

---

# pure::variants Web Client Manual

pure-systems GmbH

Version 7.0.0.685 for pure::variants 7.0

Copyright © 2003-2025 Parametric Technology GmbH

2025

## Table of Contents

1. Introduction .....	2
1.1. Overview .....	2
1.2. Operating Concept and Navigation .....	2
1.3. Access Management .....	3
1.4. Operating Modes .....	4
1.5. Multi Language Support of Models .....	5
1.6. History of Model Changes .....	5
2. Login Page .....	6
3. Home Page .....	6
3.1. Projects List .....	7
3.2. Create Project .....	8
4. Project Page .....	8
4.1. Project Item Details .....	9
4.2. Open (Model, Configuration Space) .....	10
4.3. Open Matrix (Variant Model, Configuration Space) .....	10
4.4. Variant Transformation (Variant Model, Configuration Space) .....	10
4.5. Delete (All but Project) .....	10
4.6. Rename (All) .....	10
4.7. Create Revision (Project) .....	10
4.8. Edit (Project) .....	11
4.9. Create Configuration Space (Project, Folder) .....	11
4.10. Create Feature Model (Project, Folder) .....	11
4.11. Create Folder (Project, Folder) .....	12
4.12. Configuration Space Properties (Configuration Space) .....	12
4.13. Create Variant Model (Configuration Space) .....	13
4.14. Clone Variant Model (Variant Model) .....	13
4.15. Compare (Model, Project) .....	14
5. Model Page .....	14
5.1. Feature Models and Family Models .....	14
5.2. Variant Models .....	21
6. Variant Configuration Wizard .....	27
7. Variant Matrix .....	29
8. Compare .....	31
9. Import Assets as Family Models Using the WebClient .....	33
10. Transformation .....	35
10.1. Transformation Services Concepts .....	35
10.2. Transformation Panel .....	35
10.3. Transformation Dashboard .....	38
10.4. Transformation Configurations .....	39
11. User Settings .....	41
12. System Administration .....	42
13. Model Picker .....	43
14. OSLC Services .....	44
14.1. Using the OSLC Configuration Management Service .....	44
14.2. Add traceability links to OSLC-capable Tools .....	47

15. Known Limitations of the pure::variants Web Client .....	49
--	----

## 1. Introduction

The pure::variants Web Client provides an installation free, web browser-based way for users to access pure::variants models and perform specific use cases outside of pure::variants Enterprise Desktop Client. These use cases cover authoring, i.e., constructing feature and family models, as well as variant models. Furthermore, besides managing variant projects, it is possible to run and supervise variant transformations from within the Web Client.

The pure::variants Enterprise Web Client shall be considered as complementary to the functionality provided by pure::variants Enterprise Desktop Client.

### 1.1. Overview

The Web Client is structured into following pages with dedicated functionality:

- Login Page
- Home Page
- Project Page
- Model Page
- Variant Matrix
- Variant Configuration Wizard
- Transformation Dashboard
- User Settings
- System Administration

This help document provides detailed information about each of the pages. This document does not cover the general pure::variants modelling concepts and languages. Such information is provided in the separately available pure::variants User's Guide. Information regarding setting up and managing a pure::variants Web Client Deployment is contained in the pure::variants Setup Guide.

### 1.2. Operating Concept and Navigation

Primarily the *Navigation Bar* on the top of the pages as well as context sensitive actions related to models or model elements can be used for navigation. Additionally, hyperlinks provide direct access to resources.

**Figure 1. Navigation Bar**

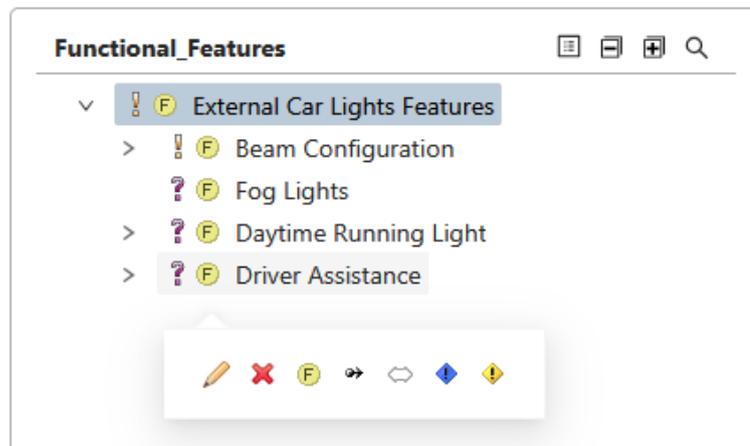


Selecting the pure::variants Logo on the navigation bar returns the user to the [Home Page](#). The Home Page is also loaded in error cases, e.g., when the targeted resource does not exist anymore. For administrators the navigation bar provides an additional menu item (  ) next to the logout to *open the [System Administration](#) Page*.

Depending on the context, the page opens in a new browser tab, or the content is loaded into the existing tab.

### Tree Item Actions

The actions related to model and project tree items are shown when hovering over the element using the cursor, then the action needs to be selected by clicking the desired item.

**Figure 2. Hover-over Menu**

Clicking on an item in the navigation tree of the Project and Model pages selects this item and loads its item details in the right Item Details area of these pages. In the [Project Page](#), the tree permits multiple selections by clicking on additional items with the left mouse button with the CTRL key is being pressed. In [Model Pages](#), multiple selection of items is not available.

When hovering over one of multiple selected items in the project page, the actions shown will be executed on all selected items, when hovering over a not selected item, the action applies only to the currently hovered-over item.

Numerous fields in the Item Details sections are editable in place in addition to the actions as provided in the trees.

Concurrent editing of the same item by different users is managed according to the *first come, first served* concept. Web Client displays a notification about the concurrent change to the second user trying to save changes. This way the second user can secure the modified content e.g. by copying it to the clipboard before closing the editor.

### 1.3. Access Management

Access to pure::variants resources using the Web Client is regulated according to the user roles and access rights as defined by the administrators in the connected model server.

When working with the Web Client, the following special roles are being supported:

- **Reader** role: User, who has the *Reader* role assigned, can only view resources, but are not able to perform any changes to them. All change actions are rejected by the system. This includes all kind of changes to project items, including changing selections in variant models. Transformations cannot be scheduled by a user with *Reader* role.

Additionally, for users with *Reader* role, the functions of web-based tool integrations provided by pure::variants Connectors will be disabled in connected tools.

Users with *Reader* role are not counted towards the maximum permitted licensed users count when accessing pure::variants through the pure::variants Web Client.

- **Transformation Administrator** role: Users with this role can manage certain aspects of the pure::variants Web Transformation Service. They see transformation jobs of all users of the connected pure::variants Transformation Service and are able to cancel or delete these. They are also able to set the operational status of the services. For more information, see the section on [Transformation](#) in this document.
- **Project Administrator** role: Users with this role can access the [System Administration](#) Page as well as can change the message shown to all users in the Information area of the [Login](#) and [Home](#) pages

### Change Access Rights

Access rights are available for projects, folders, configuration spaces and models, changing them is possible on the Project Page, in the corresponding [Project Item Details](#) section.

### Figure 3. Change Access Rights



The Permissions section provides following options:

- **Effective Permissions of User:** The effective permissions to the particular item are calculated based on the permissions from all roles assigned to the user, as well as those permissions directly enabled for the user. The *Effective Permission Details* section provides information about what data was used for this calculation.
- **Roles with Access:** Access can be granted to specific roles and users by assigning them to the selected item. Permissions (read, change, delete and change permission) can be also set here. If the “Apply permissions additionally to all subordinate models and elements” box is checked, the new permissions are applied recursively to all subordinate elements.

Editing is only possible for users who have change permission rights to the selected item.

## 1.4. Operating Modes

The Web Client operates in two different modes. The *Standard* mode and the *Global Configuration* mode. In *Standard* mode, a user can view and work with all projects and revisions they have access to based on the defined access rights. In *Global Configuration* mode, a user can only see and access project/revision combinations as defined in the specified OSLC Global Configuration Context.

By default, *Standard* mode is active. To work in *Global Configuration* mode, an OSLC compliant global configuration context parameter must be provided as part of the resource URL for a project or model. This happens automatically when opening the Web Client through the respective action from a Global Configuration enabled pure::variants Connector such as the pure::variants Connector for IBM DNG. The Web Client itself does not offer its own user interface for selecting Global Configurations and switching to *Global Configuration* mode.

### Navigation in Global Configuration Mode

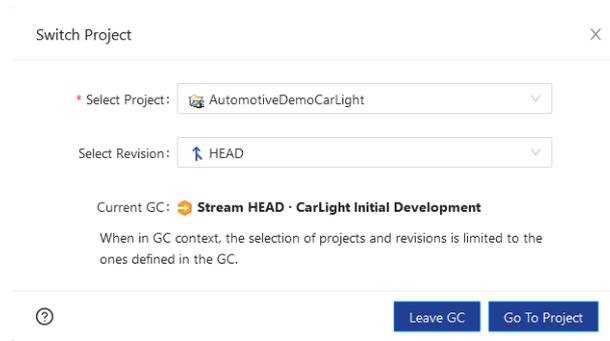
The operation in Global Configuration mode is visualized by different icons for revisions (branch: , version: ) when displayed, for instance, in the Navigation Bar. (See [Figure 4, “Navigation bar with Global Configuration context”](#)).

The *Global Configuration* mode is not affecting the operation of the Web Client Home Page. All projects are listed and opening a project from there will always open it in *Standard* mode. As the mode is selected on a per page URL basis, users may have the same project open in both modes in different browser tabs/windows.

Figure 4. Navigation bar with Global Configuration context



In Global Configuration mode, the Switch Project and Switch Revision dialogs offer only options from within the active Global Configuration context. Additionally, user can leave the active context using the **Leave GC** button that is offered on both of the dialogs. (See [Figure 5, “Switch Project Dialog in Global Configuration Mode”](#)).

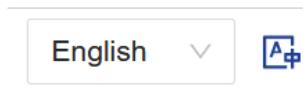
**Figure 5. Switch Project Dialog in Global Configuration Mode**

## 1.5. Multi Language Support of Models

The Web Client is able to deal with multiple languages for the visible name of items, rationales and descriptions. This allows to define Feature and Family Models, document rationales in Variant Models as well as describe Projects in more than one language.

The default language used for models can be set as *Model Language* in the *Preferences* of the [User Settings](#) Page. This setting is also stored locally in the browser, i.e. a user can define different settings in different browsers. The initial setting is *'Unspecified'*. This can be used like other languages. Typically, it is used when the language of visible names and descriptions do not play a role.

To view or edit visible names, rationales or descriptions for a particular language use the language selector in supported editors ([Figure 6, "Language Selector in Editors"](#)).

**Figure 6. Language Selector in Editors**

In addition to that already available translations can be displayed in a separate section of the editors by pressing the (A+) button. Here, any other existing content can be displayed and used as a read-only reference for the translation process. The content for multiple languages can be edited and then saved in one editing session.

In case there is no translation available in the default language for a particular item, but they exist in other languages the editors will display this as a fall back. In this case the language selectors have also that language selected. There is a precedence for fallback languages: content defined in *'Unspecified'* followed by *'English'* have priority.

Dialogs that have no multi editing support also display relevant content according to the same strategy. Editing the values here update the content in the displayed language.

Please note that editors in the Web Client support HTML formatted content only both for display and editing. Make sure to use HTML editors in Desktop Client as well to work with the same data in both tools.

## 1.6. History of Model Changes

In the Web Client model change history is available both on model and item level. The history of changes corresponding to all items of a model can be accessed by pressing the Details button (📄) available for models, while the list of changes specific to a selected item is available in the item details area.

The history entries are sorted according to change date, providing information also about the corresponding revision and the user who did the change. The list includes changes in all revisions traced back to the initial one for the particular item. Expanding an entry gives details about the change itself and the impacted element.

**Figure 7. History of Model Changes**

Following actions are available:

- **Add comment:** After selecting one or multiple history entries, a change comment can be added to them. Once added, they can not be changed anymore. This functionality is available for users with change permission rights.
- **Filter:** History comments can be filtered by user or date range. Additionally, the number of listed items can be defined.

## 2. Login Page

If users are not already authenticated, they must login before being able to access the pure::variants Web Client.

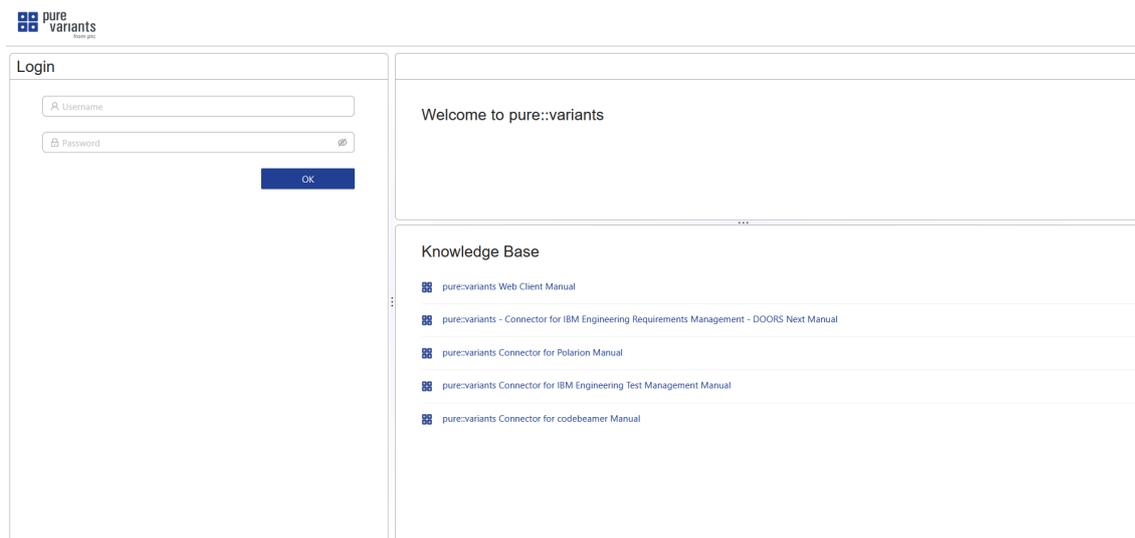
The [Login Page](#) is split into three areas, the *Login* area, the *Knowledge Base* area, and the *Information* area.

