STEP AUTOMOTIVE

REFERENCE GUIDE

STEP Trailblazer 8.3

Table of Contents

STEP Automotive Reference Guide Introduction

This guide describes specific 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 Export:** Format is not versioned (generic exporter not specific to any standard).

AutoCare

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

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

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

NAPA

- **Application Export:** Format is not versioned
- **Application Import:** Format is not versioned
- **Asset Reference Export:** Format is not versioned
- **MPCC / Attribute Export:** Format is not versioned
- **MPCC / Attribute Import:** Format is not versioned
- **Interchange Export:** Format is not versioned
- **Interchange Import:** Format is not versioned
- **Translation Import:** Format is not versioned
- **Valid Vehicles Import:** Format is not versioned

TecDoc

- **Reference Data Import:** TAF 2.4
- **Supplier Data Export:** TAF 2.4
- **Supplier Data Import:** TAF 2.4

Importing Automotive Data

The Automotive solution provides extensive import capabilities.

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

- 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
- Import Validation Rules
- Validation Error Handling

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

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

Import Framework

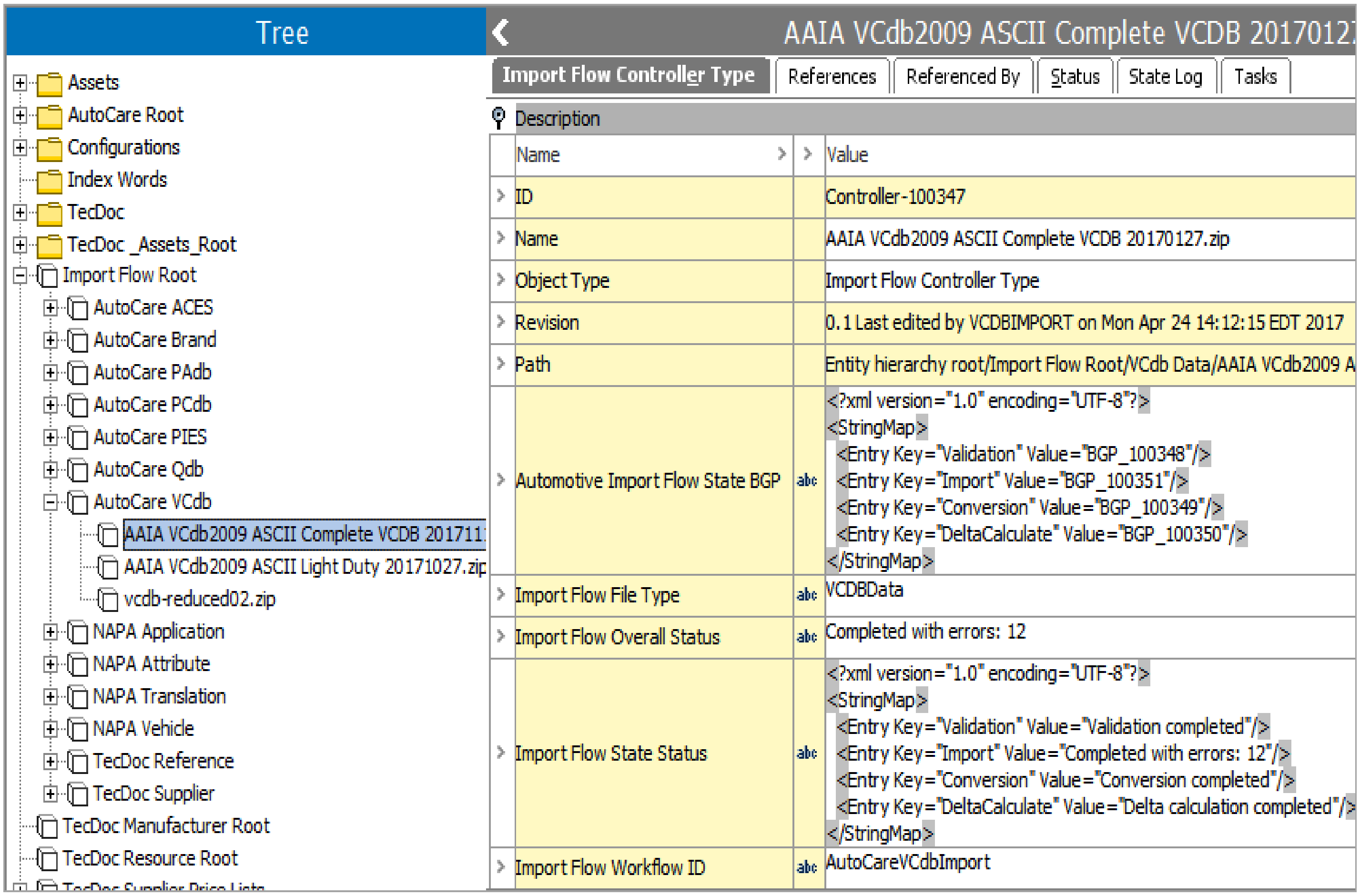
The intention of the import aspect of the Automotive core solution is to provide out-of-the-box importers for the various Automotive standards, for which each customer can then apply their own validations, business processes, and data management procedures. To do this successfully, it is crucial to understand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

This section details specifics of the Import Framework, which is applicable to all Automotive importers across all standards.

Overview

When a file is picked up from the hotfolder of an integration endpoint, the endpoint creates an entity in STEP that represents the file. This entity object is called the 'controller' and contains some basic data about the file and the file's status in the workflow.

For example:



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

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

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

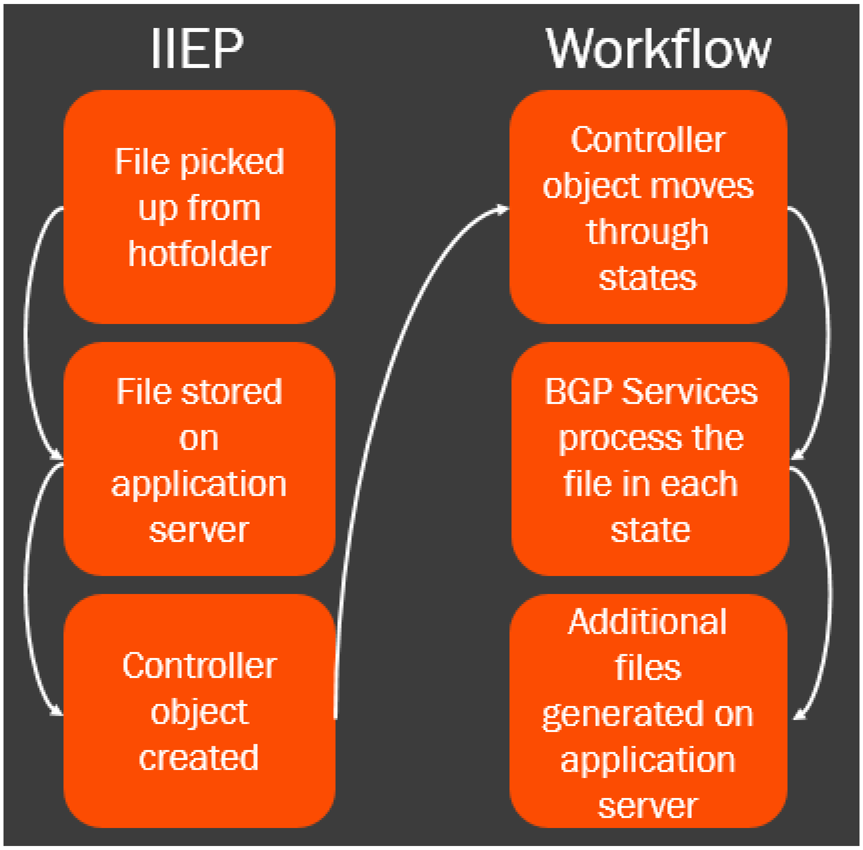
Note: All of the information displayed on the controller entity is also displayed within the workflow and controller screens in Web UI, which are discussed in the **Quick Start for Users** section of the **Automotive Quick Start Guide**.

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.

The background process service that runs as part of the Import state allows for selection of business rules that function in the same way as rules run as part of a standard import, meaning that they act on the objects being imported.

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

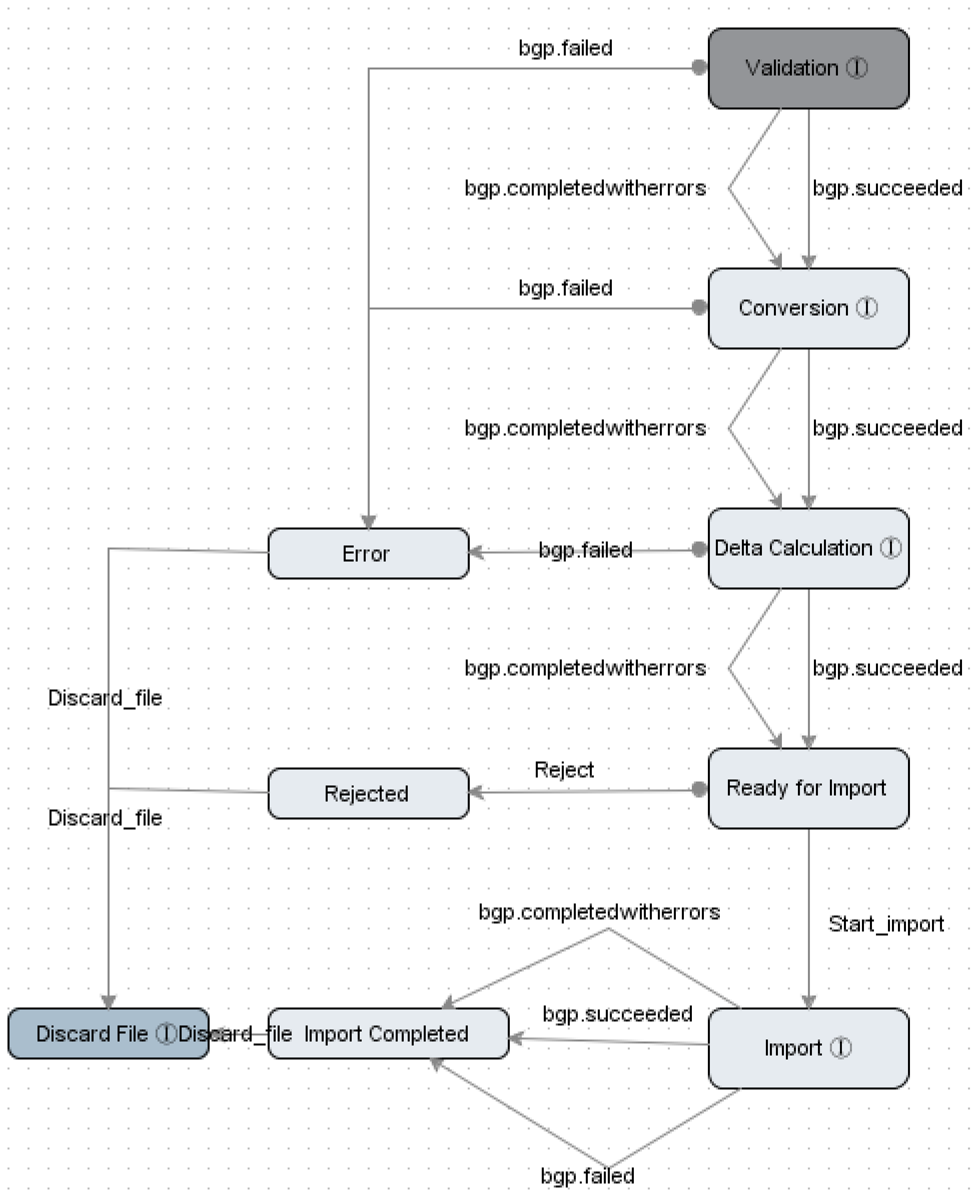

IIEP
Workflow
File picked up from hotfolder
File stored on application server
Controller object created
Controller object moves through states
BGP Services process the file in each state
Additional files generated on application server

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.

Default Workflow States and Functions

The default workflow and processing that occurs in each state is described below.



Validation State

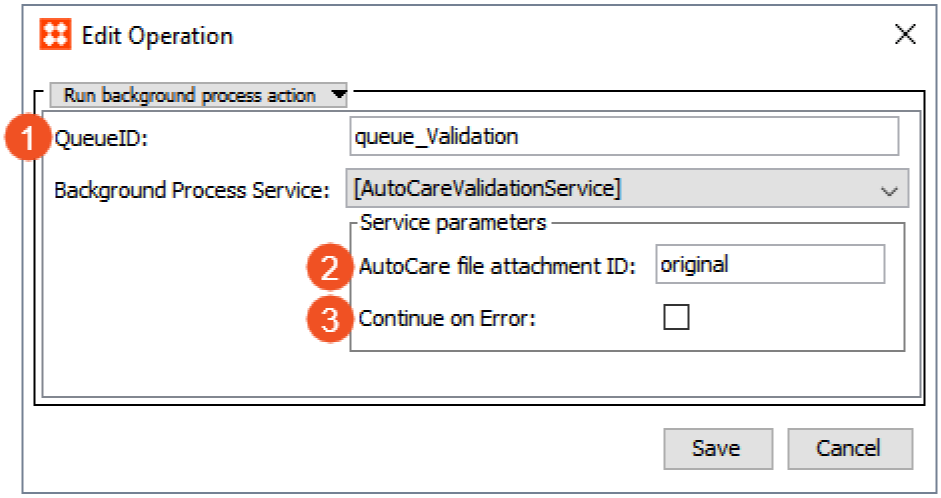
The validation state does some basic schema validation of the file. If the automotive standard (i.e., AutoCare, TecDoc, NAPA) supplies an XSD, the file format is validated against the XSD. If an XSD is not provided by the standard, the file is loosely validated to ensure the format conforms to what is expected for the file type. For more information on the validation rules for the specific import types, see the **Import Validation Rules** section of this guide.

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. The AutoCare validation service is shown as an example below.

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. All parameters are described below.



Edit Operation

Run background process action

1 QueueID: queue_Validation

Background Process Service: [AutoCareValidationService]

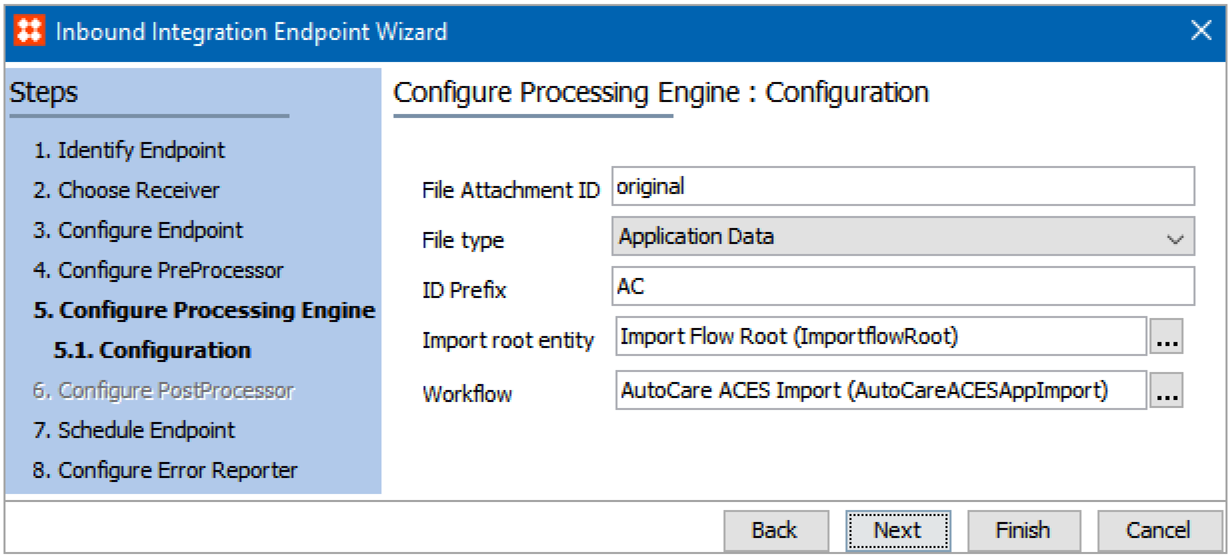
Service parameters

2 AutoCare file attachment ID: original

3 Continue on Error: □

1. **QueueID:** Required parameter for all background process actions that specifies the queue in which the background process should run, which defaults to 'queue_Validation'. 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.
2. **[Standard] file attachment ID:** Required parameter in all validation services that defaults to 'original' and forms the relationship between the file supplied to the endpoint and the file that the workflow is going to process, and is prefixed to the file name for the workflow attachment.

   Important: The [Standard] file attachment ID entry in the validation service must match the File Attachment ID value in the corresponding endpoint configuration. In this example the 'AutoCare file attachment ID:' value is 'original,' and this can be viewed in both the Edit Operation dialog (shown above) and the Inbound Integration Endpoint Wizard (shown below).


Inbound Integration Endpoint Wizard
Steps
1. Identify Endpoint
2. Choose Receiver
3. Configure Endpoint
4. Configure PreProcessor
5. Configure Processing Engine
5.1. Configuration
6. Configure PostProcessor
7. Schedule Endpoint
8. Configure Error Reporter
Back
Next
Finish
Cancel

Configure Processing Engine : Configuration

File Attachment ID: original

File type: Application Data

ID Prefix: AC

Import root entity: Import Flow Root (ImportflowRoot)

Workflow: AutoCare ACES Import (AutoCareACESAppImport)

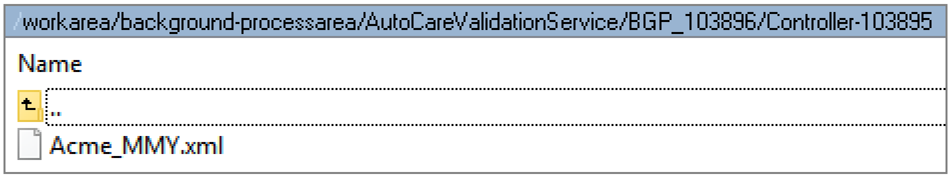
3. **Continue on error:** Specifies whether or not the file can continue to process after encountering some "allowable" data errors. Please see the **Automotive Quick Start Guide** for additional information and considerations relevant for this setting.

Validation State TecDoc Specific Parameter

- **Reference data Asset ID:** Required parameter available only in the TecDoc validation service. When importing a reference data file, the file itself is stored in the selected asset. When importing a supplier data file, the supplier data file is validated against the reference data file stored in the selected asset.

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 for example:



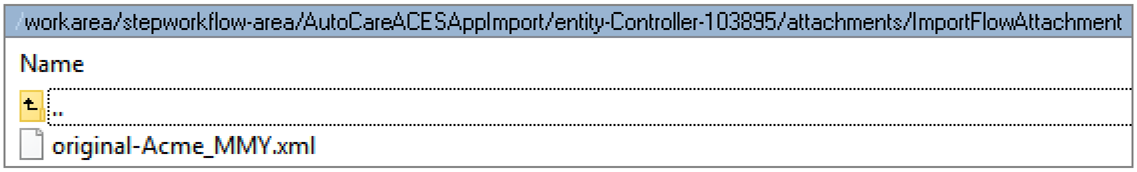
/workarea/background-processarea/AutoCareValidationService/BGP_103896/Controller-103895

Name

..

Acme_MMY.xml

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



/workarea/stepworkflow-area/AutoCareACESAppImport/entity-Controller-103895/attachments/ImportFlowAttachment

Name

..

original-Acme_MMY.xml

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

Validation State Results

The following are possible validation state results:

- **Fails:** 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). The controller will remain in the Error state until acted on by a user.

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

- **Successful:** 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.
- **Successful with Errors:** It is possible for a file to pass validation, 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. 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 'Determine Error Handling for each Import' section 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.

Conversion State

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

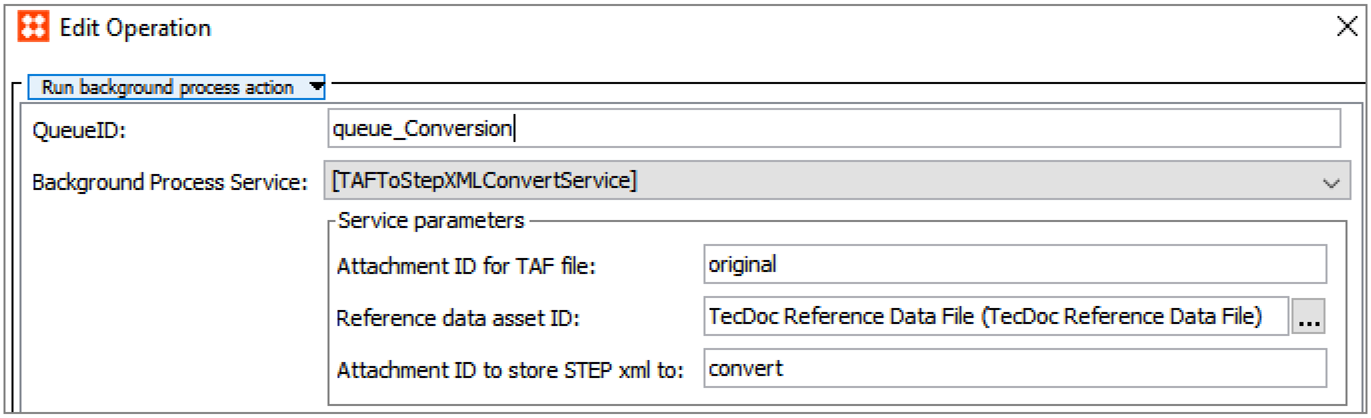
1. By converting to STEPXML, the import can be carried out using standard STEP import functionality.
2. By converting to multiple standalone files rather than a single large file, there is a performance gain in that some files can be imported in parallel.

Conversion State Common Parameters

The conversion state includes one business action (**Run background process action)** by default, which runs the conversion service as a background process. Each standard has its own conversion service and the parameters included in each are described below, beginning with those that are common to all services.

Note: Within AutoCare, there are two conversion services; one general, and one specific to ACES files.

For example, the TecDoc conversion service is shown in the screenshot below.



Edit Operation

Run background process action

QueueID: queue_Conversion

Background Process Service: [TAFToStepXMLConvertService]

Service parameters

Attachment ID for TAF file: original

Reference data asset ID: TecDoc Reference Data File (TecDoc Reference Data File)

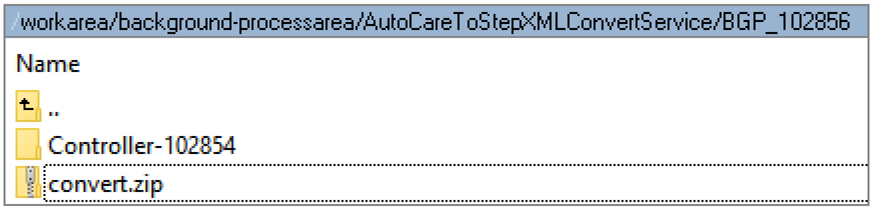
Attachment ID to store STEP xml to: convert

- **QueueID:** Required parameter for all background process actions that specifies the queue in which the background process should run, which defaults to 'queue_Conversion'. Note that this parameter is specified for the action itself, not the particular service within the action, though the outcome is the same as each action runs only a single service.
- **[Standard] file attachment ID /** Attachment ID for [Standard] 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).
- **STEP XML attachment ID /** Attachment ID to store STEP xml into**:** 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.
- **ACES Parameters:** The AutoCare ACES conversion service is unique and contains a number of additional parameters, specifically: **ACES Default FULL Import handling**, **ACES Default SUPPLIER Import handling**, **ACES Default UPDATE Import handling**, and **Regular Expression for parsing Supplier from file name**. The functionality of each of these parameters is described in the **Automotive Quick Start Guide**.
- **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).

Conversion State Function Details

As the conversion service runs a background process (BGP), a corresponding BGP folder is created on the application server at /workarea/background-processarea/[Standard]toStepXMLConvertService. The BGP folder contains a sub-folder whose name matches the STEP ID of the controller entity that is moving through the workflow and contains the originally loaded file, the same as is seen in the validation service. The conversion service also writes an additional file, which is the output of the conversion process, (e.g., a zipped file with the name specified in the STEP XML 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:



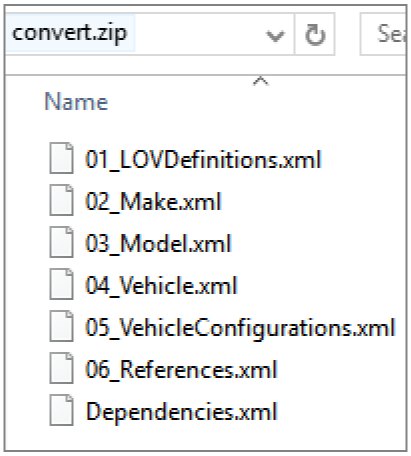
/workarea/background-processarea/AutoCareToStepXMLConvertService/BGP_102856

Name

..

Controller-102854

convert.zip



convert.zip

Name

01_LOVDefinitions.xml

02_Make.xml

03_Model.xml

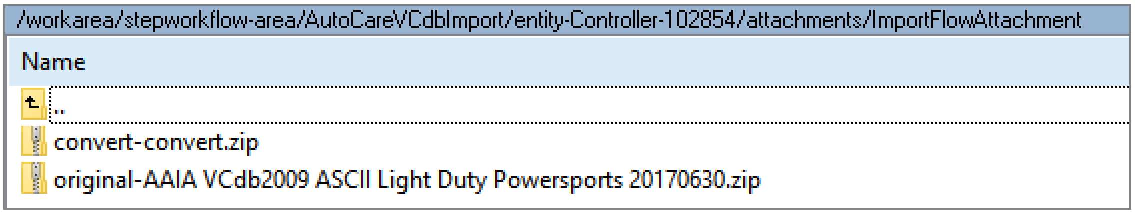
04_Vehicle.xml

05_VehicleConfigurations.xml

06_References.xml

Dependencies.xml

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:



/workarea/stepworkflow-area/AutoCareVCdbImport/entity-Controller-102854/attachments/ImportFlowAttachment

Name

..

convert-convert.zip

original-AAIA VCdb2009 ASCII Light Duty Powersports 20170630.zip

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:

- **Fails:** 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.
- **Successful:** 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.

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

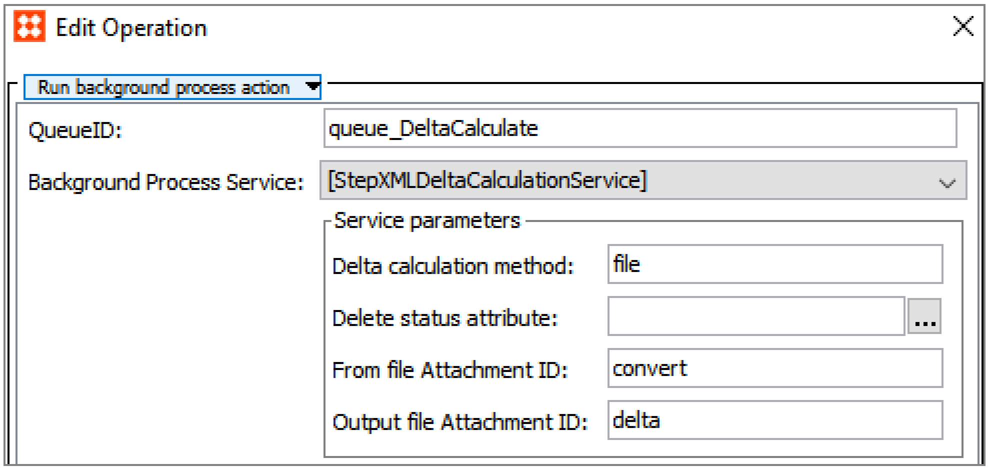
Delta Calculation State

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

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

Delta Calculation State Common Parameters

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



QueueID: queue_DeltaCalculate

Background Process Service: [StepXMLDeltaCalculationService]

Service parameters

Delta calculation method: file

Delete status attribute:

From file Attachment ID: convert

Output file Attachment ID: delta

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

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

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

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

- **Delete status attribute:** Optional parameter to select an attribute to store the indication for deletion as a result of the import. If not set, deletions will not be processed by the importer.

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

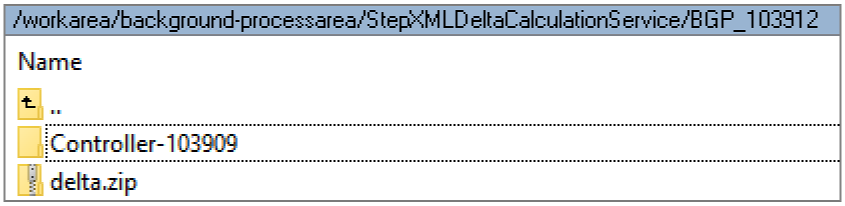
  For additional information and considerations for setting this parameter, see the **5. Update Delete Status Attribute and Delta Calculation Method in Import Workflows** section of the **Quick Start Setup for Admins**.

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

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 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:



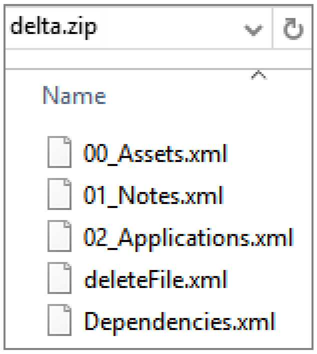
/workarea/background-processarea/StepXMLDeltaCalculationService/BGP_103912

Name

..

Controller-103909

delta.zip



delta.zip

Name

00_Assets.xml

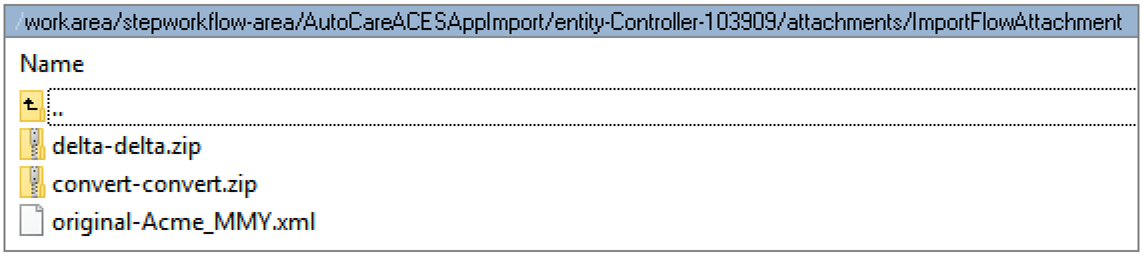
01_Notes.xml

02_Applications.xml

deleteFile.xml

Dependencies.xml

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:



/workarea/stepworkflow-area/AutoCareACESAppImport/entity-Controller-103909/attachments/ImportFlowAttachment

Name

..

delta-delta.zip

convert-convert.zip

original-Acme_MMY.xml

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

Setup Objects and Delta Calculations

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

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

Delta Calculation State Results

- **Fails:** 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.
- **Successful:** 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.
- **Successful 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 simply transition the file to the Discard File state, which contains a business rule that automatically removes the controller from the workflow so that it is no longer displayed in the workflow tasks in the Web UI.

Ready for Import State

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

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

Rejected State

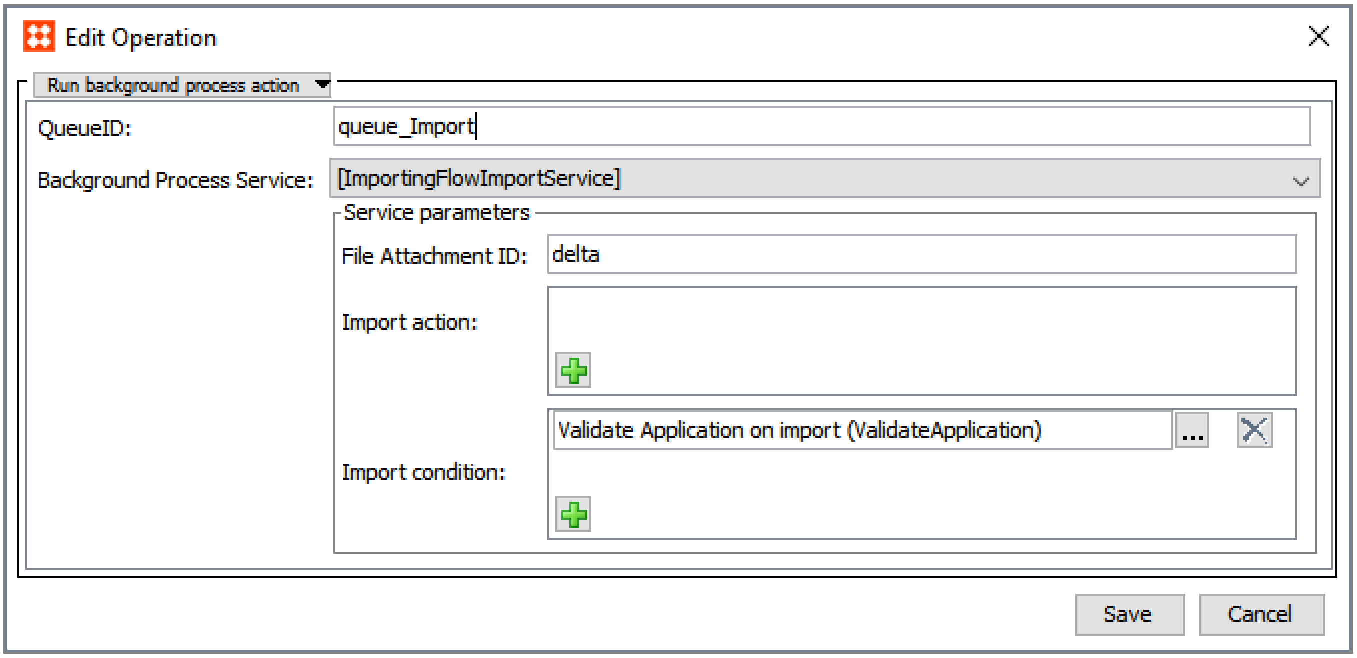
When a file has been rejected by a user, the controller is moved to the Rejected state. This indicates that the file has successfully completed all pre-import processing, but has been deemed unacceptable for import by a user, usually due to data issues. A rejected file cannot be imported unless it is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

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.



Edit Operation

Run background process action

QueueID: queue_Import

Background Process Service: [ImportingFlowImportService]

Service parameters

File Attachment ID: delta

Import action:

Import condition: Validate Application on import (ValidateApplication)

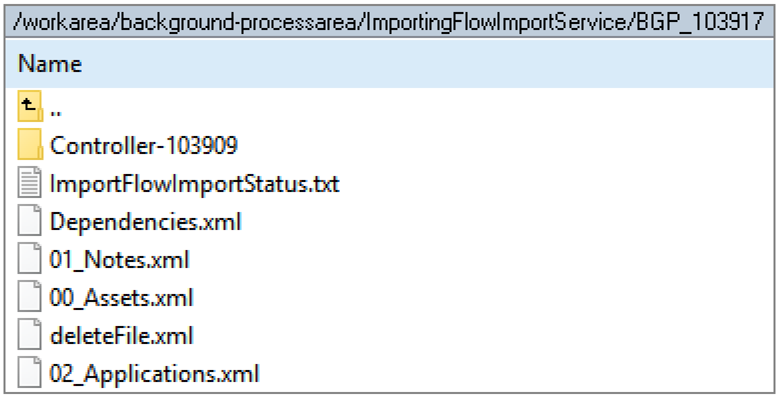
- **QueueID:** Required parameter for all background process actions that specifies the queue in which the background process should run, which defaults to 'queue_Import'. Note that this parameter is specified for the action itself, not the particular service within the action, though the outcome is 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.
- **Import action:** Optional parameter allowing administrators to select one or more business actions to be run as part of the import. By default, there are no actions included but customers may add any that they wish.
- **Import condition:** Optional parameter allowing administrators to select one of more business conditions to be run as part of the import. Importers that manage applications / linkages have one default condition applied (Validate application on import). This condition prevents applications from being imported if they do not contain a vehicle / assembly and part type in STEP.

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.

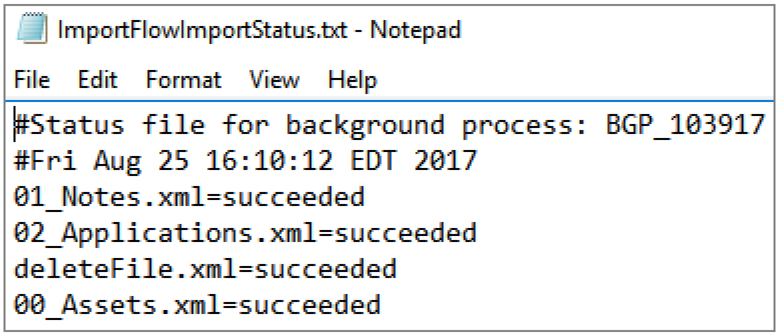
Import State Function Details

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:



```
/workarea/background-processarea/ImportingFlowImportService/BGP_103917
..
Controller-103909
ImportFlowImportStatus.txt
Dependencies.xml
01_Notes.xml
00_Assets.xml
deleteFile.xml
02_Applications.xml
```



```
#Status file for background process: BGP_103917
#Fri Aug 25 16:10:12 EDT 2017
01_Notes.xml=succeeded
02_Applications.xml=succeeded
deleteFile.xml=succeeded
00_Assets.xml=succeeded
```

Import State Results

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

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

For example, to set the AutoCare validation processes to delete after 48 hours, specify:
AutoDeleteBackgroundProcesses.AgeInHours.AutoCareValidationService = 48

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

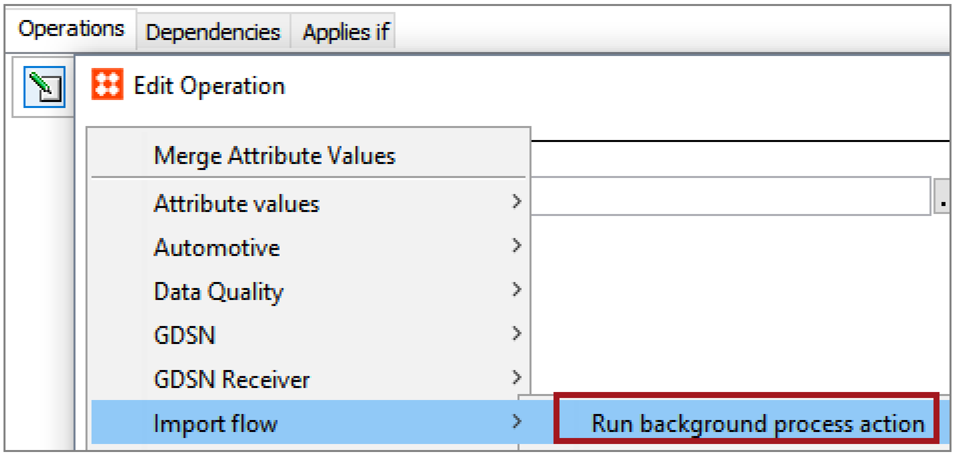
Modifying Import Framework

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

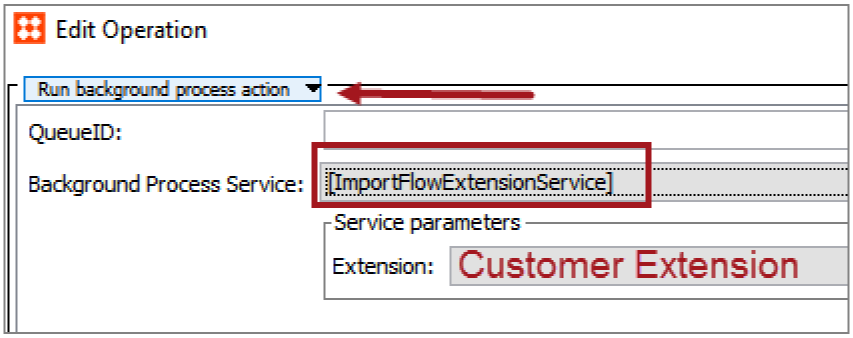
There are many out of the box business rules that can be added for advanced functionality in the importers. For more information, see the **Automotive Business Rule Plugins** section within this guide.

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.



Run background process action

QueueID:

Background Process Service: [ImportFlowExtensionService]

Service parameters

Extension: Customer Extension

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 **STEP 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.

This topic addresses the part and application ID structures used for the AutoCare, NAPA, and TecDoc standards.

Prerequisites

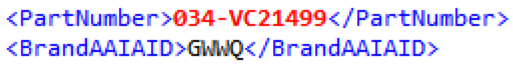
The ID structures described within this section are dependent upon the completion of the setup as described in the **2. Run Easy Setup of Import Flow Process** section of the **Automotive Quick Start Guide**.

Part ID and Application ID Structure

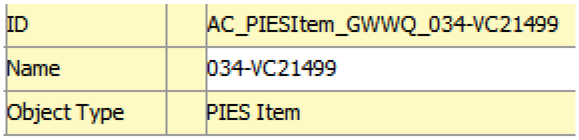
The part ID and application ID structures for each of the standards are described below. In the examples below, the import file information is displayed (when applicable) above a screenshot of the workbench ID, Name and Object Type fields.

AutoCare

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

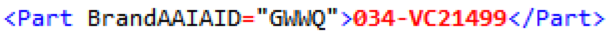


```
<PartNumber>034-VC21499</PartNumber>
<BrandAAIAID>GWWQ</BrandAAIAID>
```

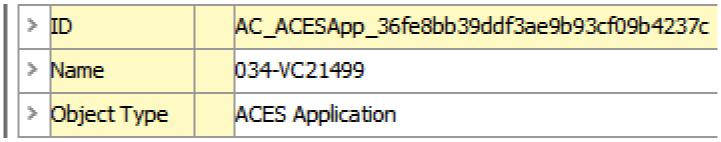


ID	AC_PIESItem_GWWQ_034-VC21499
Name	034-VC21499
Object Type	PIES Item

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



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

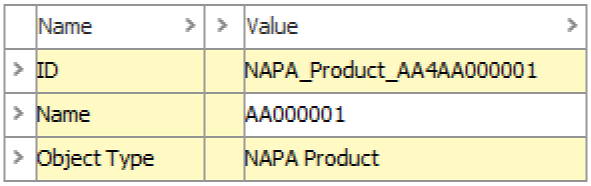


ID	AC_ACESApp_36fe8bb39ddf3ae9b93cf09b4237c
Name	034-VC21499
Object Type	ACES Application

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

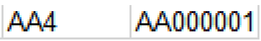
- **Part ID** = NAPA_Product_[ProductLine][PartNumber]



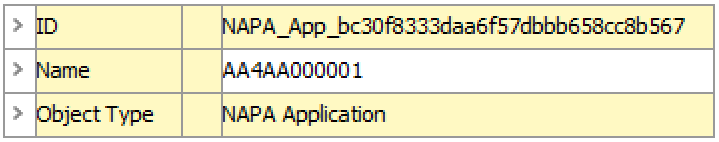
Name	Value
ID	NAPA_Product_AA4AA000001
Name	AA000001
Object Type	NAPA Product

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** = A_[hash function]



AA4	AA000001

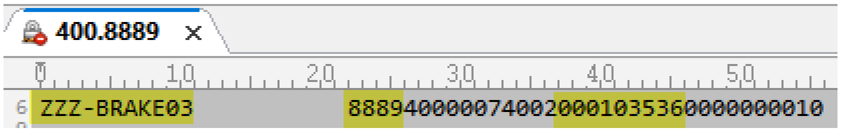


>	ID	NAPA_App_bc30f8333daa6f57dbbb658cc8b567
>	Name	AA4AA000001
>	Object Type	NAPA Application

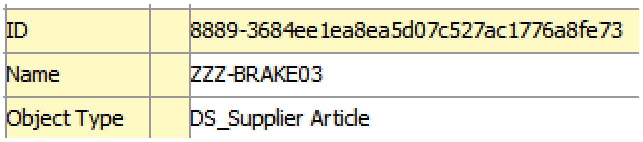
Note: Submitting the above application will search for the above part, with the ID as shown. If not found, an error will be reported and the application will not be created.

TecDoc

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

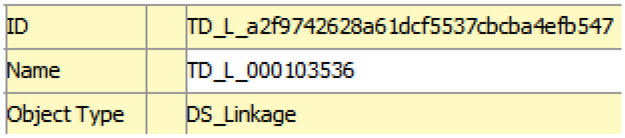


```
6 ZZZ-BRAKE03           8889400000740020001035360000000010
```



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

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



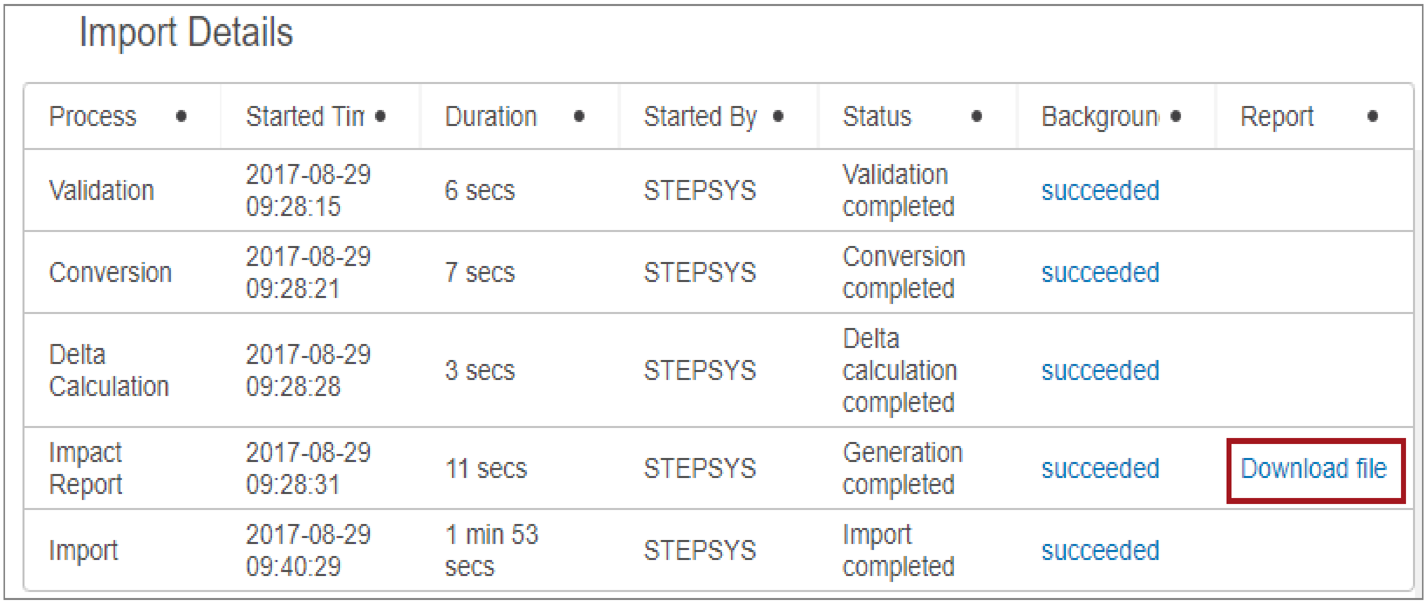
ID	TD_L_a2f9742628a61dcf5537cbcba4efb547
Name	TD_L_000103536
Object Type	DS_Linkage

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 of this guide), 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 (described above), but there is some built-in functionality to be aware of to enable users to access the report file in the Web UI.

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 as it was stored with. For the file to appear in the Report column in the Web UI, the attachmentID has to be equal to the ID of the workflow state in which the file should appear. See the ImportFlowExtensionContext interface in the Javadoc for additional information.

This of course requires that the Importflow State Report header is present in the associated workflow screen, as described in the **STEP Automotive Quick Start Guide STEP** found within the **Solution Enablement** section of **STEP Online Help**.

Import Validation Rules

When importing data into STEP for each of the automotive standards, it is important to understand the validation rules. This section describes the validation rules for the different file types within the standards listed below:

- AutoCare Import Validation Rules
- TecDoc Import Validation Rules

AutoCare Import Validation Rules

When importing AutoCare data into STEP, it is important to understand the different STEP validation rules.

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

This section addresses the STEP validation rules for the AutoCare file types listed 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 (Part Classification Database)
4. PAdb (Product Attribute Database)
5. VCdb (Vehicle Configuration Database)
6. PIES (Product Information Exchange Standards)
7. ACES (Aftermarket Catalog Enhanced Standards)

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

Qdb Import Validation Rules

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

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

Accepted File Extension: .zip

Validation Rule	Occurs	When Validation Fails	Failed Validation Message Example
All tables present	Validation state	BGP ends in an error triggering the import to enter the Error state.	The required table '[Required Table Name]' is missing in the ZIP archive or is empty.
Mandatory values present	Validation state	BGP ends in an error triggering the import to enter the Error state.	Table '[Table Name]' contains a row that misses value for the required column '[Column Name]' at line 3.
Correct number of columns (If last column is optional, then the last column can be missing without affecting validation.)	Validation state	BGP ends in an error triggering the import to enter the Error state.	Table '[Table Name]' contains a row with wrong number of columns at line 2.
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 '[Table Name]' 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 '[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. However, it does not detail the file type's data rules per the AutoCare Association. This information can be found at www.autocare.org.

For information on the supported versions, see the **Supported Versions and Formats** topic withing 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.
References between tables	Validation state	BGP ends in an error triggering the import to enter the Error state.	Table '[Table Name]' contains a row where the 'Column Name' 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 '[Table Name]' 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 '[Table Name]' contains an empty row at line 3.

PCdb Import Validation Rules

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

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

PAdb Import Validation Rules

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

For information on the supported versions, see the **Supported Versions and Formats** topic withing 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].
References between tables. (PAID, UoMList, UOMCode, MetaID)	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. or [Field Name] has a value [Value] in [File Name 1] but does not have a matching [FieldName] value in [File Name 2].
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	Warning is displayed.	Existing version '2015-09-25' is newer that

Validation Rule	Occurs	When Validation Fails	Failed Validation Message Example
	state		provided '2014-11-28.'
Empty line	Validation state	Warning is displayed.	Table 'VehicleToBedConfig' contains an empty row at line 3.
Part Type existence	Import state	All attribute links to PT skipped.	Line 10, Classification 2: No parent specified for new classification with ID 'AC_ PartTerminology_99999' Line 10, Classification 2: The classification with ID 'AC_PartTerminology_99999' was skipped.

VCdb Import Validation Rules

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

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

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

Accepted File Extension: .zip

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

PIES Import Validation Rules

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

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

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

Accepted File Extension: .xml

Validation Rule	Occurs	When Validation Fails	Failed Validation Message Example
PIES 6.5 XSD validation	Validation state	Validation fails	Error while validating file against XSD: cvc-complex-type.2.4.a: Invalid content was found starting with element '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.

Note: Brand existence is not verified within STEP.

ACES Import Validation Rules

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

For information on the supported versions, see the **Supported Versions and Formats** topic withing 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.
Engine and transmission base existence	Import state	Application is imported, but Condition is NOT imported.	Target 'AC_EngineBase_2127123321' of reference not found.
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 Action	Record skipped.	AC_PIESItem_DKGX_BADNumber does not exist for application (id=15).
Part terminology existence	Import state via Business Action	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. Also, Brand existence is not verified within STEP.

TecDoc Import Validation Rules

When importing TecDoc data into STEP it is important to understand the different STEP validation rules. This section addresses the STEP validation rules for the different TecDoc file types listed below:

- Reference Data

Note: This section does not address the details for each file type's data rules per TecAlliance.

Reference Data Import Validation Rules

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

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

Accepted File Extension: .zip

Validation Rule	Occurs	When Validation Fails	Failed Validation Message Example
Attribute validity	Import state		Error:The attribute 'TD_ATTR_BJvon' is not valid for 'classification 'TD_AXLE_CODE.'
Link type validity	Import state		Error: The link type 'TD_ UniversalGAToSearchTree(Universal)' is not valid for 'classification 'TD_GENART_07772.'
Object Type existence	Import state		ObjectType 'TD_NoAssemblyGroupSynonym' not found.

Validation Error Handling

Each import has some basic format validations applied, and each implementation must determine how these should be handled. For more information about the basic format validations, see the **Import Validation Rules** section of the **Automotive Reference Guide**. 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.

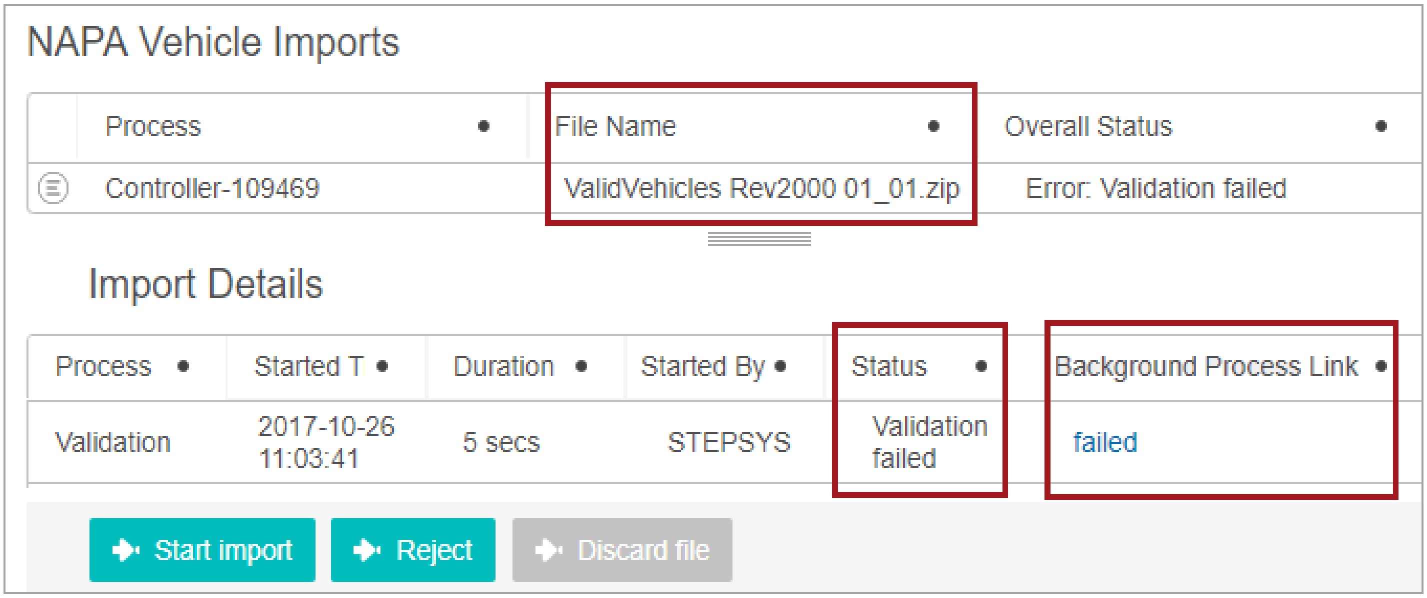
For example, when reference data includes multiple files and a data point included in one is not present in a corresponding related file (e.g., a PCdb Codemaster file includes a position that is omitted from the Position file in the same PCdb), an error will be written to the execution report of the 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 the users 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 Imports controller screen displays the import file name, if you look closely you can see that spaces were used instead of underscores. Because the 'Continue on Error' parameter was disabled at the time this import file was processed, the Import Details table displays the status as 'Validation failed,' the Background Process Link displays as 'failed,' and the Overall Status of the import displays as 'Error: Validation failed.'



