this table:

**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.




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, refer to 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

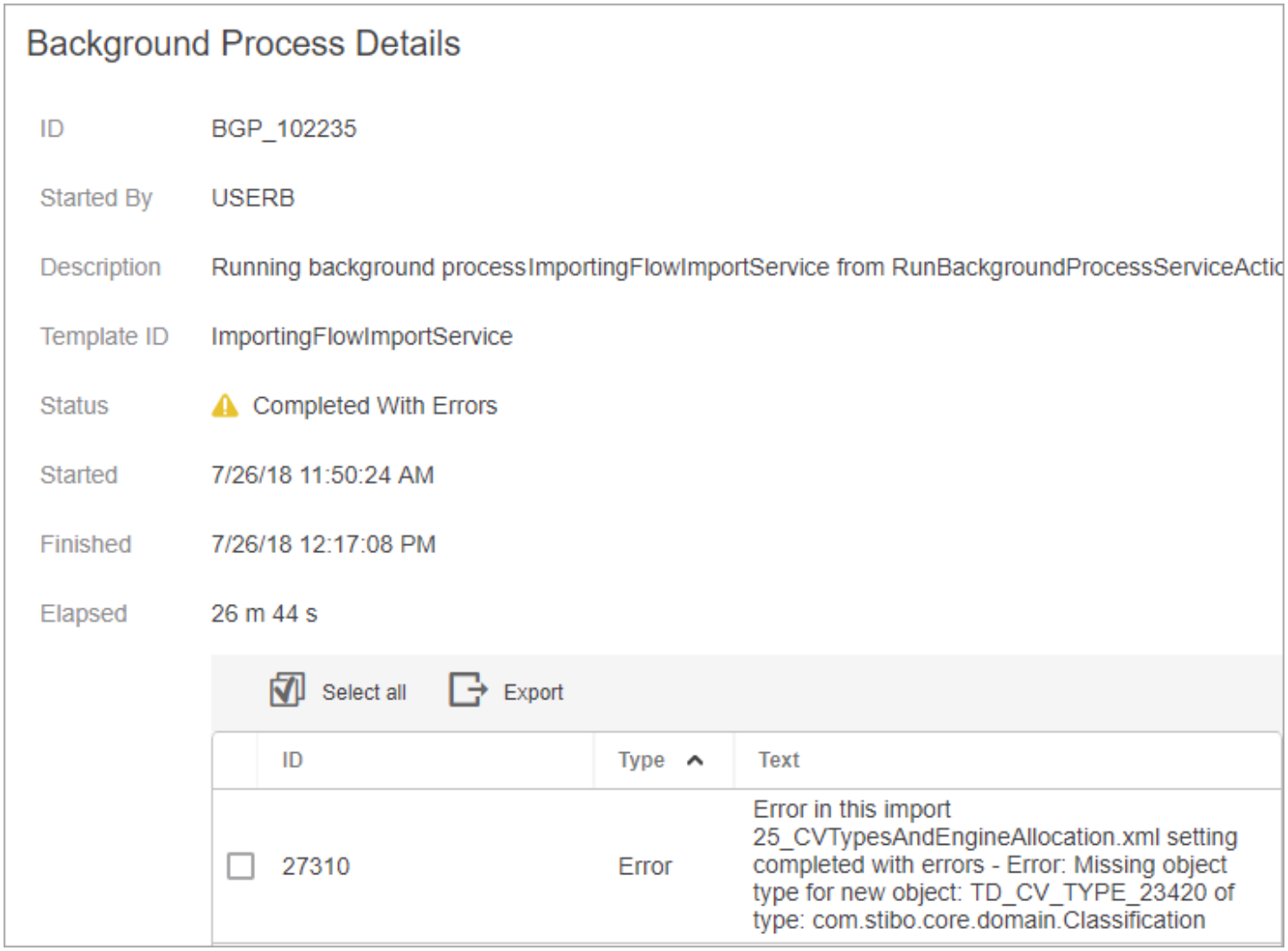
Error in this import 25_CVTypesAndEngineAllocation.xml setting completed with errors - Error: Missing object type for new object: <Object ID> of type: com.stibo.core.domain.Classification

## 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

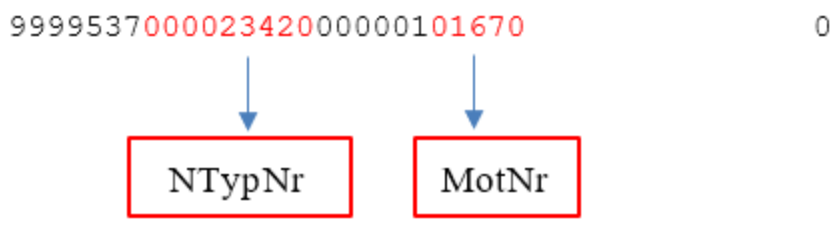
The screenshot below displays this error message example within a Background Process Details Web UI screen.




## 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.

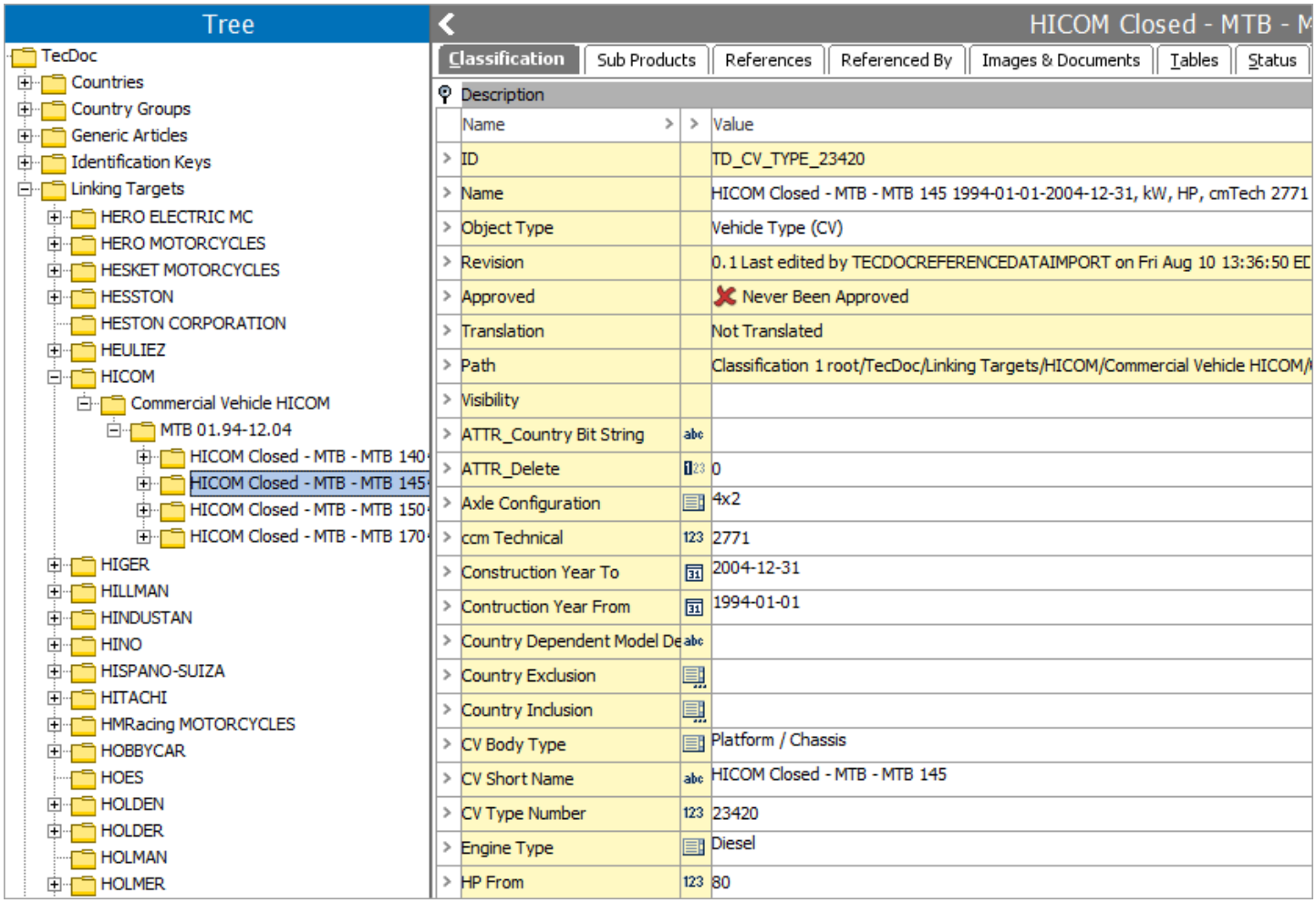


## Understanding Data Table 537

The following important Delta Keys are within this table:

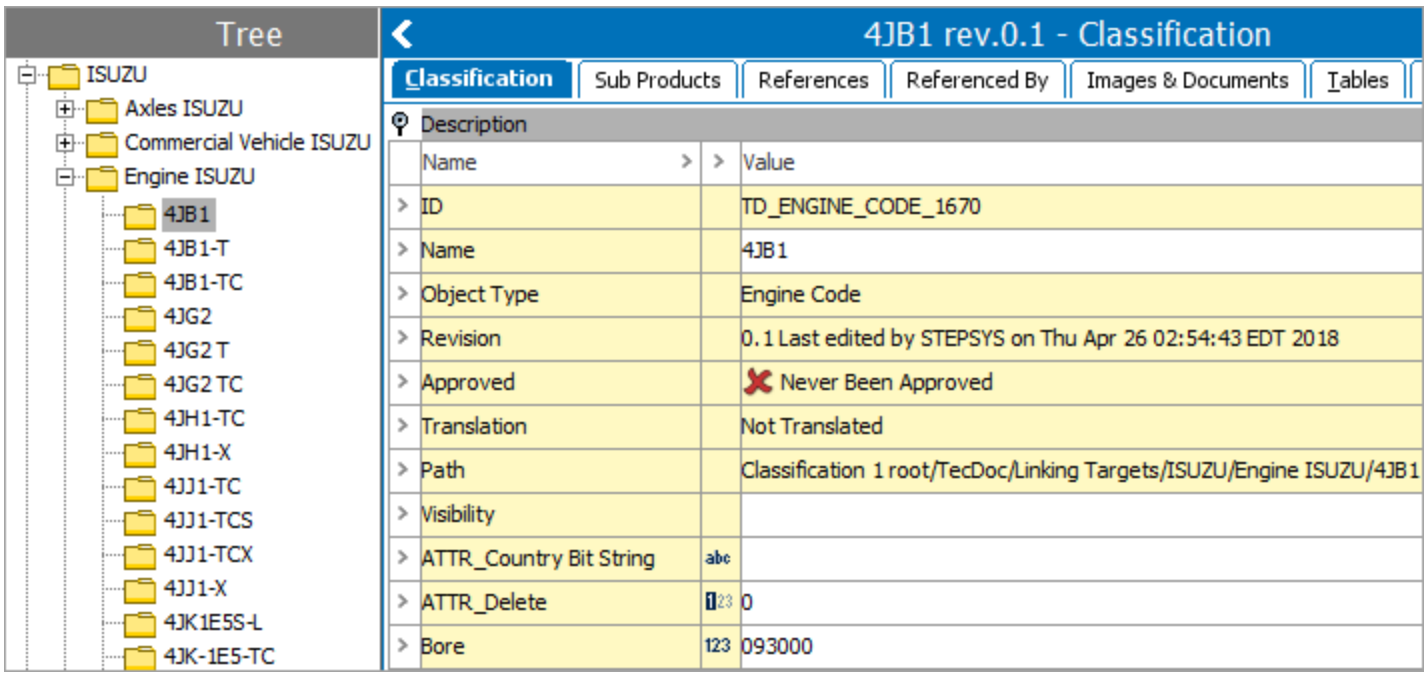
**NTypNr:** Nine digit number representing a Commercial Vehicle Type.

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).




**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).




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, refer to 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

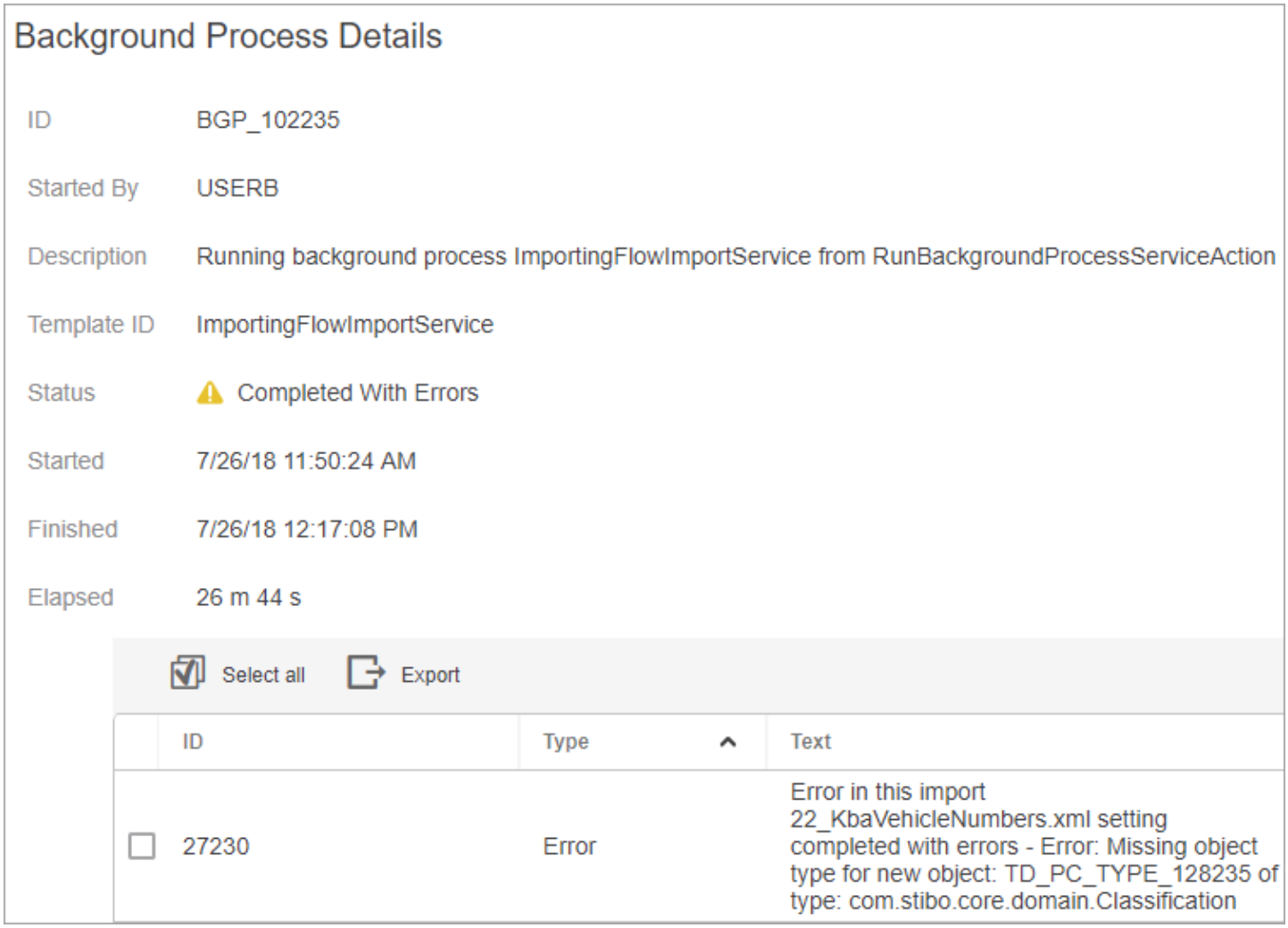
Error in this import [import file name] setting completed with errors - Error: Missing object type for new object: [<Object ID>] of type: com.stibo.core.domain.Classification

### 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

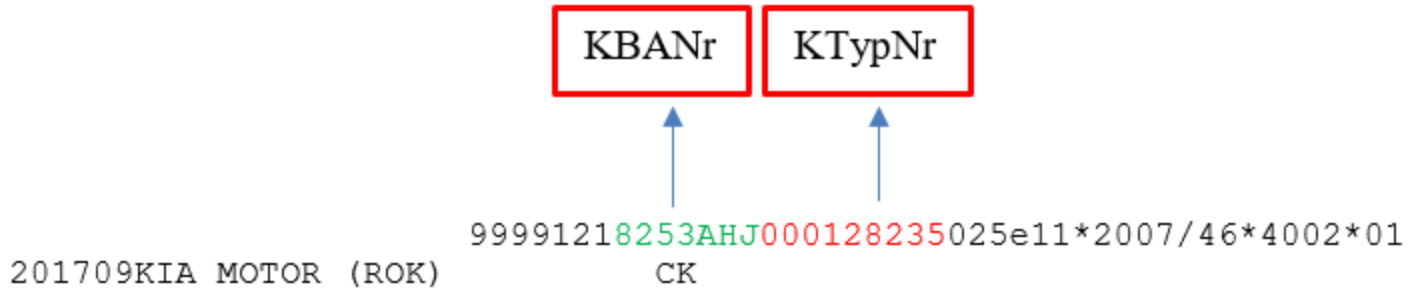
The screenshot below displays this error message example within a Background Process Details Web UI screen.




## 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

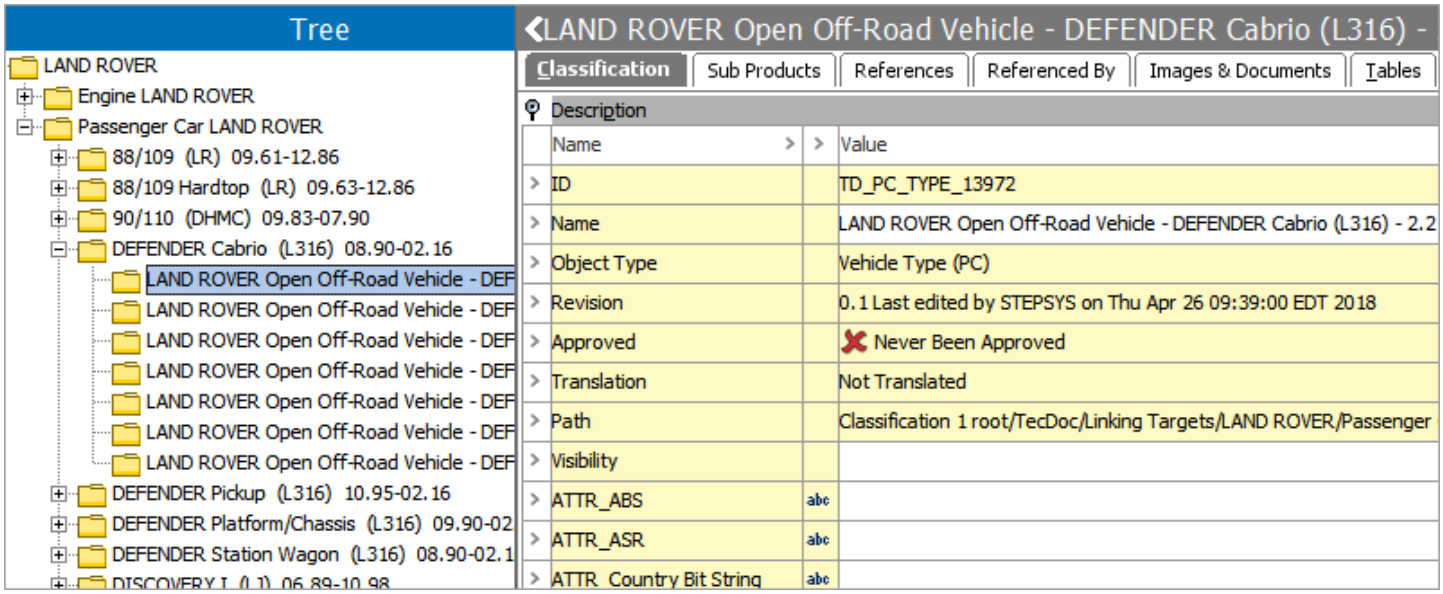
Data Table 120 lists Passenger Car Types that are featured in Delta Key **KTypNr**.

## Understanding Data Table 121

The following important Delta Keys are within Data Table 121:

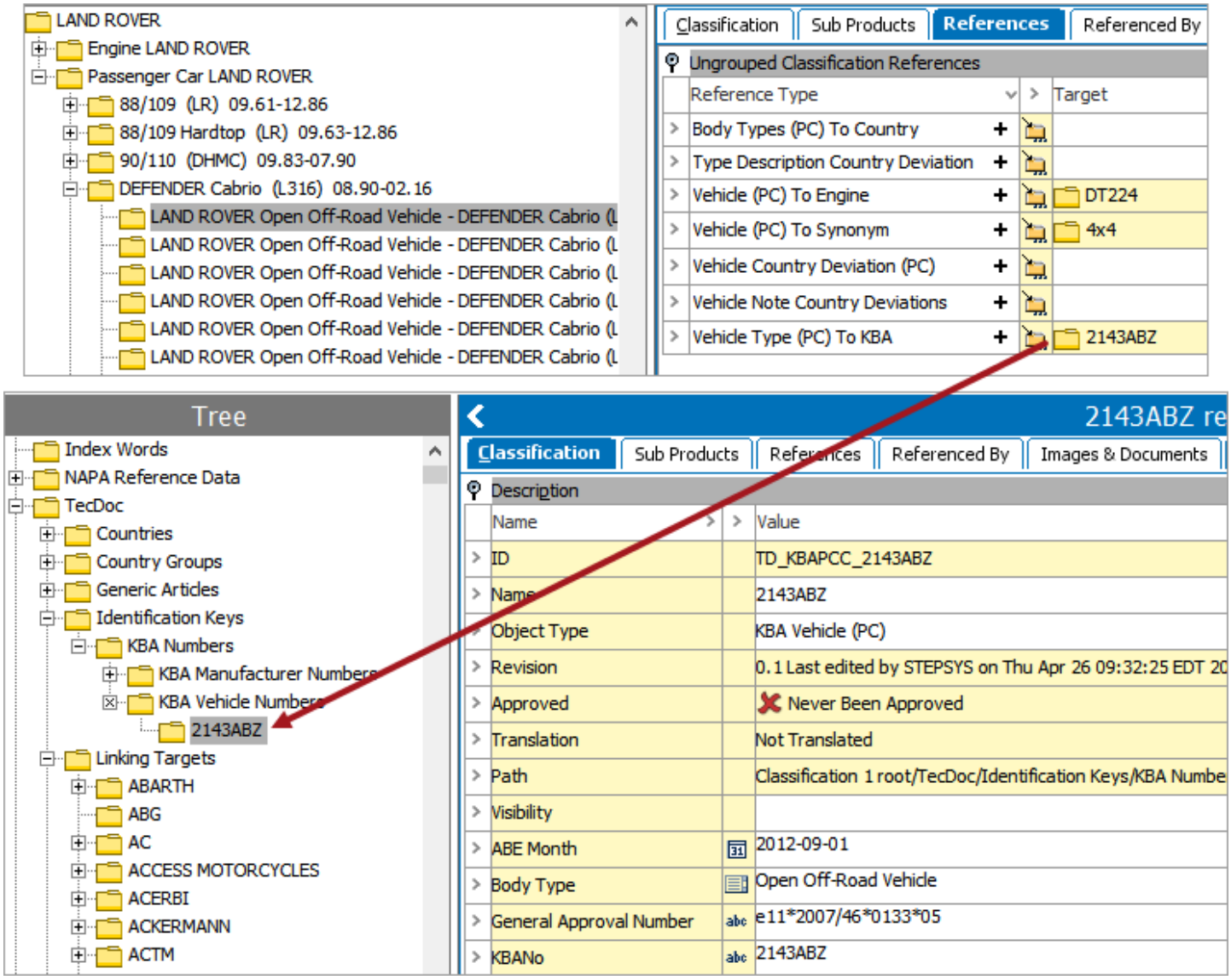
**KTypNr:** Nine digit number representing the PC Vehicle Type number. Shown above in red text.

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.




**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.




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, refer to 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, refer to 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

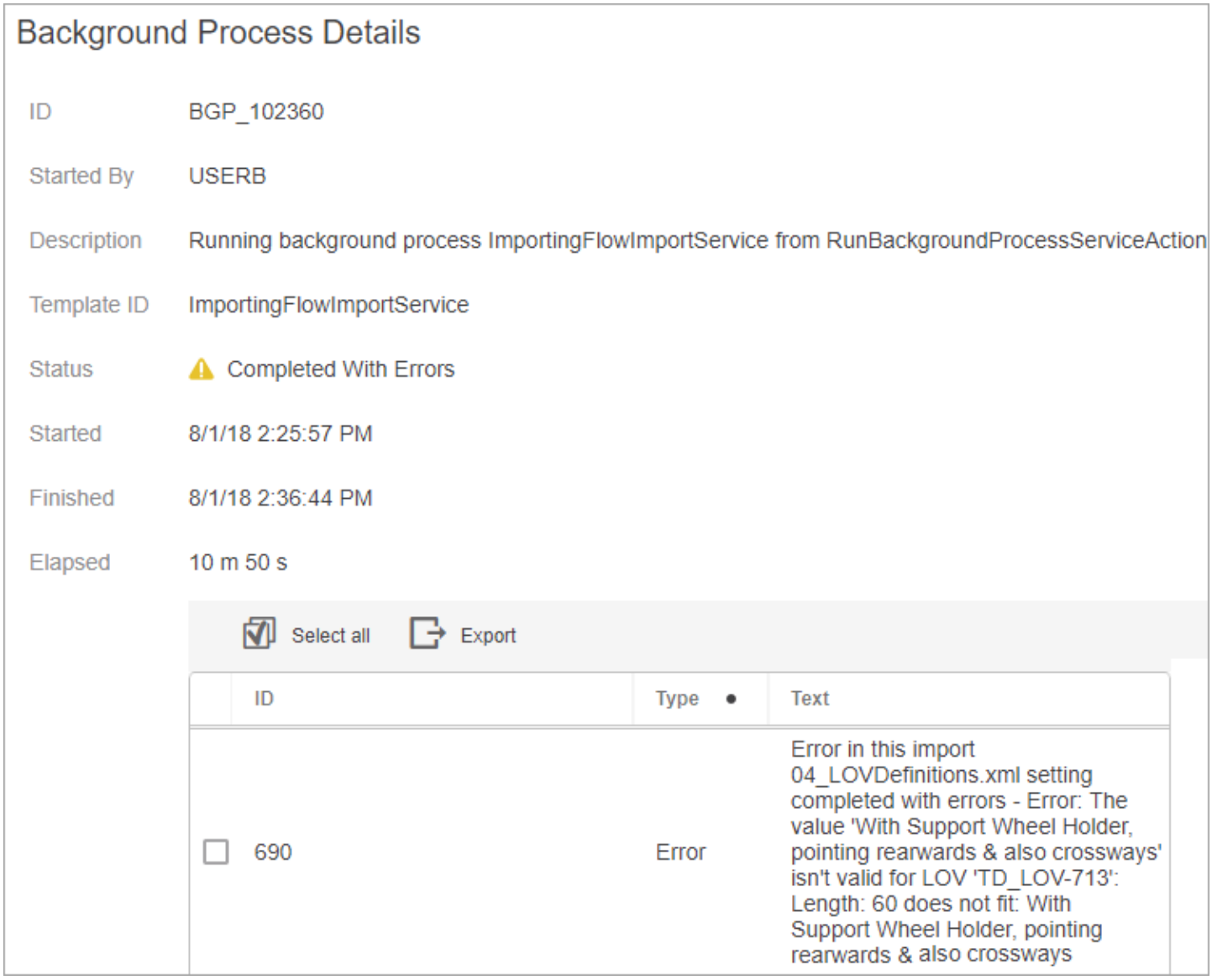
Error in this import [import file name] setting completed with errors - Error: The value [imported LOV value] isn't valid for LOV [LOV ID]: Length: [Maximum Length value] does not fit: [imported LOV value].

## 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.




## 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.

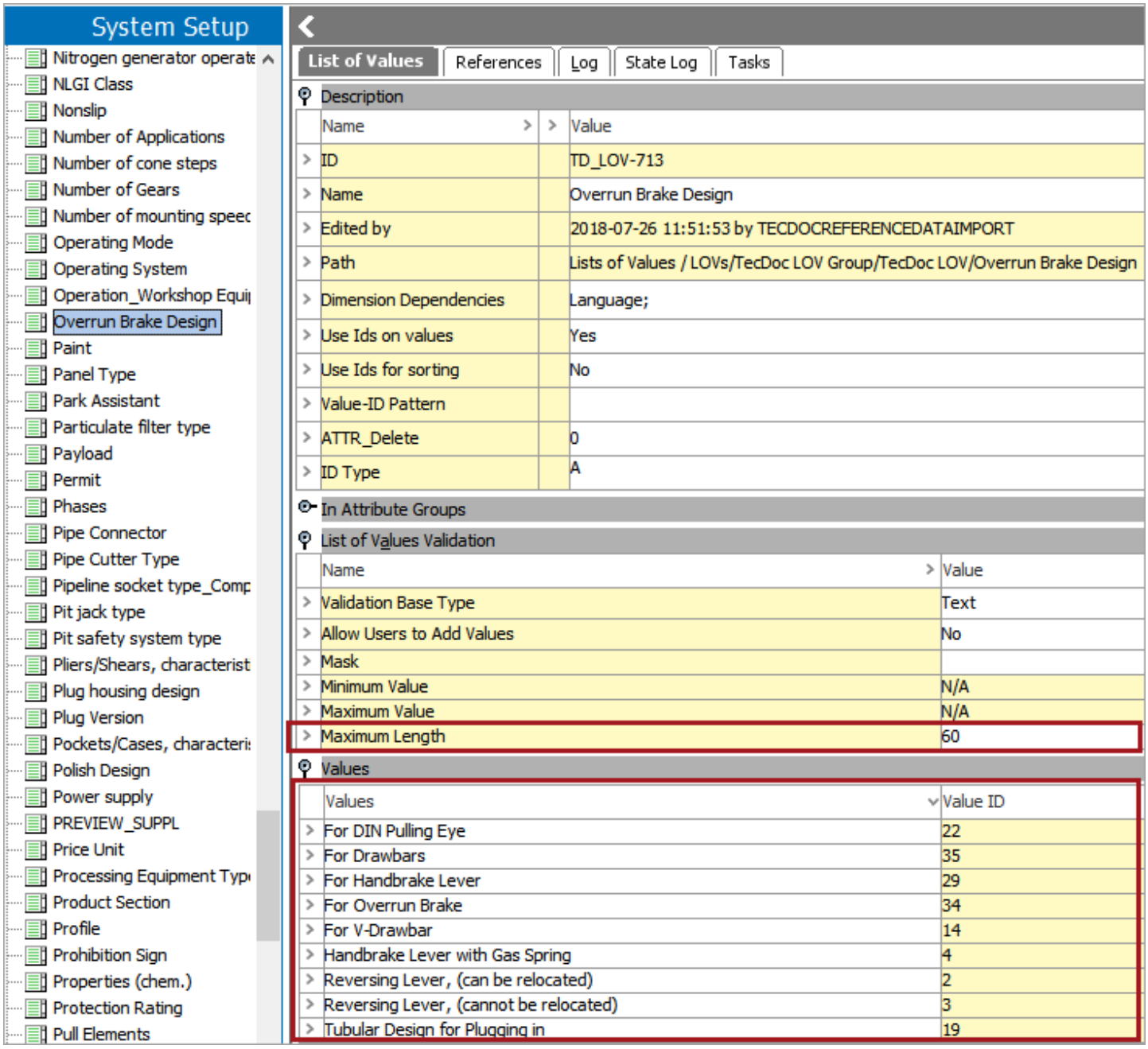
```
999903000034616004With Support Wheel Holder,
pointing rearwards & also crossways    0
```

## 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.



## 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, refer to the **Designer Access** topic within the **Web UI Getting Started** section of **Web User Interfaces** in **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, a Node Details screen must be created. For information about configuring the Node Details Screen, refer to the **Node Details Screen** topic within the **Web User Interfaces** section of **STEP Online Help**.