The Login area asks for the users' credentials. The type of credentials required to log in to the Web Client may differ depending on the service configuration. If the pure::variants Web Client is configured to use a separate Single-Sign-On service, a dedicated login button is displayed that takes the user to the respective page of the service. Alternatively, when using Single-Sign-On, the Web Client can be also configured to bypass the Login Page and directly offer the page of the login service.

Administrators may provide information to all users through the Information Area in the top right part of the page. User can also browse the Knowledge Base area on this page, which includes all connector manuals and this pure::variants Web Client Manual. Please note, that on smaller screen sizes, these areas may not be shown to the user.

Sections of the layouts can be resized on supported pages of the Web Client. The modified layouts are saved and stored locally in the browser.

**Figure 8. Login Page**



## 3. Home Page

Following a successful login, a user will be redirected to Home Page (see [Figure 9, “Home Page”](#)). This page is divided into four areas

- **Projects:** All available *projects* are listed on left. Clicking the name of a project opens this project in a new browser tab. The expand action displays the project's description as defined in the HEAD branch of this project. There are further actions available for the projects that are described in the [Projects List](#) section of this document.

Based on the central configuration of the pure::variants Web Client Deployment, projects that the user cannot read any revisions of will either appear greyed out or will not be visible in the Projects List at all (see pure::variants Setup Guide for more details on the configuration).

- **Recents:** The tabs for *Recent Projects* and *Recent Models* provide quick access to the 5 most recently viewed pure::variants models and projects a user had opened in the past in the respective browser. Moreover, the grayed out entries in the respective lists indicates that the models/projects are no more accessible by the current user, both the lists get updated automatically according to the recent activities.

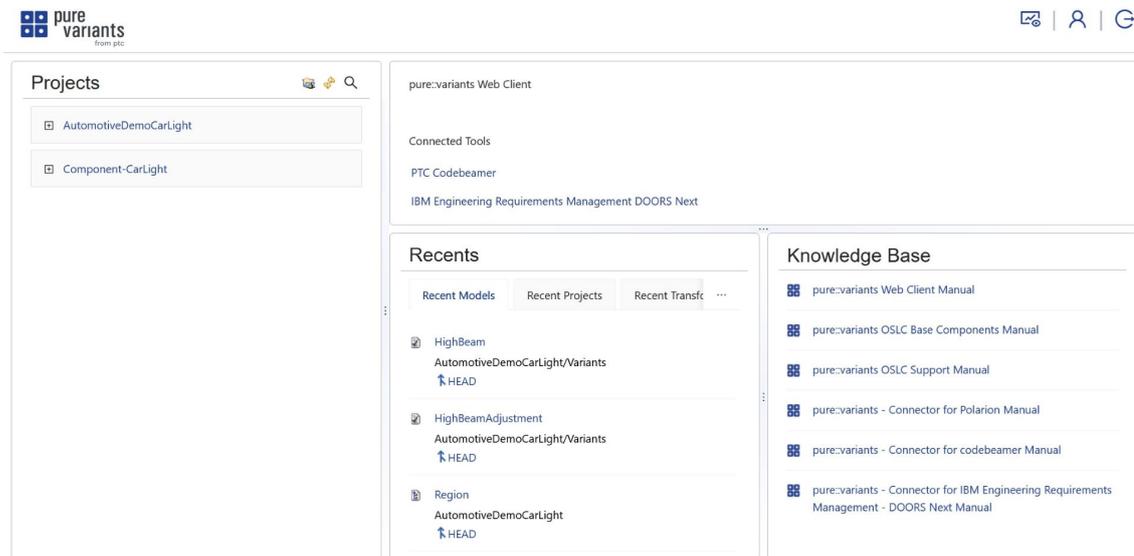
*Recent Transformations* shows the list of the last 25 transformations a user has access to. The Transformation Details Dialog can be accessed by clicking the title of a transformation in the list. See the section Transformation for more details.

*Recent Models* and *Recent Projects* data is stored locally in the browser and not in a central location. Using a different browser will thus result in a different list. *Recent Transformations* are retrieved through the service and thus shared across browsers.

- **Knowledge Base:** The *Knowledge Base* section provides access to all available online help. Typically these are the pure::variants Web Client Manual and manuals for installed pure::variants Connector. This area can also be accessed from *Login Page* before being logged in.
- **Information Area:** This area is located in the top right part of the page. It provides information to all users of the Web Client. Users with Project Administrator role can change this message using the Edit action (Pencil icon in the top right corner of this area).

The home page also provides the quick navigation button to access the [Transformation Dashboard](#), [User Settings](#) and [Log Out](#) by clicking on respective icons on top right corner of the Navigation Bar. For Administrators, the [System Administration](#) Page is also available.

**Figure 9. Home Page**



### 3.1. Projects List

There are three actions available in the top bar of the Projects List:

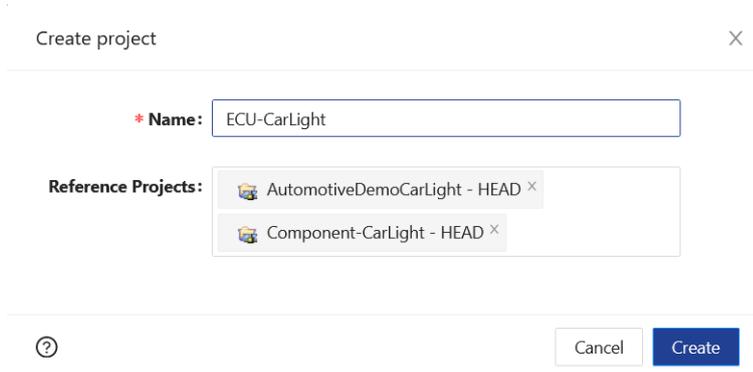
- **Create Project:** *Create* new projects with providing project name and optionally selecting referenced projects.
- **Search:** *Search* based on the search string, project items with matching names are displayed.
- **Refresh:** *Refresh* the list of projects.

Hovering over the project list entries, the action to delete the project is provided. Similarly, edit and delete options are available for the project descriptions (see [Figure 10, “Editing Actions of the Projects List”](#)).

**Figure 10. Editing Actions of the Projects List**

### 3.2. Create Project

To create a new project, a unique name for the project needs to be provided. Optionally, the list of referenced projects can be provided here, but this can be edited later as well (see [Edit Project](#)). As a reference for a project, only one revision of the same project is allowed.

**Figure 11. Create Project Dialog**

## 4. Project Page

This page displays the structure of a single pure::variants project and provides actions to interact with project items such as models, configuration spaces and folders.

The page is split into a **Project Item Tree** in the left area and the item details area on the right. The **Item Details** area shows additional information for the currently selected project item.

On top is the **Navigation Bar** displayed. Like on a [Model Page](#) the Navigation Bar on a Project Page provides the name of the current project, revision (below the project name, icons for branch: and version: ) and model item and location in the project. These names also act as quick selectors to easily navigate between projects, revision, or switch to different models inside a project. Clicking on one of the names starts the respective selector.

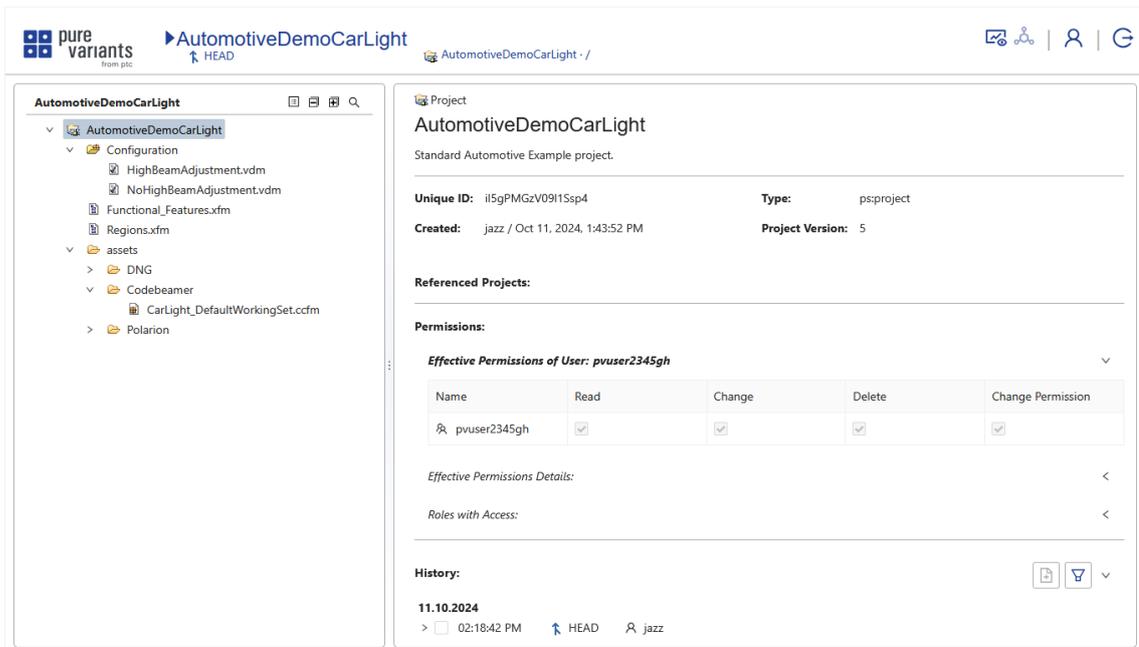
The Navigation Bar of the home page also provides quick navigation buttons to access the [Transformation Dashboard](#), [User Settings](#) and [Log Out](#).

The user can also *run Transformation* for variant models selected in the project item tree or for all models contained in a configuration space when selecting it. Further, the user can *check the transformation state of a running transformation* via the **Transformation Indicator** entry in the Navigation Bar. Hovering over the Transformation Indicator will provide the status of running transformations in an overlay menu with an additional button to start a new transformation. The transformation log can be opened by clicking on the transformation status.

There are following actions available in the top bar of the Project Item Tree:

- **Details:** Display information related to model instead of an item. History of model changes corresponding to all items is such information.
- **Collapse or Expand:** *Collapse* or *Expand* the tree below a selected project item. If no item is selected, the whole tree is expanded/collapsed.
- **Search:** *Search* based on the search string, project items with matching names are displayed. Selecting an item navigates to it in the item tree.

Figure 12. Project Page



## 4.1. Project Item Details

The item details display information about the currently selected item. Some of this information is common to all items such as the name, the internal id, the description of the item (if available), [Change History](#) related to the selected item, as well as the option to check and modify access rights including permissions to this item. Other information is specific for the type of selected item.

All item types but folders provide the capability to have a description which is displayed in the top area of the item details. Hovering over the description area provides actions to edit and delete the description of an item. The description of an item may have images and hyperlinks beside text.

Project references can be also added, modified or deleted here. Actions are available by hovering over the Project References section or the individual entries in the list.

## Convert Project Version

Project created in earlier versions (e.g. 4 or 5) can be converted to the latest available version, using the dedicated button offered for the project version. The conversion can not be undone.

Figure 13. Convert Project Version

## Create pvSCL Code Libraries

For feature and family models it is possible to create and edit a *pvSCL Code Library* to add pvSCL code that can be reused throughout the project in pvSCL expressions, including other pvSCL Code Libraries. Reuse is possible from the same or other referenced projects.

## 4.2. Open (Model, Configuration Space)

For feature models; family models; variant models; and configuration spaces including contained configuration space folders, the *Open* action (🔗) will open a new browser tab with the selected project item's page. It is possible to select multiple project items and then open up to 5 of them with a single action. Multiple selection requires to hold down the CTRL key while selecting the item with left mouse click.

## 4.3. Open Matrix (Variant Model, Configuration Space)

For configuration space project items and configuration space folders; and when multiple variants are selected, the context menu provides the *Open matrix* action (📊), which displays all models of a configuration space, configuration space folder or in case of multiple selected variants models, the selected models in a new browser tab as Variant Matrix page. Multiple selection requires to hold down the CTRL key while selecting the item with left mouse click.

## 4.4. Variant Transformation (Variant Model, Configuration Space)

Transformations can be also started using the action (🔄) after selecting a variant model or the configuration space.

## 4.5. Delete (All but Project)

The Delete action (✖) is available for all project items except for the project root item itself. Deleting a project item will also delete all child items. Multiple selection requires to hold down the CTRL key while selecting the item with left mouse click.

## 4.6. Rename (All)

The Rename action (🔄) is available for all project items including the project root item itself that renames the project. Renaming a project item does not update references to the previous name with the new name in pure::variants models.

Renaming a project is only possible in the HEAD revision.

## 4.7. Create Revision (Project)

New *Revisions* of a pure::variants project (either a writable branch or a read-only version) can be created using the *Create Revision* action (📄) in the context menu of the project item. This opens a dialog box, where a user can create branches (📄) and versions (📄).

The naming restrictions for revisions and potential conflicts with existing revision names are being checked while entering the data in the dialog. Notifications are displayed about the result of the creation process, including links to the created revisions that can be used to navigate to them.

**Figure 14. Create Revision**

The screenshot shows a dialog box titled "Create revision" with a close button (X) in the top right corner. Below the title bar, there are two radio buttons: "Branch" (selected) and "Version". Under "Branch", there is a text input field for "Branch Name" containing "Branch\_Number\_Three". Under "Version", there is a text input field for "Version Name" containing "Root\_Branch\_Number\_Three". At the bottom, there is a question mark icon, a "Cancel" button, and a "Create" button.

## 4.8. Edit (Project)

The list of referenced projects can be edited using the Edit action (✎) in the context menu of the root item of a project. The name of the project cannot be changed. As a reference for a project only one revision of the same project is allowed, and references to the project itself is not permitted.

**Figure 15. Edit Project**

Dialog: Edit project

Name: ECU-CarLight

Reference Projects:

- AutomotiveDemoCarLight - HEAD
- Component-CarLight - HEAD

Buttons: Cancel, Save

## 4.9. Create Configuration Space (Project, Folder)

A new configuration space can be added to the project by the *Create Configuration Space* action (📁) on project and folder items in the Project Page. To create a configuration space, a unique name needs to be provided using the presented dialog. All other data of the Configuration Space as well as Transformation Configuration can be defined in the WebClient later.