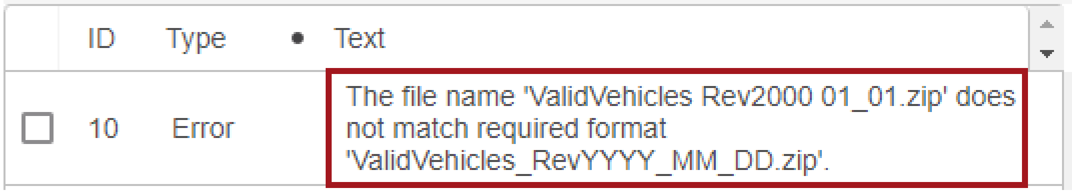
NAPA Vehicle Imports

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

Import Details

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

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



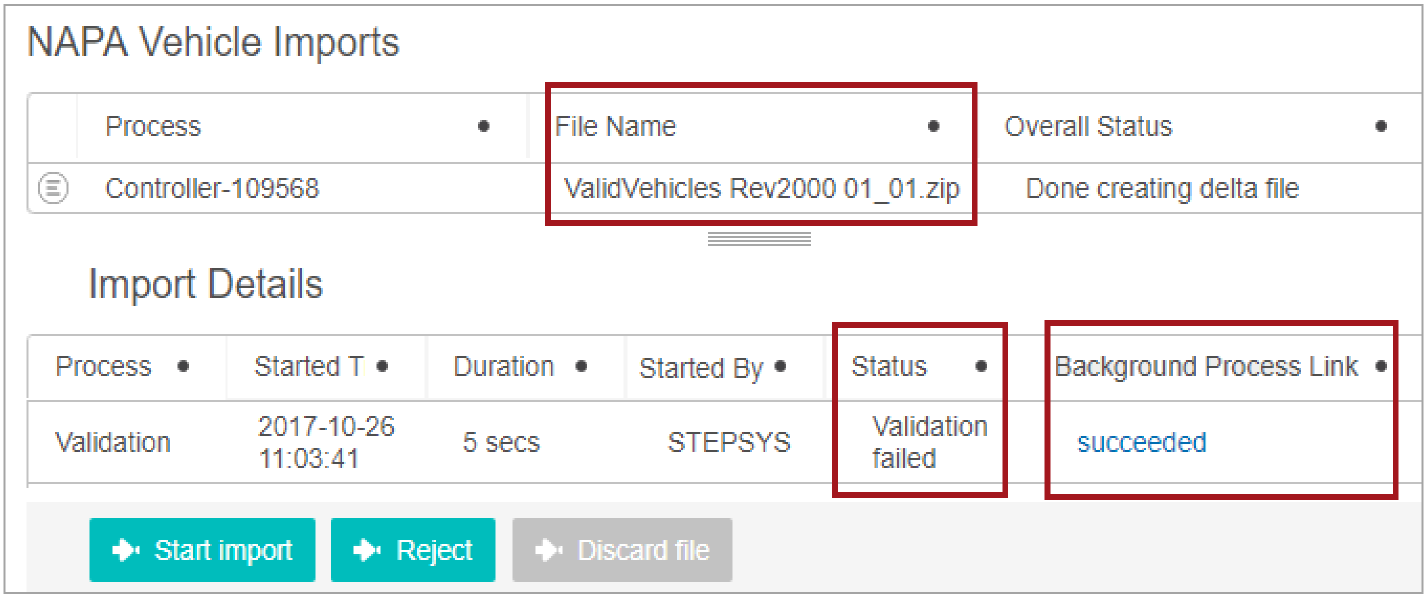
	ID	Type	Text
□	10	Error	The file name 'ValidVehicles Rev2000 01_01.zip' does not match required format 'ValidVehicles_RevYYYY_MM_DD.zip'.

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

Continue on Error Enabled

then you would want to enable the 'Continue on error' parameter. With this setting enabled, the import will continue on in the workflow, but the Validation process status will display within the Web UI import controller screen import details as 'Validation failed,' and the Background Process Link will display as 'succeeded. '

In the example below, the NAPA Vehicle Imports controller screen displays the import file name, if you look closely you can see that spaces were used instead of underscores. Because the 'Continue on Error' parameter was enabled at the time this import file was processed the Import Details table displays the status as 'Validation failed,' but the Background Process Link displays 'succeeded' and the import file was able to enter the next state of the workflow, despite the file name not meeting the validation criteria.



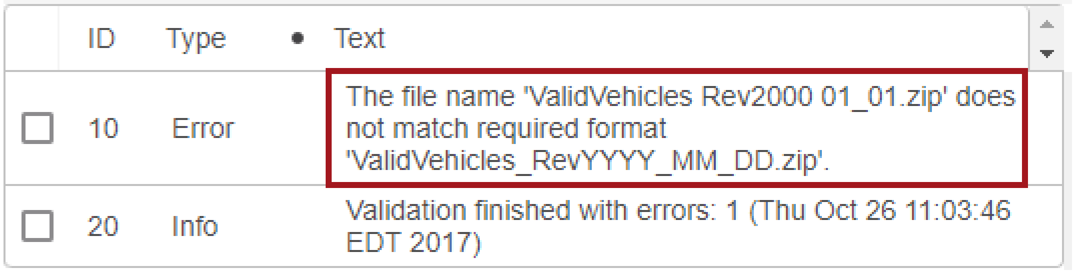
NAPA Vehicle Imports

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

Import Details

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

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



	ID	Type	●	Text
□	10	Error		The file name 'ValidVehicles Rev2000 01_01.zip' does not match required format 'ValidVehicles_RevYYYY_MM_DD.zip'.
□	20	Info		Validation finished with errors: 1 (Thu Oct 26 11:03:46 EDT 2017)

Exporting Automotive Data

The Automotive solution provides extensive export capabilities. For information on the supported versions, see the **Supported Versions and Formats** topic withing this guide. For information on the way a user initiates and manages exports, see the **Using Automotive Exporters** section of the **Automotive Quick Start Guide**.

This section addresses the following available automotive exporters:

- AutoCare ACES Application Exporter
- AutoCare PIES 6.5 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 and 3.2 are supported. The exporter requires that applications be stored in the standard AutoCare data model.

Prerequisites

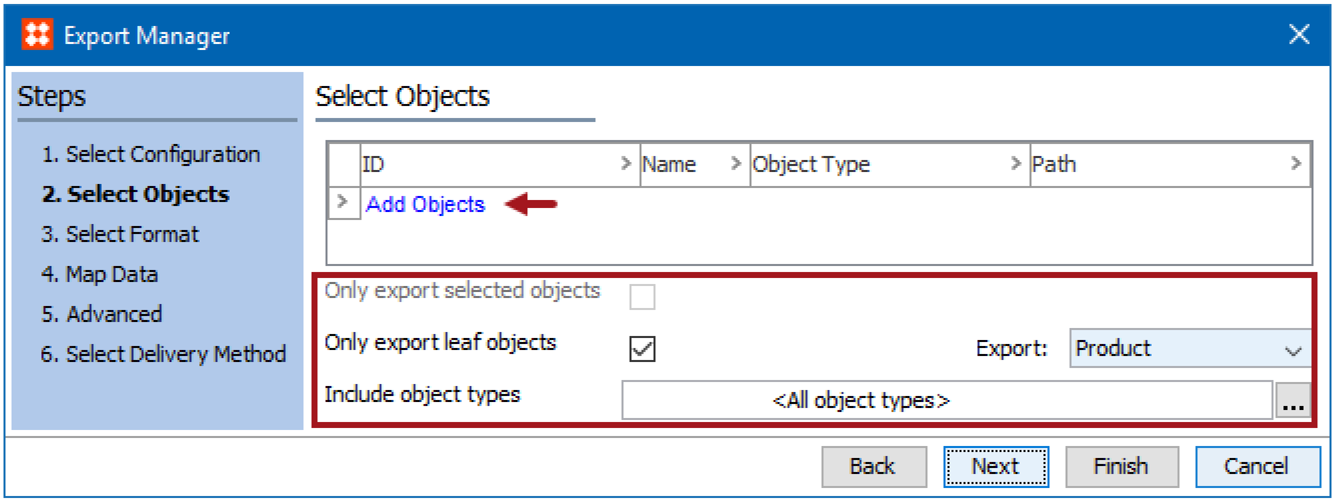
The ACES exporter runs using standard Export Manager functionality so only information specific to the ACES exporter is covered within this guide. Additional information on general Export Manager functionality is covered in the **Export Manager** topic within the **Data Exchange** section of **STEP Online Help**.

Starting the 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 context 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 screen, 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.



Only export selected objects □

Only export leaf objects ☑ Export: Product

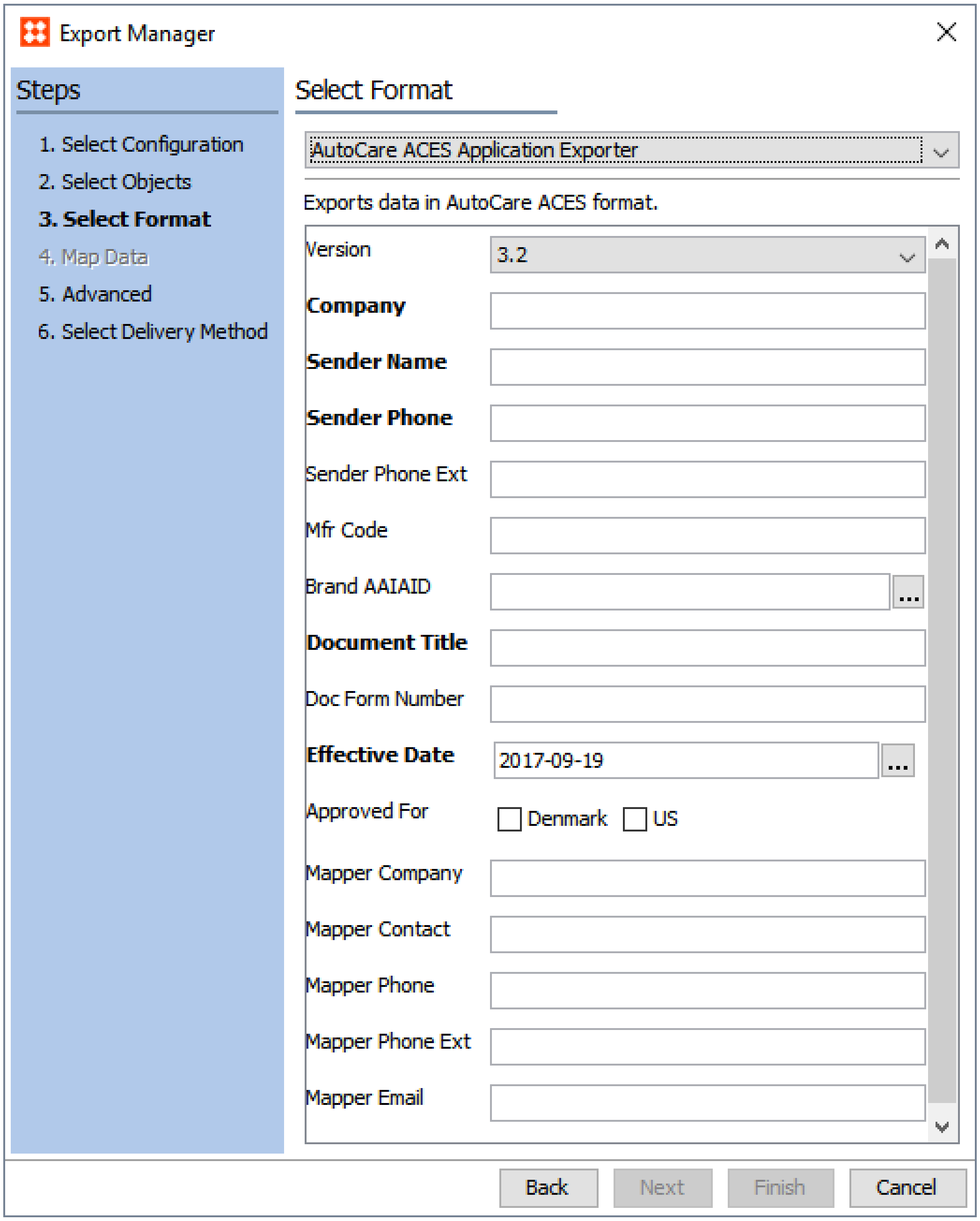
Include object types <All object types> ...

Click the **Add Objects** link to add one or more blue folders in the AutoCare PIES Products hierarchy, or one or more classification folders.

The result is that all ACES Application objects beneath the selection(s) (either as child objects or via links) are exported.

Step 3. Select Format

Select **AutoCare ACES Application Exporter** format to expose options specific to the ACES exporter. The version will default to 3.2, but version 3.0 can be selected as well from the drop-down list. All fields in bold require manual data population and the Next button will not enable until they have been populated.



Select Format

AutoCare ACES Application Exporter

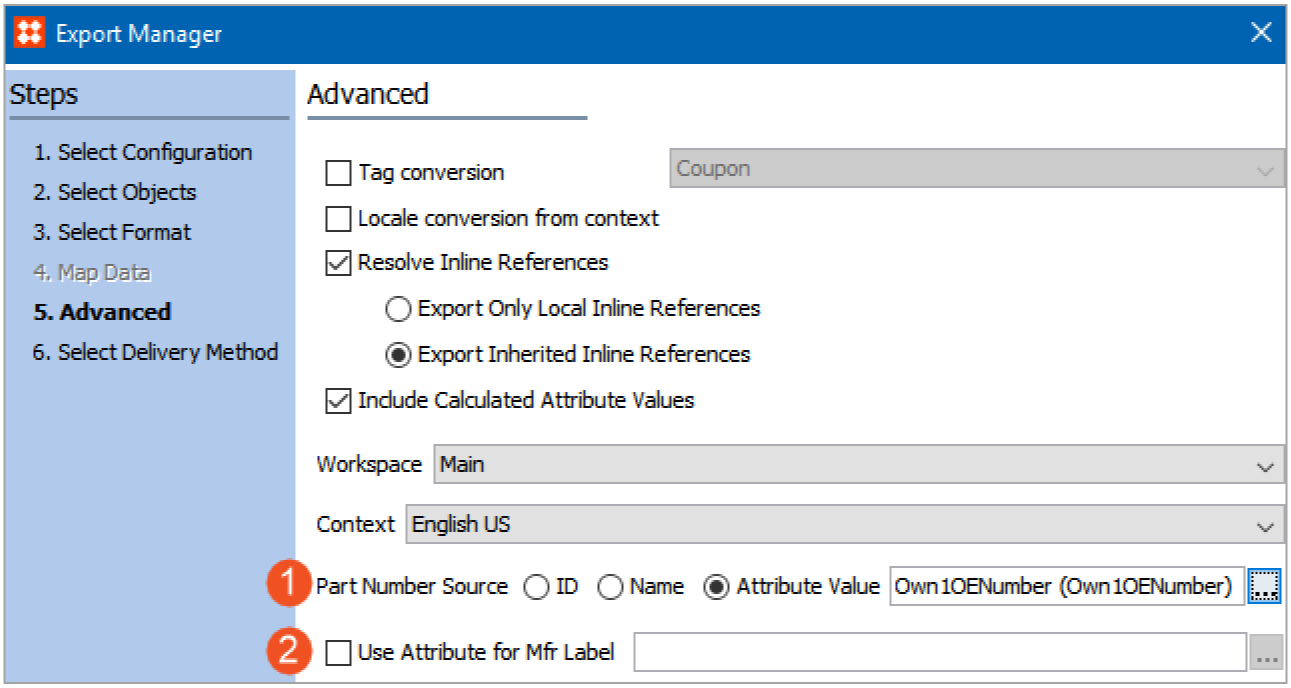
Exports data in AutoCare ACES format.

Version	3.2
Company	
Sender Name	
Sender Phone	
Sender Phone Ext	
Mfr Code	
Brand AAIAID	...
Document Title	
Doc Form Number	
Effective Date	2017-09-19 ...
Approved For	□ Denmark □ US
Mapper Company	
Mapper Contact	
Mapper Phone	
Mapper Phone Ext	
Mapper Email	

Note: Use of the Approved For field requires manual setup, which is covered in the Admin Setup portion of this Quick Start Guide.

Step 5. Advanced

The Advanced step includes two unique parameters for the ACES exporter:



Export Manager

Steps

1. Select Configuration
2. Select Objects
3. Select Format
4. Map Data
5. **Advanced**
6. Select Delivery Method

Advanced

□ Tag conversion Coupon

□ Locale conversion from context

☑ Resolve Inline References

○ Export Only Local Inline References

● Export Inherited Inline References

☑ Include Calculated Attribute Values

Workspace Main

Context English US

1 Part Number Source ○ ID ○ Name ● Attribute Value Own1OENumber (Own1OENumber)

2 □ Use Attribute for Mfr Label

1. **Part Number Source:**
   - Selecting the **ID** option inserts the STEP ID of the parental PIES part into the <Part> field of the exported file.
   - Selecting the **Name** option inserts the STEP Name of the parental PIES part into the <Part> field of the exported file.
   - Selecting the **Attribute Value** option inserts the selected attribute's value from the parental PIES part into the <Part> field of the exported file.
   - Selecting the **Use Attribute for Mfr Label** option inserts the selected attribute's value from the parental PIES part or the exported applications into the <MfrLabel> field of the exported file.
   - Data is exported from the PIES Item (object type ID = AC_PIESItem) that is parent to the applications being exported.
2. **Use Attribute for Mfr Label:**
   - 'Use Attribute for Mfr Label' data gives users the option to supply any attribute value from the parental PIES Item or the ACES application object to be populated in the <MfrLabel> tag of the application.
   - If the 'Use Attribute for Mfr Label' 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.

Sending Data Downstream

In order to send data to downstream systems in both ACES 3.0 and 3.2 formats, the AutoCare ACES Application Exporter has a Version parameter to select either 3.0 or 3.2 format.

The version will default to 3.2, but 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 XSD.

Considerations for Version 3.0 vs. 3.2

- <ACES version="X.X"> must be populated accordingly in the Header segment
- The ApprovedFor tag in the Header segment varies based on format. Assuming values of 'US' and 'DK are included:
    - 3.0:

      ```
      <ApprovedFor>DK,US</ApprovedFor>
      ```

    - 3.2:

      ```
      <ApprovedFor>

      <Country>DK</Country>

      <Country>US</Country>

      </ApprovedFor>
      ```

- 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 export, AssetName will also be populated within the application record, but an additional DigitalAsset segment with details about the asset is also included.
    - The value for AssetName comes from attribute ID=AC_PIES_ASSTAssetID on the asset.
    - The asset included in an export must be linked to the application record using any of these Image and Document Reference Types:
        - AC_ACESApplicationToInstallation
        - AC_ACESApplicationToOwnersManual
        - AC_ACESApplicationToPrimaryProductImage
        - AC_ACESApplicationToProductImage.

Note: ACES standard only supports one asset linked to each application, but STEP supports multiple assets linked per application. If more than one asset is linked to the application in STEP, only one asset is exported randomly. An entry is written to the execution report of the export for any skipped assets.

AutoCare PIES 6.5 Exporter

The AutoCare PIES 6.5 Exporter is used to export parts data in PIES 6.5 format. The exporter defaults to exporting data in the standard AutoCare data model, but configuration options are also available to export data in PIES format from an Own model.

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

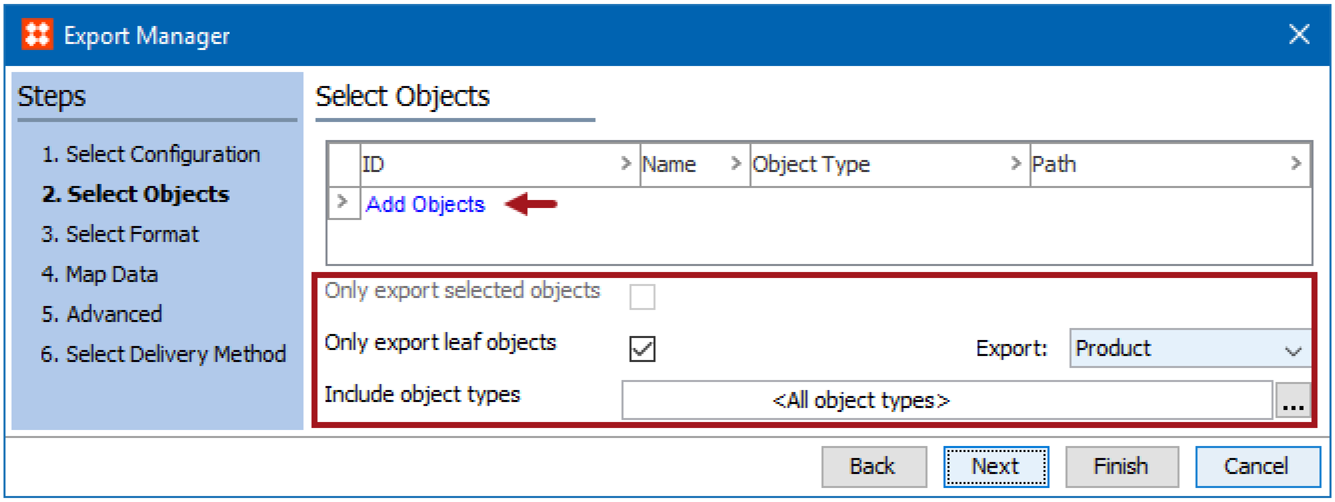
The PIES exporter runs using standard Export Manager functionality so only information specific to the PIES exporter is covered within this guide. Additional information on general Export Manager functionality is covered in the **Export Manager** topic within the **Data Exchange** section of **STEP online help**.

Starting the Exporter

The AutoCare PIES 6.5 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 context 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 screen, 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.



ID	Name	Object Type	Path
Add Objects			

Only export selected objects □

Only export leaf objects ☑

Export: Product

Include object types <All object types> ...

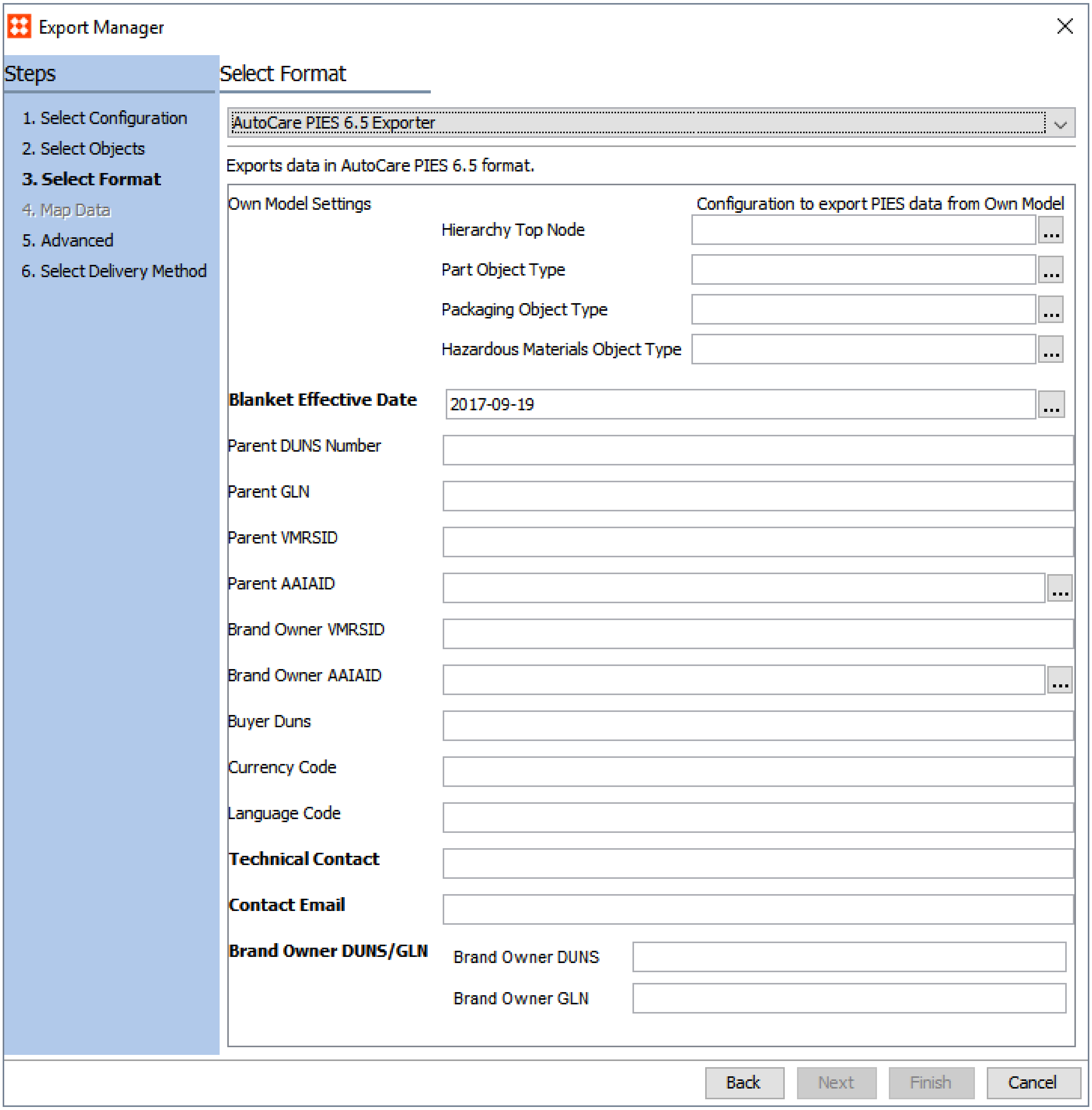
Click the **Add Objects** link to add one or more blue folders in the AutoCare PIES Products hierarchy, or one or more blue folders from an Own hierarchy, or one or more classification folders.

The result is that all PIES objects beneath the selection(s) (either as child objects or via links) are exported.

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 6.5 Exporter** to expose the PIES exporter configuration parameters. 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. In either case, the only other parameters that need to be populated are those in bold, and the Next button will not enable until data has been provided for them.



Export Manager

Steps

1. Select Configuration
2. Select Objects
3. **Select Format**
4. Map Data
5. Advanced
6. Select Delivery Method

Select Format

AutoCare PIES 6.5 Exporter

Exports data in AutoCare PIES 6.5 format.

Own Model Settings — Configuration to export PIES data from Own Model

Hierarchy Top Node	
Part Object Type	
Packaging Object Type	
Hazardous Materials Object Type	
Blanket Effective Date	2017-09-19
Parent DUNS Number	
Parent GLN	
Parent VMRSID	
Parent AAIAID	
Brand Owner VMRSID	
Brand Owner AAIAID	
Buyer Duns	
Currency Code	
Language Code	
Technical Contact	
Contact Email	
Brand Owner DUNS/GLN — Brand Owner DUNS	
Brand Owner GLN	

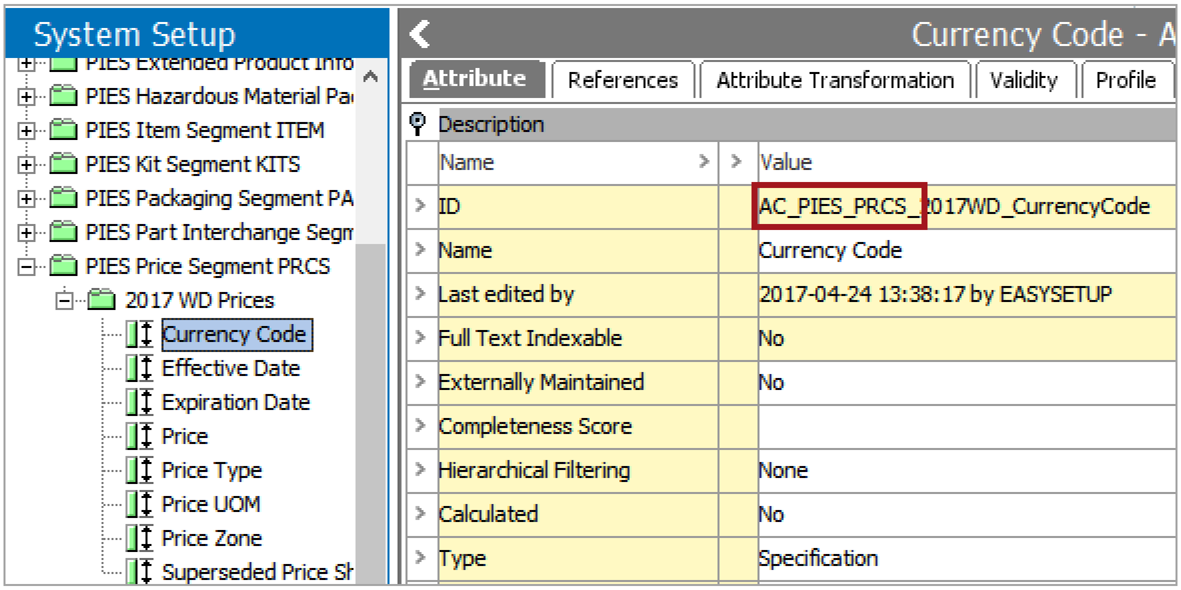
The Own Model Settings are described below, and should only be populated if data should be exported from an Own model rather than the standard AutoCare model.

Export objects (Own parts) other than PIES Items

The PIES Exporter includes the following four parameters that allow users to select their own object types to be included in the exported file:

- **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 attribute group and attributes used to store the price sheet information must use the AutoCare IDs which is prefixed with '**AC_PIES_PRCS_**. '

An example of the attribute ID structure:

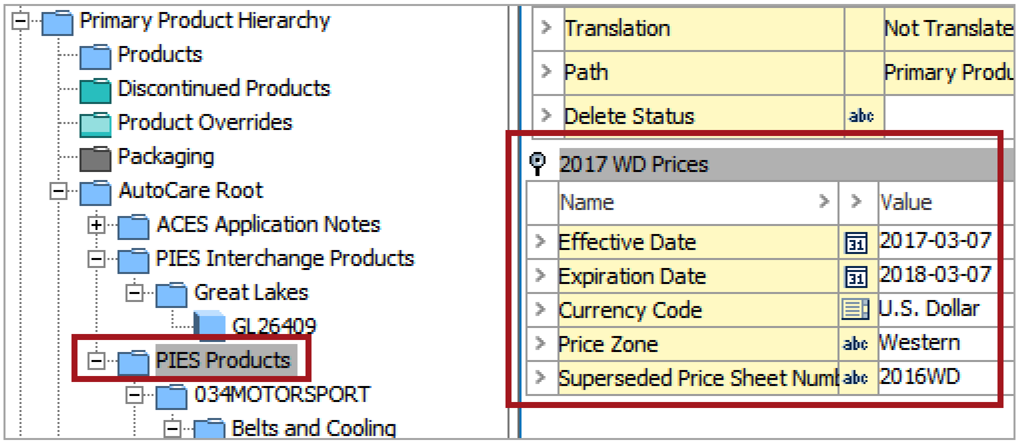


Name	Value
ID	AC_PIES_PRCS_2017WD_CurrencyCode
Name	Currency Code
Last edited by	2017-04-24 13:38:17 by EASYSETUP
Full Text Indexable	No
Externally Maintained	No
Completeness Score	
Hierarchical Filtering	None
Calculated	No
Type	Specification

PIES Importer creates 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, 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:



Translation		Not Translate
Path		Primary Produ
Delete Status	abc	

2017 WD Prices

Name		Value
Effective Date	31	2017-03-07
Expiration Date	31	2018-03-07
Currency Code		U.S. Dollar
Price Zone	abc	Western
Superseded Price Sheet Numl	abc	2016WD

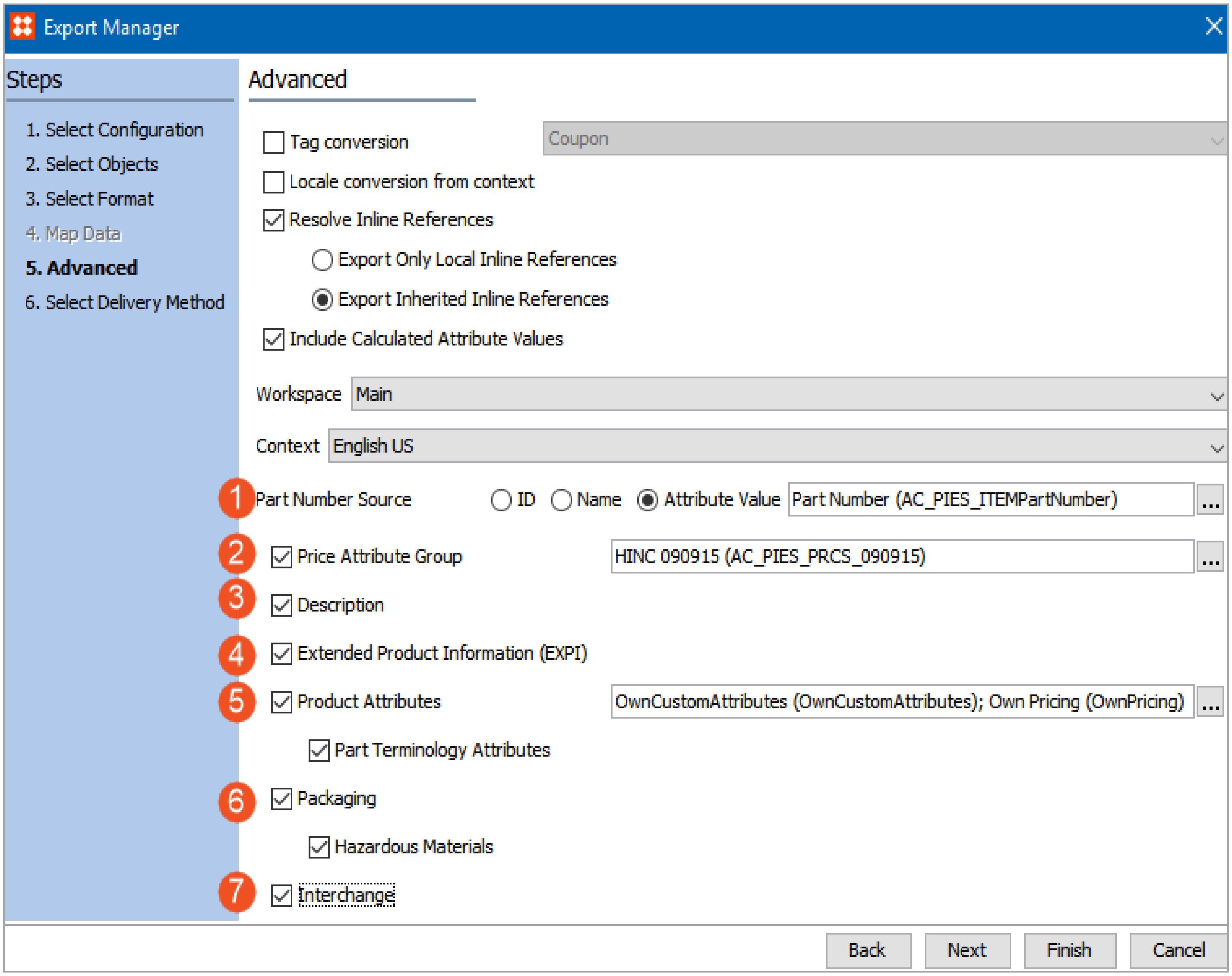
- **Part Object Type**: This is a required parameter for selection of the Own model object type representing the PIES Item.
    - The selected object must have all required PIES attributes and references (including the AC_ ProductToPartTerminology reference) from the standard model made valid on the object and populated with data.
    - If the required data is not provided, the file will fail XSD validation on downstream systems.

Note: The Part Object Type cannot be a direct child of the Hierarchy Top Node.

- **Packaging Object Type**: This is an optional parameter for selection of Packaging object type.
    - If PIES Packaging segment is to be included in an export running outside of the AutoCare PIES hierarchy, the customer packaging object(s) must exist as child to the part object.
    - If not selected, users cannot expect packaging data to export properly, and the export may fail if selections in the subsequent screen of the wizard (step 5. Advanced) try to include it.
- **Hazmat Object Type**: This is an optional parameter for selection of Hazmat object type.
    - If PIES Hazmat segment is to be included in an export running outside of the AutoCare PIES hierarchy, the Hazmat object(s) must exist as child to the Packaging object.
    - If not selected, users cannot expect hazmat data to export properly, and the export may fail if selections in the subsequent screen of the wizard (step 5. Advanced) try to include it.

Step 5. Advanced

The Advanced screen include many options that controls what data gets exported. 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.



Steps

1. Select Configuration
2. Select Objects
3. Select Format
4. Map Data
5. **Advanced**
6. Select Delivery Method

Advanced

☐ Tag conversion — Coupon

☐ Locale conversion from context

☑ Resolve Inline References

○ Export Only Local Inline References

● Export Inherited Inline References

☑ Include Calculated Attribute Values

Workspace: Main

Context: English US

Part Number Source: ○ ID ○ Name ● Attribute Value — Part Number (AC_PIES_ITEMPartNumber)

☑ Price Attribute Group — HINC 090915 (AC_PIES_PRCS_090915)

☑ Description

☑ Extended Product Information (EXPI)

☑ Product Attributes — OwnCustomAttributes (OwnCustomAttributes); Own Pricing (OwnPricing)

☑ Part Terminology Attributes

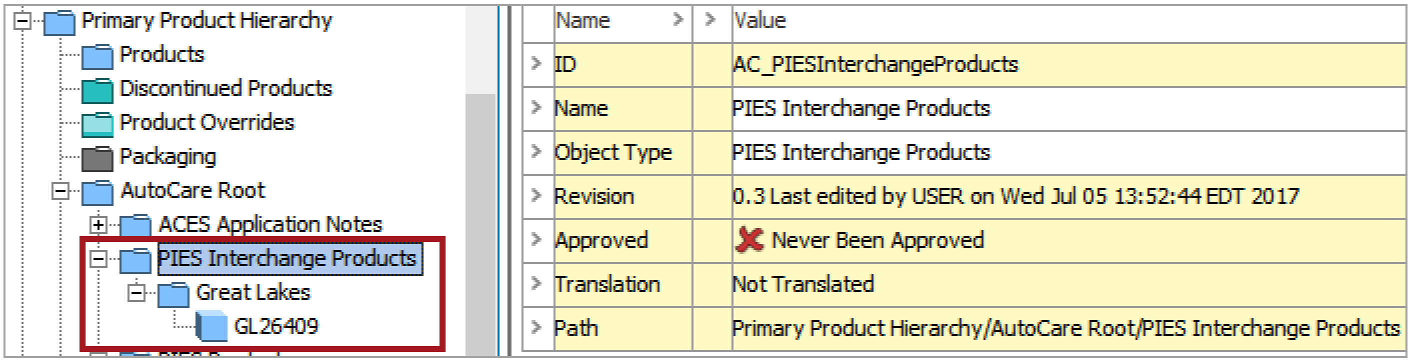
☑ Packaging

☑ Hazardous Materials

☑ Interchange

1. **Part Number Source:**
   - Selecting the **ID** option inserts the STEP ID of the PIES object in the <PartNumber> field in the exported file
   - Selecting the **Name** option inserts the STEP Name of the PIES object in the <PartNumber> field in the exported file
   - Selecting the **Attribute Value** option inserts the selected attribute's value of the PIES object in the <PartNumber> field in the exported file
2. **Price Attribute Group:** If the PIES Pricing segment *(<PriceSheets> and <Prices>)* is to be included in the export, then the attribute group that holds the PIES Pricing attributes needs to be selected in this option.
3. **Description:** If this option is selected, all standard PIES Description attributes that are valid for the PIES object and have a value will be exported in the <Descriptions> segment.

4. **Extended Product Information (EXPI):** If this option is selected, all standard PIES EXPI attributes that are valid for the PIES object and have a value will be exported in the <ExtendedInformation> segment.
5. **Product Attributes / Part Terminology Attributes:** This option allows the customer to select attribute groups that contain their own custom attributes to be included in the exported file, e.g. any attributes outside of those included in the PIES standard can be sent in this way. The values for these attributes will be included in the <ProductAttributes> segment for the PIES Item.
6. **Packaging / Hazardous Materials:** If this option is selected, exported data will be displayed in the <Packages> and <HazardousMaterial> segments.
   - If PIES Packaging and Hazmat segment 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.
7. **Interchange** - If this option is selected, exported data will be displayed in the <PartInterchangeInfo> segment.
   - The PIES Importer creates interchange records under the PIES Interchange Products node, which has a Brand and an Interchange child structure. If PIES Interchange data is to be exported, then this standard structure must be in place with the appropriate data on the interchanges (as is created with the PIES Importer).



Name	Value
ID	AC_PIESInterchangeProducts
Name	PIES Interchange Products
Object Type	PIES Interchange Products
Revision	0.3 Last edited by USER on Wed Jul 05 13:52:44 EDT 2017
Approved	✗ Never Been Approved
Translation	Not Translated
Path	Primary Product Hierarchy/AutoCare Root/PIES Interchange Products

Important: If interchanges should be included in an export running from an Own model, the Own PIES Product object type must be added as a valid source for the Product Reference with ID= AC_PIESInterchange, and a reference must exist from the Own PIES Product to the AutoCare PIES Interchange object.

NAPA Application Exporter

The NAPA Application Exporter is used to export application data that resides in a NAPA product hierarchy. Users can select an object from the NAPA product hierarchy, and the NAPA Application objects beneath the selected object (both child objects and linked objects) will be exported. The export file is then available in the NAPA Application format.

Prerequisites

NAPA Data Model Requirements

The exporter requires applications be stored in the standard NAPA data model.

NAPA Product Requirements

The NAPA Product MPCC is expected to have an ID structure prefixed with NAPA_MPCC_, and the NAPA Product is expected to have the following ID structure: NAPA_Product_[Line Abbrev][PartNumber] (i.e., NAPA_Product_FIL1515). When a NAPA Product is created, the NAPA Part Number attribute (NAPA_PartNumber) must be populated with the Part Number value, as this attribute value is used to the determine the Part Number in the NAPA Application Exporter.

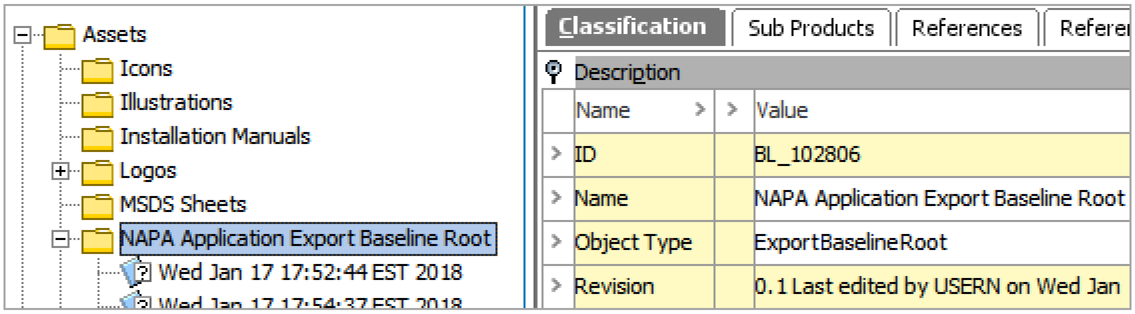
Important: Because some Product Lines only have two characters, the NAPA Product Number, if used without the overlying ID structure, might create an inconsistent pattern when trying to use the Product Number for the ID.

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.

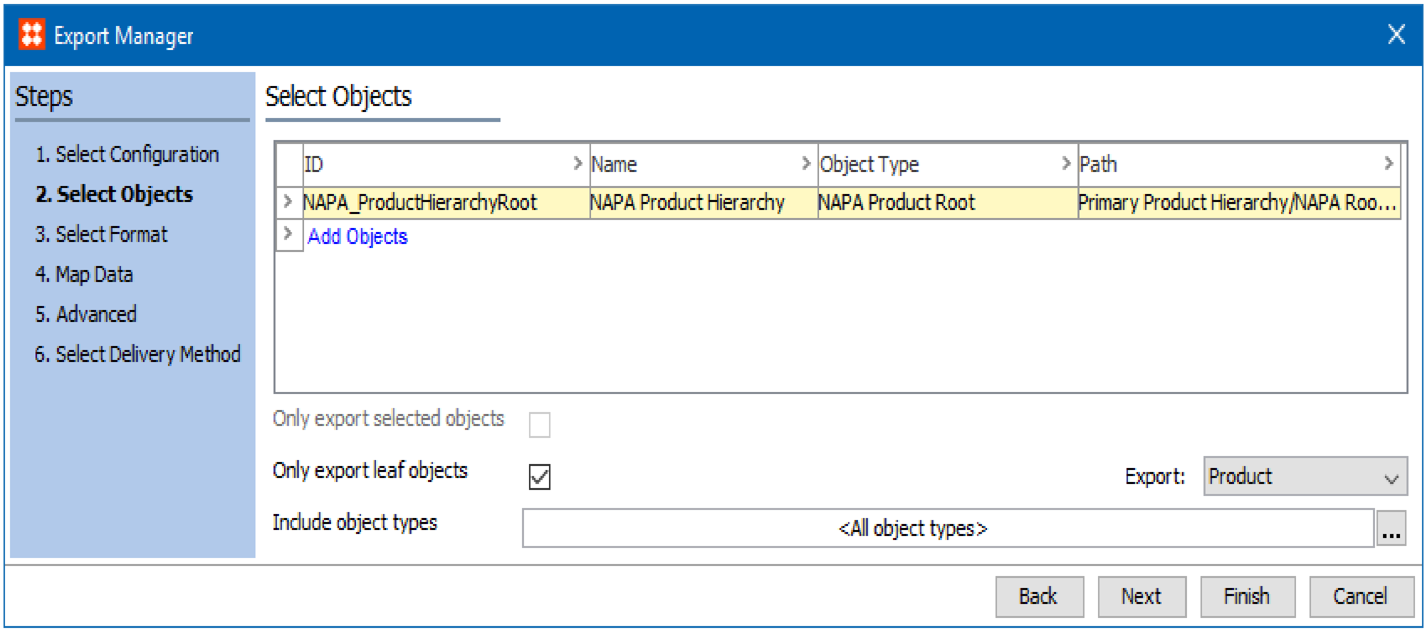


Name	Value
ID	BL_102806
Name	NAPA Application Export Baseline Root
Object Type	ExportBaselineRoot
Revision	0.1 Last edited by USERN on Wed Jan

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.



Steps

1. Select Configuration
2. **Select Objects**
3. Select Format
4. Map Data
5. Advanced
6. Select Delivery Method

Select Objects

ID	Name	Object Type	Path
NAPA_ProductHierarchyRoot	NAPA Product Hierarchy	NAPA Product Root	Primary Product Hierarchy/NAPA Roo...
Add Objects			

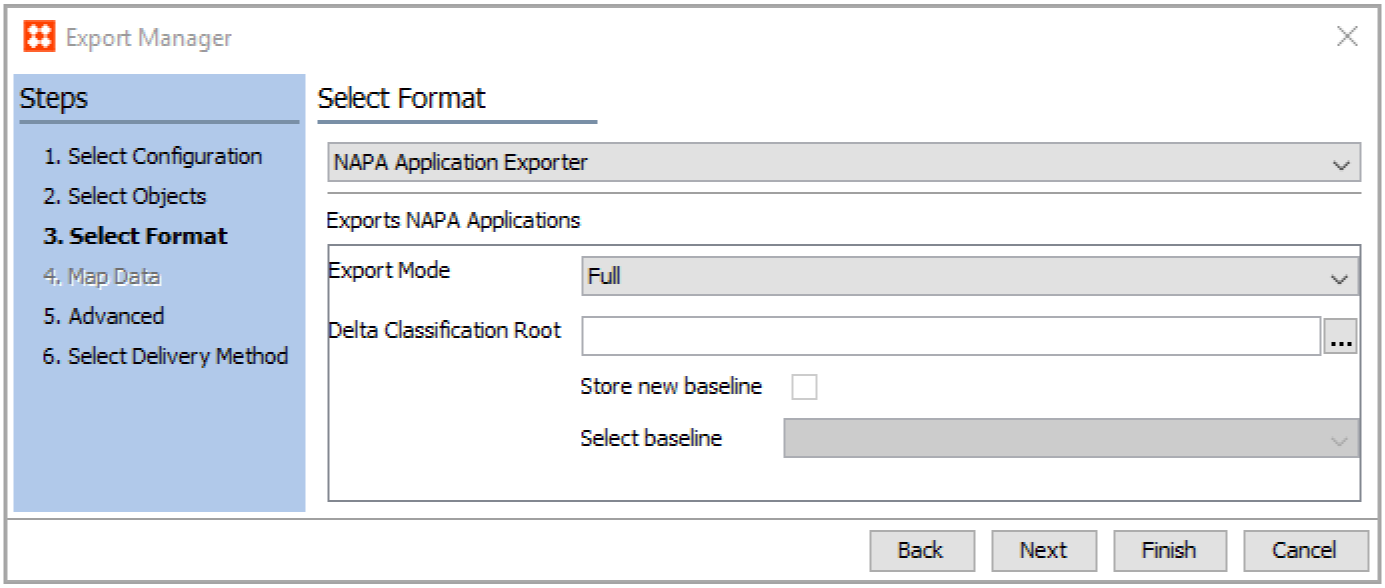
Only export selected objects □

Only export leaf objects ☑

Export: Product

Include object types <All object types> ...

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.



Steps

1. Select Configuration
2. Select Objects
3. **Select Format**
4. Map Data
5. Advanced
6. Select Delivery Method

Select Format

NAPA Application Exporter

Exports NAPA Applications

Export Mode: Full

Delta Classification Root: ...

Store new baseline □

Select baseline

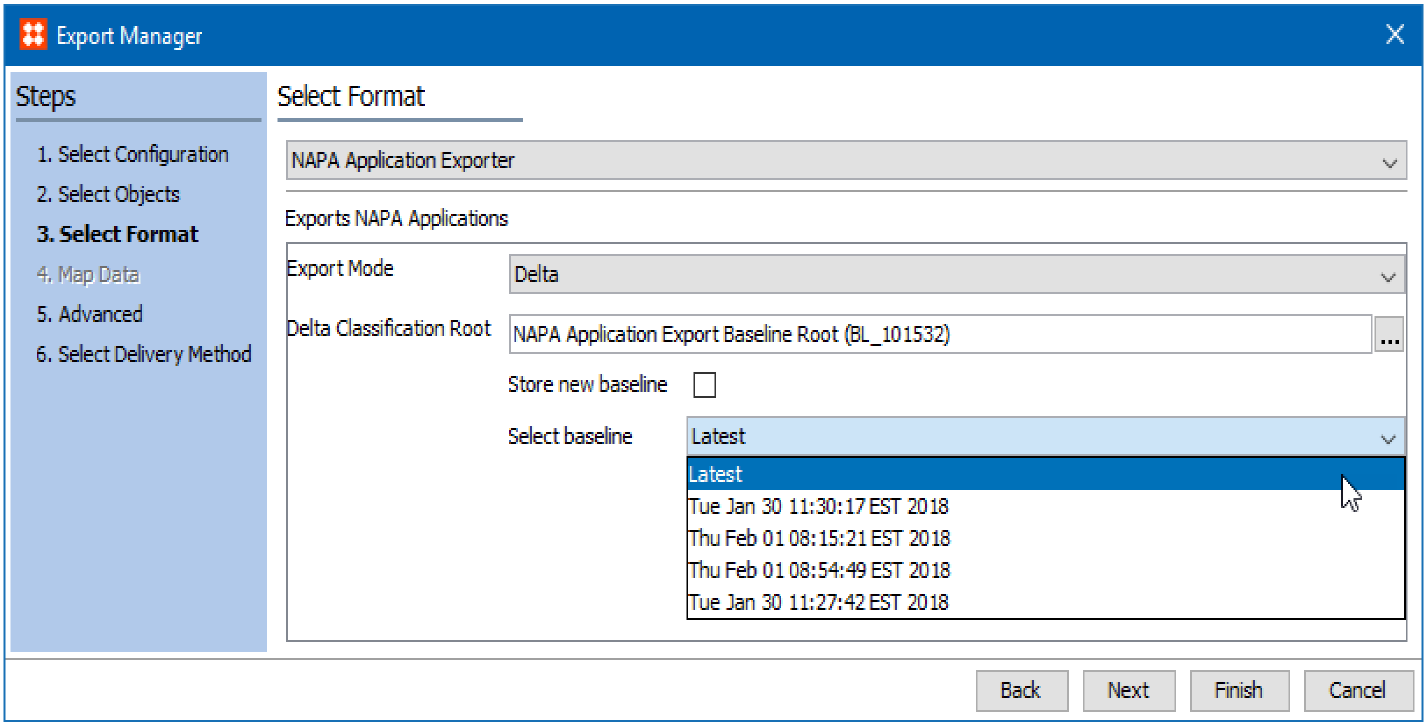
3. Select the one of the following Export Modes from the dropdown:

- **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.

3. Within the Delta Classification Root parameter, click the ellipsis button **(...)** to find and select the alternate classification object (ExportBaselineRoot). Once the Delta Classification Root is populated, the Select baseline parameter dropdown will be enabled.
4. Select the desired baseline from the dropdown list. In the example below, four previous baselines and the 'Latest' baseline can be selected. (Optionally, the 'Store new baseline' checkbox can be enabled to create a new baseline).



Select Format

NAPA Application Exporter

Exports NAPA Applications

Export Mode: Delta

Delta Classification Root: NAPA Application Export Baseline Root (BL_101532)

Store new baseline □

Select baseline: Latest

- Latest
- Tue Jan 30 11:30:17 EST 2018
- Thu Feb 01 08:15:21 EST 2018
- Thu Feb 01 08:54:49 EST 2018
- Tue Jan 30 11:27:42 EST 2018

5. Click the **Next** button and/or **Finish** button to complete the Export Manager wizard. For more information on the core parameters within the Export Manager, see the **Export Manager** topic within the Data Exchange section of Online Help.

Automotive Business Rule Plugins

A number of business rule plugins are provided to assist in extending the core Automotive solution. This section addresses many of the Automotive business rule plugins as they are available when configuring Business Actions and Business Conditions.

Prerequisites

It is assumed that the admin user has knowledge of STEP administrative functions and experience working in System Setup, including creating and editing business rules, workflows, and Web UIs. This section targets only the specific information needed for a knowledgeable STEP admin user to identify and configure the Automotive-specific business rule plugins. For more introductory material of these concepts, see the **Business Rules**, **Workflows**, and **Web User Interfaces** sections of **STEP Online Help**.

Business Action Operations

Menu	Operation Name	Components required for use	Description
Automotive	Change assembly	Application Editor, Bulk Update action button	Allows users to use a Bulk Updates action button within an Application Editor to change the assembly / vehicle of one or more existing applications. Will not change the assembly of a missing application. For more information, see **Business Action: Change Assembly**.
Automotive	Change part	Application Editor, Bulk Update action button	Allows users to use a Bulk Updates action button within an Application Editor to change the part for a selected application. For more information, see **Business Action: Change Part**.
Automotive	Change part type	Application Editor, Bulk Update action button	Allows users to use a Bulk Updates action button within an Application Editor to change the part type for a selected application. Will not change the part type of a missing application. For more information, see **Business Action: Change Part Type**.
Automotive	Copy application to other assembly	Application Editor, Bulk Update action button	Allows users to use a Bulk Updates action button within an Application Editor to copy one or more existing applications to another assembly / vehicle. For more information, see **Business Action: Copy Application to Other Assembly**.
Automotive	Copy application to other part	Application Editor, Bulk Update action button	Allows users to use a Bulk Updates action button within an Application Editor to copy one or more existing applications to another part. For more information, see **Business Action: Copy Application to Other Part**.
Automotive	Copy applications to related parts	Application Editor, Bulk Update action button	Allows users to use a Bulk Updates action button within an Application Editor to copy one or more existing applications to one or more related parts. For more information, see **Business Action: Copy Applications to Related Parts**.
Automotive	Copy classification hierarchy to product hierarchy	Anywhere a bulk update can be run	Allows users to update the product hierarchy to match the classification hierarchy. The 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.

Menu	Operation Name	Components required for use	Description
			CAUTION: Object Types need to be similar in that they should have the same ID (without the prefix).
Automotive	Move ACES Applications for PIES Part		Moves all ACES applications from one PIES Item to another by following a reference between the parts. For more information, see **Business Action: Move ACES Applications for PIES Part**.
Automotive	Set Condition Links on Part Types		Links application conditions / options to part types to assist in configuring display options in the Web UI Application Record Editor. For more information, see **Business Action: Set Condition Links on Part Types.**
Automotive	Sync ACES Applications between PIES Parts		Synchronizes all ACES applications between two PIES Items by following a reference between the parts. For more information, see **Business Action: Sync ACES Applications Between PIES Parts**.
Automotive	Unique application record constraint		Calculates a unique value based on the part, assembly and the conditions for the applications.
Import flow	Set import status attributes	Workflow: Import State	Allows for the implementation of change flags; a way for users to view what data has been created or changed due to an import. Must be used within the Import state of a workflow. For more information, see **Business Action: Set Import Status Attributes**.

