
pure::variants - Connector for AUTOSAR Manual

Parametric Technology GmbH

Version 7.0.0.685 for pure::variants 7.0

Copyright © 2003-2025 Parametric Technology GmbH

2025

Table of Contents

1. Introduction	1
1.1. Software Requirements	1
1.2. Installation	2
1.3. About this manual	2
2. Using the pure::variant Connector for AUTOSAR	2
2.1. Starting pure::variants	2
2.2. How the pure::variants Connector for AUTOSAR works	2
3. Tasks	2
3.1. Creating Variability Information	2
3.2. Create AUTOSAR Project Variants	9
3.3. Preview Variable Elements and Variant Results	12
3.4. Changing Variability Information	15
3.5. Checking Variability Information	16
4. Reference	16
4.1. Transformation Module Parameter	16
4.2. Transformation Modes	17
4.3. Variation Point Binding Times	17
4.4. Storing a Condition on an AUTOSAR Element	17
4.5. Storing Value Definitions to a System Constant	17
4.6. Remove Element rules	18
4.7. Project Renaming	20
5. Autosar Feature Model Exchange Format Support	21
5.1. Import models from Autosar Feature Model Exchange Format (ARXML)	21
5.2. Export Models to Autosar Feature Model Exchange Format (ARXML)	23
5.3. Transform a variant model into Autosar Feature Model Exchange Format	26
5.4. Limitations	27

1. Introduction

The pure::variants Connector for AUTOSAR enables pure::variants to flexibly create multiple differing variants from a single AUTOSAR master model. This provides a boost in development efficiency since instead of having to merge changes in slightly variations of the base project, the change is applied once to the master project and then all relevant variants are automatically generated by pure::variants.

The Connector supports AUTOSAR 3.x and AUTOSAR 4.0 arxml based models. It is tool independent and can be used with any standard compliant AUTOSAR tool chain. To simplify management of variability information in AUTOSAR it contains an optional standalone user interface for handling variability information in arxml files.

1.1. Software Requirements

The Connector for AUTOSAR is an extension for pure::variants and is available on all supported platforms.

In order to use a user interface integration the installation of the pure::variants Connector for Artop is required.

1.2. Installation

Please consult section **pure::variants Connectors** in the **pure::variants Setup Guide** for detailed information on how to install the connector (menu **Help** -> **Help Contents** and then **pure::variants Setup Guide** -> **pure::variants Connectors**).

1.3. About this manual

The reader is expected to have basic knowledge about and experiences with pure::variants. This document is available in online help as well as in printable PDF format [here](#).

2. Using the pure::variant Connector for AUTOSAR

2.1. Starting pure::variants

Depending on the installation method used either start the pure::variants-enabled Eclipse or under Windows select the **pure::variants** item from the **program** menu.

If the **Variant Management** perspective is not already activated, do so by selecting it from **Open Perspective** -> **Other...** in the **Window** menu.

2.2. How the pure::variants Connector for AUTOSAR works

The connector for AUTOSAR works for AUTOSAR 3 and AUTOSAR 4 projects. It detects the version automatically and performs the required actions.

The Connector use a AUTOSAR runtime master model to derive multiple variants from it. Each of the variants is derived from the master runtime model by removing AUTOSAR elements not relevant in the respective variant. Removal of elements is controlled by pure::variants restriction conditions. Conditions can be added to AUTOSAR elements by the master model author(s). The rules are evaluated by pure::variants based on a variant model which captures the variant specific feature selection. Elements having restrictions conditions which evaluate to 'false' are removed including dependent elements (see [the section called "Remove Elements"](#)).

Removal is either executed by pure::variants directly by creating new AUTOSAR arxml files ('**Remove Elements**' mode) or by generating AUTOSAR 4 compliant variant and variability information ('**Generate Variation Point**' mode, only for AUTOSAR 4 files).

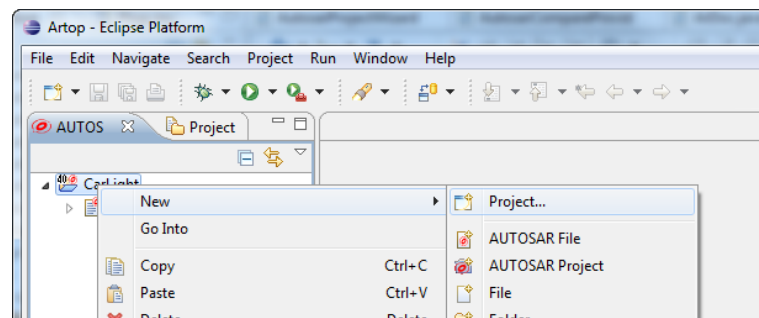
3. Tasks

3.1. Creating Variability Information

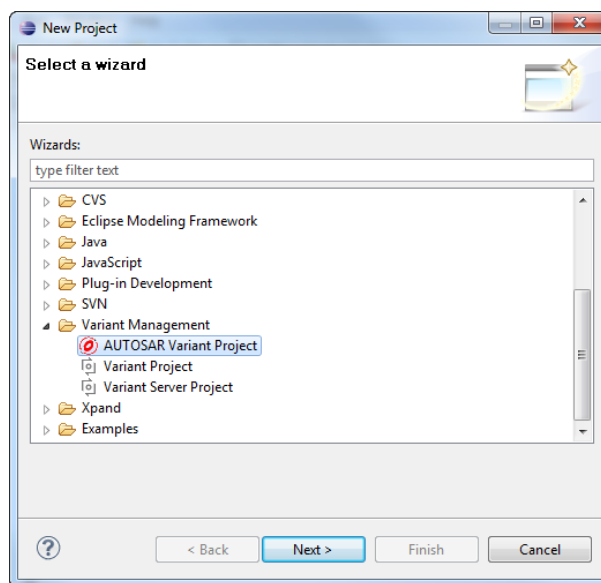
Create AUTOSAR Variant Project

In order to create AUTOSAR project variants it is required to setup an associated pure::variants project. This project holds the feature model to describe variability information used in the AUTOSAR project. It also contains a family model which refers to all AUTOSAR files which should be part of the variant generation process. A configuration space holds the variant description models which defines the different variants.

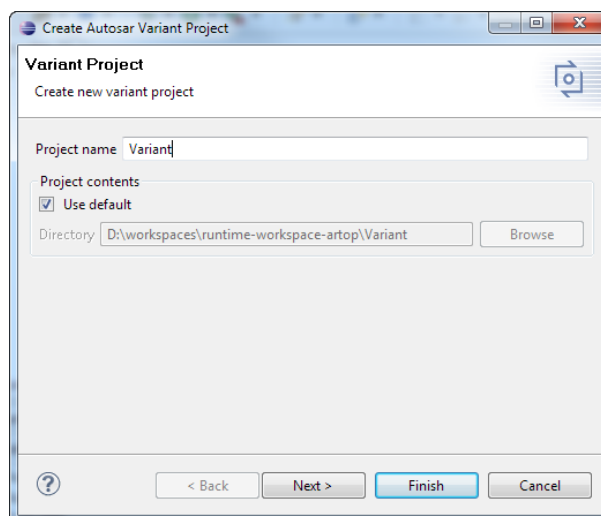
To create a new AUTOSAR variant project for an existing AUTOSAR project select the project and start the New Project Wizard by choosing **New** -> **Project...** in the context menu of the project explorer.

Figure 1. Start New Project Wizard

On the next page select **AUTOSAR Variant Project** and press **Next**.

Figure 2. New AUTOSAR Variant Project

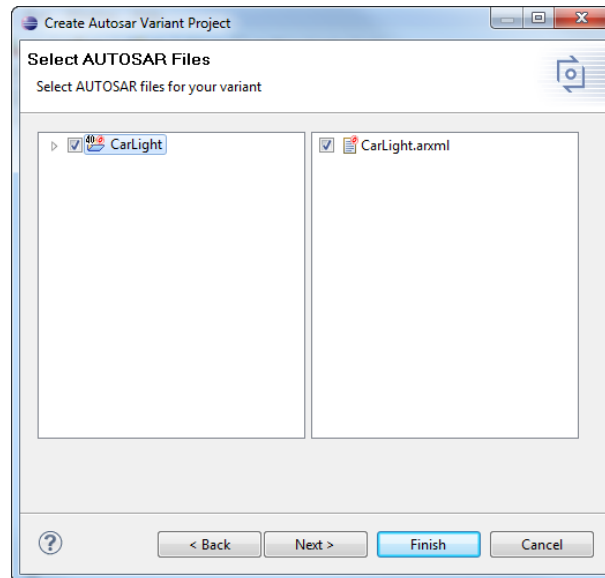
Enter the name for the variant project. For all other options the defaults can be used. Press **Next**.

Figure 3. Enter name for AUTOSAR Variant Project

The next page shows all projects containing AUTOSAR project files. Please check the project and the files, which should be used for the variant creation. If a project was already selected in the project explorer the same project

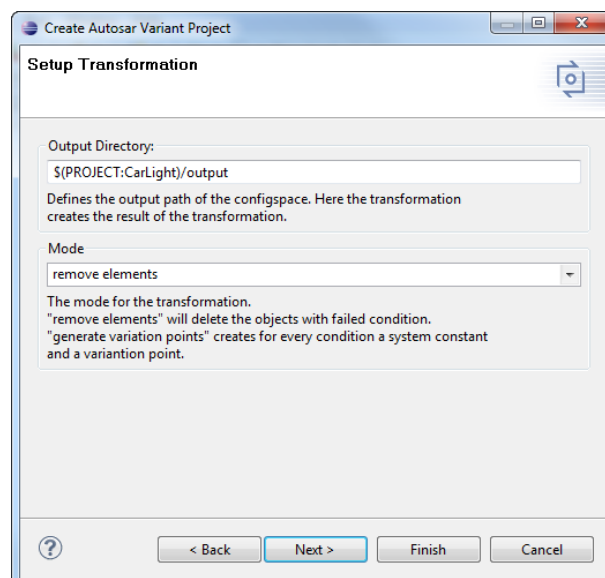
and all its AUTOSAR files are already checked. You can check as many files as you like. The files can also stay in different projects. Press **Next** after all needed files are checked.