## 4.10. Create Feature Model (Project, Folder)

A new Feature Model can be created in an existing project by the *Create Feature Model* action (📁) in the context menu of the project item or project folders outside of configuration spaces. The creation is supported by a wizard (see Figure 16, “Create Feature Model”). More information on the Feature Model concept is provided in the [Feature/Family Model Page](#) section in this document and in the pure::variants User Guide.

**Figure 16. Create Feature Model**

Dialog: New Feature Model

Tabs: General, Destination, Config Spaces

Message: Please enter a unique feature model name.

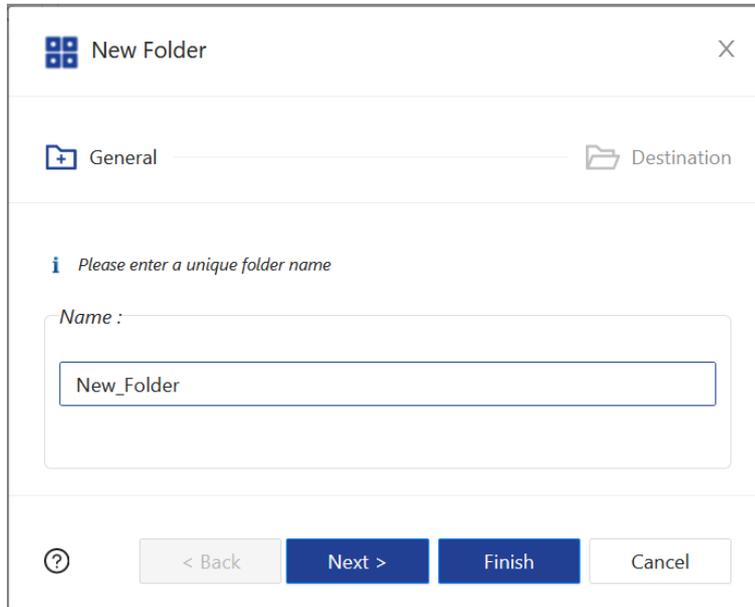
Name: Baselight\_EMEA

Buttons: < Back, Next >, Finish, Cancel

## 4.11. Create Folder (Project, Folder)

A new folder can be added to the project by the *Create Folder* action  on project and folder items in the Project Page. A wizard (see [Figure 17, “Create Folder”](#)) guides the user through the process of folder creation and permits to select any available destination for the new folder. The default destination is the project item from which the action was started.

**Figure 17. Create Folder**

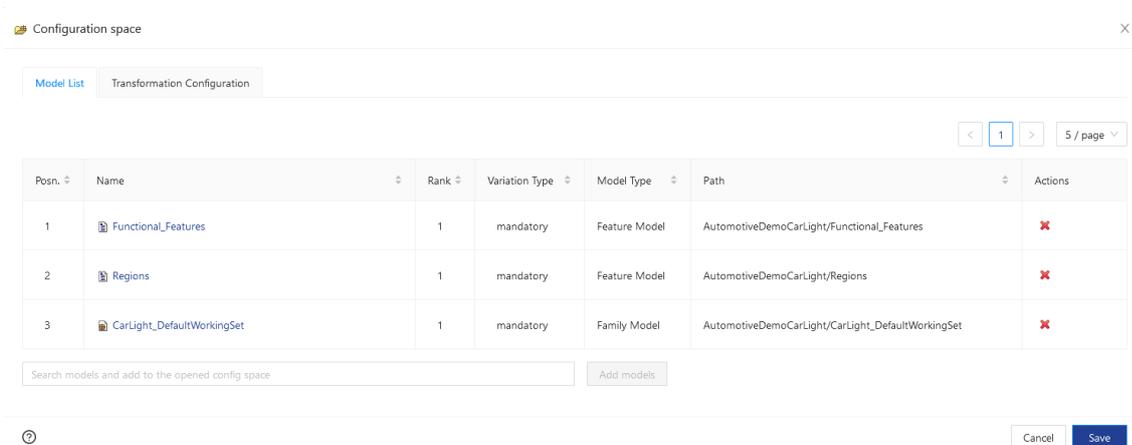


## 4.12. Configuration Space Properties (Configuration Space)

The *Configuration Space Properties* are accessed through the Properties action  available on configuration space project items in the Project Page.

The **Model List Tab** can be used to *add/delete models* which are members of the configuration space model list. (See [Figure 18, “Configuration Space Properties - Model List”](#)). The search field at bottom left of the dialog displays available models with matching names from its containing projects and all projects which this project is referencing.

**Figure 18. Configuration Space Properties - Model List**



Posn.	Name	Rank	Variation Type	Model Type	Path	Actions
1	Functional_Features	1	mandatory	Feature Model	AutomotiveDemoCarLight/Functional_Features	
2	Regions	1	mandatory	Feature Model	AutomotiveDemoCarLight/Regions	
3	CarLight_DefaultWorkingSet	1	mandatory	Family Model	AutomotiveDemoCarLight/CarLight_DefaultWorkingSet	

Using the **Transformation Configuration Tab** parameter values of the transformation module instances can be set. (See [Figure 19, “Configuration Space Properties - Transformation Configuration”](#)). First, the transformation

needs to be selected, then the specific transformation module for which the transformation module instance parameters get listed. Mandatory attributes must have a value otherwise the changes can not be saved. Attribute values are displayed read only for those transformation module instance types that are not supported in the WebClient.

**Figure 19. Configuration Space Properties - Transformation Configuration**

The screenshot shows a web-based configuration interface for a transformation module. It is titled "Configuration space" and has a close button (X) in the top right. There are two tabs: "Model List" and "Transformation Configuration".

Under "Transformation Configuration", there are three main sections:

- Select transformation:** A dropdown menu currently showing "DNG\_Stream".
- Module instances:** A list of radio buttons. "IBM Rational DOORS NG Module" is selected, and "Global Configuration Transformation" is unselected.
- Module instance parameters:** A form with several fields:
  - VariantRoot:** /variants
  - Modus:** Variant Enumeration
  - StreamName:** \$(VARIANT)\_\$(QUALIFIER)
  - PerformPartialText...:** false
  - Name:** (empty field)

At the bottom right, there are "Cancel" and "Save" buttons.

### 4.13. Create Variant Model (Configuration Space)

A new *Variant Model* can be created through the *Create Variant Model* action (🔗) on a configuration space or folder inside a configuration space directly from the [Variant Model Page](#). More information on the *Variant Model* concept is provided in the pure::variants User Guide.

A variant model is created by giving a variant model name, destination of the model, inheritance and type of configuration, i.e., full configuration or partial configuration (see [Figure 20, "Create Variant Model"](#)).

**Figure 20. Create Variant Model**

The screenshot shows a wizard window titled "New Variant Model" with a close button (X) in the top right. It has four tabs: "General", "Destination", "Inheritance", and "Configuration". The "General" tab is selected.

Below the tabs, there is a warning message: "Empty strings not permitted for a variant model name." Below this is a "Name:" label followed by a text input field containing the placeholder text "Enter Variant Model Name ...".

At the bottom, there is a navigation bar with a help icon (?), a "< Back" button, a "Next >" button (highlighted in blue), a "Finish" button, and a "Cancel" button.

### 4.14. Clone Variant Model (Variant Model)

An existing variant model can be cloned, resulting in a new variant model with identical content, but different name and id through the *Clone Variant Model* action (🔗). Once cloned, any changes in a cloned variant model have no impact on the original variant model it was cloned from. The action starts a wizard which guides through the cloning process. A clone may be saved in any available configuration space in its project or any folder inside a configuration space.

When creating a clone of the variant model, the user may assign a new unique model name (see [Figure 20, “Create Variant Model”](#)) if the suggested default name is not suitable.

**Figure 21. Clone Variant Model**

## 4.15. Compare (Model, Project)

Using the *Compare with* action () , revisions or baselines of feature, family and variant models, as well as project models can be compared with each other. Differences can be viewed and selected changes can be merged to the other revision using the dedicated Compare Page. Further, selecting multiple variant models from the same revision or baseline can be compared with each other (select 2 models for a two way, or 3 models for a three way compare).

In *Global Configuration* mode, another Global Configuration *streams or baselines* can be selected as revisions to compare with.

More information on the Compare functionality is provided in the [Compare](#) section in this document.

## 5. Model Page

The *Model Page* can be used to view and edit pure::variants feature models, family models and variant models.

Please consult the sub-chapter about [Feature Models and Family Models](#) for details on how to use the Model Page for viewing and editing feature models and family models.

Read the sub-chapter about [Variant Models](#) for details on how to view and edit variant models via the Model Page.

### 5.1. Feature Models and Family Models

When opening a feature model or family model of a pure::variants project via the *Open* action () on the [Project Page](#), the Model Page for feature models and family models opens in a separate browser tab. The Model Page consists of two vertically oriented areas and a main menu at the top.

The **Model Item Tree** in the left area displays the content of the opened model, including model elements (e.g., Features, Components, Parts, ...), Attributes, Values, Relations, Restrictions and Constraints. The name of the model is displayed on the top left of this area while the top right provides buttons to *collapse*, *expand* and *search* the content of the model.

The **Item Details** area on the right displays the *details* (e.g., Unique Name, Class/Type, Model, Created, Changed) of the model item that the user has selected in the Model Item Tree.

The **Navigation Bar** on top of the Model Page shows (from left to right) the pure::variants logo, the project name and revision of the currently opened model, the model name as well as menu entries for the Problem Indicator, the Validation button, the Toggle Automatic Validation button, the Transformation Dashboard, the User Settings and Logout.

Via the pure::variants logo in the Navigation Bar the user can *navigate back* to the [Home Page](#).

By clicking on the project name the user can *switch to another project and to another revision*. This will navigate the user to the Project Page of the selected project in the selected revision.

The user can *switch to a different revision of the current project* without leaving the model page by directly clicking on the revision name in the Navigation Bar.

By clicking on the model name in the Navigation Bar the user activates a drop-down model *search* for the current project and can directly *switch to a different model* of this project. The model will be opened in the currently active browser tab.

The user can navigate to all information, warning and error markers of the variant model via the **Problem Indicator** . Hovering over the Problem Indicator will provide a list *with all information, warning, and error markers* of the feature or family model concerning model validation. Clicking on one of the markers in the list will switch the focus of the Model Item Tree to the model item to which the marker is pointing. The markers are also directly listed in the Model Item Tree area. The user can assess the suggested quick fixes to resolve an error by clicking on the error marker in the Model Item Tree.

The user can *trigger the validation of the feature or family model* manually by clicking on the **Validate** menu entry  in the Navigation Bar.

The user can turn the *automatic validation of the variant model* on or off after each configuration step by making use of the **Toggle Automatic Validation and Evaluation** menu entry  in the Navigation Bar.

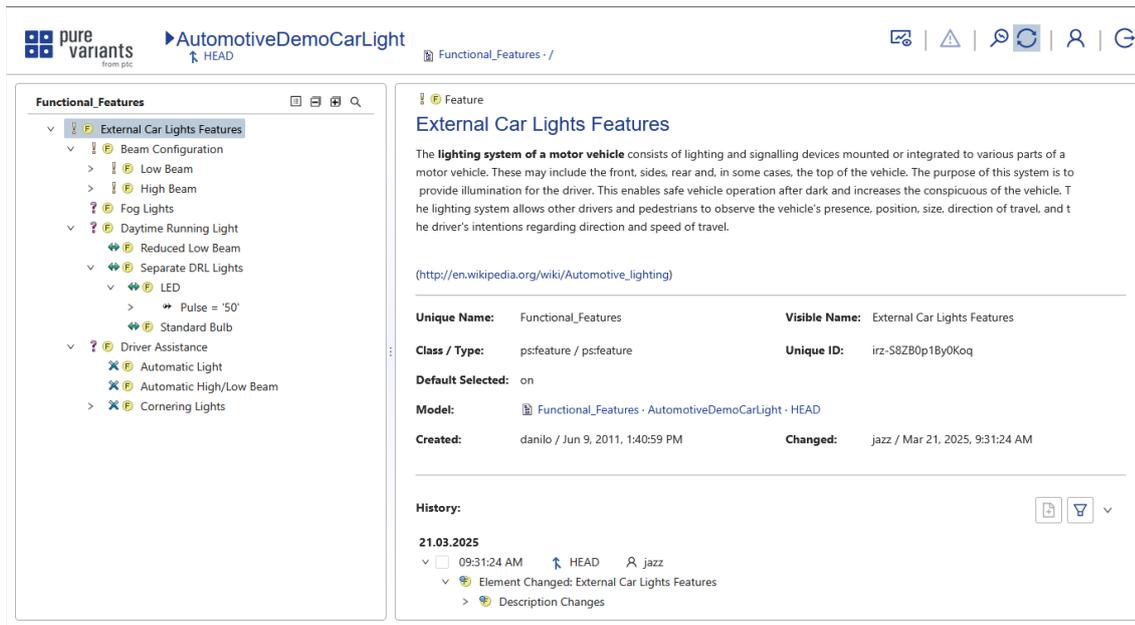
The user can *open the Transformation Dashboard* in a separate browser tab by clicking on its menu entry  in the Navigation Bar.

The user can *open the User Settings* by clicking on the respective menu entry  in the Navigation Bar. The User Settings will open in the currently active browser tab.

By clicking on the Logout menu entry  in the Navigation Bar the user can *end the Web Client session*. The user will have to log in again via the [Login Page](#) to continue working with the Web Client.

There are different actions which can be performed on model items of feature models and family models which will be explained in the following sub-chapters.