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 Homepage Widget, including additional information about working with these widgets, refer to File Loading Widget topic, and Status Selector Homepage Widget topic, both within the **Web User Interfaces** section of the **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, refer to 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.

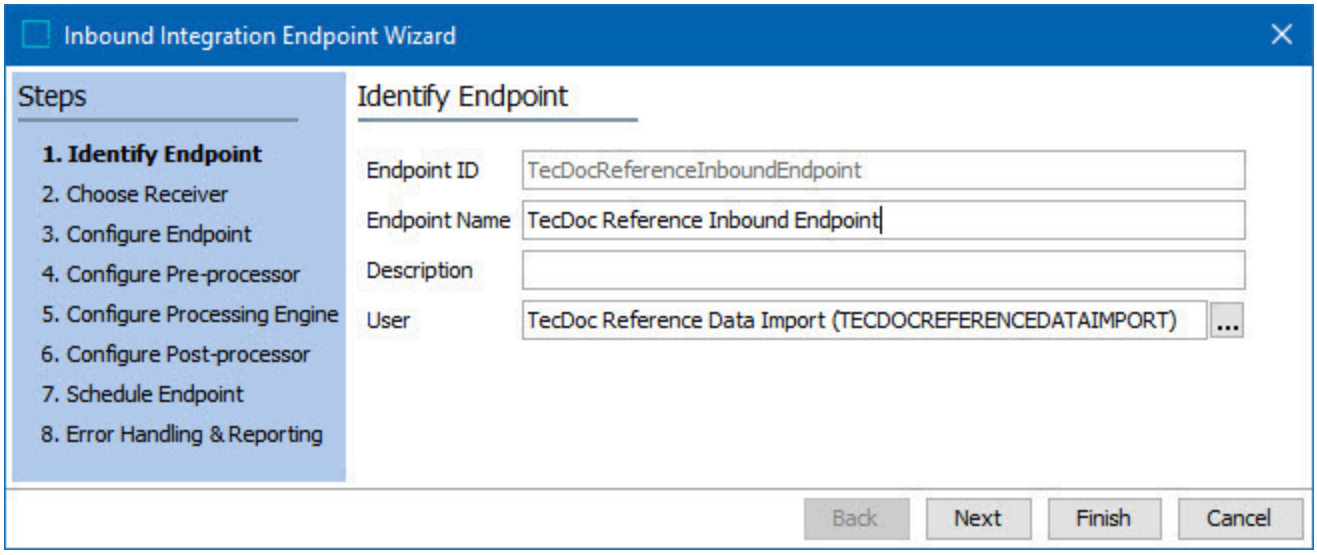
## Prerequisites

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, refer to the **Inbound Integration Endpoints** topic within the **Data Exchange** section of **STEP Online Help**.

## Configuration Steps

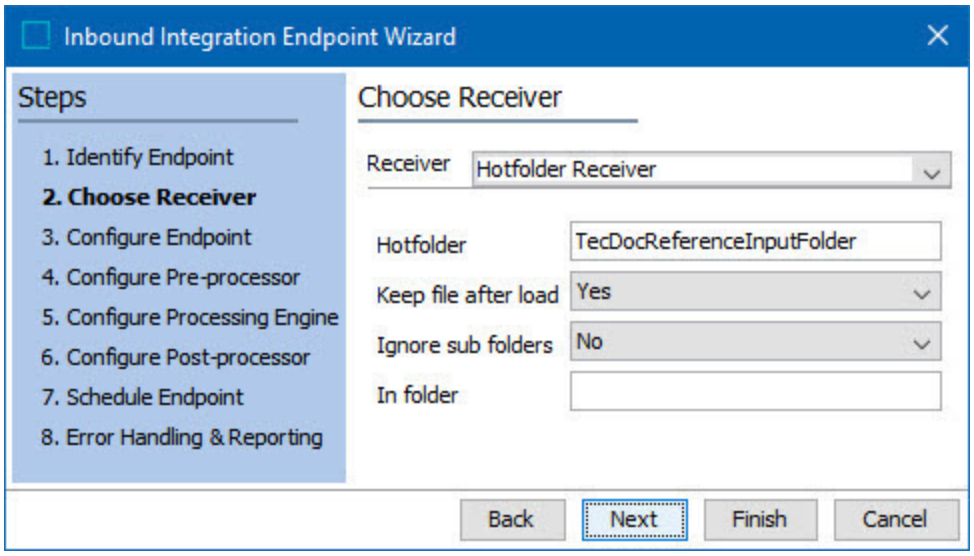
1. Go to System Setup, select and then right-click the **Inbound Integrations Endpoints** setup group, and click **Create Inbound Integration Endpoint**.
2. Once the Inbound Integration Endpoint Wizard displays, populate each parameter with values that best identify the IIEP. By default, all parameters 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.



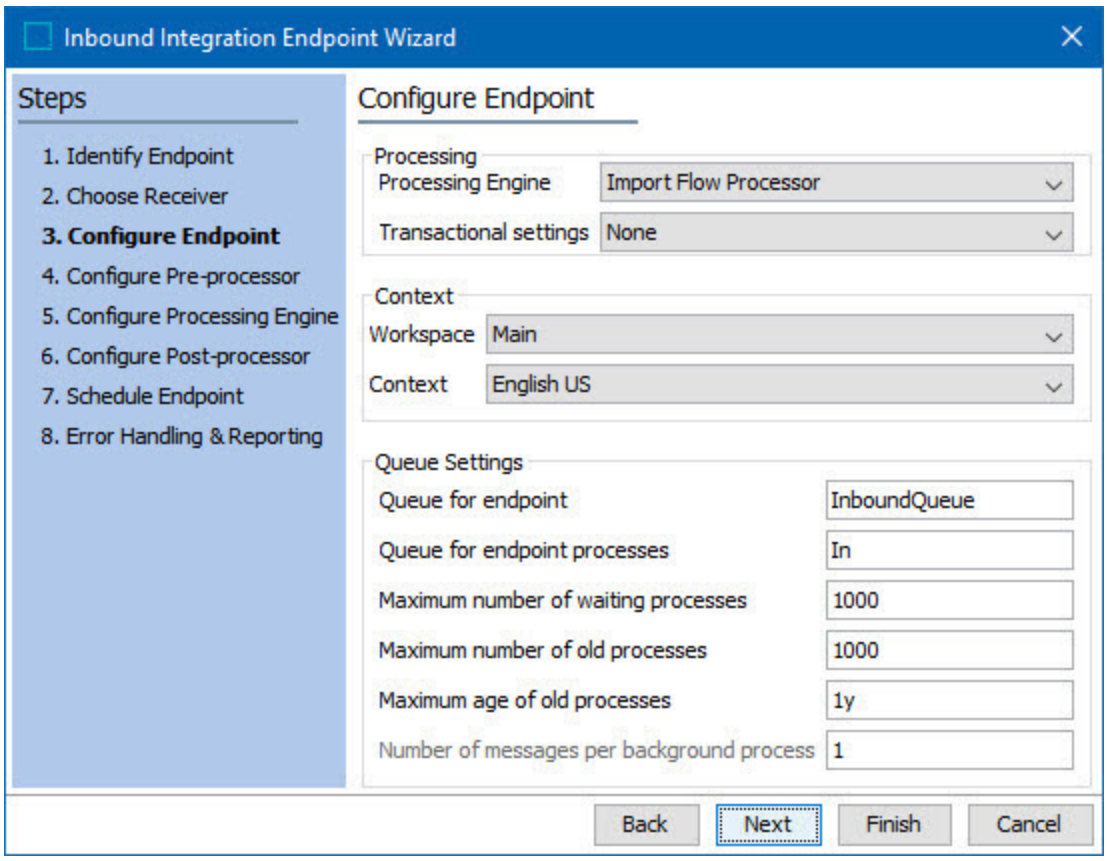
For more information about the parameters, refer to the **IIEP - Identify Endpoint** topic within the **Inbound Integration Endpoint** section of **STEP Online Help**.

3. 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.




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

4. 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

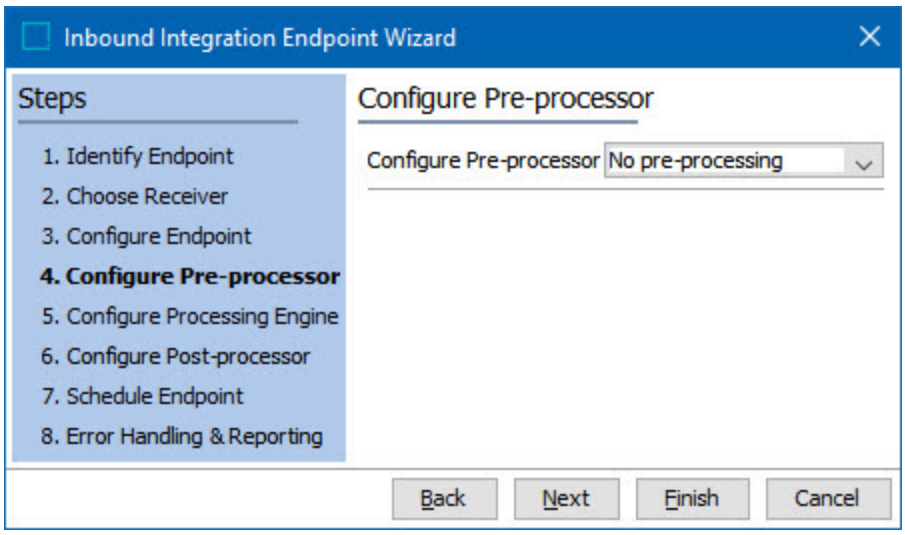



**Important:** The Import Flow Processor is only responsible for picking up files from the configured hotfolder, creating an Entity controller object, and updating the Entity with the Import Flow State BGP's and Status. The Import Flow Processor works with the Background Process Service for each workflow 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.

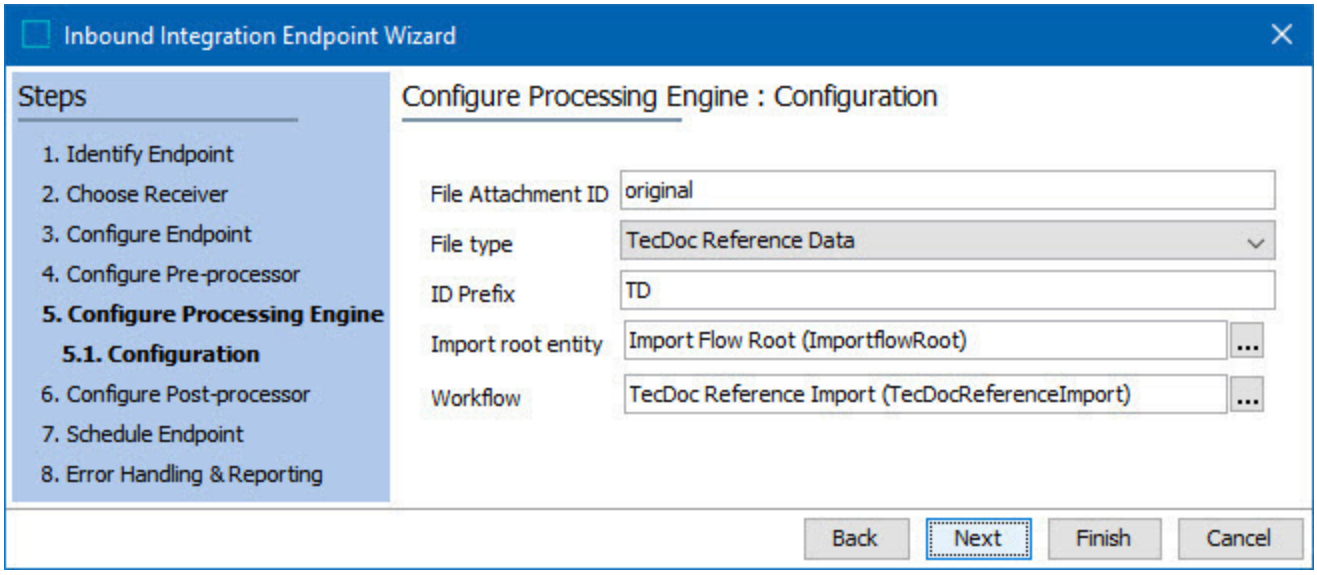
  For more information about the parameters, refer to the **IIEP - Configure Endpoint** topic within the **Inbound Integration Endpoint** section of **STEP Online Help**.

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




For more information about the parameter, refer to the **IIEP - Configure Pre-processor** topic within the **Inbound Integration Endpoint** section of **STEP Online Help**.

6. 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.



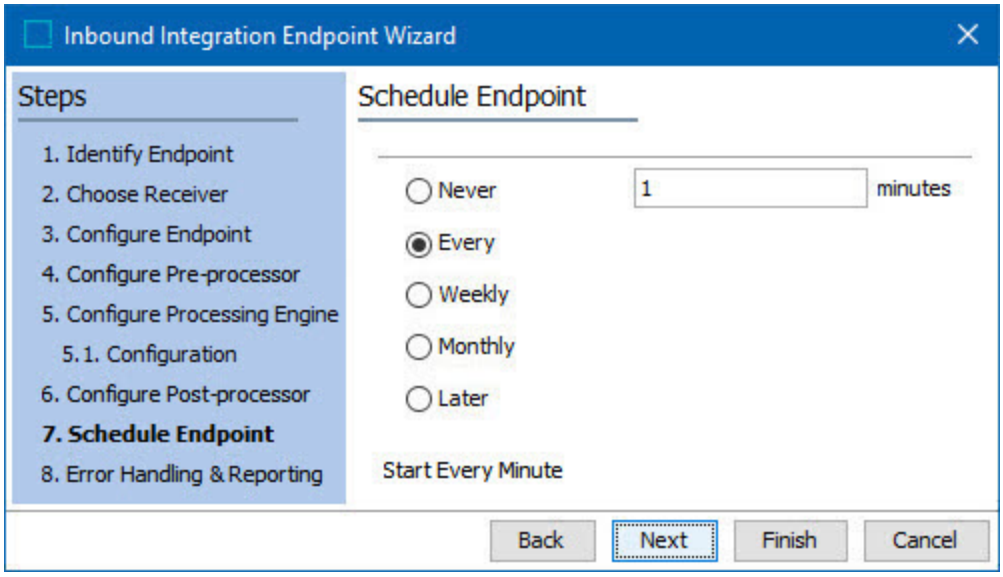

The prior Configure Endpoint step determines the options available for the Configure Processing Engine step.

- 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.

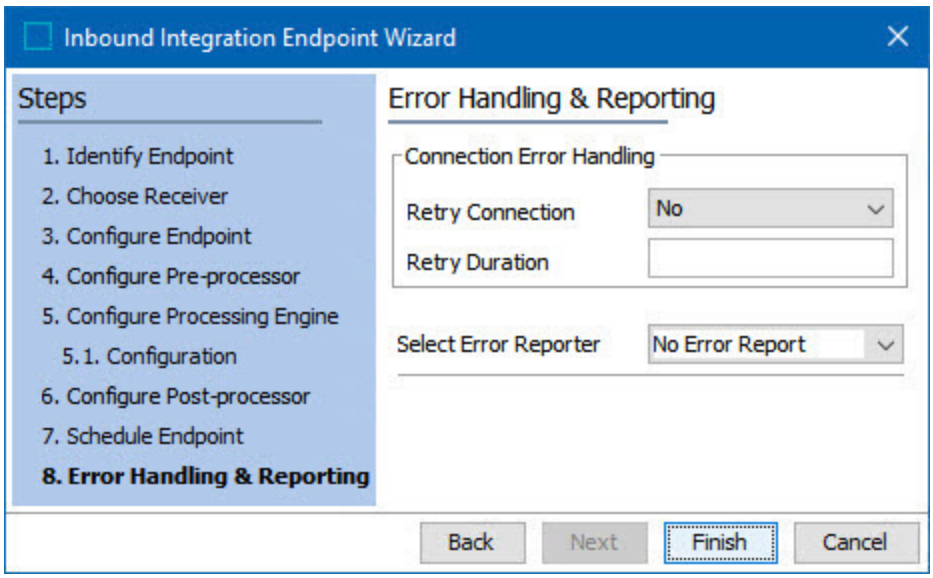
7. Click the **Next** button, and the Schedule Endpoint parameters will display. By default, 'Never' is selected. Optionally, update the values to those shown below.




> **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, refer to the **IIEP - Schedule Endpoint** topic within the **Inbound Integration Endpoint** section of **STEP Online Help**.

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




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

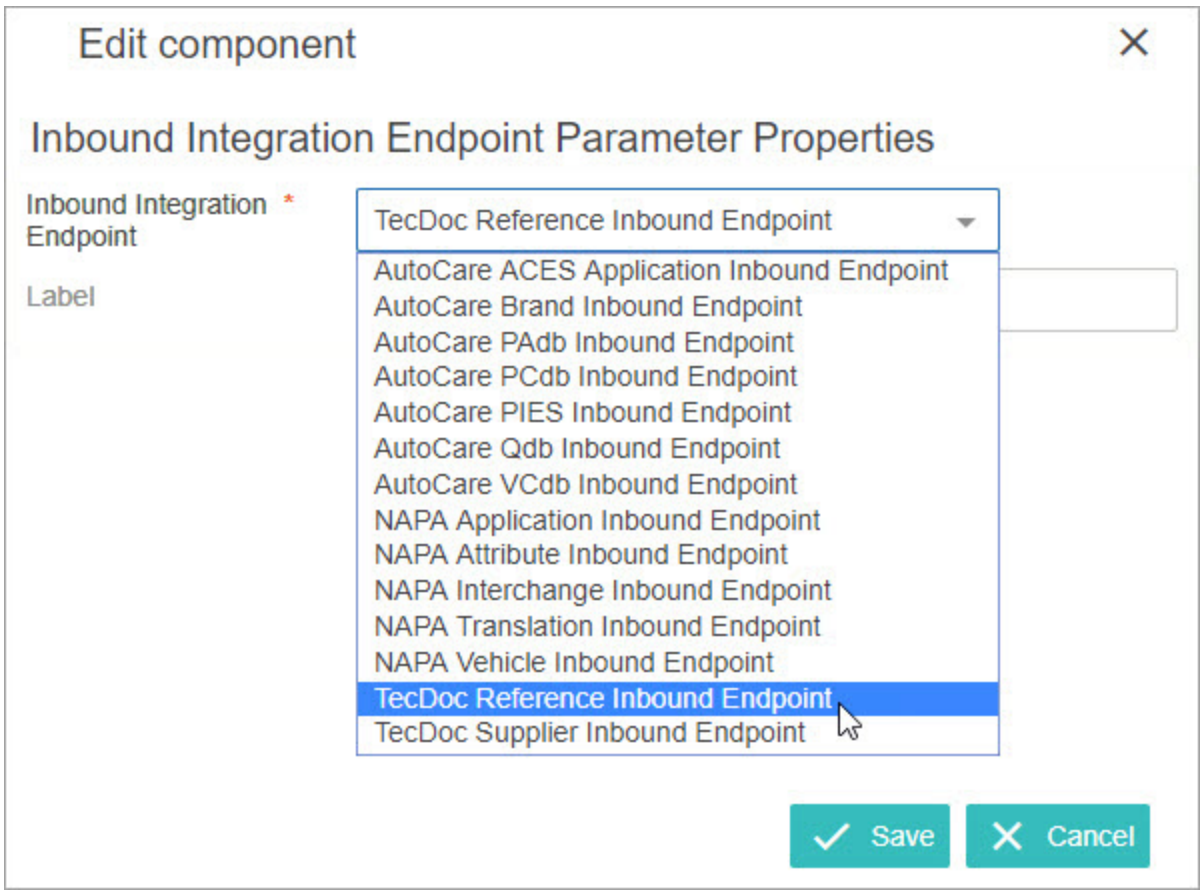
## Configuration

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, refer to 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).

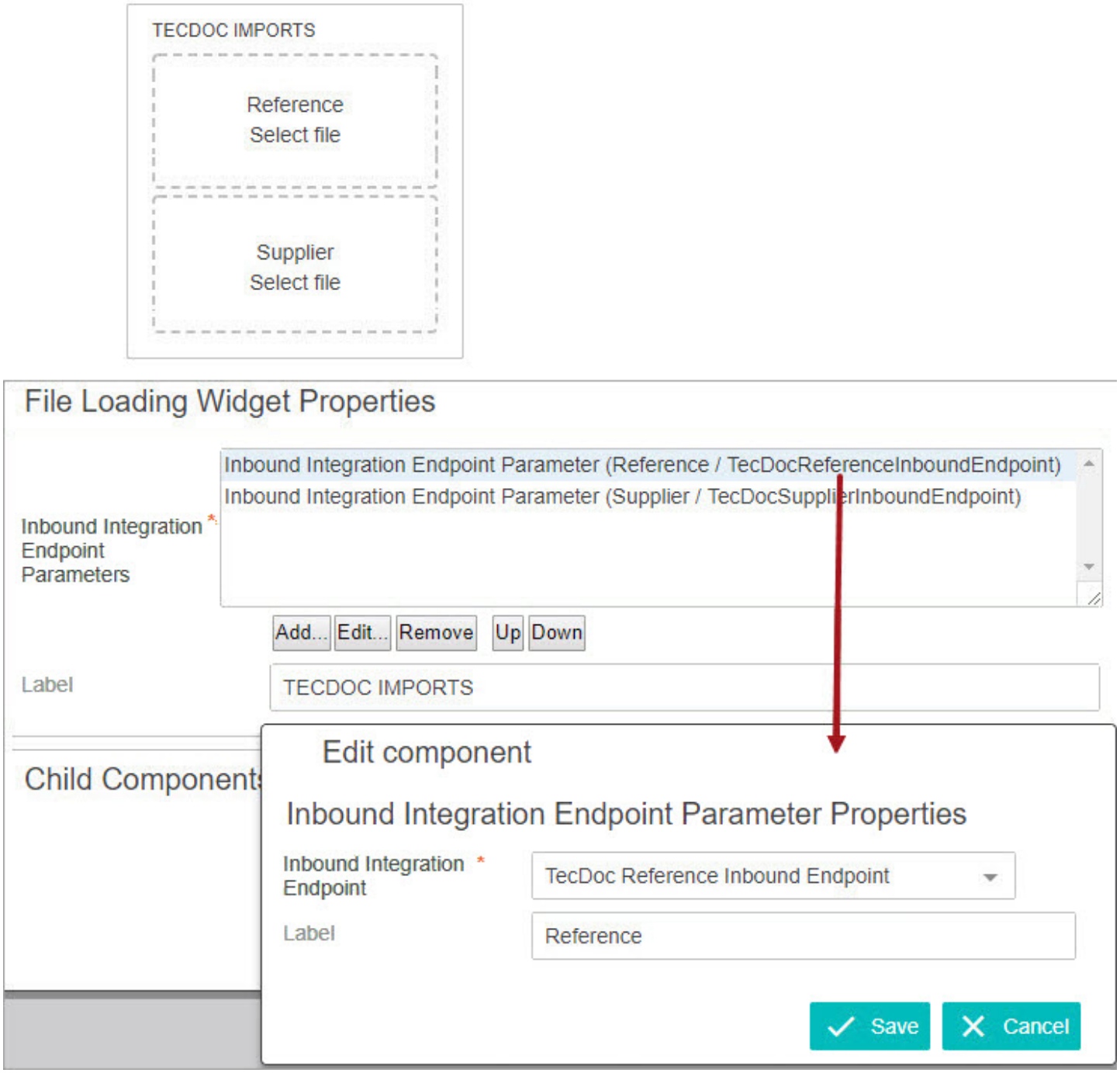



**Note:** If the desired IIEP does not display in the dropdown, then it can be created using the steps described in the **Configuring an IIEP for TecDoc Supplier Data Imports** topic.

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.




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

For general information about the File Loading Widget and Status Selector Homepage Widget, including additional information about working with these widgets, refer to File Loading Widget topic, and Status Selector Homepage Widget topic, both within the **Web User Interfaces** section of the **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, a Node Details screen must be created. For information about configuring the Node Details Screen, refer to the **Node Details Screen** topic within the **Web User Interfaces** section of **STEP Online Help**.

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, refer to 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 Homepage 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.

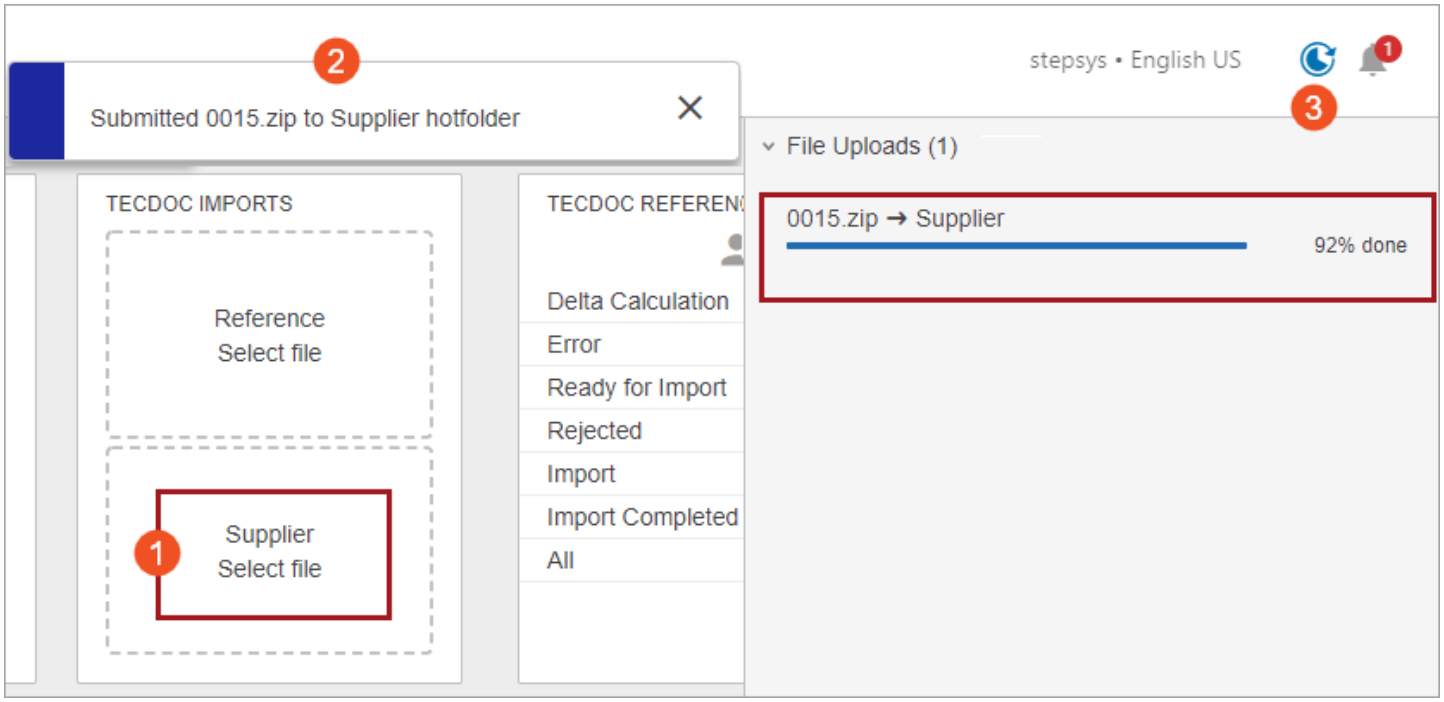
#### Using the TecDoc Supplier Data Importer in Web UI

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, refer to 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 'Recent background processes' side panel.

   In the example below, the 'Recent background processes' side panel is expanded and the file, '0015.zip' is 92% processed.




For more information on using the 'Recent background processes' side panel, refer to the **Recent Background Processes Side Panel** 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.

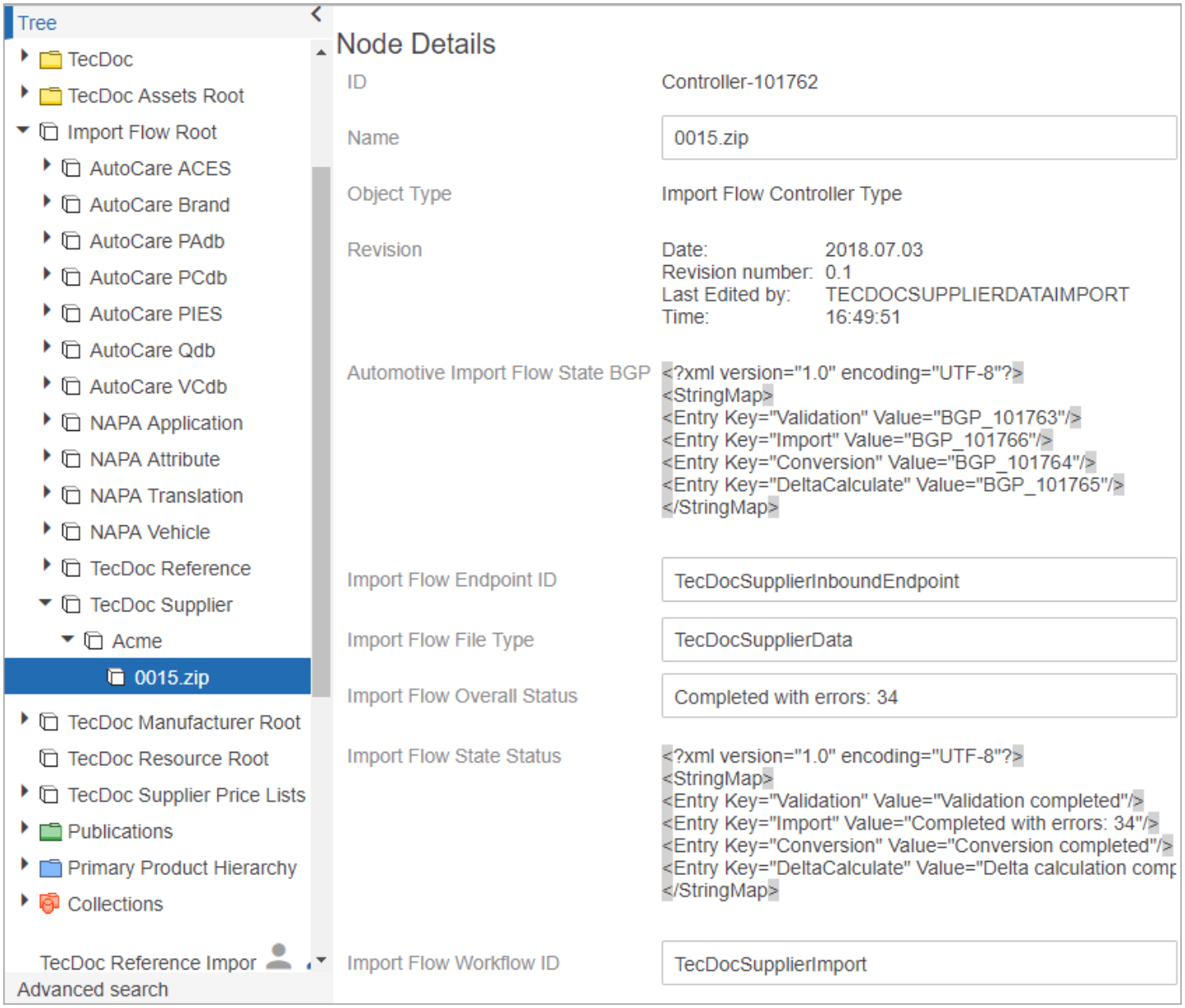
**Important:** It is critical to understand that it is only the Controller Entity that is in the workflow - the objects being acted on (created / updated / deleted) via information supplied in the import file are not in the workflow. Therefore, running standard business actions acting on current object 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, refer to the **Import State** topic within the **Default Workflow States and Functions** section.