Business Condition Operations

Menu	Operation Name	Description
Automotive	Validate Application	Prevents any applications from being imported that do not have both a part type and vehicle / assembly that exists in the STEP database. When a record is found that does not meet the condition, an error is written to the execution report of the import process and the record is not imported. This condition is automatically created and added to the Import state of the relevant workflow when the Easy setup of the import process has been run.

Business Action: Change Assembly

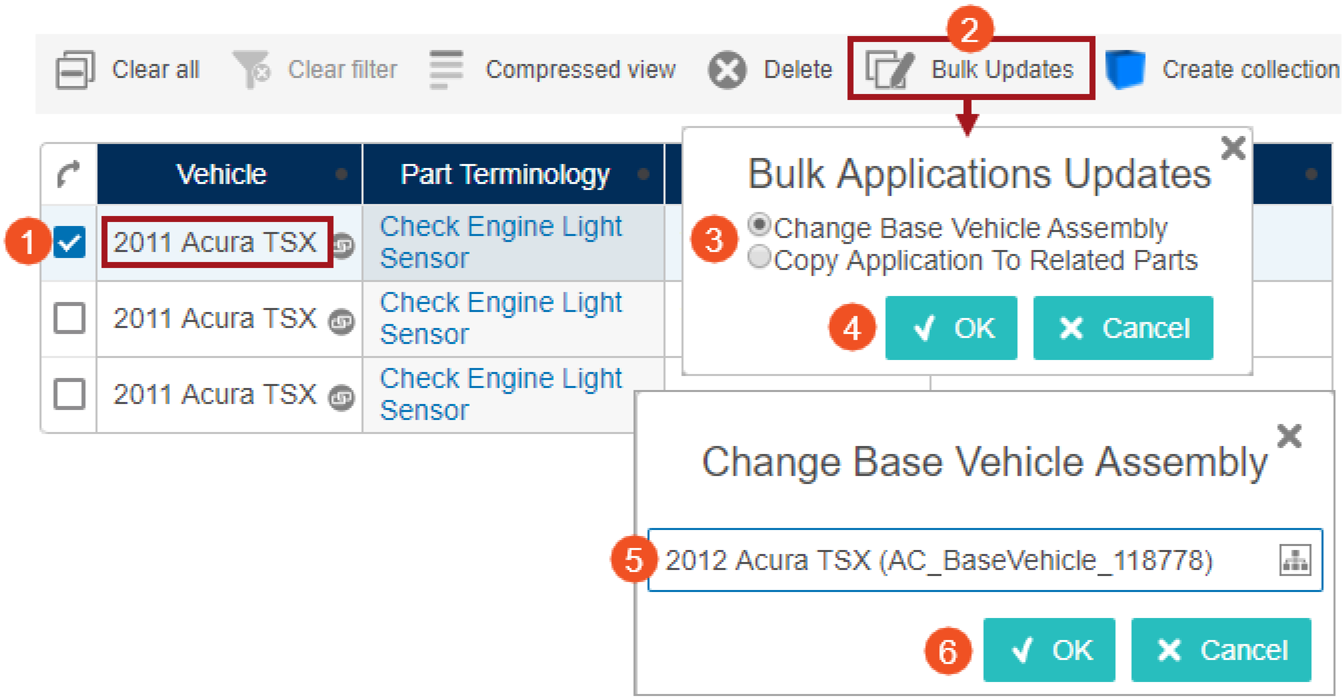
This automotive business action allows users to use a Bulk Updates action button within an Application Editor to change the assembly / vehicle of one or more existing applications. However, it will not change the assembly / vehicle of a missing application, and thus any missing applications selected when the bulk update is run will be ignored. For more information, see the **Missing Coverage** topic within this guide. Setup is required within both STEP Workbench and Web UI for the action to be available to users.

This section addresses steps necessary when:

- Using the Configured Change Assembly Business Action in Web UI
- Configuring the Change Assembly Business Action in Workbench
- Configuring the Change Assembly Business Action in Web UI

Using the Configured Change Assembly Business Action in Web UI

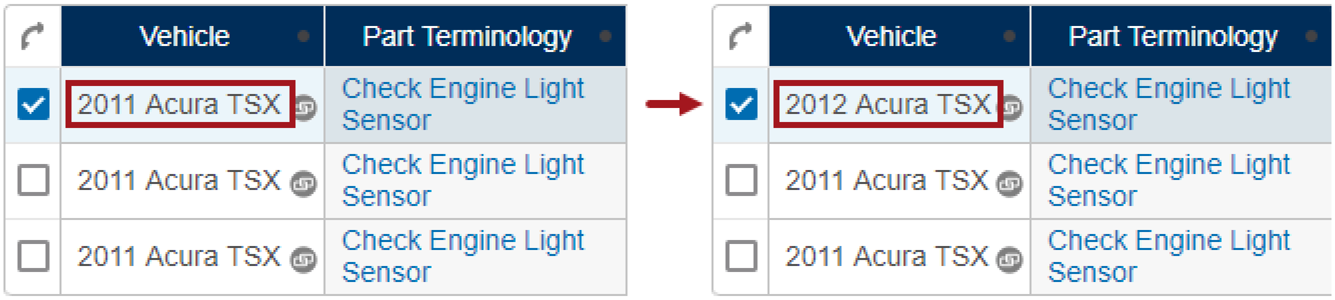
Once configured, a Bulk Updates button within an Application Editor can be used to change the assembly for one or more existing application. Below are the steps for using the business action in Web UI.


Clear all
Clear filter
Compressed view
Delete
Bulk Updates
Create collection
2
Vehicle
Part Terminology
1
2011 Acura TSX
Check Engine Light Sensor
2011 Acura TSX
Check Engine Light Sensor
2011 Acura TSX
Check Engine Light Sensor
Bulk Applications Updates
3
Change Base Vehicle Assembly
Copy Application To Related Parts
4
OK
Cancel
Change Base Vehicle Assembly
5
2012 Acura TSX (AC_BaseVehicle_118778)
6
OK
Cancel

Note: If only one bulk update is configured, then the Bulk Applications Updates dialog will not display.

1. Within the configured Application Editor, 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 Editor results table.

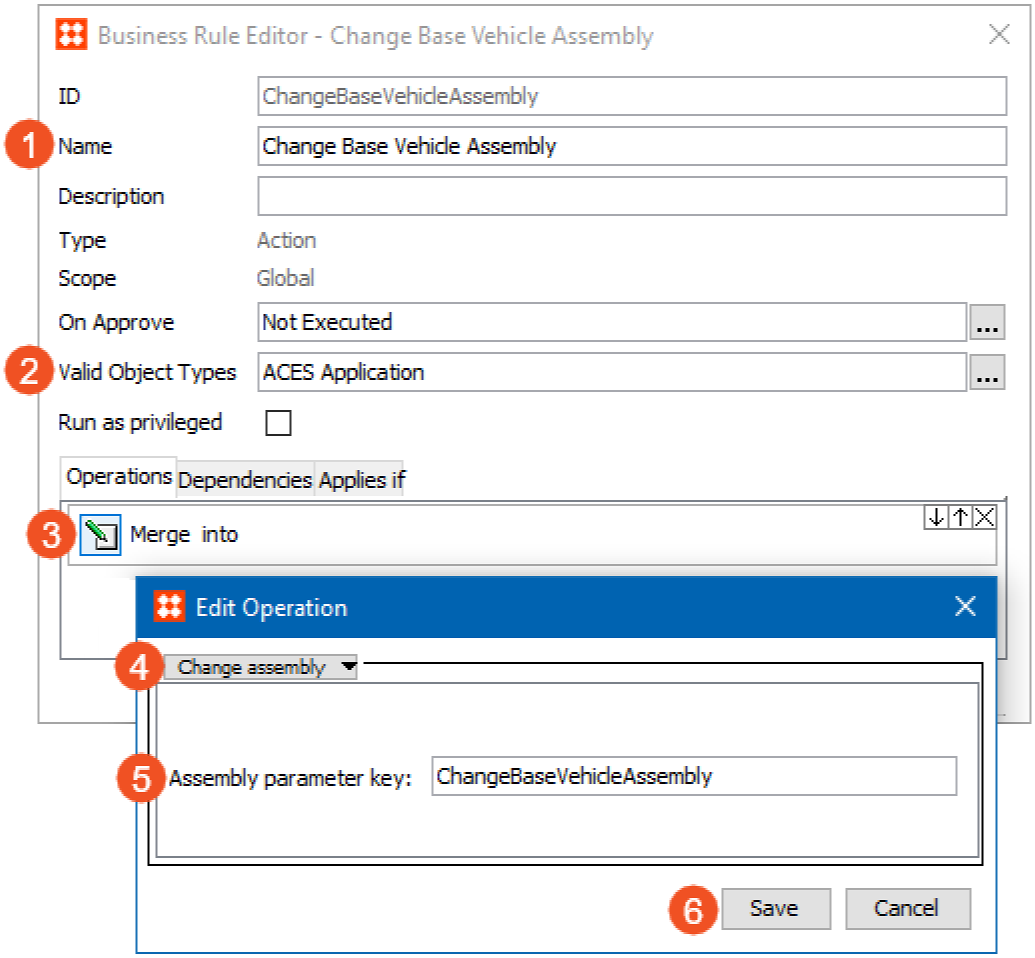


	Vehicle	Part Terminology
☑	2011 Acura TSX	Check Engine Light Sensor
□	2011 Acura TSX	Check Engine Light Sensor
□	2011 Acura TSX	Check Engine Light Sensor

	Vehicle	Part Terminology
☑	2012 Acura TSX	Check Engine Light Sensor
□	2011 Acura TSX	Check Engine Light Sensor
□	2011 Acura TSX	Check Engine Light Sensor

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.



ID: ChangeBaseVehicleAssembly
Name: Change Base Vehicle Assembly
Description:
Type: Action
Scope: Global
On Approve: Not Executed
Valid Object Types: ACES Application
Run as privileged: □
Merge into
Change assembly
Assembly parameter key: ChangeBaseVehicleAssembly

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 Editor 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.

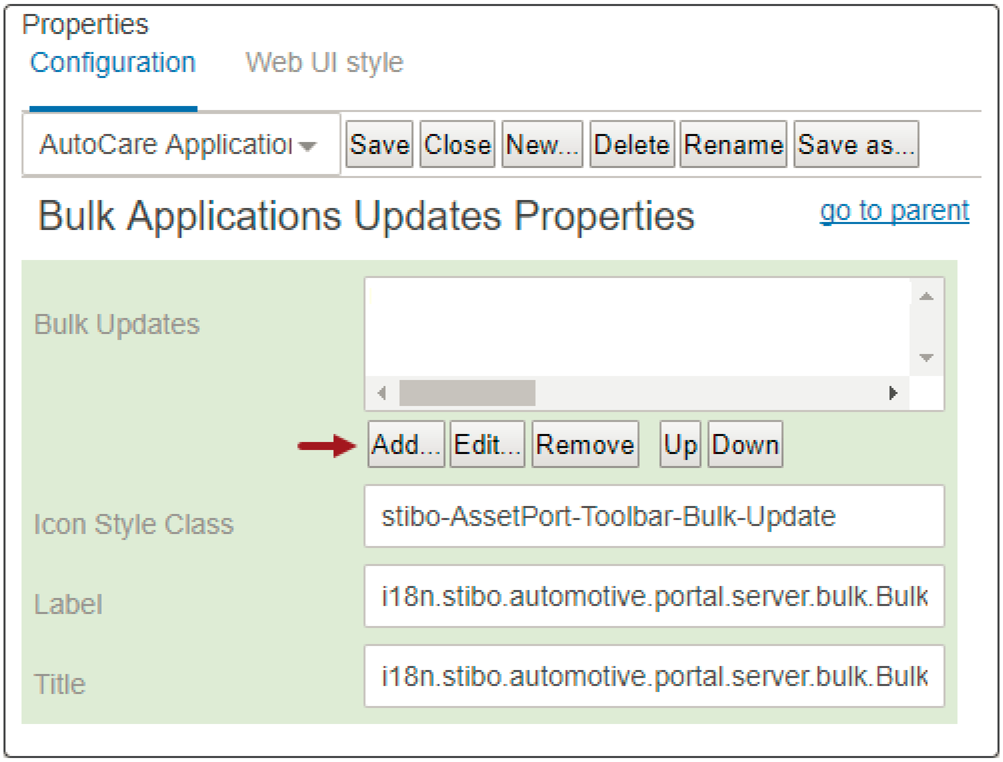
6. Click the **Save** button and continue to the next topic, **Configuring the Change Assembly Business Action in Web UI**.

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 Editor 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).



Bulk Applications Updates Properties

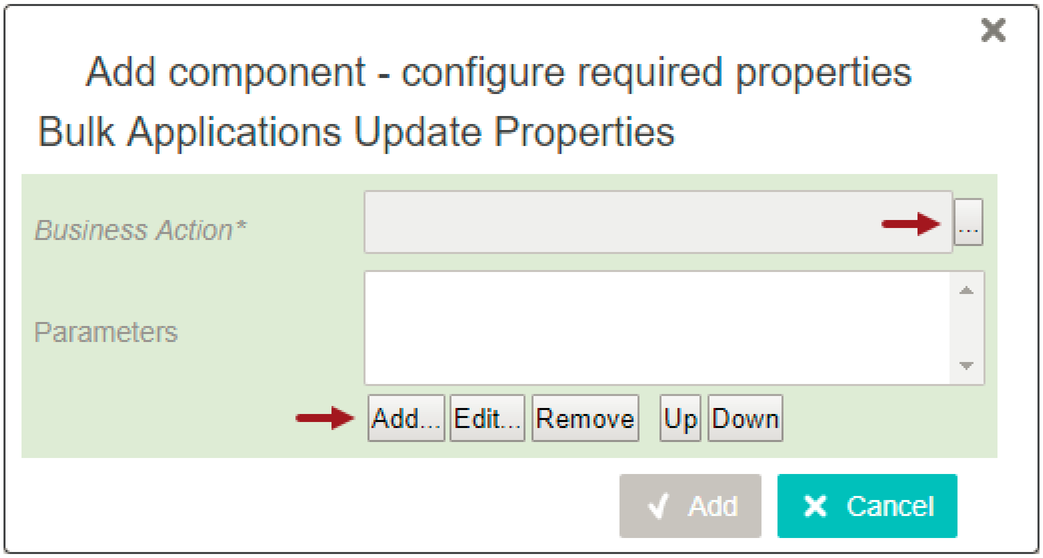
Bulk Updates

Icon Style Class: stibo-AssetPort-Toolbar-Bulk-Update

Label: i18n.stibo.automotive.portal.server.bulk.Bulk

Title: i18n.stibo.automotive.portal.server.bulk.Bulk

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).



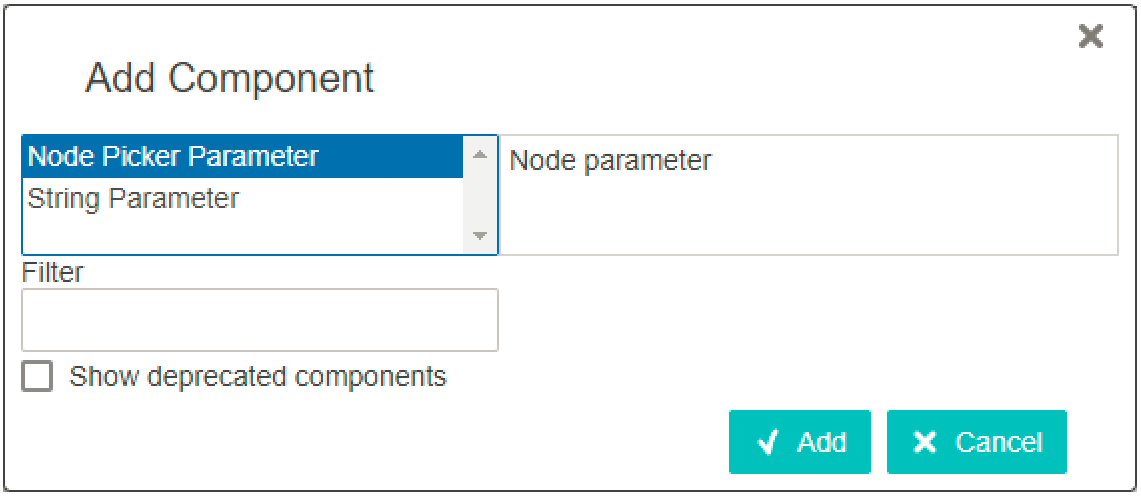
Add component - configure required properties

Bulk Applications Update Properties

Business Action*

Parameters

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).



Add Component

Node Picker Parameter

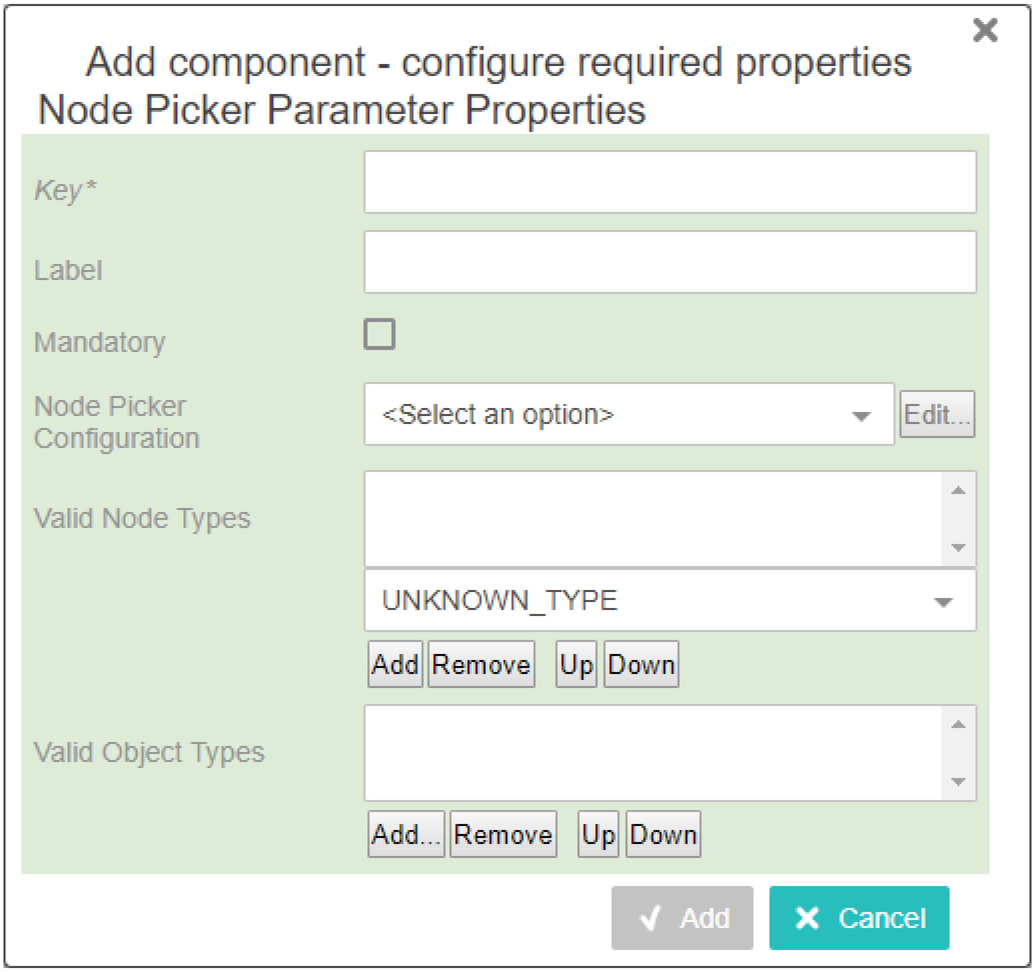
String Parameter

Node parameter

Filter

□ Show deprecated components

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



Add component - configure required properties

Node Picker Parameter Properties

Key*

Label

Mandatory ☐

Node Picker Configuration <Select an option>

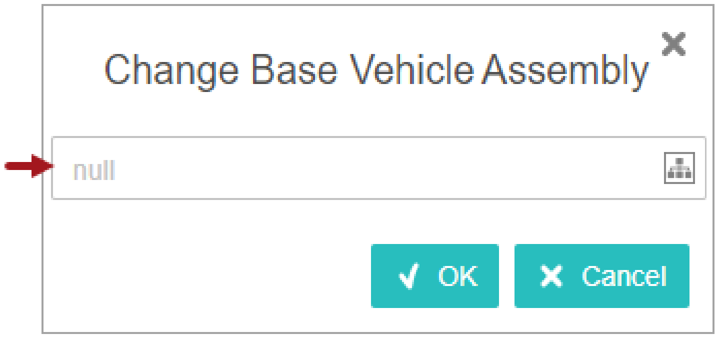
Valid Node Types UNKNOWN_TYPE

Valid Object Types

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.

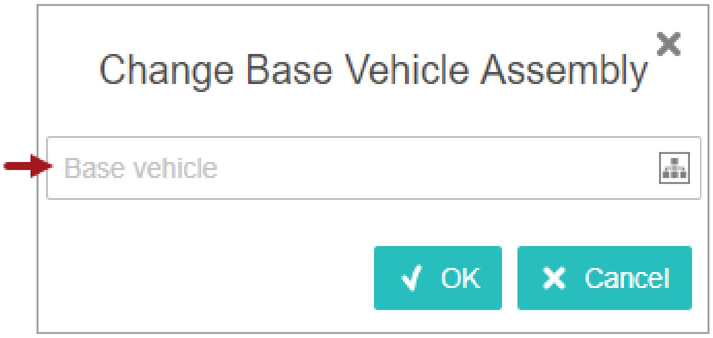
Important: If this key is not entered exactly in both places, then the business action will not properly function within the Web UI.

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.



Change Base Vehicle Assembly

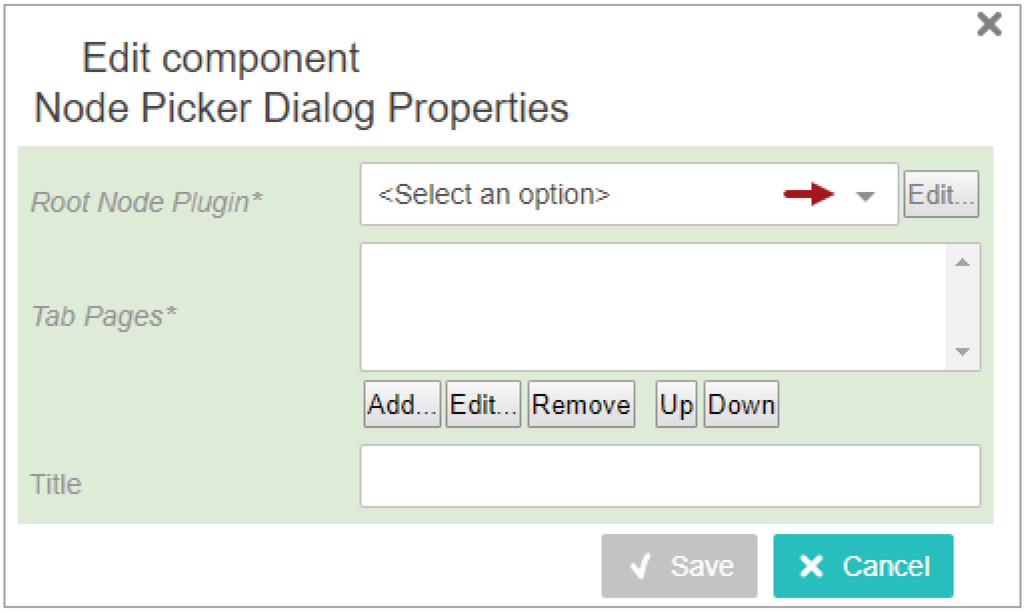
null



Change Base Vehicle Assembly

Base vehicle

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.



Edit component
Node Picker Dialog Properties

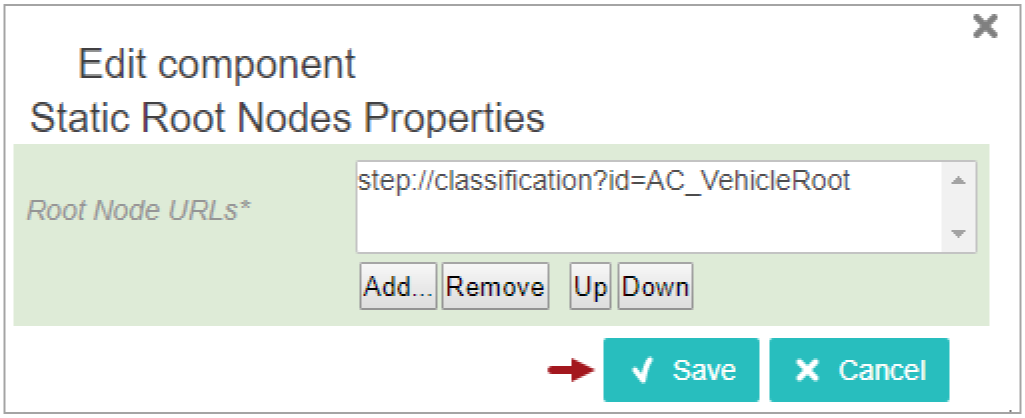
*Root Node Plugin** <Select an option>

*Tab Pages**

Title

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.

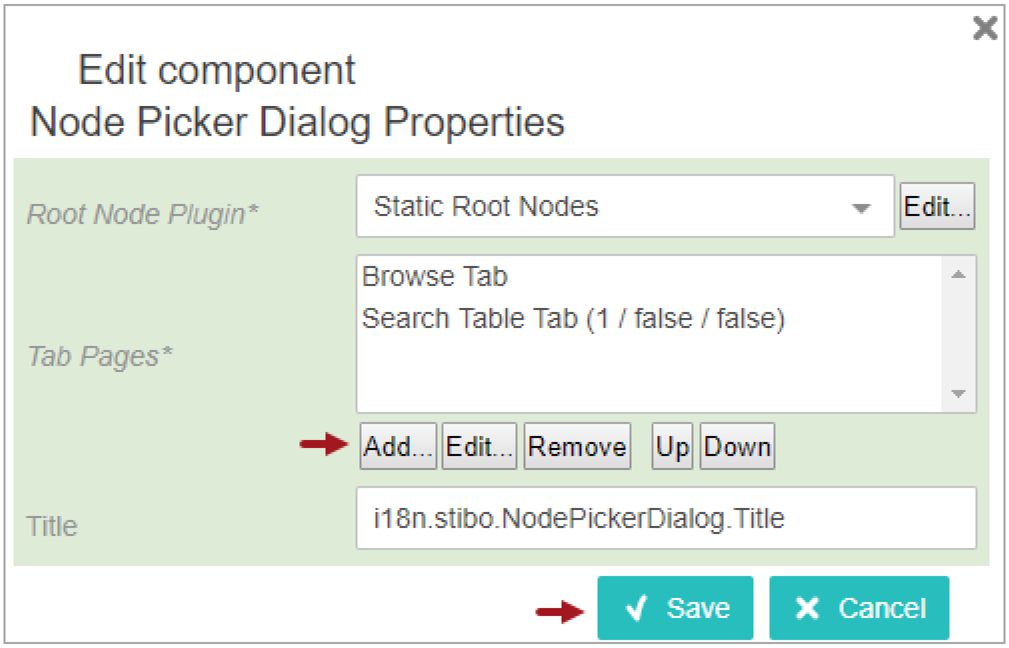
11. Click the **Add** button beneath the Root Nodes URLs parameter > select the root nodes that the user should be able to choose from when changing the part type (for this example AC_VehicleRoot is used) > click the **OK** button to close the dialog, and return to the 'Edit component' for 'Static Root Nodes Properties' dialog. Optionally, repeat this step to add additional nodes for a user to browse from when looking for a part type.



Edit component
Static Root Nodes Properties

Root Node URLs*: step://classification?id=AC_VehicleRoot

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.



Edit component
Node Picker Dialog Properties

Root Node Plugin*: Static Root Nodes

Tab Pages*:
Browse Tab
Search Table Tab (1 / false / false)

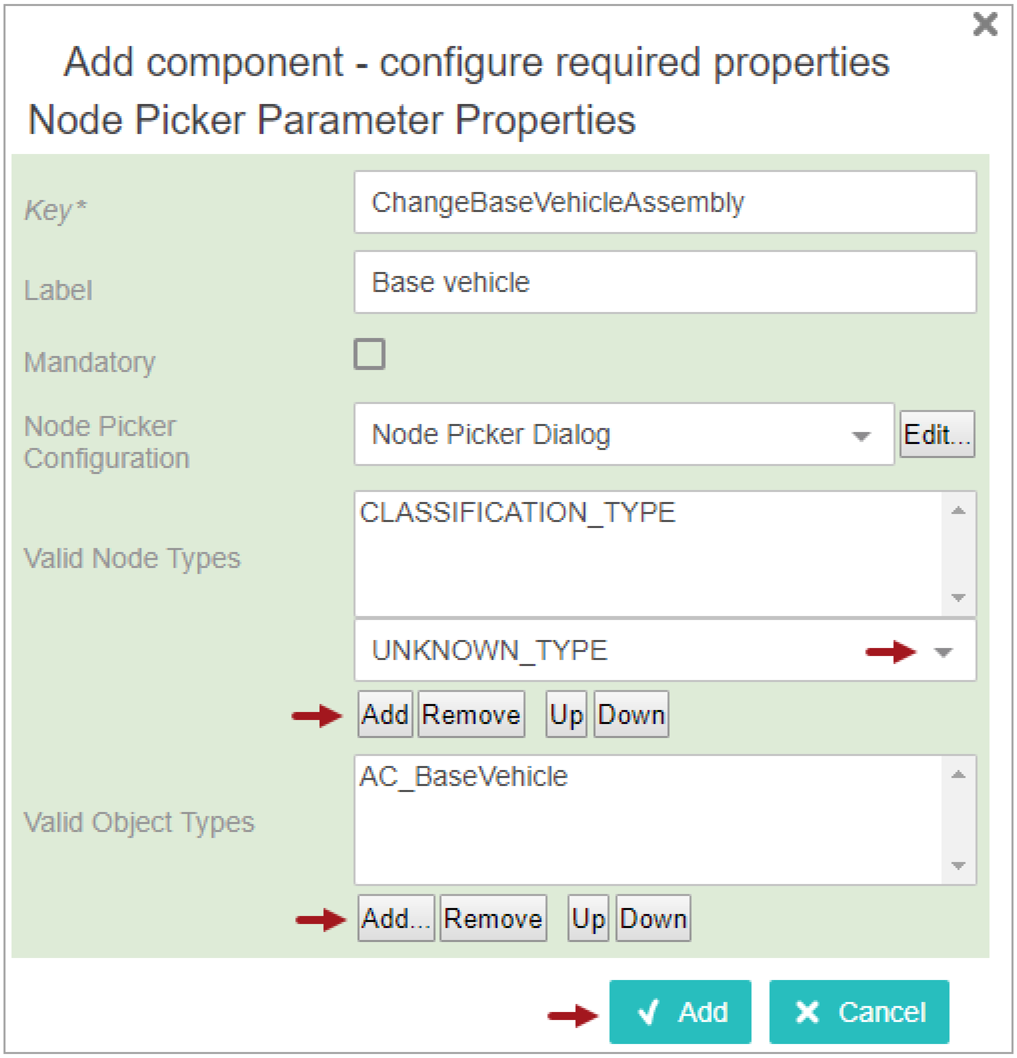
Title: i18n.stibo.NodePickerDialog.Title

Note: Leave the Title parameter blank and *after the configuration is saved* an i18n key will be populated (as shown above). For more information, see the **Localization** topic within the **Administration Portal** section of the **STEP Online Help**. Otherwise, text entered within the Title parameter will display in place of the default 'Select Node(s)' title for the dialog used to select the vehicle assembly.

14. Click the **Save** button, and the 'Add component - configure required properties' for the 'Node Picker Parameter Properties' dialog will display with the Node Picker Configuration parameter populated.

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

16. Click the **Add** button beneath the Valid Object Types parameter > select the desired valid object types > Click the **OK** button to return to the 'Add component - configure required properties' for the 'Node Picker Parameter Properties' dialog.



Add component - configure required properties

Node Picker Parameter Properties

Key* ChangeBaseVehicleAssembly

Label Base vehicle

Mandatory □

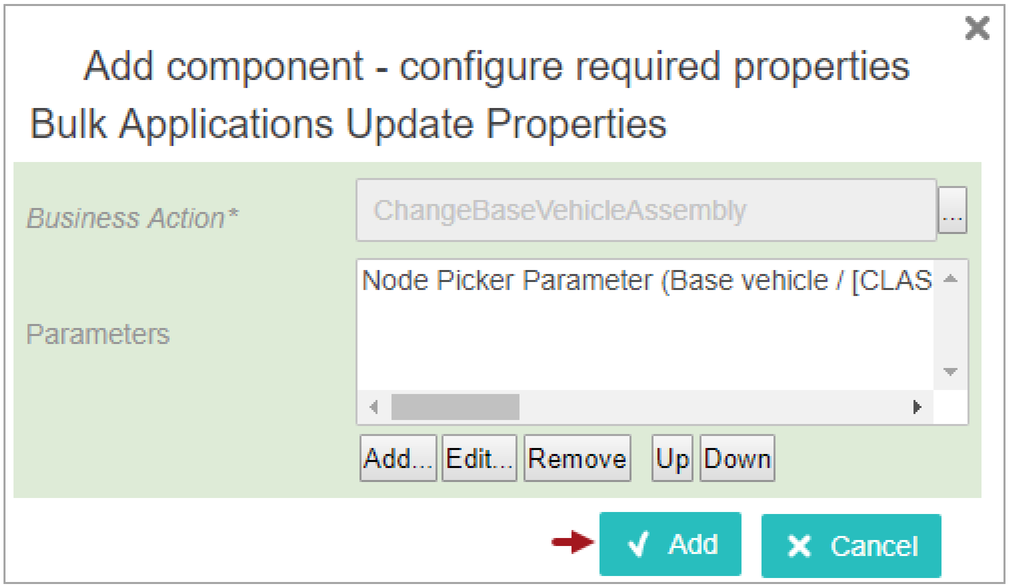
Node Picker Configuration Node Picker Dialog

Valid Node Types CLASSIFICATION_TYPE

UNKNOWN_TYPE

Valid Object Types AC_BaseVehicle

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



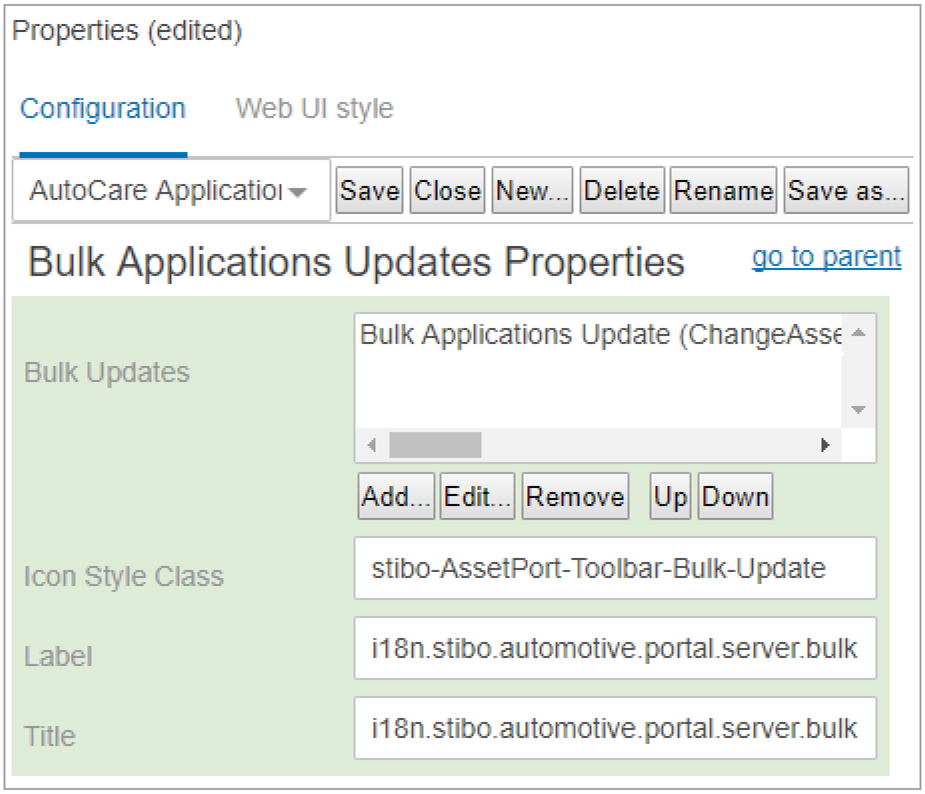
Add component - configure required properties

Bulk Applications Update Properties

Business Action*: ChangeBaseVehicleAssembly

Parameters: Node Picker Parameter (Base vehicle / [CLAS

18. Click the **Add** button, and the 'Bulk Applications Updates Properties' dialog will display with the newly added Bulk Update listed.



Properties (edited)

Bulk Applications Updates Properties

Bulk Updates: Bulk Applications Update (ChangeAsse

Icon Style Class: stibo-AssetPort-Toolbar-Bulk-Update

Label: i18n.stibo.automotive.portal.server.bulk

Title: i18n.stibo.automotive.portal.server.bulk

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

To use the newly configured business action, see the **Using the Configured Change Assembly Business Action in Web UI** section of the **Business Action: Change Assembly** topic.

Business Action: Change Part Type

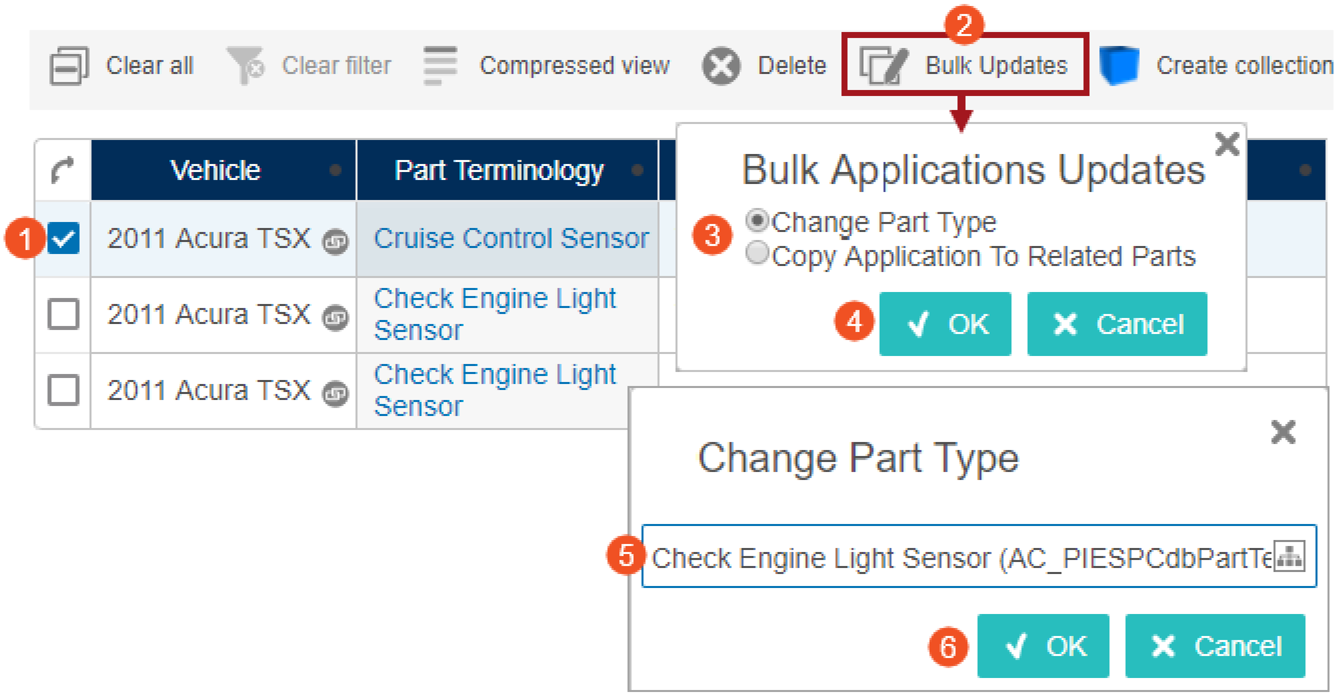
This automotive business action allows users to change the part type of one or more existing applications by clicking a Bulk Updates action button within an Application Editor. However, it cannot change the part type of a missing application, and thus any missing applications selected when the bulk update is run will be ignored. For more information, see the **Missing Coverage** topic within this guide. Setup is required within both STEP Workbench and Web UI for the action to be available to users.

This section addresses steps necessary when:

- Using the Configured Change Part Type Business Action in Web UI
- Configuring the Change Part Type Business Action in Workbench
- Configuring the Change Part Type Business Action in Web UI

Using the Configured Change Part Type Business Action in Web UI

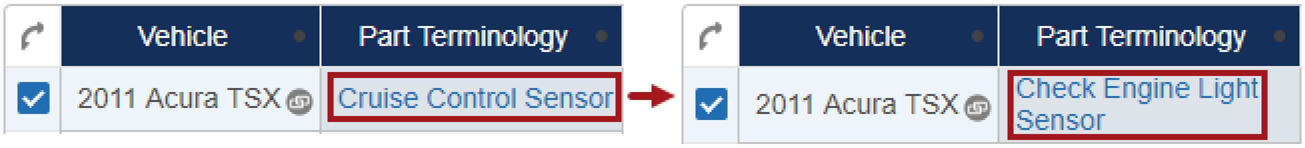
Once configured, a Bulk Updates button within an Application Editor can be used to change the part type for an existing application. Below are the steps for using the business action in Web UI.


Clear all
Clear filter
Compressed view
Delete
Bulk Updates
Create collection
Vehicle
Part Terminology
1
2
3
4
5
6
☑ 2011 Acura TSX | Cruise Control Sensor
□ 2011 Acura TSX | Check Engine Light Sensor
□ 2011 Acura TSX | Check Engine Light Sensor
Bulk Applications Updates
● Change Part Type
○ Copy Application To Related Parts
✓ OK
✕ Cancel
Change Part Type
Check Engine Light Sensor (AC_PIESPCdbPartTe
✓ OK
✕ Cancel

Note: If only one bulk update is configured, then the Bulk Applications Updates dialog will not display.

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

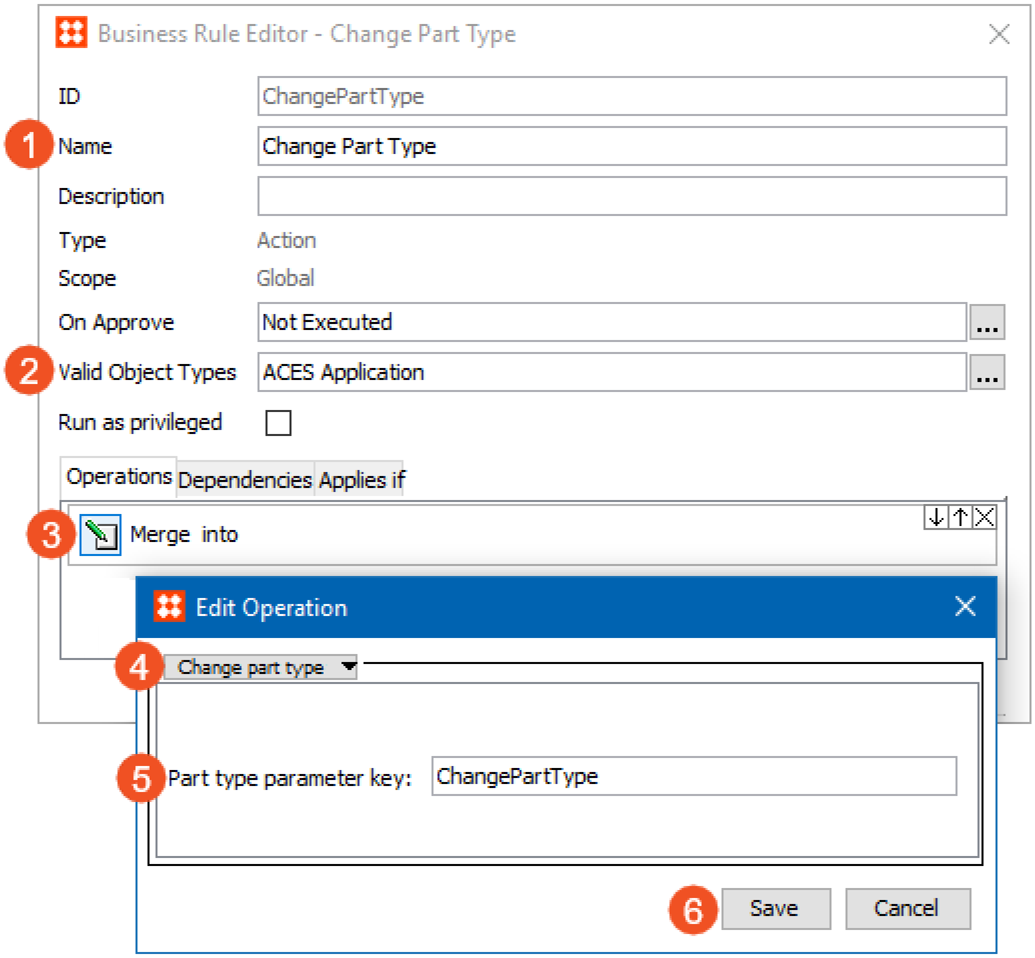


	Vehicle	Part Terminology
☑	2011 Acura TSX	Cruise Control Sensor

	Vehicle	Part Terminology
☑	2011 Acura TSX	Check Engine Light Sensor

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.



Business Rule Editor - Change Part Type

ID: ChangePartType

Name: Change Part Type

Description:

Type: Action

Scope: Global

On Approve: Not Executed

Valid Object Types: ACES Application

Run as privileged: □

Part type parameter key: ChangePartType

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 Editor 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 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 part type** operation, and the parameter 'Part Type 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.

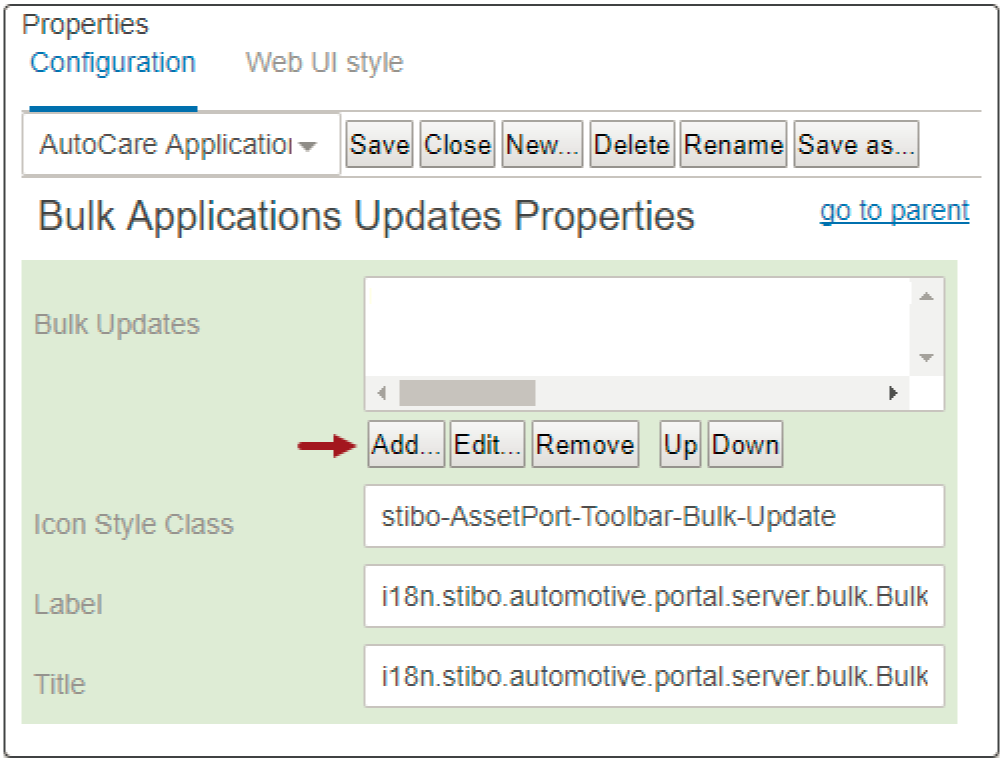
6. Click the **Save** button and continue to the next topic, **Configuring the Change Part Type Business Action in Web UI**.

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 Editor 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).



Bulk Applications Updates Properties

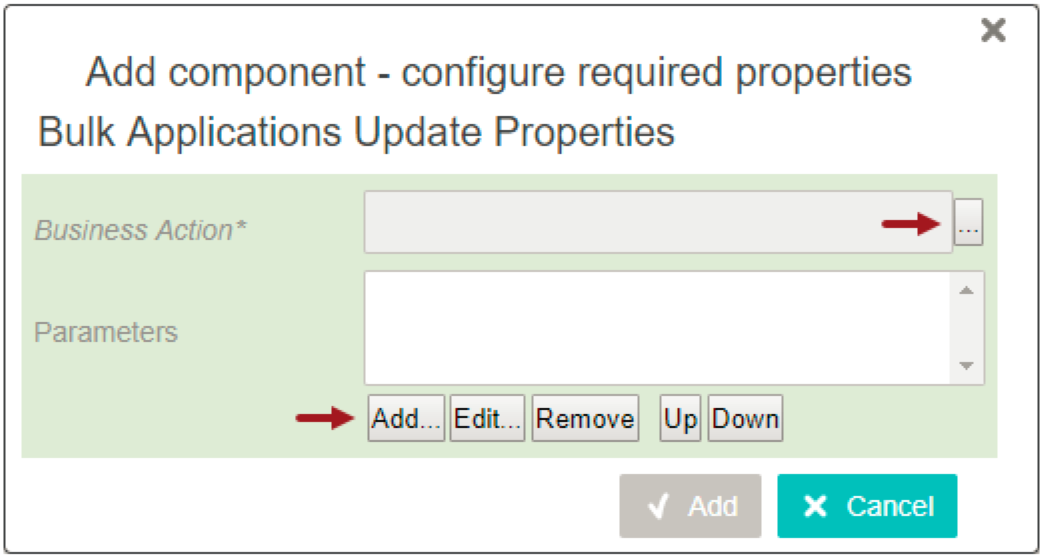
Bulk Updates

Icon Style Class: stibo-AssetPort-Toolbar-Bulk-Update

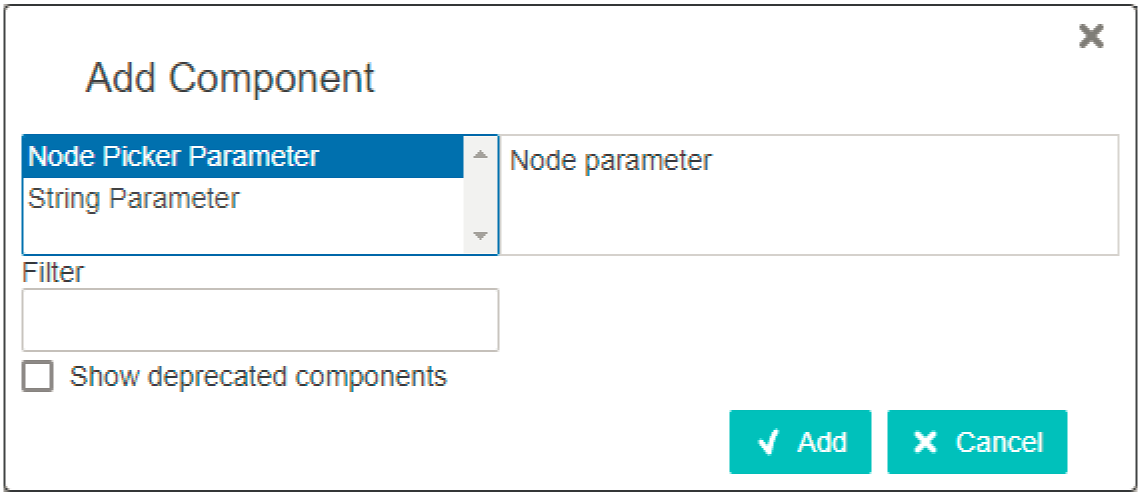
Label: i18n.stibo.automotive.portal.server.bulk.Bulk

Title: i18n.stibo.automotive.portal.server.bulk.Bulk

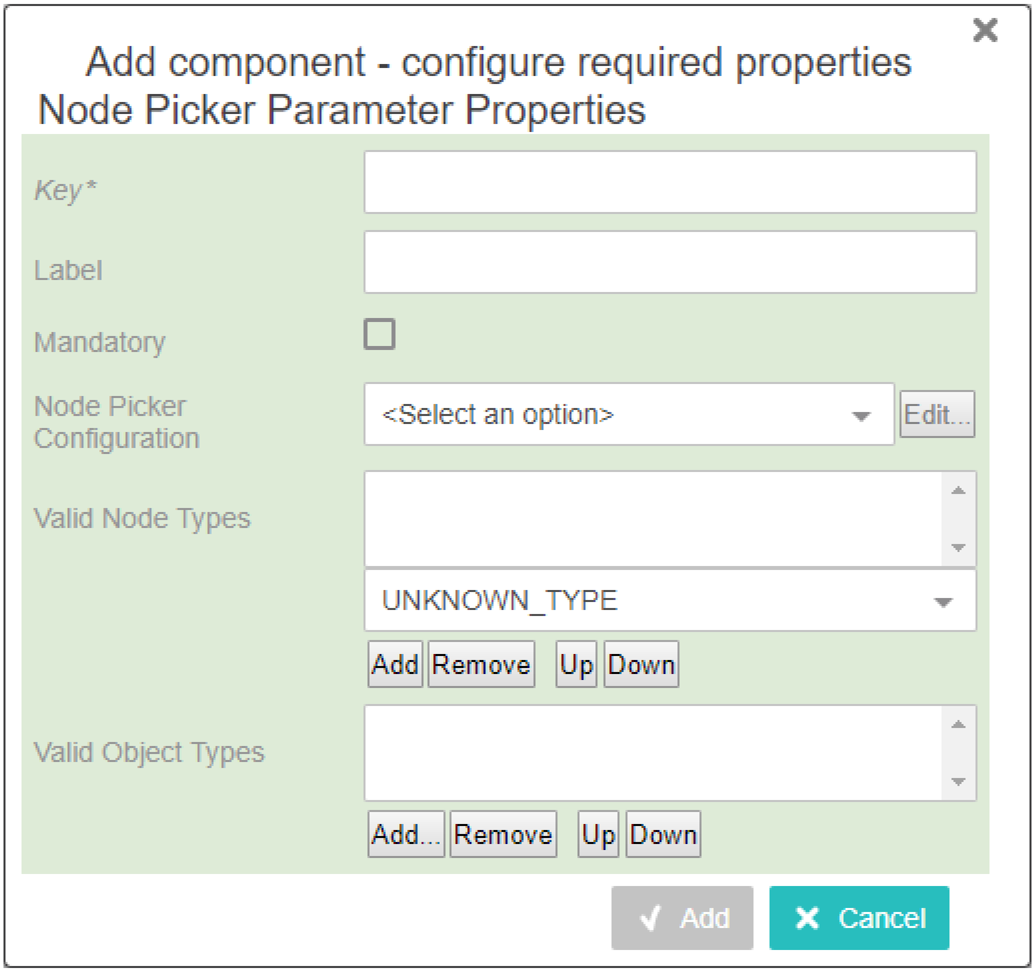
3. Click the **Add** button beneath the Bulk Updates parameter, and the 'Add component - configure required properties' for the 'Bulk Applications Update Properties' dialog will display (as shown below).


Add component - configure required properties
Bulk Applications Update Properties
Business Action*
...
Parameters
Add...
Edit...
Remove
Up
Down
Add
Cancel

4. Click the ellipsis button **(...)** next to the Business Action parameter > select the business action previously created > click the **OK** button to close the dialog, and return to the Bulk Applications Update Properties dialog. The selected action is displayed in the Business Action parameter.
5. Click the **Add** button beneath the Parameters field, and the Add Component dialog will display (as shown below).