**Figure 22. Model Page for Feature Model**

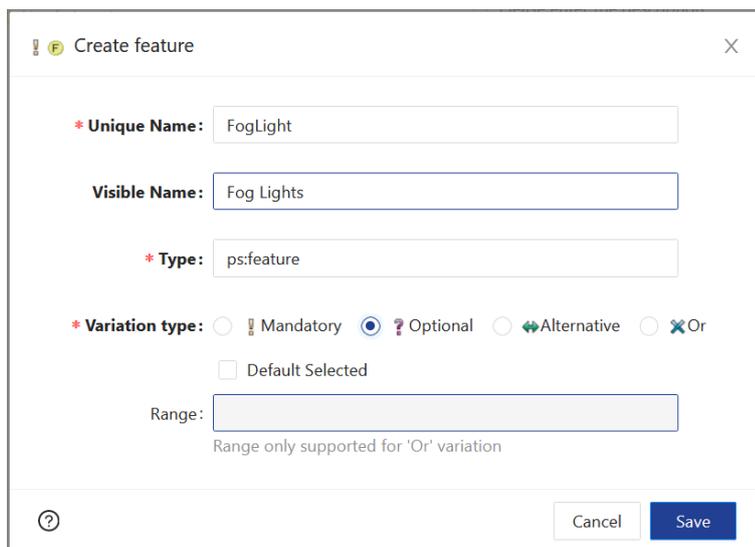


## Create / Edit Element

If users have the rights to change a model, then they can *add and edit model elements via the context menu* of existing model elements in the left area of the Model Page. Different elements will be available to add depending on whether the user opened a Feature Model (*add Feature* ) or a Family Model (*add Component*  / *Part*  / *Source*  ). When the user triggers one of these actions, a dialog will show up wherein they must set a Unique Name, Type and Variation Type for the element. Optionally, the user can set a Visible Name for the element and make the element Default Selected. In case the Variation Type *Or* has been selected, the user can manipulate the range of the *Or* group.

Depending on the selected type of the element, the dialog may be dynamically extended by additional fields. This is the case for the types of *Variant Reference*, *Variant Instance* and *Variant Collection* which can be used to model hierarchical variability.

**Figure 23. Create Feature**



## Variant Collection

The *Variant Collection* is a link in a feature model to another configuration space. The variant models defined in this configuration space are automatically linked. The selections in the linked variant models are locked and unchangeable in the resulting variant hierarchy. When *adding a new variant collection* to a feature model, the user must select a configuration space as well as the collection type. The variant models of the selected configuration space will take over the variation type that has been selected as Collection Type. They will show up as children of the Variant Collection during the configuration of variant models based on the feature model to which the Variant Collection has been added to.

**Figure 24. Variant Collection**

**\* Unique Name:** FogLight

**Visible Name:** Please input visible name

**\* Type:** ps:variantcollection

**\* Variation type:**  Mandatory  Optional  Alternative  Or

Default Selected

Range:

Range only supported for 'Or' variation

**\* Config Space:** Please enter or select a config space

Please select a configuration space

**\* Collection type:**  Mandatory  Optional  Alternative  Or

Restricted To Variants: Start typing to see available variants..

Cancel Save

### Variant Instance

A *Variant Instance* is an element in a feature model that is linked to another configuration space. In a variant model of a configuration space referring this feature model, it is possible to create a new linked variant model below the Variant Instance element which will be based on the linked configuration space.

When *adding a new Variant Instance* to a feature model, the user must select a configuration space. When the user configures a variant based on this feature model, they have the action *Create Instance* available in the context menu of the Variant Instance. This action will add a new optional variant model below the Variant Instance which can be freely configured in the context of its parent variant model.

**Figure 25. Variant Instance**

The screenshot shows a 'Create feature' dialog box with the following fields and options:

- \* Unique Name:** FogLight
- Visible Name:** Please input visible name
- \* Type:** ps:instance
- \* Variation type:**
  - Mandatory
  - Optional
  - Alternative
  - Or
- Default Selected
- Range:** [Empty text box]
- Range only supported for 'Or' variation
- \* Config Space:** Please enter or select a config space (dropdown menu)
- Please select a configuration space

Buttons: ? (help), Cancel, Save

### Variant Reference

A *Variant Reference* is simply a link in a feature model to a concrete variant model of another configuration space. The selections in the linked variant model are locked and cannot be changed in the resulting variant hierarchy. When adding a new Variant Reference to a feature model, a variant model must be selected. Optionally, the Variant Reference can be defined as a *Flexible Variant Reference*.

**Figure 26. Variant Reference**

The screenshot shows a 'Create feature' dialog box with the following fields and options:

- \* Unique Name:** FogLights
- Visible Name:** Fog Lights
- \* Type:** ps:variant
- \* Variation type:**
  - Mandatory
  - Optional
  - Alternative
  - Or
- Default Selected
- Range:** [Empty text box]
- Range only supported for 'Or' variation
- \* Variant Description:** Please enter or select vdm model (dropdown menu)
- Please select a variant
- Flexible Variant Reference

Buttons: ? (help), Cancel, Save

### Create / Edit Attribute

Every model element may have an unlimited number of associated *Attributes*. A new Attribute can be added to an existing feature using the *Create Attribute* action  in the elements context menu. An existing Attribute can be edited using the *Edit Attribute* action  in the Attributes context menu. Both actions open a dialog (see [Figure 27](#),

“[Create Attribute](#)”), wherein the user must enter a unique name and type for the attribute. The user may select a Collection Type other than *None* (default selected) to turn the expected value into a *List* or *Set* of the selected type. If the selected type is a numerical one, the user may restrict the range for valid values by providing a *Range*.

Attributes can be set as fixed by *selecting the option Fixed*. The value of a fixed attribute cannot be changed manually by a user during variant configuration.

**Figure 27. Create Attribute**

## Create / Edit Constant

A *Constant* value can be added to an attribute using the *Create Constant* action  in the context menu of an attribute. Existing Constants can be edited by using the *Edit Constant* action in the context menu of the Constant. Both actions open a dialog (see [Figure 28, “Create Constant”](#)).

**Figure 28. Create Constant**

## Create / Edit Calculation

A *Calculation* can be assigned to an attribute to Calculate its value based on other decisions made during variant configuration. The result of evaluating the Calculation expression is the value of the attribute in a variant. The pure::variants Simple Constraint Language (pvSCL) is used to formulate calculations. For more information on the pvSCL language please consult the pure::variants user manual.

A Calculation can be added to an attribute using the *Create Calculation* action  in the context menu of an attributes. A Calculation can be edited using the *Edit Value* action in the context menu of a Calculation. Both

actions open a dialog (see [Figure 29, “Create Calculation”](#)) with a *pvSCL Code* field . Here, the user can type in the Calculation, supported by auto-complete functionality which can be activated by pressing CTRL+SPACE.

**Figure 29. Create Calculation**

## Create Relation

*Relations* can be defined between an element containing the relation on one side and all other elements of the same project or referenced project on another side.

A new Relation can be added to an existing element using the *Create Relation* action  in the context menu of an element. Existing Relations can be edited using the *Edit Relation* action in the context menu of a Relation. This opens a dialog (see [Figure 30, “Create Relation”](#)), where the user must enter a type and target(s) of the Relation. The user can search for targets by pressing CTRL+SPACE while the cursor is placed in the Targets field to use an auto complete functionality. Instead of selecting one of the relation types supported by pure::variants, the user may also enter their own relation type.

**Figure 30. Create Relation**

## Create / Edit Restriction

*Restrictions* are used to decide if an item of a model is a part of a variant configuration. During model evaluation, an item cannot become part of a variant configuration unless one of the restrictions defined on the item is evaluated as true. Restrictions can be defined on elements, attribute values and relations.

A new Restriction can be added to an existing model item using the *Create Restriction* action  in the context menu of the item. Existing Restrictions can be edited using the *Edit Restriction* action  in the context menu of the restriction. This opens a dialog (see [Figure 31, “Create Restriction”](#)), where the user can enter a name for the Restriction, which is optional . Via the *pvSCL Code* field, the user can formulate the Restriction. By pressing CTRL+SPACE the auto complete functionality can be activated to search for model items and operators of the *pvSCL* language.

**Figure 31. Create Restriction**

## Create Constraint

*Constraints* are used to check the integrity of variant configurations during variant model evaluation. Independent of their location in a model their scope during variant model evaluation is the whole variant model.

A new constraint can be added to an existing element using the *Create Constraint* action  in the context menu of an element. Existing constraints can be edited using the *Edit Constraint* action  in the context menu of the constraint. Both actions open a dialog (see [Figure 32, “Create Constraint”](#)), where users can enter a name for the constraint. Via the *pvSCL Code* field, users can formulate a constraint. By pressing CTRL+SPACE the auto complete functionality can be activated to search for model items and operators of the pvSCL language. If a constraint contains errors the dialog will show a *warning* message below the pvSCL Code field but will still allow the user to save the invalid constraint.

**Figure 32. Create Constraint**

## 5.2. Variant Models

When opening a variant model of a pure::variants project via the *Open* action on the [Project Page](#), the Model Page for variant models opens in a separate browser tab. The Model Page consists of two vertically oriented areas and the Navigation Bar at the top.

The **Model Item Tree** in the left area displays the content of the opened variant model, including model elements (e.g., Features, Components, Parts, ...), Attributes, Values, Relations, Restrictions, Constraints of the underlying feature models and family models as well as the selection state of each model element in the variant model. A drop-down menu with the name of the currently opened variant model is displayed right above the model content. It can be used to *navigate to the other variant models of the same configuration space*. The top right of this area provides buttons to *collapse, expand and search* the content of the variant model.

The **Item Details** area on the right displays the details (e.g., Unique Name, Class/Type, Model, Created, Changed) of the model item that the user has selected in the Model Item Tree area.

The **Navigation Bar** on top of the Model Page shows (from left to right) the pure::variants logo, the project name and revision of the currently opened model, the model name as well as menu entries for the Transformation Dashboard, the Transformation Indicator, the Problem Indicator, the Validation and Evaluation button, the Toggle Automatic Validation and Evaluation button, the Variant Configuration Wizard button, the Create Variant Model / Clone Variant Model button, the Toggle Viewing / Editing Mode button, the User Settings and the Logout button.

Via the **pure::variants logo** in the Navigation Bar the user can *navigate back to the Home Page*.

By clicking on the **project name** the user can *switch to another project and to another revision*. This will navigate the user to the **Project Page** of the selected project in the selected revision.

The user can *switch to a different revision of the current project* without leaving the model page by directly clicking on the **revision name** in the Navigation Bar.

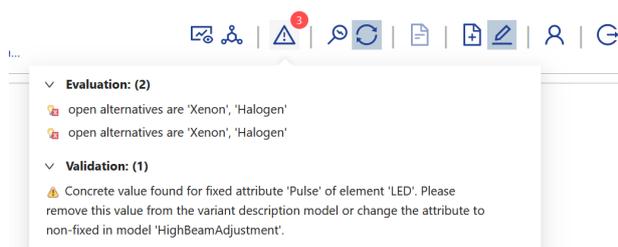
By clicking on the **model name** in the Navigation Bar the user activates a *drop-down model search for the current project* and can directly *switch to a different model of this project*. The model will be opened in the currently active browser tab.

The user can *open the Transformation Dashboard* in a separate browser tab by clicking on its menu entry  in the Navigation Bar.

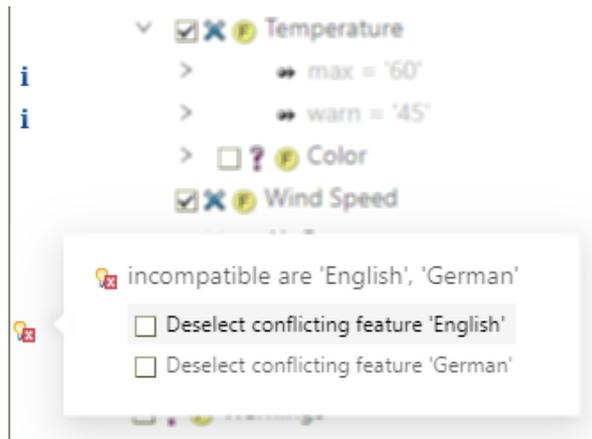
The user can *run a Transformation* for the variant model and *check the transformation state of a running transformation* via the **Transformation Indicator** entry  in the Navigation Bar. Hovering over the Transformation Indicator will provide the status of running transformations in an overlay menu with an additional button to start a new transformation. The transformation log can be opened by clicking on the transformation status.

The user can navigate to all information, warning and error markers of the variant model via the **Problem Indicator** . Hovering over the Problem Indicator will provide a list *with all information, warning, and error markers* of the variant model, in the categories evaluation and validation. Clicking on one of the markers in the list will switch the focus of the Model Item Tree to the model item to which the marker is pointing.

**Figure 33. Problem Indicator with list of error and information markers**



The markers are also directly listed in the Model Item Tree area. The user can assess the suggested quick fixes to resolve an error by clicking on the error marker in the Model Item Tree.

**Figure 34. Error Marker with suggested quick fixes**

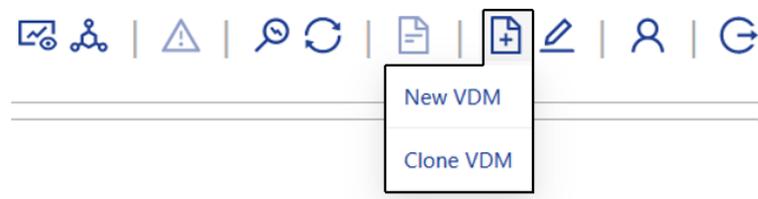
The user can *trigger the evaluation of the variant model* manually by clicking on the **Evaluate** menu entry  in the Navigation Bar.

The user can turn the *automatic evaluation of the variant model* on or off after each configuration step by making use of the **Toggle Automatic Evaluation** menu entry  in the Navigation Bar.

**Figure 35. Evaluation - On-Demand & Auto**

The user can *start the Variant Configuration Wizard* for a guided, step-by-step variant configuration for the variant model by clicking on the Variant Configuration Wizard entry  in the Navigation Bar. This option is only available if the Variant Configuration Wizard has been set up and enabled for the configuration space of the variant model.

The user can *create a new variant model* either from scratch or by *cloning an existing variant model* via the **Create Variant Model / Clone Variant Model** entry  of the Navigation Bar.

**Figure 36. Create or Clone Variant Model in Model Editor directly**

Creating a variant model opens a wizard that asks the user for a variant model name and the destination configuration space of the model. It also provides the optional choice to inherit feature selections and attribute values from an existing variant model as well as the choice to either run a full configuration or a partial configuration. For details on inheritance and partial configurations please consult the pure::variants user manual.

**Figure 37. Create Variant Model Wizard**

New Variant Model

General Destination Inheritance Configuration

Empty strings not permitted for a variant model name.