4. If a Node 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 Node Details screen has not been configured. For information about configuring the Node Details Screen, refer to the **Node Details Screen** topic within the **Web User Interfaces** section of **STEP Online Help**.

   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.

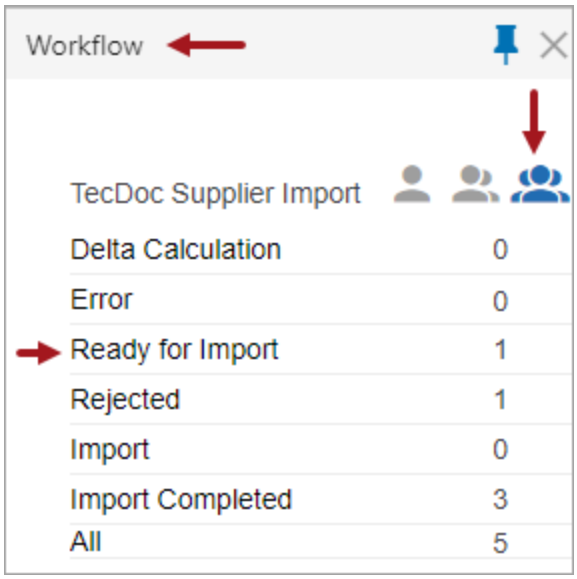



- **Name:** STEP Name of the Controller entity. This is generated from the uploaded file original name.
- **Automotive Import Flow State BGP:** Attribute used to store the IDs of the background processes.
- **Import Flow State Status:** Attribute used to store and display the status of each process (rather 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 Homepage Widget on the Web UI homepage and/or the Status Selector component on the left side panel.

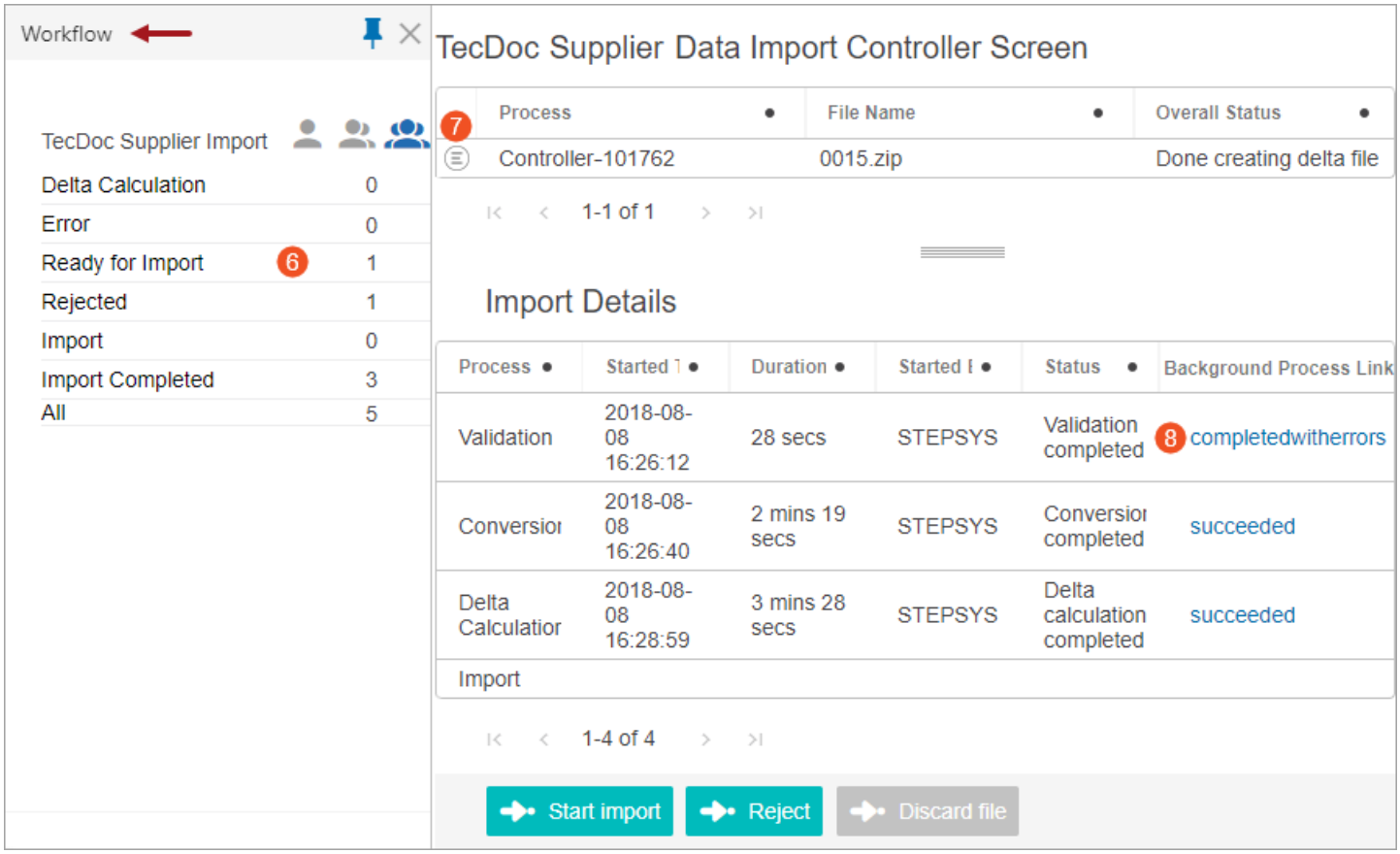
In the example below, the Workflow panel is expanded, and the **Triple user button** is selected so that all items assigned to any user are displayed. Notice that one file is in the 'Ready for Import' state, one file is in the Rejected state, and three files are in the 'Import Completed' state, for a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

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

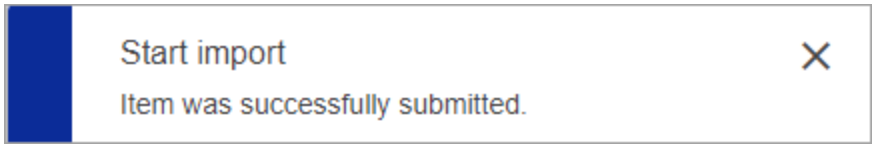



6. Clicking on the **Ready for Import** state within the Status Selector homepage and/or the left side panel widget will navigate the user to the TecDoc Supplier Data Import Controller Screen with only those Controllers in the 'Ready for Import' state displayed.
7. Clicking the Controller icon displays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

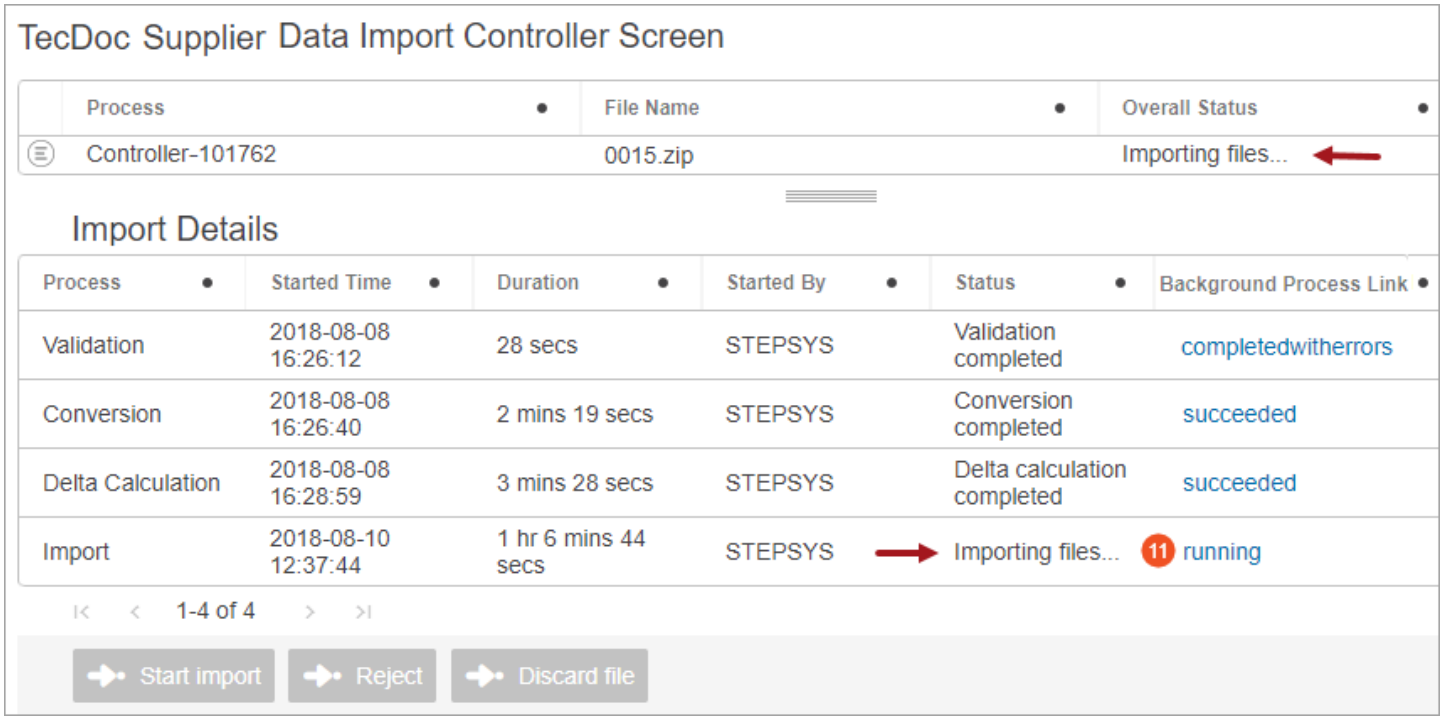



8. Optionally, before starting the import, clicking on the 'completedwitherrors' Background Process Link will display the Background Process Details screen where users can view or download the details of the validation process.
9. Click the **Reject** button to move the Controller to the Rejected state. For more information, refer to 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, refer to 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 Homepage Widget on the Web UI homepage and/or the Status Selector component in the left side panel. More detailed progress can be monitored in the import process Status column that is available within the Import Controller Screen.

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.'




11. 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, refer to the **Import Completed State** topic within the **Import Framework** section.

    For more information on automotive workflow states, refer to 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, refer to the **Designer Access** topic within the **Web UI Getting Started** section of **Web User Interfaces** in **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, a Node Details screen must be created. For information about configuring the Node Details Screen, refer to the **Node Details Screen** topic within the **Web User Interfaces** section of **STEP Online Help**.

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 Homepage Widget, including additional information about working with these widgets, refer to File Loading Widget topic, and Status Selector Homepage Widget topic, both within the **Web User Interfaces** section of the **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, refer to 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.

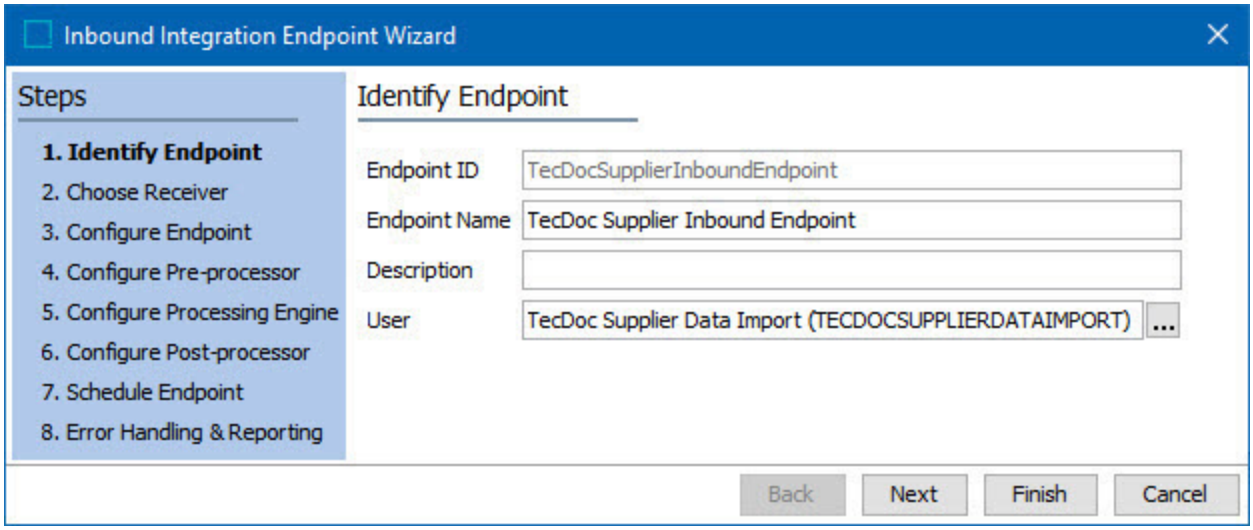
## Prerequisites

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, refer to the **Inbound Integration Endpoints** topic within the **Data Exchange** section of **STEP Online Help**.

## Configuration Steps

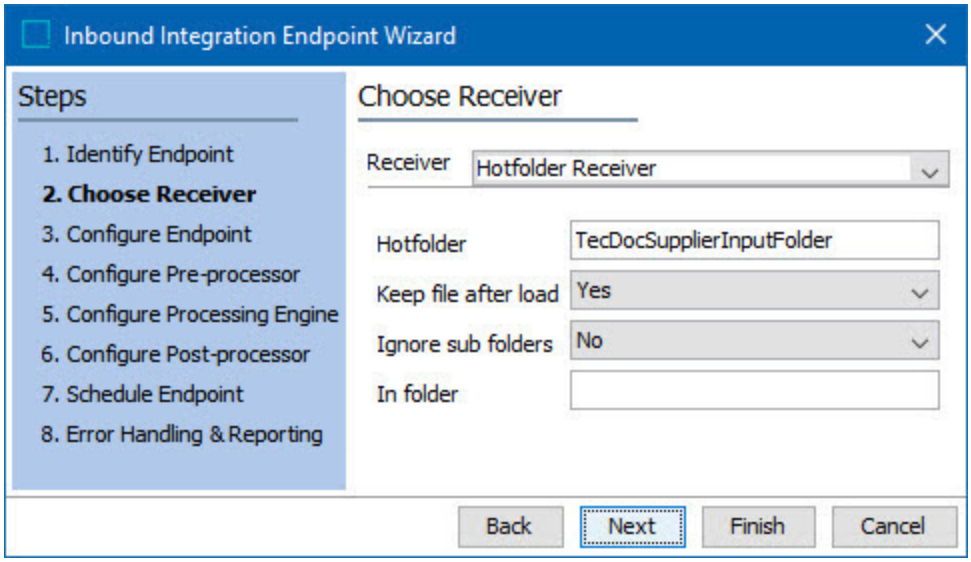
1. Go to System Setup, select and then right-click the **Inbound Integrations Endpoints** setup group, and click **Create Inbound Integration Endpoint**.
2. Once the Inbound Integration Endpoint Wizard displays, populate each parameter with values that best identify the IIEP. By default, all parameters 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.



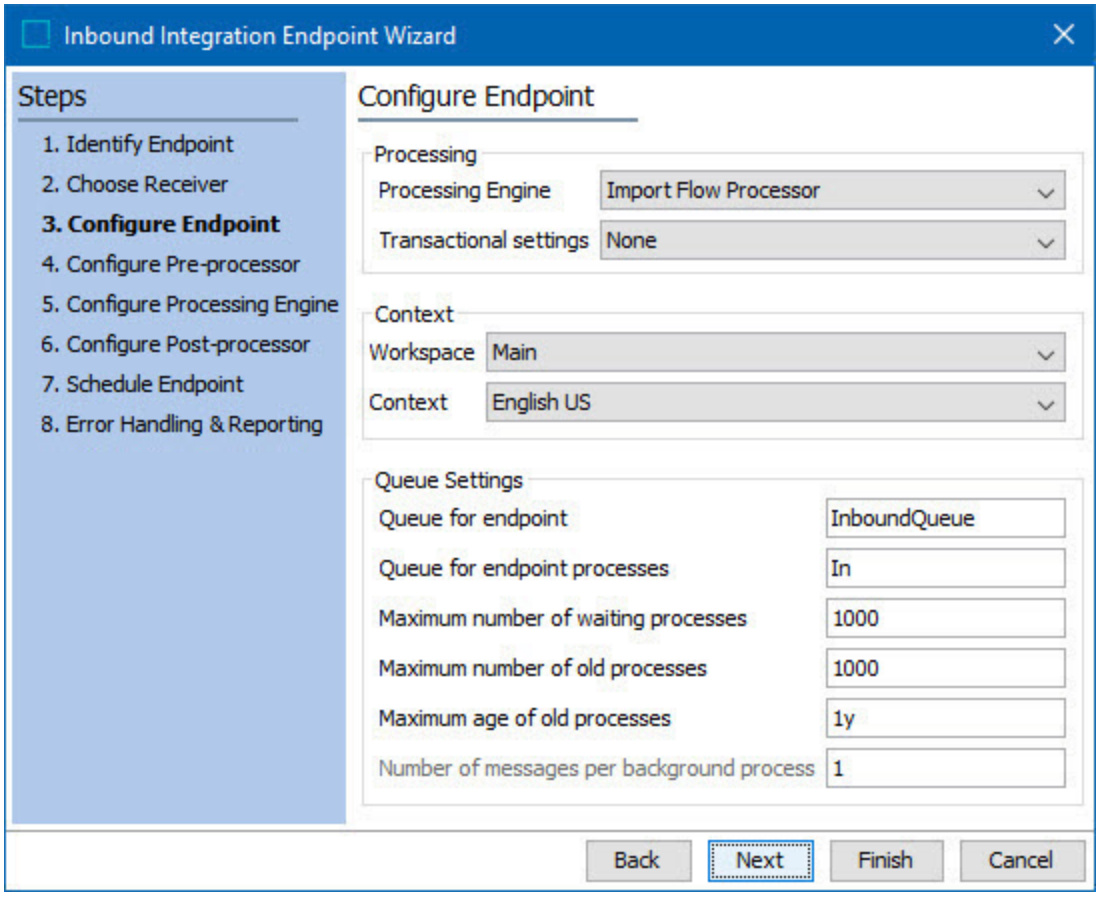

   For more information about the parameters, refer to the **IIEP - Identify Endpoint** topic within the **Inbound Integration Endpoint** section of **STEP Online Help**.

3. 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.




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

4. 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

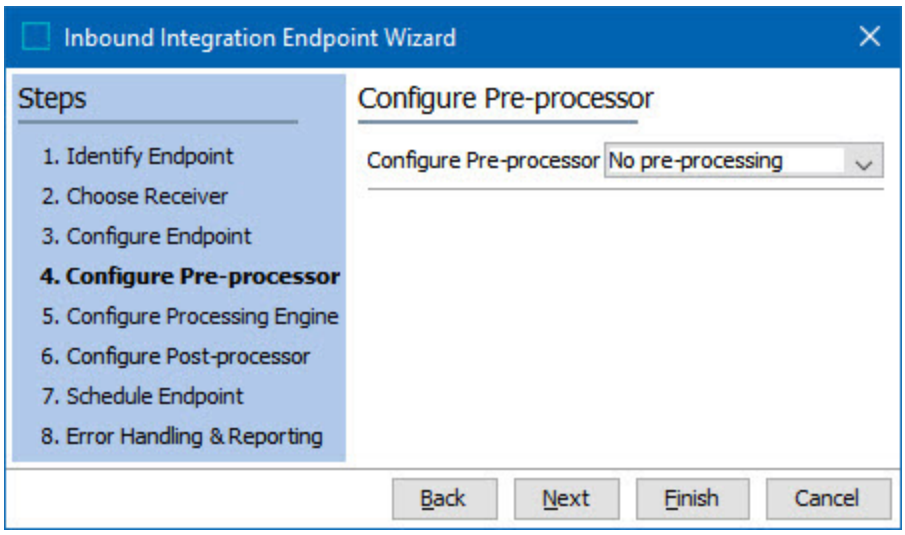



**Important:** The Import Flow Processor is only responsible for picking up files from the configured hotfolder, creating an Entity controller object, and updating the Entity with the Import Flow State BGPs and Status. The Import Flow Processor works with the Background Process Service for each workflow 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.

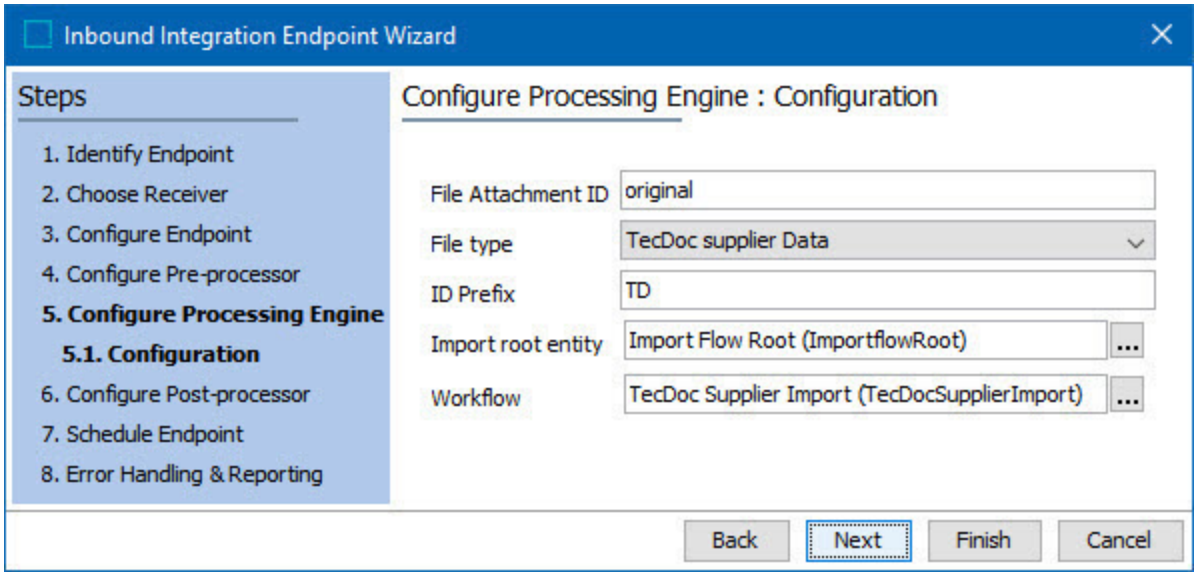
For more information about the parameters, refer to the **IIEP - Configure Endpoint** topic within the **Inbound Integration Endpoint** section of **STEP Online Help**.

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




For more information about the parameter, refer to the **IIEP - Configure Pre-processor** topic within the **Inbound Integration Endpoint** section of **STEP Online Help**.

6. 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.



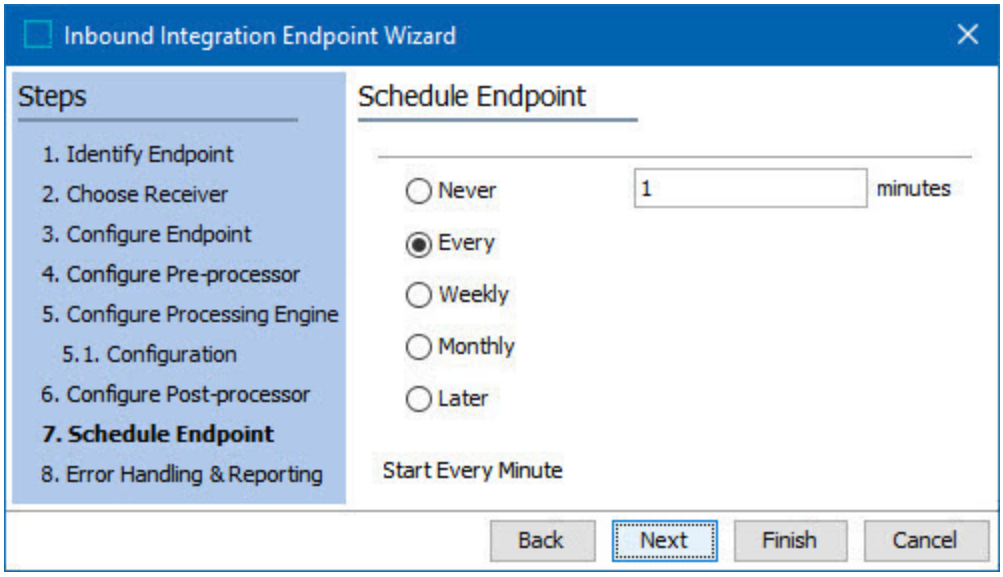

The prior Configure Endpoint step determines the options available for the Configure Processing Engine step.

- 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.

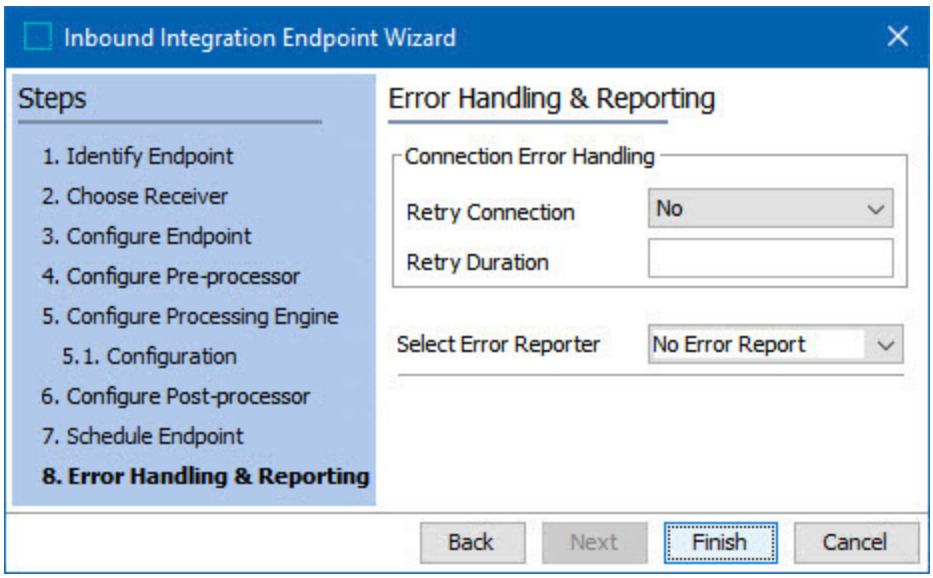
7. Click the **Next** button, and the Schedule Endpoint parameters will display. By default, 'Never' is selected. Optionally, update the values to those shown below.




> **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, refer to the **IIEP - Schedule Endpoint** topic within the **Inbound Integration Endpoint** section of **STEP Online Help**.

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




For more information about the parameter, refer to the **IIEP - Error Handling & Reporting** 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, refer to 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, refer to 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, refer to 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.

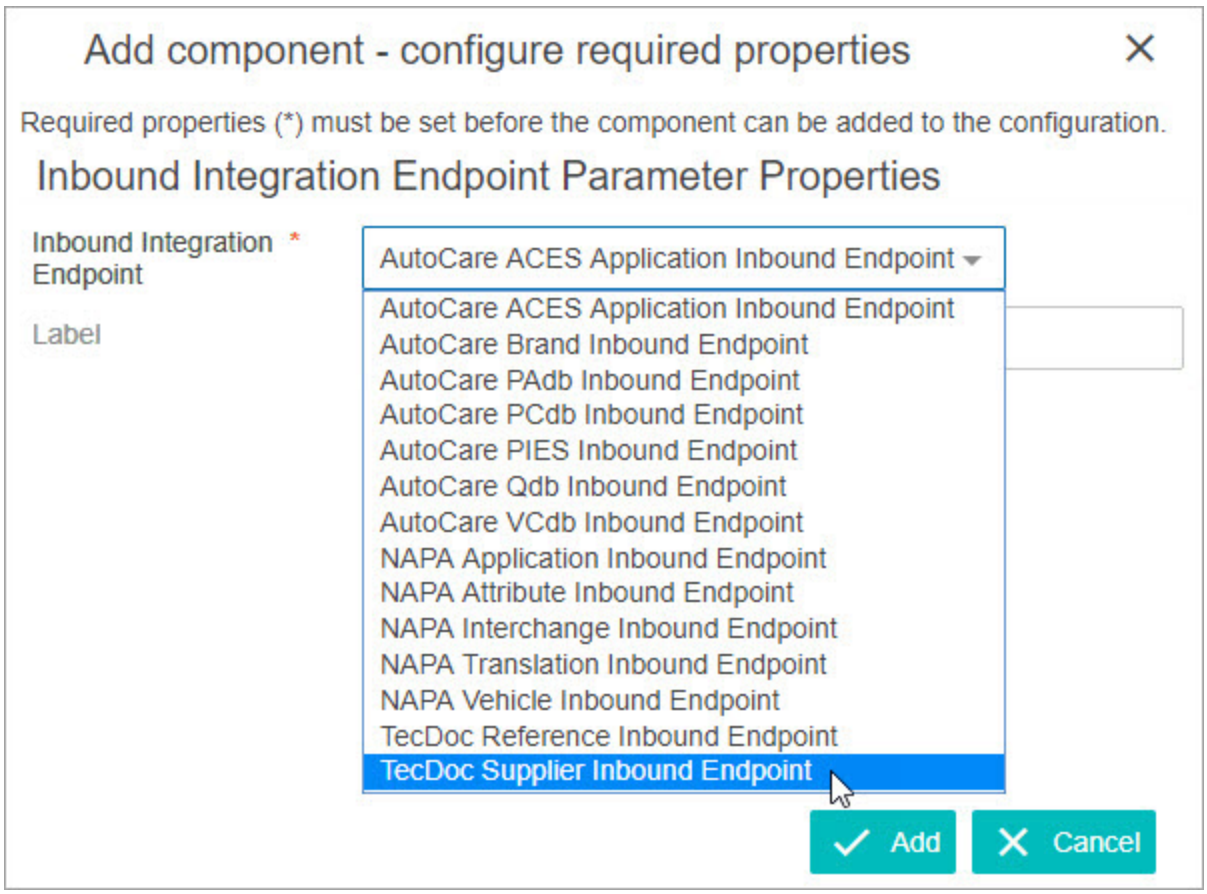
## Configuration

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, refer to 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).