Figure 4. Select AUTOSAR files

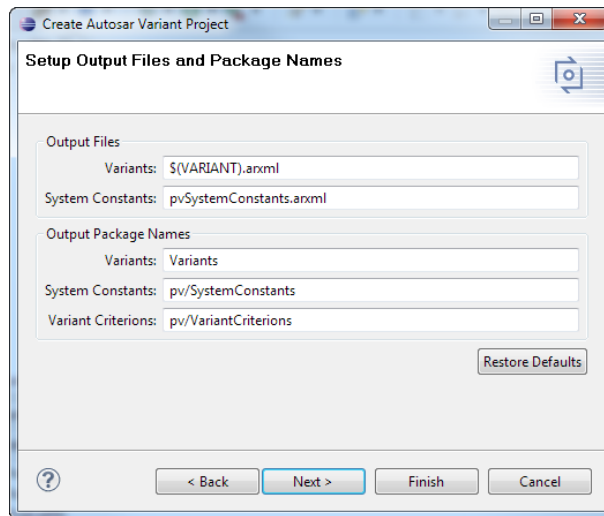


The next page shows the options for the transformation, which is used to generate the AUTOSAR project variants. The first option is the **Output Directory** enter here the folder in which the files for a variant should be saved. The **Mode** option defines how the variants will be created. The "remove elements" mode will delete all elements not part of the variant from the files. This can be used for AUTOSAR 3x and AUTOSAR 4 projects. The "generate variation points" mode will create system constants, variation points and system constant values to allow AUTOSAR 4 compatible variant definition. This can only used for AUTOSAR 4 projects.

Figure 5. Setup Transformation Options

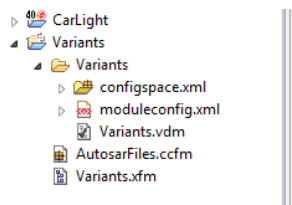


The next page shows further options for the transformation. The transformation process generates a `variants` and `System Constants` file. Therefore file names can be changed. The `*.arxml` extension has to be used. Furthermore their package names for `variants`, `System Constants` and `Variant Criteria` can be changed. The names needs to comply to the AUTOSAR package name specification.

Figure 6. Setup Transformation Options (Second Page)**Note**

The default values for output file names and output package names can simply be restored by pushing the **Restore Defaults** button.

After pressing **Finish** a new pure::variants project will be created. The created project should look like the following figure.

Figure 7. Created AUTOSAR Variant Project

To finish the project setup open the feature model and create your features. These features are later used in the conditions to describe when a specific AUTOSAR element is part of a variant.

Note

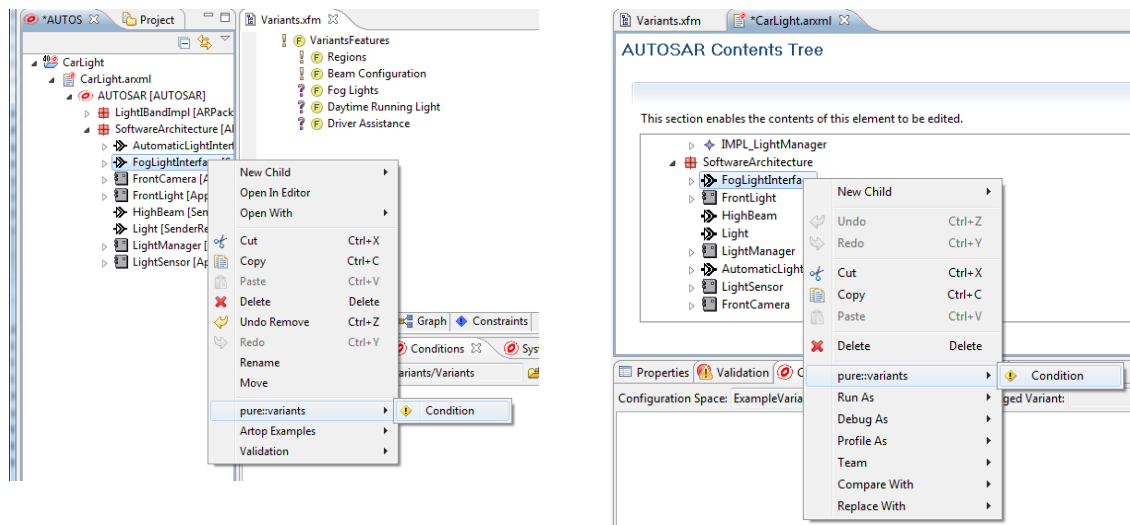
The created project is only visible in the **Project Explorer** or the **Variants Projects** views.

Create Condition on AUTOSAR Element

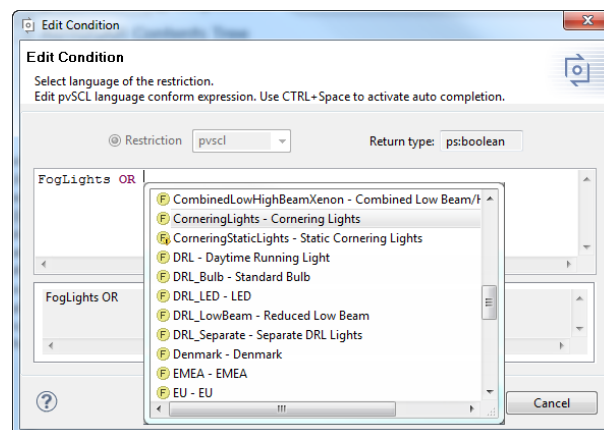
To decide if an AUTOSAR element is part of a variant a condition can be stored on any element which is able to hold ADMIN-DATA.

Using Context Menu in any Artop View/Editor

Conditions can easily be added by using the **pure::variants** -> **Condition** context menu. It is available on any AUTOSAR element which can hold a condition in any editor or view.

Figure 8. Create Condition by Context Menu

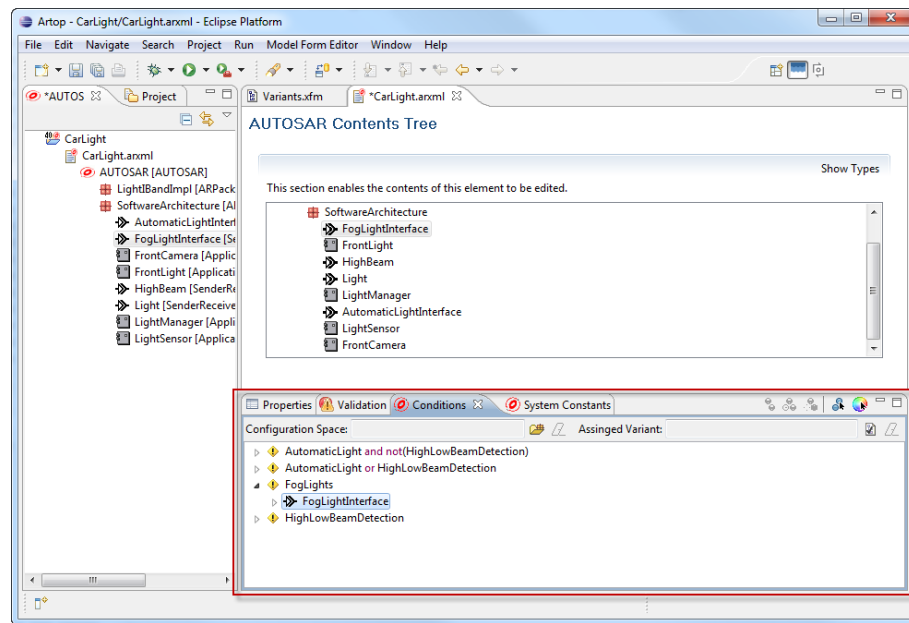
The menu opens the condition editor. If the AUTOSAR object has already a condition assigned the editor shows the condition code. The editor supports writing conditions by a proposal showing all available features from the feature model. To open the proposal press **Ctrl+Space**.

Figure 9. Condition Editor Showing the Proposal

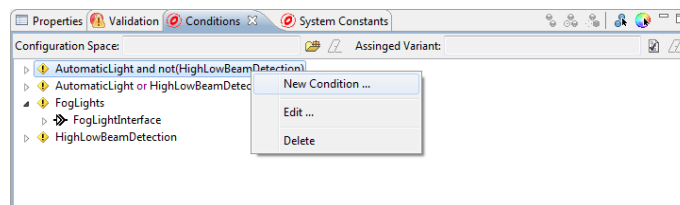
Pressing **OK** will store the new or changed condition into the AUTOSAR object.

Using the Conditions View

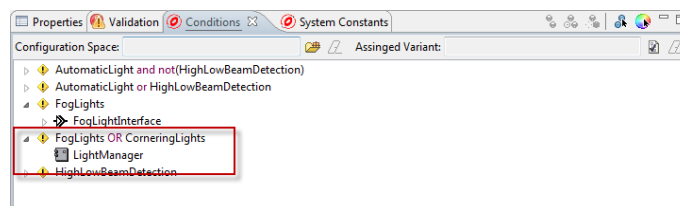
The **Conditions View** help to work with conditions inside AUTOSAR projects. It shows all conditions found in the files. Below the conditions all AUTOSAR elements which are restricted by this condition are shown.

Figure 10. Conditions View

The view can also be used to create conditions and assign conditions to elements. To create a new condition use the context menu **New Condition ...** of the view.

Figure 11. New Condition

The condition editor comes up and allows to create a new condition. After pressing **OK** the newly created condition appears in the view. You can now drop AUTOSAR elements from any view or editor on that condition. The condition is then set to the dropped element.