Name :

Enter Variant Model Name ...

< Back Next > Finish Cancel

The action *Clone Variant Model* can be used to create a variant model and take over all feature selections and attribute values of the currently displayed variant model. In contrast to inheriting from another variant, all feature selections and attribute values can be changed after initially cloning a variant.

In the Clone Variant Model wizard, the user must provide a variant model name and choose a configuration space or a subfolder as the destination for the new variant model.

**Figure 38. Clone Variant Model Wizard**

Clone Variant Model

General Destination

Empty strings not permitted for a variant model name.

Name :

Enter Variant Model Name ...

< Back Next > Finish Cancel

The user can *switch between viewing and editing mode* of the Model Page for Variant Models via the **Toggle Viewing / Editing Mode** entry  in the Navigation Bar.

**Figure 39. Menu entries in the Navigation Bar of the Model View for Variant Models**

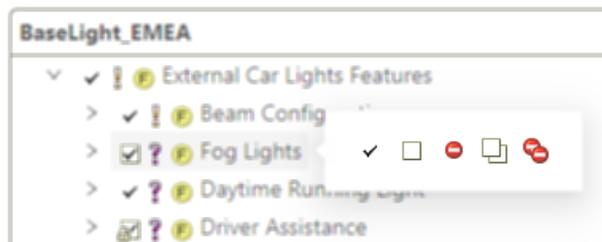
The user can *open the User Settings* by clicking on the respective menu entry  in the Navigation Bar. The User Settings will open in the currently active browser tab.

By clicking on the **Logout** menu entry  in the Navigation Bar the user can *end the Web Client session*. The user will have to log in again via the [Login Page](#) to continue working with the Web Client.

There are different actions which can be performed on variant models via the Model Item Tree of the Model Page which will be explained in the following sub-chapters.

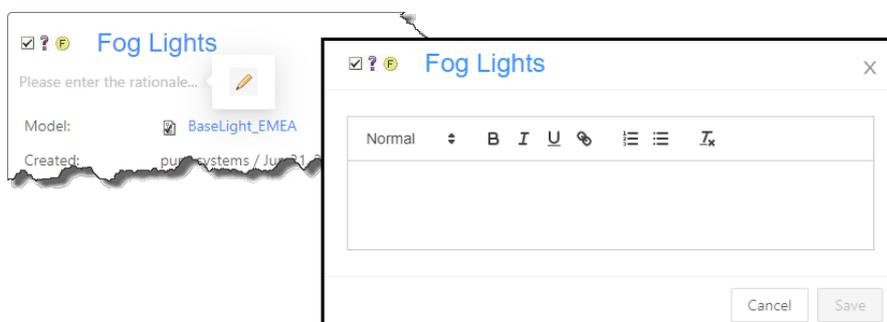
## Change Element Selection State (Variant Model)

In the editing mode of the model page for variant models, the user can *select, unselect, and exclude elements and subtrees* of the variant model via the context menu of an element in the Model Item Tree. Selecting a feature is a typical example of an element selection in the variant model.

**Figure 40. Actions - Selection configuration**

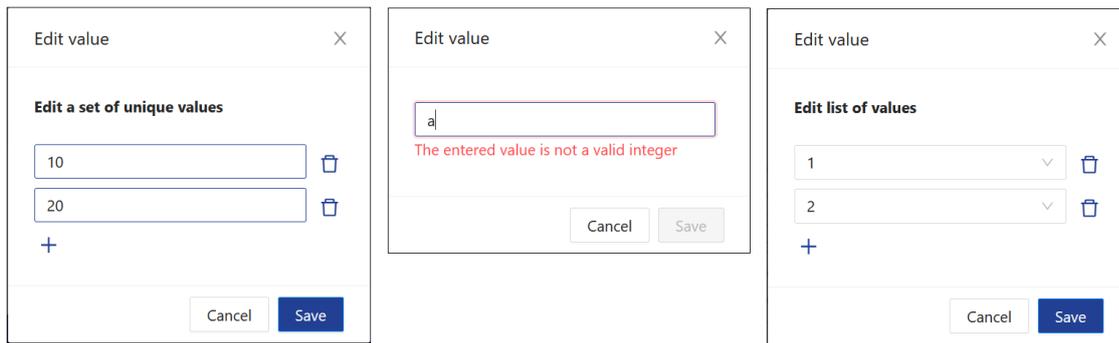
## Edit Rationale (Variant Model)

In the Item Details area, the user can add and edit the rationale for an element selection. This can help other users to understand why an element has been selected in or excluded from a variant.

**Figure 41. Edit the Rationale**

## Edit Attribute Value (Variant Model)

In the editing mode of the model page for variant models, the values of non-fixed attributes can be edited by the user. Depending on the attribute type either one or multiple values can be entered by the user or selected from the drop-down menu of the add value field(s).

**Figure 42. Edit Attribute**

## Set Attribute Value as New Default (Variant Model)

It is possible to set an attribute value as defined by the user in a variant model to be the new default for all inheriting models. This provides increased flexibility when defining inheritance structures.

**Figure 43. Set and Unset a Value as Default**

This property of an attribute can be reset as well. By doing this, the value is considered again as a regular user-set value.

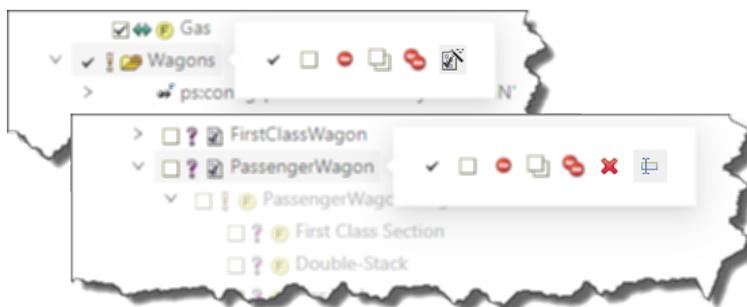
Please note that this functionality is available starting from project version 7.

## Hierarchical Variability (Variant Model)

While in editing mode the user can create, configure, and rename nested variant models that are children of a *Variant Instance* via the context menu of these variant models. The user can add a new nested variant model below a variant instance via the context menu of the variant instance.

Below *Variant Collections* in a variant model all the variants of the linked configuration space will show up as selectable elements with frozen feature selections and attribute values. The user can *select*, *unselect*, and *exclude* these nested variant models from the parent variant model via their context menu.

Variant References show up as frozen and nested variant models in the parent variant model.

**Figure 44. Actions - Variant Instances**

## Create Variant Model Constraint

*Variant Model Constraints* have the same concept as *Constraints* with the difference that their scope is a specific Variant Model.

A new variant model constraint can be added to a variant model using the *Create variant model constraint* action  in the context menu of the root element of the model used and listed first in the variant model. Existing variant model constraints can be edited or deleted using the corresponding actions in the context menu of the constraint.

## 6. Variant Configuration Wizard

In addition to configuring variants via the [Variant Model Page](#) in edit mode, the pure::variants Web Client offers a guided, step-wise configuration of variants via the *Variant Configuration Wizard*.

In the configuration space a Configuration Wizard Setting Model (see [Figure 45](#), “[Open Variant Configuration Configuration Settings Model](#)”) can be added.

**Figure 45. Open Variant Configuration Configuration Settings Model**

### Variant Configuration Wizard Settings Model:

WeatherStationVariantConfigurationModel · Weather Station Example demo · HEAD

If a configuration space is configured to use a Variant Configuration Wizard Setting Model, the Variant Model Page in edit mode enables an additional action named *Open Variant Configuration Wizard* (see [Figure 46](#), “[Open Variant Configuration Wizard](#)”).

**Figure 46. Open Variant Configuration Wizard**



The wizard is divided into two areas. The left area displays a list with all the configuration steps that are part of the guided configuration. The main area to the right is the *Configuration* area. When the Variant Configuration Wizard is started for a variant, the start page of the wizard is displayed in the configuration area (See [Figure 47](#), “[Variant Configuration Wizard: Start Page](#)”).

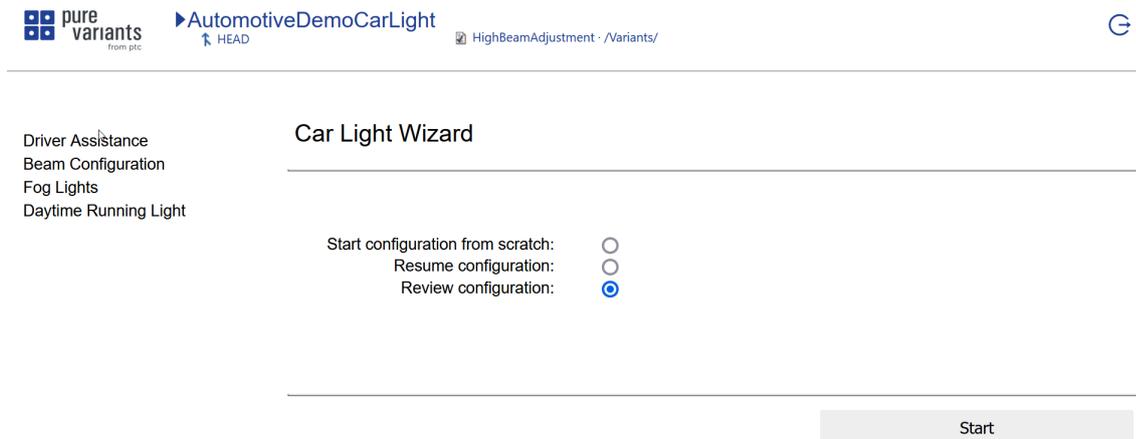
The Navigation Bar displays information about the Variant Model being edited using the wizard, as well as the Project and the Revision in a read only mode.

Depending on the configuration of the Variant Configuration Wizard, the start page lists one or more of the following start-up options.

- **Start configuration from scratch:** Discards all previous selections and resets the variant model to its initial, unconfigured state.
- **Resume configuration:** Resumes the configuration at the point where the user left the configuration the last time.
- **Review configuration:** Allows the user to view the configuration without being able to change any selection. This is the only mode which allows the user to navigate through the pages without changing selections.

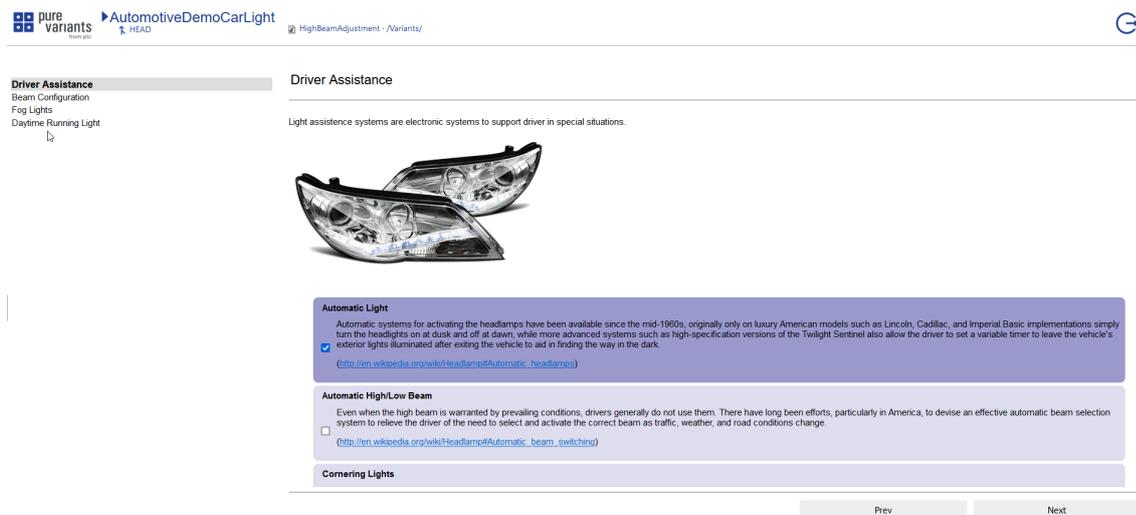
After clicking the Start button, the user is guided through the configuration process step by step. Each configuration step is displayed separately in the Configuration area, listing all the configuration items that are necessary to complete the corresponding configuration step (See [Figure 49](#), “[Variant Configuration Wizard: Finish Page](#)”).

Figure 47. Variant Configuration Wizard: Start Page



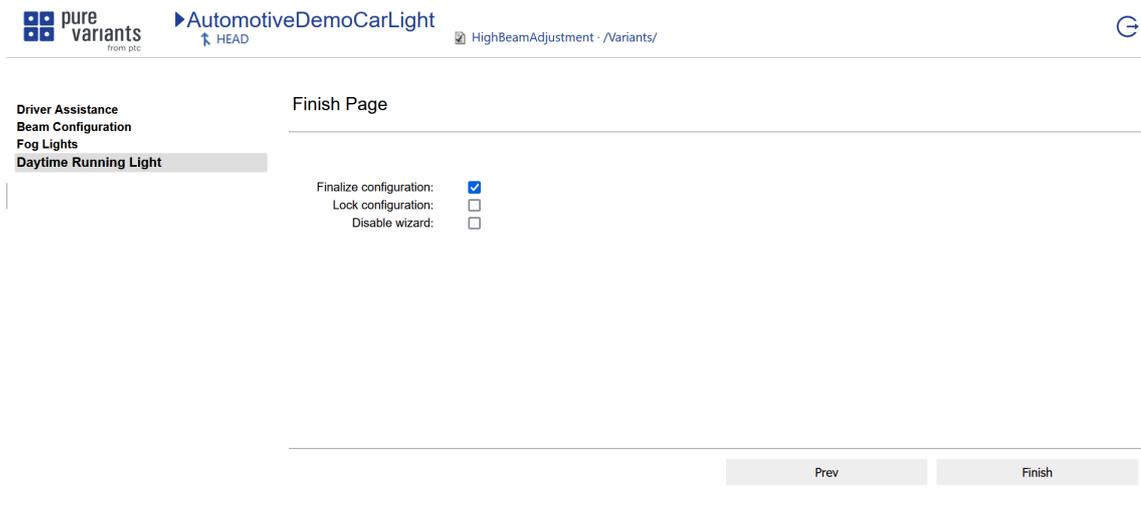
If a configuration item has an associated description, this description is shown below the item. In addition to individual configuration items, a configuration step itself can also have a description. This description is shown at the top of the page.