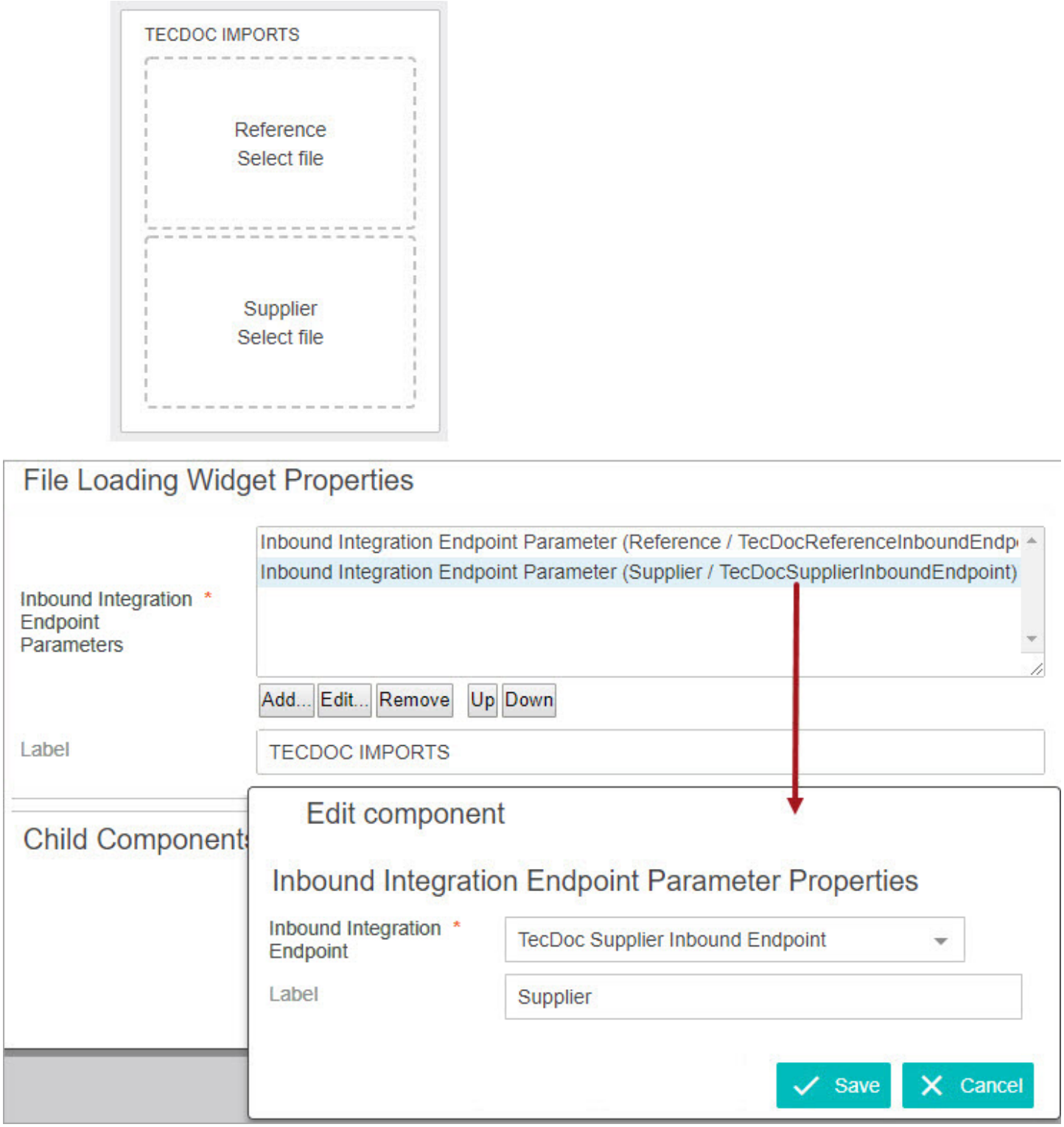



**Note:** If the desired IIEP does not display in the dropdown, then it can be created using the steps described in the **Configuring an IIEP for Supplier Data Imports** topic.

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.




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

# Reference Delegation Screen

The Reference Delegation Screen is a Web UI screen type that, when configured with a reference type and a screen ID, will follow the reference from the source object to its target and display the information of the target object. It allows you to take any active selection, follow a reference on that selection, and display any screen configured for the target of the reference (noting that the selected screen may show data from the target itself, data from the children of the target (Automotive specific using the Application Editor Screen), data from objects referenced by the target, etc).

One common use case of the Reference Delegation Screen within the Automotive solution is to view the application records of any other standard model from the OWN model. There are no explicit limits on how the screen can be used, but it was created specifically to satisfy two use cases:

1. View applications belonging to a standard's part (example, AutoCare) while the OWN part is selected.
2. View product data on a standard's part while the OWN part is selected.

For example, the screen can be configured to display applications from a PIES Item when an OWN part is selected. To do this, the reference parameter must specify a reference that has the OWN part as a source and the PIES Item as a target, and a screen must be selected that displays applications for a part. The 'AutoCare Application Editor Screen (Parts)' exists via easy setup creation and displays applications when a part is selected, so this screen would be a good candidate. Once configured properly, the Reference Delegation Screen can then be added to any existing screen mapped to display data from the OWN part. An example of this configuration would be a Node Details screen using Tab Control in which the Sub Screen Tab Page configuration specifies the Reference Delegation Screen.

The above is just an example, and as stated, there are a variety of use cases for which this screen can be applied. The only limitation to its use is that **the selected reference must be of a reference type that allows single references only**. If a selected reference is of a reference type that allows multiple references, an error is displayed.

## Prerequisites

It is expected that anyone configuring the Reference Delegation Screen is familiar with the Web UI Designer, as basic concepts for working with the designer are not covered in this section. In addition, the user must have appropriate privileges to access the designer. Additional information can be found in the **Designer Access** section of the **Web User Interfaces** documentation.

The target screen that is used to display the target object information must already be created.

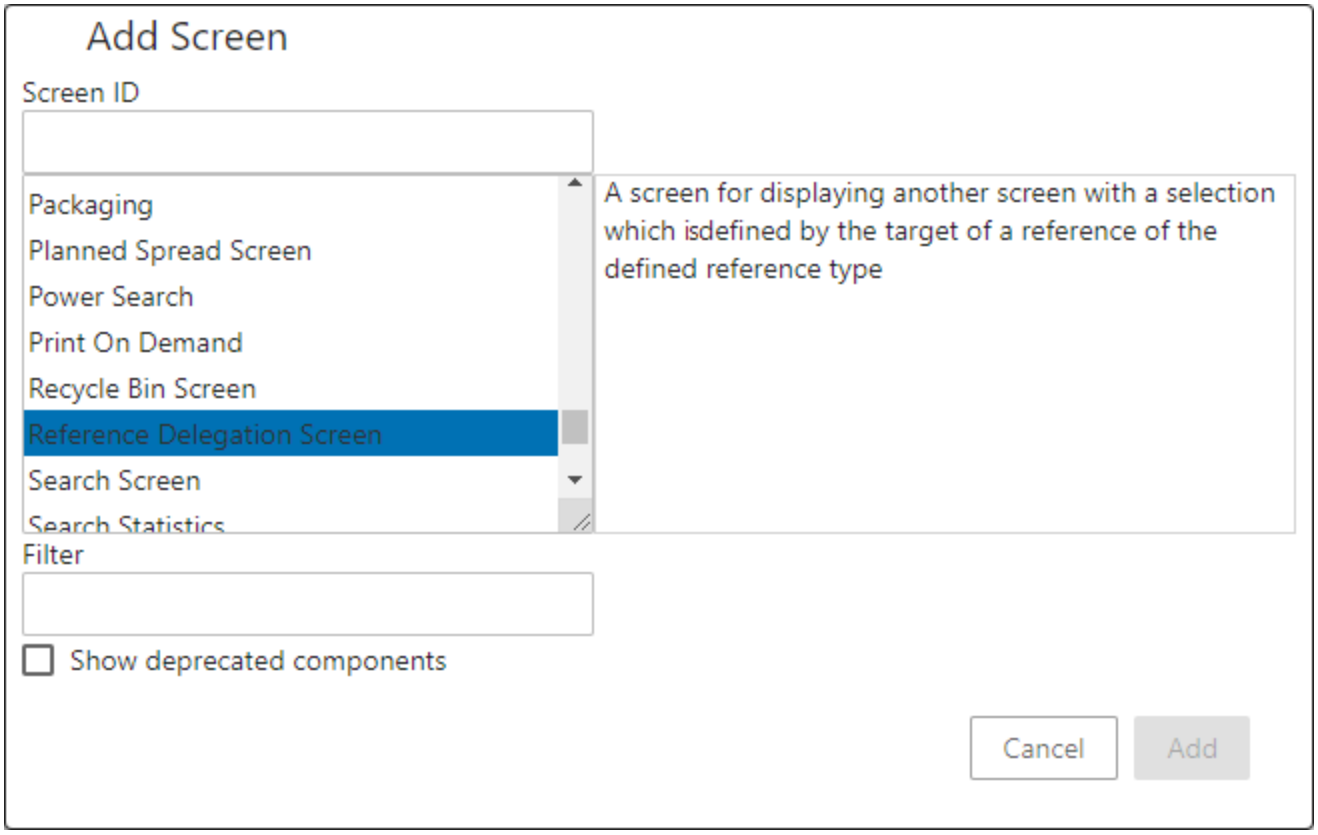
A reference must already be established between the source and the target object using the reference type that will be configured in the Reference Delegation Screen.

## Configuring Reference Delegation Screen

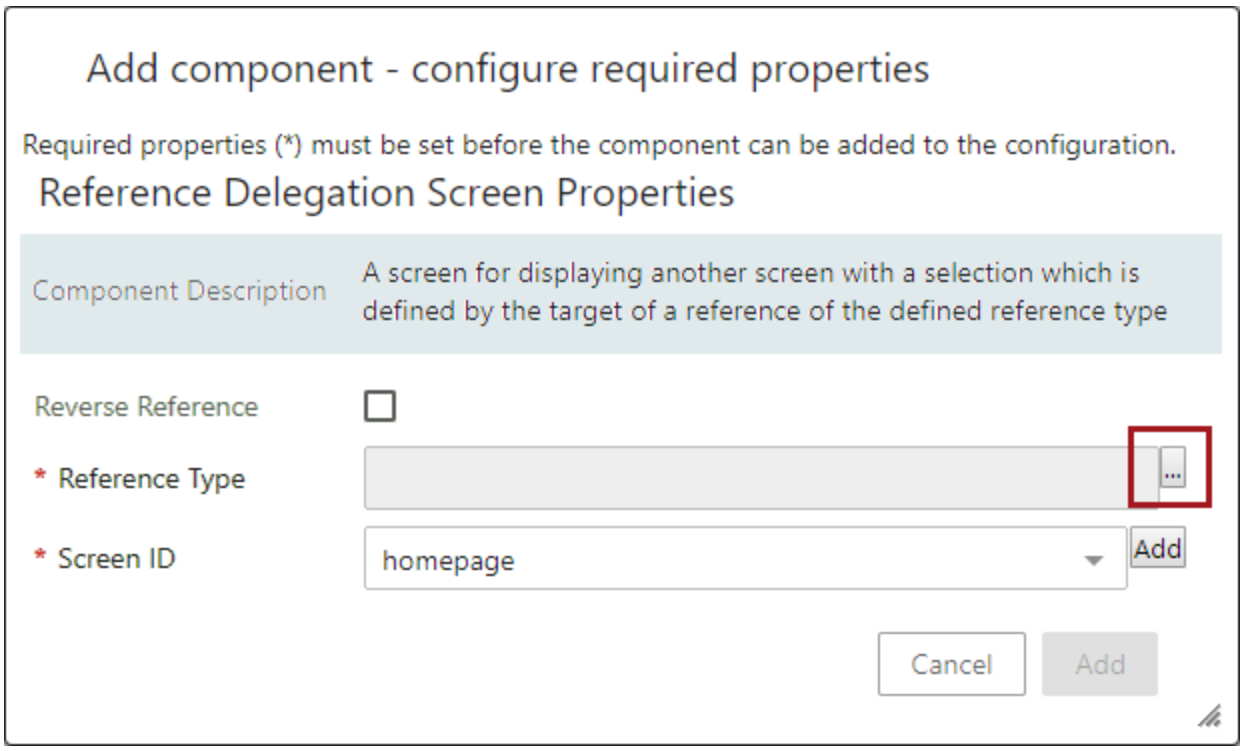
To create and configure a Reference Delegation Screen, users must first determine the reference type that links the source and the target object, and the screen that displays the target object details. Once this has been determined, users may begin the configuration process.

1. Create a new ‘Reference Delegation Screen.’ Steps for creating a new screen are outlined in the **Creating a New Screen** section described within the **Design Mode Basics** topic in the **Web User Interfaces** documentation of **STEP Online Help**.

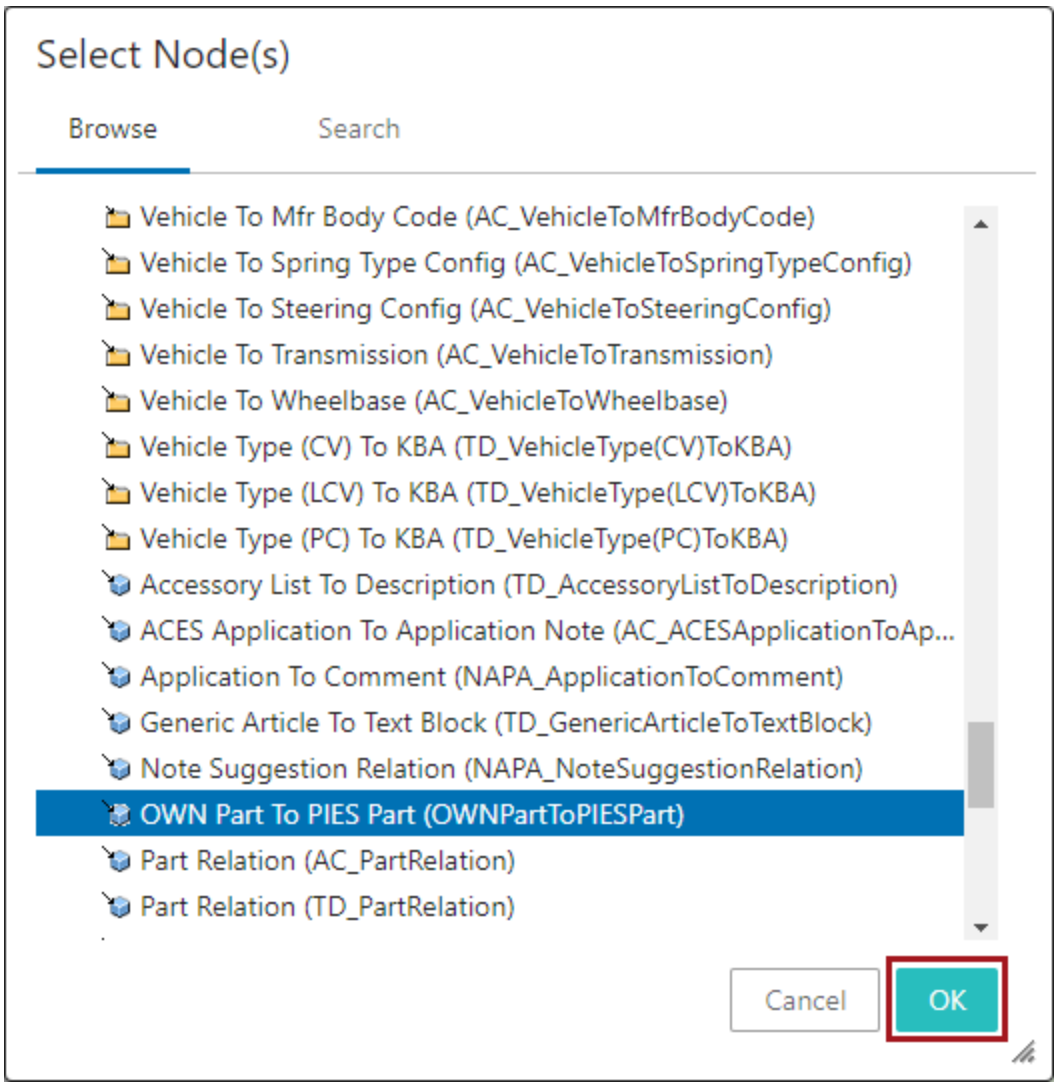
   Select 'Reference Delegation Screen' as shown below:



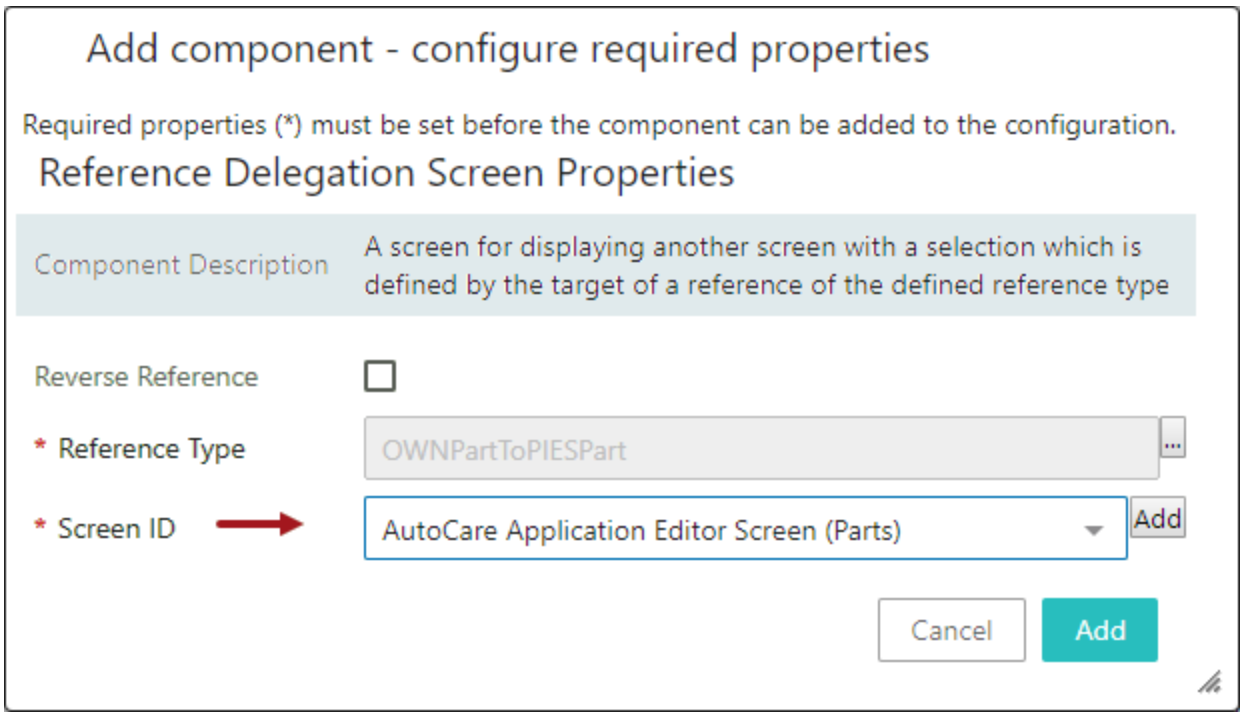

2. After creating the new ‘Reference Delegation Screen', users must determine which reference type must be used to follow from the source to the target object. Click the ellipsis button (...) to bring up a Node Selector dialog.




3. Using either the 'Search' or the 'Browse' option, select the reference type that will be used to follow from the source object to its target. In the example below, reference type ID 'OWNPartToPIESPart' is selected. When the reference type has been selected, click **OK**.



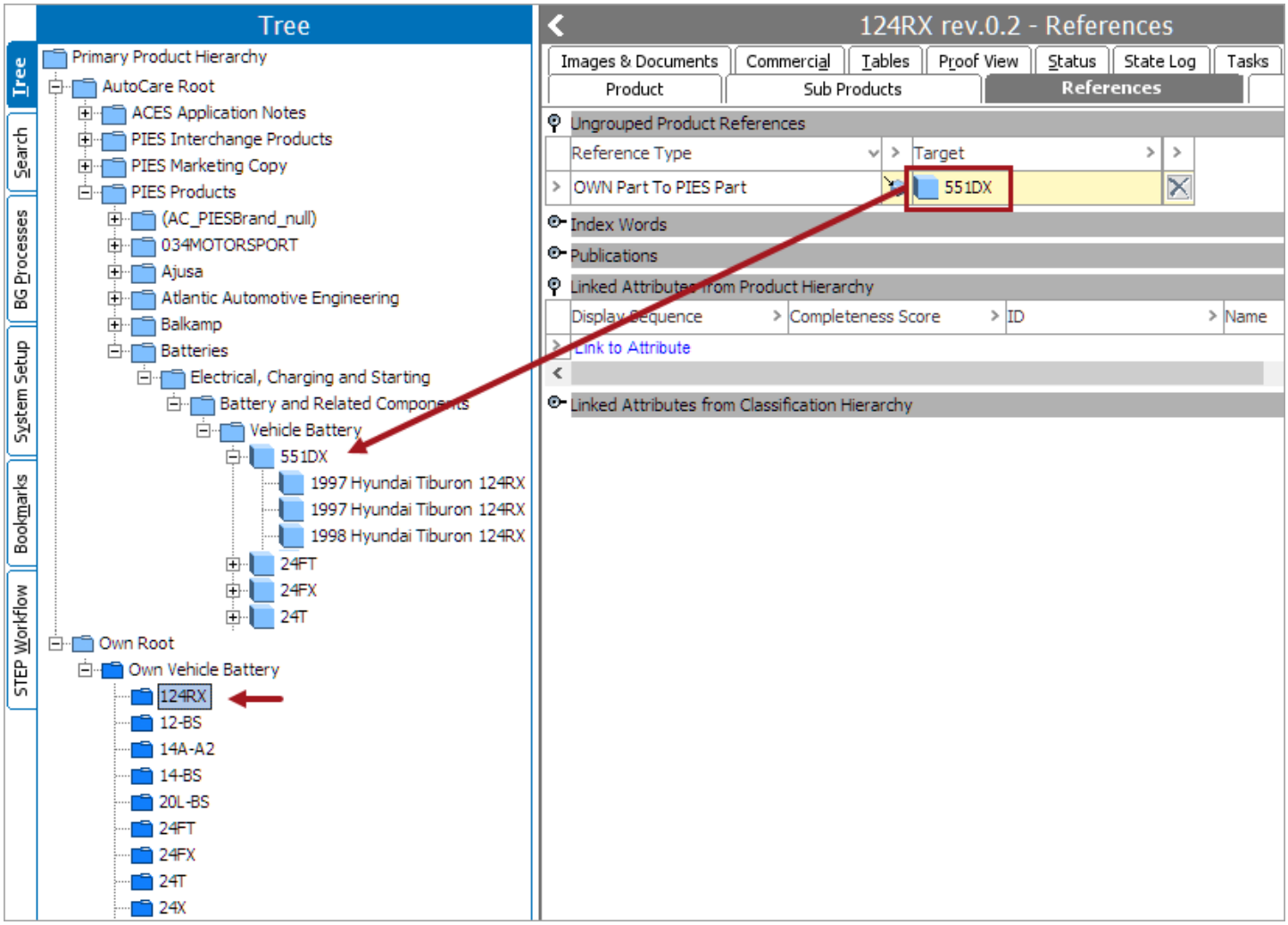

4. In the dropdown next to the Screen ID field, select the required screen from the available list. This determines the screen that will be used to display the target object information. In the example below, we have selected the screen entitled, 'AutoCare Application Editor Screen (Parts).'




5. Check the Reverse Reference parameter only if required. Selecting this field will work the other way of the reference type defined in the Reference Type parameter above by using the 'Referenced By' tab to find the target. Selecting this parameter will follow the reference on the selected object, and display any screen configured for the children objects of the target referenced by the source object. Click **Add** to finish creating the screen.
6. Once configured, set the Reference Delegation Screen as a Sub Screen Tab Page. Steps for setting up as a Sub Screen Tab Page are outlined in the **Tab Pages** topic within in the **Web User Interfaces** documentation of **STEP Online Help**.
7. Save and close the designer.

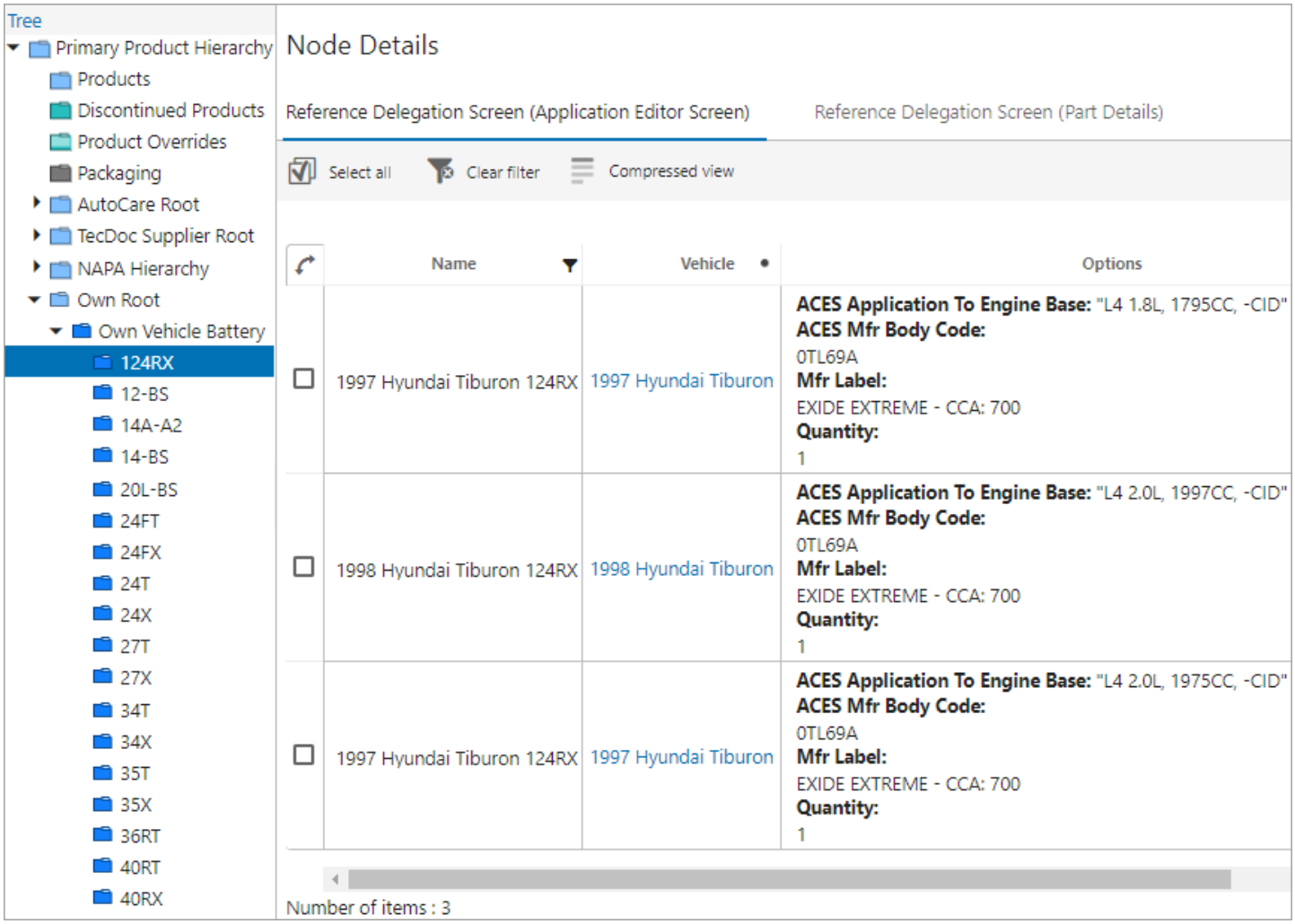
## Use Case

Shown below is an example where the OWN part '124RX' is referenced to a PIES Item '551DX' (AutoCare standard) through the 'OWNPartToPIESPart' reference type. The target object PIES Item has few application records as its child objects.



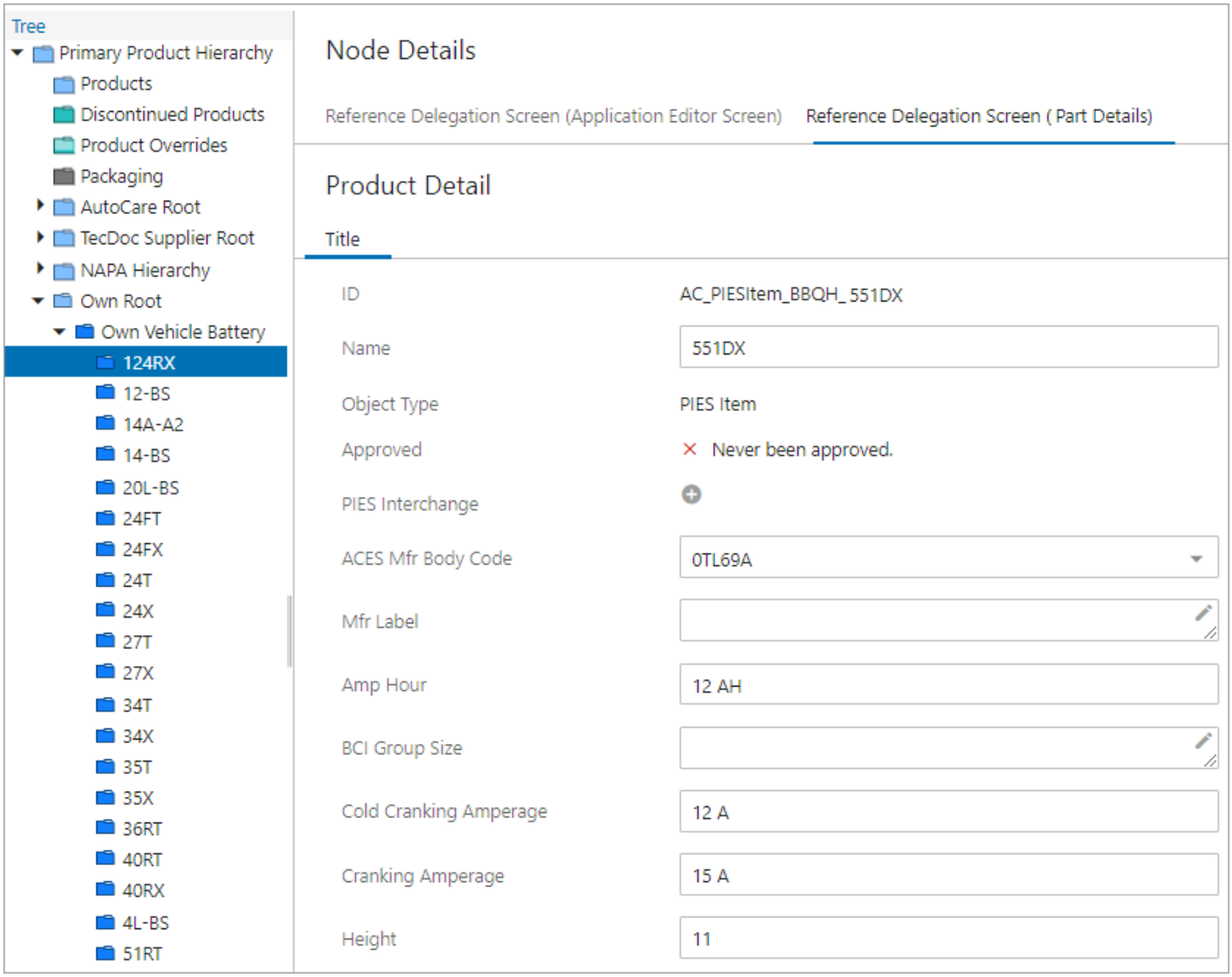

## Example One

When the Reference Delegation Screen is configured as a Sub Screen Tab Page within the source Node Details screen to display the Application Editor Screen of the target object, the application records belonging to the PIES Item are displayed when the OWN part is selected, as shown below.