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



Add component - configure required properties
Node Picker Parameter Properties

Key*

Label

Mandatory □

Node Picker Configuration <Select an option>

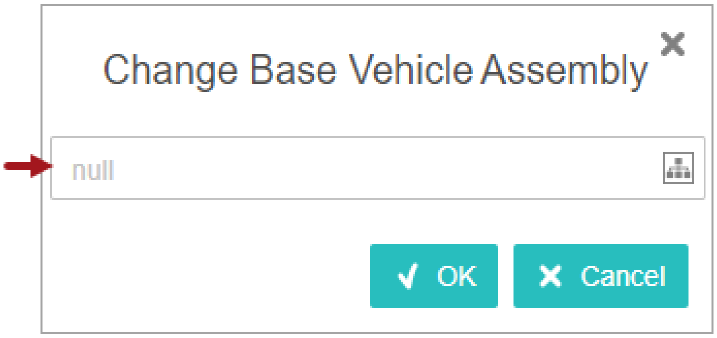
Valid Node Types UNKNOWN_TYPE

Valid Object Types

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.

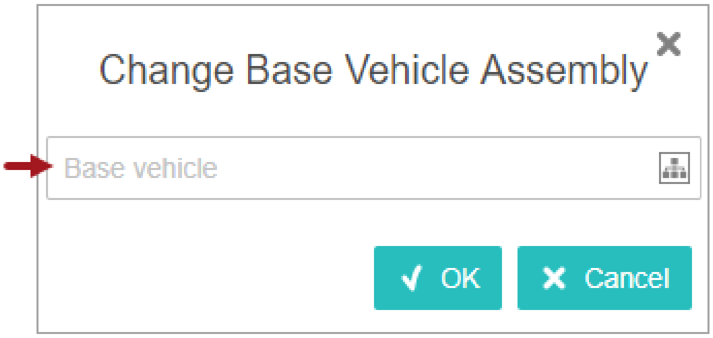
Important: If this key is not entered exactly in both places, then the business action will not properly function within the Web UI.

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.



Change Base Vehicle Assembly

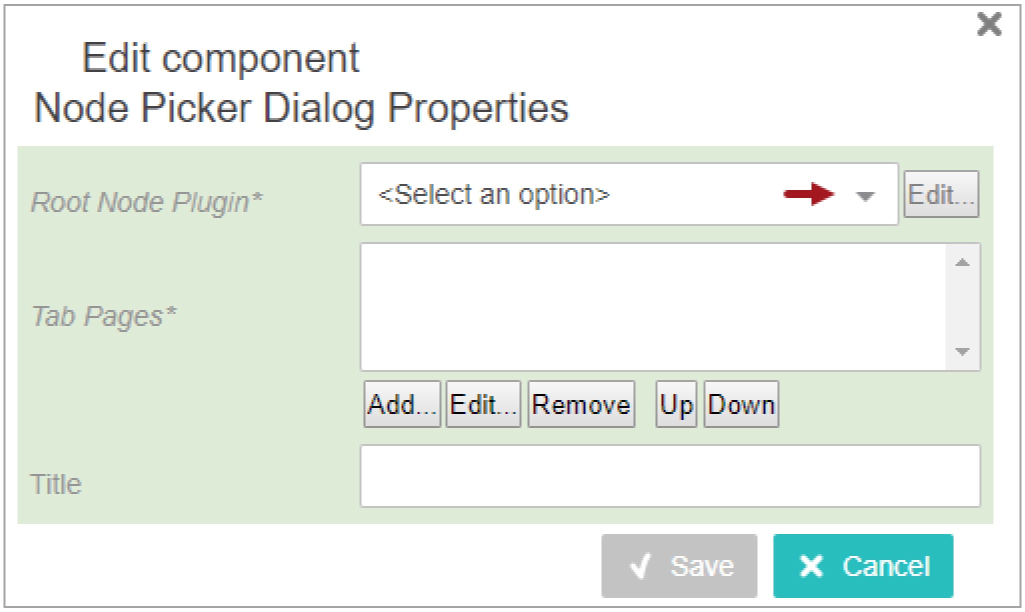
null



Change Base Vehicle Assembly

Base vehicle

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.



Edit component
Node Picker Dialog Properties

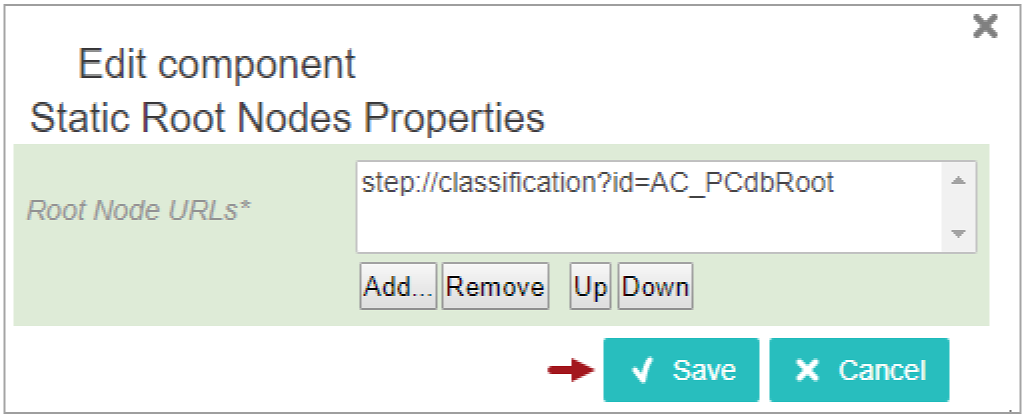
*Root Node Plugin** <Select an option>

*Tab Pages**

Title

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.

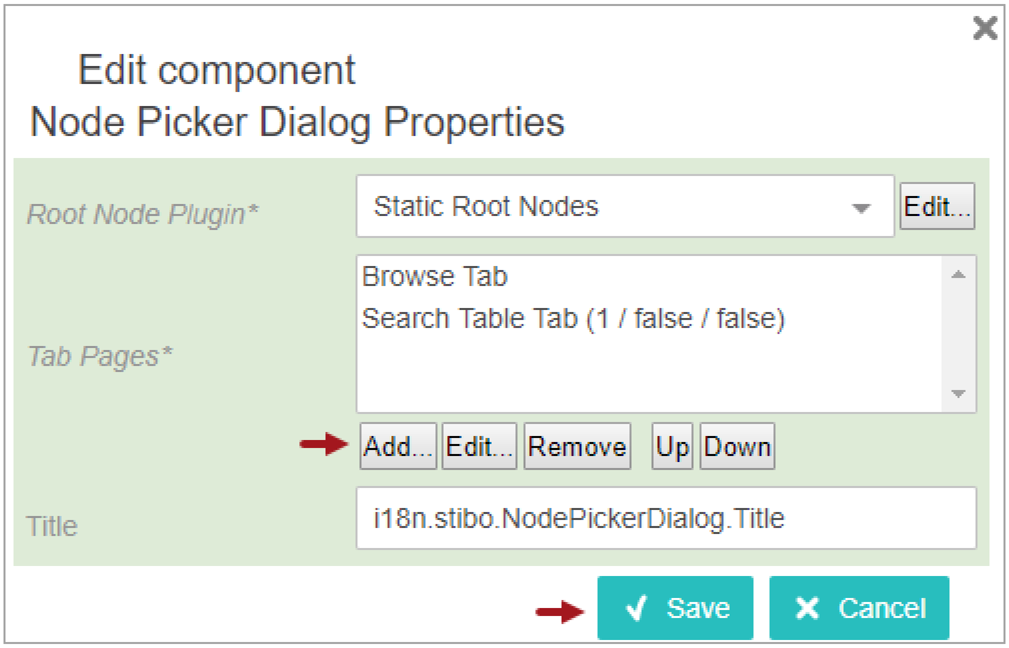
11. Click the **Add** button beneath the Root Nodes URLs parameter > select the Root Nodes that the user should be able to choose from when changing the part type (For this example AC_PCdbRoot is used) > click the **OK** button to close the dialog, and return to the 'Edit component' for 'Static Root Nodes Properties' dialog. Optionally repeat this step to add additional nodes for a user to browse from when looking for a part type.



Edit component
Static Root Nodes Properties

Root Node URLs*: step://classification?id=AC_PCdbRoot

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.



Edit component
Node Picker Dialog Properties

Root Node Plugin*: Static Root Nodes

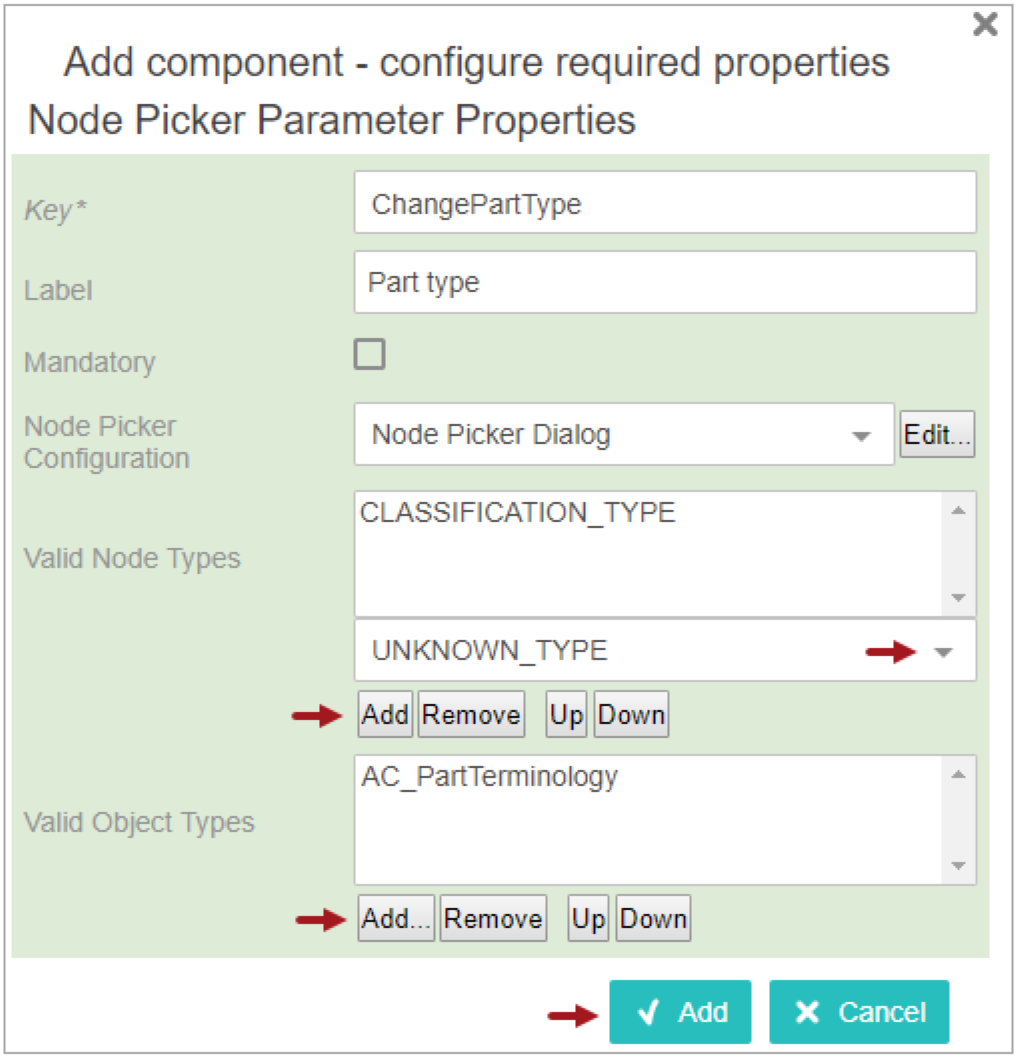
Tab Pages*:
Browse Tab
Search Table Tab (1 / false / false)

Title: i18n.stibo.NodePickerDialog.Title

Note: Leave the Title parameter blank and *after the configuration is saved* an i18n key will be populated (as shown above). For more information, see the **Localization** topic within the **Administration Portal** section of the **STEP Online Help**. Otherwise, text entered within the Title parameter will display in place of the default 'Select Node(s)' title for the dialog used to select the vehicle assembly.

14. Click the **Save** button, and the 'Add component - configure required properties' for the 'Node Picker Parameter Properties' dialog will display with the Node Picker Configuration parameter populated.

15. Use the dropdown located beneath the Valid Node Types parameter to select **CLASSIFICATION_TYPE** > click the **Add** button beneath the Valid Node Types parameter so that CLASSIFICATION_TYPE is displayed within the Valid Node Types parameter (as shown below).
16. Click the **Add** button beneath the Valid Object Types parameter > select the desired valid object types (For this example AC_PartTerminology is used)> click the **OK** button to return to the 'Add component - configure required properties' for the 'Node Picker Parameter Properties' dialog.



Add component - configure required properties

Node Picker Parameter Properties

Key* ChangePartType

Label Part type

Mandatory □

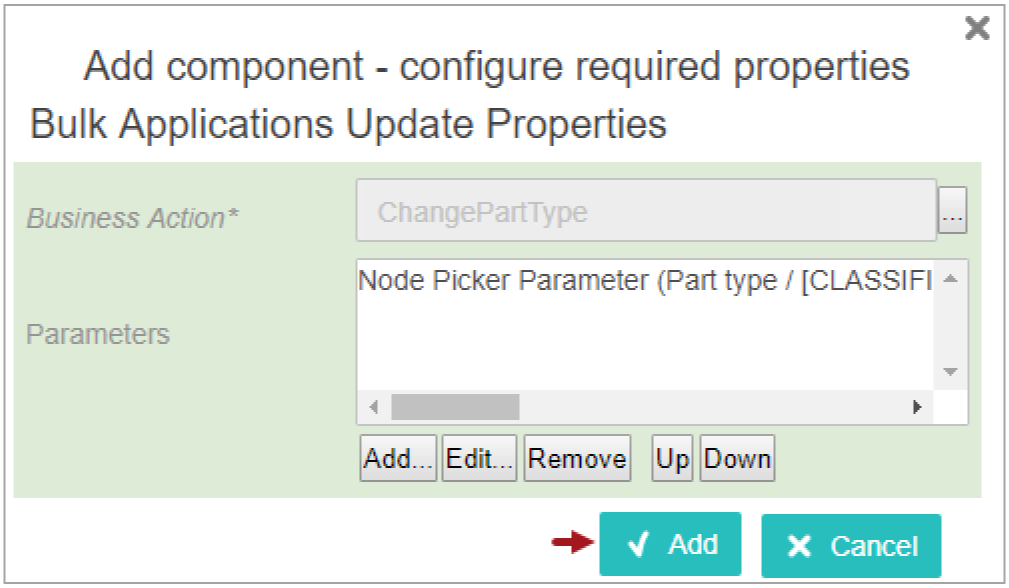
Node Picker Configuration Node Picker Dialog

Valid Node Types CLASSIFICATION_TYPE

UNKNOWN_TYPE

Valid Object Types AC_PartTerminology

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



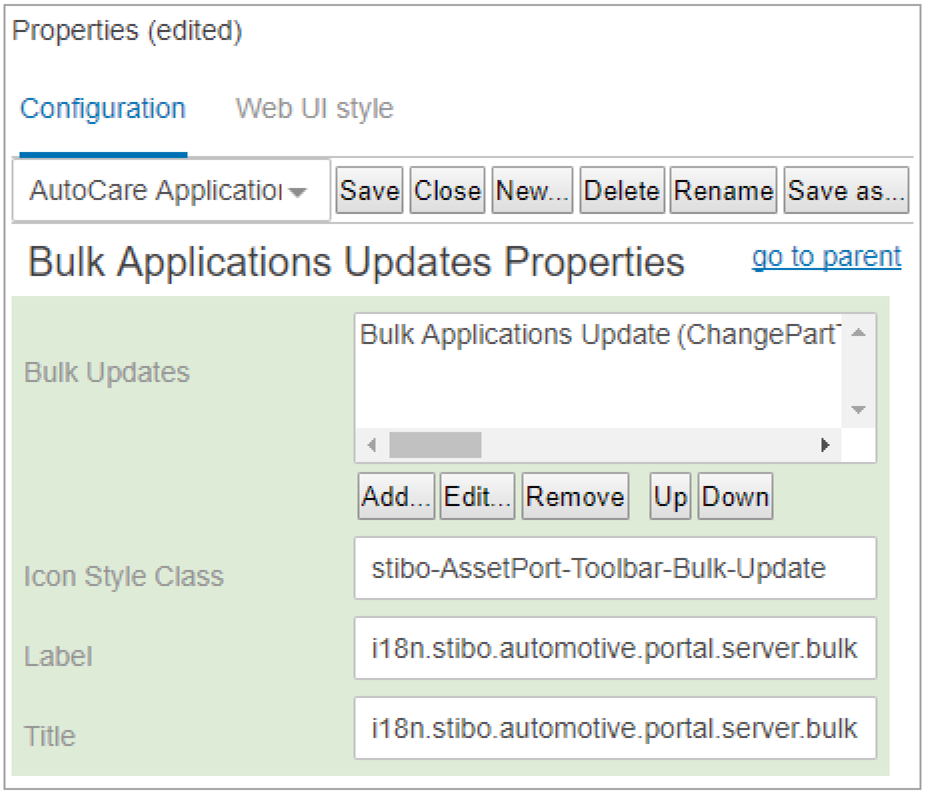
Add component - configure required properties

Bulk Applications Update Properties

Business Action* ChangePartType

Parameters Node Picker Parameter (Part type / [CLASSIFI

18. Click the **Add** button, and the 'Bulk Applications Updates Properties' dialog will display with the newly added Bulk Update listed.



Properties (edited)

Bulk Applications Updates Properties

Bulk Updates Bulk Applications Update (ChangePart

Icon Style Class stibo-AssetPort-Toolbar-Bulk-Update

Label i18n.stibo.automotive.portal.server.bulk

Title i18n.stibo.automotive.portal.server.bulk

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

To use the newly configured business action, see the **Using the Configured Change Part Type Business Action in Web UI** section of the **Business Action: Change Part Type** topic.

Business Action: Change Part

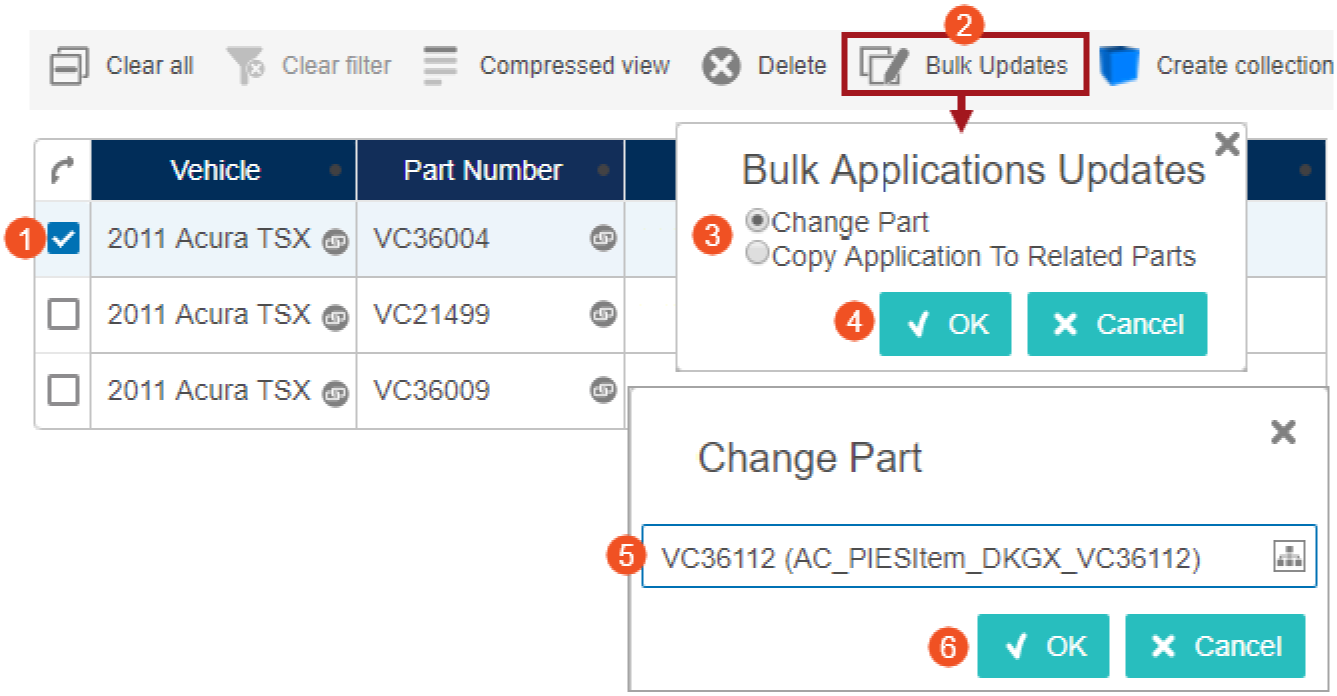
This automotive business action allows users to change the part of an existing application by clicking a Bulk Updates action button within an Application Editor. However, it cannot change the part of a missing application, and thus any missing applications selected when the bulk update is run will be ignored. For more information, see the **Missing Coverage** topic within this guide. Setup is required within both STEP Workbench and Web UI for the action to be available to users.

This section addresses steps necessary when:

- Using the Configured Change Part Business Action in Web UI
- Configuring the Change Part Business Action in Workbench
- Configuring the Change Part Business Action in Web UI

Using the Configured Change Part Business Action in Web UI

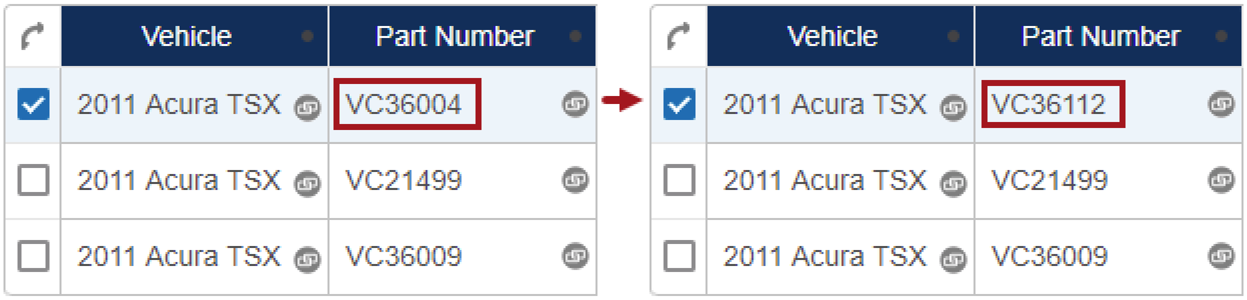
Once configured, a Bulk Updates button within an Application Editor can be used to change the part for an existing application. Below are the steps for using the business action in Web UI.



Note: If only one bulk update is configured, then the Bulk Applications Updates dialog will not display.

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

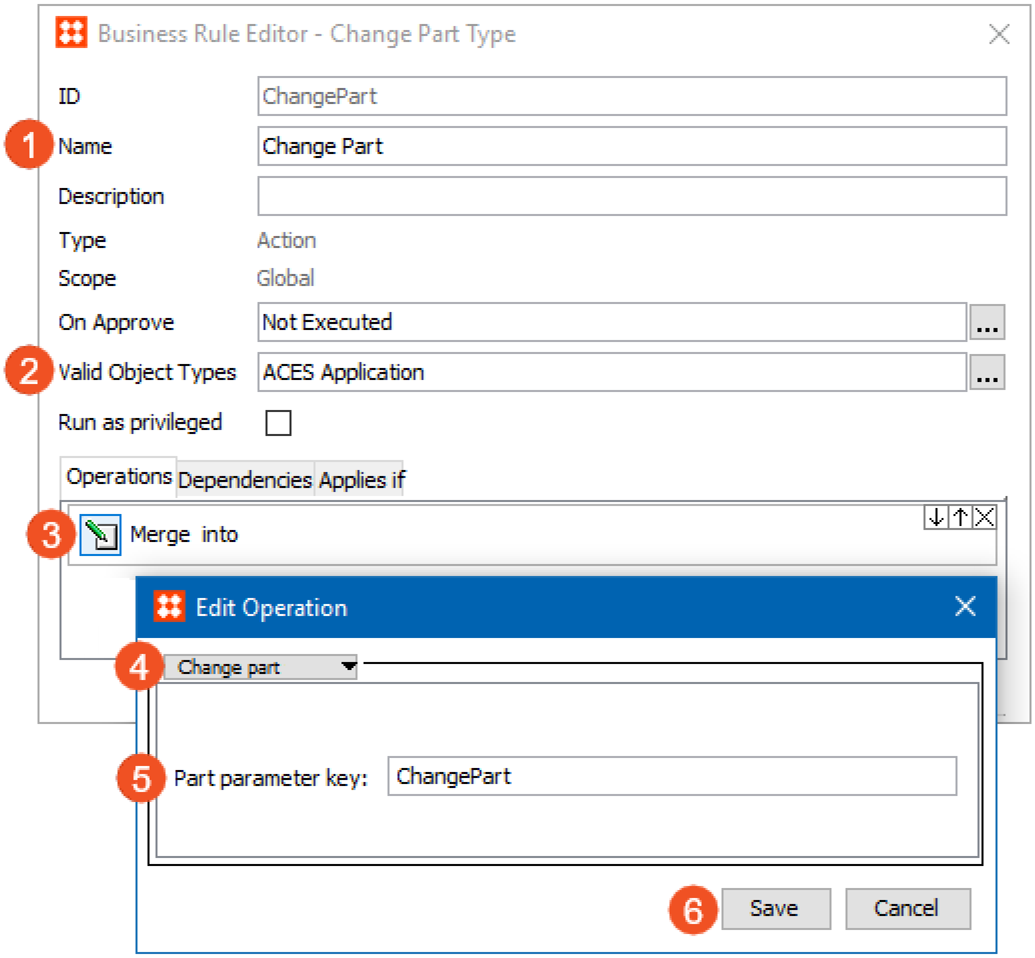


	Vehicle	Part Number
☑	2011 Acura TSX	VC36004
□	2011 Acura TSX	VC21499
□	2011 Acura TSX	VC36009

	Vehicle	Part Number
☑	2011 Acura TSX	VC36112
□	2011 Acura TSX	VC21499
□	2011 Acura TSX	VC36009

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.



Business Rule Editor - Change Part Type

ID: ChangePart

Name: Change Part

Description:

Type: Action

Scope: Global

On Approve: Not Executed

Valid Object Types: ACES Application

Run as privileged: □

Merge into

Edit Operation

Change part

Part parameter key: ChangePart

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 Editor 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 Change Part Business Action in Web UI** section of the **Business Action: Change Part** 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 Part** operation, and the parameter 'Part 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 Part Business Action in Web UI**.

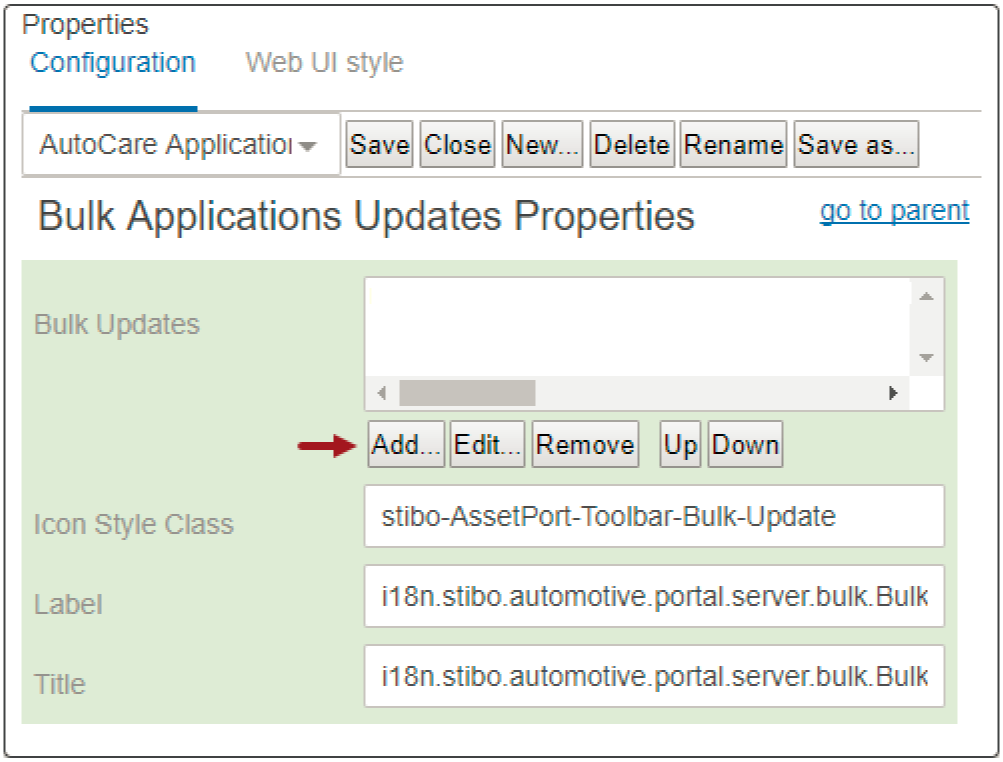
Configuring the Change Part Business Action in Web UI

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 Editor 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).



AutoCare Applicatioı

Bulk Applications Updates Properties

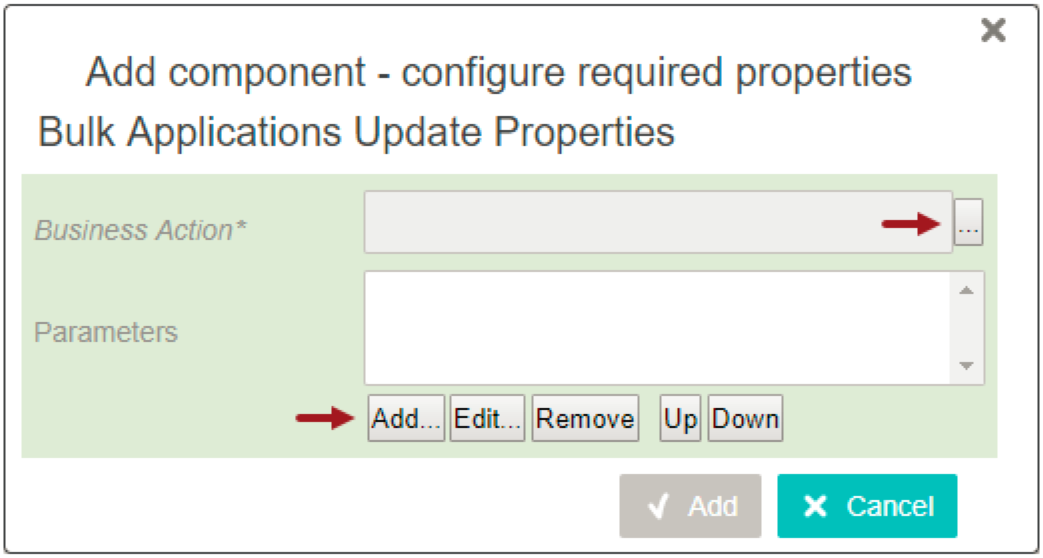
Bulk Updates

Icon Style Class: stibo-AssetPort-Toolbar-Bulk-Update

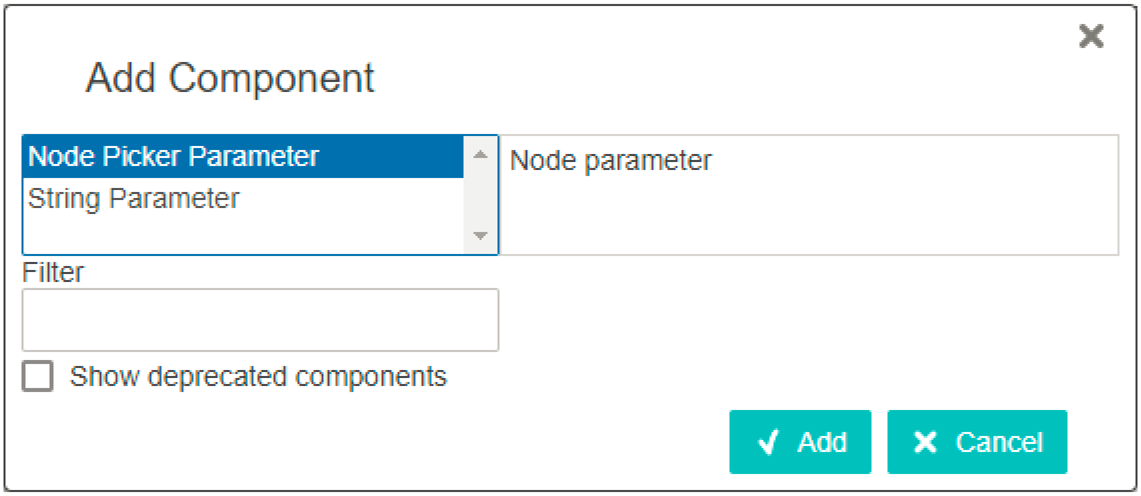
Label: i18n.stibo.automotive.portal.server.bulk.Bulk

Title: i18n.stibo.automotive.portal.server.bulk.Bulk

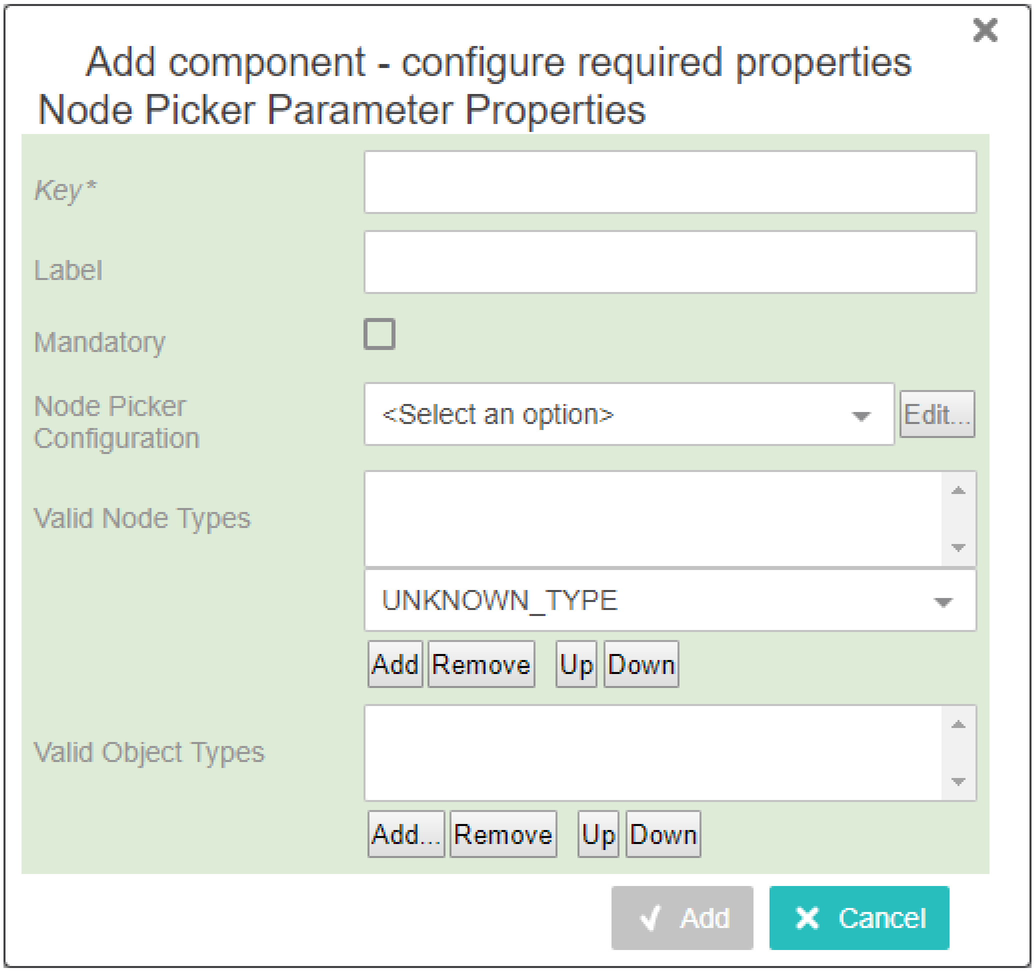
3. Click the **Add** button beneath the Bulk Updates parameter, and the 'Add component - configure required properties' for the 'Bulk Applications Update Properties' dialog will display (as shown below).


Add component - configure required properties
Bulk Applications Update Properties
Business Action*
...
Parameters
Add...
Edit...
Remove
Up
Down
✓ Add
✕ Cancel

4. Click the ellipsis button **(...)** next to the Business Action parameter > select the business action previously created > click the **OK** button to close the dialog, and return to the Bulk Applications Update Properties dialog. The selected action is displayed in the Business Action parameter.
5. Click the **Add** button beneath the Parameters field, and the Add Component dialog will display (as shown below).



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



Add component - configure required properties

Node Picker Parameter Properties

Key*

Label

Mandatory □

Node Picker Configuration <Select an option>

Valid Node Types

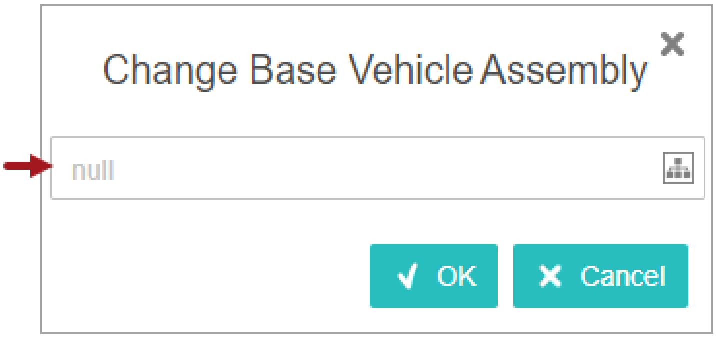
UNKNOWN_TYPE

Valid Object Types

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 Business Action in Workbench** topic. This is the only required parameter.

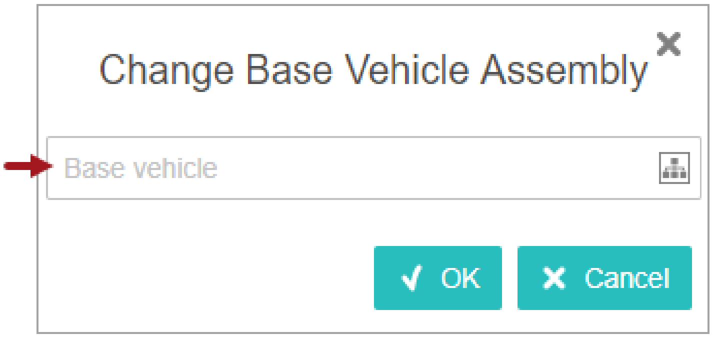
Important: If this key is not entered exactly in both places, then the business action will not properly function within the Web UI.

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.



Change Base Vehicle Assembly

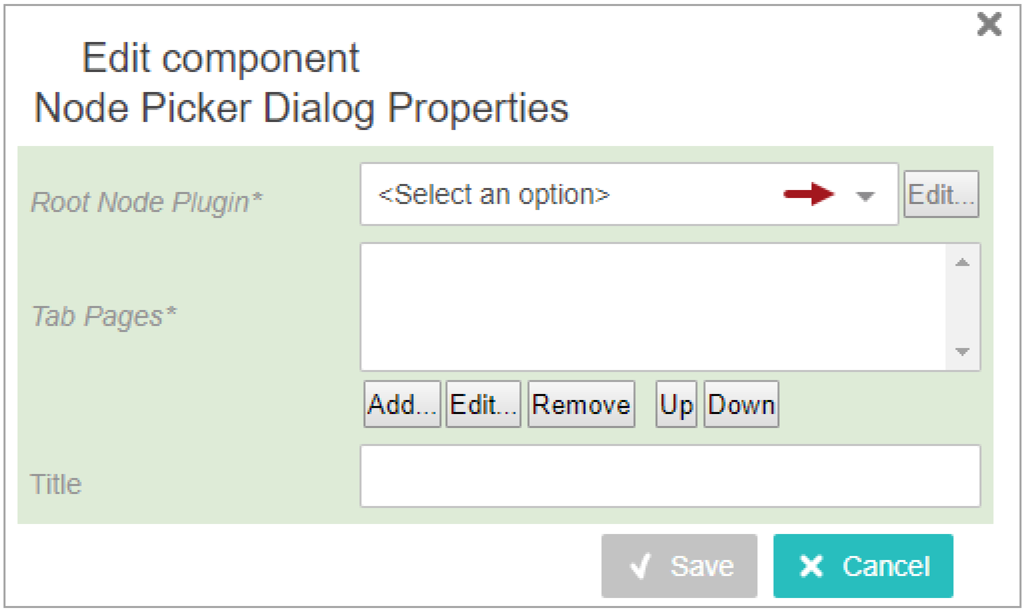
null



Change Base Vehicle Assembly

Base vehicle

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.

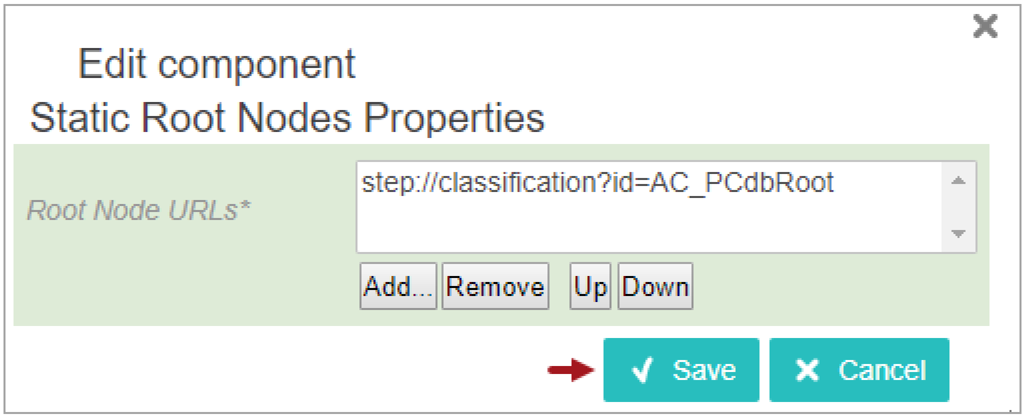


Edit component
Node Picker Dialog Properties

*Root Node Plugin**	<Select an option>
*Tab Pages**	
Title	

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.

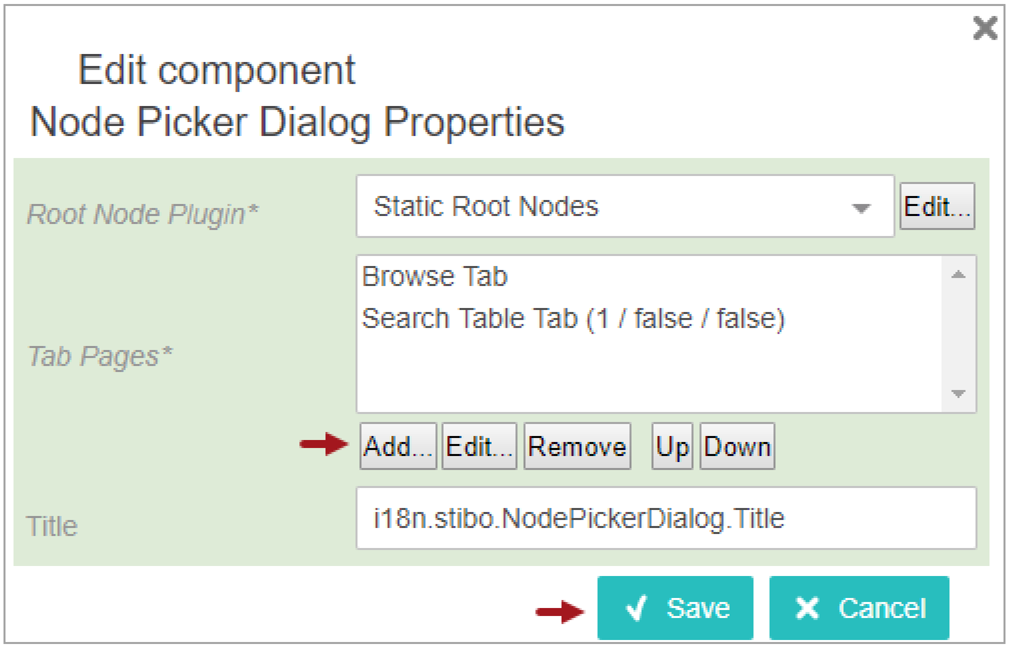
11. Click the **Add** button beneath the Root Nodes URLs parameter > select the Root Nodes that the user should be able to choose from when changing the part (For this example AC_PCdbRoot is used) > click the **OK** button to close the dialog, and return to the 'Edit component' for 'Static Root Nodes Properties' dialog. Optionally repeat this step to add additional nodes for a user to browse from when looking for a part type.



Edit component
Static Root Nodes Properties

Root Node URLs*: step://classification?id=AC_PCdbRoot

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.



Edit component
Node Picker Dialog Properties

Root Node Plugin*: Static Root Nodes

Tab Pages*: Browse Tab
Search Table Tab (1 / false / false)

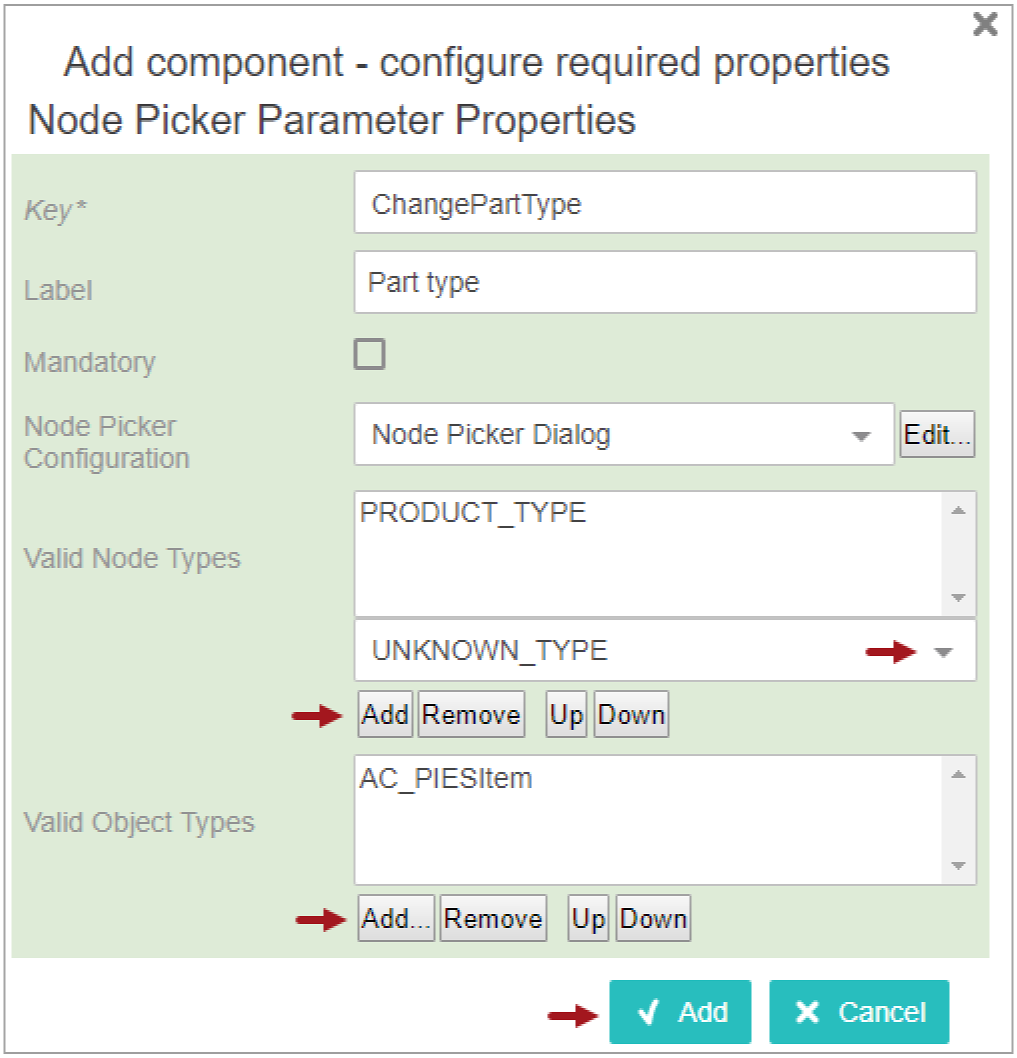
Title: i18n.stibo.NodePickerDialog.Title

Note: Leave the Title parameter blank and *after the configuration is saved* an i18n key will be populated (as shown above). For more information, see the **Localization** topic within the **Administration Portal** section of the **STEP Online Help**. Otherwise, text entered within the Title parameter will display in place of the default 'Select Node(s)' title for the dialog used to select the vehicle assembly.

14. Click the **Save** button, and the 'Add component - configure required properties' for the 'Node Picker Parameter Properties' dialog will display with the Node Picker Configuration parameter populated.

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

16. Click the **Add** button beneath the Valid Object Types parameter > select the desired valid object types (For this example AC_PIESItem is used) > click the **OK** button to return to the 'Add component - configure required properties' for the 'Node Picker Parameter Properties' dialog.

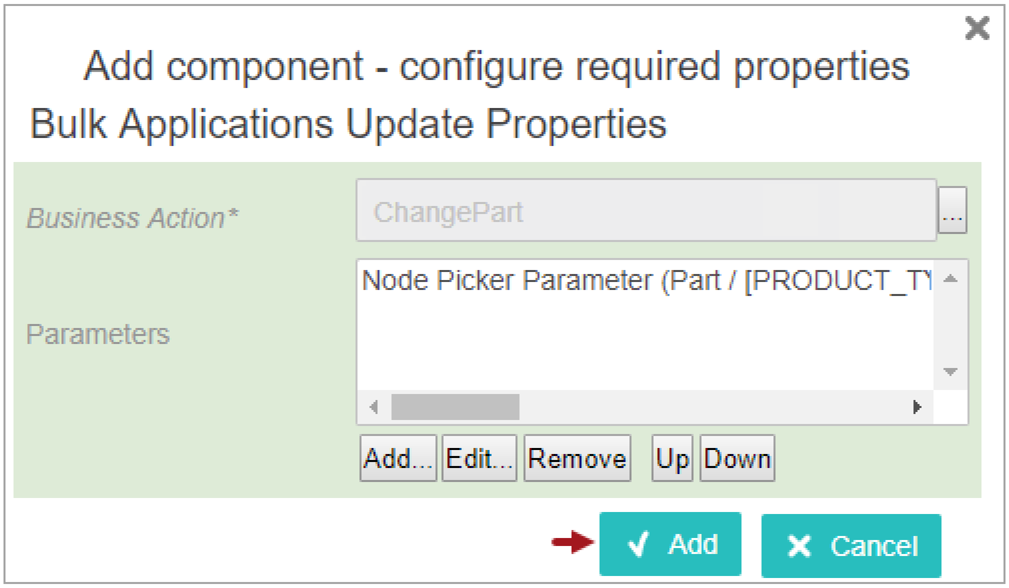


Add component - configure required properties

Node Picker Parameter Properties

*Key**	ChangePartType
Label	Part type
Mandatory	□
Node Picker Configuration	Node Picker Dialog
Valid Node Types	PRODUCT_TYPE
	UNKNOWN_TYPE
Valid Object Types	AC_PIESItem

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



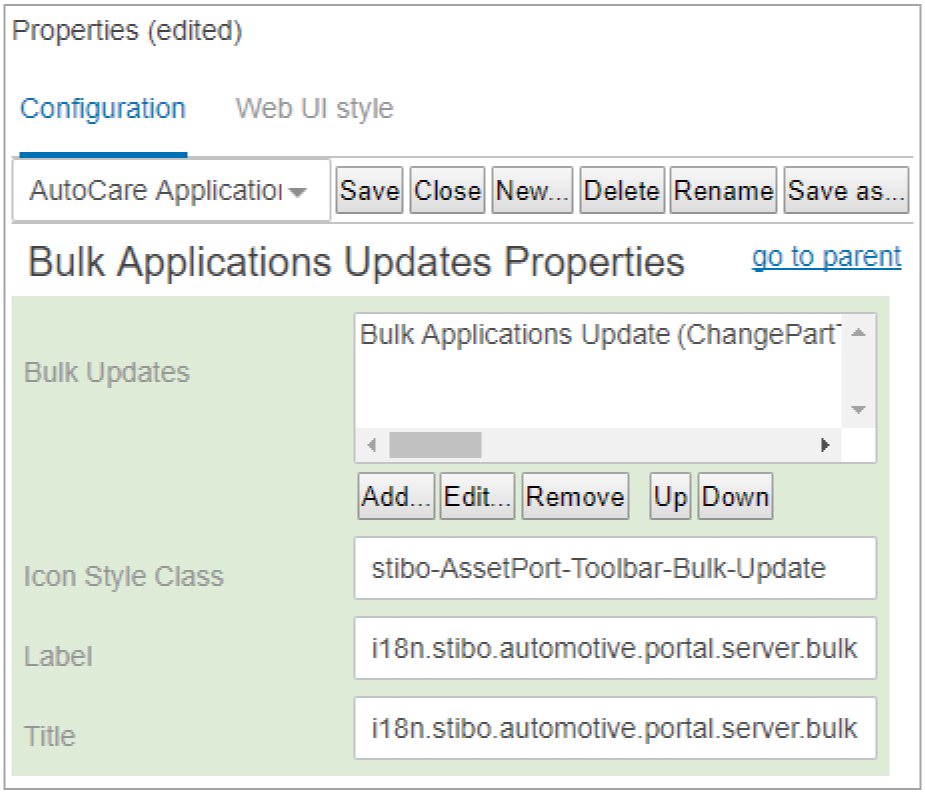
Add component - configure required properties

Bulk Applications Update Properties

Business Action*: ChangePart

Parameters: Node Picker Parameter (Part / [PRODUCT_TY

18. Click the **Add** button, and the 'Bulk Applications Updates Properties' dialog will display with the newly added Bulk Update listed.



Properties (edited)

Bulk Applications Updates Properties

Bulk Updates: Bulk Applications Update (ChangePart

Icon Style Class: stibo-AssetPort-Toolbar-Bulk-Update

Label: i18n.stibo.automotive.portal.server.bulk

Title: i18n.stibo.automotive.portal.server.bulk

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

To use the newly configured business action, see the **Using the Configured Change Part Business Action in Web UI** section of the **Business Action: Change Part** topic.