Figure 12. New Condition - Element dropped

Using XML Editor

Conditions are written in pvSCL (pure::variants Simple Constraint Language, see pure::variants User's Guide) and stored into the ADMIN-DATA of an element. An example is shown in the following figure.

```
<ADMIN-DATA>
  <SDSG>
    <SDG GID="VARIATION-POINT">
      <SD GID="SW-SYSCOND">FogLight OR CorneringLights</SD>
    </SDG>
  </SDSG>
```

</ADMIN-DATA>

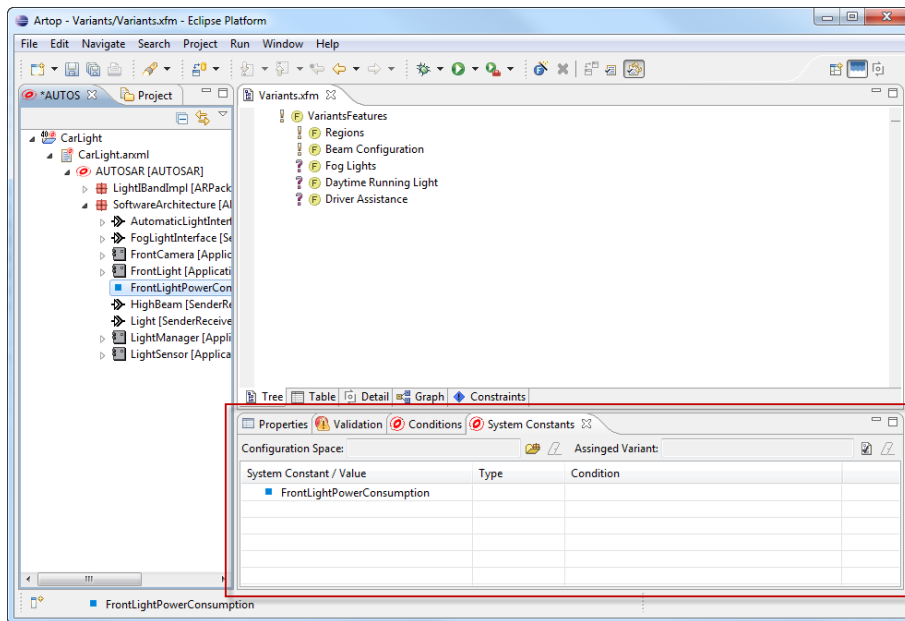
You can use any XML editor to create this data inside the AUTOSAR file.

Create Value Definition on System Constant

Using System Constants View

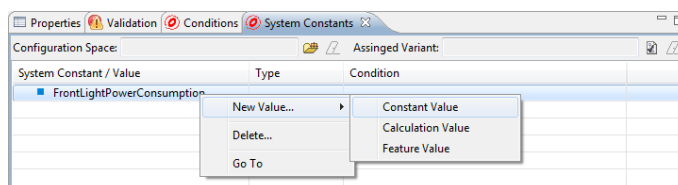
The **System Constants View** provides user interface for defining and changing variant depending value definitions for system constants. It shows all existing system constants and their value definitions.

Figure 13. System Constants View



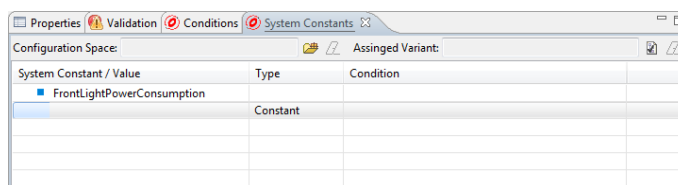
To add a value definition to a system constant select it in the view and open the context menu. Choose **New Value ... -> Constant Value** from the menu.

Figure 14. New System Constant Value



A new value row is added to the table below the selected system constant.

Figure 15. New System Constant Value Row Added



Double click the first column of the row and enter your value. If the value should only appear if a certain condition is valid you need to add the condition into the last column. Double click the condition column and open the

Condition Editor by pressing the small button on the right side. The condition editor opens. Enter the condition and press **OK**.

Figure 16. New System Constant Value with Condition Added

System Constant / Value	Type	Condition
FrontLightPowerConsumption 70	Constant	LowBeamXenon and HighBeamXenon

You can add as many values as you like. Later during variant creation the values will be one after the other evaluated. The first value which condition is valid or which has no condition will be used for the variant. This allows to have always a default value by leave the condition empty on the last value.

Figure 17. New System Constant Value with Default

System Constant / Value	Type	Condition
FrontLightPowerConsumption 70	Constant	LowBeamXenon and HighBeamXenon
110	Constant	LowBeamHalogen and HighBeamHalogen
80	Constant	

The order of the values can be easily changed by drag and drop the values.

Using XML Editor

Value definition can be written with any XML editor. The listing below shows the structure of the example from the previous section.

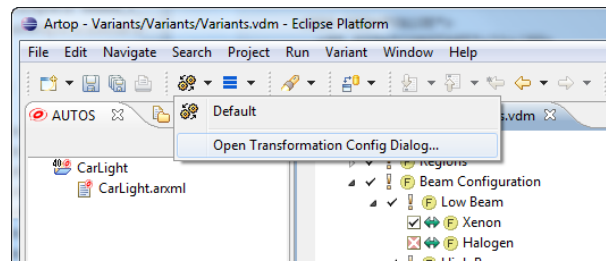
```
<ADMIN-DATA>
  <SDGS>
    <SDG GID="VALUE-DEFINITIONS">
      <SDG GID="VALUE">
        <SD GID="CONSTANT">70</SD>
        <SD GID="SW-SYSCOND">LowBeamXenon and HighBeamXenon</SD>
      </SDG>
      <SDG GID="VALUE">
        <SD GID="CONSTANT">110</SD>
        <SD GID="SW-SYSCOND">LowBeamHalogen and HighBeamHalogen</SD>
      </SDG>
      <SDG GID="VALUE">
        <SD GID="CONSTANT">80</SD>
      </SDG>
    </SDGS>
  </ADMIN-DATA>
```

3.2. Create AUTOSAR Project Variants

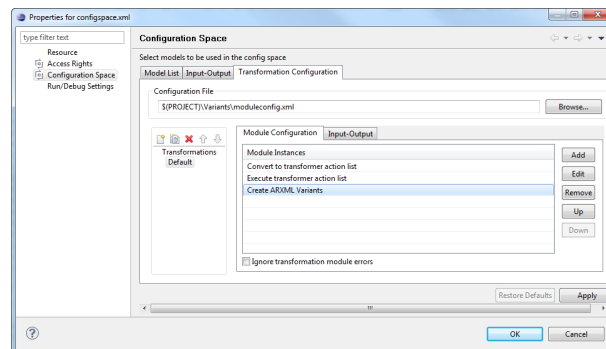
Variants of AUTOSAR project will be created by opening the variant description model and create a valid feature configuration. A variant is then generated by performing a transformation of the variant.

Remove Elements

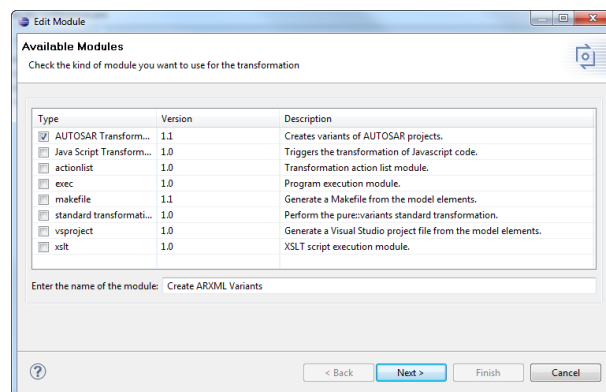
To remove elements from the AUTOSAR file please ensure that the module configuration contains the **AUTOSAR Transformation Module** and the "remove elements" mode is activated. Open the transformation configuration by expanding the **Transform** button and select **Open Transformation Config Dialog ...**

Figure 18. Open Transformation Configuration Dialog

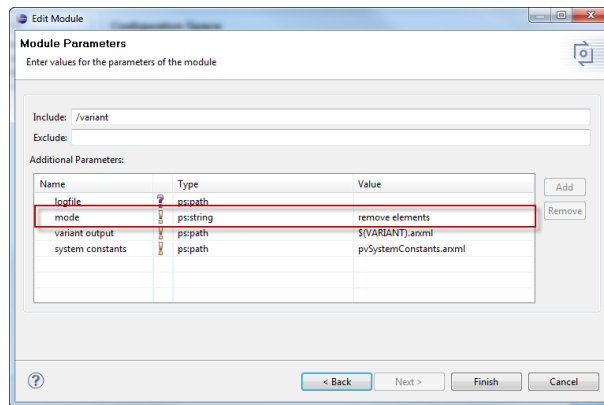
Double click the **Create ARXML Variants** in the module configuration list.

Figure 19. Select AUTOSAR Transformation Module

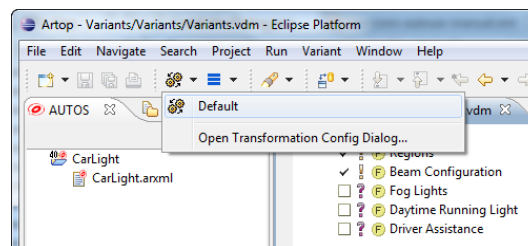
Leave the **AUTOSAR Transformation Module** checked and press **Next**.

Figure 20. Select AUTOSAR Transformation Module

Ensure that the option mode is set to "remove elements" and press **Finish**.

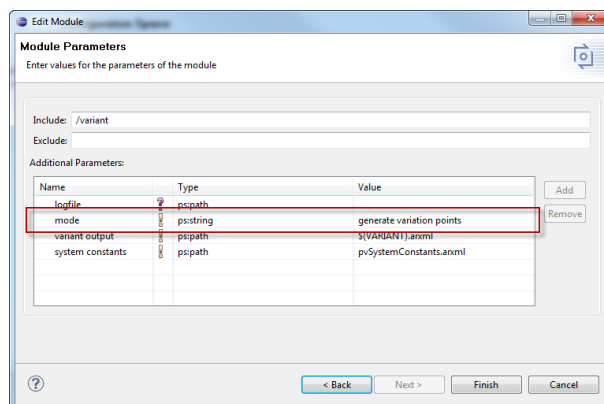
Figure 21. Setup AUTOSAR Transformation Options

To start the transformation choose the configuration (standard named Default) from the drop-down menu of the transformation button. The generated variant of the AUTOSAR project can be found in the output directory after the transformation finished.