## Example Two

When the Reference Delegation Screen is configured as a Sub Screen Tab Page within the source Node Details screen to display the product data of the target object, the information available within the Part Details screen belonging to the PIES Item is displayed when the OWN part is selected, as shown below.


Tree
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
24FT
24FX
24T
24X
27T
27X
34T
34X
35T
35X
36RT
40RT
40RX
4L-BS
51RT
Node Details
Reference Delegation Screen (Application Editor Screen)
Reference Delegation Screen ( Part Details)
Product Detail
Title
ID
AC_PIESItem_BBQH_ 551DX
Name
551DX
Object Type
PIES Item
Approved
Never been approved.
PIES Interchange
ACES Mfr Body Code
0TL69A
Mfr Label
Amp Hour
12 AH
BCI Group Size
Cold Cranking Amperage
12 A
Cranking Amperage
15 A
Height
11

# Search by Card Screen

The Search by Card Screen uses the sync between search cards functionality to provide an easy-to-use intelligent search interface. This makes it possible for the user to narrow down the search result and only display the relevant result. This screen can be configured to define the most likely used filter groups for the search in the Web UI. For example, if the user needs to find a vehicle based on the Name, Make, Model, Year, or any attribute value, and the search cards are configured to be available for each of these search criteria in the Search by Card Screen, then users can refine the list of objects based on one or more of the available filters.

The top half of the Search by Card 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

The Search by Card Screen topics in online help include:

- Search by Card Screen Initial Configuration
- Using Search by Card Screen Interface

The topics listed below cover more information on Search by Card configurations, going beyond the basic setups described in the introductory material.

- Using Search by Cards Screen with Data Onboarding Tasks

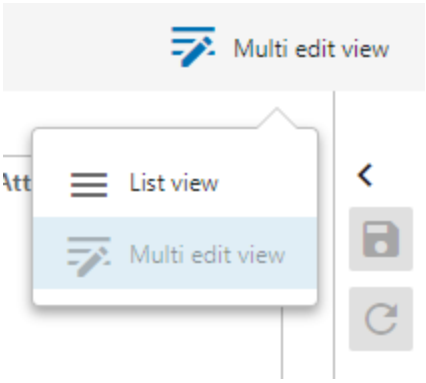
**Note:** The Search by Card Screen is highly configurable via the Web UI Designer, and more than one Search by Card Screen can be created within each Web UI. Therefore, it is expected that a Web UI administrator will configure a Search by Card Screen in a manner that best meets their user's needs. For more information on configuring the Search by Card Screen, refer to the **Search by Card Initial Configuration** topic.

## Search by Card Screen Interface Overview

The top half of the Search by Card 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. The Search by Card Screen is composed of the following:

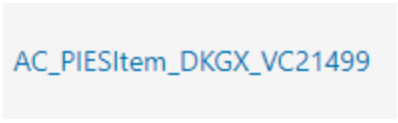
1. **Title:** Users can edit the Title in Design mode with any text string.
2. **Card based search panel:** Use cards to narrow down the search result and only display relevant nodes. Users can configure the cards to use attributes, parent hierarchy, or Name. All of the configured cards are synchronized, and the search criteria can be entered into the cards in any order, and the available search criteria will display in the typeahead dropdown based upon the criteria that have been entered within the other cards.
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, refer to the **Saved Search Tool** topic within the **Application Manager** section of this guide.
4. **Clear All link:** Allows users to clear all search criteria and results that are displayed in the results table. The Clear All link remains disabled until at least one criterion for one of the search cards is populated. When text is typed into a search card, but is not yet selected, clicking the link will also remove the text typed into the search card. Additionally, a criterion can be cleared one at a time by clicking the '**X**' to the right of a value within a search card. Clearing the criteria in either way does not update the search results. Once any unnecessary criteria are removed, the Search button must be used to update the results table.

5. **Search button:** When clicked, the Search button will use the criteria that are provided within the search cards, and then provide results matching the search criteria in the results table. The Search button remains disabled until at least one search card is populated.
6. **Results table toolbar:** Allows users to configure action buttons, such as bulk update or export, relative to the Search by Card results table. Action buttons can be configured using the Web UI Designer. For more information about action buttons, refer to the **Action Buttons** section of the **Web User Interfaces** guide within **STEP Online Help**.
7. **Display Mode:** Allows user to change the display modes. The list of available display modes can be edited as needed by accessing the Search by Cards Node List Properties Child Component Display Modes parameter. For more information on the list of available display modes, refer to the **Node List Component** topic within the **Web User Interfaces** section of **STEP Online Help**.

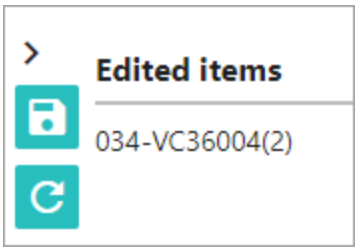



8. **Flip table icon:** Allows users to change or flip the orientation of the table.
9. **Row selection boxes:** Allows for selection of one or more rows within the table. Selecting at least one row can change the action buttons that display within the toolbar.

10. **Hyperlinks:** 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.



11. **Results table:** Displays the list of search results. A Node List component is used to display the results of selections made within the intelligent search interface. Each column label can be edited as needed by accessing the Search by Cards Node List Properties Headers parameter.
12. **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 below, it is obvious that the 034-VC36004(2) object 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.




13. **Number of items:** List the total number of search result items that are listed within the results table.

# Search by Card Initial Configuration

Search by Card allows users to find specific data by using the sync between search cards functionality to narrow down the search result and only display the relevant result.

Search by Card properties can be edited at any time. However, it is best to configure the properties early on. Removing search criteria later may impact searches you have already saved. If removed, the saved search still remains accessible, but the Search button will not be enabled.

The Search by Card link can be added to the Web UI homepage for convenient user access. In addition, the search criteria panel itself is configurable so Web UI designers can present users with criteria that are relevant to their job. It is better to define the search criteria in a manner that reduces the risk of unintended and restricted searches.

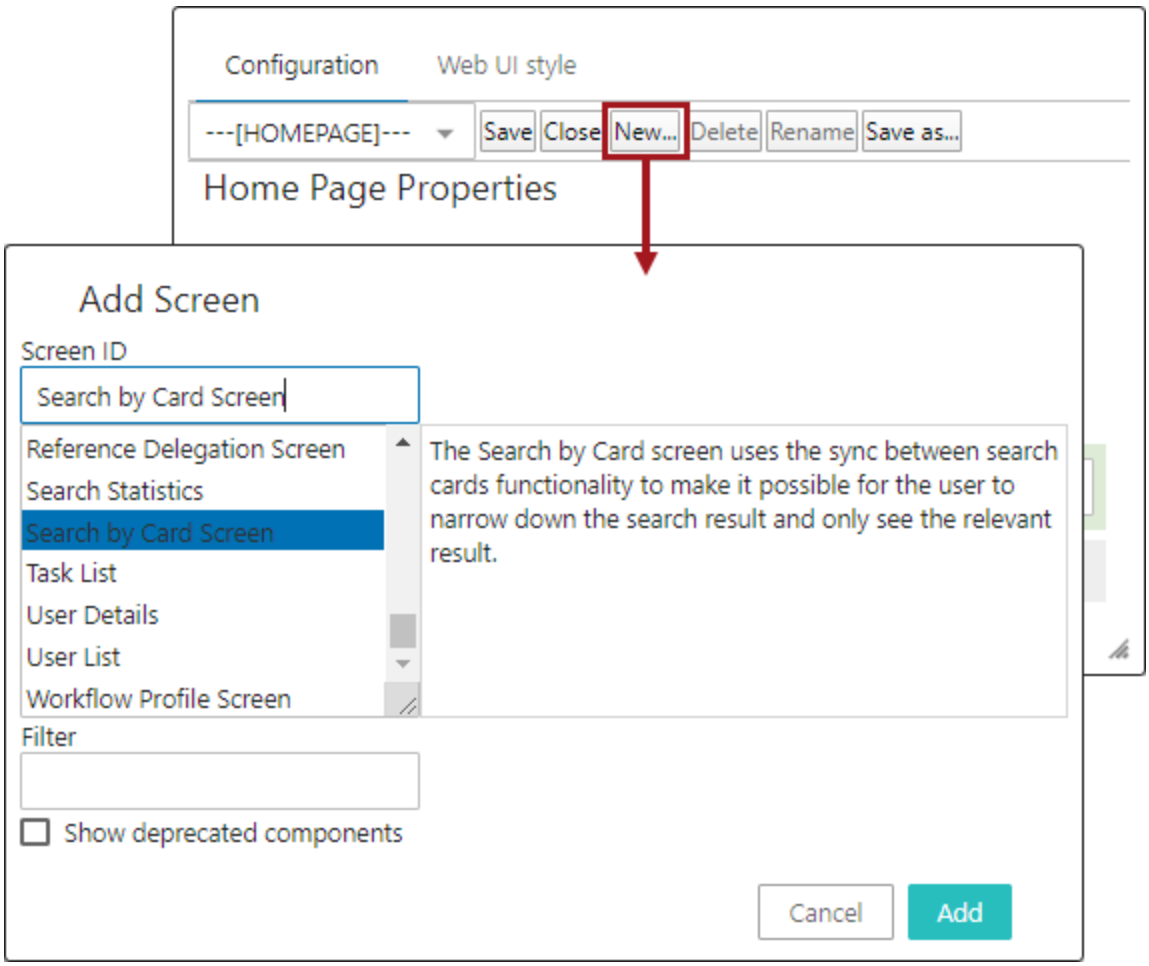
## Configuration Prerequisites

It is expected that anyone configuring the Search by Card 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** documentation.

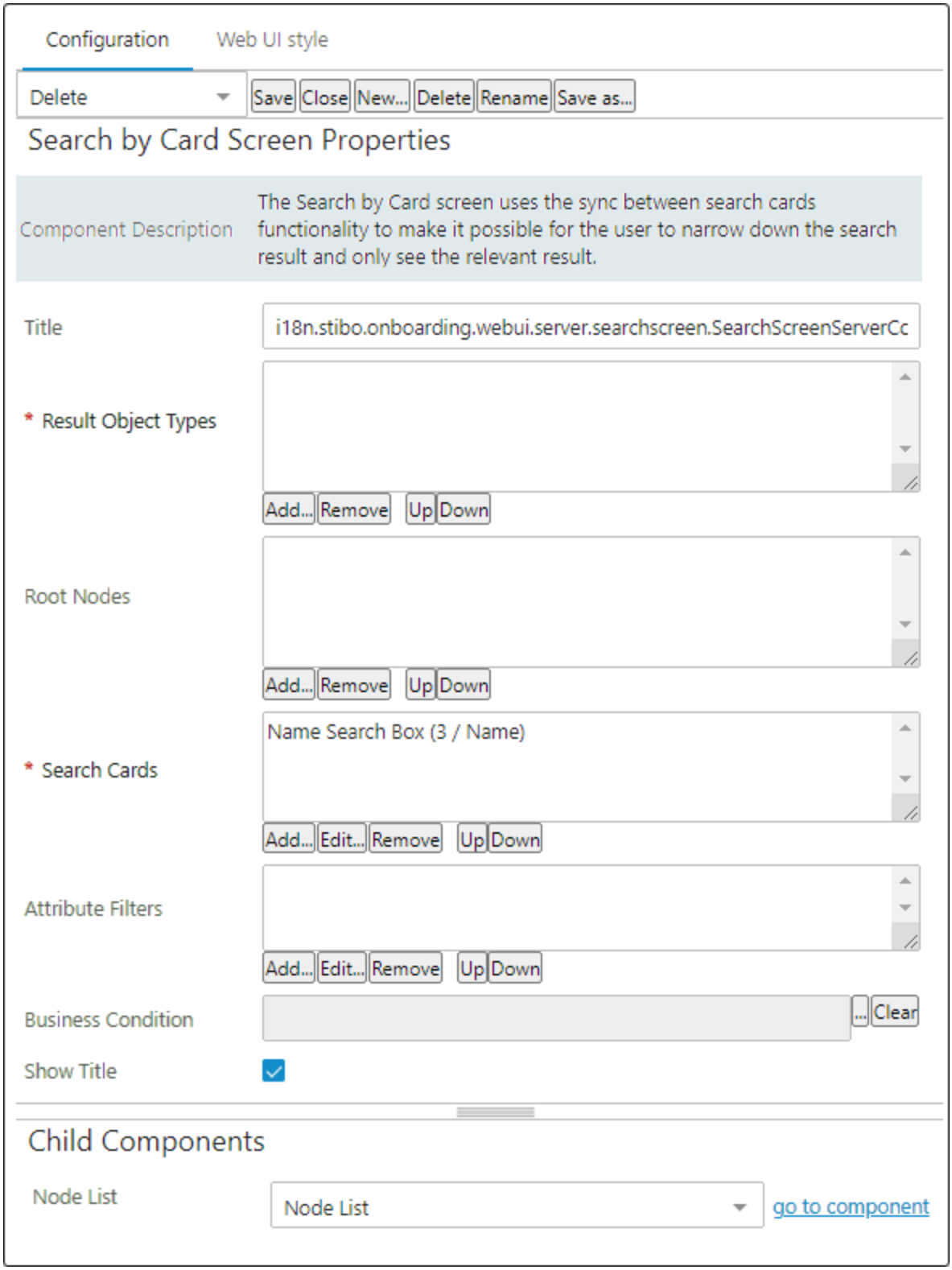
## Configuration Process

In the designer, create a new screen by selecting **New**. Select the **Search by Card Screen** and create a name for the screen in the 'Screen ID' field. In the example below, the Screen ID is 'Search by Card Screen.' Click **Add**.

The Screen ID you use should be something descriptive (e.g., Search by Card or Card Search) since this will be the name shown on any design component list in the Web UI. Screen IDs cannot be reused and must be unique.




Once this screen is created, the Search by Card Screen Properties designer screen displays, which contains the following parameters for configuration:




**Title:** To make changes to the default title in the screen, add the custom text to the Title field. If nothing is added, the Title default will display as 'Search screen.'

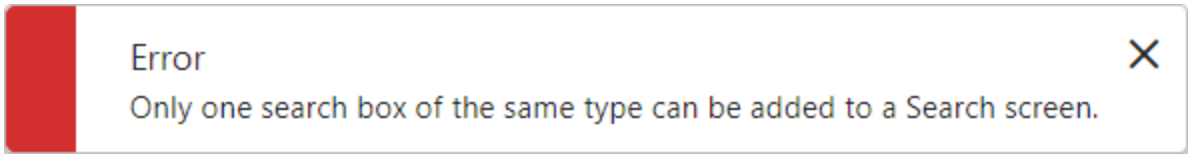
**Result Object Types:** Allows search results to display only results of the listed object type. This parameter cannot be left blank. Click the Add button below the parameter field to browse or search for the necessary object types.

**Root Nodes:** Defines the root node under which the search needs to be performed when searching / filtering for specific data. Only results from those folders that pertain to the root nodes that are defined within this parameter will be displayed. In addition, once the root node(s) is defined, the search input field displays only suggestions of objects that belong in the root node that is defined within this parameter. Click the **Add** button below the parameter field to browse or search for the necessary root node object(s).

**Search Cards:** This parameter allows adding multiple search boxes that are displayed as cards within the Search by Card Screen. Of the available options, Name Search Box and Hierarchy Search Box components can be added only once and Attribute Search Box can be added multiple times. The order that the search boxes are listed determines the order that they will be displayed within the Search by Card Screen. All of the search boxes require additional configuration. Clicking the Add button next to the Search Cards parameter displays an Add Component dialog with the option to add one of the following search boxes: Attribute Search Box, Hierarchy Search Box, and Name Search Box. Steps for configuring each search field can be found in their respective sections mentioned below:

- Configuring Name Search Box
- Configuring Hierarchy Search Box
- Configuring Attribute Search Box

Because Name Search Box and Hierarchy Search Box components can only be configured once for each Search by Card Screen, adding the search box of the same type again will cause the Error dialog as shown below to display, and the Search Cards parameter will remain unchanged.




**Attribute Filters:** Allows users to define attributes and their values. The objects that hold these attributes with the defined values will be excluded from the search result.

**Business Condition:** Optional parameter allowing users to select a business condition to be run as part of the search process.

**Show Title:** Optional parameter when checked, the title is displayed in the screen; if unchecked, the title and associated white space is removed from the screen.

**Node List:** The Search by Card Screen Properties is pre-configured with a Node List in Table Display Mode. Select the 'Node List' option in the dropdown and click on the 'go to component' to add more table header components in the Node List component. For more information, refer to the **Node List Component** topic within the **Web User Interfaces** section of **STEP Online Help**.

## Adding Search by Card to the Homepage

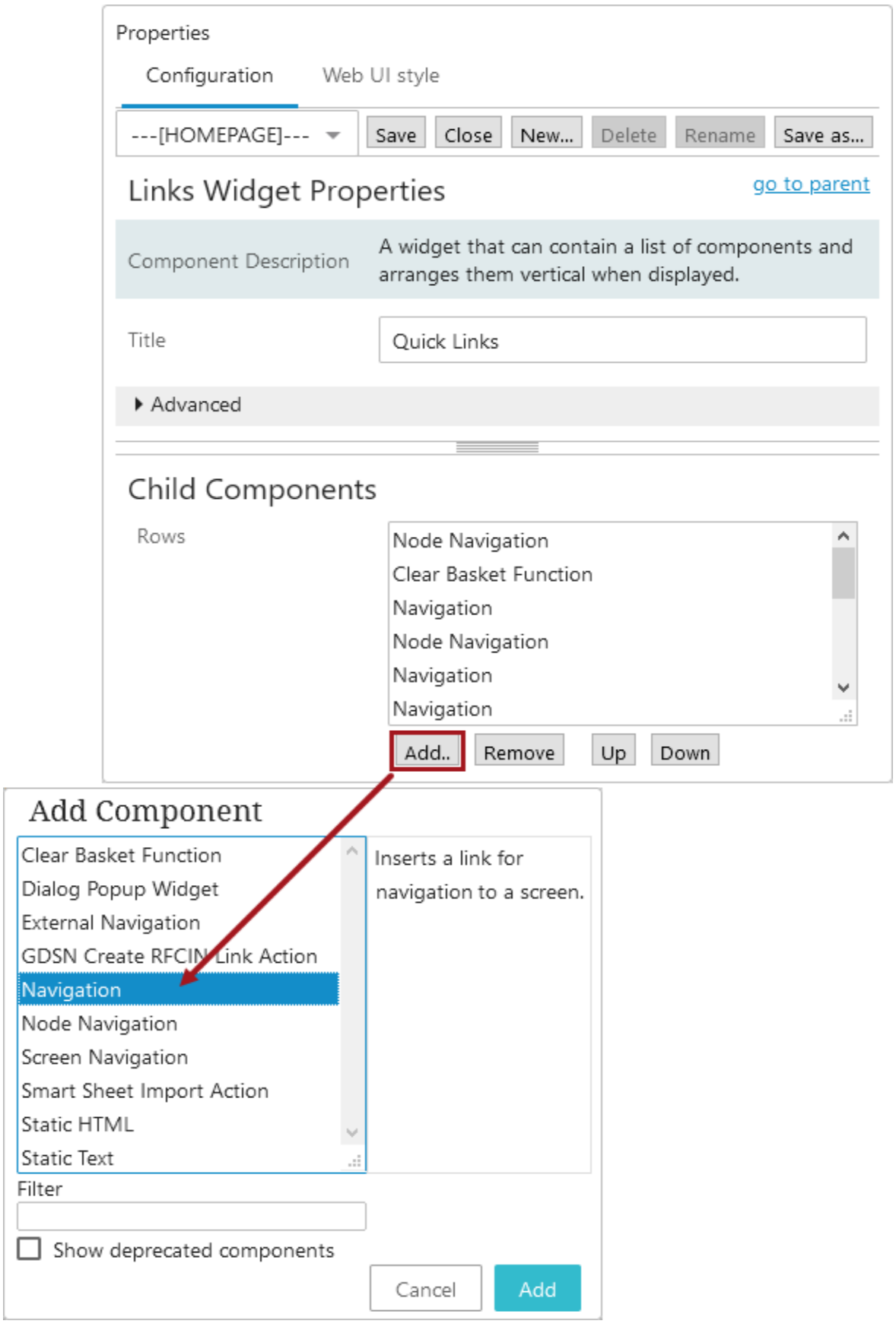
To add a link to Search by Card on the homepage:

1. Log in to the Web UI. User must have privileges to edit Web UI configurations.
2. Enter design mode.
3. The link must now be configured to display in the **Links Widget**.

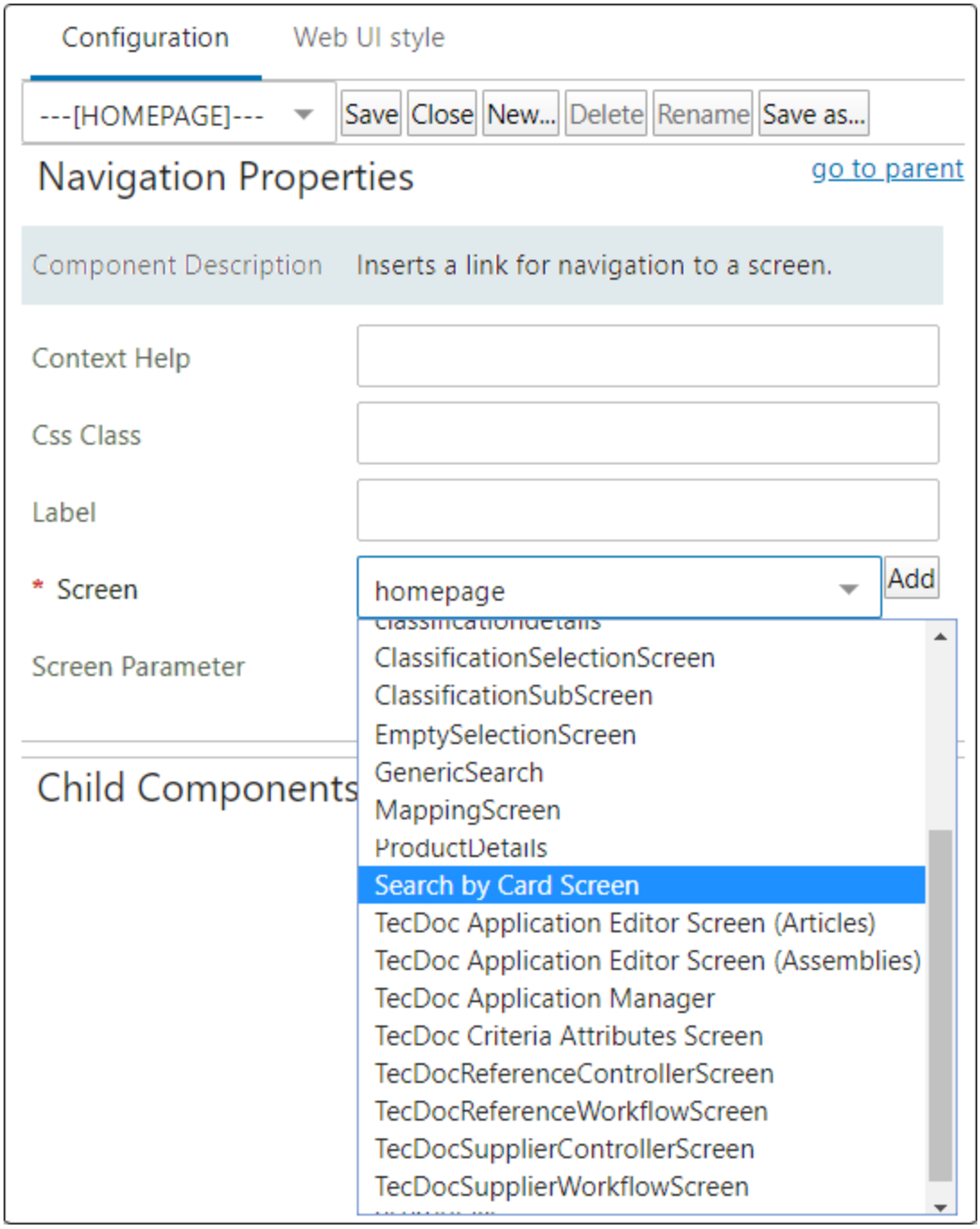
   Select HOMEPAGE from the screen dropdown in the designer and select the Links Widget in the applicable Widget Grid (found under Home Page Properties > Child Components > Content).

   If the Links Widget has never been configured to display on the Homepage, it will need to be added to a Widget Grid. Details on how to do this can be found in the **Adding Widgets to a Homepage** topic in the **Getting Started** documentation of the STEP Online Help.

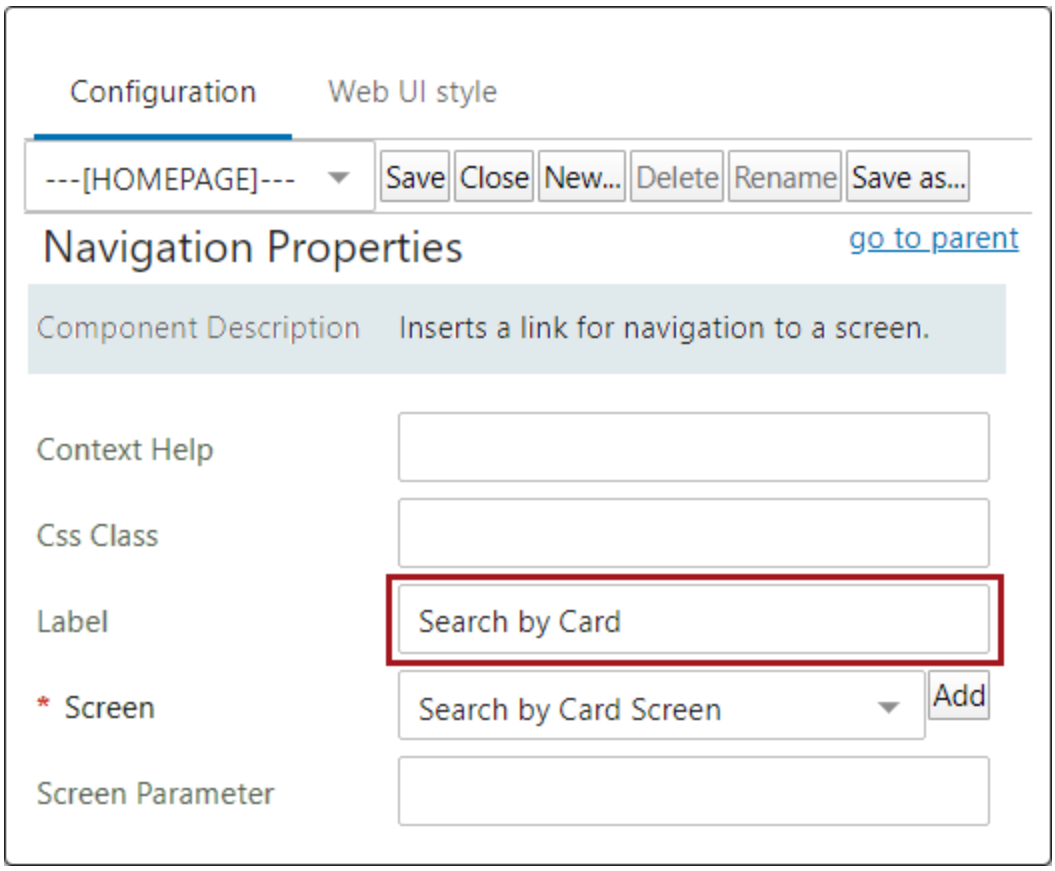
4. Within Links Widget Properties, go down to the Child Components section and click the **Add** button.


Properties
Configuration
Web UI style
---[HOMEPAGE]---
Save
Close
New...
Delete
Rename
Save as...
Links Widget Properties
go to parent
Component Description
A widget that can contain a list of components and arranges them vertical when displayed.
Title
Quick Links
Advanced
Child Components
Rows
Node Navigation
Clear Basket Function
Navigation
Node Navigation
Navigation
Navigation
Add..
Remove
Up
Down
Add Component
Clear Basket Function
Dialog Popup Widget
External Navigation
GDSN Create RFCIN Link Action
Navigation
Node Navigation
Screen Navigation
Smart Sheet Import Action
Static HTML
Static Text
Inserts a link for navigation to a screen.
Filter
Show deprecated components
Cancel
Add

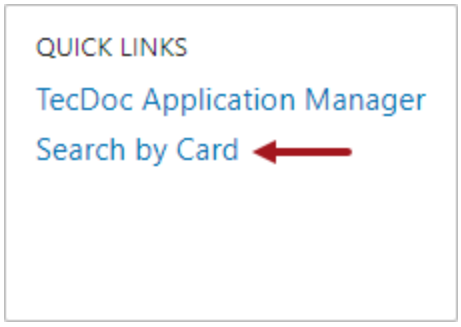
5. Choose the Navigation component and click **Add**.
6. On the Navigation properties screen, click the dropdown available towards the right of the Screen field.
7. Select the desired Search by Card Screen.




8. Fill in the Label field with the name that will show on the links widget, then click **Add**.




9. You can move the Links Widget Child Components up and down based on your preference for how they show on the homepage.
10. Click **Save** in the Properties Configuration dialog box.
11. The Advanced Search link will now show under Quick Links and is identified by the name you gave the link in Step 8.



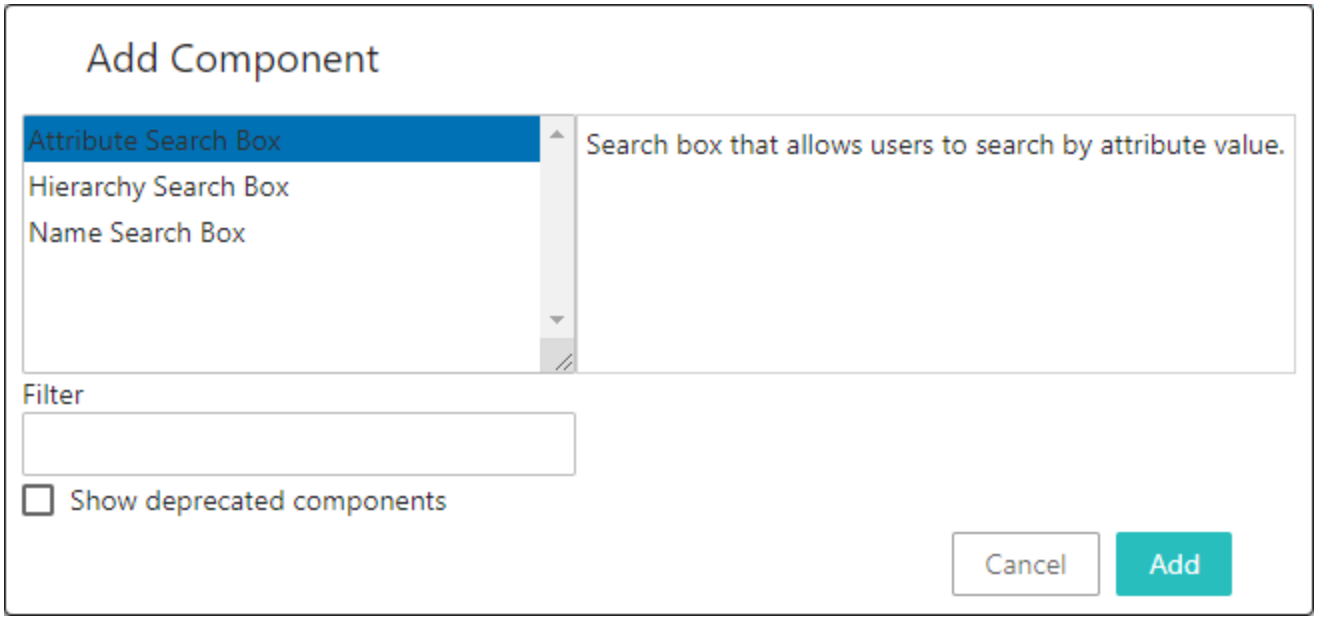

**Note:** Additional information regarding STEP Web UI Designer (entering design mode, selecting screens to edit, and more) can be found in the **Web User Interfaces** / **Getting Started** documentation of the STEP Online Help.