Business Action: Copy Application to Other Assembly

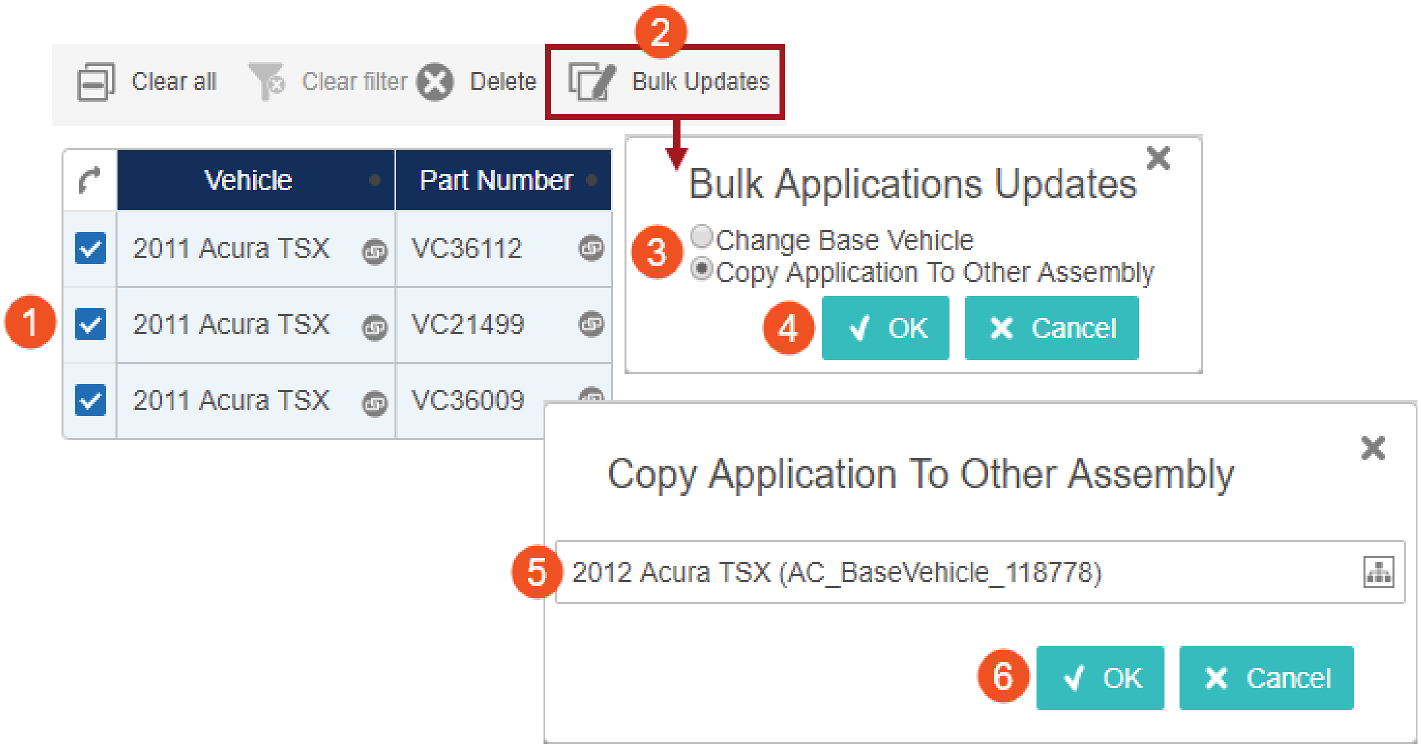
This automotive business action allows users to use a Bulk Updates action button within an Application Editor to copy one or more existing applications to another assembly / vehicle. However, it will not copy an assembly / vehicle that is missing an application, and thus any missing applications selected when the bulk update is run will be ignored. For more information, see the **Missing Coverage** topic within this guide. Setup is required within both STEP Workbench and Web UI for the action to be available to users.

This section addresses steps necessary when:

- Using the Configured Copy Application to Other Assembly Business Action in Web UI
- Configuring the Copy Application to Other Assembly Business Action in Workbench
- Configuring the Copy Application to Other Assembly Business Action in Web UI

Using the Configured Copy Application to Other Assembly Business Action in Web UI

Once configured, a Bulk Updates button within an Application Editor 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.



	Vehicle	Part Number
☑	2011 Acura TSX	VC36112
☑	2011 Acura TSX	VC21499
☑	2011 Acura TSX	VC36009

Bulk Applications Updates

○ Change Base Vehicle
● Copy Application To Other Assembly

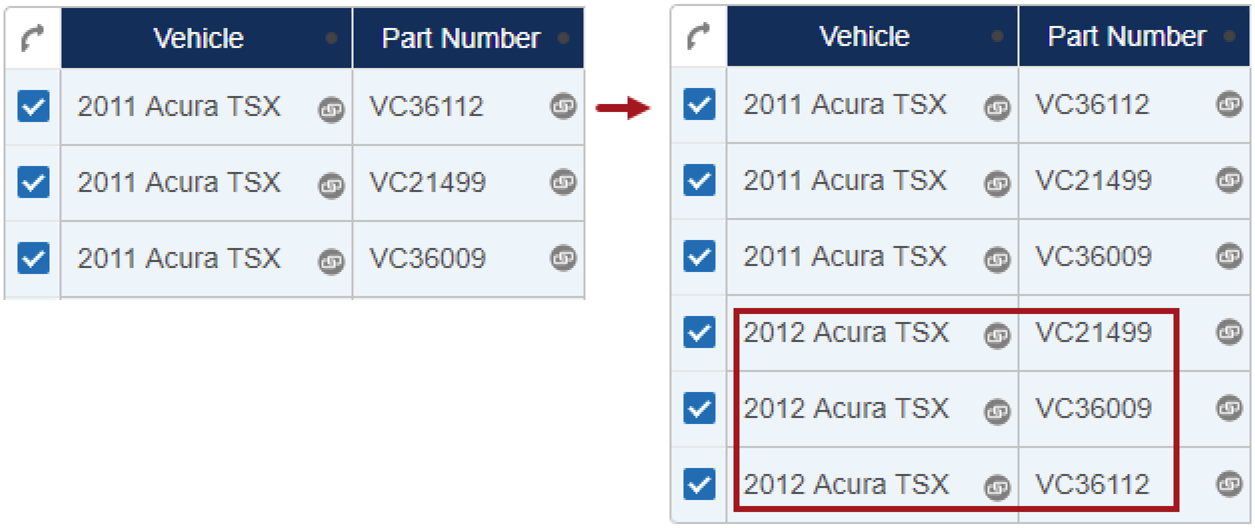
Copy Application To Other Assembly

2012 Acura TSX (AC_BaseVehicle_118778)

Note: If only one bulk update is configured, then the Bulk Applications Updates dialog will not display.

1. Within the configured Application Editor, 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 Editor results table.

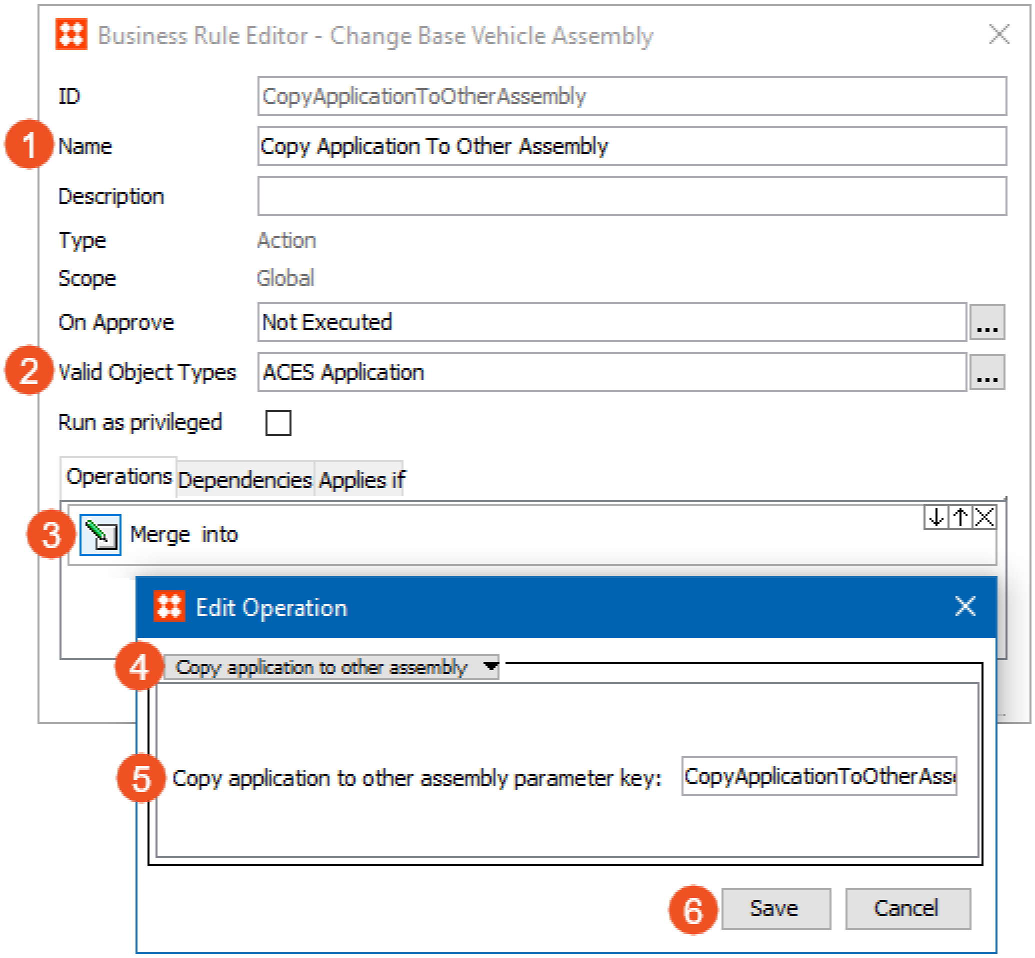


	Vehicle	Part Number
☑	2011 Acura TSX	VC36112
☑	2011 Acura TSX	VC21499
☑	2011 Acura TSX	VC36009

	Vehicle	Part Number
☑	2011 Acura TSX	VC36112
☑	2011 Acura TSX	VC21499
☑	2011 Acura TSX	VC36009
☑	2012 Acura TSX	VC21499
☑	2012 Acura TSX	VC36009
☑	2012 Acura TSX	VC36112

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.



Business Rule Editor - Change Base Vehicle Assembly

ID: CopyApplicationToOtherAssembly

Name: Copy Application To Other Assembly

Description:

Type: Action

Scope: Global

On Approve: Not Executed

Valid Object Types: ACES Application

Run as privileged □

5 Copy application to other assembly parameter key: CopyApplicationToOtherAss

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 Editor 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 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 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.

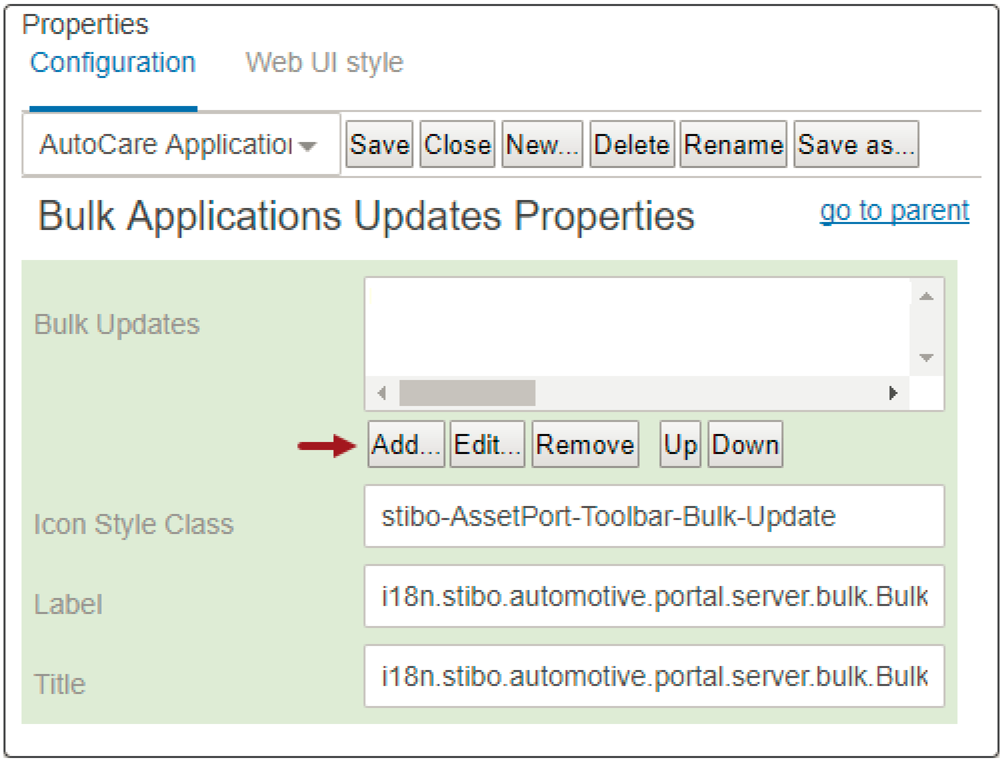
6. Click the **Save** button and continue to the next topic, **Copy Application to Other Assembly Business Action in Web UI**.

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 Editor 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).



Bulk Applications Updates Properties

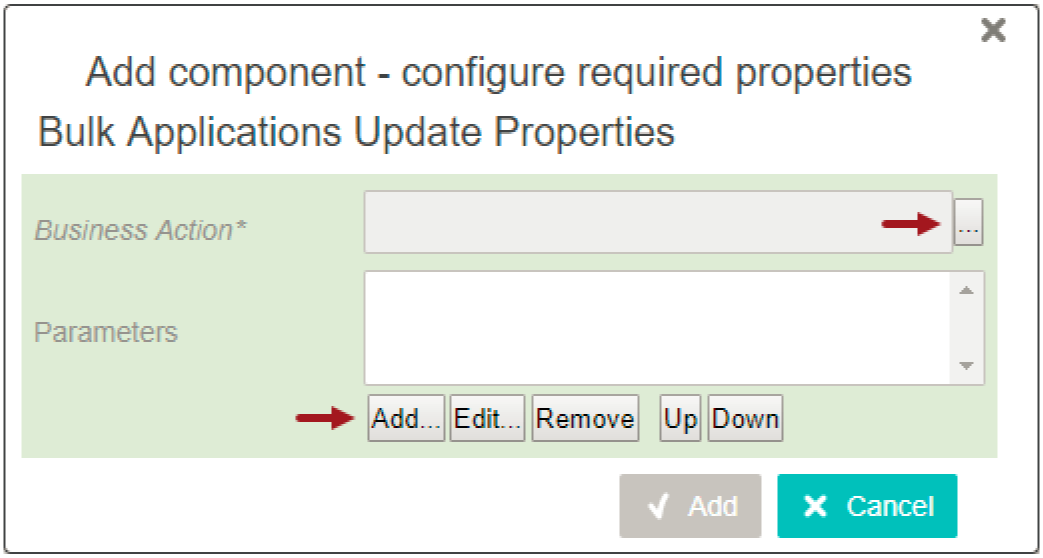
Bulk Updates

Icon Style Class: stibo-AssetPort-Toolbar-Bulk-Update

Label: i18n.stibo.automotive.portal.server.bulk.Bulk

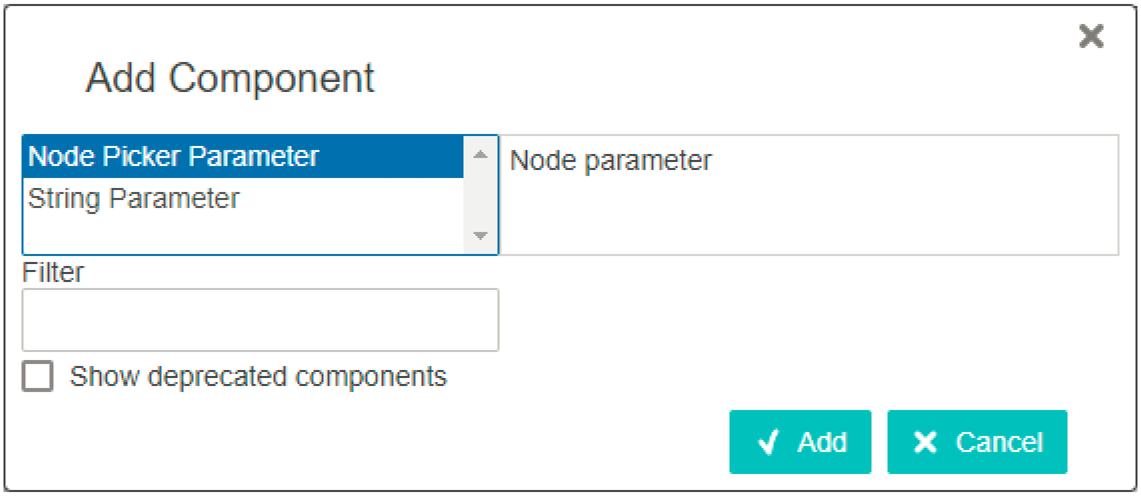
Title: i18n.stibo.automotive.portal.server.bulk.Bulk

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).

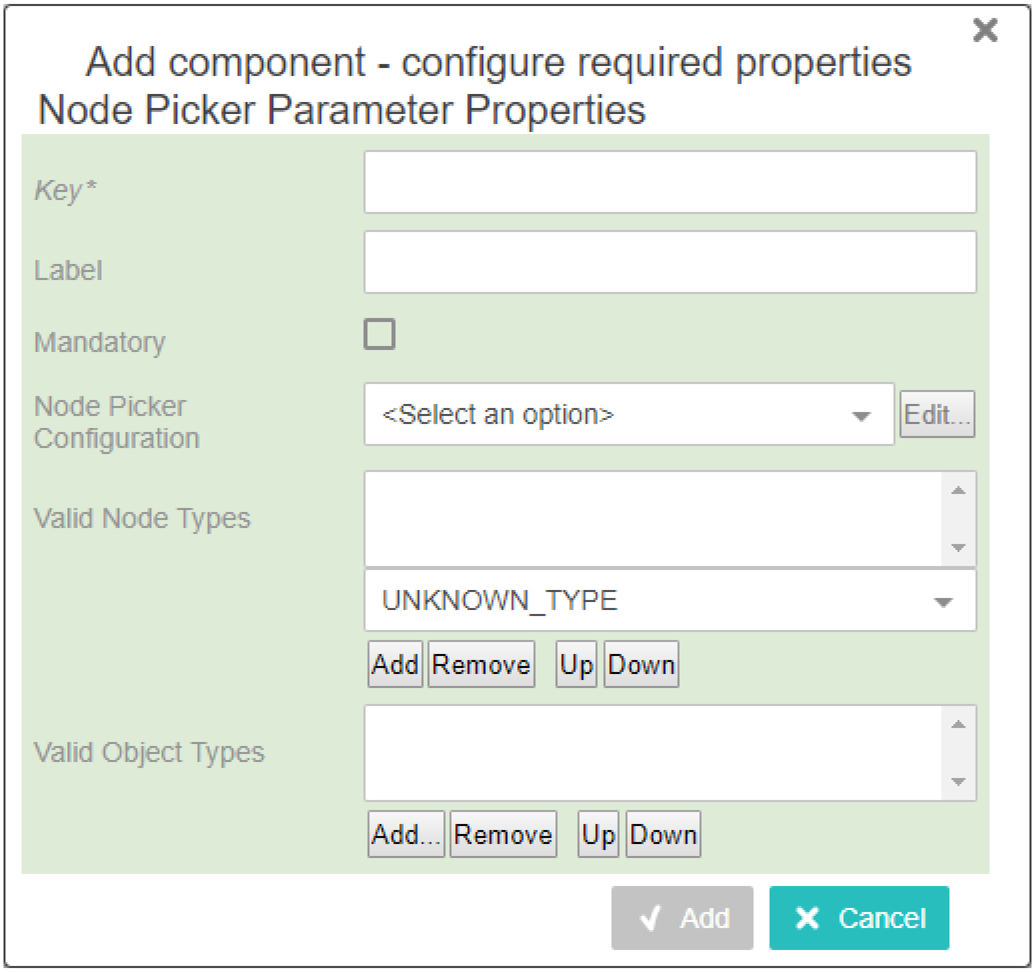


×
Add component - configure required properties
Bulk Applications Update Properties
Business Action*
...
Parameters
Add...
Edit...
Remove
Up
Down
✓ Add
✕ Cancel

4. Click the ellipsis button **(...)** next to the Business Action parameter > select the business action previously created > click the **OK** button to close the dialog, and return to the Bulk Applications Update Properties dialog. The selected action is displayed in the Business Action parameter.
5. Click the **Add** button beneath the Parameters field, and the Add Component dialog will display (as shown below).



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



Add component - configure required properties
Node Picker Parameter Properties

Key*

Label

Mandatory □

Node Picker Configuration <Select an option> Edit...

Valid Node Types

UNKNOWN_TYPE

Add Remove Up Down

Valid Object Types

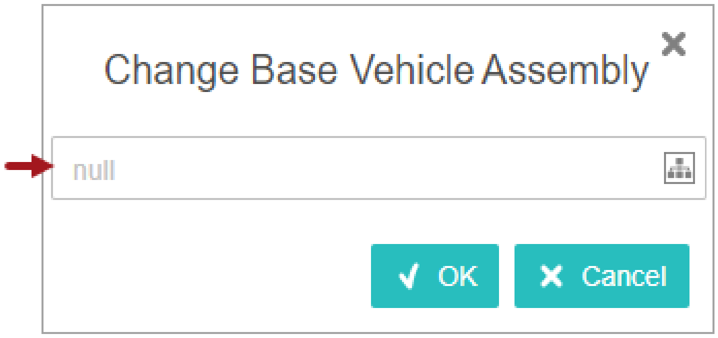
Add... Remove Up Down

✓ Add ✕ Cancel

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.

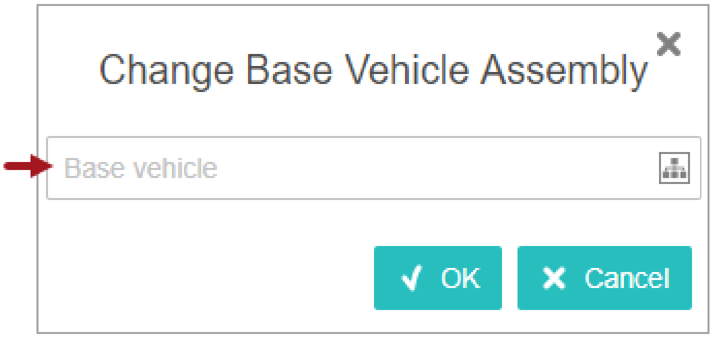
Important: If this key is not entered exactly in both places, then the business action will not properly function within the Web UI.

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.



Change Base Vehicle Assembly

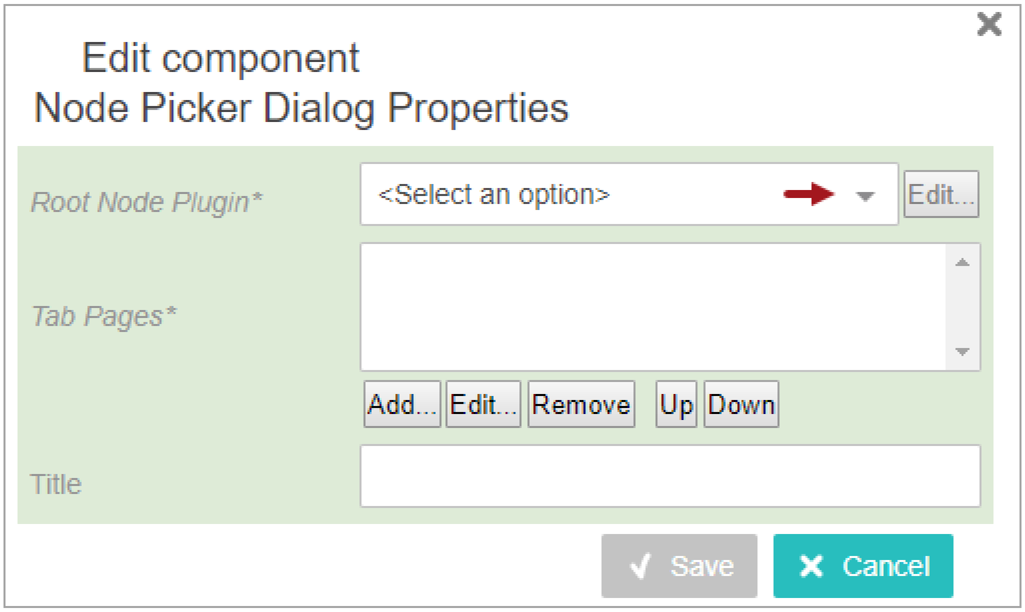
null



Change Base Vehicle Assembly

Base vehicle

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.



Edit component
Node Picker Dialog Properties

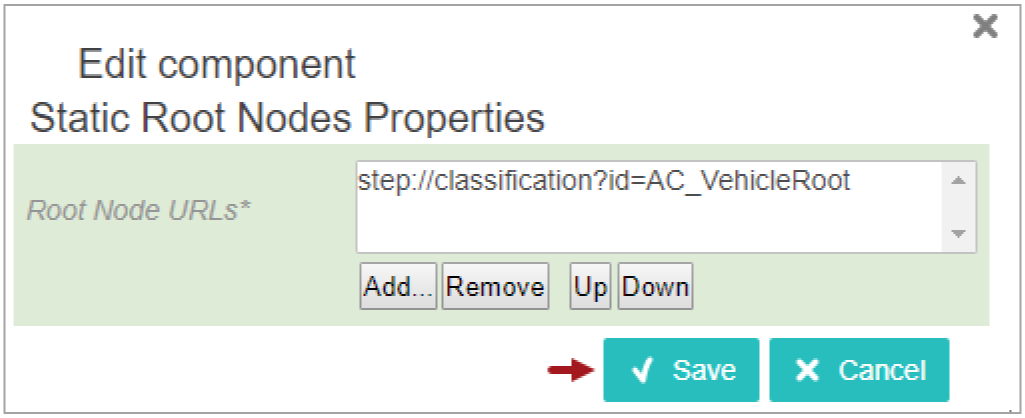
*Root Node Plugin** <Select an option>

*Tab Pages**

Title

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.

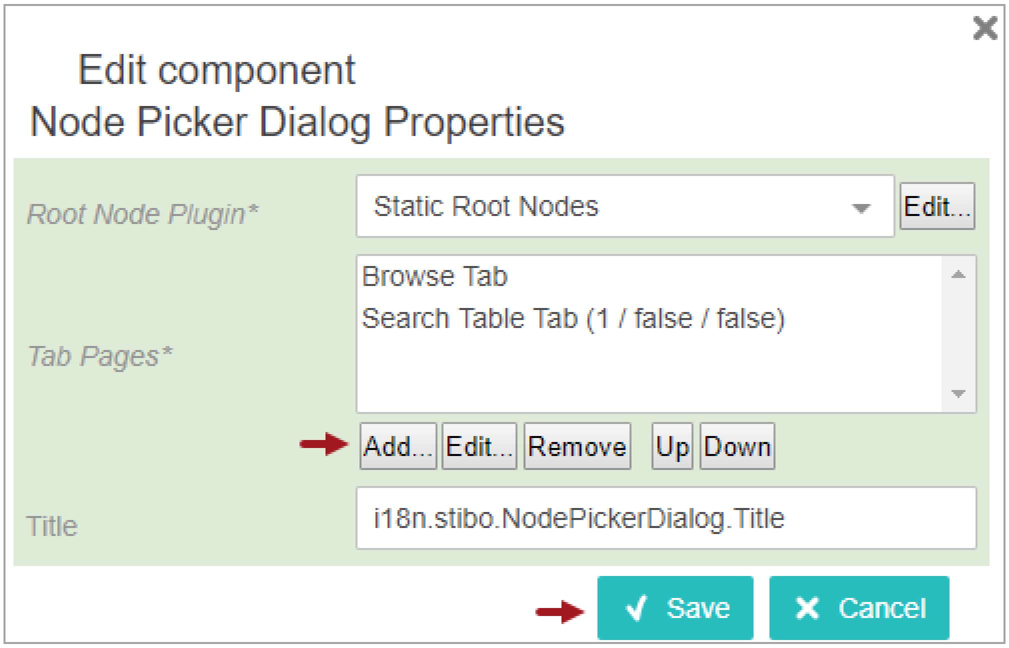
11. Click the **Add** button beneath the Root Nodes URLs parameter > select the Root Nodes that the user should be able to choose from when changing the part type (For this example AC_VehicleRoot is used) > click the **OK** button to close the dialog, and return to the 'Edit component' for 'Static Root Nodes Properties' dialog. Optionally repeat this step to add additional nodes for a user to browse from when looking for a part type.



Edit component
Static Root Nodes Properties

Root Node URLs*: step://classification?id=AC_VehicleRoot

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.



Edit component
Node Picker Dialog Properties

Root Node Plugin*: Static Root Nodes

Tab Pages*:
Browse Tab
Search Table Tab (1 / false / false)

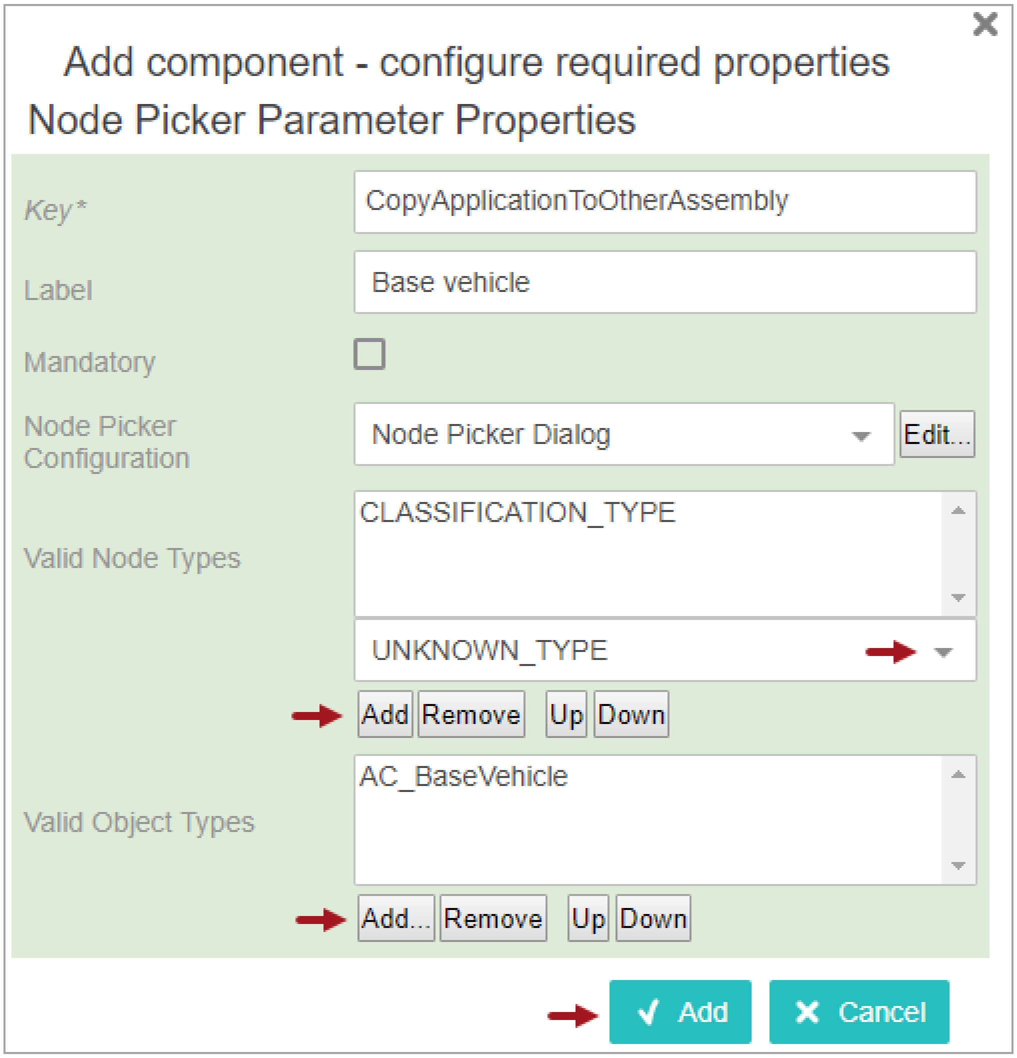
Title: i18n.stibo.NodePickerDialog.Title

Note: Leave the Title parameter blank and *after the configuration is saved* an i18n key will be populated (as shown above). For more information, see the **Localization** topic within the **Administration Portal** section of the **STEP Online Help**. Otherwise, text entered within the Title parameter will display in place of the default 'Select Node(s)' title for the dialog used to select the vehicle assembly.

14. Click the **Save** button, and the 'Add component - configure required properties' for the 'Node Picker Parameter Properties' dialog will display with the Node Picker Configuration parameter populated.

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

16. Click the **Add** button beneath the Valid Object Types parameter > select the desired valid object types > click the **OK** button to return to the 'Add component - configure required properties' for the 'Node Picker Parameter Properties' dialog.



Add component - configure required properties

Node Picker Parameter Properties

Key* CopyApplicationToOtherAssembly

Label Base vehicle

Mandatory □

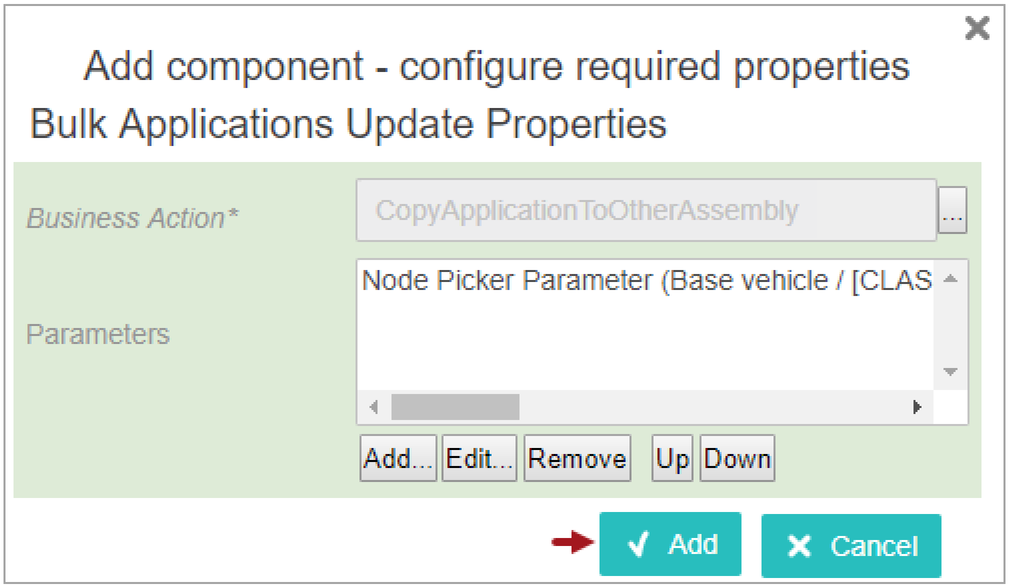
Node Picker Configuration Node Picker Dialog

Valid Node Types CLASSIFICATION_TYPE

UNKNOWN_TYPE

Valid Object Types AC_BaseVehicle

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



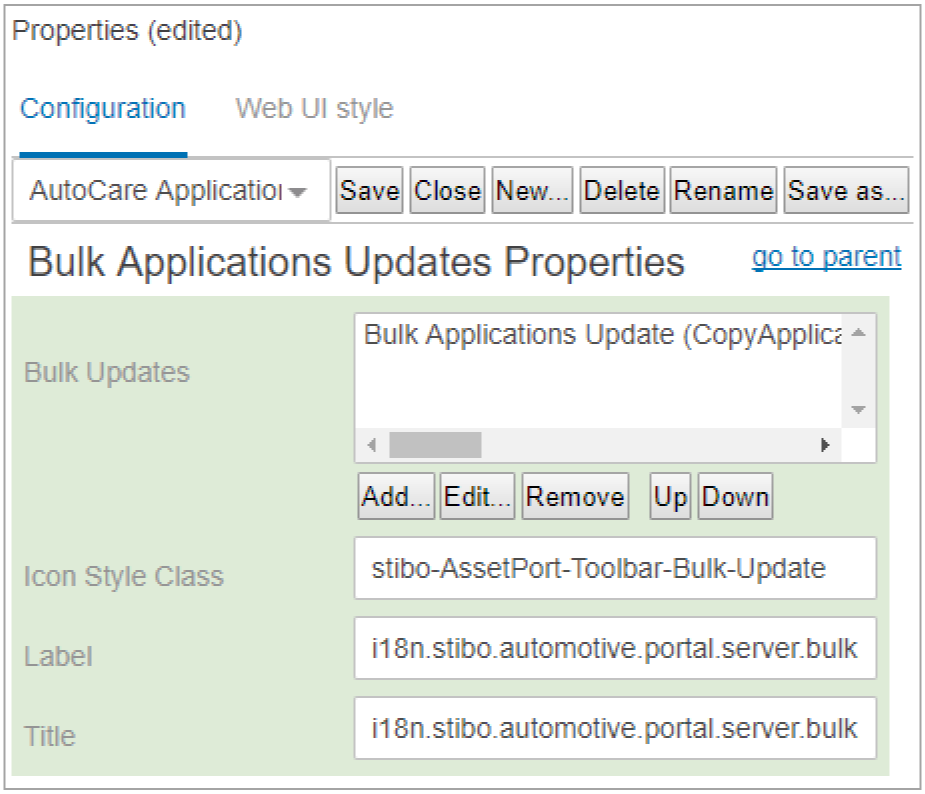
Add component - configure required properties

Bulk Applications Update Properties

Business Action*: CopyApplicationToOtherAssembly

Parameters: Node Picker Parameter (Base vehicle / [CLAS

18. Click the **Add** button, and the 'Bulk Applications Updates Properties' dialog will display with the newly added Bulk Update listed.



Properties (edited)

Bulk Applications Updates Properties

Bulk Updates: Bulk Applications Update (CopyApplica

Icon Style Class: stibo-AssetPort-Toolbar-Bulk-Update

Label: i18n.stibo.automotive.portal.server.bulk

Title: i18n.stibo.automotive.portal.server.bulk

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

To use the newly configured business action, see the **Using the Configured 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

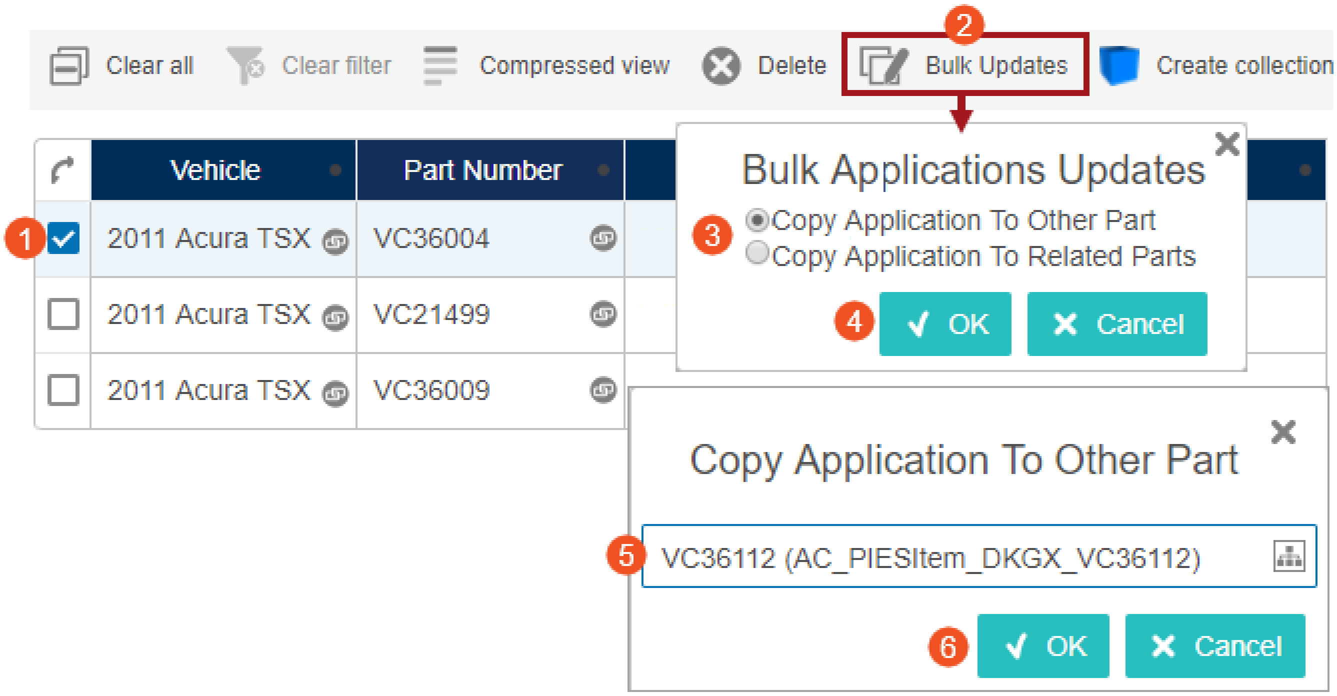
This automotive business action allows users to copy one or more existing applications to another part by clicking a Bulk Updates action button within an Application Editor. However, it cannot copy the part of a missing application, and thus any missing applications selected when the bulk update is run will be ignored. For more information, see the **Missing Coverage** topic within this guide. Setup is required within both STEP Workbench and Web UI for the action to be available to users.

This section addresses steps necessary when:

- Using the Configured Copy Application to Other Part Business Action in Web UI
- Configuring the Copy Application to Other Part Business Action in Workbench
- Configuring the Copy Application to Other Part Business Action in Web UI

Using the Configured Copy Application to Other Part Business Action in Web UI

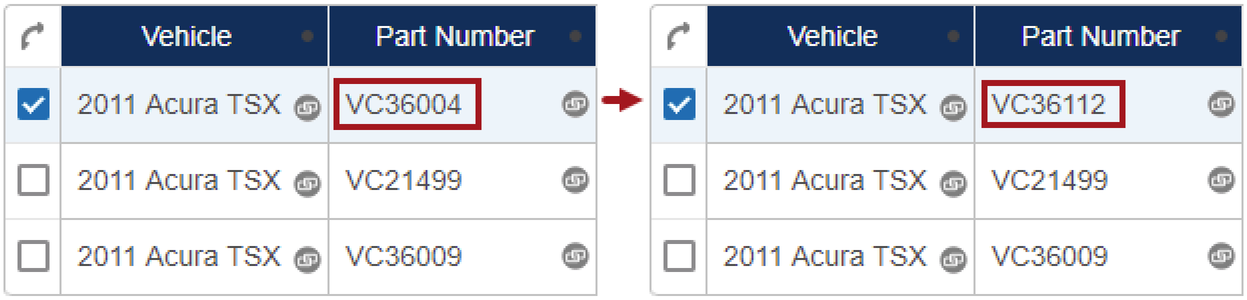
Once configured, a Bulk Updates button within an Application Editor can be used to copy one or more existing applications to another part. Below are the steps for using the business action in Web UI.



Note: If only one bulk update is configured, then the Bulk Applications Updates dialog will not display.

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

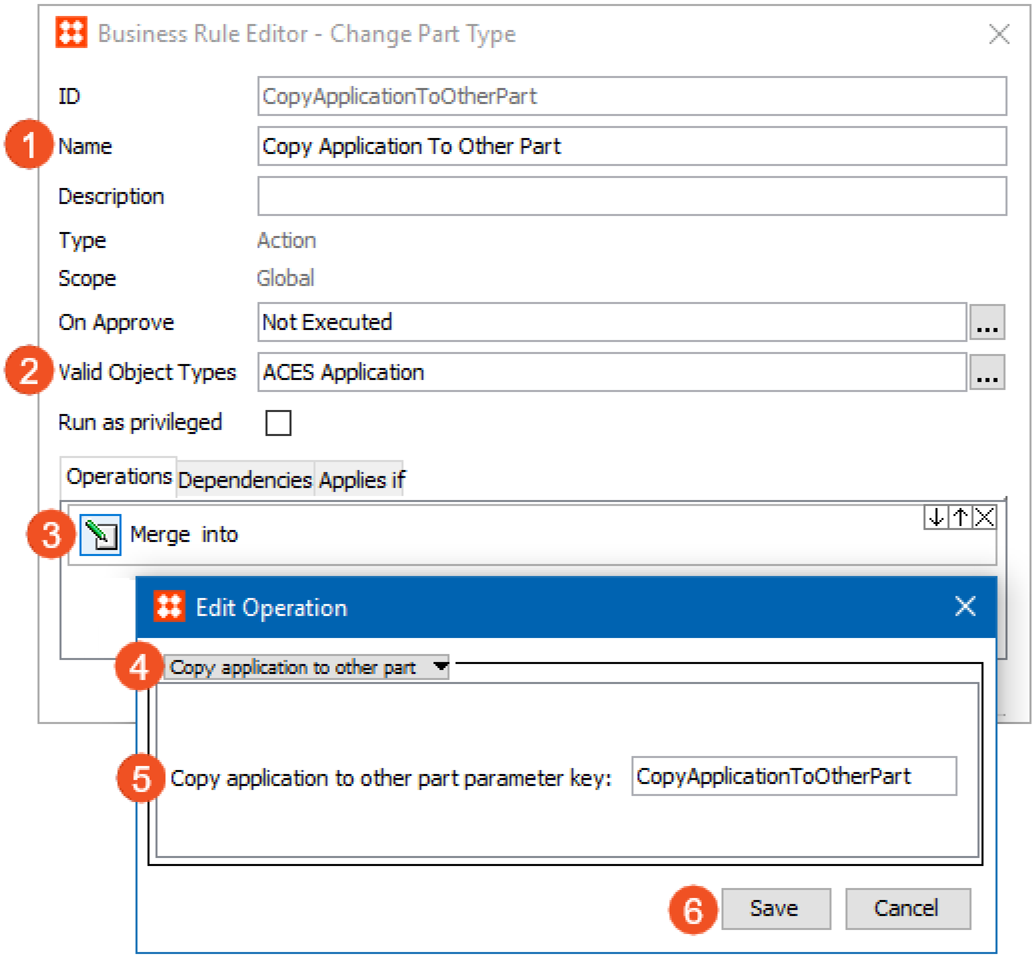


	Vehicle	Part Number
☑	2011 Acura TSX	VC36004
□	2011 Acura TSX	VC21499
□	2011 Acura TSX	VC36009

	Vehicle	Part Number
☑	2011 Acura TSX	VC36112
□	2011 Acura TSX	VC21499
□	2011 Acura TSX	VC36009

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.



Business Rule Editor - Change Part Type

ID: CopyApplicationToOtherPart

1 Name: Copy Application To Other Part

Description:

Type: Action

Scope: Global

On Approve: Not Executed

2 Valid Object Types: ACES Application

Run as privileged: □

3 Merge into

Edit Operation

4 Copy application to other part

5 Copy application to other part parameter key: CopyApplicationToOtherPart

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 Editor 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 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 application to other part** operation, and the parameter 'Copy application to other part 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.

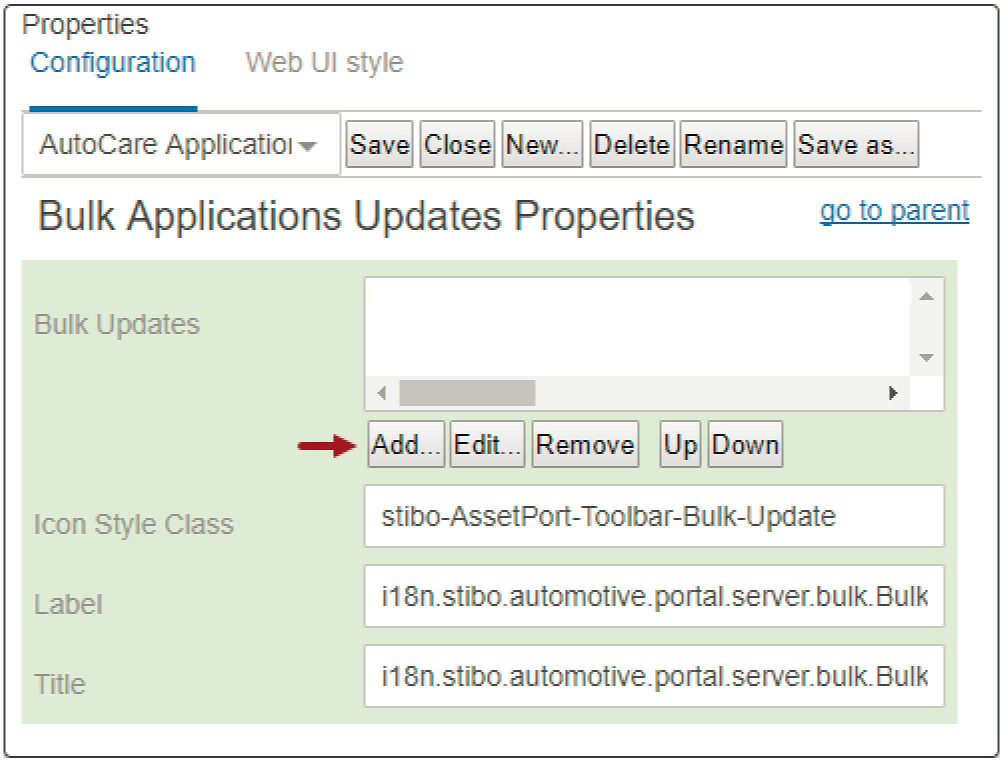
6. Click the **Save** button and continue to the next topic, **Configuring the Copy Application to Other Part Business Action in Web UI**.

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 Editor 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).



Bulk Applications Updates Properties

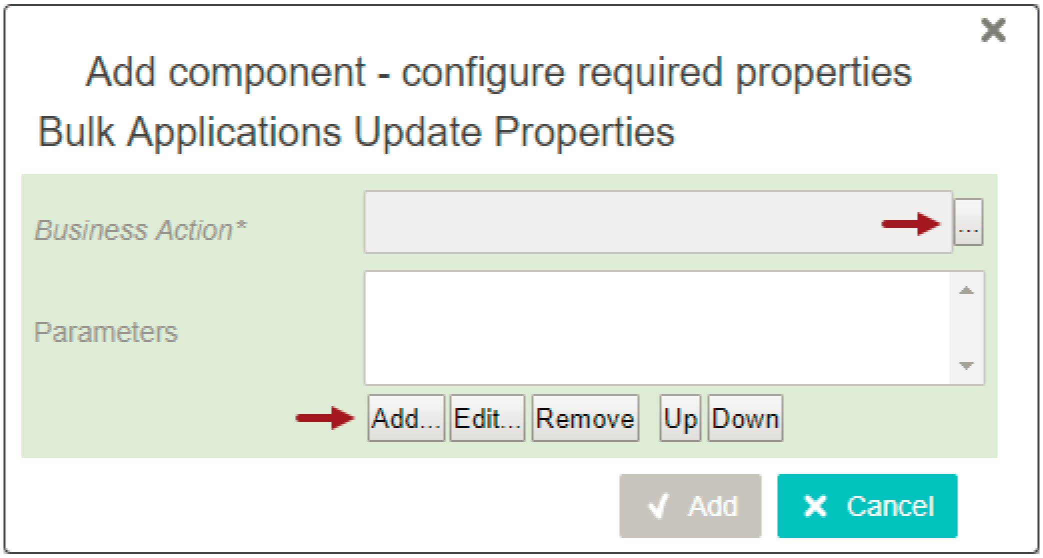
Bulk Updates

Icon Style Class: stibo-AssetPort-Toolbar-Bulk-Update

Label: i18n.stibo.automotive.portal.server.bulk.Bulk

Title: i18n.stibo.automotive.portal.server.bulk.Bulk

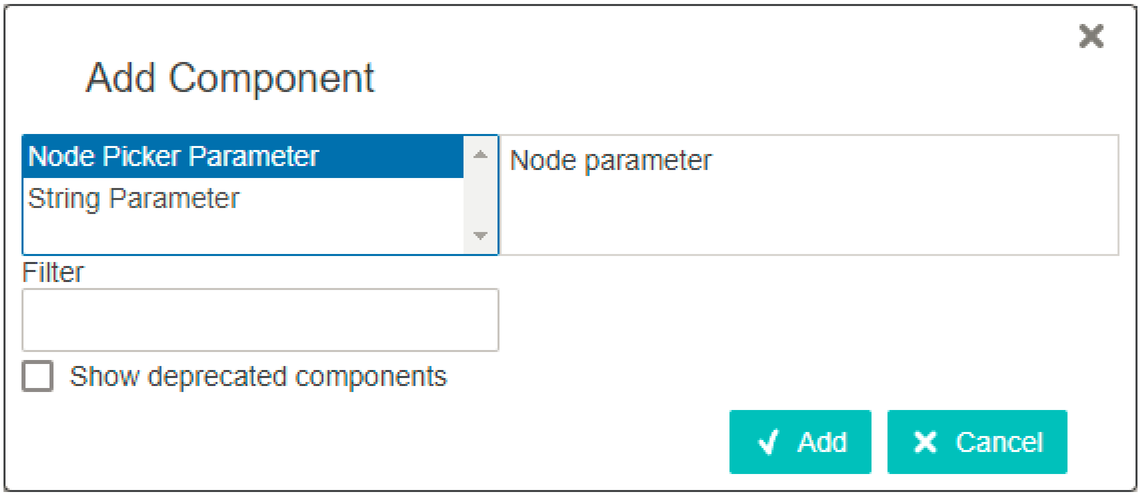
3. Click the **Add** button beneath the Bulk Updates parameter, and the 'Add component - configure required properties' for the 'Bulk Applications Update Properties' dialog will display (as shown below).



Add component - configure required properties

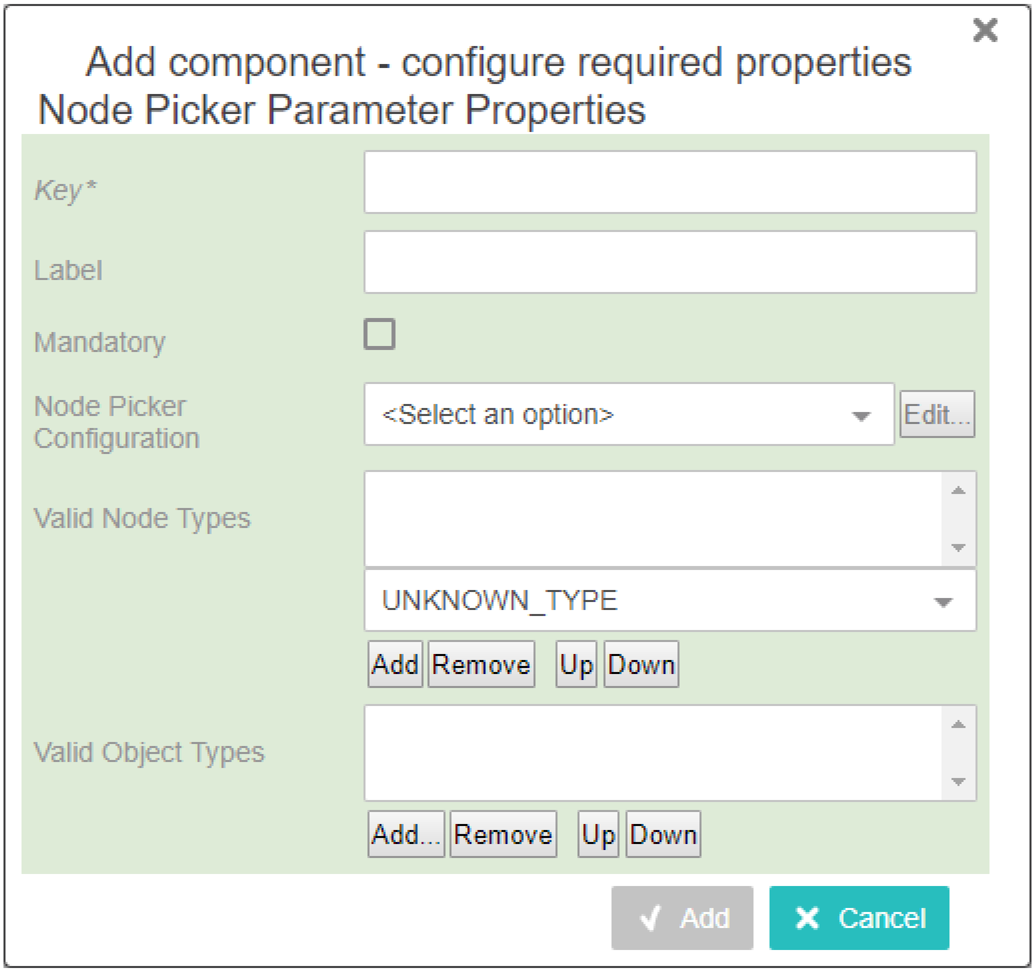
Bulk Applications Update Properties

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).



Add Component

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



Add component - configure required properties

Node Picker Parameter Properties

Key *

Label

Mandatory □

Node Picker Configuration <Select an option>

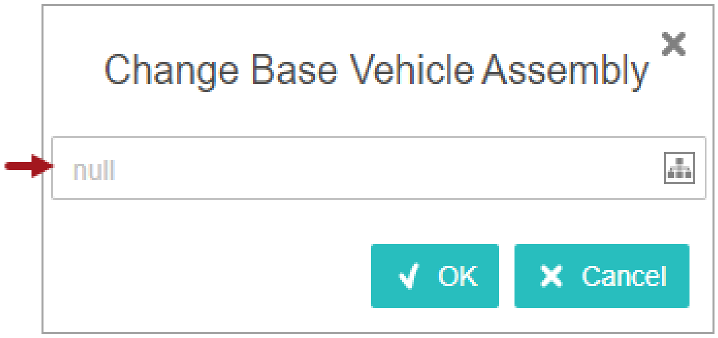
Valid Node Types UNKNOWN_TYPE

Valid Object Types

7. Within the Key parameter, enter the exact key created within the 'Part type 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.