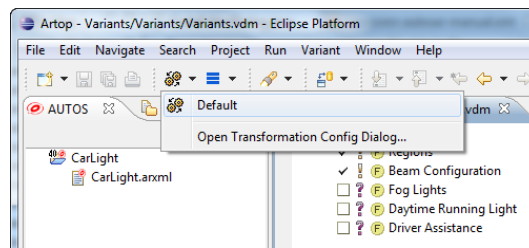
Figure 22. Start Transformation

Create Variation Points and Predefined Variant (AUTOSAR 4 only)

An AUTOSAR 4 compatible variant generation can be performed by using the "generate variation points" mode. For this the Mode option in the **AUTOSAR Transformation Module** need to be set to "generate variation points". Open the transformation module options page as described in the section called "Remove Elements".

Figure 23. Setup AUTOSAR Transformation Options

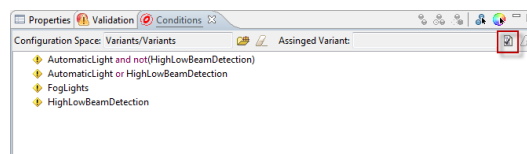
To start the transformation choose the configuration (standard named Default) from the drop-down menu of the transformation button. The generated variant of the AUTOSAR project can be found in the output directory after the transformation finished.

Figure 24. Start Transformation

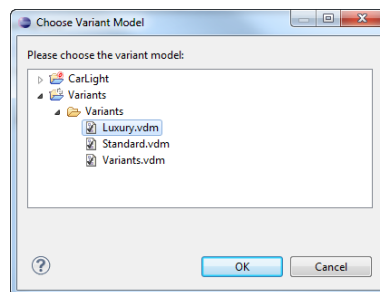
3.3. Preview Variable Elements and Variant Results

Select Variant Description Model for Preview a Variant

To enable the preview of variants a variant description model needs to be selected. Open one of the Artop based views (e.g. Conditions View) and press the **Assign Variant** button.

Figure 25. Assign Variant Description Model

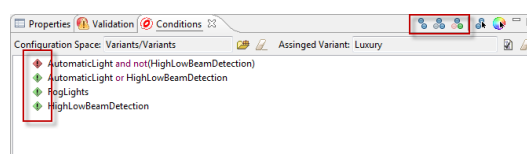
This opens a file selection dialog. Please select the model and press **OK**.

Figure 26. Select Variant Description Model

The variant description model is loaded and evaluated. The **Conditions View** changes the images in front of the conditions. Before a variant is assigned the images are yellow. After loading a variant the images are:

- green if the condition evaluates to `true`
- red if the condition evaluates to `false`

The different preview modes are also enabled.

Figure 27. Conditions View with Assigned Variant

The **System Constants View** is updated to show the values for the variant behind the name of the system constant. The images of the conditions are also changed to reflect the result of the variant.

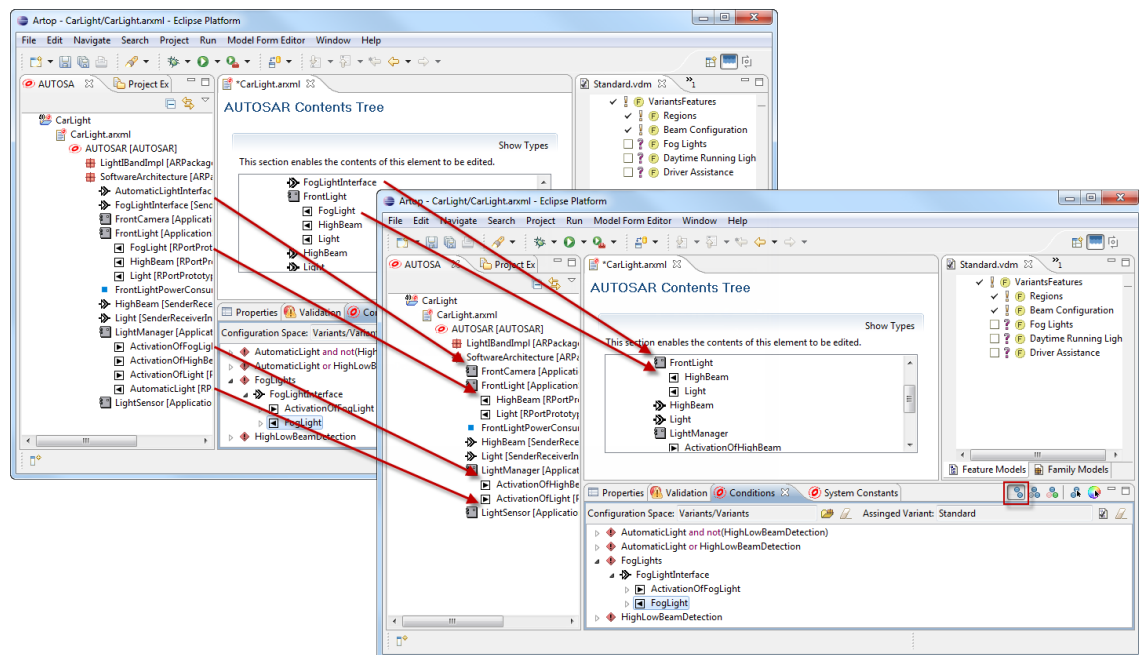
Figure 28. System Constants View with Assigned Variant

System Constant / Value	Type	Condition
FrontLightPowerConsumption = 70	Constant	LowBeam/Xenon and HighBeam/Xenon
110	Constant	LowBeam/Halogen and HighBeam/Halogen
80	Constant	

The variant description model can be assigned in the **Conditions View** as well as in the **System Constants View**. It will change the variant for all views. All views get updated if the selection in the assigned variant description model is changed.

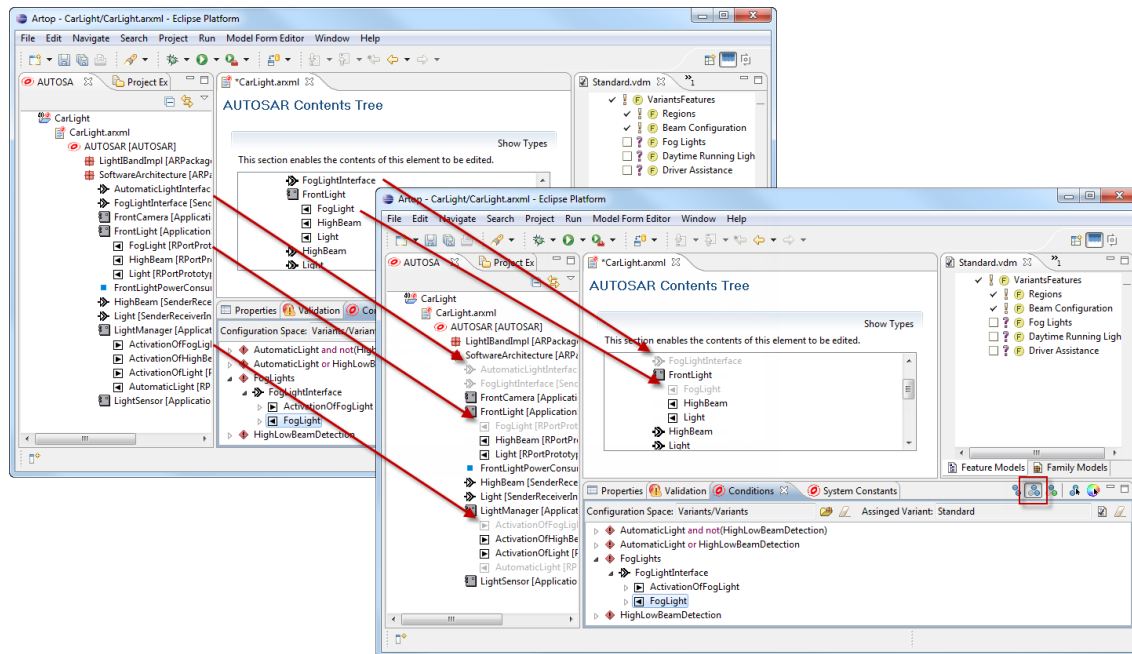
Filter Elements to Remove

After activating the **Filter Variant** mode in the **Conditions View** all views and editors showing the AUTOSAR system are updated to hide elements which are not part of the variant. If the feature selection in the assigned variant is changed the views and editors are updated immediately to reflect the new variant.

Figure 29. Preview Variant: Filtered Elements

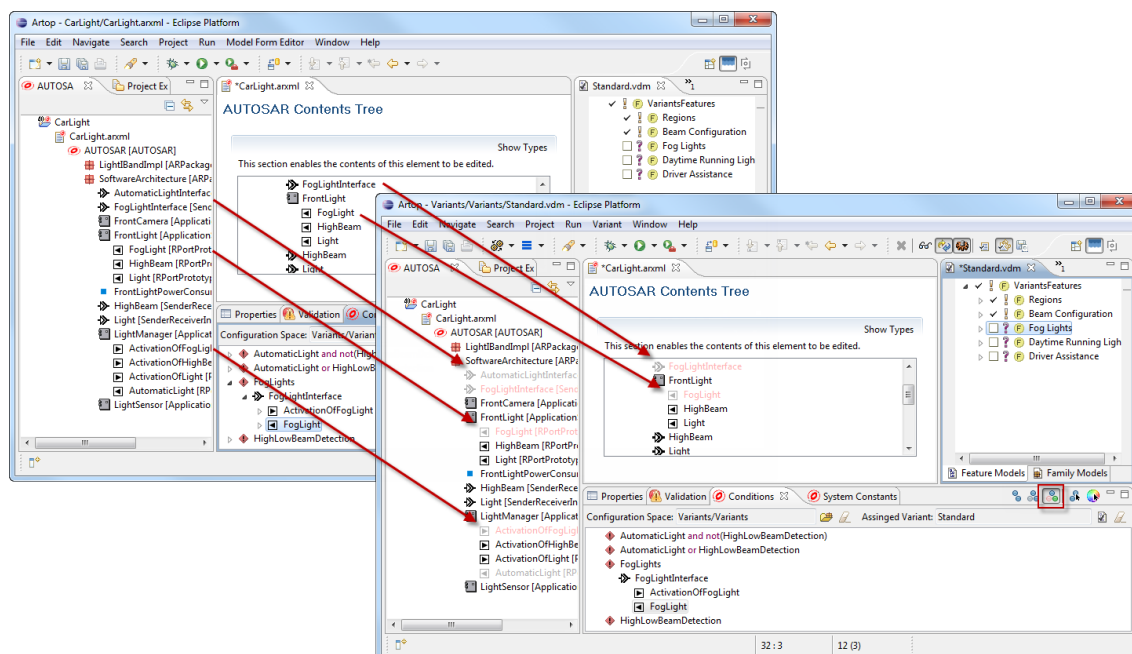
Gray out Elements to Remove

After activating the **Show Variant** mode in the **Conditions View** all views and editors showing the AUTOSAR system are updated to gray out elements which are not part of the variant. If the feature selection in the assigned variant is changed the views and editors are updated immediately to reflect the new variant.