Figure 48. Variant Configuration Wizard: Step Page



After all configuration steps are done, the *Finish page* is shown (see Figure 49, “Variant Configuration Wizard: Finish Page”). The finish page lists the following options:

- **Finalize configuration:** It automatically converts derived selections and values into user selections and values. The effect of this conversion is that all configuration decisions made in the wizard, even those that were computed by the auto resolver, are treated as if they were made manually by the user. As such, the auto resolver will not change these decisions accidentally if the variant model is reopened later. The only possibility to revise these configuration choices is through explicit user interaction.
- **Lock configuration:** It locks all user selections so they cannot be changed later.
- **Disable wizard:** It disables the wizard for the currently configured Variant Model. This means the wizard is not shown, if the Variant Model is opened again. Pressing the Finish button performs the selected actions and saves the Variant Model. Please note, that this option has no effect in the pure::variants Web Client, but it will be applied when the respective variant model is being opened in the pure::variants Desktop Client.

**Figure 49. Variant Configuration Wizard: Finish Page**

## 7. Variant Matrix

The *Variant Matrix* page enables users to quickly review selections across multiple variants in a *Configuration Space*, change them, and to run transformations. Filtering capabilities support the user when analyzing the variant model configurations.

The *Variant Matrix Page* (see [Figure 50, “Variant Matrix”](#)) can be opened through the *Open Matrix* action (🗒) in the context menu of configuration space and variant model items from [Project Page](#).

After opening the Variant Matrix, the page loads and evaluates all selected variants automatically.

There are three actions available in the top bar of the Model Elements column:

- **Collapse or Expand:** *Collapse* or *Expand* the tree below a selected item. If no item is selected, all trees are expanded/collapsed.
- **Search:** *Search* Based on the search string, model items with matching names are displayed. Selecting an item navigates to it in the item tree.

The Issue indicator in the Navigation Bar provides a quick glance at the number of evaluation problems reported for the selected variant models. Hovering over the Issue indicator opens the Issue Panel, from which a user can navigate to the affected matrix row in Variant Matrix tree table. Here in front of the items name, a marker provides access to the known issues of the selected item and its children. Each variant column has an individual problem indicator which displays the problem count and opens the problem list specifically for the variant model represented by this column upon hovering over it. It also provides navigation to the affected row by selecting the respective problem entry.

The user can *switch between viewing and editing mode* of the Variant Matrix via the **Toggle Viewing / Editing Mode** entry  in the Navigation Bar.

Selections in of variant model can be changed by actions displayed when hovering over a table cell. In case of problems associated with the selection of a cell, a color indicates this, and the action display includes information about the issue(s) and access to quick fixes, if available.

Clicking on the displayed column name opens the Variant Model page for that variant model in a new tab.

**Figure 50. Variant Matrix**

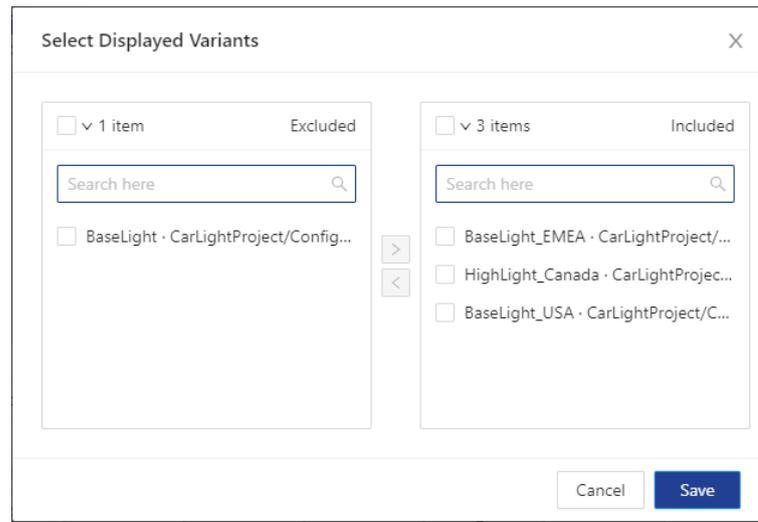
Model Elements	Level	HighBeamAdjustment	BaseLight_USA	HighLight_USA	BaseLight_EMEA
Functional_Features					
External Car Lights Featur...		✓	✓	✓	✓
Beam Configuration	1	☑	✓	✓	✓
Fog Lights	2	☐	☐	☐	☐
Daytime Running Light	3	✓	✓	✓	✓
Driver Assistance	4	☐	☐	☐	☐
Regions					
Regions		✓	✓	✓	✓
EMEA	1	☑	☑	☑	✓
North America	2	☐	☐	☐	☑
CarLight_DefaultWorkingSet					
CarLight_DefaultWorking...		✓	✓	✓	✓

The slide out Matrix Action Side Bar can be opened by clicking on slider icon in the top left box of the matrix (see Figure 51, “Variant Matrix: Side Bar”). It provides several actions such as changing variant displayed in the matrix, enabling or clearing the quick filters through the related actions, it also offers actions to expand or collapse the subtree of a selected variant model item or the full variant model, if no model item is selected, and finally, it offers the action to enable/disable tooltip over the feature model elements inside the Matrix.

**Figure 51. Variant Matrix: Side Bar**

Level	HighBeam	BaseLight_EMEA	BaseLight_USA	HighLight_Canada
tur...	✓	✓	✓	✓
1	✓	✓	✓	✓

The *Select Displayed Variants* action provides a convenient way to change the scope of the Variant Matrix. The respective Selected Displayed Variant dialog covers all variant models defined in the enclosing configuration space. Actions like filtering, evaluation or transformations are always only applied to the variants selected for display.

**Figure 52. Variant Matrix - Select Displayed Variants**

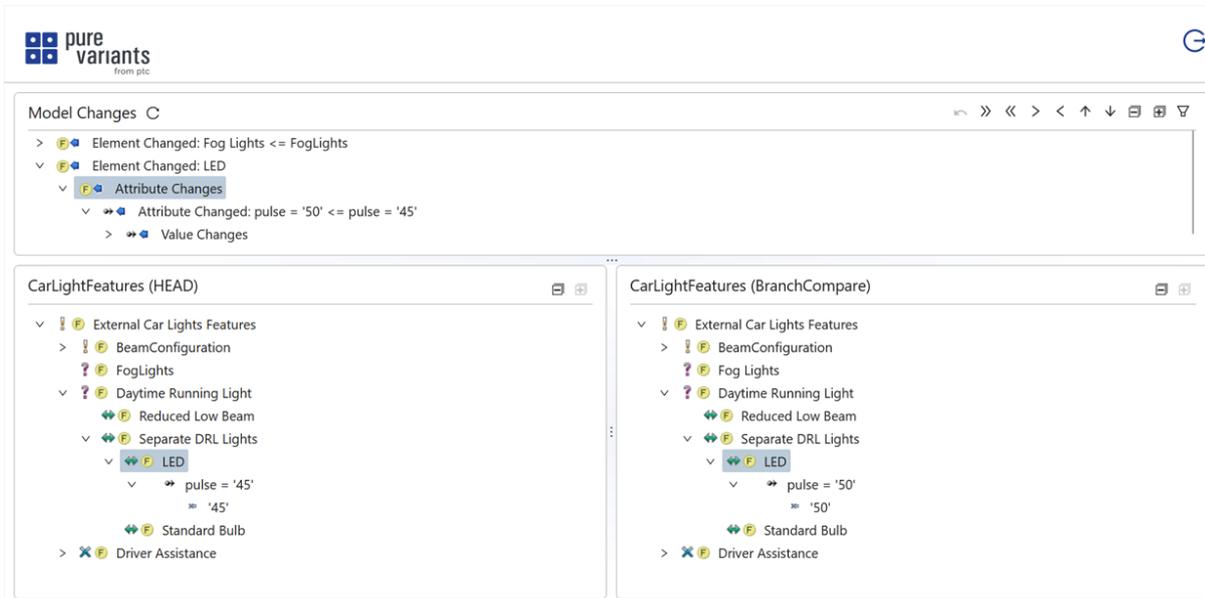
*Quick Filters:* The Variant Matrix can be filtered based on the selection states of features in the individual Variant models by clicking the sidebar on left side of the screen. The supported Quick Filters reduce the displayed rows to all rows which have identical selection status in all models (*Commonalities*), all rows where at least one variant has a different selection state (*Differences*) or all rows where exactly one variant has a different selection state (*Uniqueness*). The *Errors* filter lists all rows which have at least one variant reporting an evaluation problem for the row's model item. Filters are mutually exclusive, i.e., only one of the filters can be activated. The *Clear Filter* action disables all filtering, and all model item rows are being displayed.

Trough the *Transformation* indicator (⚙️) in the Navigation Bar the user gets access to the *Transformation Panel* to start and monitor *multiple transformations* together for the selected Variant Models in a Variant Matrix. See the [Transformation](#) section for a comprehensive description of the Transformation in the Web Client.

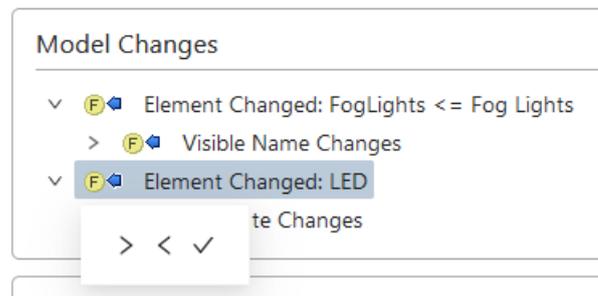
## 8. Compare

The Compare Page has two major layouts. When comparing models, the upper section (**Model Changes**) displays the patch tree listing the differences between the compared models or model revisions in a structured way. The labels for the patch consist of a brief patch description, the label of the patched model item and a concrete visualization of the old and the new value if it makes sense. The lower part displays the model trees in the respective revisions. (See [Figure 53, "Compare of Models"](#))

Differences can be reviewed and managed by selecting a patch in the Model Changes section. Upon selection, the impacted elements are highlighted in the trees below.

**Figure 53. Compare of Models**

The hover over menu provides direct actions to be performed for the selected patch. (See [Figure 54, “Hover Over Menu for Patches”](#))

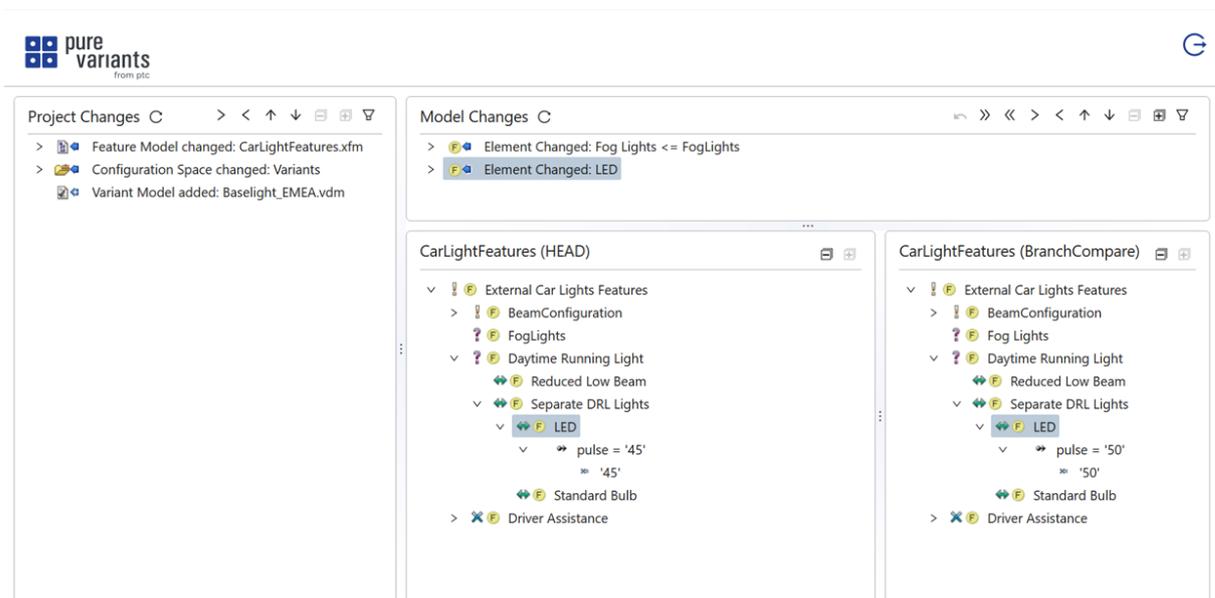
**Figure 54. Hover Over Menu for Patches**

The actions **merge right** (>) and **merge left** (<) allow the user to individually take over a patch either to the right or to the left model. After application, the patch is greyed out in the list of Model Changes. The action **marked as merged** (v) allows the user to accept and thus keep the difference between the two models.

The menu displayed in the header of the Model Changes section additionally offer actions to **merge all** non conflicting patches to the **right** (») or to the **left** (‹). An applied patch can be also **undone** (↶). Undo is available for all model types, except when comparing different variant models with each other. User can skip between the patches using the **next** (↷) and **previous** (↶) actions, while the **expand** (⊞) and **collapse** (⊟) actions can be used to show or hide all sub items of the patches. The **search** (🔍) action supports the User to search for a keyword within the changes. The compare session can be restarted by selecting the **refresh compare** (🔄) action. The refresh action starts a new compare session, as opposed to only closing the tab and reopening the same compare again.

Other layout is used when comparing project models. (See [Figure 55, “Compare of Projects”](#)) In this case, the dedicated **Project Changes** section displays the differences on project level, including removal or addition of models and the list of changed models. When selecting a patch of a changed model, the Model Changes section appears where the user can review and manage the model changes separately. Alternatively, the actions **merge right**, **merge left**, **marked as merged** either through the menu of the Project Changes section or the hover over menu. Similarly to the Model Changes, user can skip between the patches using the **next** and **previous** actions, while the compare session for the project can be restarted by selecting the **refresh compare** action.