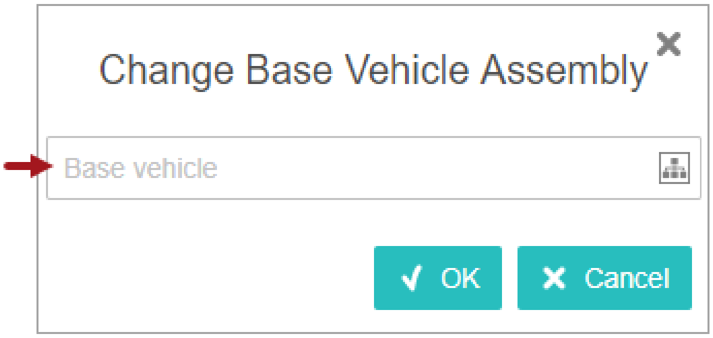
Important: If this key is not entered exactly in both places, then the business action will not properly function within the Web UI.

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.



Change Base Vehicle Assembly

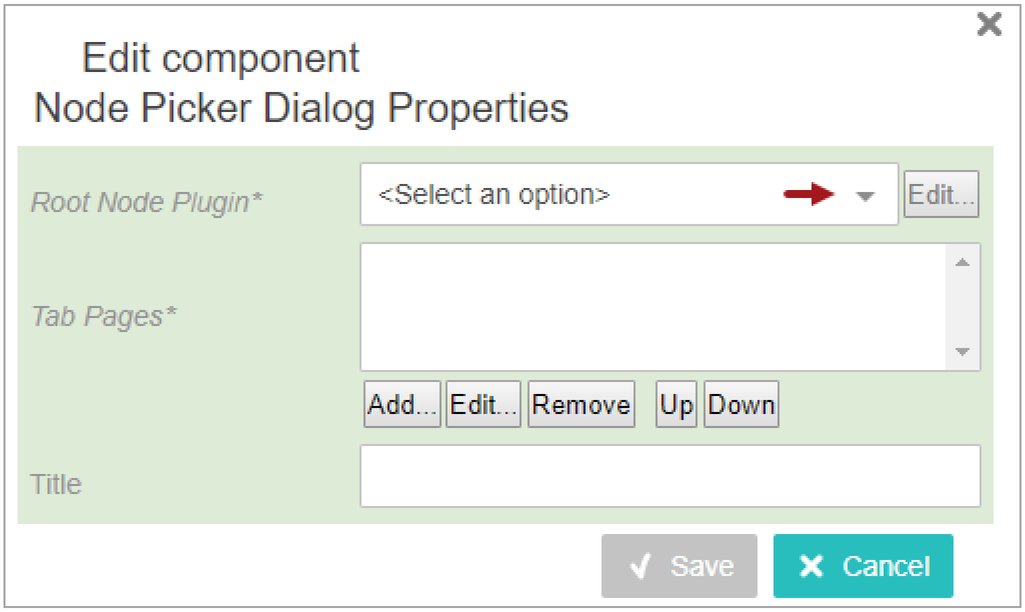
null



Change Base Vehicle Assembly

Base vehicle

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.



Edit component
Node Picker Dialog Properties

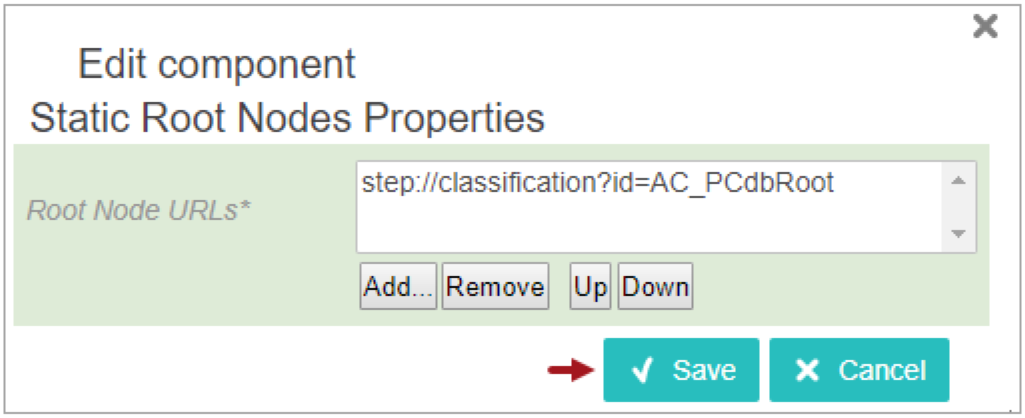
*Root Node Plugin** <Select an option>

*Tab Pages**

Title

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.

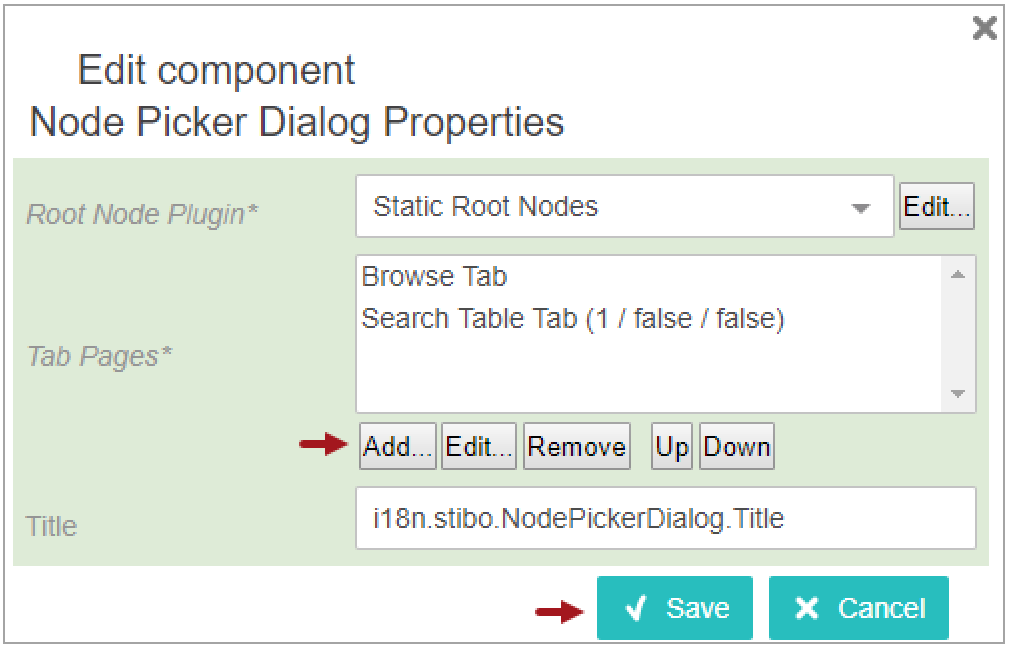
11. Click the **Add** button beneath the Root Nodes URLs parameter > select the Root Nodes that the user should be able to choose from when changing the part (For this example AC_PCdbRoot is used) > click the **OK** button to close the dialog, and return to the 'Edit component' for 'Static Root Nodes Properties' dialog. Optionally repeat this step to add additional nodes for a user to browse from when looking for a part type.


Edit component

Static Root Nodes Properties

Root Node URLs*: step://classification?id=AC_PCdbRoot

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.


Edit component

Node Picker Dialog Properties

Root Node Plugin*: Static Root Nodes

Tab Pages*: Browse Tab
Search Table Tab (1 / false / false)

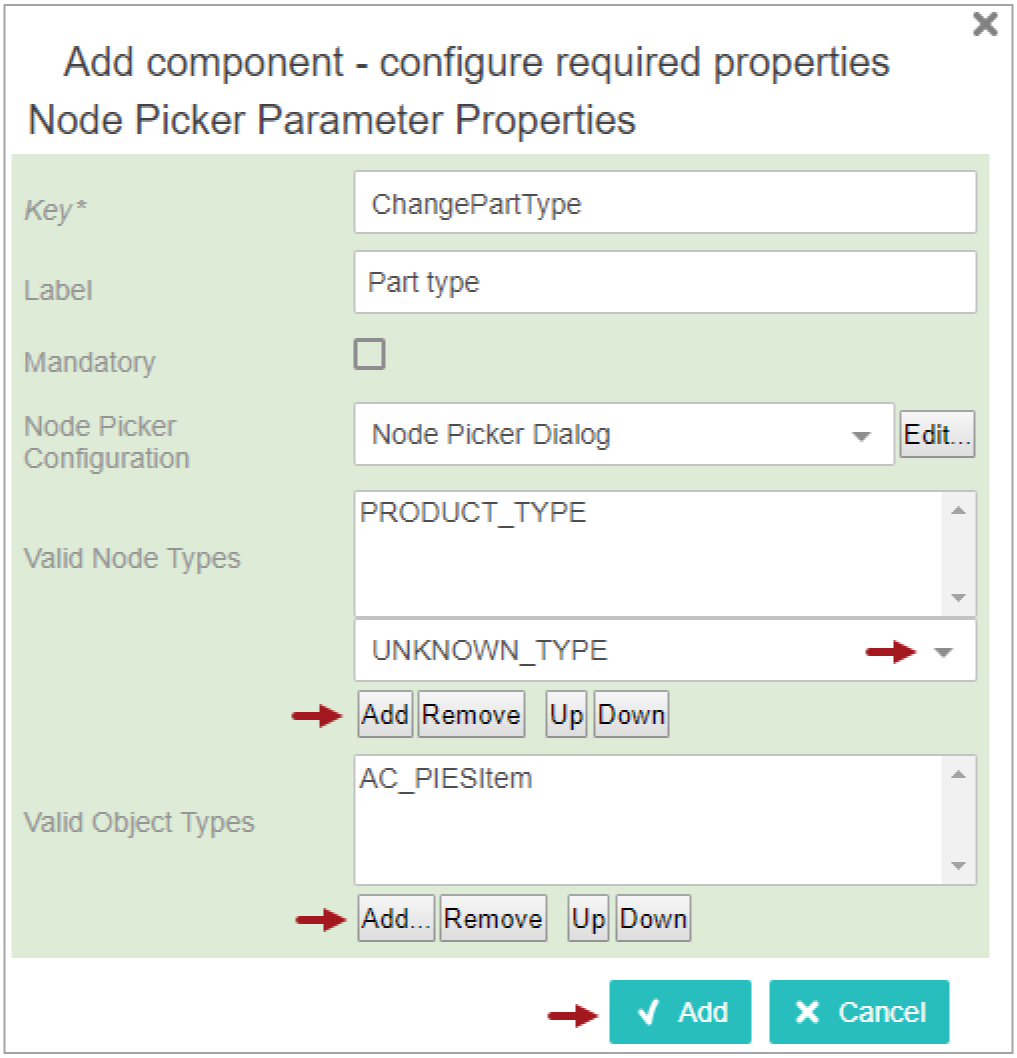
Title: i18n.stibo.NodePickerDialog.Title

Note: Leave the Title parameter blank and *after the configuration is saved* an i18n key will be populated (as shown above). For more information, see the **Localization** topic within the **Administration Portal** section of the **STEP Online Help**. Otherwise, text entered within the Title parameter will display in place of the default 'Select Node(s)' title for the dialog used to select the vehicle assembly.

14. Click the **Save** button, and the 'Add component - configure required properties' for the 'Node Picker Parameter Properties' dialog will display with the Node Picker Configuration parameter populated.

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

16. Click the **Add** button beneath the Valid Object Types parameter > select the desired valid object types (For this example AC_PIESItem is used)> click the **OK** button to return to the 'Add component - configure required properties' for the 'Node Picker Parameter Properties' dialog.

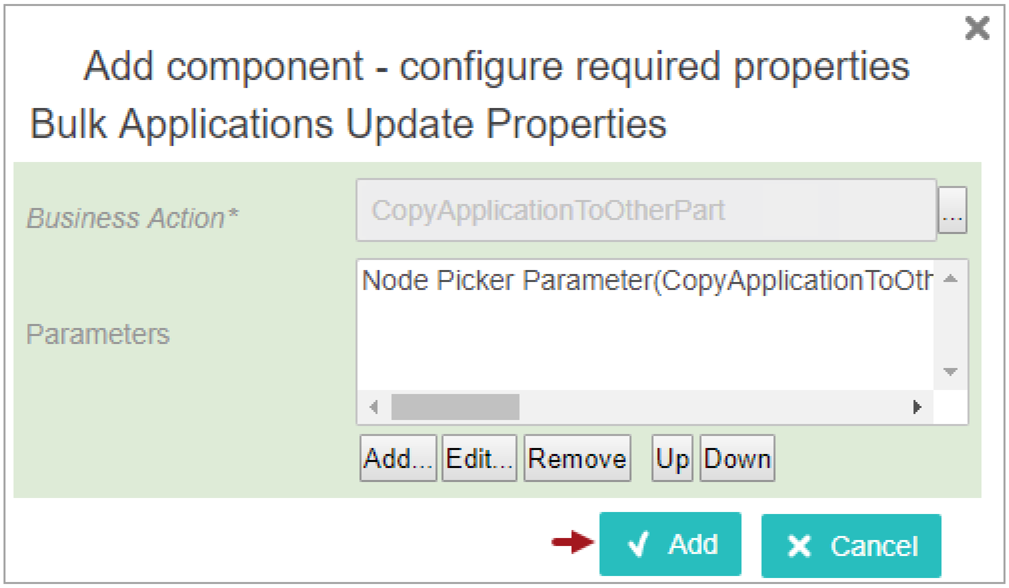


Add component - configure required properties

Node Picker Parameter Properties

*Key**	ChangePartType
Label	Part type
Mandatory	□
Node Picker Configuration	Node Picker Dialog
Valid Node Types	PRODUCT_TYPE UNKNOWN_TYPE
Valid Object Types	AC_PIESItem

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



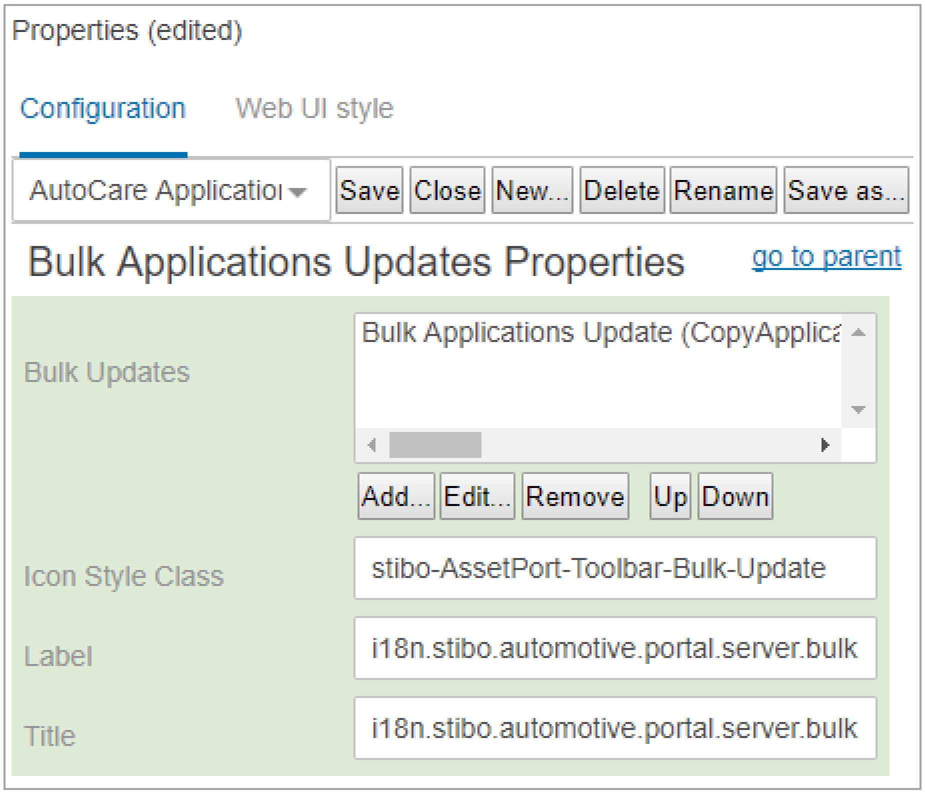
Add component - configure required properties

Bulk Applications Update Properties

Business Action*: CopyApplicationToOtherPart

Parameters: Node Picker Parameter(CopyApplicationToOth

18. Click the **Add** button, and the 'Bulk Applications Updates Properties' dialog will display with the newly added Bulk Update listed.



Properties (edited)

Bulk Applications Updates Properties

Bulk Updates: Bulk Applications Update (CopyApplica

Icon Style Class: stibo-AssetPort-Toolbar-Bulk-Update

Label: i18n.stibo.automotive.portal.server.bulk

Title: i18n.stibo.automotive.portal.server.bulk

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

To use the newly configured business action, see the **Using the Configured 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 Editor. However, it cannot copy the part of a missing application, and thus any missing applications selected when the bulk update is run will be ignored. For more information, see the **Missing Coverage** topic within this guide. Setup is required within both STEP Workbench and Web UI for the action to be available to users.

This section addresses steps necessary when:

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

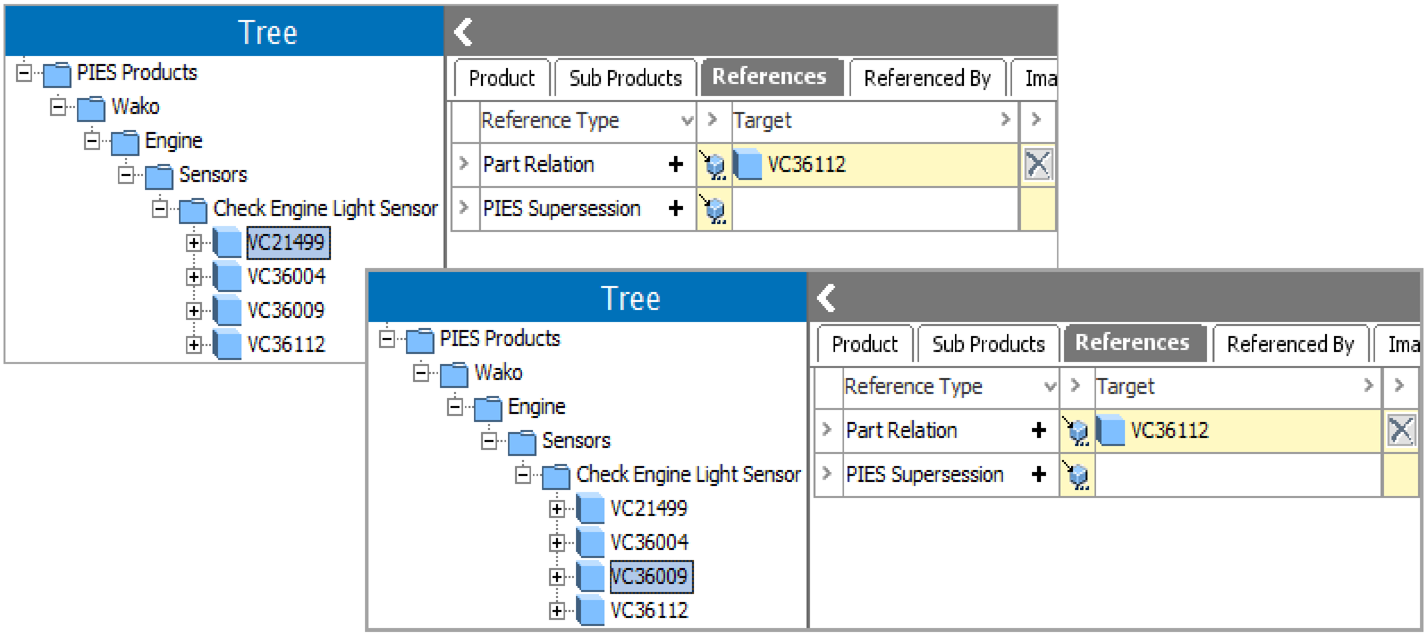
Using the Configured Copy Applications to Related Parts Business Action in Web UI

Once configured, a Bulk Updates button within an Application Editor 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.'



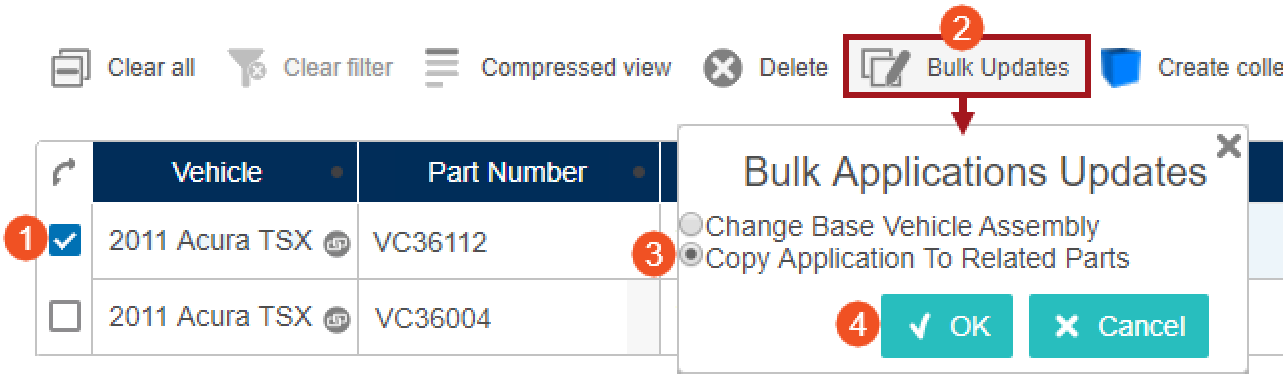
Reference Type	Target
Part Relation	VC36112
PIES Supersession	

Reference Type	Target
Part Relation	VC36112
PIES Supersession	

With the above configuration in place, the Application Editor 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).

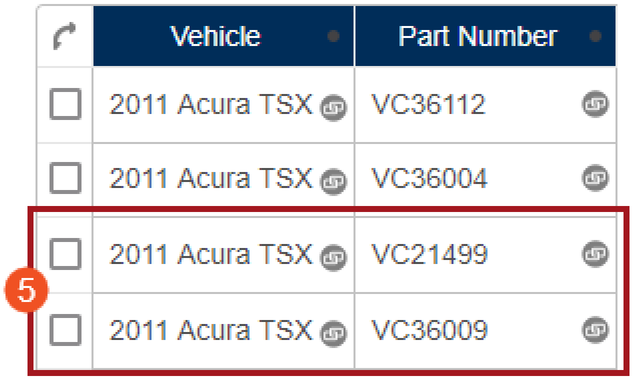
Steps for Using the Business Action in Web UI

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 Editor, 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 Editor results table. For this example, notice that two new applications for the related parts have been created (as shown below).



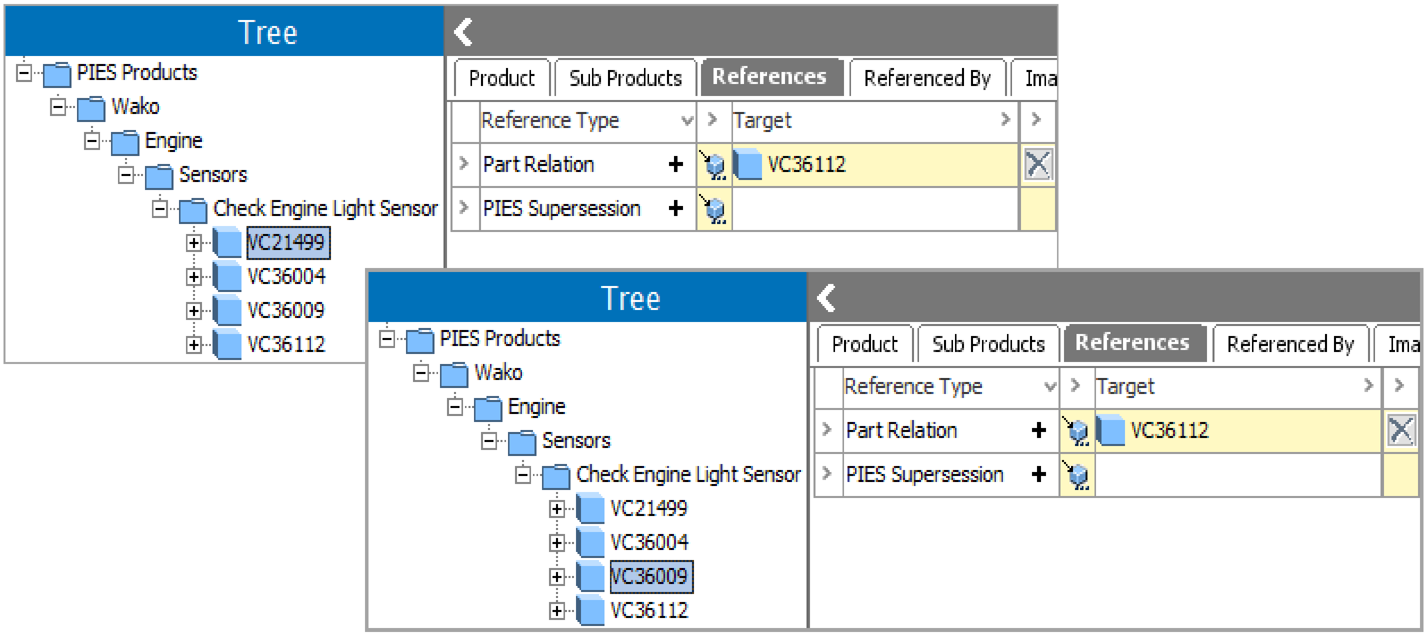
	Vehicle	Part Number
□	2011 Acura TSX	VC36112
□	2011 Acura TSX	VC36004
□	2011 Acura TSX	VC21499
□	2011 Acura TSX	VC36009

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.

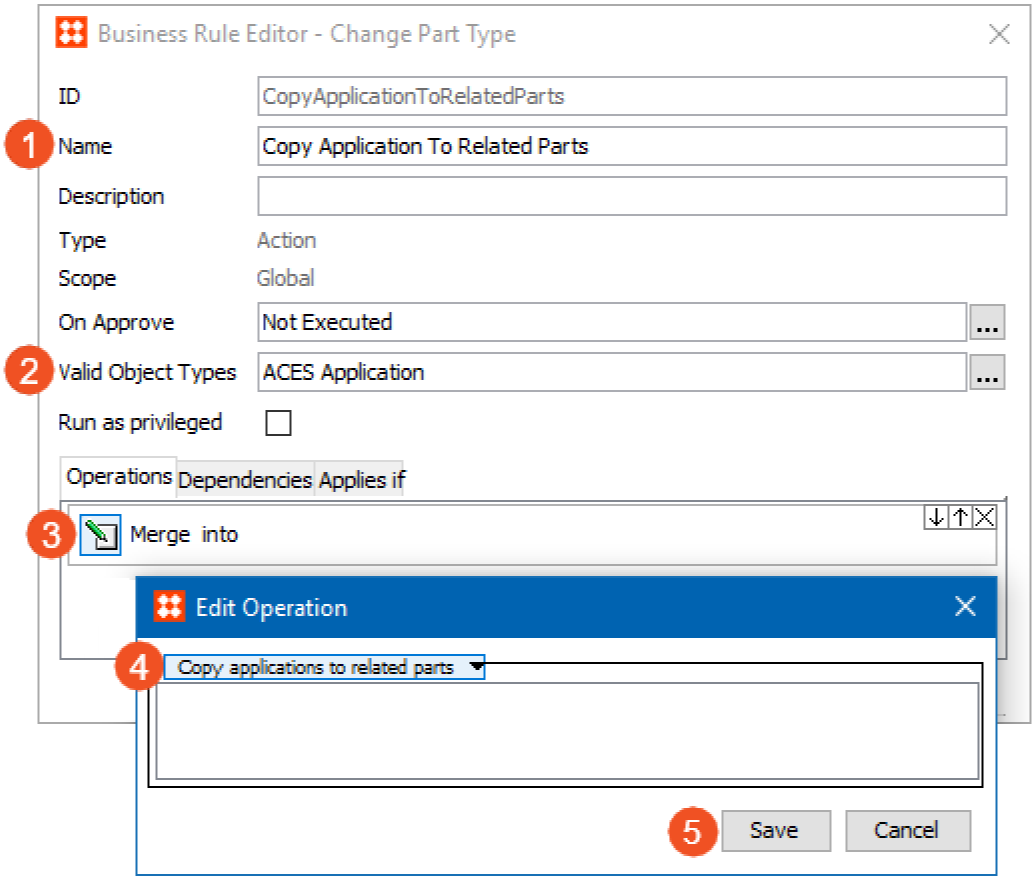


Reference Type	Target
Part Relation	VC36112
PIES Supersession	

Reference Type	Target
Part Relation	VC36112
PIES Supersession	

Configuring the Business Action

Before the business action can be used within Web UI, the business action must be created. Below are the steps required to create the business action.



Business Rule Editor - Change Part Type

ID: CopyApplicationToRelatedParts

Name: Copy Application To Related Parts

Description:

Type: Action

Scope: Global

On Approve: Not Executed

Valid Object Types: ACES Application

Run as privileged: □

Operations | Dependencies | Applies if

Merge into

Edit Operation

Copy applications to related parts

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 Editor 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**.

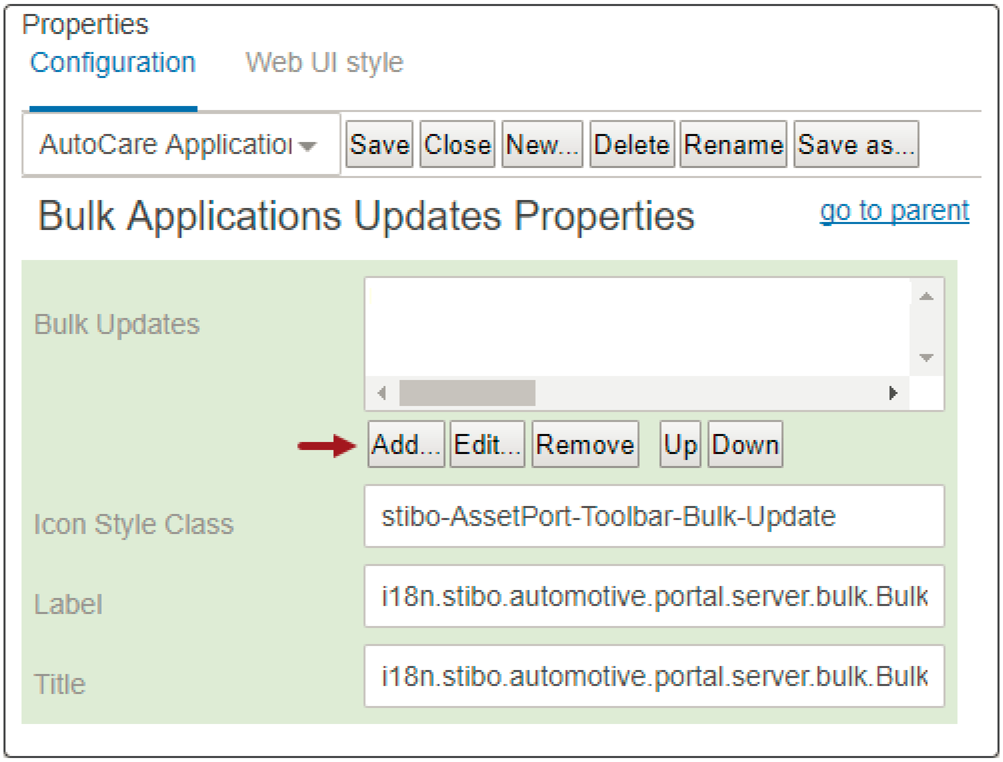
Configuring the Copy Applications to Related Parts Business Action in Web UI

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 Editor 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).



Bulk Applications Updates Properties

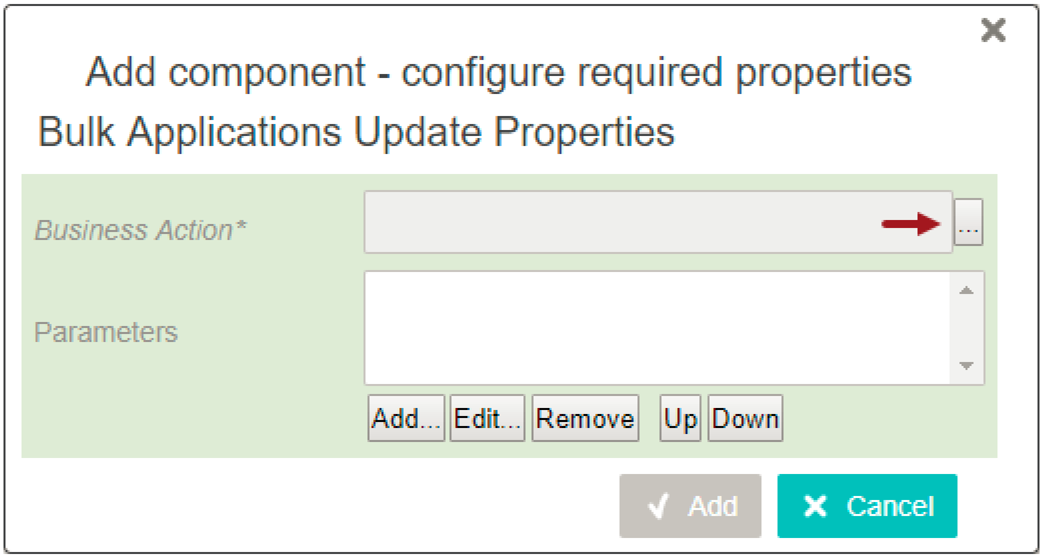
Bulk Updates

Icon Style Class: stibo-AssetPort-Toolbar-Bulk-Update

Label: i18n.stibo.automotive.portal.server.bulk.Bulk

Title: i18n.stibo.automotive.portal.server.bulk.Bulk

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).



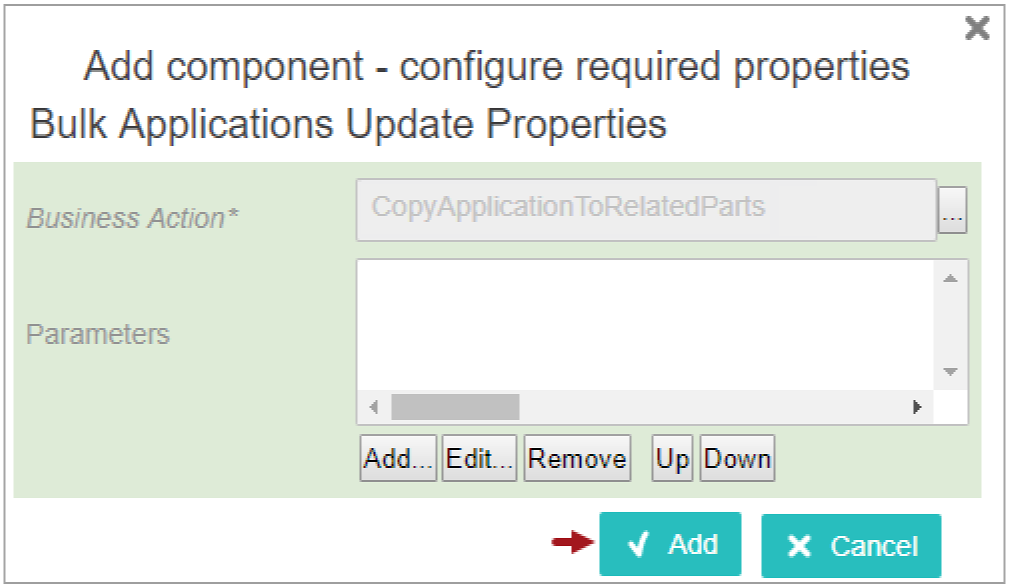
Add component - configure required properties

Bulk Applications Update Properties

Business Action*:

Parameters:

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.



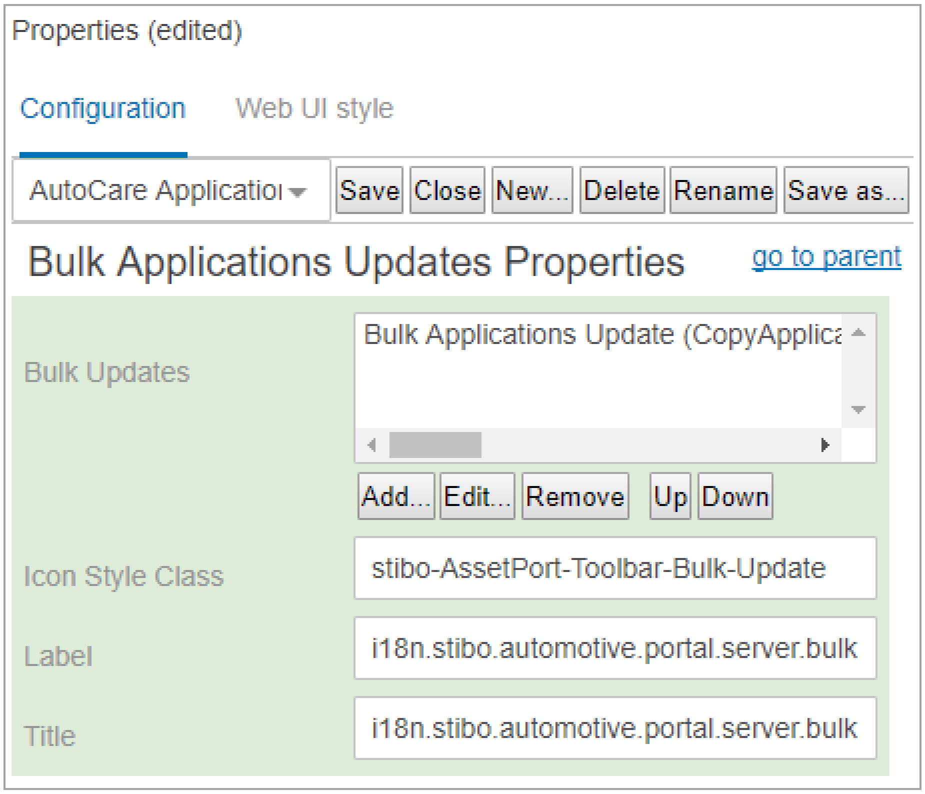
Add component - configure required properties

Bulk Applications Update Properties

Business Action*: CopyApplicationToRelatedParts

Parameters:

5. Click the **Add** button, and the 'Bulk Applications Updates Properties' dialog will display with the newly added Bulk Update listed.



Properties (edited)

Bulk Applications Updates Properties

Bulk Updates: Bulk Applications Update (CopyApplica

Icon Style Class: stibo-AssetPort-Toolbar-Bulk-Update

Label: i18n.stibo.automotive.portal.server.bulk

Title: i18n.stibo.automotive.portal.server.bulk

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

To use the newly configured business action, see the **Using the Configured Copy Applications to Related Parts Business Action in Web UI** section of the **Business Action: Copy Applications to Related Parts** topic.

Business Action: Move ACES Applications for PIES Part

This rule is a business action found under the Automotive menu that will move all ACES applications from one PIES Item to another by following a reference between the parts. The rule has a single parameter (Reference Type), which requires selection of a reference. For the rule to function properly, the selected reference must be a product reference with the PIES Item object type (ID=AC_PIESItem) as both the source and the target, and the rule must be set with the PIES Item as the valid object type.

The action evaluates all ACES applications that are child to the selected part on which the rule is being executed and moves the child applications from the current object to the target(s) of the reference identified in the rule. If the PIES Item has no reference of this type, no action is taken and the existing applications will remain in place. If the PIES Item has one or multiple targets of the reference, the applications will be duplicated to all targets and removed from the source object.

When a move occurs, the application is moved with all data on the original record retained, except the part number. The part that the application is being moved to is applied as the part number in both the application name and the hash function of the application ID, as well as within the ACES Replacement Context metadata value. The brand element of the replacement context is also updated if the part that the application is being moved to is from another brand (indicated by the value of the attribute with ID=AC_PIES_ITEMBrandAAIAID on the PIES Item).

Important: ACES supports only a single asset being linked to any application. If multiple assets are present on an application, all will be linked to the moved record, but only one will be accounted for in the application ID. This could lead to duplicate applications in the system so it is important to ensure that the standard is adhered to and only one asset is present for any application.

Note that this rule cannot be used to move applications outside of the AutoCare model (using PIES Item and ACES Application object types) as the functionality for construction of the application ID and other related data relies on the standard model.

Business Action: Set Condition Links on Part Types

This rule is a business action found under the Automotive menu that serves to link application conditions / options to part types to assist in configuring display options in the Web UI Application Record Editor (application search screen). Additional information on the display settings can be found in the Application Editor 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.

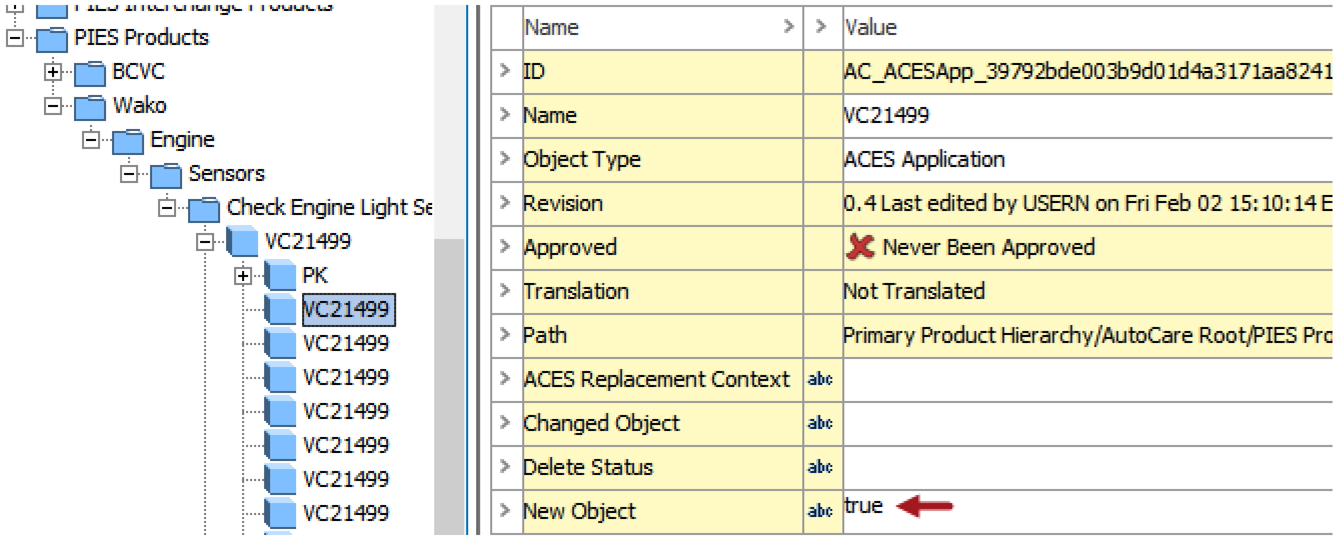
If attribute or reference links already exist on any part type for the populated conditions, the existing links are not altered by running the rule - only new links are added.

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.

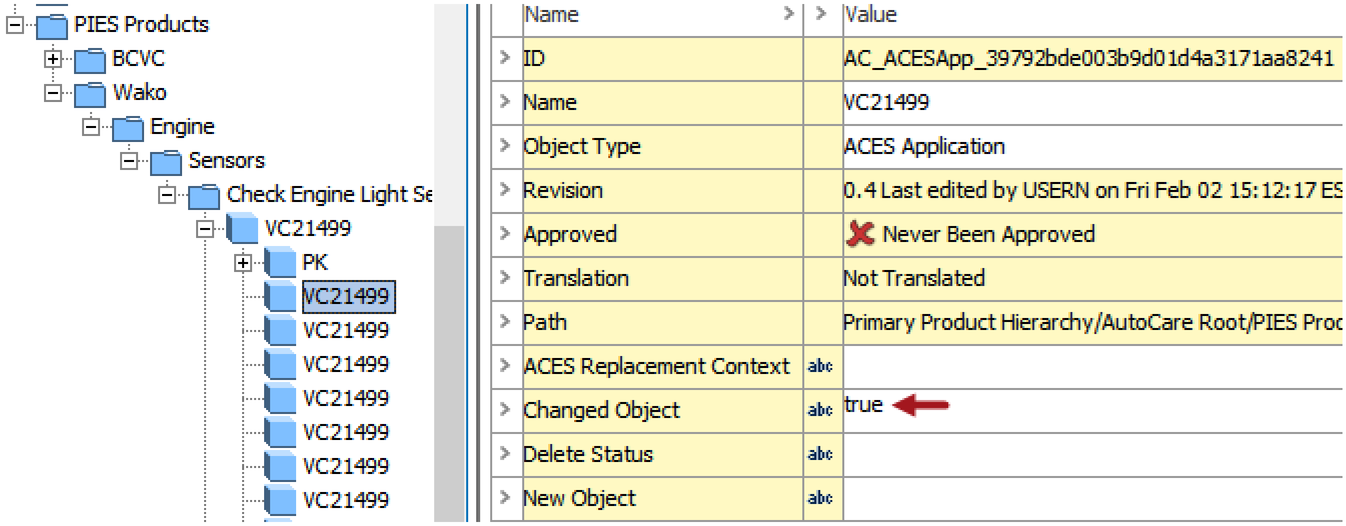
In the example below, an ACES import created a new application for part VC21499. Because the application is new, the configured attribute New Object displays the value of 'true.'



PIES Interchange Products
PIES Products
BCVC
Wako
Engine
Sensors
Check Engine Light Se
VC21499
PK
VC21499
VC21499
VC21499
VC21499
VC21499
VC21499
VC21499

Name		Value
ID		AC_ACESApp_39792bde003b9d01d4a3171aa8241
Name		VC21499
Object Type		ACES Application
Revision		0.4 Last edited by USERN on Fri Feb 02 15:10:14 E
Approved		✗ Never Been Approved
Translation		Not Translated
Path		Primary Product Hierarchy/AutoCare Root/PIES Prc
ACES Replacement Context	abc	
Changed Object	abc	
Delete Status	abc	
New Object	abc	true ←

If the application had previously existed and had information changed during the import, then only the Changed Object attribute value would display 'true.'



Name		Value
ID		AC_ACESApp_39792bde003b9d01d4a3171aa8241
Name		VC21499
Object Type		ACES Application
Revision		0.4 Last edited by USERN on Fri Feb 02 15:12:17 ES
Approved		✗ Never Been Approved
Translation		Not Translated
Path		Primary Product Hierarchy/AutoCare Root/PIES Proc
ACES Replacement Context	abc	
Changed Object	abc	true
Delete Status	abc	
New Object	abc	

Important: This business action can ONLY be used within the Import state of a workflow. Additionally, an error will occur when delete flags are used for TecDoc reference data on an 8.2 system. It is recommended that change flags for TecDoc reference data only be implemented on 8.3 (or newer) releases.

Configuring Change Flags for New and/or Updated Objects

To properly configure this business action to track new and/or updated objects, the following is needed:

- Create / Identify Valid Attributes
- Configure a Business Action to Use the Set Import Status Attributes Operation
- Update Workflow Settings

Create / Identify Valid Attributes

1. Create the necessary attributes (or verify that they exist) to be used to store the new and/or changed data (i.e. Delete Status, New Object, Changed Object).
2. Make sure the attributes are valid for the object(s) being managed by the import.

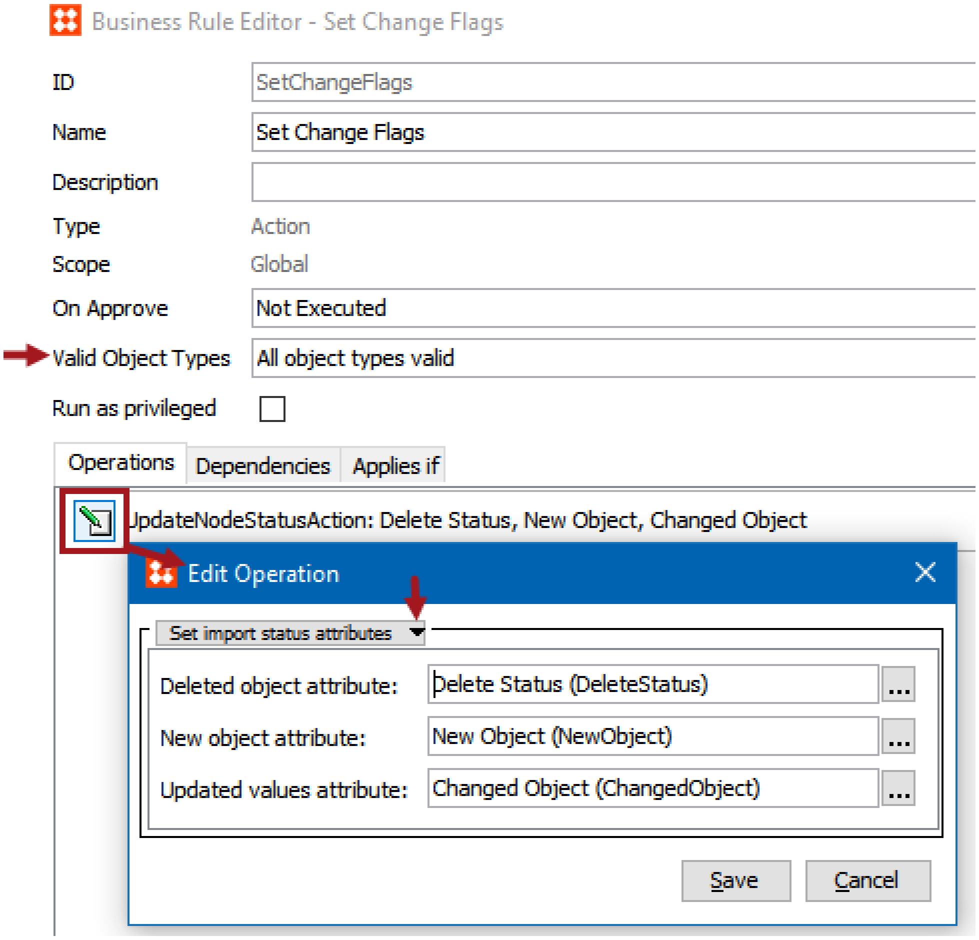
   For example, an attribute used in the AutoCare ACES Application Importer must be valid on ACES Application objects. Alternatively, an attribute used in the AutoCare PCdb Importer must be valid on all nodes in the PCdb classification hierarchy.

Configure a Business Action to Use the Set Import Status Attributes Operation

1. Create a business action and give it a name that users can easily identify. In the example below, Set Change Flags is used.
2. Click the ellipsis button **(...)** to the right of the Valid Object Types parameter to find and select the necessary object type(s). In the Set Change Flags example below, ACES Application 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 an attribute to be used to identify when an object should be marked for deletion. The attribute that is used in this parameter should be the same attribute that is configured in the delta calculation. For the example below, 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 below, 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 below, the Changed Object (created / identified in the previous section) is selected.

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



Business Rule Editor - Set Change Flags

ID: SetChangeFlags

Name: Set Change Flags

Description:

Type: Action

Scope: Global

On Approve: Not Executed

Valid Object Types: All object types valid

Run as privileged: □

UpdateNodeStatusAction: Delete Status, New Object, Changed Object

Edit Operation

Set import status attributes

Deleted object attribute: Delete Status (DeleteStatus)

New object attribute: New Object (NewObject)

Updated values attribute: Changed Object (ChangedObject)

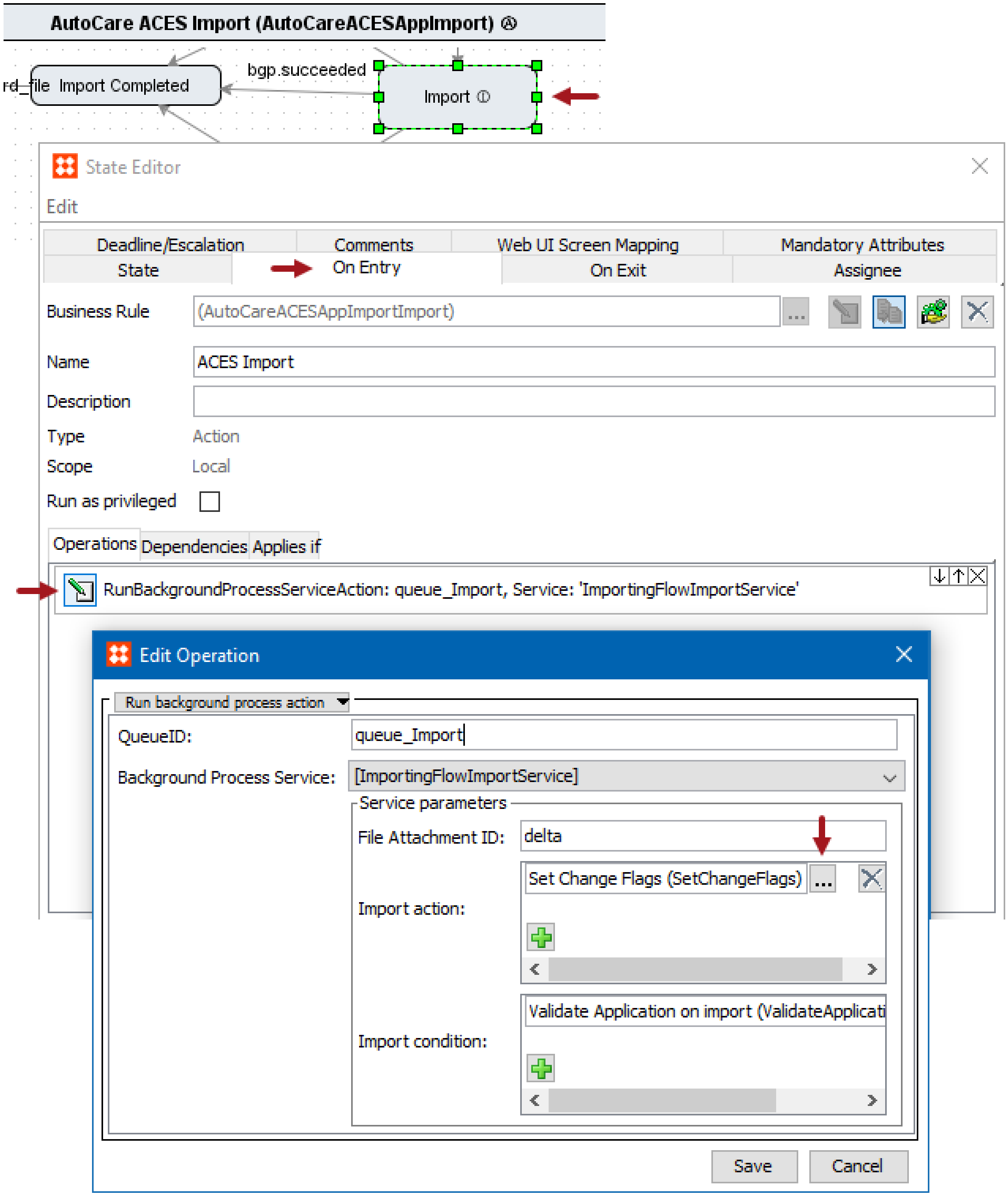
For more information on creating and/or editing business rules, see the **Creating a Business Rule or Library** topic, and the **Editing a Business Rule** topic within the **Business Rules** guide.

Update Workflow Settings

Once the business rule is configured, the necessary workflows need to be updated to use the business rule. Below are the steps to update a workflow to use the business rule created in the previous section.

Important: Because the business rule created in the previous section is configured to work with ACES Application valid object types, the ACES Import Workflow is used for the example below. If the business rule is configured to also work with TecDoc and/or NAPA Applications, then those import workflows would need to also be edited.

1. Go to workbench > System Setup > Workflows > select a workflow.
2. Right-click, and select **Edit STEP Workflow**.
3. On the STEP Workflow Designer window, double click on the **Import** state to open the State Editor.
4. Select the **On Entry** tab, and within the Operations tab, click the Edit icon to display the Edit Operations Dialog (as shown below).