Figure 30. Preview Variant: Gray Out Elements

Show Variant Delta

After activating the **Show Variant Delta** mode in the **Conditions View** all views and editors showing the AUTOSAR system are updated to lighten elements which are not part of the variant. If the selection is changed the views and editors are updated immediately to reflect the new variant. The color of elements which are previously part of the variant but are now removed is changed to light red. The color of elements which are new part of the variant is changed to green. This allows to see the impact of a selection change.

Figure 31. Preview Variant: Variant Delta

Navigate to AUTOSAR element

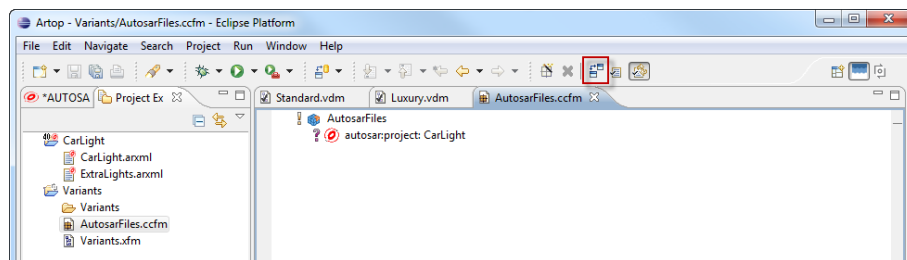
You can navigate to any AUTOSAR element shown in the **Conditions** or **System Constants View** by double click on the element or use the context menu **Show Object** item. All views and editor showing the AUTOSAR system will show and select this element.

3.4. Changing Variability Information

Add new AUTOSAR File

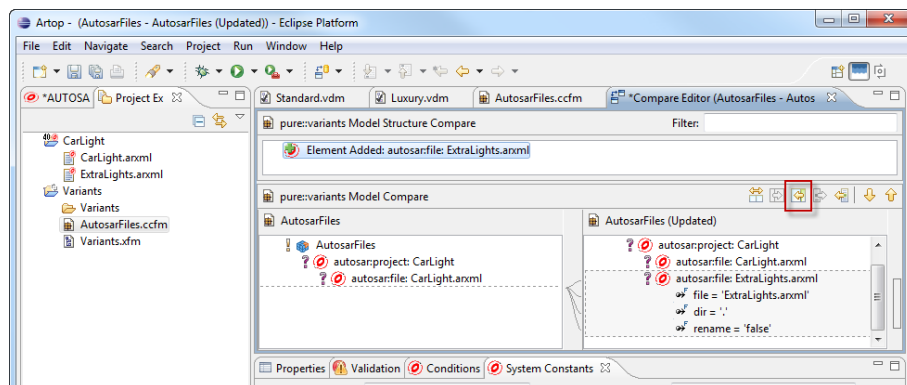
To add a new AUTOSAR file to the project open the `AutosarFiles.ccfm` model and press the **Synchronize** button in the tool bar.

Figure 32. Synchronize new AUTOSAR files



The synchronization process scans the AUTOSAR project for changes and opens the **Merge Editor**. The editor shows all changes in the project. The model is updated by merging the changes. Press the "**Copy all non-Confliction Changes**" button to add the new file to the project and the save editor content. For more information how to work with the **Merge Editor** see pure::variants Users Guide section 6.6 *Comparing Models*.

Figure 33. Add new File to Project



Change a Condition for a single Element

Change the condition on a single element by selecting it in any AUTOSAR view or editor and open the context menu. By choosing the **pure::variants** -> **Condition** action the condition editor comes up and allows to edit the condition. After pressing **OK** the newly created condition is stored and comes up in the **Conditions View**. The element is relocated to the new condition.

Note

Conditions with no assigned elements will be removed after closing your Artop instance.

Change a Condition for all Elements

Select the condition in the **Conditions View** and open the **Edit ...** context menu. The condition editor with the actual condition comes up. Edit the condition and press **OK**. The condition will be updated on all elements.

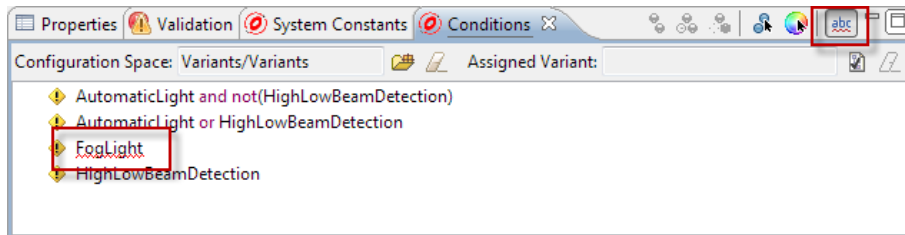
3.5. Checking Variability Information

In both views is a **Check Errors** button exists. It allows to scan for any violations affecting the syntax or semantic of pvSCL expressions. The button effects all views. So enabling the check in **Conditions View** will also enable the check for **System Constants View**.

Check Errors in the Conditions View

Check all conditions present in the **Condition View** by enabling the **Check Errors** button. If an error appears the affected condition is marked. Errors are caused either by missing feature(s) or wrong pvSCL syntax.

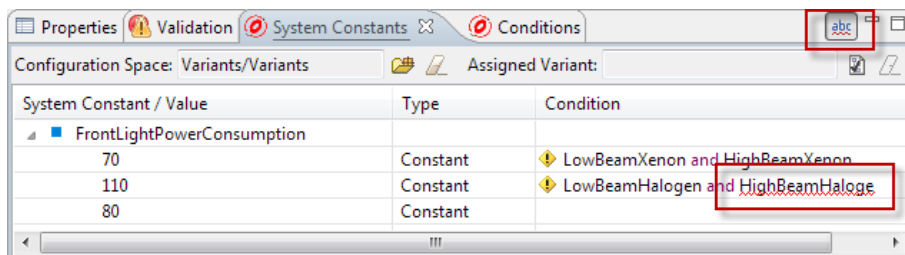
Figure 34. Check Conditions for Errors



Check Errors in the System Constants View

Check all Values of all System Constants present in the **System Constants View** having a condition. These will be checked in the same way as described in the section called “Check Errors in the Conditions View”. Additionally values of type **Feature** or **Calculation** are checked, too.

Figure 35. Check Values of System Constants for Errors



4. Reference

4.1. Transformation Module Parameter

Parameter	Meaning
mode	The mode for the transformation. For the possible modes see Section 4.2, “Transformation Modes”
variant output	The output file for all variants data. The file contains all calculated system constant values. It also contains the predefined variant (default "\${VARIANT}.arxml")
variant package	The package for the variant data. (default "Variants")
system constants	The file for all by pure::variants generated system constants. System constants will be created only if transformation parameter "mode" is set to "generate variation points". (default "pvSystemConstants.arxml")
system constants package	The package for the generated system constants. (default "pv/SystemConstants")
variant criteria package	The package for the generated variant criteria. (default "pv/VariantCriteria")
condition constant prefix	The prefix for all generated System Constants of Conditions. (default "PVC_")






Parameter	Meaning
dependency constant prefix	The prefix for all generated System Constants resulting from dependency chain analyses. (default "PVR_")
logfile	Optional name of the logfile to record remove operations

4.2. Transformation Modes

Mode	Meaning
Remove Elements	The remove mode simply removes all elements not part of the variant from the AUTOSAR document. This mode works with AUTOSAR 3.x and AUOSAR 4.
Generate Variation Points	This mode creates AUTOSAR 4 compatible variant definitions. It generates a <code>System Constant</code> for every element with a condition. This <code>System Constant</code> is used in a <code>Variation Point</code> definition on that element. Also for connected elements a <code>System Constant</code> is created and referred in the attached <code>Variation Point</code> definition. For a variant a <code>System Constant Value Set</code> for the generated <code>System Constants</code> will be created. Depending on the presents of the element in this variant the value is set to 0 or 1. This <code>System Constant Value Set</code> is then referred by the created <code>Predefined Variant</code> .

4.3. Variation Point Binding Times

When using the "Generate Variation Points" mode the binding times of the variation points will be calculated from feature binding times. Hence the binding time of a feature is represented by the features type. See table below for a list of all supported binding times. The binding time of the variation point is set to the latest binding time of all features referenced by the condition.

Image	Feature Type	AUTOSAR Binding Time
	autosar:SystemDesign	SYSTEM-DESIGN-TIME
	autosar:CodeGeneration	CODE-GENERATION-TIME
	autosar:PreCompile	PRE-COMPILE-TIME
	autosar:Link	LINK-TIME
	autosar:PostBuild	POST-BUILD

4.4. Storing a Condition on an AUTOSAR Element

Condition rules are written in pvSCL (pure::variants Simple Constraint Language, see pure::variants User's Guide) and stored in the `ADMIN-DATA` of an element. An example is shown in the following figure.

```
<ADMIN-DATA>
  <SDSG>
    <SDG GID="VARIATION-POINT">
      <SD GID="SW-SYSCOND">FeatureA OR FeatureB</SD>
    </SDG>
  </SDSG>
</ADMIN-DATA>
```

4.5. Storing Value Definitions to a System Constant

The definition for value calculation is stored into the `ADMIN-DATA` of the system constant. Every defined value can have a condition attached. During the evaluation the first value which condition is true will be used for the variant. A value is always used if it has no condition. This behavior allows to define the default value as last value without a condition.

There are three different value types possible.