Search by Card Screen is now set up and ready for use.

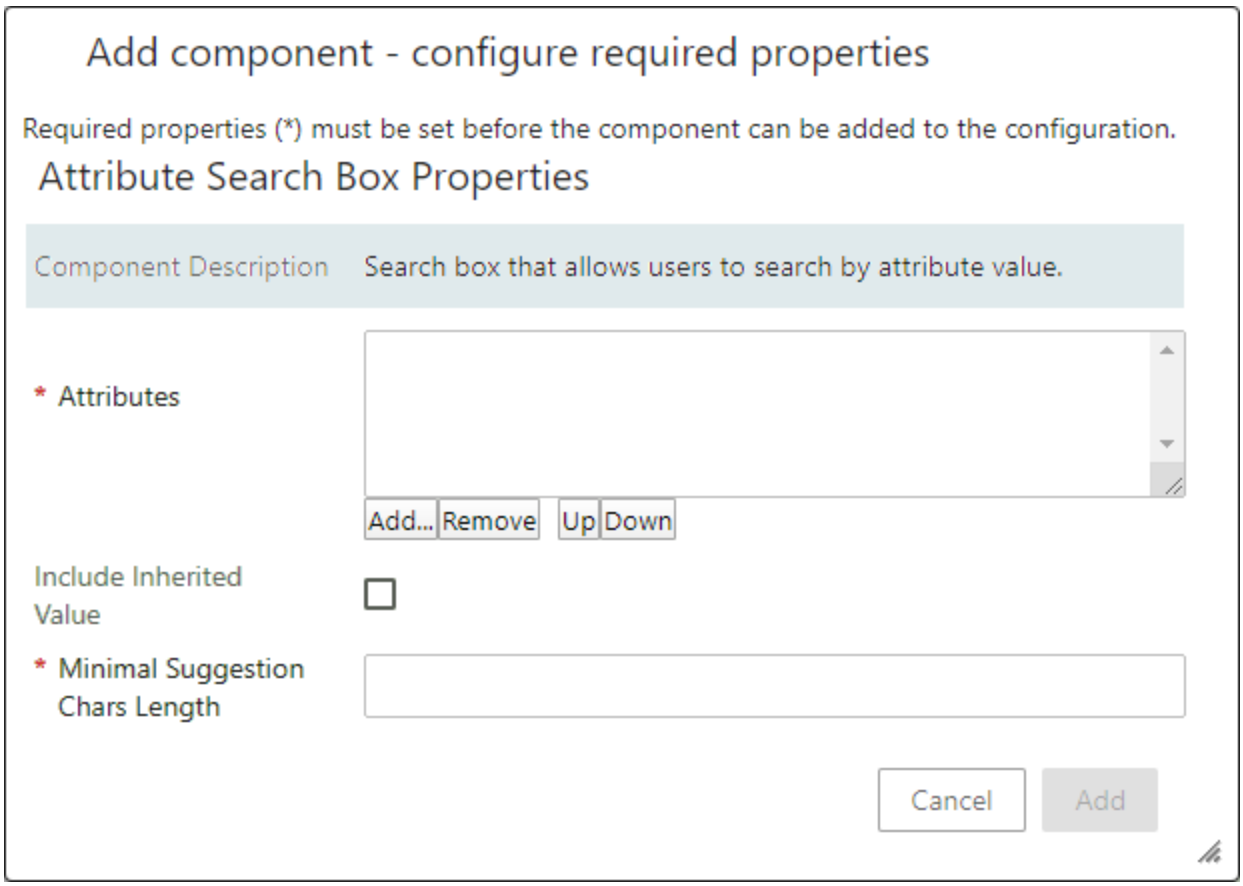
## Configuring the Attribute Search Box

The Attribute Search Box will search for attribute values that are valid for the object type that is intended for search. For example, in the AutoCare standard, if a Base Vehicle object is being searched, then the Attribute Search Box will by default search only on the attribute(s) that are valid for the Base Vehicle objects to display (e.g. VCdb Year). If a root node has been configured in the Root Nodes parameter, then the Attribute Search Box will only search below the configured root node(s). To add the Attribute Search Box component within the Search Cards parameter follow the instructions below.

1. From the Search by Card Screen Properties dialog, click the **Add** button next to the Search Cards parameter field. The Add Component dialog will display.




2. Click **Attribute Search Box** from the list of components.
3. Click the **Add** button, and the Attribute Search Box Properties will display, as shown below.



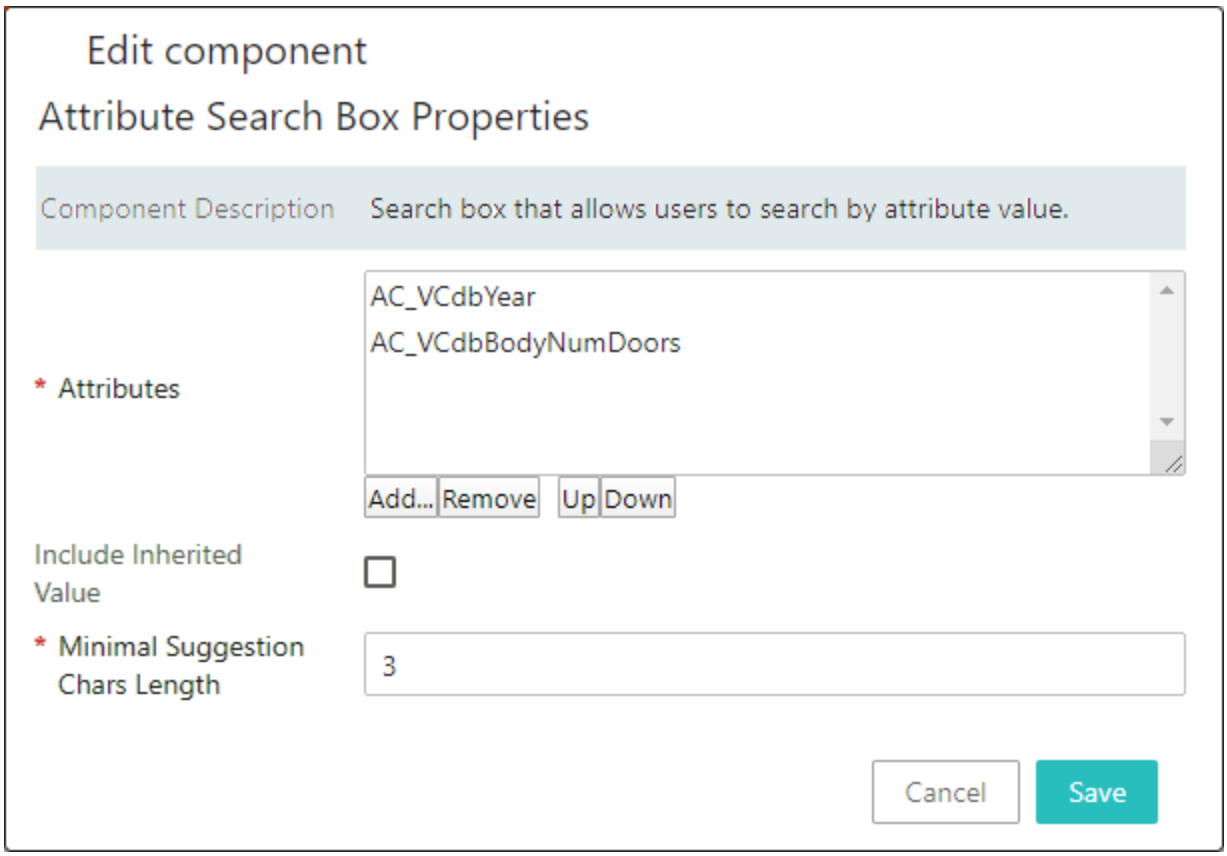

**Attributes:** This required field allows users to add one or more attribute criteria to their search. All attribute validation base types (e.g., Text, Number, List Of Values) are allowed. Click the ellipsis button (...) to find and select the required attribute. The selected attribute must be valid for the object type that is intended for search, and the attribute value must be populated on that object in order for it to display correctly. The name of the attribute will display in the configured search box.

**Include Inherited Value:** This checkbox determines the requirement to include the inherited attribute values in its search criteria. Check the field to enable searching of the inherited attribute values in the search card.

**Minimal Suggestion Length:** This required field determines the minimum number of characters that must be entered into the typeahead field so the associated card starts suggesting relevant values in the dropdown.

4. Once the required parameters are populated, click the **Add** button to return to the Search by Card Screen Properties and **Save** the configuration. Users may add additional search fields, if needed.

   Below is an example of a configured Attribute Search Box component:



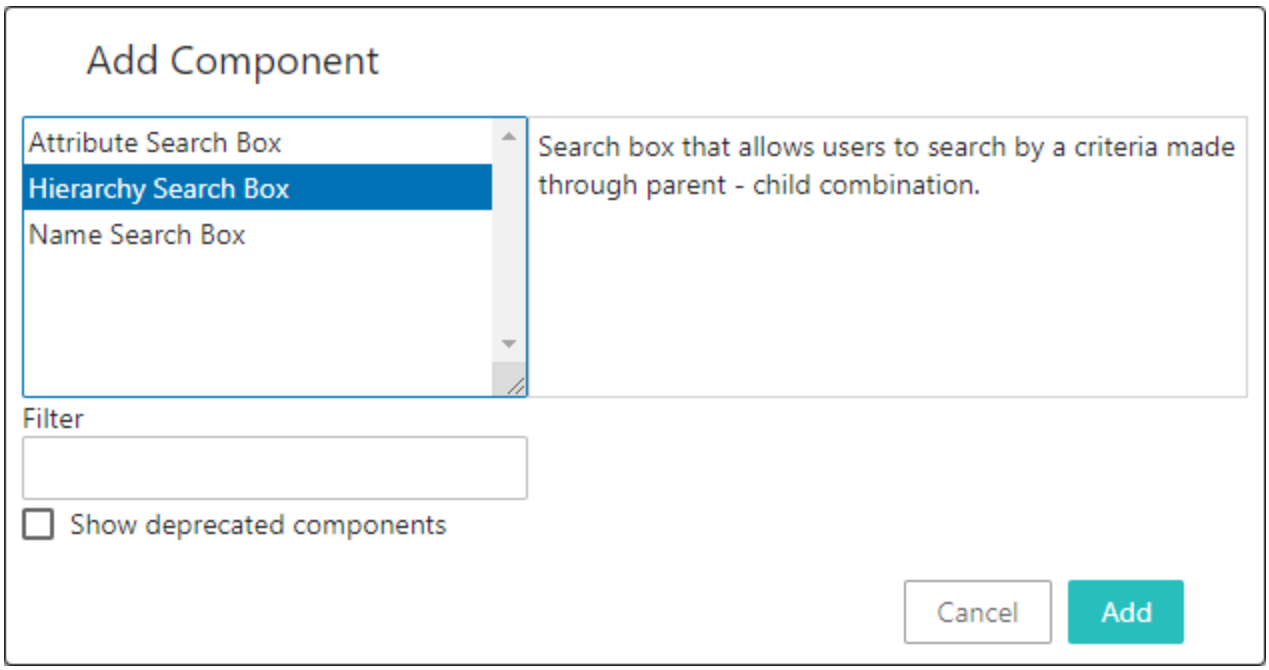

## Configuring the Hierarchy Search Box

The Hierarchy Search Box will search for the parent folder(s) of the searched object in order to more efficiently gather relevant search results. For example, in the AutoCare standard, if a Vehicle object is to be searched for, then the Hierarchy Search Box can be configured to search for the Make/Model to display all of the Vehicle objects (Vehicle) for the selected Make/Model.

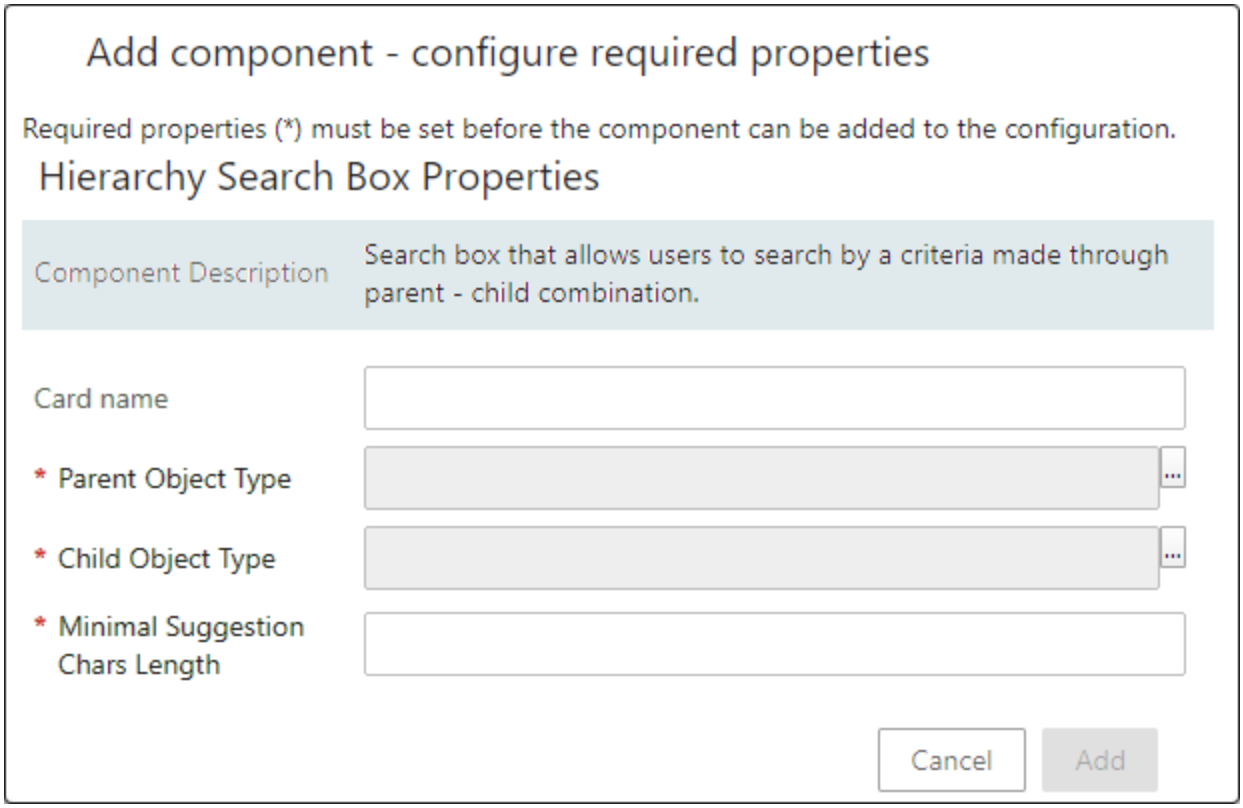
If a root node has been configured in the Root Nodes parameter, then the Hierarchy Search Box will only search below the configured root node(s). For example, the Light Duty Vehicle root node can be added so that the search will only be performed to include Light Duty vehicles and exclude all other vehicle types (Medium/Heavy Duty, Powersports, etc.).

To add and configure the Hierarchy Search Box component within the Search Cards parameter, follow the instructions below:

1. From the Search by Card Screen Properties dialog, click the **Add** button next to the Search Cards parameter field. The Add Component dialog will display.




2. Click **Hierarchy Search Box** from the list of components.
3. Click the **Add** button, and the Hierarchy Search Box Properties will display.



**Card name:** This optional field lets the user provide a label for the card associated with the Hierarchy Search Box. It is a good practice to label the card that best communicates to the user what criteria are available for selection within the card. By default, the label is set to 'Hierarchy.'

**Parent Object Type:** Determines the parent object type that a user can select within the Hierarchy card. Click the ellipsis button (...) to find and select the object type that should be displayed for the user as a parent object to select within the typeahead dropdown.

The defined object type should meet all of the following criteria:

- Should exist as any top-level parent folder for the object type that is intended for search.
- Should be present in the hierarchy of the root node defined in the Root Nodes parameter.
- Cannot be the object type that is intended for search.

**Child Object Type:** Determines the child object type that a user can select within the Hierarchy card. Click the ellipsis button (...) to find and select the object type that should be displayed for a user as a child object to select within the typeahead dropdown.

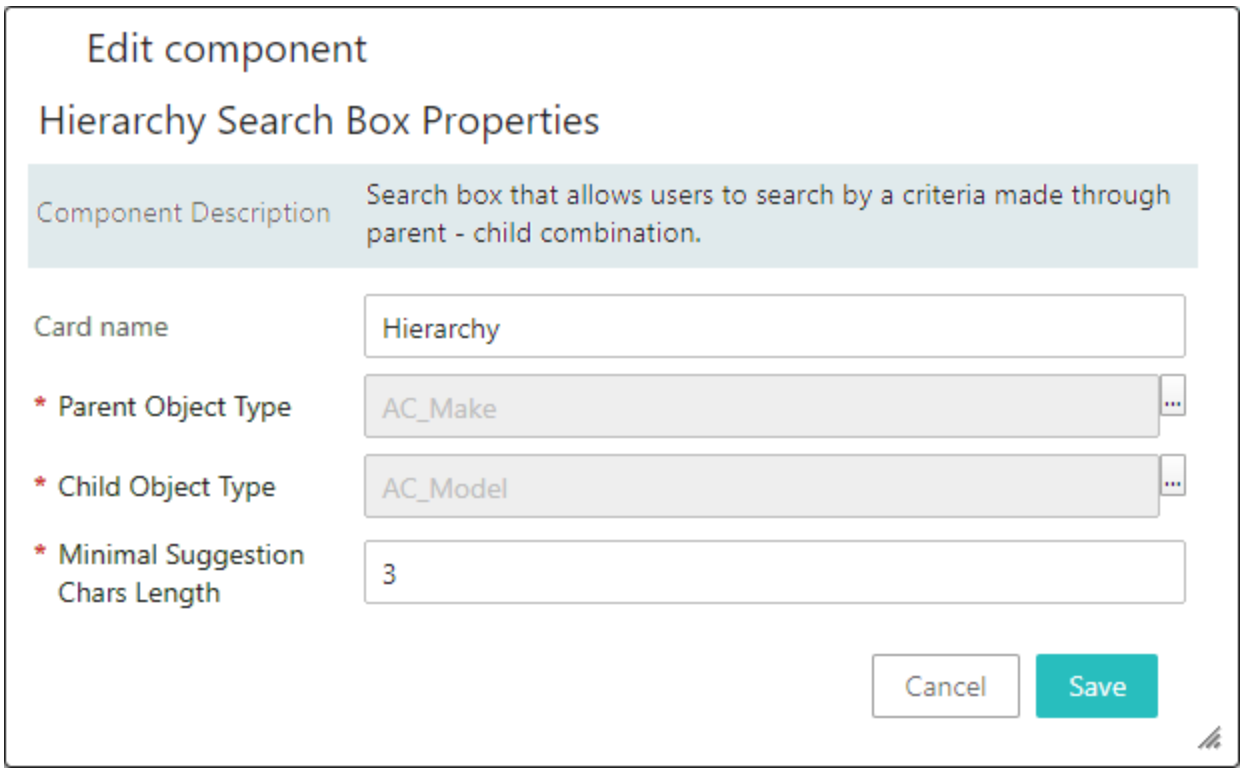
The defined object type should meet all of the following criteria:

- Should exist as any top-level parent folder for the object type that is intended for search.
- Should exist as any lower-level folder below the object type defined in the Parent Object Type parameter.
- Should be present in the hierarchy of the root node defined in the Root Nodes parameter.
- Cannot be the object type that is intended for search.

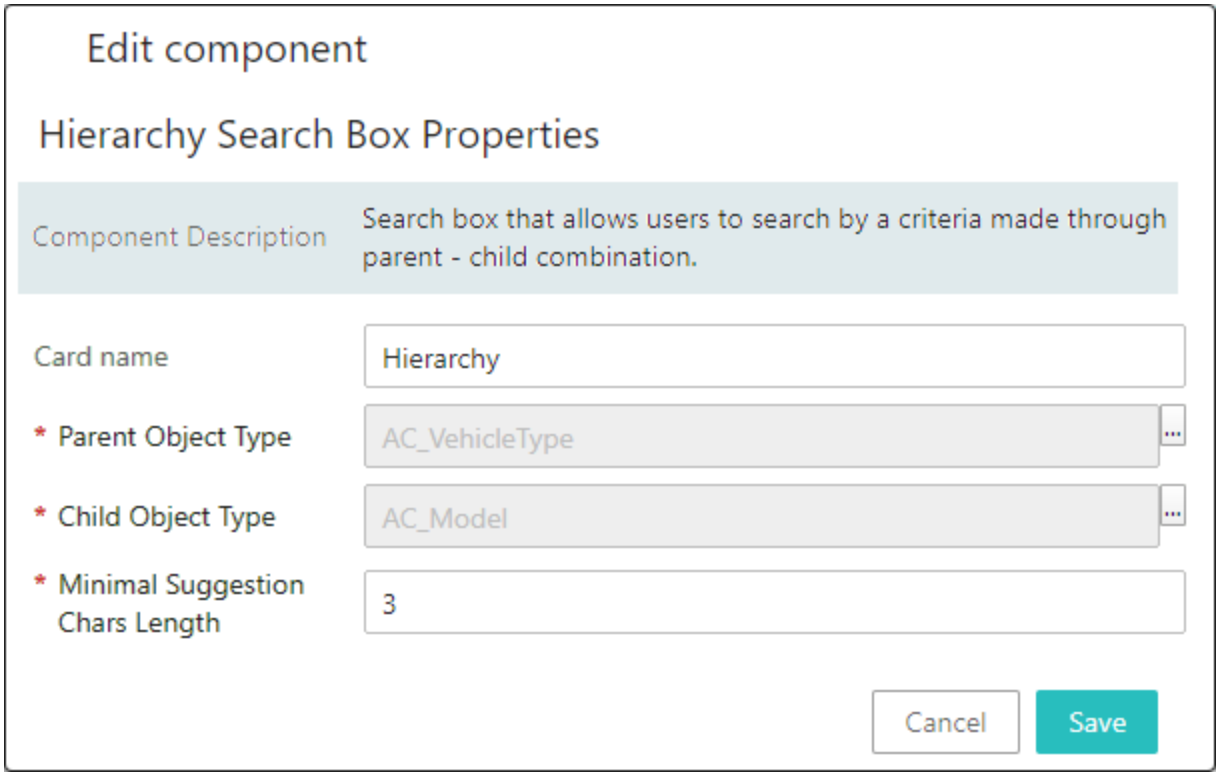
**Minimal Suggestion Length:** This required field determines the minimum number of characters that must be entered into the typeahead field so the associated card starts suggesting relevant object names in the dropdown.

4. Once the required parameters are populated, click the **Add** button (or **Save** button) to return to the Hierarchy Search Box Properties and **Save** the configuration.

   Below is an example of a configured Hierarchy Search Box component:



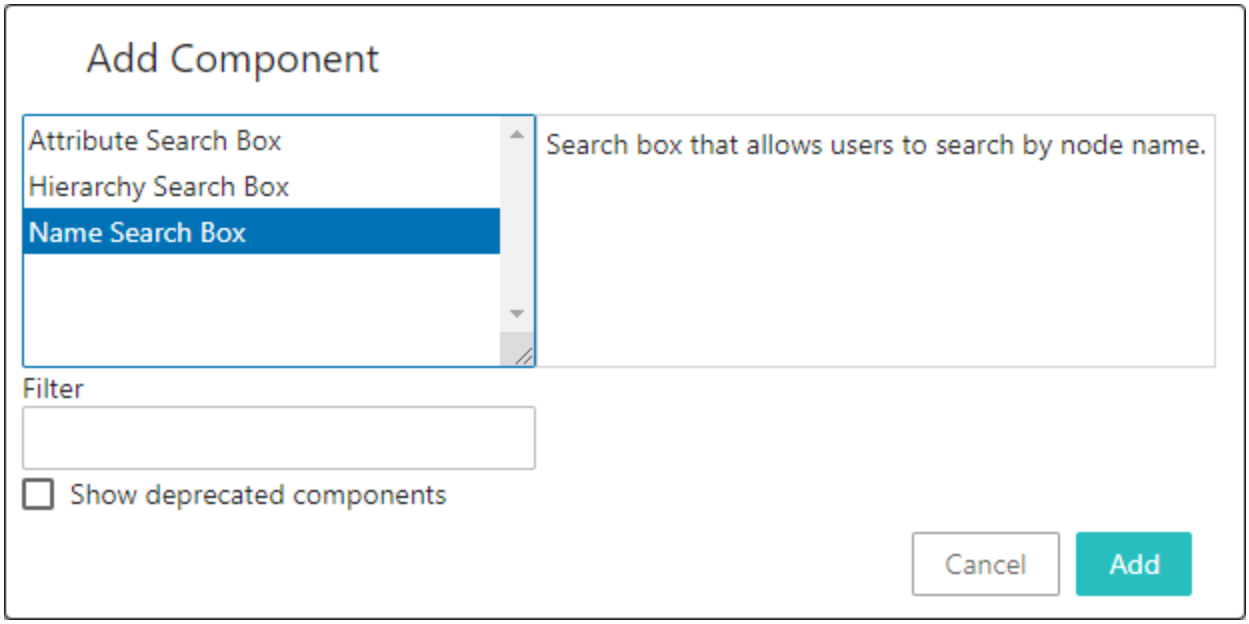

Below is another example of a configured Hierarchy Search Box component:



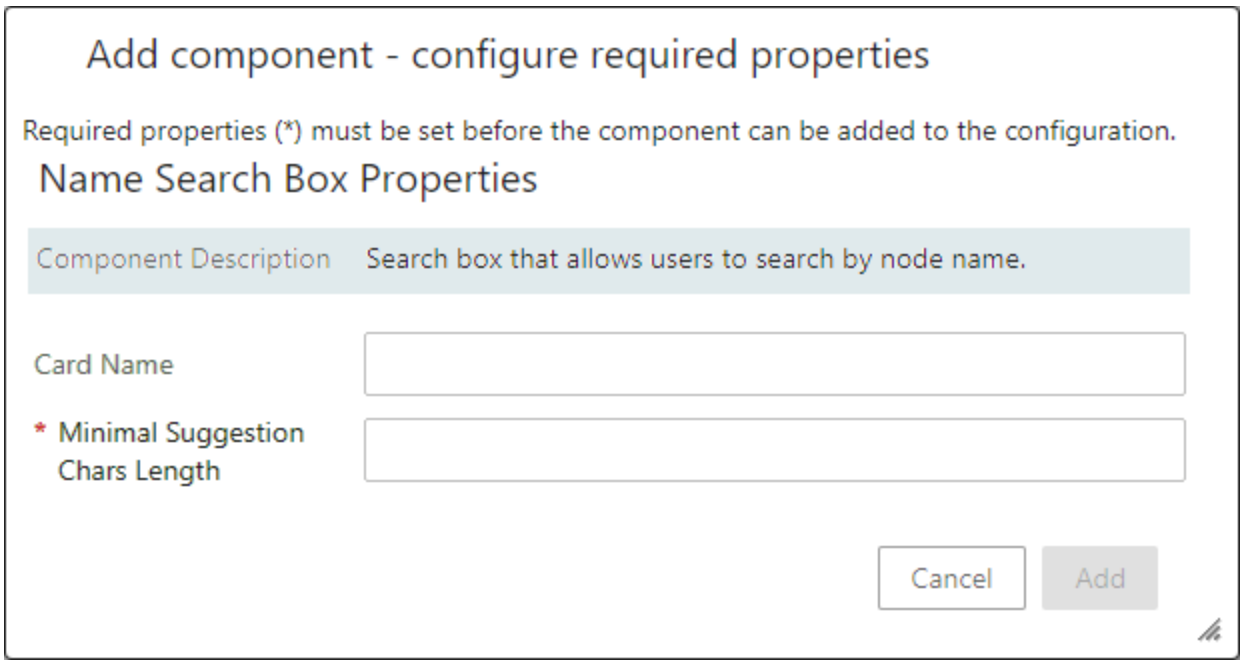
## Configuring the Name Search Box

The Name search box will search for the name of the object that is intended for search. For example, if the Search by Card Screen is configured to search for the Base Vehicle objects, then the Name search box will by default search on the Name of the Base Vehicle objects to display. If a root node has been configured in the Root Nodes parameter, then the Name search card will only search below the configured root node(s). To add the Name Search Box component within the Search Cards parameter:

1. From the Search by Card Screen Properties dialog, click the **Add** button next to the Search Cards parameter field and the Add Component dialog will display.




2. Click **Name Search Box** from the list of components.
3. Click the **Add** button, and the Name Search Box Properties will display, as shown below.

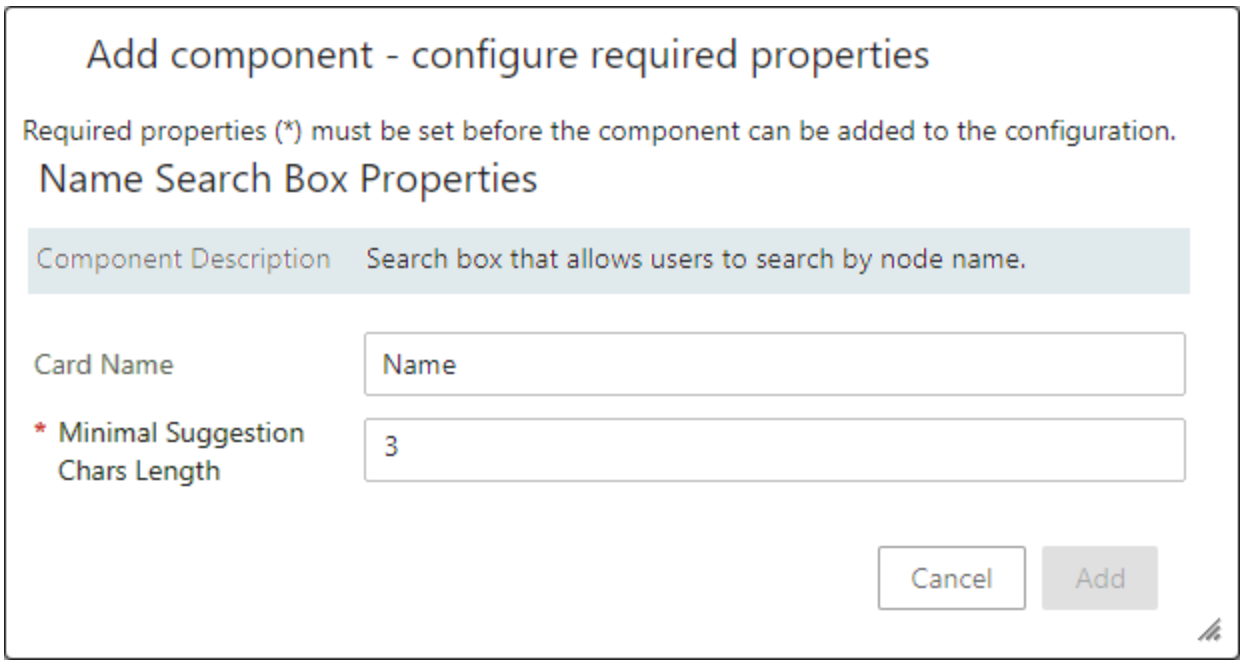



**Card Name:** This optional field allows the user to provide a label for the card associated with the Name Search Box that best communicates to the user what criteria are available for selection within the card. By default, the label is set to 'Name.'