Business Rule (AutoCareACESAppImportImport)

Name ACES Import

Description

Type Action

Scope Local

Run as privileged □

RunBackgroundProcessServiceAction: queue_Import, Service: 'ImportingFlowImportService'

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

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: 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. 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.

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.

Application Editor

The Application Editor is created by easy setup (System Setup > Component Models > Automotive - [Standard] Model > 4. Configure [Standard] Web UI). Once the appropriate easy setup action is run, the editor screen (ID= [Standard] Application Editor Search Screen) exists in the Web UI but needs to be made accessible for end users, as described in the **Automotive Quick Start Guide**. Access to the screen will likely be provided via a Quick Links widget, but is at the discretion of the administrator, thus it is not described within this guide.

Within this section, the following topics provide additional information on some of the complexities of configuring and working with the application editor:

- Using the Application Editor
- Automotive Validation Paths
- Controlling Display of Conditions in the Editor
- Making Application Editing Screens Read-Only
- Missing Coverage

Using the Application Editor

This section addresses the following functionality to be aware of when using the Application Editor:

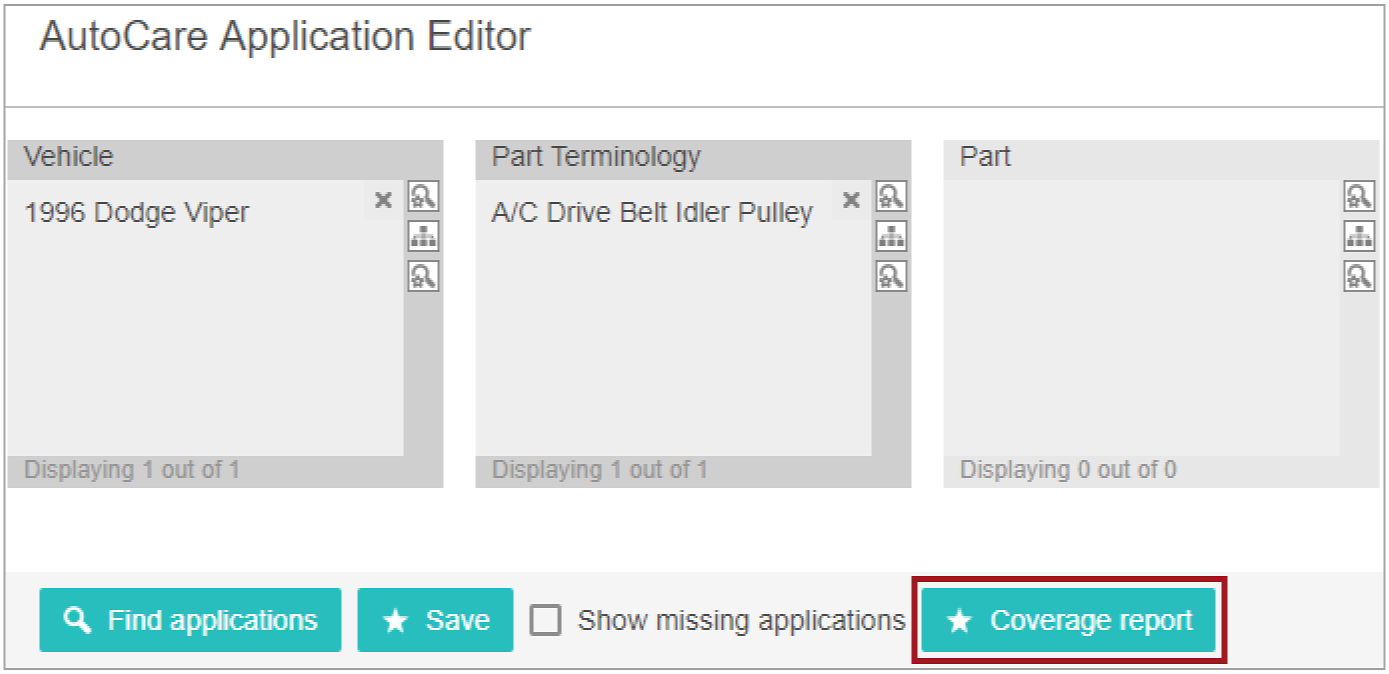
- Deleting Applications
- Running a Coverage Report
- Using Mult-Select
- Using the Value Editor
- Variations Across Standards

Deleting Applications

After preforming a vehicle search, a user can selected the application records to be deleted, and then click the delete button. This will delete the selected applications. However, if a user has searched for and selected missing applications, then clicked the delete button, no change to the missing applications will occur since an application did not exist for deletion.

Running a Coverage Report

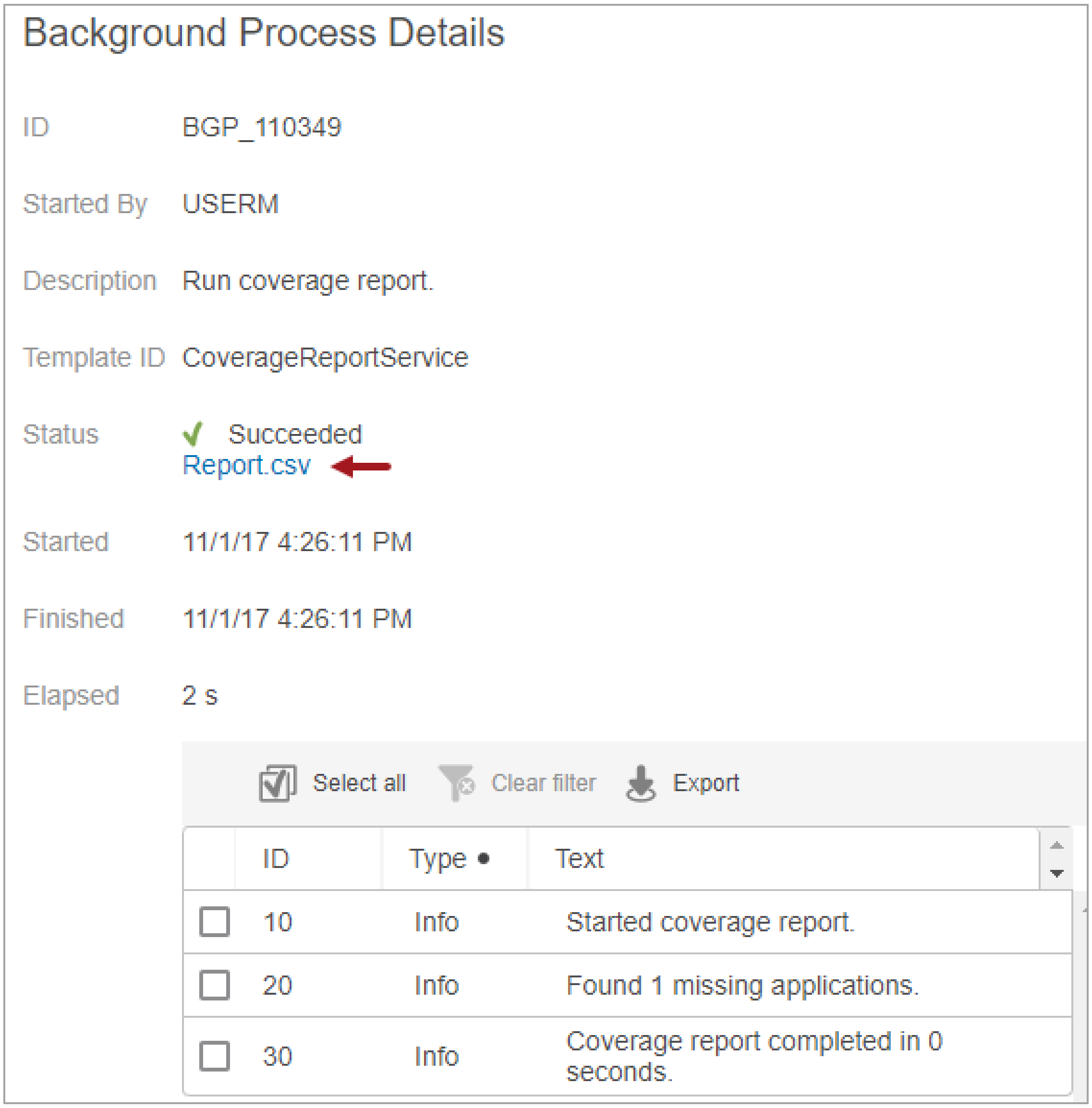
In an application editor, it is possible to run a coverage report for missing applications. The coverage report is available in a .CSV file format and will list the missing application's Base Vehicle and Part Terminology IDs.



Below are the steps to run a coverage report:

1. Enter both a vehicle and part type selection within the respective finders.
2. Click the Coverage report button, and a background process notification will display.
3. Click the background process ID to view the Background Process Details screen.

4. Once the background process has succeeded, the background process table will list the details of the process and beneath the Status label, a 'Report.csv' link will display as shown below.



Background Process Details

ID BGP_110349

Started By USERM

Description Run coverage report.

Template ID CoverageReportService

Status ✓ Succeeded
Report.csv

Started 11/1/17 4:26:11 PM

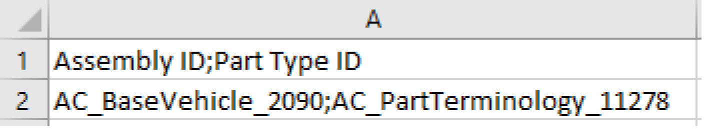
Finished 11/1/17 4:26:11 PM

Elapsed 2 s

	ID	Type ●	Text
□	10	Info	Started coverage report.
□	20	Info	Found 1 missing applications.
□	30	Info	Coverage report completed in 0 seconds.

5. Click the 'Report.csv' link to download the report.

6. When the .CSV file is opened, the missing application's Base Vehicle and Part Terminology IDs will display as shown below.



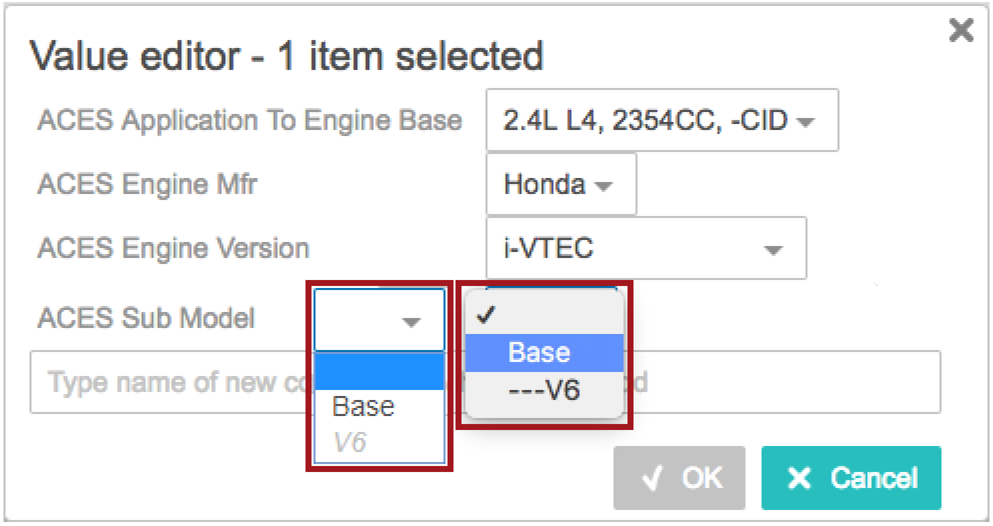
	A
1	Assembly ID;Part Type ID
2	AC_BaseVehicle_2090;AC_PartTerminology_11278

Using Multi-Select

In the application editor, 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 I have three records selected and I double click the cell in the middle record, those are the data values that will display.

Using the Value Editor

Invalid values (options) do not change style (grey out, italic font) on Mac computers, but are instead represented with three dashes (---)before the invalid value. For example, in the screenshot below the ACES Sub Model dropdown on the left is from a PC using the Chrome browser, and the dropdown on the right is from a Mac using the Chrome browser. This problem exist only on Mac computers for both Safari and Chrome browsers.



Value editor - 1 item selected

ACES Application To Engine Base: 2.4L L4, 2354CC, -CID

ACES Engine Mfr: Honda

ACES Engine Version: i-VTEC

ACES Sub Model

Type name of new c

Variations Across Standards

There are terminology variations in the application editors for each standard, as well as minor variations in the available headers.

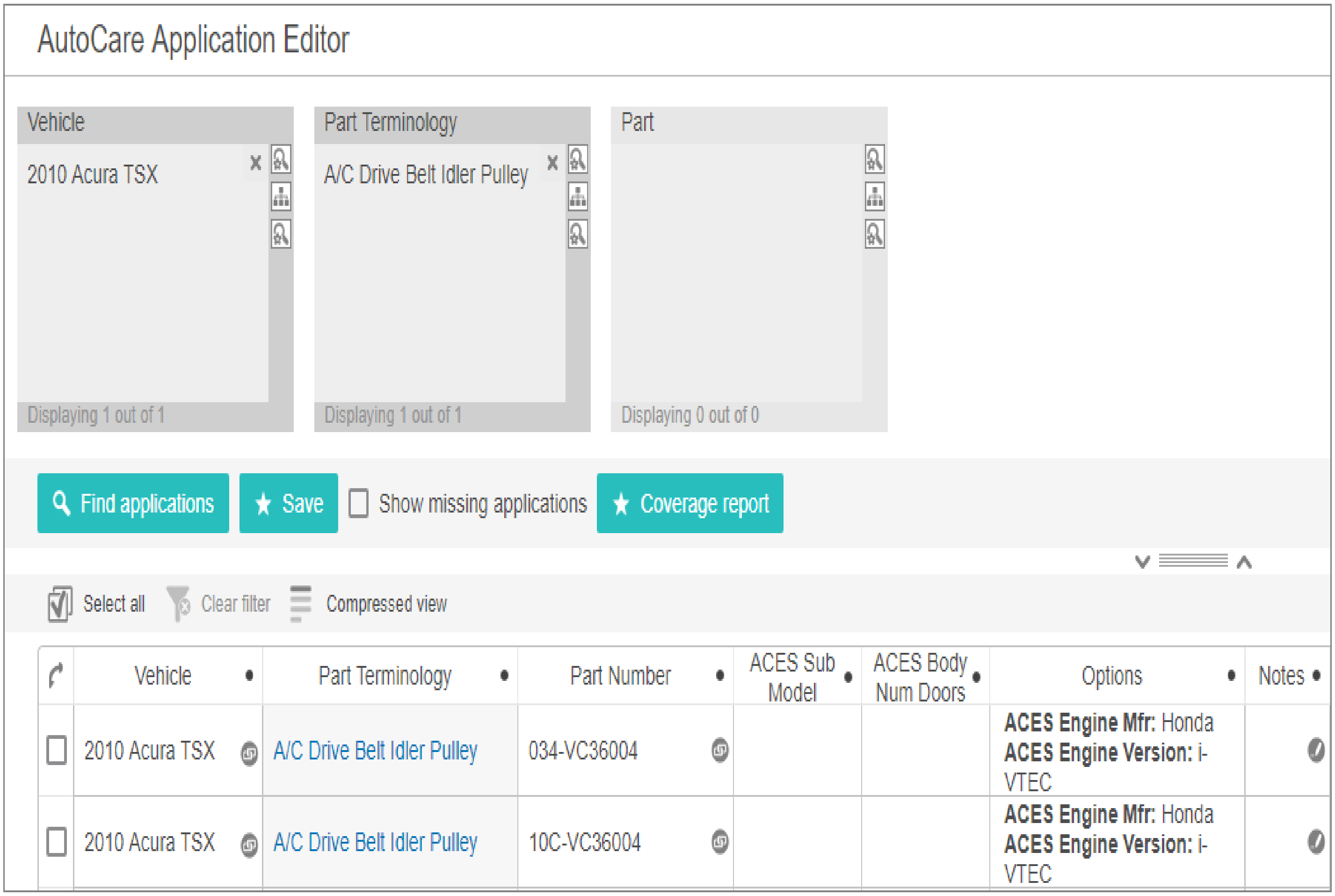
For example:

- TecDoc editor offers search options for Assembly, Generic Article, and Article.
- AutoCare editor offers search options labeled as Vehicle, Part Terminology, and Part.
- AutoCare editor contains a Qualifiers header, whereas a TecDoc editor does not.

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 (unless a read-only configuration is applied).

For example, an AutoCare editor is shown below.



Vehicle: 2010 Acura TSX

Part Terminology: A/C Drive Belt Idler Pulley

☐ Show missing applications

	Vehicle	Part Terminology	Part Number	ACES Sub Model	ACES Body Num Doors	Options	Notes
☐	2010 Acura TSX	A/C Drive Belt Idler Pulley	034-VC36004			**ACES Engine Mfr:** Honda **ACES Engine Version:** i-VTEC	
☐	2010 Acura TSX	A/C Drive Belt Idler Pulley	10C-VC36004			**ACES Engine Mfr:** Honda **ACES Engine Version:** i-VTEC	

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.

Allow Applications Outside Part Type

When using an Application Editor, if a part number is found using the part type search, but the part number is being applicated to a part type that differs from the one used to search, an error dialog will display informing the user that the part cannot be linked because the part type is different. In other words, when using the application editor to applicate parts, a parts part type must match the applications part type.

However, it is possible to override this, allowing users to select a part number with a part type that does not match the part type searched for application. In other words, the Application Editor will ignore the searched part type when assigning the part number to the application. This is helpful when working with a part that can only have a single part type, but can be applicated for multiple part types.

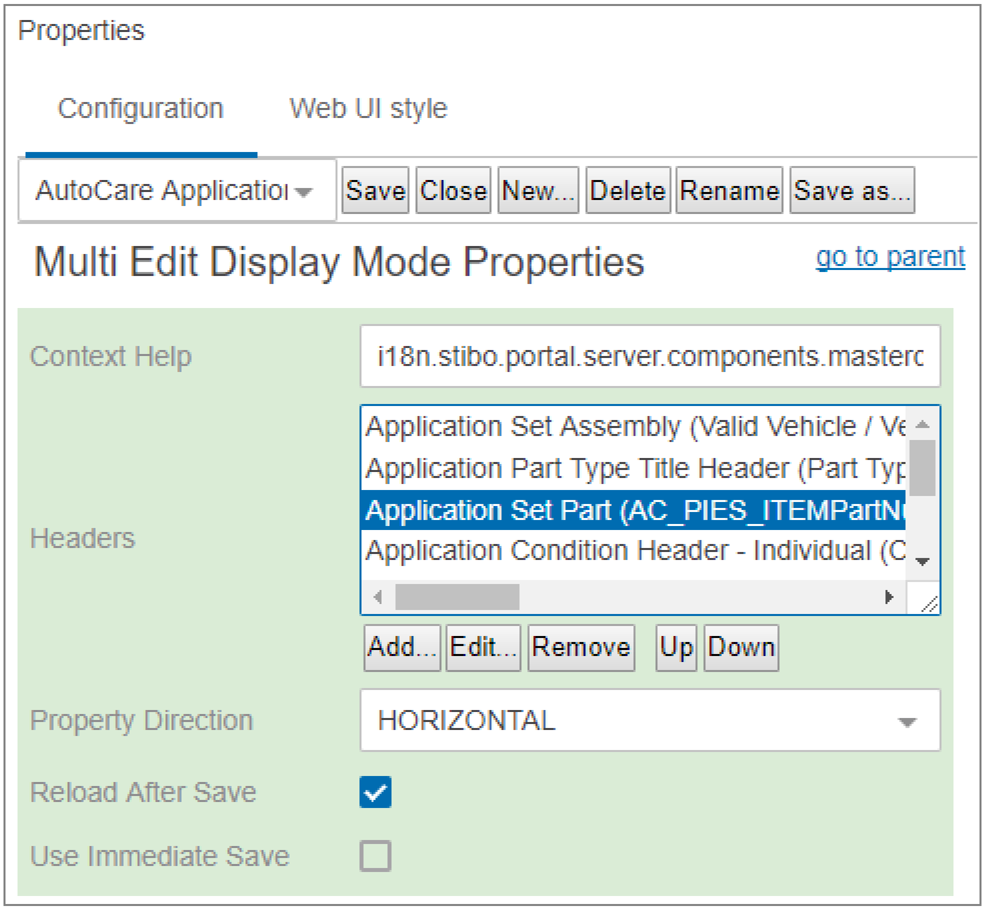
Prerequisites

The new parameter will need to be added by running **Automotive - [Standard] Model > 4. Configure [Standard] WebUI** Easy Setup.

Enabling the Allow Applications Outside Part Type

Below are the steps to enable the Allow Applications Outside Part Type parameter.

1. Go to the Application Editor's Multi Edit Display mode Properties > Headers.
2. Select **Application Set Part**.



Properties

Configuration   Web UI style

AutoCare Applicatio▾ | Save | Close | New... | Delete | Rename | Save as...

Multi Edit Display Mode Properties   go to parent

Context Help: i18n.stibo.portal.server.components.masterc

Headers:
Application Set Assembly (Valid Vehicle / Ve
Application Part Type Title Header (Part Typ
Application Set Part (AC_PIES_ITEMPartNu
Application Condition Header - Individual (C

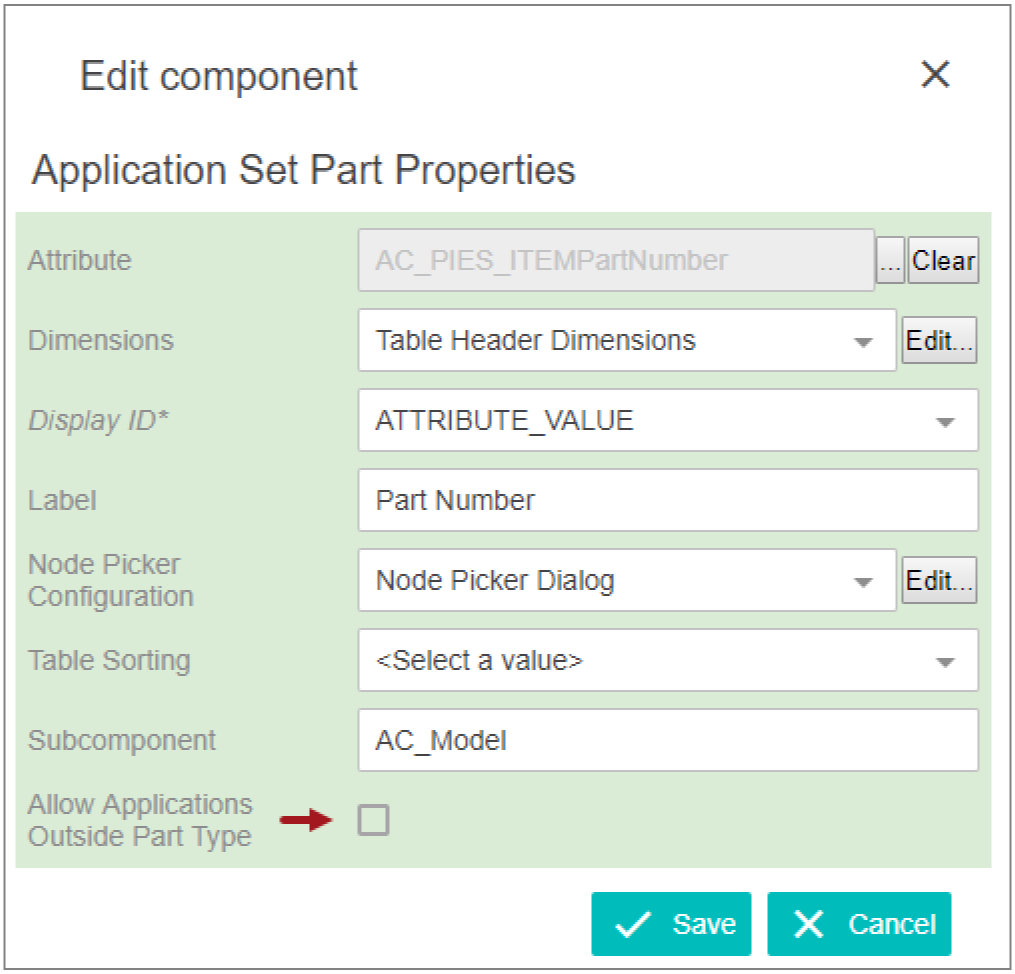
Add... | Edit... | Remove | Up | Down

Property Direction: HORIZONTAL

Reload After Save: ☑

Use Immediate Save: □

3. Click the **Edit** button, and the Application Set Part Properties will display.
4. Click the checkbox for the **Allow Applications Outside Part Type**.



Edit component

Application Set Part Properties

Attribute	AC_PIES_ITEMPartNumber
Dimensions	Table Header Dimensions
*Display ID**	ATTRIBUTE_VALUE
Label	Part Number
Node Picker Configuration	Node Picker Dialog
Table Sorting	<Select a value>
Subcomponent	AC_Model
Allow Applications Outside Part Type	□

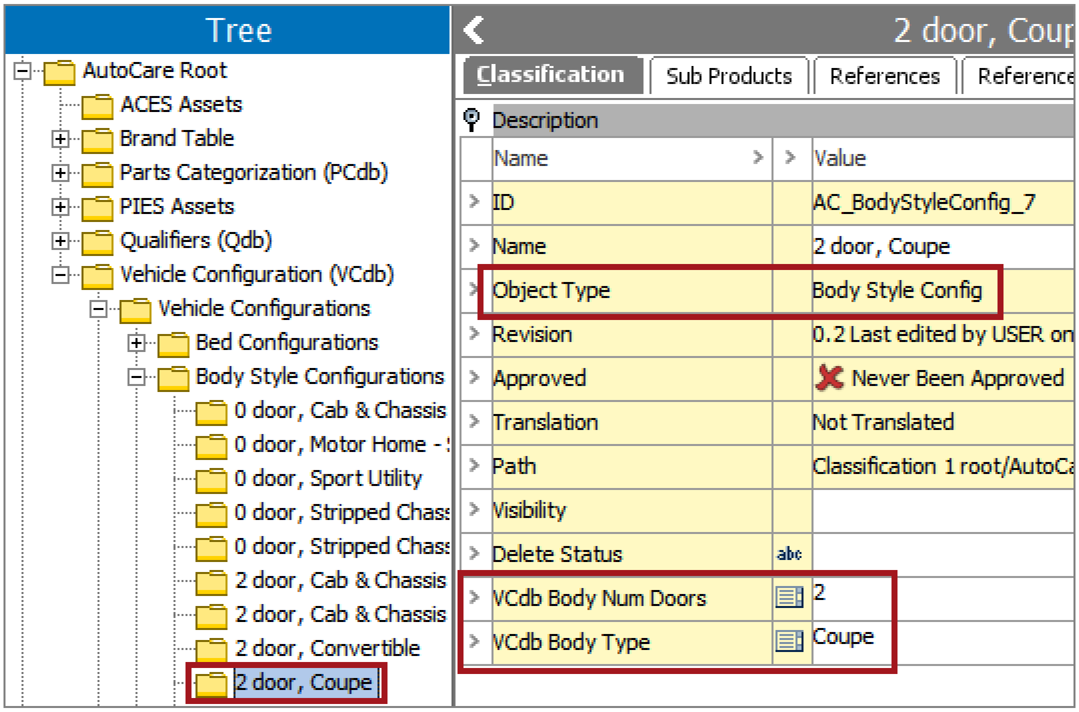
5. Click the **Save** and **Close** buttons to save and close the designer.

Automotive Validation Paths

A validation path concept is used to validate application data created using the Application Editor in Web UI. The premise behind the concept is that all of the standards provide some level of "valid configuration" data for vehicles (e.g., a vehicle with two doors can be considered a coupe, but a vehicle with four doors cannot be a coupe and may instead be a sedan). A vehicle may be available in both two and four door configurations, but then is only valid with the corresponding body style.

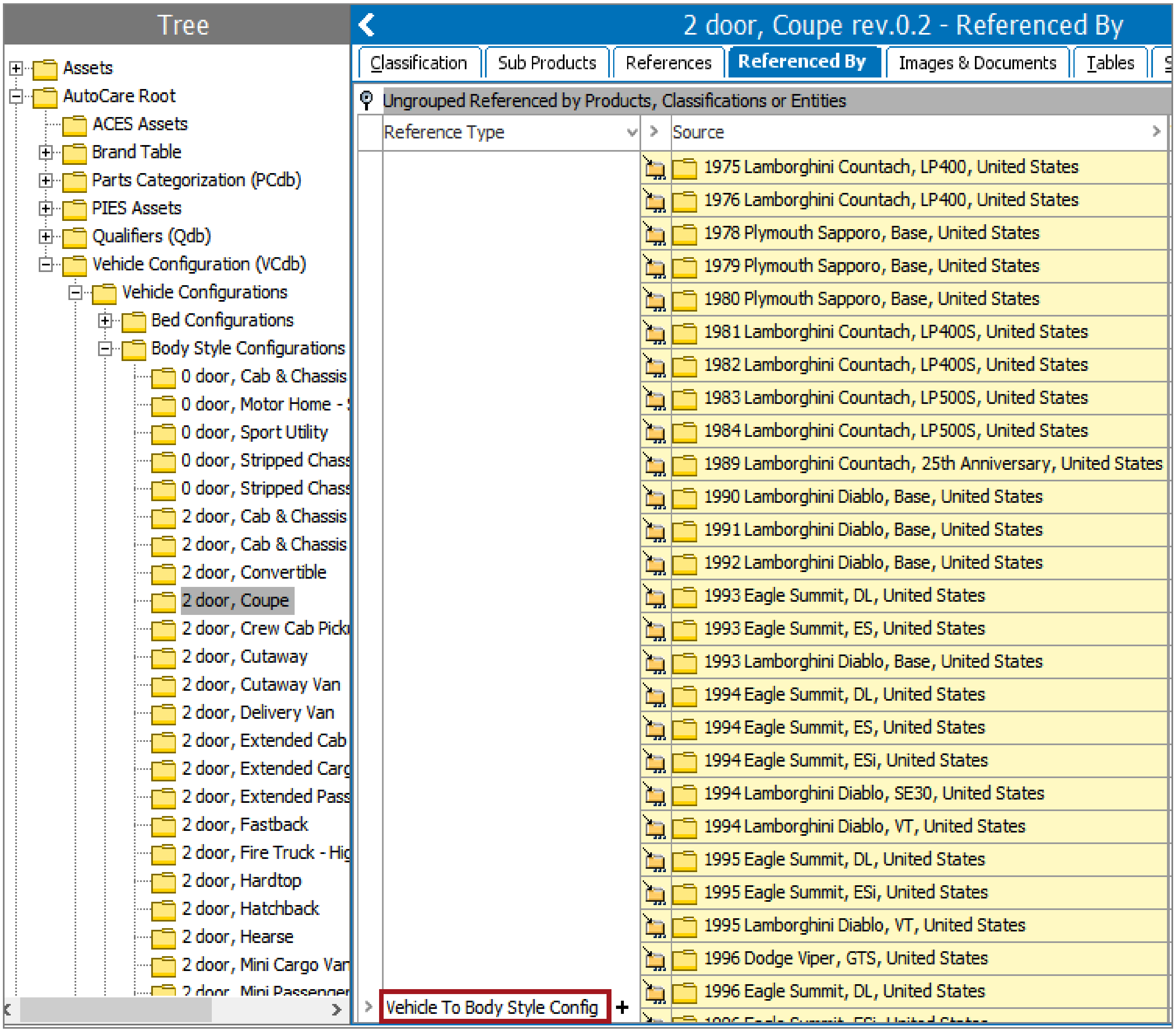
The validation data is represented in STEP via a series of objects, references, and attributes, which are then applied to individual applications as either attributes or references.

Consider the 'VCdb Body Num Doors' and 'VCdb Body Type' options in the AutoCare model. These are modeled via classification objects of the 'Body Style Config' type that hold the data for both options. For example, there is a valid configuration of a vehicle with '2' doors and a 'Coupe' body type, named '2 door, Coupe.'



Name		Value
ID		AC_BodyStyleConfig_7
Name		2 door, Coupe
Object Type		Body Style Config
Revision		0.2 Last edited by USER on
Approved		✗ Never Been Approved
Translation		Not Translated
Path		Classification 1 root/AutoCa
Visibility		
Delete Status	abc	
VCdb Body Num Doors	☰	2
VCdb Body Type	☰	Coupe

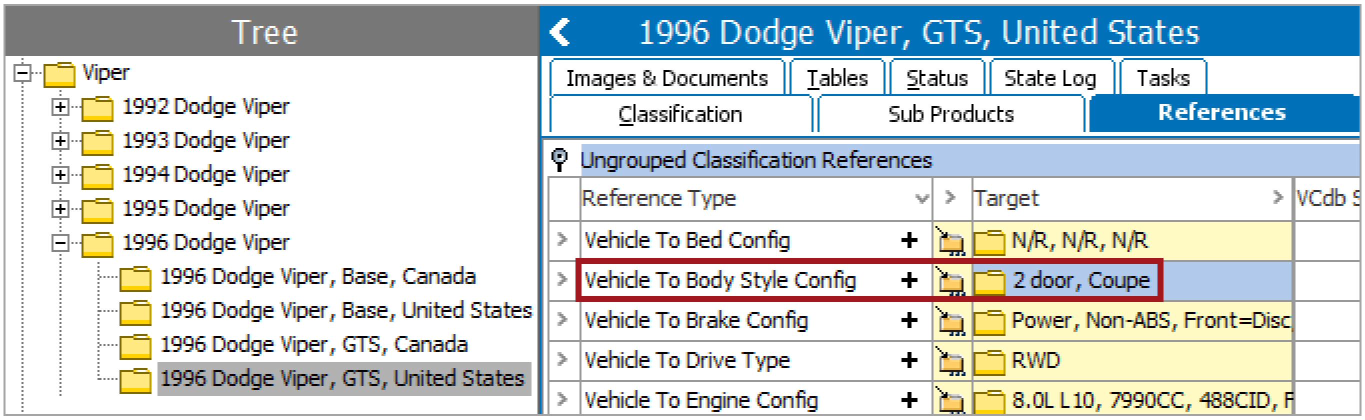
All vehicles with the '2 door, Coupe' body style reference this configuration using the 'Vehicle To Body Style Config' reference (as shown below).


Tree
2 door, Coupe rev.0.2 - Referenced By
Classification
Sub Products
References
Referenced By
Images & Documents
Tables
Ungrouped Referenced by Products, Classifications or Entities

- Assets
- AutoCare Root
  - ACES Assets
  - Brand Table
  - Parts Categorization (PCdb)
  - PIES Assets
  - Qualifiers (Qdb)
  - Vehicle Configuration (VCdb)
    - Vehicle Configurations
      - Bed Configurations
      - Body Style Configurations
        - 0 door, Cab & Chassis
        - 0 door, Motor Home - !
        - 0 door, Sport Utility
        - 0 door, Stripped Chass
        - 0 door, Stripped Chass
        - 2 door, Cab & Chassis
        - 2 door, Cab & Chassis
        - 2 door, Convertible
        - 2 door, Coupe
        - 2 door, Crew Cab Pick
        - 2 door, Cutaway
        - 2 door, Cutaway Van
        - 2 door, Delivery Van
        - 2 door, Extended Cab
        - 2 door, Extended Carg
        - 2 door, Extended Pass
        - 2 door, Fastback
        - 2 door, Fire Truck - Hig
        - 2 door, Hardtop
        - 2 door, Hatchback
        - 2 door, Hearse
        - 2 door, Mini Cargo Var
        - 2 door, Mini Passenger

Reference Type	Source
Vehicle To Body Style Config	1975 Lamborghini Countach, LP400, United States 1976 Lamborghini Countach, LP400, United States 1978 Plymouth Sapporo, Base, United States 1979 Plymouth Sapporo, Base, United States 1980 Plymouth Sapporo, Base, United States 1981 Lamborghini Countach, LP400S, United States 1982 Lamborghini Countach, LP400S, United States 1983 Lamborghini Countach, LP500S, United States 1984 Lamborghini Countach, LP500S, United States 1989 Lamborghini Countach, 25th Anniversary, United States 1990 Lamborghini Diablo, Base, United States 1991 Lamborghini Diablo, Base, United States 1992 Lamborghini Diablo, Base, United States 1993 Eagle Summit, DL, United States 1993 Eagle Summit, ES, United States 1993 Lamborghini Diablo, Base, United States 1994 Eagle Summit, DL, United States 1994 Eagle Summit, ES, United States 1994 Eagle Summit, ESi, United States 1994 Lamborghini Diablo, SE30, United States 1994 Lamborghini Diablo, VT, United States 1995 Eagle Summit, DL, United States 1995 Eagle Summit, ESi, United States 1995 Lamborghini Diablo, VT, United States 1996 Dodge Viper, GTS, United States 1996 Eagle Summit, DL, United States 1996 Eagle Summit, ESi, United States

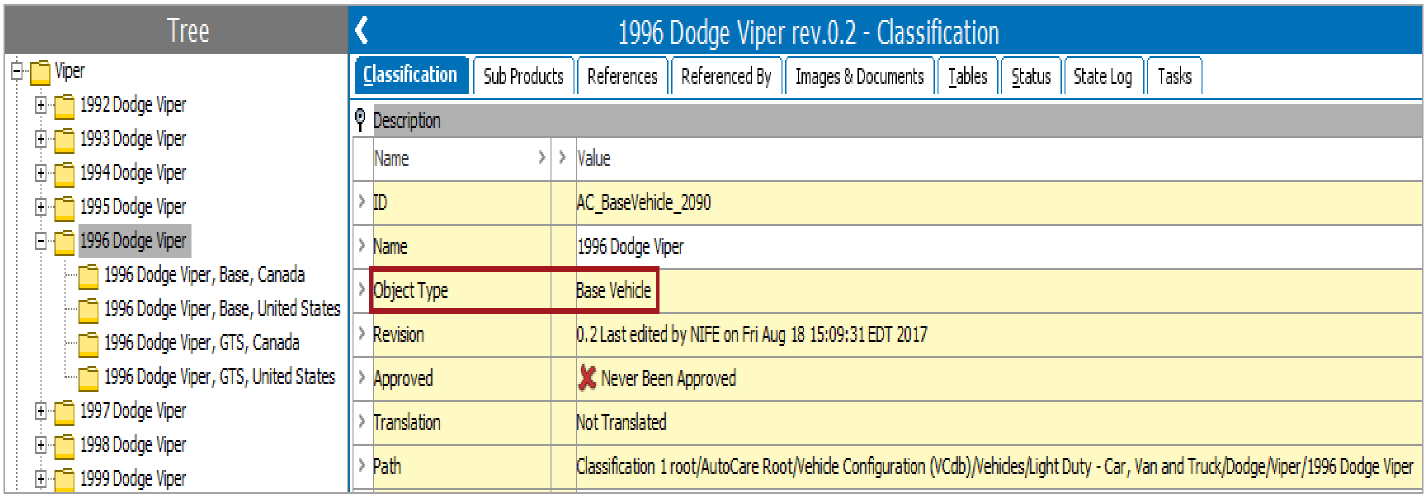
A vehicle can have more than one 'Vehicle To Body Style Config' reference and if so, it will display as multiple references on the vehicle. But let's consider the '1996 Dodge Viper GTS, United States' which has only one valid 'Vehicle To Body Style Config' reference (as shown below).


Tree
Viper
1992 Dodge Viper
1993 Dodge Viper
1994 Dodge Viper
1995 Dodge Viper
1996 Dodge Viper
1996 Dodge Viper, Base, Canada
1996 Dodge Viper, Base, United States
1996 Dodge Viper, GTS, Canada
1996 Dodge Viper, GTS, United States
1996 Dodge Viper, GTS, United States
Images & Documents
Tables
Status
State Log
Tasks
Classification
Sub Products
References
Ungrouped Classification References

Reference Type	Target	VCdb S
Vehicle To Bed Config	N/R, N/R, N/R	
Vehicle To Body Style Config	2 door, Coupe	
Vehicle To Brake Config	Power, Non-ABS, Front=Disc,	
Vehicle To Drive Type	RWD	
Vehicle To Engine Config	8.0L L10, 7990CC, 488CID, F	

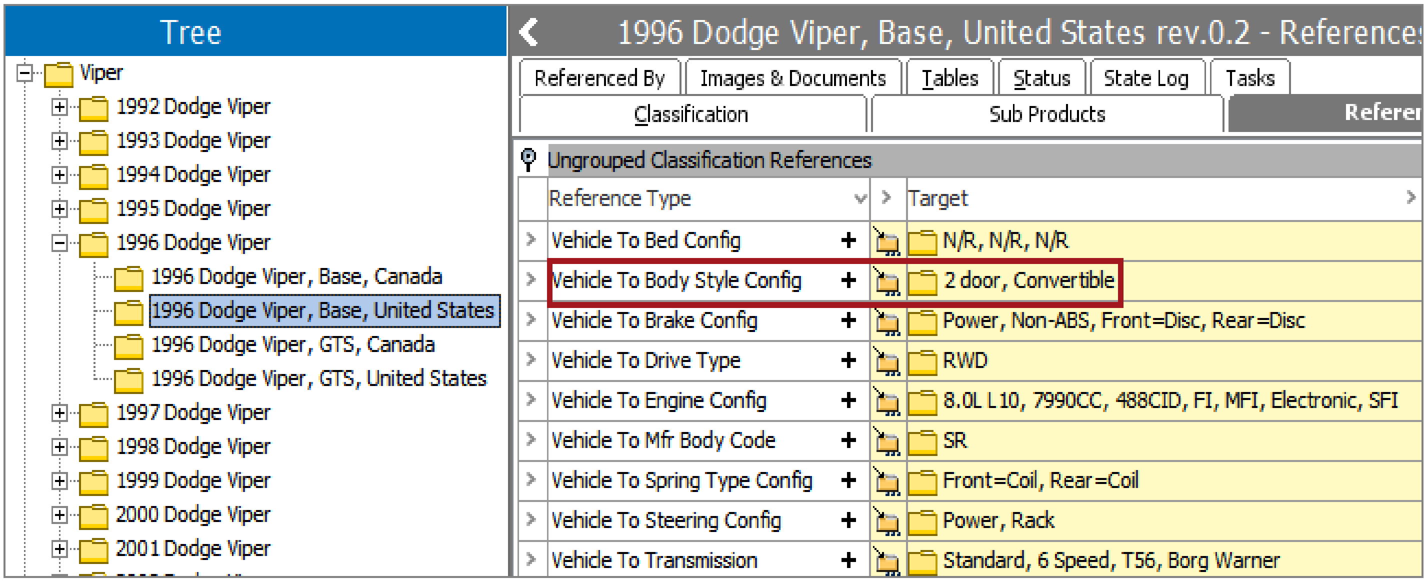
As there is only one valid body style for this vehicle, it would be invalid for an application to be supplied for the vehicle where any other body style was implied (via providing a Body Num Doors value other than 2, or a Body Type value other than Coupe).

However, applications are applied to Base Vehicle object types (not Vehicles themselves). In this case, that means an application is not applied to the Vehicle object type mentioned above ('1996 Dodge Viper, GTS, United States'), but instead applied to the Base Vehicle object type '1996 Dodge Viper' (shown below with four Vehicle object type children).



Name	Value
ID	AC_BaseVehicle_2090
Name	1996 Dodge Viper
Object Type	Base Vehicle
Revision	0.2 Last edited by NIFE on Fri Aug 18 15:09:31 EDT 2017
Approved	✗ Never Been Approved
Translation	Not Translated
Path	Classification 1 root/AutoCare Root/Vehicle Configuration (VCdb)/Vehicles/Light Duty - Car, Van and Truck/Dodge/Viper/1996 Dodge Viper

In order to know all valid body style configurations for a Base Vehicle object type, we must evaluate all child vehicles (the 'Base' and 'GTS' vehicles for the United States and Canada). In doing this, it can be seen that the 1996 Dodge Viper Base models have a '2 door, Convertible' body style (as shown below) whereas the 1996 Dodge Viper GTS models have the 2 door, Coupe body style (as previously mentioned).



Tree

- Viper
  - 1992 Dodge Viper
  - 1993 Dodge Viper
  - 1994 Dodge Viper
  - 1995 Dodge Viper
  - 1996 Dodge Viper
    - 1996 Dodge Viper, Base, Canada
    - 1996 Dodge Viper, Base, United States
    - 1996 Dodge Viper, GTS, Canada
    - 1996 Dodge Viper, GTS, United States
  - 1997 Dodge Viper
  - 1998 Dodge Viper
  - 1999 Dodge Viper
  - 2000 Dodge Viper
  - 2001 Dodge Viper

1996 Dodge Viper, Base, United States rev.0.2 - References

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

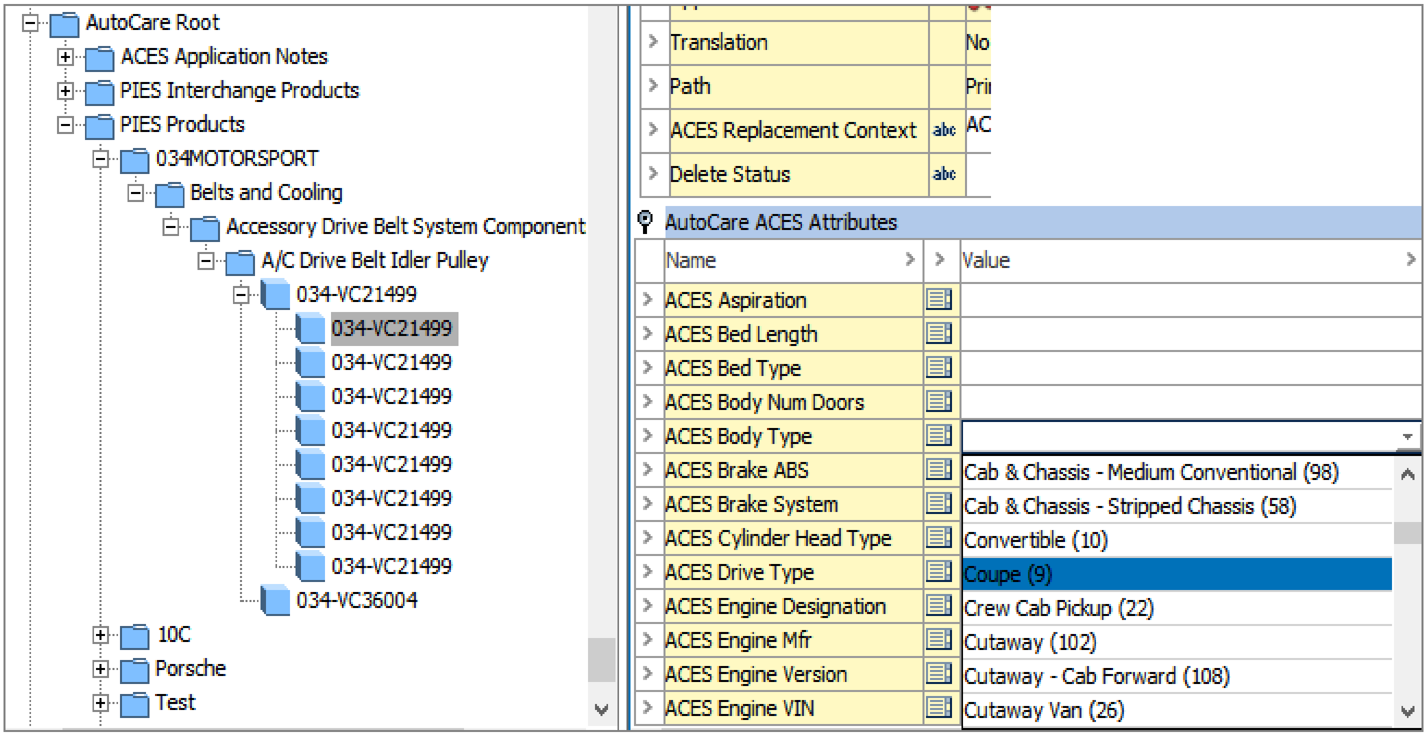
Classification | Sub Products | Referen

Ungrouped Classification References

Reference Type	Target
Vehicle To Bed Config	N/R, N/R, N/R
Vehicle To Body Style Config	2 door, Convertible
Vehicle To Brake Config	Power, Non-ABS, Front=Disc, Rear=Disc
Vehicle To Drive Type	RWD
Vehicle To Engine Config	8.0L L10, 7990CC, 488CID, FI, MFI, Electronic, SFI
Vehicle To Mfr Body Code	SR
Vehicle To Spring Type Config	Front=Coil, Rear=Coil
Vehicle To Steering Config	Power, Rack
Vehicle To Transmission	Standard, 6 Speed, T56, Borg Warner

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

On an application, an attribute or reference exists to model each data point in a valid configuration. For the reference type 'Vehicle To Body Style Config' we have an attribute for 'ACES Body Num Doors' and an attribute for 'ACES Body Type,' similar to what we saw on the configuration classification object. In the workbench example below, we can see that all available body types can be selected.



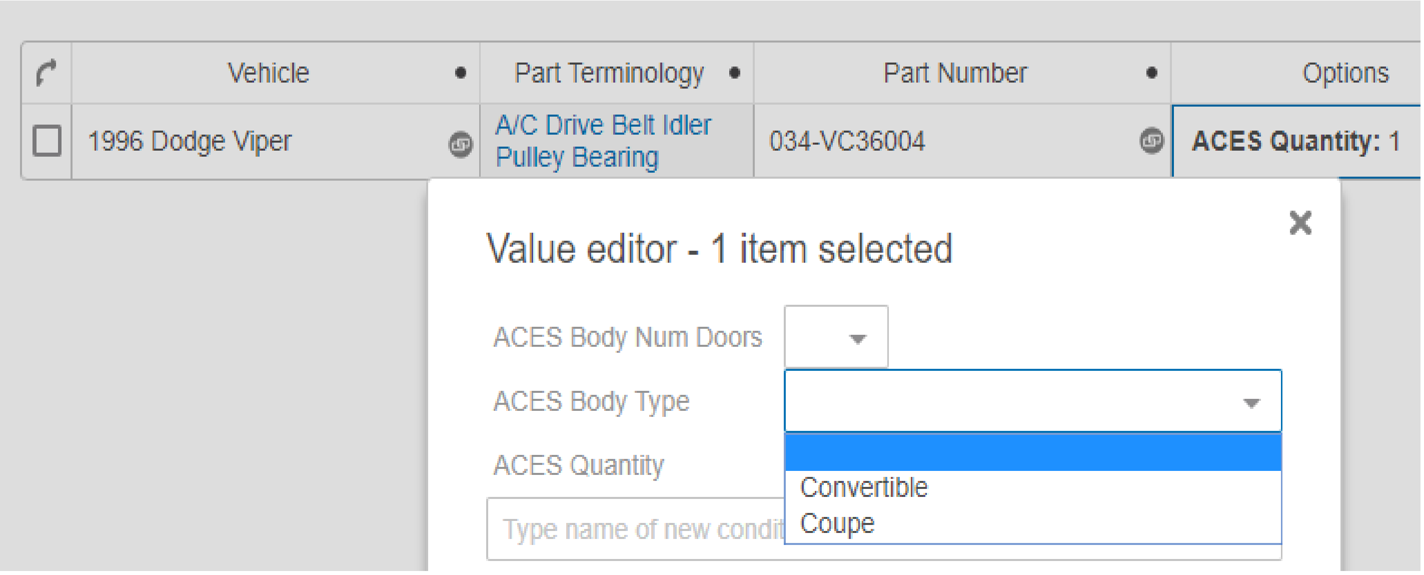
- AutoCare Root
  - ACES Application Notes
  - PIES Interchange Products
  - PIES Products
    - 034MOTORSPORT
      - Belts and Cooling
        - Accessory Drive Belt System Component
          - A/C Drive Belt Idler Pulley
            - 034-VC21499
              - 034-VC21499
              - 034-VC21499
              - 034-VC21499
              - 034-VC21499
              - 034-VC21499
              - 034-VC21499
              - 034-VC21499
              - 034-VC21499
            - 034-VC36004
    - 10C
    - Porsche
    - Test

Translation	No
Path	Pri
ACES Replacement Context	AC
Delete Status	

AutoCare ACES Attributes

Name	Value
ACES Aspiration	
ACES Bed Length	
ACES Bed Type	
ACES Body Num Doors	
ACES Body Type	
ACES Brake ABS	Cab & Chassis - Medium Conventional (98)
ACES Brake System	Cab & Chassis - Stripped Chassis (58)
ACES Cylinder Head Type	Convertible (10)
ACES Drive Type	Coupe (9)
ACES Engine Designation	Crew Cab Pickup (22)
ACES Engine Mfr	Cutaway (102)
ACES Engine Version	Cutaway - Cab Forward (108)
ACES Engine VIN	Cutaway Van (26)

However, applications are not intended to be edited in the workbench. The application editor is instead provided, and there we can see that only the Coupe and Convertible options are available when we are working with a 1996 Dodge Viper vehicle:



	Vehicle	Part Terminology	Part Number	Options
□	1996 Dodge Viper	A/C Drive Belt Idler Pulley Bearing	034-VC36004	**ACES Quantity:** 1

Value editor - 1 item selected

ACES Body Num Doors

ACES Body Type

- Convertible
- Coupe

ACES Quantity

Type name of new condit

Important: It is the validation path concept that drives the filtered display in the application record editor.

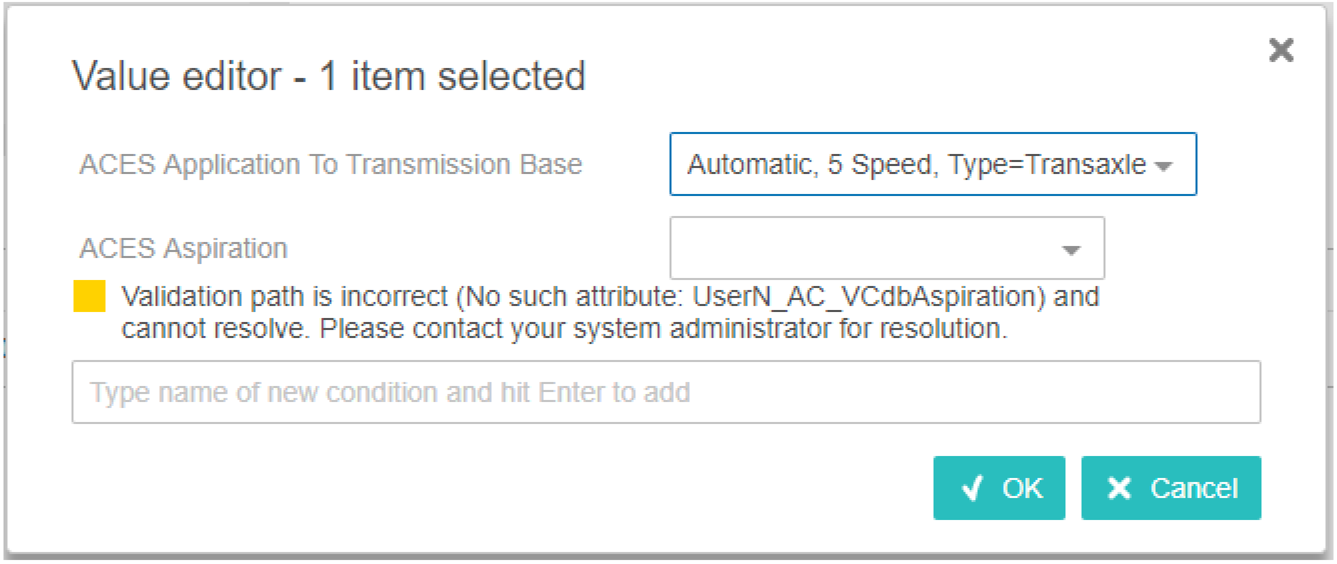
Error Handling in Application Editor

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.