Type	Meaning	Example
Constant	a simple string or number	ON, OFF, 5, 2277
Calculation	a attribute of a feature	FeatureA->attribute
Feature	direct mapping from feature to boolean value. Results in 1 if feature is selected. 0 in any other case.	FeatureA

An example is shown in the following figure.

```

<ADMIN-DATA>
  <SDGS>
    <SDG GID="VALUE-DEFINITIONS">
      <SDG GID="VALUE">
        <SD GID="CONSTANT">10</SD>
        <SD GID="SW-SYSCOND">FeatureA OR FeatureB</SD>
      </SDG>
      <SDG GID="VALUE">
        <SD GID="CALCULATION">FeatureWithAttribute->Attribute</SD>
        <SD GID="SW-SYSCOND">FeatureWithAttribute</SD>
      </SDG>
      <SDG GID="VALUE">
        <SD GID="CONSTANT">0</SD>
      </SDG>
    </SDGS>
  </ADMIN-DATA>

```

4.6. Remove Element rules

The following table shows which related elements are removed when a special elements is removed:

AUTOSAR 3 Element to remove	Related AUTOSAR 3 Elements to remove
R-PORT-PROTOTYPE P-PORT-PROTOTYPE	OPERATION-INVOKED-EVENT DATA-RECEIVED-EVENT DATA-RECEIVED-ERROR-EVENT DATA-SEND-COMPLETED-EVENT MODE-SWITCH-EVENT ASYNCHRONOUS-SERVER-CALL-RETURNS-EVENT MODE-SWITCHED-ACK-EVENT PORT-API-OPTION DATA-READ-ACCESS DATA-RECEIVE-POINT DATA-WRITE-ACCESS DATA-SEND-POINT SYNCHRONOUS-SERVER-CALL-POINT ASYNCHRONOUS-SERVER-CALL-POINT CALPRM-ACCESS MODE-SWITCH-POINT
OPERATION-INVOKED-EVENT DATA-RECEIVED-EVENT DATA-RECEIVED-ERROR-EVENT DATA-SEND-COMPLETED-EVENT MODE-SWITCHED-EVENT ASYNCHRONOUS-SERVER-CALL-RETURNS-EVENT MODE-SWITCH-ACK-EVENT	If the RUNNABLE-ENTITY referenced by the removed event has no more references, then the RUNNABLE-ENTITY will also be removed. RUNNABLE-ENTITY
RUNNABLE-ENTITY	OPERATION-INVOKED-EVENT DATA-RECEIVED-EVENT DATA-RECEIVED-ERROR-EVENT DATA-SEND-COMPLETED-EVENT MODE-SWITCHED-EVENT

AUTOSAR 3 Element to remove	Related AUTOSAR 3 Elements to remove
	ASYNCHRONOUS-SERVER-CALL-RETURNS-EVENT MODE-SWITCH-ACK-EVENT
INTER-RUNNABLE-VARIABLE	READ-VARIABLE-REF WRITTEN-VARIABLE-REF
DATA-ELEMENT-PROTOTYPE	OPERATION-INVOKED-EVENT DATA-RECEIVED-EVENT DATA-RECEIVED-ERROR-EVENT DATA-SEND-COMPLETED-EVENT MODE-SWITCHED-EVENT ASYNCHRONOUS-SERVER-CALL-RETURNS-EVENT MODE-SWITCH-ACK-EVENT PORT-API-OPTION DATA-READ-ACCESS DATA-RECEIVE-POINT DATA-WRITE-ACCESS DATA-SEND-POINT SYNCHRONOUS-SERVER-CALL-POINT ASYNCHRONOUS-SERVER-CALL-POINT CALPRM-ACCESS MODE-SWITCH-POINT QUEUED-SENDER-COM-SPEC UNQUEUED-SENDER-COM-SPEC QUEUED-RECEIVER-COM-SPEC UNQUEUED-RECEIVER-COM-SPEC If no more DATA-ELEMENT-PROTOTYPE elements left, the interface will also be removed. CALPRM-INTERFACE CLIENT-SERVER-INTERFACE SENDER-RECEIVER-INTERFACE
CALPRM-INTERFACE CLIENT-SERVER-INTERFACE SENDER-RECEIVER-INTERFACE	R-PORT-PROTOTYPE P-PORT-PROTOTYPE

AUTOSAR 4 Element to remove	Related AUTOSAR 4 Elements to remove
R-PORT-PROTOTYPE P-PORT-PROTOTYPE	OPERATION-INVOKED-EVENT DATA-RECEIVED-EVENT DATA-RECEIVED-ERROR-EVENT DATA-SEND-COMPLETED-EVENT SWC_MODE-SWITCH-EVENT ASYNCHRONOUS-SERVER-CALL-RETURNS-EVENT MODE-SWITCHED-ACK-EVENT PORT-API-OPTION VARIABLE-ACCESS SYNCHRONOUS-SERVER-CALL-POINT ASYNCHRONOUS-SERVER-CALL-POINT PARAMETER-ACCESS MODE-SWITCH-POINT
OPERATION-INVOKED-EVENT DATA-RECEIVED-EVENT DATA-RECEIVED-ERROR-EVENT DATA-SEND-COMPLETED-EVENT SWC_MODE-SWITCH-EVENT ASYNCHRONOUS-SERVER-CALL-RETURNS-EVENT	If the RUNNABLE-ENTITY referenced by the removed event has no more references, then the RUNNABLE-ENTITY will also be removed. RUNNABLE-ENTITY

AUTOSAR 4 Element to remove	Related AUTOSAR 4 Elements to remove
MODE-SWITCHED-ACK-EVENT	
RUNNABLE-ENTITY	OPERATION-INVOKED-EVENT DATA-RECEIVED-EVENT DATA-RECEIVED-ERROR-EVENT DATA-SEND-COMPLETED-EVENT SWC_MODE-SWITCH-EVENT ASYNCHRONOUS-SERVER-CALL-RETURNS-EVENT MODE-SWITCH-ACK-EVENT
IMPLICIT-INTER-RUNNABLE-VARIABLE EXPLICIT-INTER-RUNNABLE-VARIABLE	VARIABLE_ACCESS
VARIABLE-DATA-PROTOTYPE	OPERATION-INVOKED-EVENT DATA-RECEIVED-EVENT DATA-RECEIVED-ERROR-EVENT DATA-SEND-COMPLETED-EVENT MODE-SWITCHED-EVENT ASYNCHRONOUS-SERVER-CALL-RETURNS-EVENT MODE-SWITCH-ACK-EVENT PORT-API-OPTION VARIABLE-ACCESS SYNCHRONOUS-SERVER-CALL-POINT ASYNCHRONOUS-SERVER-CALL-POINT PARAMETER-ACCESS MODE-SWITCH-POINT QUEUED-SENDER-COM-SPEC NONQUEUED-SENDER-COM-SPEC QUEUED-RECEIVER-COM-SPEC NONQUEUED-RECEIVER-COM-SPEC If no more VARIABLE-DATA-PROTOTYPE elements left, the interface will also be removed. PARAMETER-INTERFACE CLIENT-SERVER-INTERFACE SENDER-RECEIVER-INTERFACE
PARAMETER-INTERFACE CLIENT-SERVER-INTERFACE SENDER-RECEIVER-INTERFACE	R-PORT-PROTOTYPE P-PORT-PROTOTYPE

4.7. Project Renaming

The pure::variants Connector for AUTOSAR support renaming of the projects. If the `rename` attribute on a `autosar:file` is set to `TRUE` then the following additional steps will be performed:

1. The names of AUTOSAR 3 elements `APPLICATION-SOFTWARE-COMPONENT-TYPE`, `COMPLEX-DEVICE-DRIVER-COMPONENT-TYPE`, `ECU-ABSTRACTION-SW-COMPONENT-TYPE`, `SENSOR-ACTUATOR-SW-COMPONENT-TYPE`, `SERVICE-SW-COMPONENT-TYPE`, `SWC-INTERNAL-BEHAVIOR` and `SWC-IMPLEMENTATION` will be extended by the name of the variant model.
2. The names of AUTOSAR 4 elements `APPLICATION-SW-COMPONENT-TYPE`, `COMPLEX-DEVICE-DRIVER-SW-COMPONENT-TYPE`, `ECU-ABSTRACTION-COMPONENT-TYPE`, `SENSOR-ACTUATOR-SOFTWARE-COMPONENT-TYPE`, `SERVICE-COMPONENT-TYPE`, `INTERNAL-BEHAVIOR` and `SWC-IMPLEMENTATION` will be extended by the name of the variant model.
3. All interface names will be extended by the name of the variant model.
4. All references to renamed elements are updated.

5. The file name of the ARXML file is extended by the variant model name.

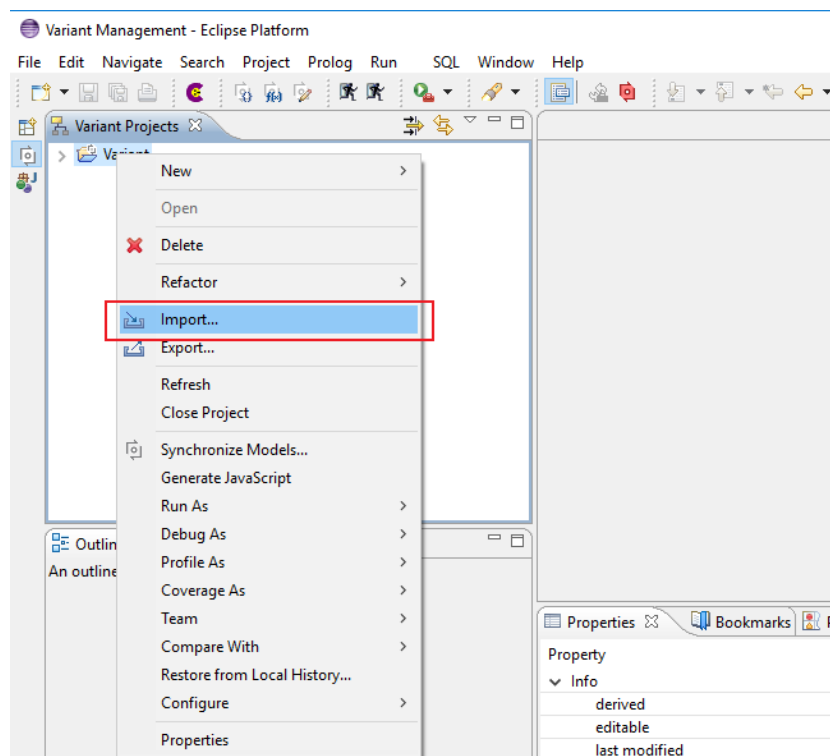
5. Autosar Feature Model Exchange Format Support

This supports the users to import feature and variant models from the Autosar Feature Model Exchange Format (ARXML) and also export the feature and variant models to the Autosar Feature Model Exchange Format (ARXML). The Autosar Feature Model Exchange Format (ARXML) files describes the structure and data of the models.