**Minimal Suggestion Length:** This required field determines the minimum number of characters that needs to be typed in the typeahead field so that the associated card starts suggesting the names in the dropdown matching with the typed characters.

4. Once the component is populated, click the **Add** button to return to the Search by Card Screen Properties and Save the configuration. Optionally add another search box.

   Below is an example of a configured Name Search Box component:

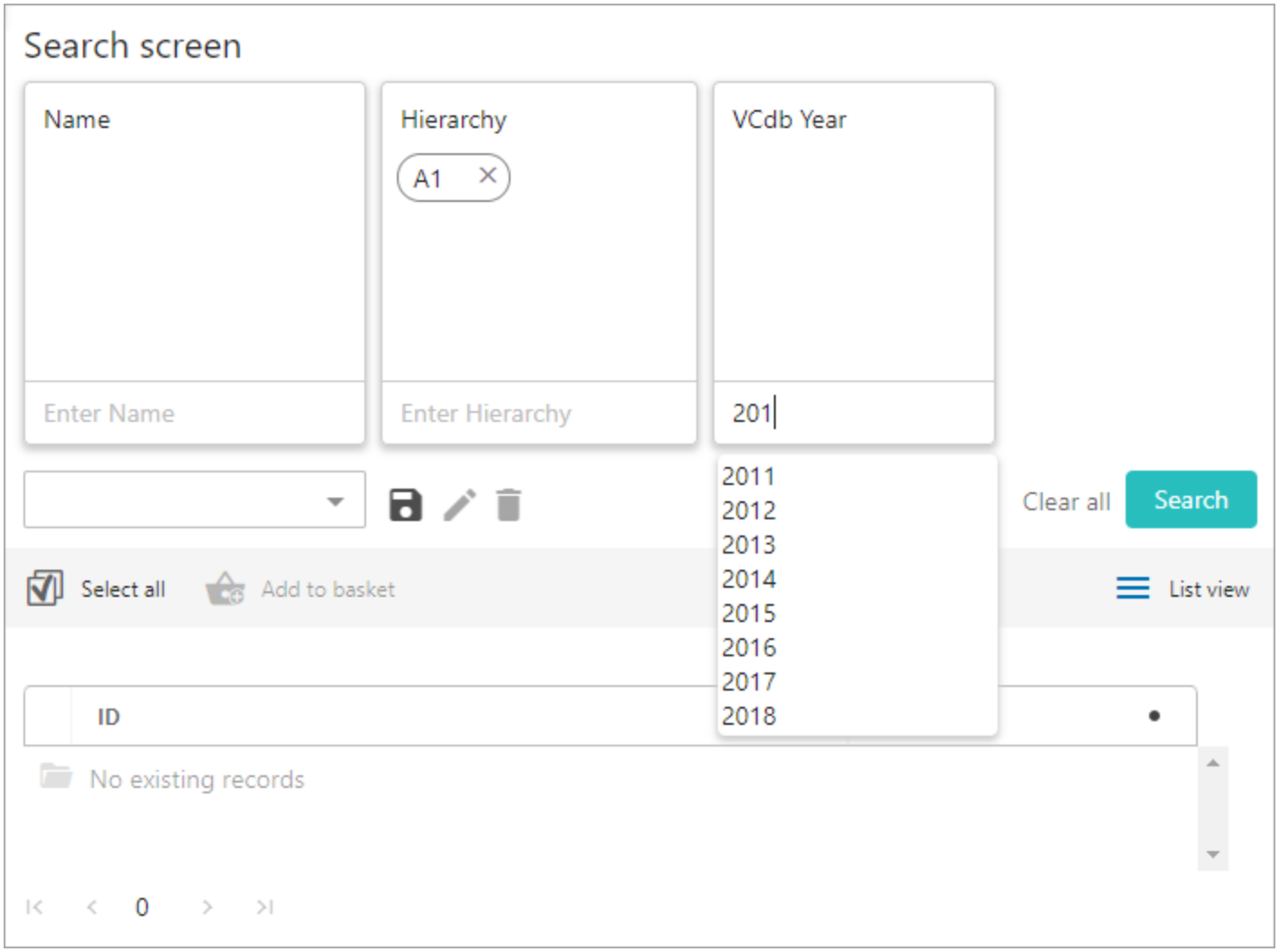



## Using Search by Card Screen Interface

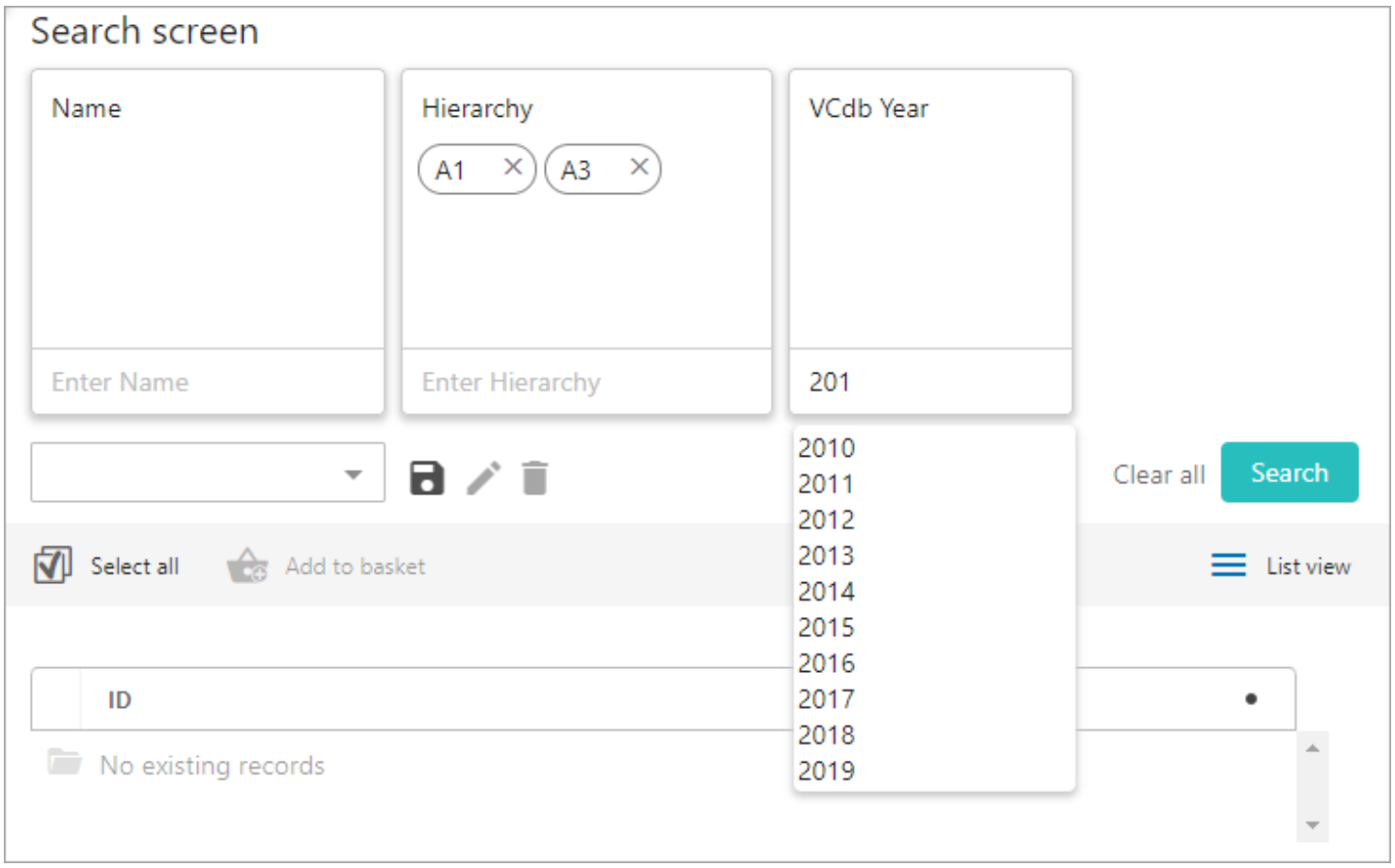
The Search by Card Screen intelligent search interface has been carefully designed to allow users to quickly and easily select criteria for valid combinations of Name, parent hierarchy, and attribute values. This allows users to create accurate search criteria on the fly and benefit from the customized search results. This is accomplished by allowing users to select only valid criteria across multiple search cards per the configuration of the Search by Card Screen.

Each Search by Card Screen can be configured to display up to five search cards. Of the available options, Name Search Box and Hierarchy Search Box components can be added only once and Attribute Search Box can be added multiple times. Within each search card, a typeahead field is used to view and select valid search criteria. Because only valid criteria can be selected within each typeahead field, the Search by Card Screen provides automated intelligence for each search. Thanks to the synchronization of the search cards, the valid criteria is considered upon each selection and within any of the displayed search cards, no matter the order of entry.

For example, if the Hierarchy search card contains a Base Vehicle that was only produced from 2011-2018, then the typeahead field within the attribute search card (configured to display VCdb Year attribute) will only allow to enter those years for selection (notice that the initial characters of the attribute value have to be typed in the type ahead field). Because Audi (Make) has A1 (Model) beginning with the 2011 model year, the available years are displayed as 2011-2018 when ‘A1’ is selected within the Hierarchy search card (as shown below).




However, if an additional Model (A3 ) is added to the Hierarchy search card, then the available years display as 2010-2019 (notice that the initial characters that are typed in the typeahead field are '201').




This section addresses the following:

- **Using Search Cards**
- **Using Hierarchy Search Card**
- **Using Attribute Search Card**
- **Using Name Search Card**

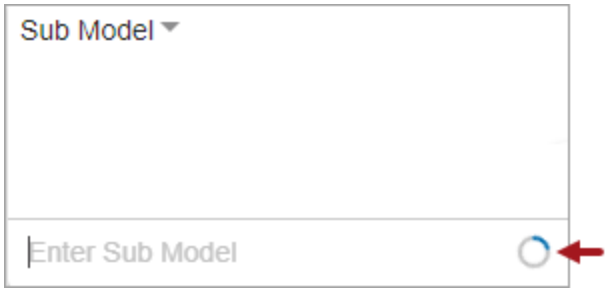
## Using Search Cards

Though each search card contains its own typeahead field, and each typeahead field allows the user to enter data specific to that search card, common keyboard shortcuts are compatible within these typeahead fields. The functionality that is specific to each search card is addressed in subsequent sections of this guide. This section addresses the following functionalities that are consistent across the Search by Card Screen search cards:

- Searching for Criteria
- Selecting Valid Criteria
- Removing Criteria

### Searching for Criteria

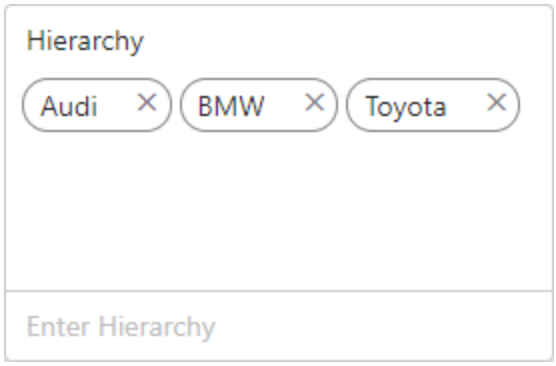
The cursor is activated for each search card by clicking within the typeahead field at the bottom of the card. Text typed into the field is used to filter and then display only valid criteria within a dropdown list. Typing an asterisk (*) displays any available criteria for selection in the dropdown. 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 card 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 card. Each selected criterion will display as a tile within the search card. In the example below, three criterion tiles have been added to the Hierarchy search card.




Once a criterion is selected to display within a search card, it will no longer display within the dropdown for selection. This allows users to focus on valid options they have not yet selected.

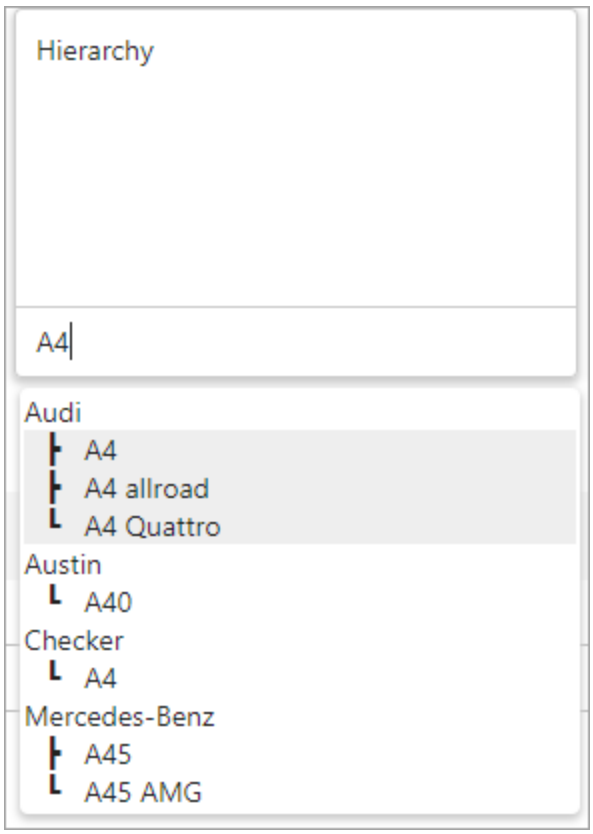
The Tab key on the keyboard can be used to move the cursor from left to right across different search cards. Holding the keyboard Shift key down while pressing the Tab key will move the cursor from right to left.

### Selecting Multiple Criteria

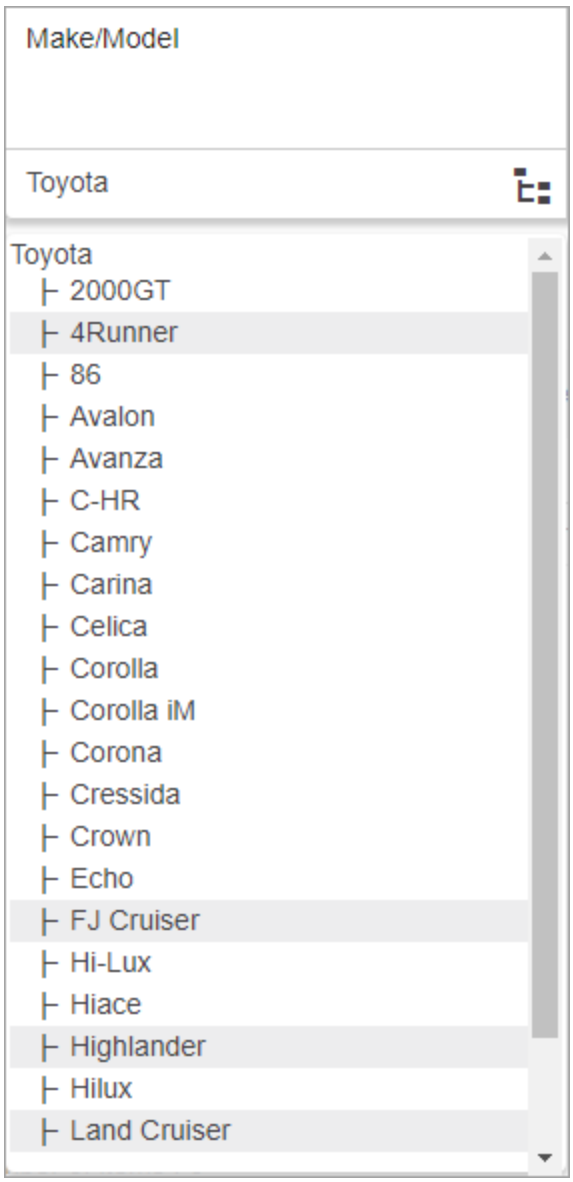
Within a dropdown list, multiple criteria (either as a group or scattered throughout the list) can be selected and added to a search card. Holding down the Shift key on a keyboard will enable the selection of multiple criteria. While holding down the Shift key, the mouse pointer can be used to click 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 card.

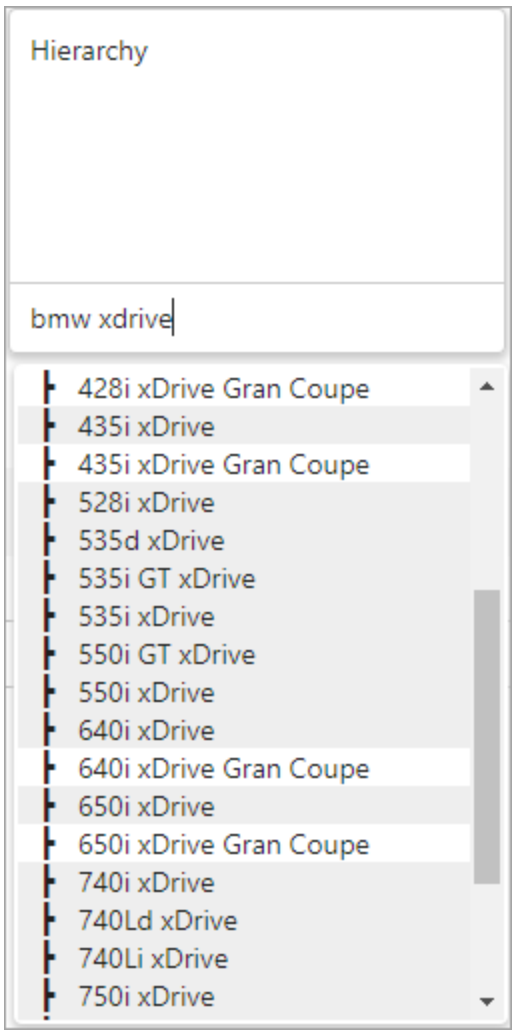
In the example below, after a list of valid criteria options displayed are within the dropdown list, the down arrow key was used to highlight the Audi A4 option, the Shift key was held down, and the down arrow key was used to highlight the two additional Audi model options.




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 card.




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 card, or the 'Clear all' link can be used to remove all selected criteria. To remove a criterion from a search card using the mouse, click the 'X' at the right of a criterion tile.

### Minimum Search Criteria

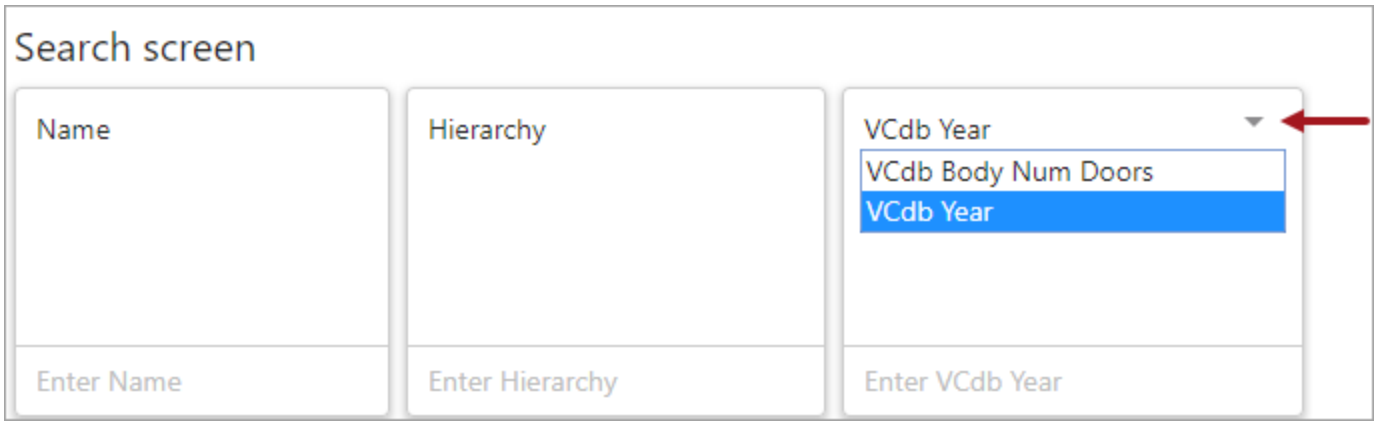
Though only one value within one search card is required for the Search button to be enabled, the recommended practice is to provide as much details as possible prior to running a search. When broad search criteria are provided, results can take much longer than expected to display.

## Using Attribute Search Card

The Attribute search card allows users to add one or more attribute criteria to their search. This section addresses functionality that is specific to the Attribute search card. For more information on the general functionality found within the search cards, refer to the **Using Search Cards** topic within this guide.

When the cursor is placed within the typeahead field for the Attribute search card, and all other search cards are blank, the dropdown will not display until at least the initial characters are typed into the typeahead field. The minimum character(s) requirement for typeahead is configured in design mode. For more on configuration, refer to the **Configuring the Attribute Search Box** topic within this guide.

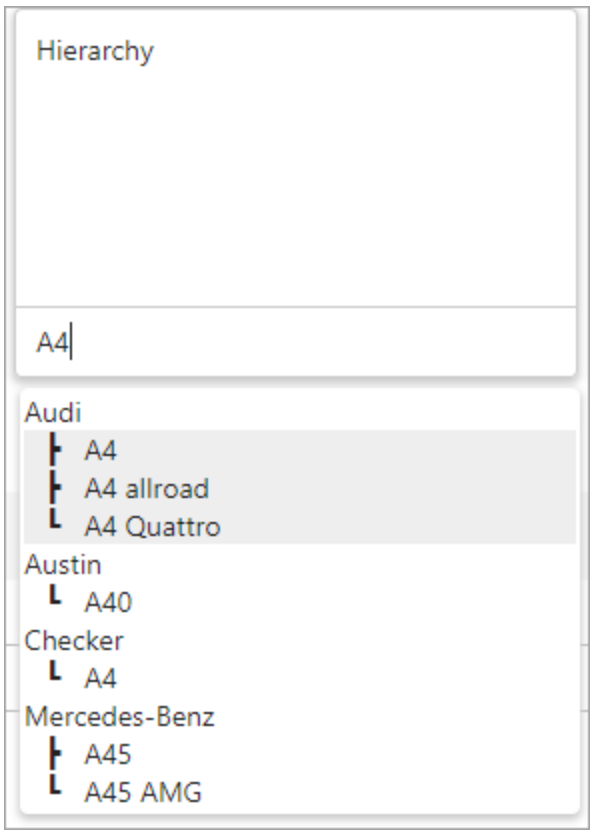
When multiple attributes are defined to be in the search criteria within the same Attribute search card, a dropdown will be displayed next to the title of the search card that allows users to select the required attribute search criteria (as shown below).



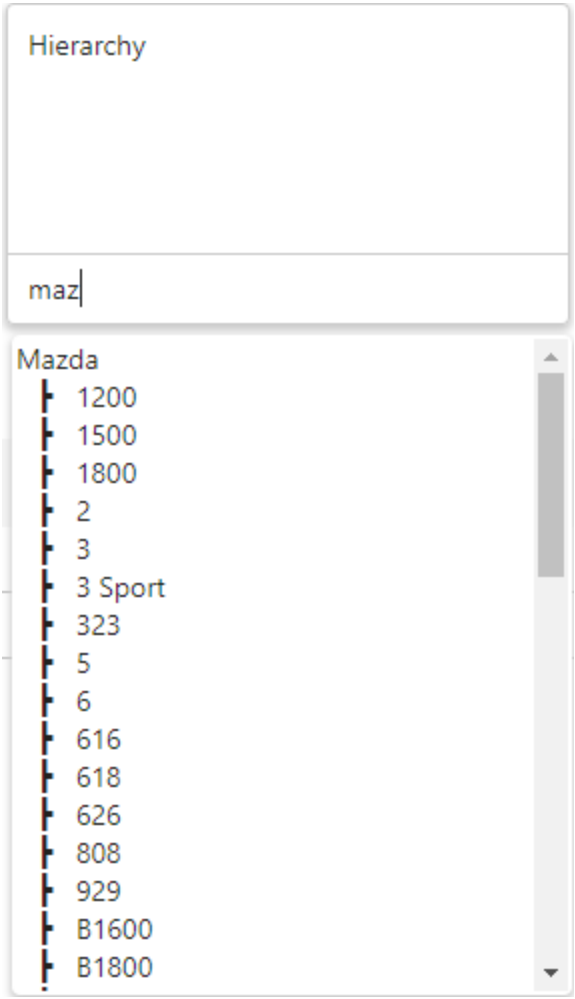

## Using Hierarchy Search Card

The Hierarchy search card allows users to add parent hierarchy criteria to their search. The Hierarchy search card lets users search based on their parent objects in order to narrow down the search. This section addresses functionality that is specific to the Hierarchy search card. For more information on the general functionality found within the search cards, refer to the **Using Search Cards** topic within this guide.

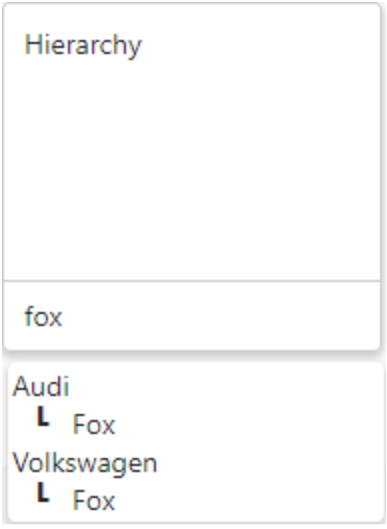
Within the Hierarchy search card, the typeahead field can be used to search for parent criteria. The dropdown list will display the criteria options alphabetically by parent, and then by child. For example, if 'A4' is typed into the field, then the dropdown list displays the Audi, Austin, Checker, and Mercedes-Benz parents alphabetically, and the children that match the information typed into the field (A4) will be listed alphabetically below their respective parents (as shown below).



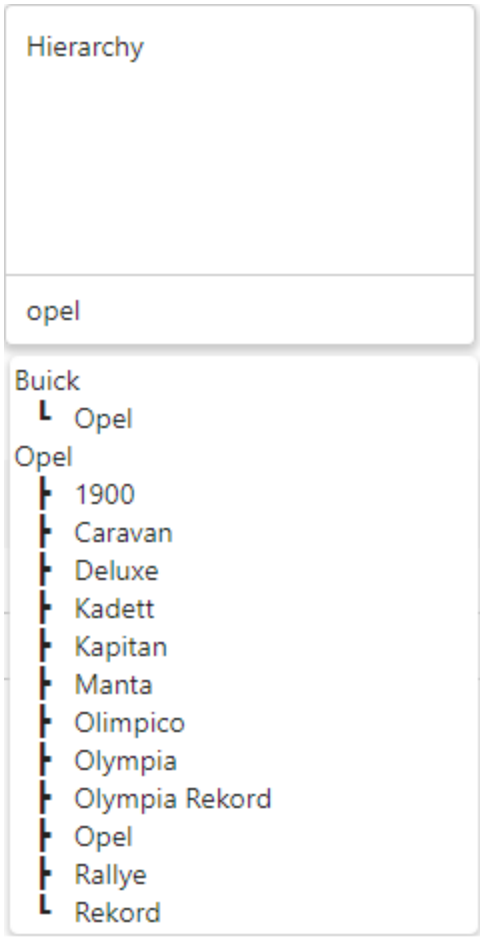

If the text typed in matches a parent only, then all children below that parent will be displayed. For example, if 'maz' is typed into the field, then the dropdown list will display the Mazda parent with all its available children listed below.



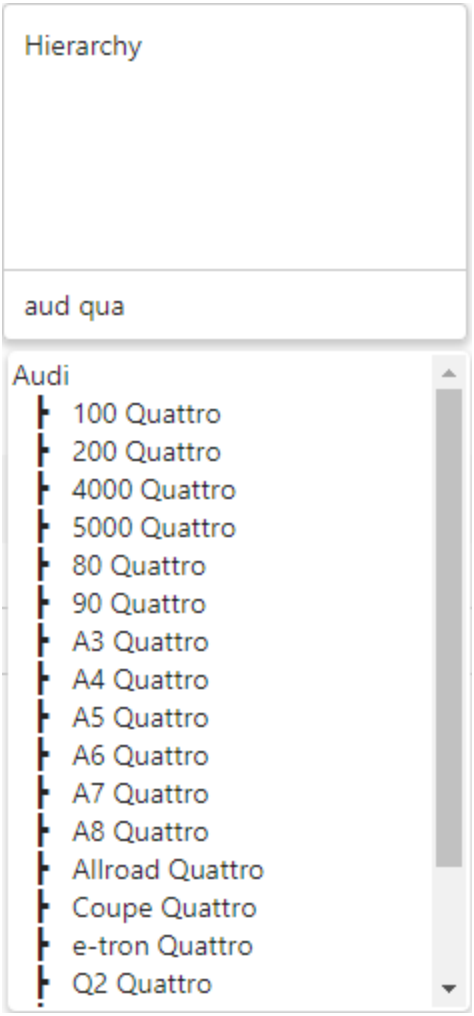

If the text typed in matches a child only, then the parent for the matching child is displayed with only the child that matches the search string. For example, if 'fox' is typed into the field, then the dropdown list will display the Audi and Volkswagen parents with the 'Fox' children listed below.



If the text typed in matches the parent and child, then the matching parents and/or children will display within the dropdown for selection. For example, if 'opel' is typed into the field, then the dropdown list will display the Buick parent with the Opel child beneath, and then the Opel parent will display with all of its available children below.



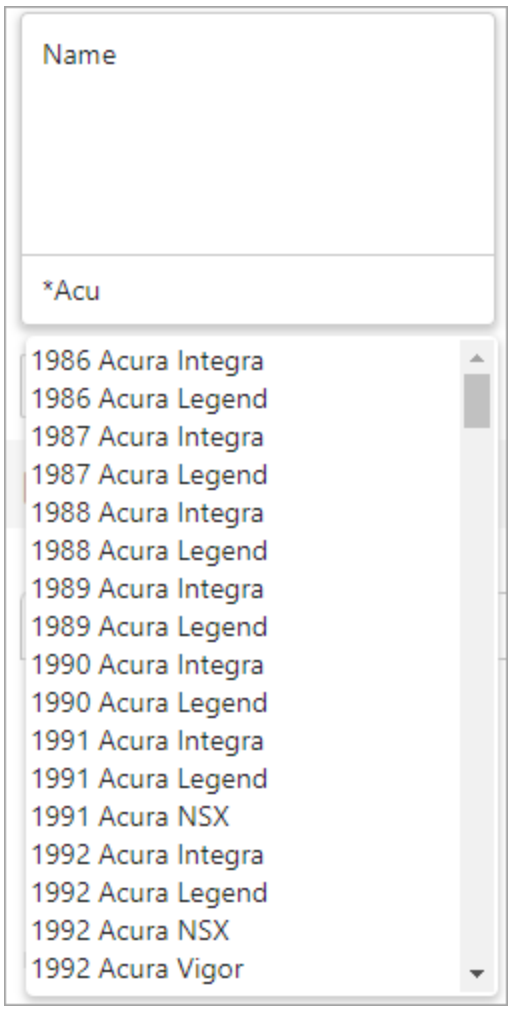

To search for a parent with a specific child, using the keyboard space bar, groupings of two or more characters can be typed into the field to narrow down the search. The keyboard space bar allows for the following text to conduct a 'contains' search for parents with specific children. For example, to display all the Audi models containing Quattro, the following can be entered into the parent / child field: 'aud [space bar] qua.'