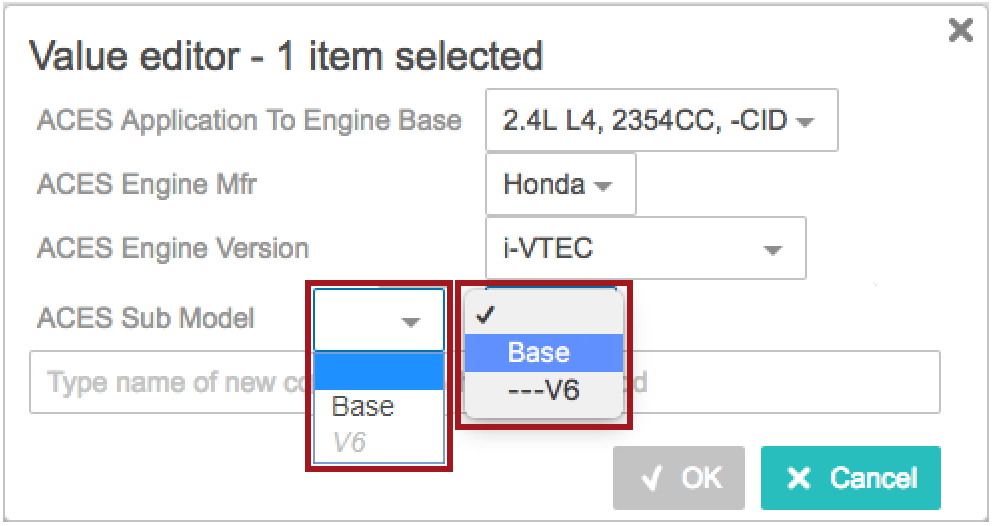
Value editor - 1 item selected

ACES Application To Transmission Base: Automatic, 5 Speed, Type=Transaxle

ACES Aspiration:

🟨 Validation path is incorrect (No such attribute: UserN_AC_VCdbAspiration) and cannot resolve. Please contact your system administrator for resolution.

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

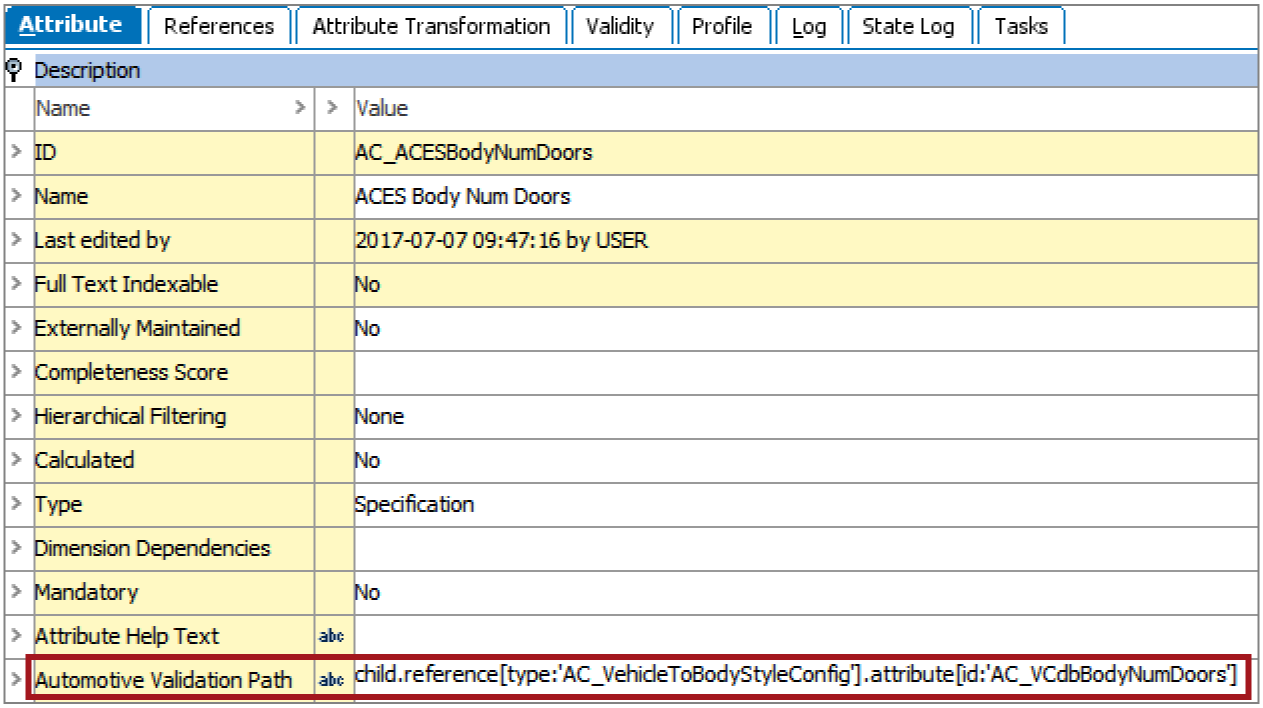
ACES Engine Version: i-VTEC

ACES Sub Model:

Validation Paths for Attributes

When the data model uses an attribute to store the vehicle option / criteria, the 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. For example:



Name		Value
ID		AC_ACESBodyNumDoors
Name		ACES Body Num Doors
Last edited by		2017-07-07 09:47:16 by USER
Full Text Indexable		No
Externally Maintained		No
Completeness Score		
Hierarchical Filtering		None
Calculated		No
Type		Specification
Dimension Dependencies		
Mandatory		No
Attribute Help Text	abc	
Automotive Validation Path	abc	child.reference[type:'AC_VehicleToBodyStyleConfig'].attribute[id:'AC_VCdbBodyNumDoors']

Regardless of the model / standard you are working in, the system always evaluates the validation path by beginning with the base vehicle / assembly object used in application records.

In the above example, the data within the data model comes by looking at the base vehicle, examining all vehicle children of the base vehicle, following the Vehicle To Body Style Config (AC_VehicleToBodyStyleConfig) reference on the vehicle to any targets, and evaluating the value of the VCdb Body Num Doors attribute (AC_VCdbBodyNumDoors) on that object.

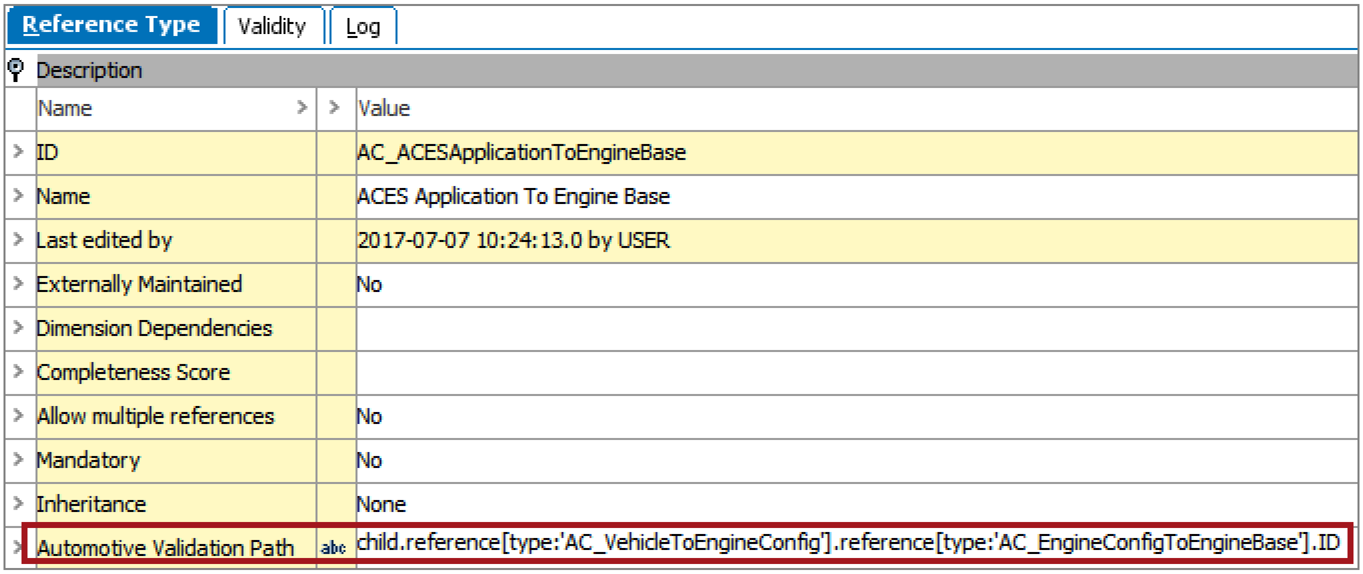
The syntax of the validation path is a series of commands, each separated by a period (.). Using the child, reference, ID (shown below), and attribute elements (including IDs of the applicable references and attributes to be followed), a validation path can be applied to any attribute for which the data is modeled in STEP. When the vehicle option is modeled using an attribute, the final element of the syntax should always be an attribute.

Note: Within the AutoCare model, validation paths are populated on all standard ACES vehicle options as part of easy setup when the attribute modeling the option is first created. If a validation path is subsequently deleted, it can be reapplied manually using the syntax described above. Alternatively, the attribute can be deleted and easy setup can be re-run (System Setup > Component Models > Automotive - AutoCare Model > 1. Configure AutoCare Data Model). However, deleting the attribute will also delete all existing data for the attribute so it is recommended to re-populate deleted validation paths manually on production environments.

Validation Paths for References

When the data model uses a reference to store the vehicle option / criteria, the 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). The validation path attribute can then be populated on any reference. For example:



Name		Value
ID		AC_ACESApplicationToEngineBase
Name		ACES Application To Engine Base
Last edited by		2017-07-07 10:24:13.0 by USER
Externally Maintained		No
Dimension Dependencies		
Completeness Score		
Allow multiple references		No
Mandatory		No
Inheritance		None
Automotive Validation Path	abc	child.reference[type:'AC_VehicleToEngineConfig'].reference[type:'AC_EngineConfigToEngineBase'].ID

In the above example, the data within the data model comes by looking at the base vehicle, examining all vehicle children of the base vehicle, following the Vehicle To Engine Config (AC_VehicleToEngineConfig) reference on the vehicle to any targets, following the Engine Config To Engine Base reference (AC_EngineConfigToEngineBase) from those initial targets (which are now considered sources of the Engine Config To Engine Base reference), and identifying the target of that reference via it's ID.

The syntax of the 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), a validation path can be applied to any attribute 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.

Note: Within the AutoCare model, validation paths are populated on all standard ACES vehicle options as part of easy setup when the reference modeling option is first created. If a validation path is subsequently deleted, it can be reapplied manually using the syntax described above. Alternatively, the reference can be deleted and easy setup can be re-run (System Setup > Component Models > Automotive - AutoCare Model > 1. Configure AutoCare Data Model). However, deleting the reference will also delete all existing data for the reference so it is recommended to re-populate deleted validation paths manually on production environments.

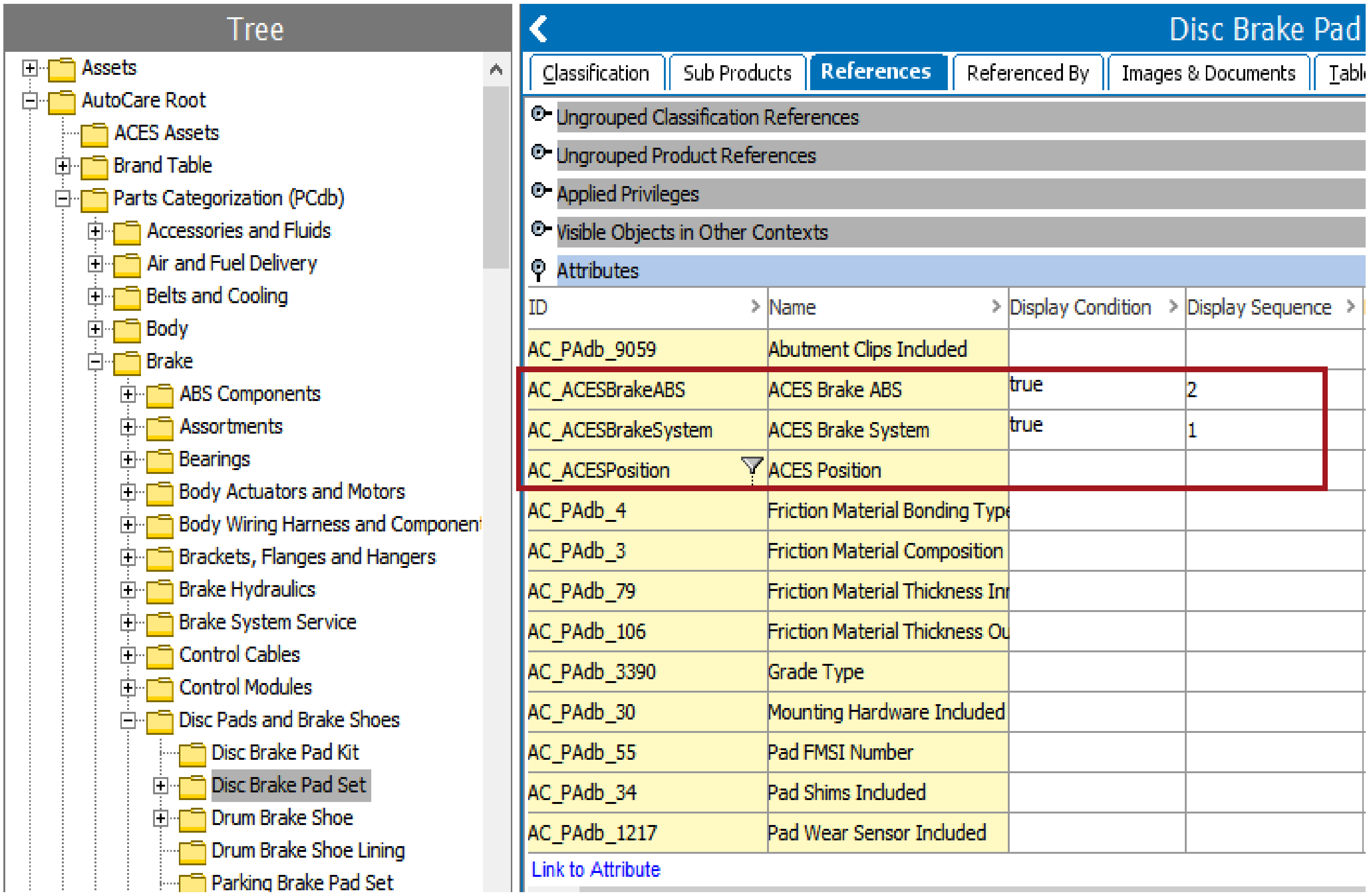
Controlling Display of Conditions in the Editor

The application editor is configured with two columns to display vehicle option data on applications:

- **Application Condition Header - Individual**: This is used to allow table headers to be expanded to display conditions in individual columns. Condition attributes that are linked to the part type of the application and have metadata DisplayCondition=true will be displayed in their own column in the results table using this header.
- **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 appears by default in the value editor or must be searched for, is all 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. For example:



Tree

- Assets
- AutoCare Root
  - ACES Assets
  - Brand Table
  - Parts Categorization (PCdb)
    - Accessories and Fluids
    - Air and Fuel Delivery
    - Belts and Cooling
    - Body
    - Brake
      - ABS Components
      - Assortments
      - Bearings
      - Body Actuators and Motors
      - Body Wiring Harness and Component
      - Brackets, Flanges and Hangers
      - Brake Hydraulics
      - Brake System Service
      - Control Cables
      - Control Modules
      - Disc Pads and Brake Shoes
        - Disc Brake Pad Kit
        - Disc Brake Pad Set
        - Drum Brake Shoe
        - Drum Brake Shoe Lining
        - Parking Brake Pad Set

Disc Brake Pad

Ungrouped Classification References

Ungrouped Product References

Applied Privileges

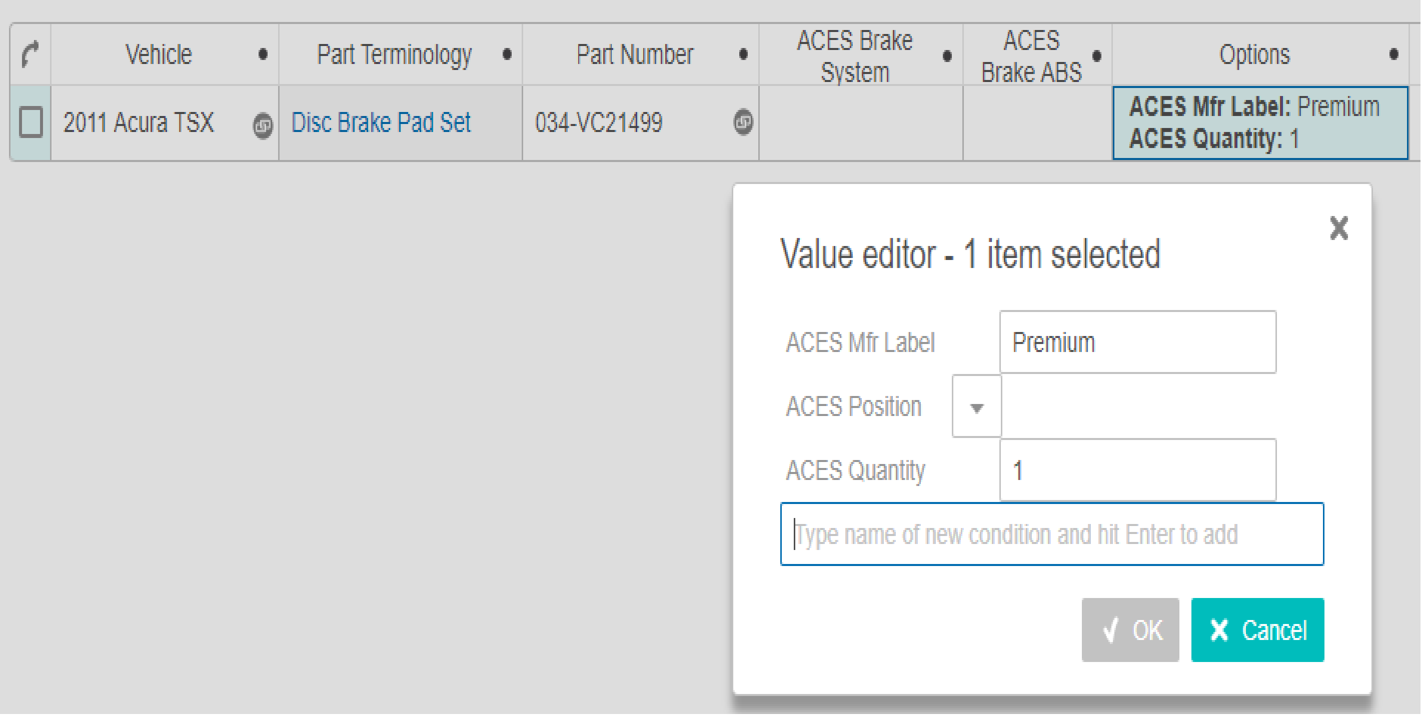
Visible Objects in Other Contexts

Attributes

ID	Name	Display Condition	Display Sequence
AC_PAdb_9059	Abutment Clips Included		
AC_ACESBrakeABS	ACES Brake ABS	true	2
AC_ACESBrakeSystem	ACES Brake System	true	1
AC_ACESPosition	ACES Position		
AC_PAdb_4	Friction Material Bonding Type		
AC_PAdb_3	Friction Material Composition		
AC_PAdb_79	Friction Material Thickness Inr		
AC_PAdb_106	Friction Material Thickness Ou		
AC_PAdb_3390	Grade Type		
AC_PAdb_30	Mounting Hardware Included		
AC_PAdb_55	Pad FMSI Number		
AC_PAdb_34	Pad Shims Included		
AC_PAdb_1217	Pad Wear Sensor Included		

Link to Attribute

Brake ABS and Brake System both are linked to the part type and have a true Display Condition so they are displayed in their own columns within the editor, in the order indicated by the Display Sequence. Mfr Label and 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. 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.



	Vehicle	Part Terminology	Part Number	ACES Brake System	ACES Brake ABS	Options
□	2011 Acura TSX	Disc Brake Pad Set	034-VC21499			**ACES Mfr Label:** Premium **ACES Quantity:** 1

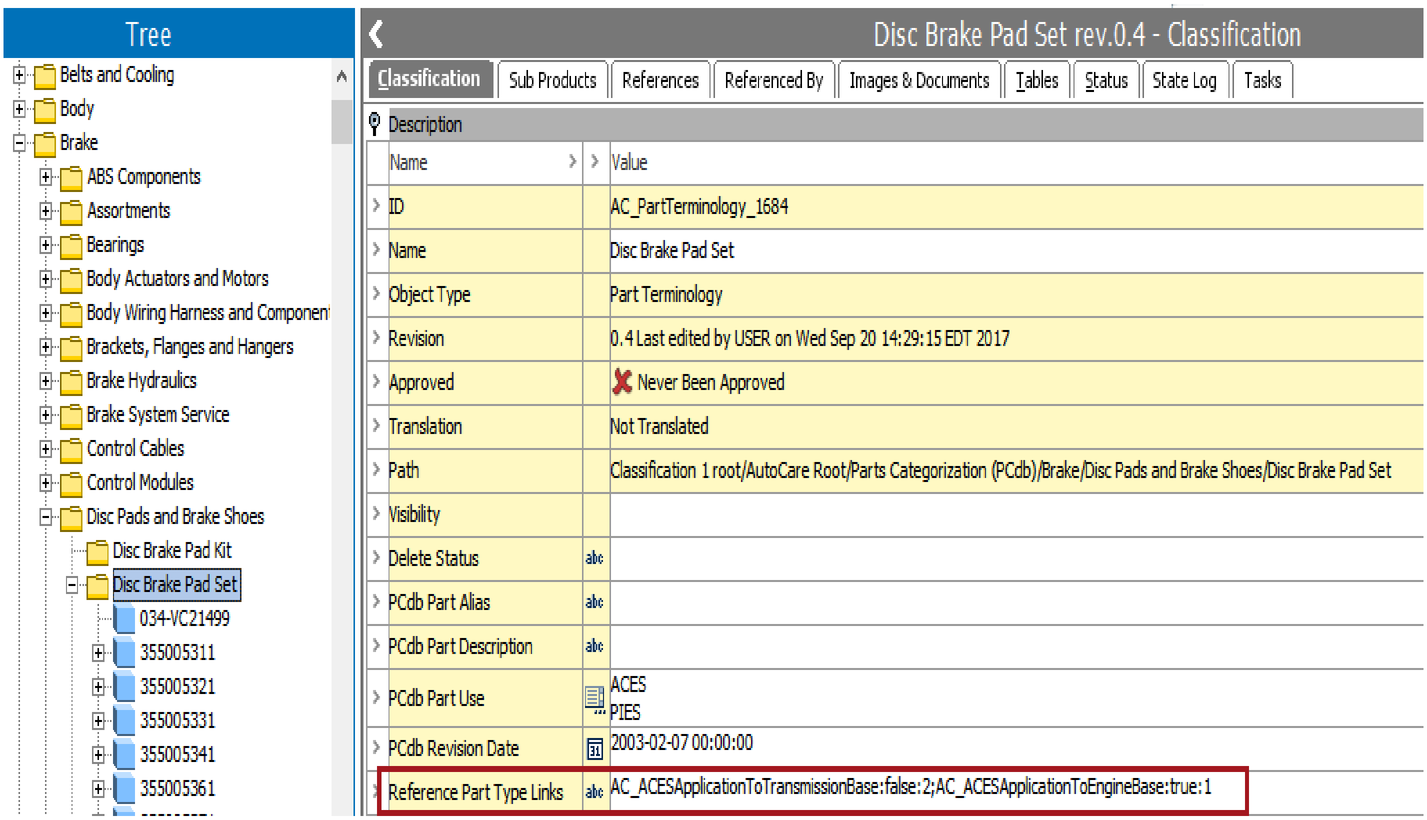
Value editor - 1 item selected

ACES Mfr Label	Premium
ACES Position	
ACES Quantity	1

The behavior of the 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 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:

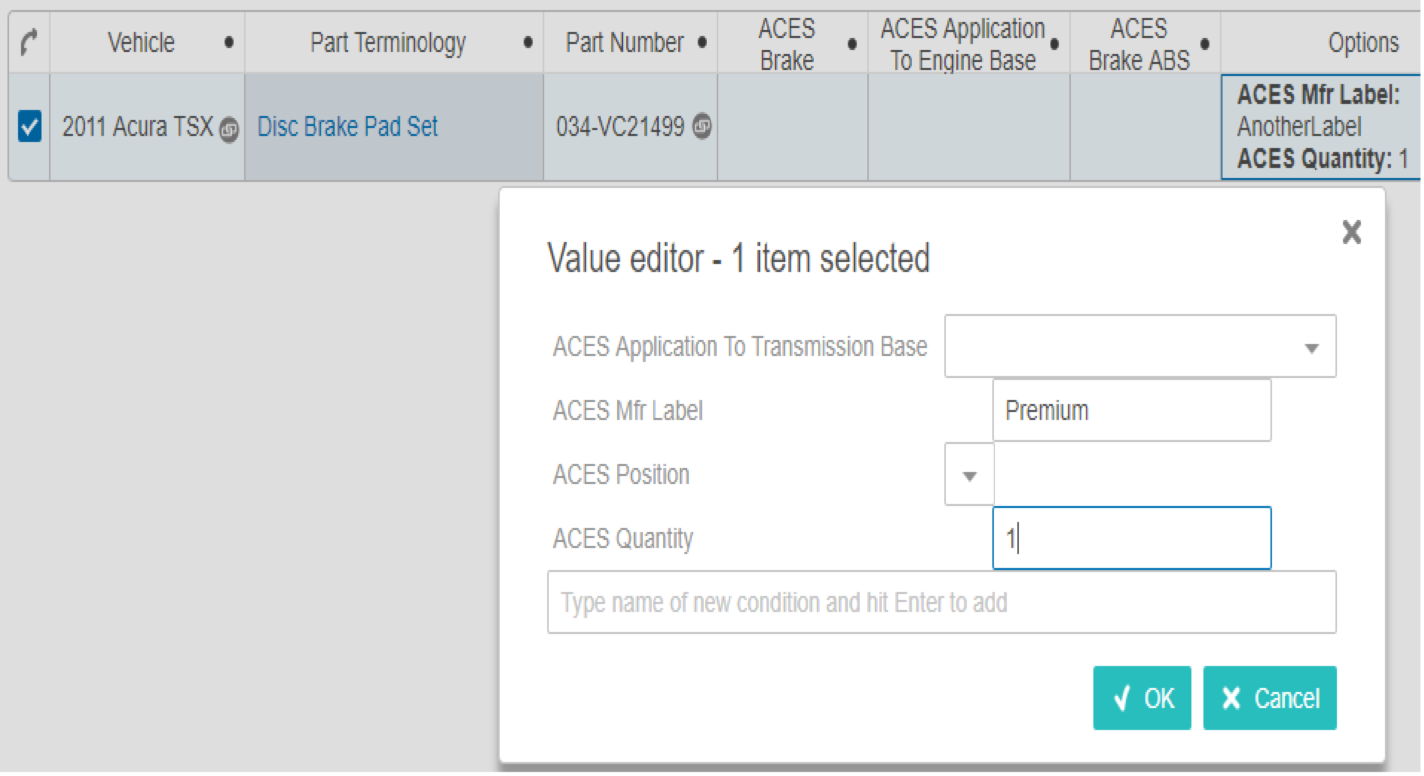


- Belts and Cooling
- Body
- Brake
  - ABS Components
  - Assortments
  - Bearings
  - Body Actuators and Motors
  - Body Wiring Harness and Componen
  - Brackets, Flanges and Hangers
  - Brake Hydraulics
  - Brake System Service
  - Control Cables
  - Control Modules
  - Disc Pads and Brake Shoes
    - Disc Brake Pad Kit
    - Disc Brake Pad Set
      - 034-VC21499
      - 355005311
      - 355005321
      - 355005331
      - 355005341
      - 355005361

Description

Name		Value
ID		AC_PartTerminology_1684
Name		Disc Brake Pad Set
Object Type		Part Terminology
Revision		0.4 Last edited by USER on Wed Sep 20 14:29:15 EDT 2017
Approved		✗ Never Been Approved
Translation		Not Translated
Path		Classification 1 root/AutoCare Root/Parts Categorization (PCdb)/Brake/Disc Pads and Brake Shoes/Disc Brake Pad Set
Visibility		
Delete Status	abc	
PCdb Part Alias	abc	
PCdb Part Description	abc	
PCdb Part Use		ACES PIES
PCdb Revision Date		2003-02-07 00:00:00
Reference Part Type Links	abc	AC_ACESApplicationToTransmissionBase:false:2;AC_ACESApplicationToEngineBase:true:1

Since the transmission reference is set to false and the engine reference is set to true, we see the engine option in its own column in the editor, and the transmission option displayed only within the value editor.



	Vehicle	Part Terminology	Part Number	ACES Brake	ACES Application To Engine Base	ACES Brake ABS	Options
☑	2011 Acura TSX	Disc Brake Pad Set	034-VC21499				**ACES Mfr Label:** AnotherLabel **ACES Quantity:** 1

Value editor - 1 item selected

ACES Application To Transmission Base	
ACES Mfr Label	Premium
ACES Position	
ACES Quantity	1

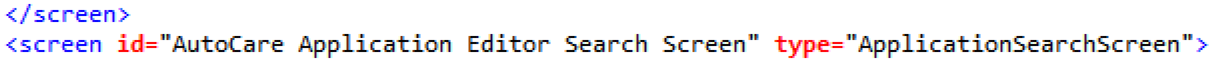
Type name of new condition and hit Enter to add

Making Application Editor Screens Read-Only

The Application Editor is designed for editing application records, and is therefore provided by easy setup with full editing capabilities. However, the editor also includes useful search features, independent of the editing functionality, and implementations may wish to take advantage of the lookup features without allowing application editing. To enable use of the lookup functionality without also enabling editing, the application editor must be configured to use the Compare Display Mode.

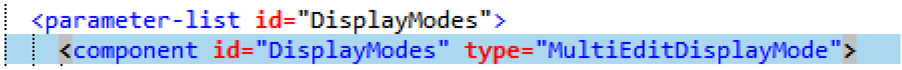
To implement a read-only application editor, follow the steps below. The modification can be done in an existing editor screen, or by duplicating an existing editor screen in the designer, then following the steps below (in the case that both a read-only and a fully editable option are desired within the same Web UI). Note that it is not recommended to make the configuration changes described below within the Web UI designer as this will require re-configuration of all existing headers.

1. Find the Web UI in the workbench, right-click, and select Edit configuration. You may choose to use an internal or external editor, but external may be preferred so that search functionality is available.
2. Find the segment with the appropriate screen ID in the format [Standard] Application Editor Search Screen, for example:



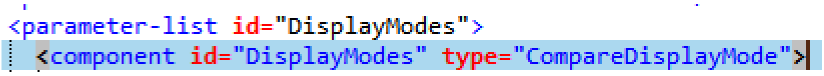
```
</screen>
<screen id="AutoCare Application Editor Search Screen" type="ApplicationSearchScreen">
```

3. Within the identified screen, find the **Display Modes** setting within the Node List settings:



```
 <parameter-list id="DisplayModes">
   <component id="DisplayModes" type="MultiEditDisplayMode">
```

4. Change the display modes type to **CompareDisplayMode**:



```
<parameter-list id="DisplayModes">
  <component id="DisplayModes" type="CompareDisplayMode">
```

5. Save the edited configuration.

Note: The '[Standard] Application Editor Screen (Parts)' and '[Standard] Application Editor Screen (Vehicles)' can both be made read-only in the same way (by updating the configuration to use the Compare Display Mode).

Missing Coverage

When using the Application Editor search feature, and both a vehicle and part type selection have been made, then missing coverage functionality is available by enabling the 'Show missing applications' checkbox.

When the 'Show missing applications' checkbox is enabled and the Application Editor Search 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 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, then the additional configurations of the vehicle are displayed. For more information about validation paths, see the **Automotive Validation Paths** topic within this guide.

In other words, when missing coverage is run, 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).

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 Submodels, each of which is available with Gas or Diesel fuel types, and each fuel type has a valid configuration with each Submodel; totaling four valid configurations of this vehicle. Further, assume that only Submodel 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 Submodel 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 Record
One record where Submodel and Fuel Type are null	None, as the existing record is understood to work for all Submodel and Fuel Type options
One record where Submodel is populated with Submodel 1 and Fuel Type is null	One record: Submodel 2 + null Fuel Type
One record where Fuel Type is populated with Gas and Submodel is null	One record: null Submodel + Diesel
One record where Submodel is populated with Submodel 1 and Fuel Type is Gas	Three records: Submodel 2 + Diesel, Submodel 1 + Diesel, Submodel 2 + Gas
Two records: Submodel 1 + null Fuel Type, Submodel 2 + Gas	One record: Submodel 2 + Diesel
No records	One record with both Submodel and Fuel Type as null

Application Manager

Application Manager allows Web UI users to search, view, create, edit, and delete part application data. Users can also export reports to Excel using the new Application Coverage Report button, and manage interchange data using different tools within the results table.

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 below.

This section addresses:

- Application Manager Overview
- Configuring Application Manager

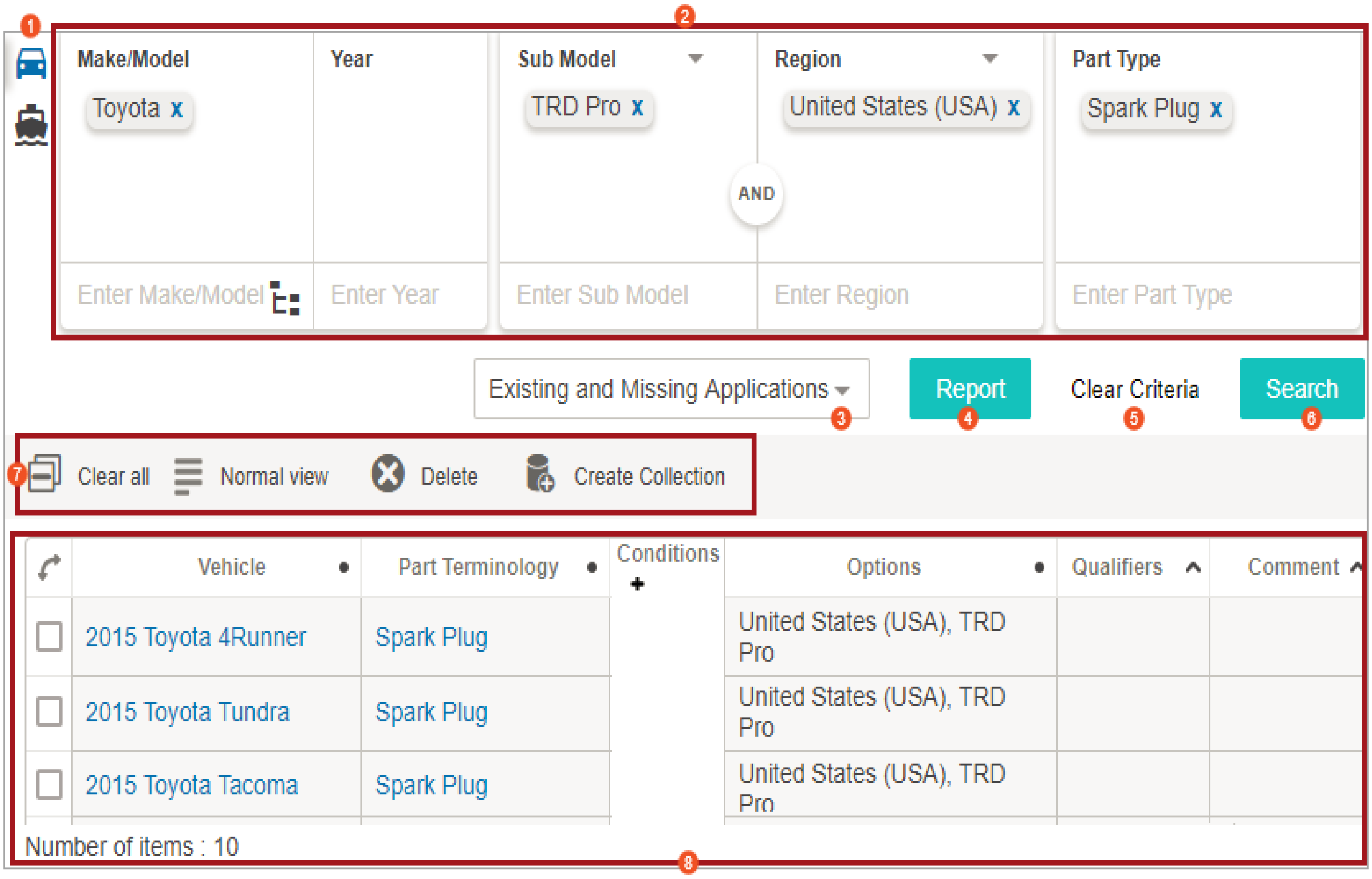
Application Manager Overview

Because the Application Manager offers many powerful features, this section will first provide an overview of the features within the screen, and then details for each of the features.

The Application Manager screen is comprised of the following:

1. Vehicle Type icon
2. Vehicle Type search panel
3. Existing and Missing Applications dropdown
4. Application Coverage Report button
5. Clear Criteria link
6. Search button
7. Results table toolbar
8. Results table

In the example below, an Application Manager is used to find existing and missing spark plug applications for Toyota's made in the United States with the TRD Pro sub model option.



Make/Model: Toyota x

Year

Sub Model: TRD Pro x

Region: United States (USA) x

Part Type: Spark Plug x

Existing and Missing Applications

	Vehicle	Part Terminology	Conditions	Options	Qualifiers	Comment
□	2015 Toyota 4Runner	Spark Plug		United States (USA), TRD Pro		
□	2015 Toyota Tundra	Spark Plug		United States (USA), TRD Pro		
□	2015 Toyota Tacoma	Spark Plug		United States (USA), TRD Pro		

Number of items : 10

1. 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. The following four icons are available for selection within the Web UI Designer: 🚗, 🚌, 🚢, 🚲. Each Application Manager must have at least one vehicle type icon. However, up to five icons can be displayed within the same Application Manager. Therefore up to five Vehicle Type search panels can be specifically configured to best meet the needs of different search types.

2. Vehicle Type search panel: The Vehicle Type search panel is highly configurable, and can consist of one to four search boxes. Because 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.

3. Existing and Missing Applications dropdown: The 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 'Show 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.

4. Application Coverage Report button: The Application Coverage Report button allows users to export an Application Coverage Report. The Application Coverage Report provides results based upon the combination of the selected search criteria and the Existing and Missing Applications dropdown selection in an Excel spreadsheet (XLSX file type). Because the Application Coverage Report is built to work with the intelligent search interface of the Application Manager, it is extremely configurable, and allows users the flexibility to choose the report criteria that best suits their needs on demand, and then easily export the search criteria results. The Report button remains disabled until at least one criteria for one of the search boxes is populated.

5. Clear Criteria link: When clicked, the Clear Criteria link allows users to easily clear the search criteria entered in all the search boxes. The Clear Criteria link remains disabled until at least one criteria for one of the search boxes is populated. When text is typed into a search box, but is not yet selected, clicking the Clear Criteria Link will also remove the text typed into the search box. However, clicking the Clear Criteria link does not clear the results table. When the results table is populated with previous search results, once the Search button is clicked to initiate a new search, the results table will only display results related to that search. Additionally, criterion can be cleared one at a time by clicking the 'x' to the right of a value within a search box. Clearing criterion does not update the search results. Once the unnecessary criterion are removed, the Report or Search buttons must be used to update the search results.

6. Search button: When clicked, the Search button will consider the criteria provided within the search boxes and the selection made for the Existing and Missing Applications dropdown, and provide any results in the results table. The Search button remains disabled until at least one criterion for one of the 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, simply enter 'Spark Plugs' for the Part Type search box criterion, and with the 'Existing Application Only' selected, click the Search button, and all the existing applications for Spark Plugs will display in the results table. However, to view missing applications for this type of broad search would be very time consuming, therefore it is recommended to use additional search criteria, or to use the Application Coverage Report button, instead of the Search button.

7. Results table toolbar: The results table toolbar consists of action buttons relative to the Application Manager results table. The action buttons displayed within the toolbar can be configured using the Web UI Designer. Different action buttons will display once one or more records are selected within the results table. For more information about action buttons, see the **Action Buttons** section of the **Web User Interfaces** guide within **STEP Online Help**.

8. Results table: The results table displays the combined results of the criteria included within the Vehicle Type search panel, and the selection made within the Existing and Missing Applications dropdown. Once the Search button has been clicked, valid results will display within the results table. However, it is possible to click the Search button and have zero results displayed. 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 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.

Configuring Application Manager

Once the Easy Setup actions for at least one standard have been run, the Application Manager Screen can be added to a Web UI and configured for that standard. If more than one standard's data will be accessed using an Application Manager, the Easy Setup actions for that standard must be run for the Application Manager Screen to function properly.

Below are the steps necessary to configure an Application Manager.

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. Login 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 the 'Application Finder Screen,' enter a screen ID, click the Add button, and the Application Finder Screen Properties dialog will display.
4. The Vehicle Types parameter is required. Populating this parameter allows for a Vehicle Type icon and Vehicle Type search panel to display within the Application Manager. Up to five Vehicle Type search panels can be added to each Application Manager. Clicking the **Add** button below the Vehicle Types parameter will display the Vehicle Type Panel Properties.

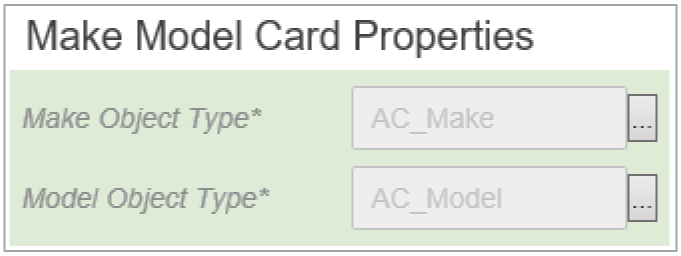
Note: If a vehicle search type is created, and the base vehicle is later removed from STEP, then the vehicle search type created within the Application Manager will no longer display. If the base vehicle is re-created, then the Application Manager will display as it did before the base vehicle was deleted.

5. Within the Vehicle Type Panel Properties dialog, the following parameters are required:

   Criterion Cards: Allows one to four search boxes to be added to the Vehicle Type Search Panel. Only one search box type is required, and each search box type can be added only once. Clicking the Add button below the Criterion Cards parameter will display an Add Component dialog with the option to add one of the following search box types: Make / Model, Options Group, Part Type, and Year. The Part Type search box type does not require additional configuration, whereas the other search box types do.

   To add the Make / Model search box type, select **Make Model Card** from the list of components, click the **Add** button, and the Make Model Card Properties will display. Once the required parameters are populated, click the **Save** button to return to the Vehicle Type Panel Properties, and optionally add another search box type, or move on to the next required parameter.

   Below is an example of how to configure the Make / Model search box type for the AutoCare standard.



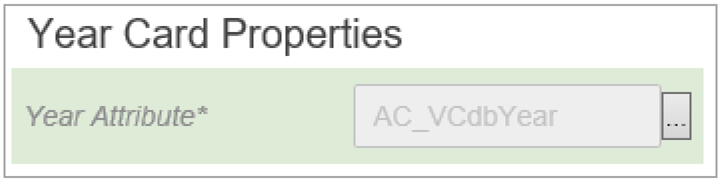
Make Model Card Properties

Make Object Type* AC_Make

Model Object Type* AC_Model

   To add the Year search box type, select **Year Card** from the list of components, click the **Add** button, and the Year Card Properties will display. Once the required parameters are populated, click the **Save** button to return to the Vehicle Type Panel Properties, and optionally add another search box type, or move on to the next required parameter.

   Below is an example of how to configure the Year search box type.



Year Card Properties

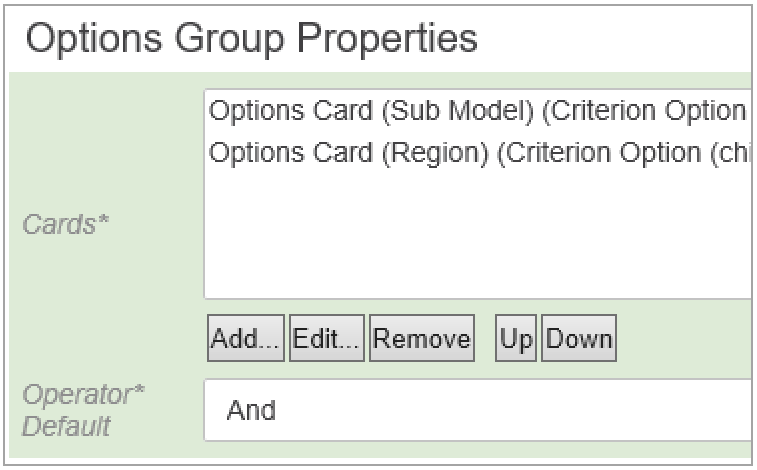
Year Attribute*: AC_VCdbYear

To add the Options Group search box type, select **Options Group** from the list of components, click the **Add** button, and the Options Group Properties will display the following required parameters:

Cards: 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.

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 how to configure the Options Group search box properties.



Options Group Properties

Cards*:
Options Card (Sub Model) (Criterion Option
Options Card (Region) (Criterion Option (chi

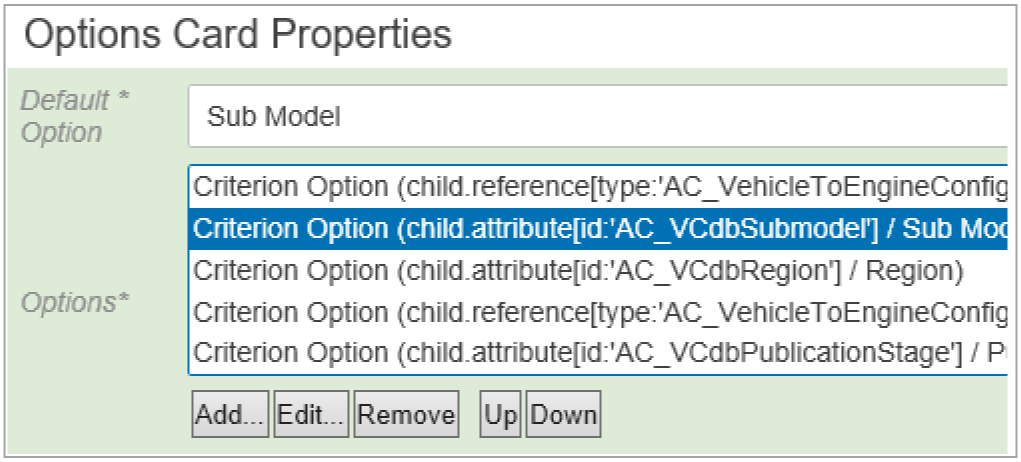
Operator* Default: And

To add an Options Card to the Options Group, click the **Add** button beneath the Cards parameter, and the Options Card Properties will display the following required parameters:

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 the Default Option parameter.

Options: Must be populated with one or more options.

Below is an example of how to configure the Options Group search box properties.



Options Card Properties

Default Option*: Sub Model

Options*:

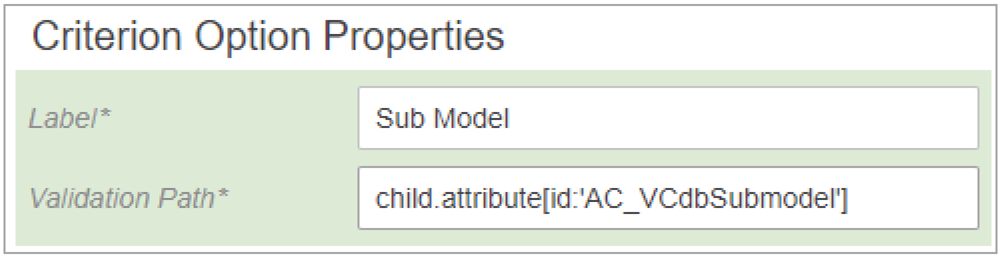
- Criterion Option (child.reference[type:'AC_VehicleToEngineConfig
- Criterion Option (child.attribute[id:'AC_VCdbSubmodel'] / Sub Moc
- Criterion Option (child.attribute[id:'AC_VCdbRegion'] / Region)
- Criterion Option (child.reference[type:'AC_VehicleToEngineConfig
- Criterion Option (child.attribute[id:'AC_VCdbPublicationStage'] / P

To add an option to an Option Card, click the **Add** button, and the Criterion Options Properties dialog will display the following required parameters:

Label: Provide a label that best describes the option being added to the Option Card / 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.

Below is an example of how to configure the Criterion Option Properties for the ACES attribute, Sub Model.

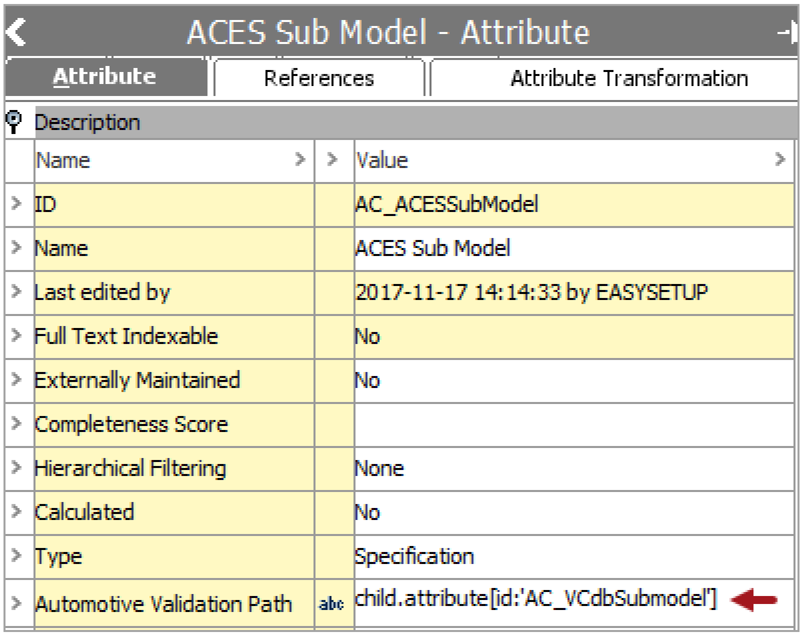


Criterion Option Properties

Label*: Sub Model

Validation Path*: child.attribute[id:'AC_VCdbSubmodel']

The Automotive Validation Path for the ACES Sub Model can be found in the workbench by navigating to System Setup > Attribute Groups > AutoCare Attributes > AutoCare Aces Attributes > ACES Sub Model.



Name		Value
ID		AC_ACESSubModel
Name		ACES Sub Model
Last edited by		2017-11-17 14:14:33 by EASYSETUP
Full Text Indexable		No
Externally Maintained		No
Completeness Score		
Hierarchical Filtering		None
Calculated		No
Type		Specification
Automotive Validation Path	abc	child.attribute[id:'AC_VCdbSubmodel']

Once the required parameters are populated, click the **Add** button to return to the Options Card Properties, and optionally add additional options for the Options Card, or click the **Add** button and return to the Options Group Properties.

Optionally repeat the steps above to add an additional Options Card, or click the **Add** button to return to the Vehicle Type Panel Properties. At a maximum, only two Options Cards can be added to a Vehicle Type Panel.

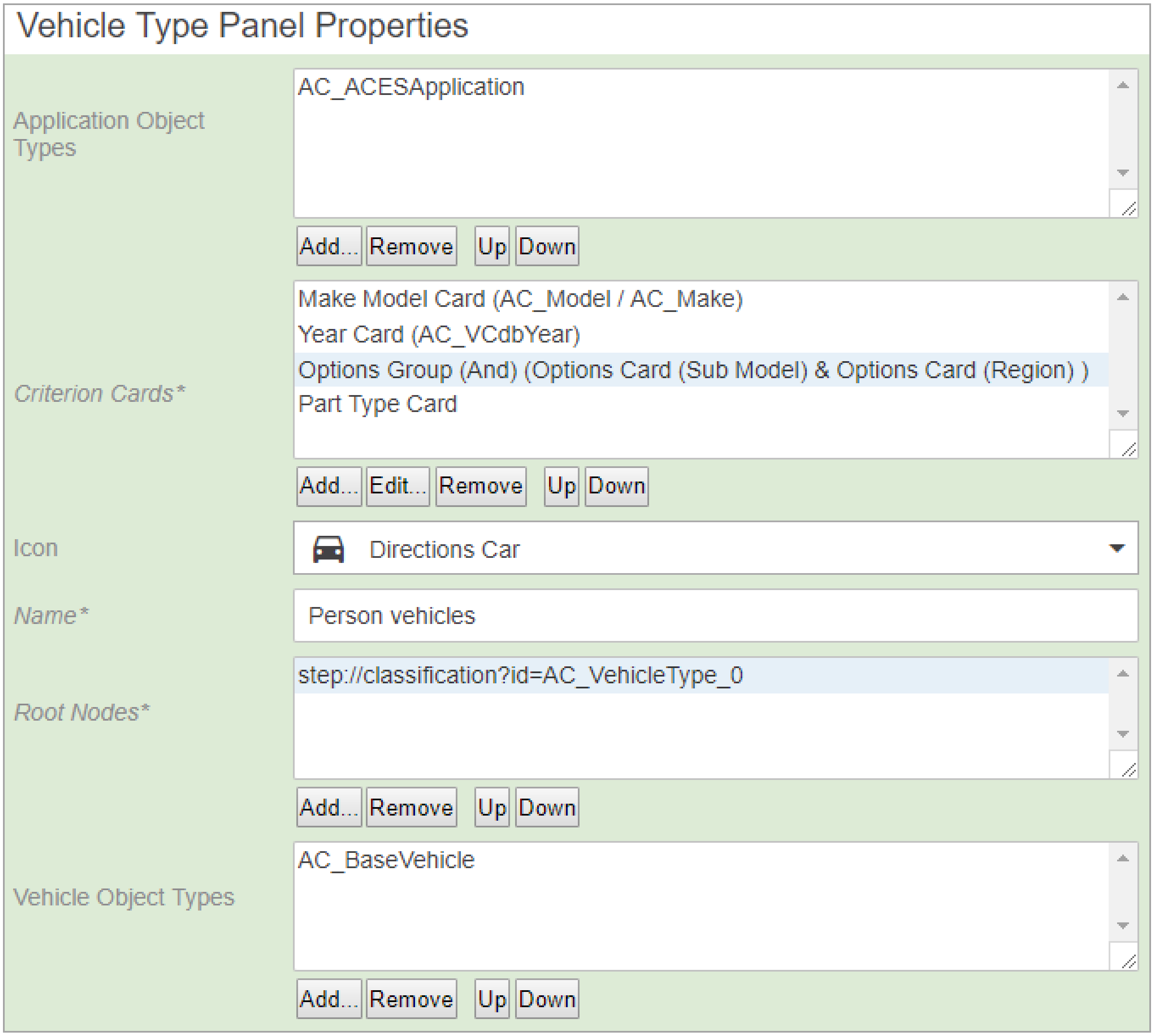
Once the required parameters are populated, click the **Add** button to return to the Vehicle Type Panel Properties, and optionally add another search box type, or move on to the next required parameter.

Icon: Allows selection of one of the four Vehicle Type icons: 🚗, 🚌, 🚢, 🏍. The Vehicle Type icon is a visual representation of the Vehicle Type search panel. Clicking a vehicle type icon displays the 'Vehicle Type search panel' configured specifically for that vehicle type (i.e., Personal Cars, Buses, Marine, Street Bikes). Hovering over the icon will display the name configured for the Vehicle Type search panel. Each Application Manager must have at least one vehicle type icon. However, up to five icons can be displayed within the same Application Manager. Therefore up to five Vehicle Type search panels can be specifically configured to best meet the needs of different search types.

Name: Allows for a custom name for the Vehicle Type search panel to display when a user hovers over the Vehicle Type icon.

Root Nodes: Allows for the restriction of root nodes to be used within the Application Manager. The root nodes listed should correspond to the Icon and Name parameters.

Below is an example of how to configure the Application Manager Vehicle Type Panel Properties for AutoCare.



Vehicle Type Panel Properties

Application Object Types: AC_ACESApplication

Criterion Cards*:
- Make Model Card (AC_Model / AC_Make)
- Year Card (AC_VCdbYear)
- Options Group (And) (Options Card (Sub Model) & Options Card (Region))
- Part Type Card

Icon: Directions Car

Name*: Person vehicles

Root Nodes*: step://classification?id=AC_VehicleType_0

Vehicle Object Types: AC_BaseVehicle

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