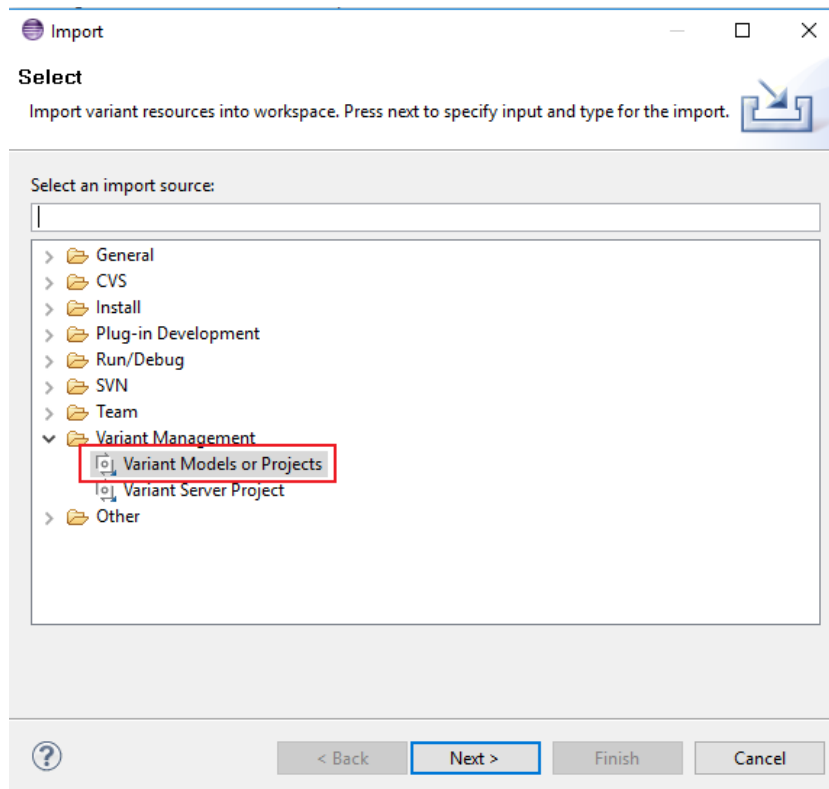
5.1. Import models from Autosar Feature Model Exchange Format (ARXML)

This feature accepts an Autosar Feature Model Exchange Format (ARXML) file as an input. The file will contain the necessary information to create the models. To Import the file, start Import Wizard by choosing **Import** in the context menu of the **Variant Projects** explorer.

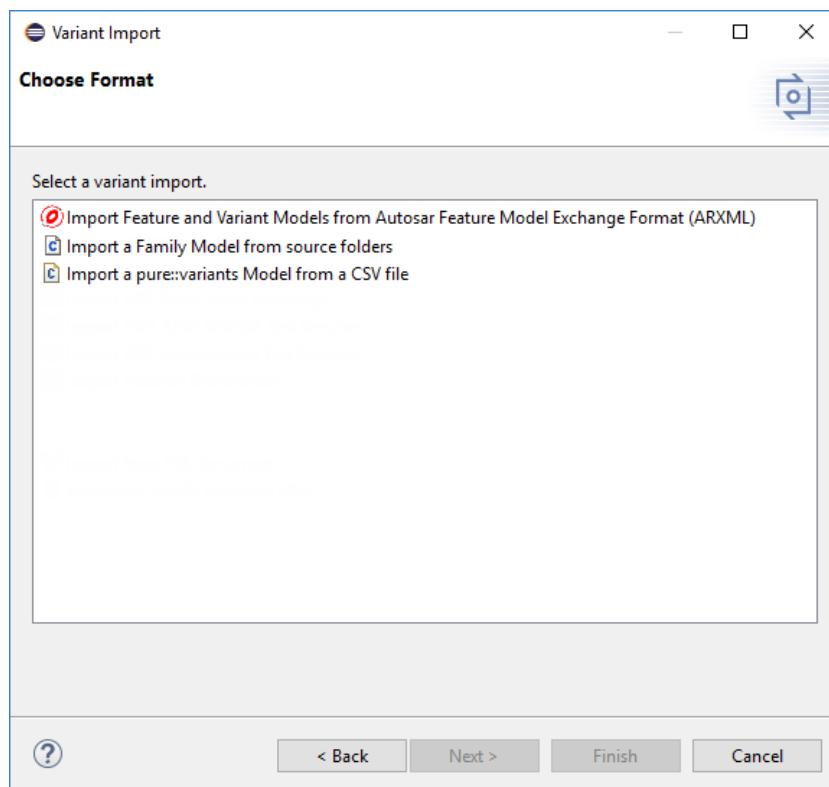
Figure 36. Start Import



On the next page, select the **Variant Models or Projects** option and click on **Next**.

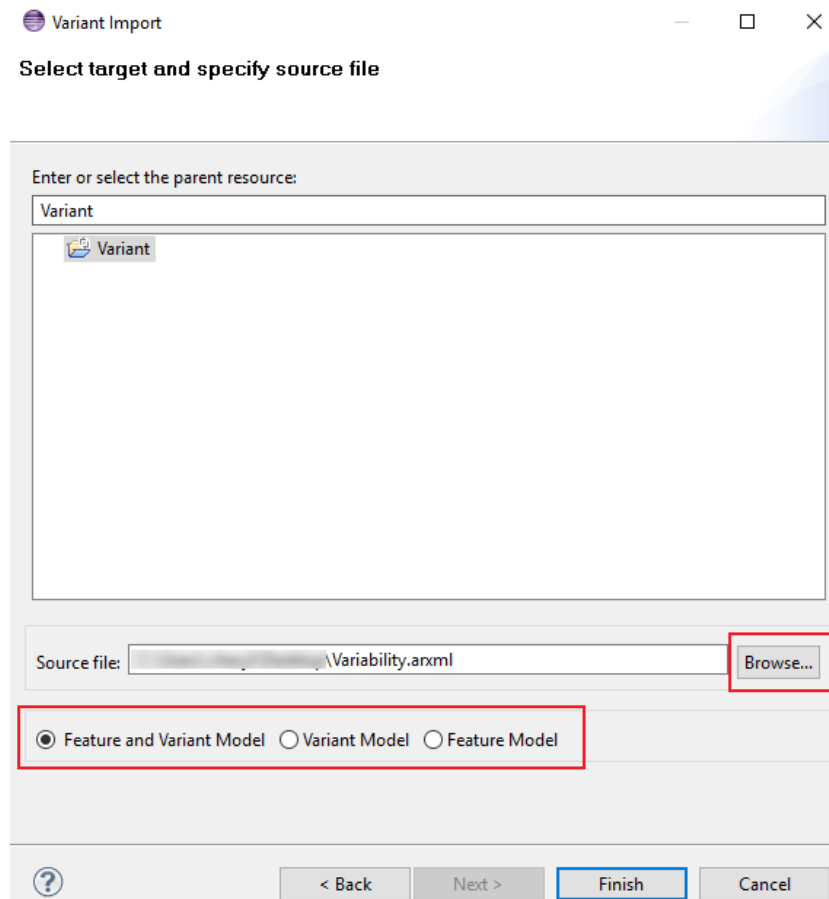
Figure 37. Start Import

On the next page, Choose **Import feature models and Variant models from ARXML** option and click on **Next**.

Figure 38. Start Import

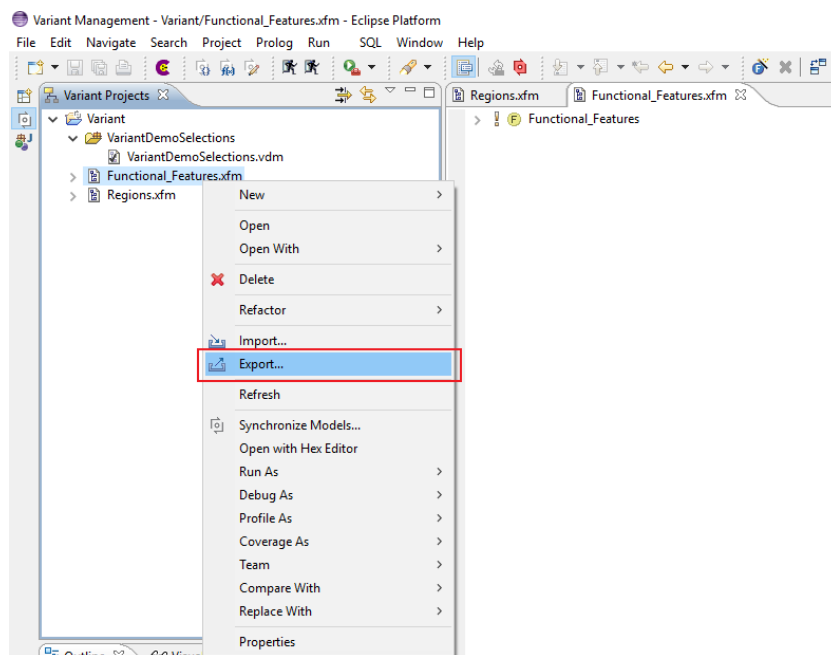
On the next page, Choose the target folder, in which the models will be created. Browse to the ARXML file that shall be imported. Click on one of the three model creation options (1. Feature and Variant Model, 2. Variant Model, 3. Feature Model). This will create the specified models accordingly. Then click on **Next**. For option 1 and 2 there is also a configuration space created and the created feature models are added as used models automatically.