Figure 55. Compare of Projects



More information about the generic concept of compare and merge as well as the algorithm is provided in the section **Comparing Models** of the pure::variants User Guide.

## 9. Import Assets as Family Models Using the WebClient

The process of importing assets from connected tools as well as updating family models already imported is started using the pure::variants integration.

The step to select assets and trigger the import or the update is tool specific, and is described in detail in the section **Importing and Updating Models using the Web Client** of the respective pure::variants Connector.

After the **import** is triggered in the integration, the Web Client displays a dedicated wizard to support the next steps.

First, the project, the project revision and the target folder needs to be selected where the family model (or models) shall be created. (See [Figure 56, “Select Target Location for the Family Model”](#)) Models can be stored either in the project opened in the pure::variants integration or in any of its referenced projects. In *Global Configuration Mode* only those referenced projects and revisions are offered that are included in the active Global Configuration context.

**Figure 56. Select Target Location for the Family Model**

Depending on the connected tool, either one separate family model is created for each selected asset or a common one. In later case, the name of the family model can be also specified at this step.

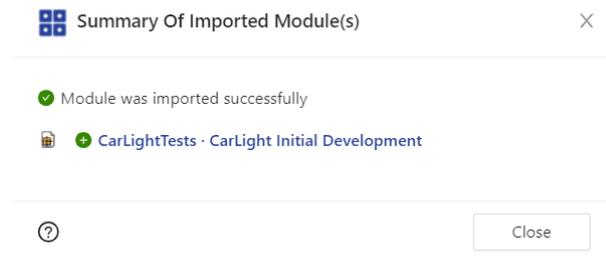
Some connected tools require additional import rules to be defined by the user, for this the wizard offers a dedicated page. (see [Figure 57, “Define Import Rules”](#)) In case multiple assets are imported at the same time, the same rules will be applied to all of them.

**Figure 57. Define Import Rules**

The last page displays a summary of the assets that are going to be included in the new family models.

During the **update** process the import rules can be modified, while the location and the name of the family models can not be changed.

After the import has finished, a summary page (See [Figure 58, “Summary of the Imported Module\(s\)”](#)) is displayed showing the status of the family model (or models) creation.

**Figure 58. Summary of the Imported Module(s)**

## 10. Transformation

The following section describes the interaction with an optional element of a pure::variants Deployment, the pure::variants Transformation Service. This service, if configured, enables users of the Web Client to schedule variant transformations on server side without the need to setup anything locally.

Not all available pure::variants transformation modules are compatible with the pure::variants Transformation Service. Please check the documentation for the pure::variants Connectors and the pure::variants Setup for compatibility information with the Transformation Service and information how to enable the service usage.

Information regarding administrative configuration of the Transformation Service is not in the scope of this document, please see the pure::variants Setup Guide for more information on this topic.

### 10.1. Transformation Services Concepts

The Transformation Services receives transformation request from the Web Client through the user interface actions describe below, or via a REST API (not described in this document). Each request, if accepted, becomes a job. Each job is placed in a centrally managed transformation job queue. The Transformation Coordination Services is responsible for selecting scheduled jobs for execution and for assigning these to Transformation Runners, which are controlling the execution of a single job through a Transformation Executor, running in a containerized execution environment provided by Docker. Multiple Transformation Runners can be provided, enabling parallel / concurrent processing of multiple transformation jobs of different Web Client users.

Transformation Executor report their progress and status through the Transformation Runner back to the Transformation Coordination Service.

The lifecycle of each transformation job has three different states: *Waiting*, *Running*, and *Done*. Waiting jobs have not yet been assigned to a Runner. As soon as a job is assigned to a Transformation Runner, it changes into the *Running* state. Once the transformation is done or was stopped (cancelled by a user or aborted due to errors), the state is *Done*. The result information of a transformation provides information about the transition reason into the Done state (CANCELLED, ABORTED, SUCCESS, FAILED).

### 10.2. Transformation Panel

The Transformation Panel provides an easy user interface to the Transformation Service. It can be accessed through the Transformation indicator (🔗) on the Navigation Bar of each Variant Model or Variant Matrix page. It provides details of the currently and previously scheduled **transformation** jobs for the active Variant Model / Variant Matrix. It also serves as management interface for creating and managing these jobs. Please note, this Panel only shows the details of the transformations which are triggered from the same page.

The panel provides two operations on the complete list of transformation jobs. The **Delete Done** action discards the execution information for all finished transformations listed in the panel. Please note, that discarded execution information cannot be recovered after deletion.

Additionally, cancellation of the execution of all waiting and running transformation jobs list in this panel can be requested by the **Cancel All** action. Cancellation of a job may not happen immediately.

The same actions can be executed individually for each job through the action buttons shown at the end of each job entry in the panel. The displayed action changes based on the jobs state from Cancel to Delete once the job reaches the Done state.

A detailed log for a job can be accessed from the panel by clicking on the respective job entry in the panel, presented in the **Transformation Details Dialog**.

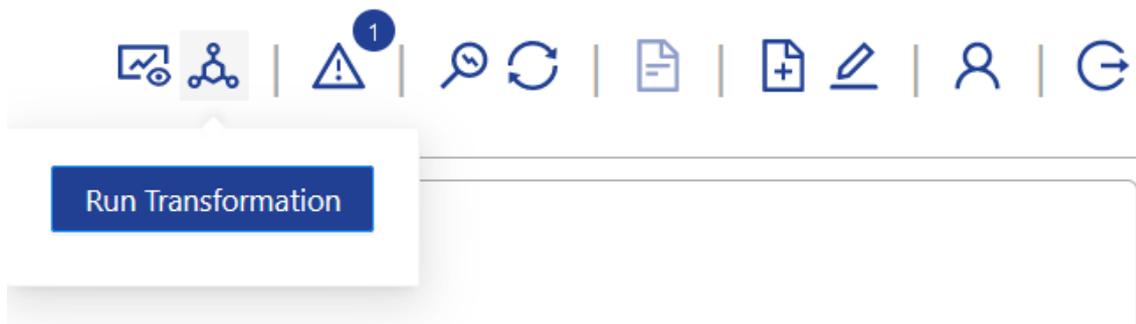
**Figure 59. Transformation Panel**



## Run Transformation (Variant Model)

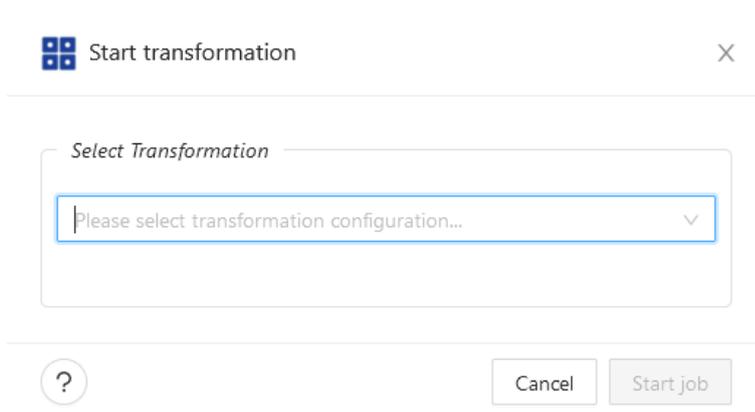
A transformation for the specific Variant Model can be triggered by opening the Transformation panel using the Transformation indicator (🔗) in the Navigation Bar and clicking on *Run Transformation* button on the Variant Model page (see [Figure 60, “Run Transformation”](#)).

**Figure 60. Run Transformation**



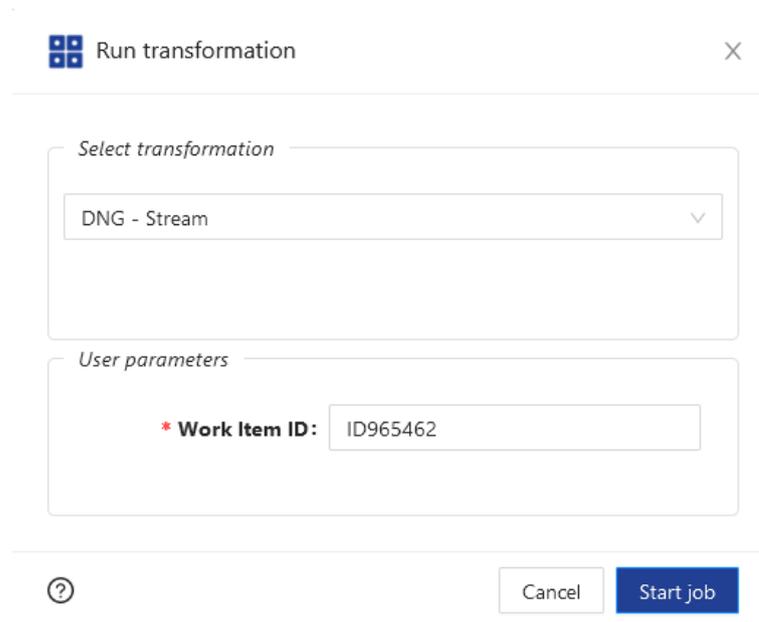
The *Run Transformation* dialog (see [Figure 61, “Create Transformation”](#)) provides the list of available transformation configurations. The transformation request is scheduled by clicking the "Start job" button after selecting the transformation. The definition of transformation configurations can be done using the pure::variants Desktop Client side. Please note, that the presented list may also contain transformation configurations not suitable for server-side execution. It is recommended to use an appropriate naming scheme for transformation configurations to guide the user when selecting transformation configurations for execution.

**Figure 61. Create Transformation**



If in case, there are any user-defined parameters configured for any of the available transformation configurations then dialog also provides the capability to set the value for those parameters (see [Figure 62, “Create Transformation with User Parameters”](#)). In case a parameter value is defined as mandatory and no value is set, an appropriate error validation message will be shown. Only after setting all required values for the user parameters it is possible to launch transformation job.

**Figure 62. Create Transformation with User Parameters**



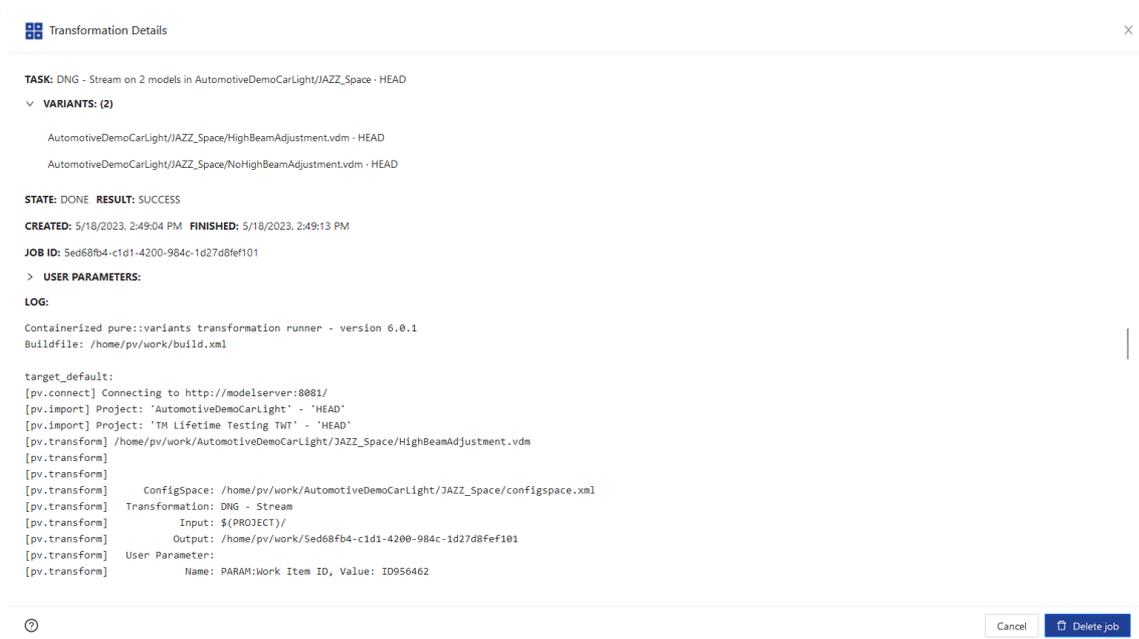
The screenshot shows a dialog box titled "Run transformation" with a close button (X) in the top right corner. The dialog is divided into two sections. The first section, "Select transformation", contains a dropdown menu with "DNG - Stream" selected. The second section, "User parameters", contains a field labeled "\* Work Item ID:" with the value "ID965462". At the bottom of the dialog, there is a help icon (question mark), a "Cancel" button, and a "Start job" button.

User parameters can be configured using the pure::variants Enterprise Desktop Client.

In case any of the Variant Models subject to transformation is invalid, an error message is shown and the transformation can not be started unless the user decides so by setting the checkbox "Transform despite invalid variant".

## Transformation Details Dialog

The Transformation Details Dialog provides basic information about the currently selected job as well as the complete transformation log. Further, user can perform job related actions, i.e. cancel a running job or delete the job once it is finished.

**Figure 63. Transformation Details Dialog**

The Transformation Details Dialog can be accessed from the Transformation Panel, from Transformation Dashboard and also from the Recent Transformations list on the Home Page.

## 10.3. Transformation Dashboard

The *Transformation Dashboard* provides a comprehensive management and monitoring interface for the Transformation Service. All the content in the Transformation Dashboard updates in real time. Users without the Transformation Administrator role have access to all transformation jobs they have started. Users with the special role *Transformation Administrator* can see all jobs known to the Transformation Coordination Service and have in addition the right to start, pause or disable the service globally. The Transformation Dashboard can be accessed from the Navigation Bar action () , which opens the dashboard in a new tab.

The Transformation Dashboard has two main areas (see [Figure 65, “Transformation Dashboard”](#)). The lower Table area shows either the list of Transformation Jobs or the Transformation Runner list. The content of these tables updates automatically via a live updating service.

- *Transformation Job Table* shows the details of each running, waiting and done jobs. It provides filters to limit the displayed information. Upon selection of table rows *Delete* and *Cancel* actions can be executed on the selected jobs. Cancelling a job can only be done on jobs in the Waiting or Running state. Deletion is only applicable to jobs in the Done state.