## Using Name Search Card

The Name search card allows users to add a name criteria to their search. This section addresses functionality that is specific to the Name search card. For more information on the general functionality found within the search cards, refer to the **Using Search Cards** topic located in this documentation.

Within the Name search card, the typeahead field can be used to search for the name of the object that is intended for search, and the use of an asterisk will allow for a wildcard search. Only Names are searched, not IDs. The dropdown list displays criteria options alphabetically. For example, if '*Acu' is typed into the field, then the dropdown list displays the object name (of the object type that is defined in the design mode) containing 'Acu' alphabetically (as shown below).




# Using Search by Cards Screen with Data Onboarding Tasks

While the Search by Card Screen is designed to perform a search based on the valid combinations of Name, parent hierarchy, and attribute values, one use case for this Search by Card Screen in Onboarding task is to display the searched result object on Onboarding Comparison Screen by using Forward Screen Action button.

This topic describes how to navigate an object displayed in the search result table onto Onboarding Comparison Screen that is used in the Onboarding tasks.

## Prerequisites

This topic assumes the following:

- An understanding of how to configure the Forward Screen Action button as a toolbar component in the Search by Cards Screen. For more information, refer to the **Forward Screen Action Component** topic within the **Onboarding Comparison Screen** section of this guide.
- An understanding of how to perform a search in the Search by Cards Screen. Refer to the **Using Search by Card Screen Interface** topic for more information.

## Navigating to the Onboarding Comparison Screen

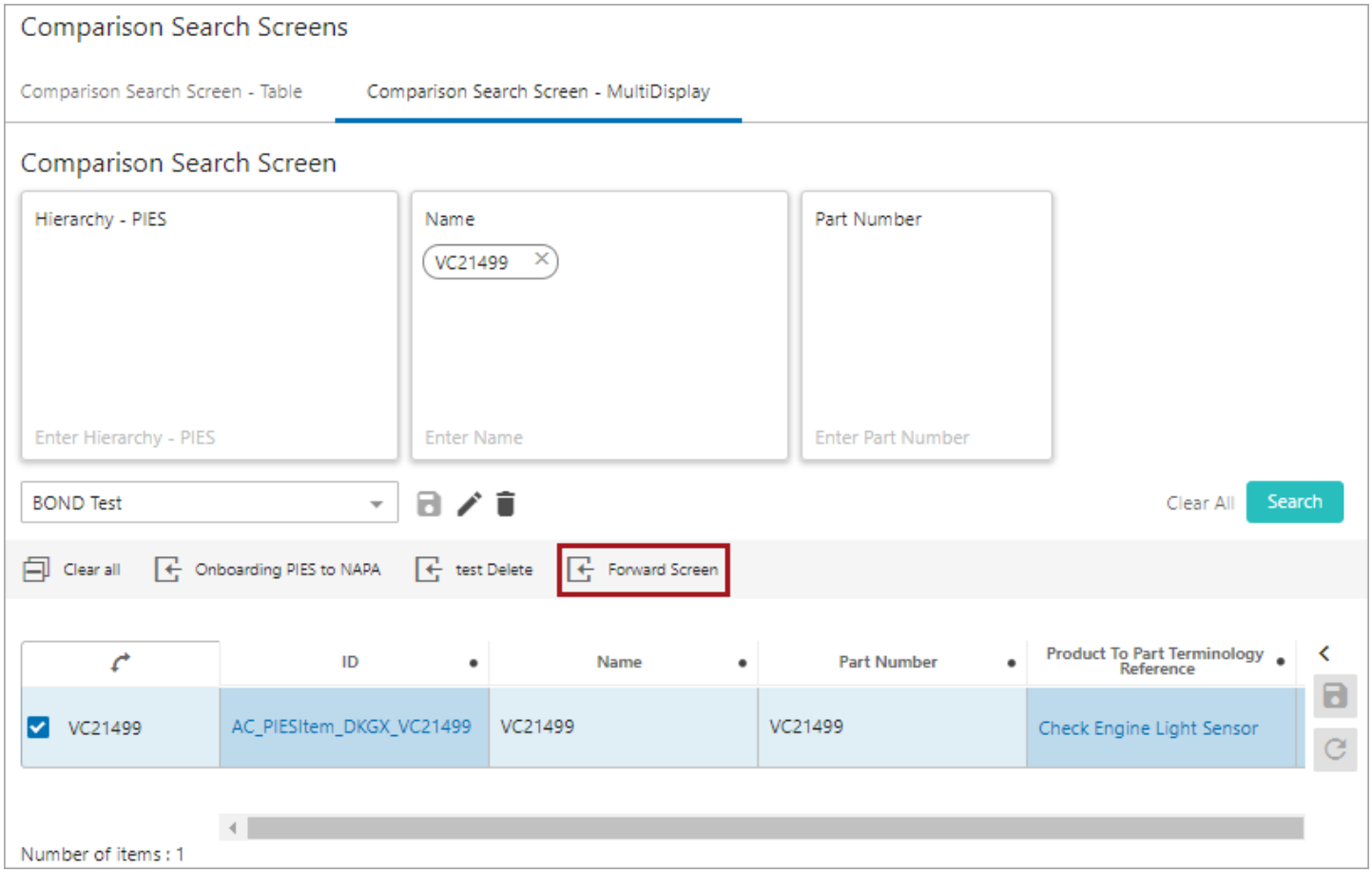
When accessing the Onboarding Comparison Screen via a Forward Screen Action button, the default screen behavior varies depending on whether or not the Target object exists. Also, the presence of multiple Target objects for the selected Source object does not allow to display the Onboarding Comparison Screen. If there exists no Target object, or if there exists multiple Target objects for the selected Source object, then the following error message will be displayed on the screen.

The Selected Mapping Configuration AC_PIESItem_DKGX_5700_OIL_SOURCE_03 failed unexpectedly Cannot load target for source: AC_PIESItem_DKGX_5700_OIL_SOURCE_03.

Follow these steps to access the Onboarding Comparison Screen from the Search by Cards screen:

1. Perform the search following the steps outlined in the **Using Search Cards** topic.
2. Select the required object from the search result that needs to be routed to the Onboarding Comparison Screen.
3. Click the **Forward Screen Action** button (

) that is available in the toolbar of the screen.



# STEP Path Component

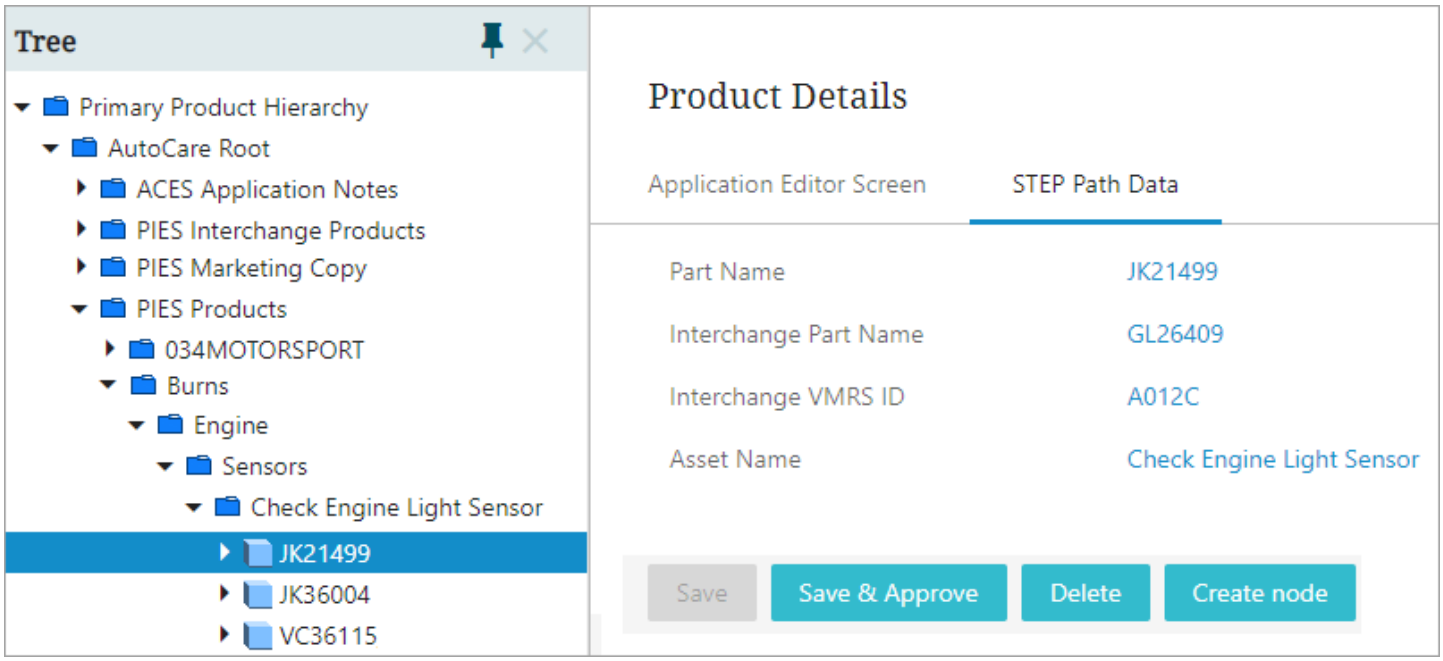
The STEP Path component can be added to a Node Editor component to retrieve data points from referenced objects based on the path that is defined within the STEP Path parameter. The premise behind this concept is that when a valid STEP Path is provided, the Node Editor can retrieve and display the ID, Name, attribute values, or references of any object related to the current object.

One significant benefit to using a STEP Path component is that users can view the desired data of the referenced object by standing on the current object screen without having to navigate to the referenced object.

For example, if the user is viewing object A but wants just the attribute value for object B, the STEP Path component can pull that attribute value into the Node Editor on the viewed object for quick reference.

Further, the user can also configure the path to access data points by following a series of objects that reference one another. For example, if object A is referencing object B and object B is referencing object C, then users can view the ID, Name, or attribute value of object C while viewing object A.

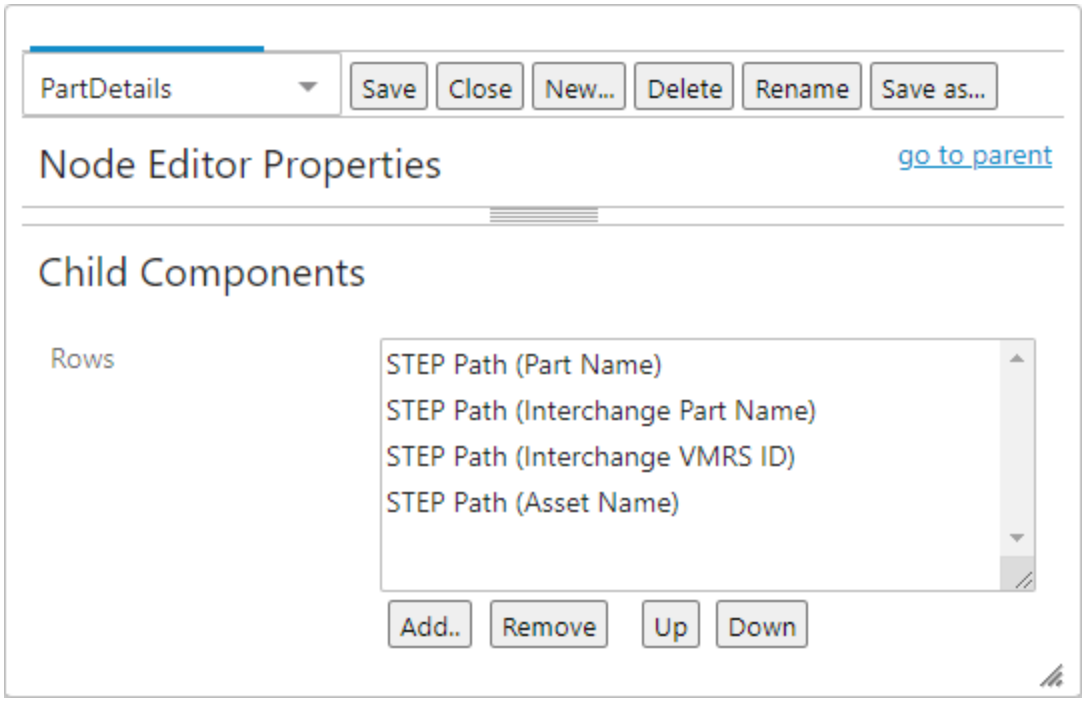
In the example below, the object JK21499 is referenced to an interchange part 'GL26409' and an asset 'Check Engine Light Sensor'. The screen has been configured to display the Name of the current object (Part Name), Name of the interchange part (Interchange Part Name), an attribute value belonging to the interchange part (Interchange VMRS ID), and the Name of the referenced asset (Asset Name) are displayed using STEP Path components within a Node Editor on a Node Details screen.




The defined STEP Path always begins by evaluating the current object to follow the defined path, thereby retrieving information that lies at the end of the path. Though the information retrieved and displayed on the current object is read-only, the displayed value will be hyperlink text. Clicking the hyperlink will navigate the user to the object from which the displayed value has been retrieved.

For example, If the current object is configured to display an attribute value of a referenced object, clicking that value will follow the reference type defined in the validation path and displays the screen configured for the referenced object where the attribute with that value exists.

The screenshot below displays the Web UI designer settings used to enable the example given above.




**Note:** The STEP Path component does not retrieve and display any assets. It can only display the ID / Name / attribute values / references of the asset.

## Prerequisites

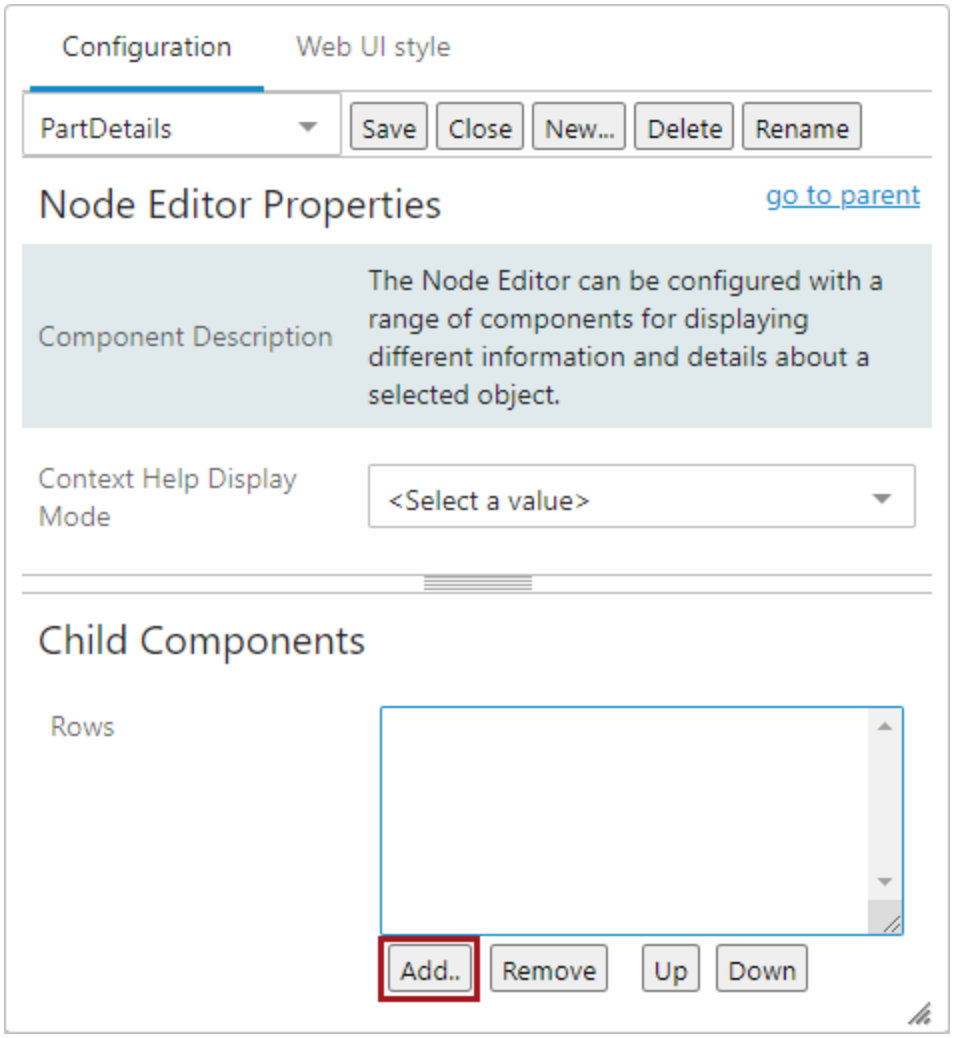
It is expected that anyone configuring the STEP Path component is familiar with the Web UI Designer, as basic concepts for working with the designer are not covered in this section. In addition, users must have appropriate privileges to access the designer. Additional information can be found in the **Designer Access** section of the **Web User Interfaces** documentation.

## Adding a STEP Path component to a Node Editor

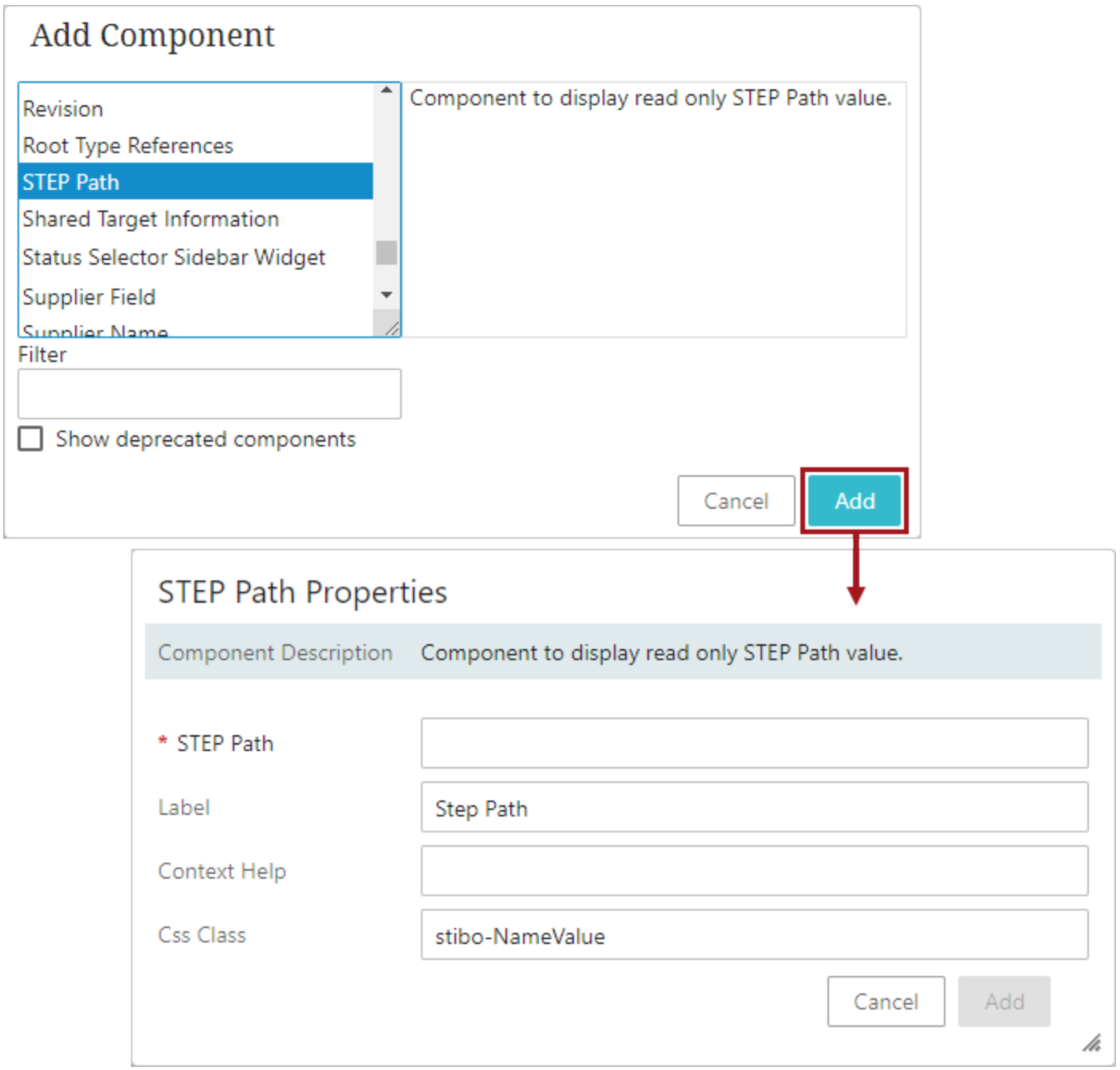
STEP Path components are added and configured in the Child Components section of the Node Editor Properties. STEP Paths can be added and removed using the buttons below the Rows field, and the data points they retrieve from the referenced objects will display on the screen based on how they are ordered in the Rows field. They can be reordered by selecting the relevant STEP Path and clicking the Up and Down buttons.

Steps for adding a STEP Path component to a Node Editor, as well as all parameters used to configure a STEP Path component, are described below.

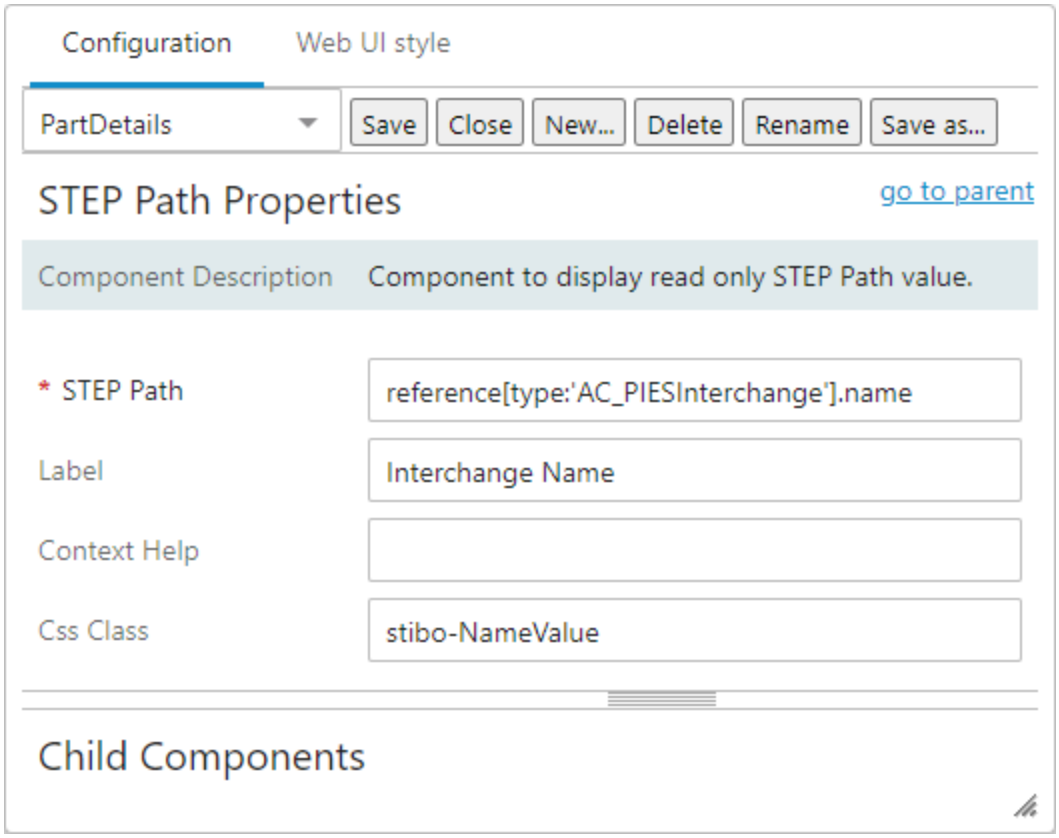
1. In the designer, select your **Node Details screen**, navigate to your **Node Editor component**, and the Node Editor Properties will display. Click the **Add** button.




2. In the Add Component dialog, select the STEP Path component, click the **Add** button, and the STEP Path Properties dialog will display.



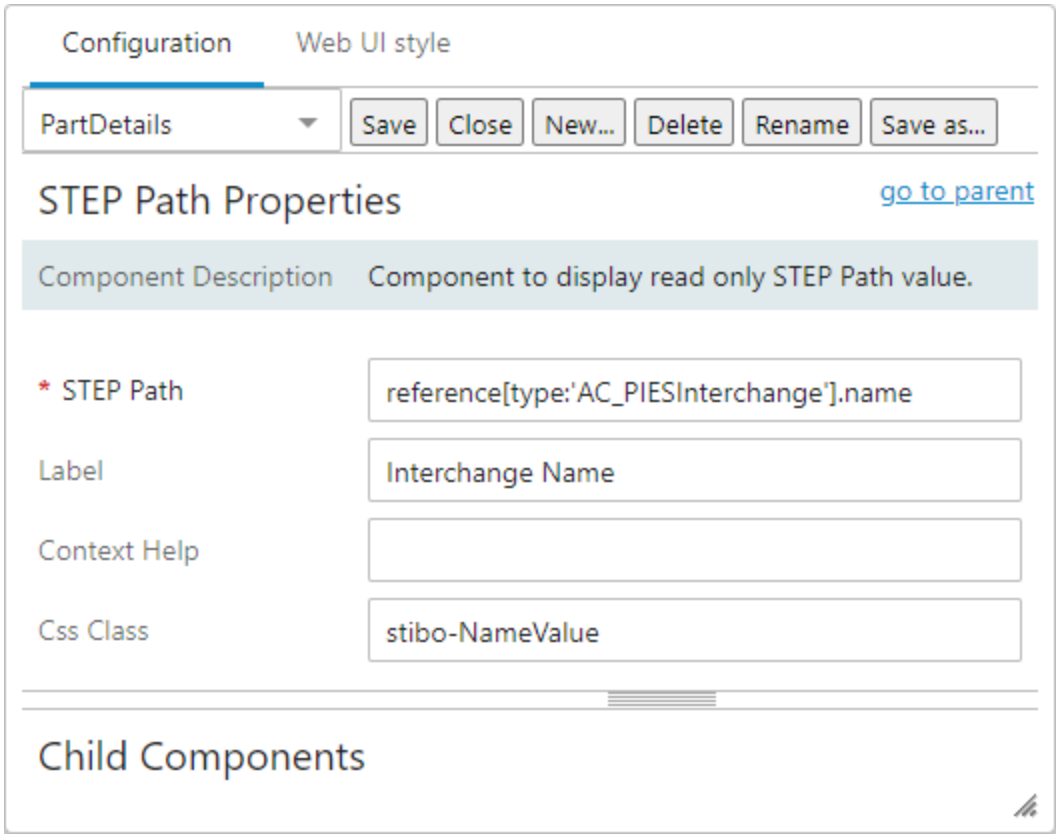

3. The only mandatory parameter is **STEP Path**. In the example below, the STEP Path parameter is populated to retrieve the interchange part's name.




4. Once the appropriate validation path is defined, continue configuring the non-required parameters that are available in the STEP Path Properties dialog.
5. Click the **Add** button, the dialog will close, and the newly added STEP Path component will display as the last row of the Node Editor Child Components Rows.
6. Optionally, select the newly created STEP Path, and then use the **Up** and **Down** buttons to order the display of the newly added STEP Path component as desired.
7. Click the **Save** and **Close** buttons to exit the designer.

## Configuring the STEP Path component

The STEP Path component enables some configuration, and a description of each of the configuration parameters can be found below.




- **STEP Path:** The only mandatory parameter for this component. In this field, the user can define a validation path for retrieving desired data points through the use of the syntax of child, reference, reference by, ID, name, text, parent, and attribute elements. The uses of each type of element can be found in the 'STEP Path Elements' section below.
- **Label:** Override the data point name by adding new text that will display in its place. The default label for the STEP Path component is 'STEP Path.'
- **Context Help:** Enter help text to display when a user hovers over the displayed data point in the screen. This parameter can provide information to end users about how the data point is being retrieved and displayed on the screen.
- **Css Class:** Enter a css-style name to be set as the primary style for this component. The default css-style name is 'stibo-NameValue.'

## STEP Path Elements

Users can form a syntax for following elements to define a STEP Path. The combination of elements depends on what data must be retrieved from the directly or indirectly related object(s).

| Element | Function |
|---|---|
| **id** | This element retrieves the ID of the defined object. |
| **name** | This element retrieves the Name of the defined object.<br><br>For example:<br><br>reference[type:'AC_PIESInterchange'].name |
| **attribute[id:'[id]']** | This element helps to retrieve the attribute value(s). Further configuration is required to determine the attribute. Within the **attribute[id:'[id]']** element, the user must replace the string [id] with the required attribute ID. For example:<br><br>attribute[id:'AC_ACESBodyNumDoors']. |
| **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. |
| **child[objecttype:'[id]']** | This element allows users to retrieve data from the child of any object. 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. Within the **child [objecttype:'[id]']** element, the user must replace the string **[id]** with the actual ID of the child's object type. This element should always be accompanied by **id** / **name** / **attribute[id:' [id]']** / **reference[type:'[id]']** elements.<br><br>In the example below, the **child[objecttype:'[id]']** and **id** elements are used to retrieve the ID of the current object's child that belongs to the 'AC_ACESApplication' object type:<br><br>child[objecttype:'AC_ACESApplication'].id |
| **reference[type:'[id]']** | This element retrieves the data from the references of the current 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. Within the **reference[type:'[id]']** element, the |

| Element | Function |
|---|---|
| | user must replace the string **[id]** with the actual ID of the reference type. This element can always be accompanied by **id** / **name** / **attribute[id:'[id]']** / **reference[type:'[id]']** elements.<br><br>In the example below, the **reference[type:'[id]']** and **id** elements are used to retrieve the ID of the target object that is referenced via 'AC_PIESInterchange' reference type:<br><br>reference[type:'AC_PIESInterchange'].id |
| **referenceBy[type:'[id]']** | This element retrieves the data from the source object when a current object is a target object of the configured reference. Further configuration is required to define the reference type and must define what value of the defined 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.<br><br>In the example below, the **referenceBy[type:'[id]']** and **id** elements are used to retrieve the ID of the source object that is referenced via 'OWNPartToPIESPart' reference type:<br><br>referencedBy[type:'OWNPartToPIESPart'].id |
| **text['[text]']** | This element lets the user enter a static text of the user's choice. |

For example, if the user needs to retrieve the value from ACES Body Num Doors attribute (ID = AC_ ACESBodyNumDoors) stored at the child level object (object type = AC_ACESApplication) of the current object, the STEP path is configured as:

```
child[objecttype:'AC_ACESApplication'].attribute[id:'AC_ACESBodyNumDoors']
```

This Source STEP Path is determined by:

- Identifying the current object.
- Examining all children of the current object with object type 'AC_ACESApplication.'
- Evaluating the child object's 'ACES Body Num Doors' (AC_ACESBodyNumDoors) attribute value.