Figure 39. Select target and specify source file

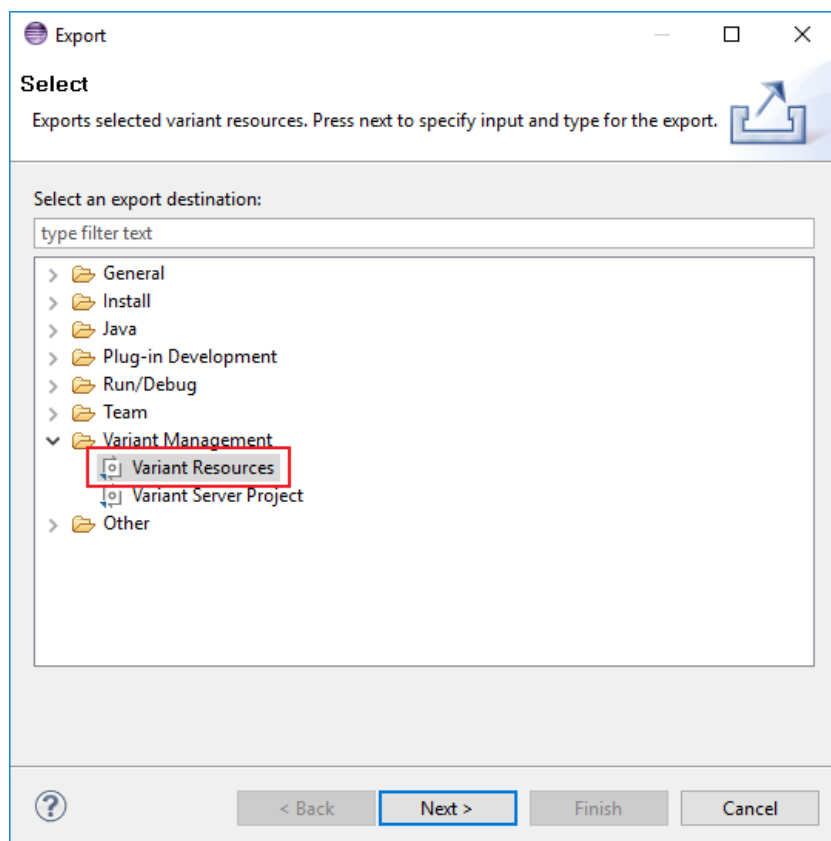


5.2. Export Models to Autosar Feature Model Exchange Format (ARXML)

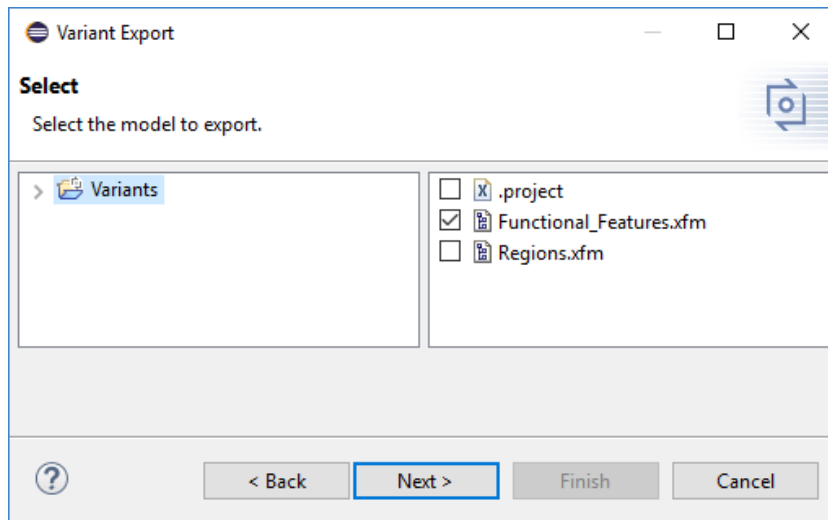
Feature models can be exported to the Autosar Feature Model Exchange Format (ARXML) by the **Export Wizard**. To export the feature models use the **Export...** action in the context menu of the **Variant Projects** view.

Figure 40. Feature Model Export 1

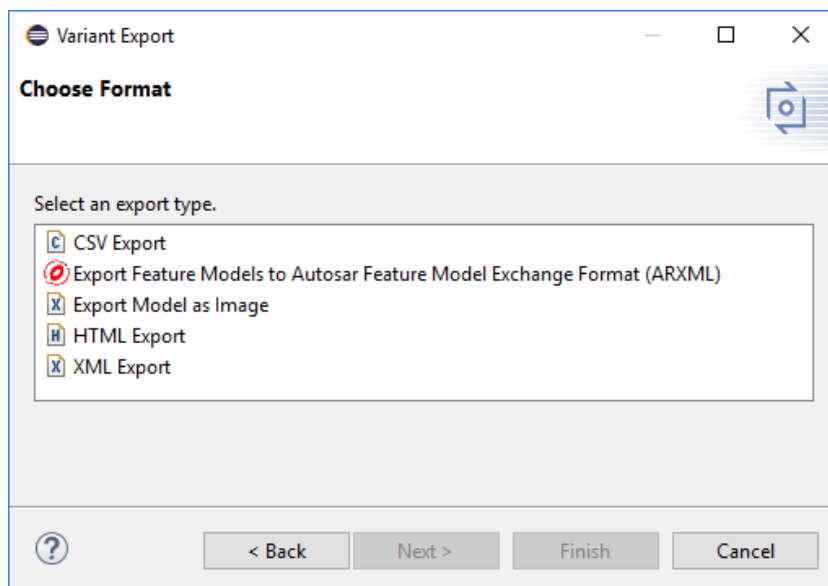
On the Export Wizard Page, Select the **Variant Management** -> **Variant Resources** . Then click on **Next** .

Figure 41. Feature Model Export 2

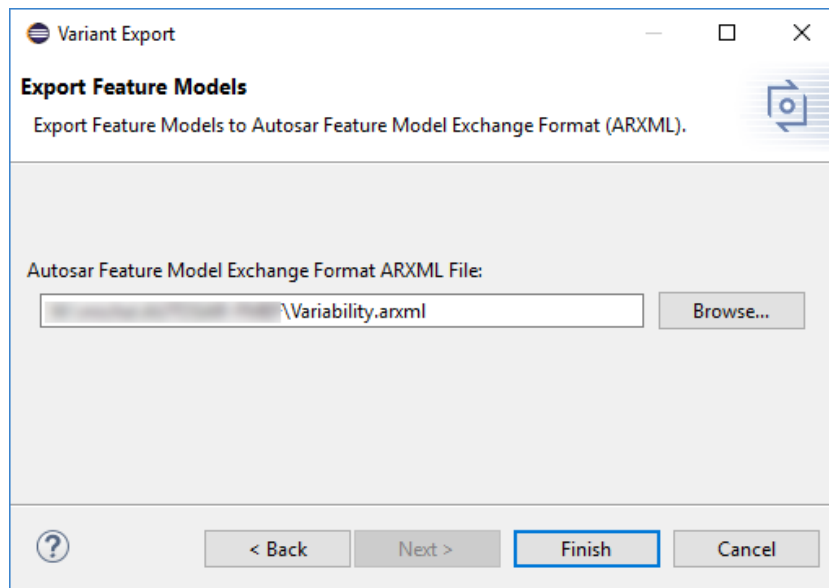
On the next page select the model that shall be exported. Only one model can be selected. It has to be a feature model. Click on **Next**.

Figure 42. Feature Model Export 3

On the next page select the **Export Feature Models to Autosar Feature Model Exchange Format (ARXML)** export format. Click on **Next**.

Figure 43. Feature Model Export 4

On the next page, browse to a new or already existing ARXML file. Click on **Finish** to start the export. The model will be exported into this file.

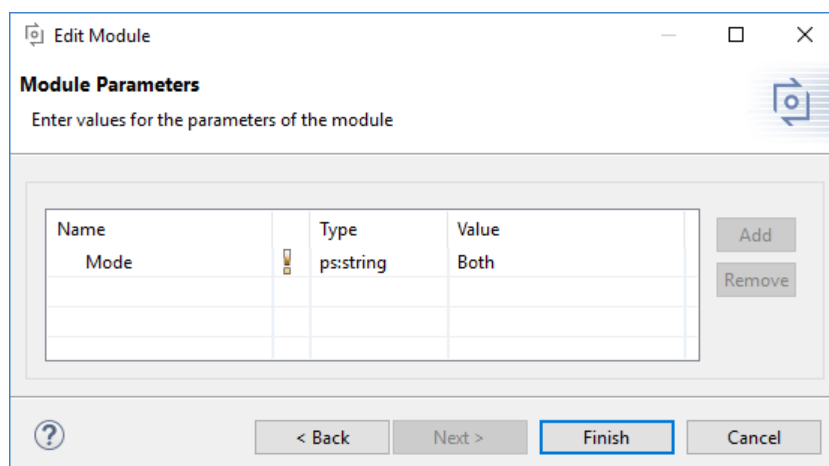
Figure 44. Select the target file

5.3. Transform a variant model into Autosar Feature Model Exchange Format

To transform a variant the configuration space has to be configured to add the **Autosar Feature Model Exchange Format (ARXML) Transformation** module. The transformation creates a new ARXML file in the output directory. The filename is <variant name>.arxml. This can not be configured.

The transformation module has the following parameter, which are used to configure the transformation:

- Mode - This parameter defines, which transformation mode is used
 - Both - All feature models of the variant and the selection information are exported during the transformation.
 - Selections - This mode exports the selection information of the transformed. The feature models are not exported in this mode.
 - Feature Models - This mode exporters the feature models of the variant only. The selection information is not exported.

Figure 45. Autosar Feature Model Exchange Format (ARXML) Transformation Module

5.4. Limitations

- If the user adds restrictions to the pure::variants model elements, the exporter ignores the restrictions and exports the model with all other features of the model.
- If an attribute contains multiple values, only the first value is considered as the default value and hence only one value will be exported to the ARXML file. The other values will be lost during export.
- Attribute calculations will not be exported.
- There are only two relation types supported (ps:requires and ps:conflicts). Relations with different types will be ignored.
- If a feature contains more than one long name, the first value will be considered. The other values will be lost during import.