In case of multiple Variant Models were transformed in the scope of the same job, hovering over the variant column entry indicating the count of Variant Models a popup window will list the Variant Models (see [Figure 64, “List of Multiple Variant Models Transformed”](#)).

- *Transformation Runner Table* shows the list of all configured runners, independent of their activity status.

**Figure 64. List of Multiple Variant Models Transformed**

Job	State	Variant	Transformation	Config Space
5ed68f...	✔	HighBea... 2	DNG - Stream	JAZZ_Space
2cc6e9...	✔			CB_Space
50f629...	✔			CB_Space
8848b...	✔	HighBea...	WorkingSet	CB_Space

The upper Status area also shows live data received from the Transformation Coordination Service. It is divided in 3 boxed sections:

- **Service:** This box shows the current operational status of the service and provides controls to change the service status. Please note, pausing or disabling service will let already running transformations complete proceed to the Done state. In the Paused state, the service will still accept job requests, but will not schedule them for execution. In Disabled state, the service will reject transformation requests. If an immediate stopping of running jobs is required, use the provide Cancel action for the jobs in the Transformation Job table.
- **Jobs:** This box shows statistical data for transformation jobs known to the service. It also serves as selector for the lower table area. Clicking into the box displays the Transformation Job table in the lower area.
- **Runners:** This box shows the current activity status of the configured transformation runners. Active runners are currently executing transformation jobs, Idle runners are ready accept jobs for execution, and Offline runners are configured but are currently not connected to the Transformation Coordination service. It also serves as selector for the lower table area. Clicking into the box displays the Transformation Runner table in the lower area.

**Figure 65. Transformation Dashboard**

Job	State	Variant	Transformation	Config Space	Project	Revision	User	Result	Created	Finished	Duration	Waiting	Runner
f769b62...	✔	HighBeamAdju...	DNG - Stream	JAZZ_Space	Automotive...	HEAD	jazz	SUCCESS	3/8/2023, 3:32:10 PM	3/8/2023, 3:33:39 PM	00h : 01m : 29s	00h : 00m : 41s	runner-1
9e69e84-3...	✔	NoHighBeamA...	Create VRM	JAZZ_Space	Automotive...	HEAD	jazz	SUCCESS	3/8/2023, 3:31:55 PM	3/8/2023, 3:32:51 PM	00h : 00m : 56s	00h : 00m : 42s	:
5fd783d-d...	✔	NoHighBeamA...	DNG - Stream	JAZZ_Space	Automotive...	HEAD	jazz	SUCCESS	3/8/2023, 3:31:46 PM	3/8/2023, 3:32:37 PM	00h : 00m : 51s	00h : 00m : 00s	:
68805894-...	✘	BaseLight_USA...	Enum	CB_Space	Automotive...	HEAD	jazz	FAILED	3/8/2023, 3:29:44 PM	3/8/2023, 3:29:58 PM	00h : 00m : 14s	00h : 00m : 00s	:
2aaa7dc6-...	✔	HighBeamAdju...	DNG - Stream	JAZZ_Space	Automotive...	HEAD	jazz	SUCCESS	3/8/2023, 2:45:14 PM	3/8/2023, 2:46:05 PM	00h : 00m : 51s	00h : 00m : 01s	:
edc1bca-c...	✘	HighBeam.vdm	WorkingSet	CB_Space	Automotive...	HEAD	jazz	FAILED	3/8/2023, 2:44:08 PM	3/8/2023, 2:44:23 PM	00h : 00m : 15s	00h : 00m : 01s	:

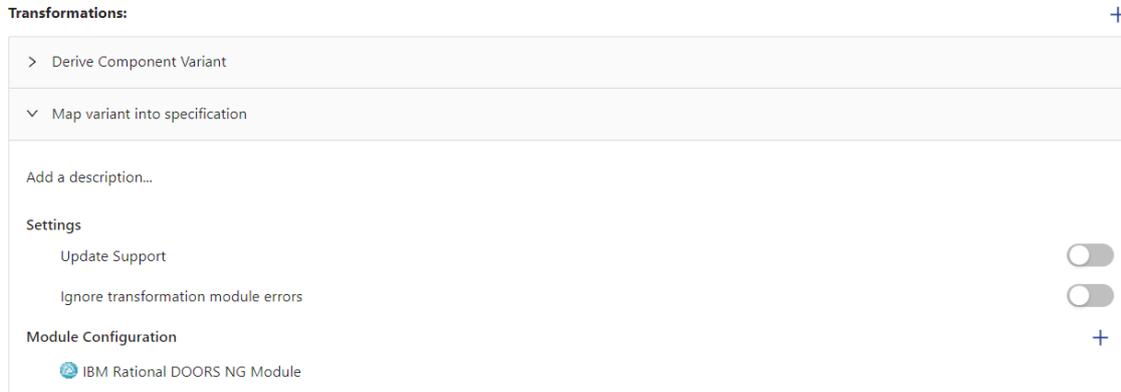
## 10.4. Transformation Configurations

In the Configuration Space a number of Transformation Configurations can be defined. Use the Add button (+) in the Transformations section to create a new **Transformation Configuration** (see [Figure 66, “Transformation Configuration”](#)).

If the **Update Support** button is toggled, the created output of transformation modules for a given variant has to support variant update scenario. In that case an already existing output for this variant may not be overwritten while transformation but can be updated afterwards with the newly created output.

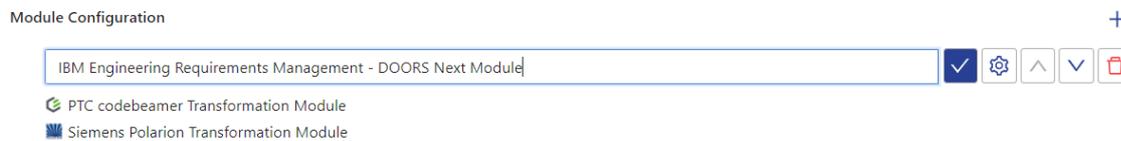
If the **Ignore transformation module errors** button is toggled, errors reported by transformation modules do not cause the current transformation to be aborted. Use this option with caution, it may lead to invalid transformation results.

**Figure 66. Transformation Configuration**

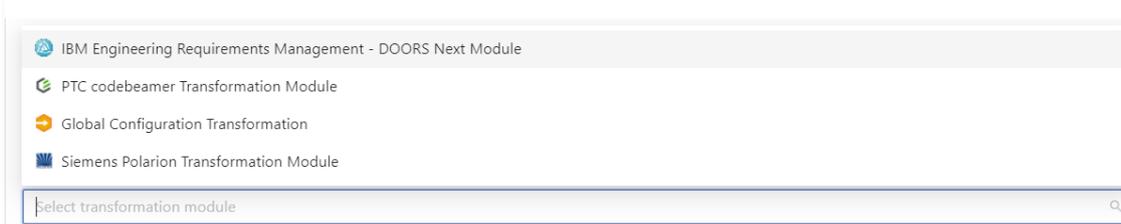


A **Module Configuration** (see [Figure 67, “Transformation Module Configuration”](#)) consists of a list of configured modules. A configured module can be selected by clicking the Add button (see [Figure 68, “Selection of Configured Modules”](#)). Since many modules have dependencies on other modules they must be executed in a specific order. The order of execution of the transformation modules is specified by the order in the Configured Modules list and by the kind of modules. This order in the list can be changed using the Up and Down buttons.

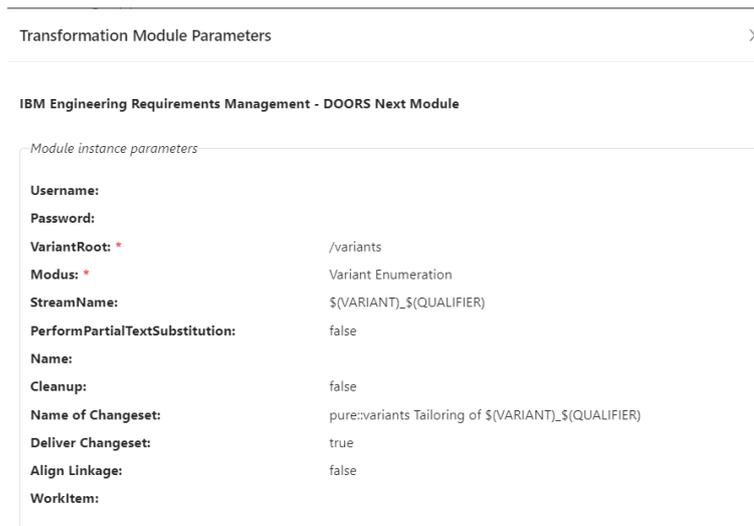
**Figure 67. Transformation Module Configuration**



**Figure 68. Selection of Configured Modules**



The (⚙️) button opens a "Transformation Module Parameters" dialog (see [Figure 69, “Transformation Module Parameters”](#)) showing the **module parameters**.

**Figure 69. Transformation Module Parameters**

## 11. User Settings

The *User Settings* page provides information about the current user such as the assigned roles; enables the user to define some user specific preferences for the operation of the Web Client; and, if not using an external authentication service, allows users to change their pure::variants password.

The *Profile* section displays read-only administrative information for the current logged in user. The Roles list all pure::variants roles for the user. Based on these roles, pure::variants calculates the effective permissions for each project and model item.

The *Password* section provides a self-service for changing the user password on the connected pure::variants Model Server if the authentication is not an external service such as a Single-Sign-On service or LDAP. With an external authentication service, this section is disabled, and passwords must be changed through the external service.

The *Preferences* section lets users define certain settings, which modify the behavior of the Web Client for this user. These preferences are stored locally in browser, they are not stored on server side. I.e., a user can define different preferences in different browsers.

Edit Mode	Defines whether a new Variant Model page or the Variant Matrix is opened in Read mode (default) or in Edit mode. For users with <i>Reader</i> role, this setting is not relevant.
Auto Validate and Evaluate	Defines how a new Model page handles model validation in case of Feature or Family Models (or validation <i>and</i> evaluation in case of a Variant Model). By default, the user must manually request a validation or a validation and evaluation respectively. Selecting this setting enables immediate validation or validation and evaluation when opening the model or after each change ( <i>Auto Validate and Evaluate</i> mode).
Label	Changes how model elements are being displayed in the model pages. Different choices allow a user to adjust the display model item labels according to the personal needs. The default setting is to display the Visible Name with a fall back to the Unique Name or Element id. Other labelling strategies provide different combinations of these three parts of element identification such as the combined display of Visible Name and Unique name.
Matrix Feature Tooltip	Define whether or not to activate the tooltip over the feature model elements on the Variant Matrix page. By default this setting is activated.

Model Language

Set the language of the models to be displayed or edited. By default this setting is *'Unspecified'*.

**Figure 70. User Settings: Preferences**

The screenshot shows the 'User Settings: Preferences' interface. On the left is a sidebar with a 'Collapse' button and menu items for 'Profile', 'Password', 'Preferences' (highlighted), and 'About'. The main area is titled 'User Settings' and 'Preferences >'. It lists several settings:

- Edit Mode**: Includes a checkbox for 'Open VDM or Matrix in Edit Mode'.
- Auto Validate and Evaluate**: Includes a checked checkbox for 'Enable Auto Validate and Evaluate'.
- Label**: Includes a dropdown menu currently set to 'Visible Name'.
- Matrix Feature Tooltip**: Includes a checked checkbox for 'Enable Feature Tooltip'.
- Model Language**: Includes a dropdown menu currently set to 'English'.

The *About* section list version information of pure::variants components and services.

## 12. System Administration

The *System Administration* page provides the option for Administrators to manage users and roles.

The *Users* section lists all users defined in the system with search and filtering possibility. When selecting a user from the list, the details of the users are displayed on the right and can be edited. In the details section users can be also disabled or enabled, as well as deleted (see [Figure 71, “System Administration: Manage Users”](#)).

**Figure 71. System Administration: Manage Users**



An initial password can be set that is to be used for authentication in case the Web Client is configured this way ([Login Page](#)).

In addition to this, permission roles can be assigned to the user that are used to manage access rights to model items. Similarly, functional roles can be enabled for a user (see [Access Management](#)). Creation of a new user is possible using the dedicated button.

The *Roles* section lists all roles defined in the system. When selecting a role from the list, the details of the role are displayed on the right and can be edited. A role can be also disabled or enabled, as well as deleted.

## Figure 72. System Administration: Manage Roles

Both permission and functional roles can be defined here, and users can be directly assigned to them. Creation of a new role is possible using the dedicated button.

## 13. Model Picker

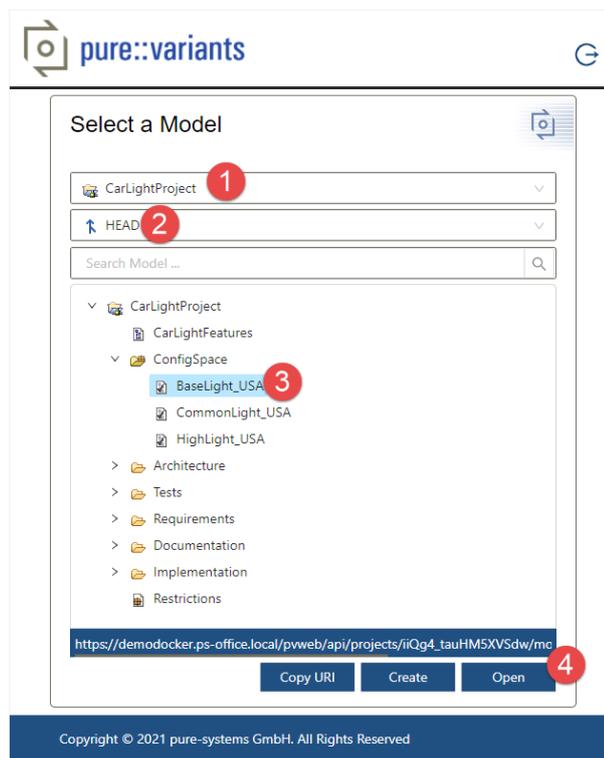
The Model Picker provides a simplified alternative to the Web Client to access to pure::variants projects and models using a web browser, including the option to use functionality or to navigate to specific pages of the Web Client.

In order to open the Model Picker in the browser, please point your browser to the following link:

```
https://[pv-server-FQDN]:[port]/pv/ui/picker
```

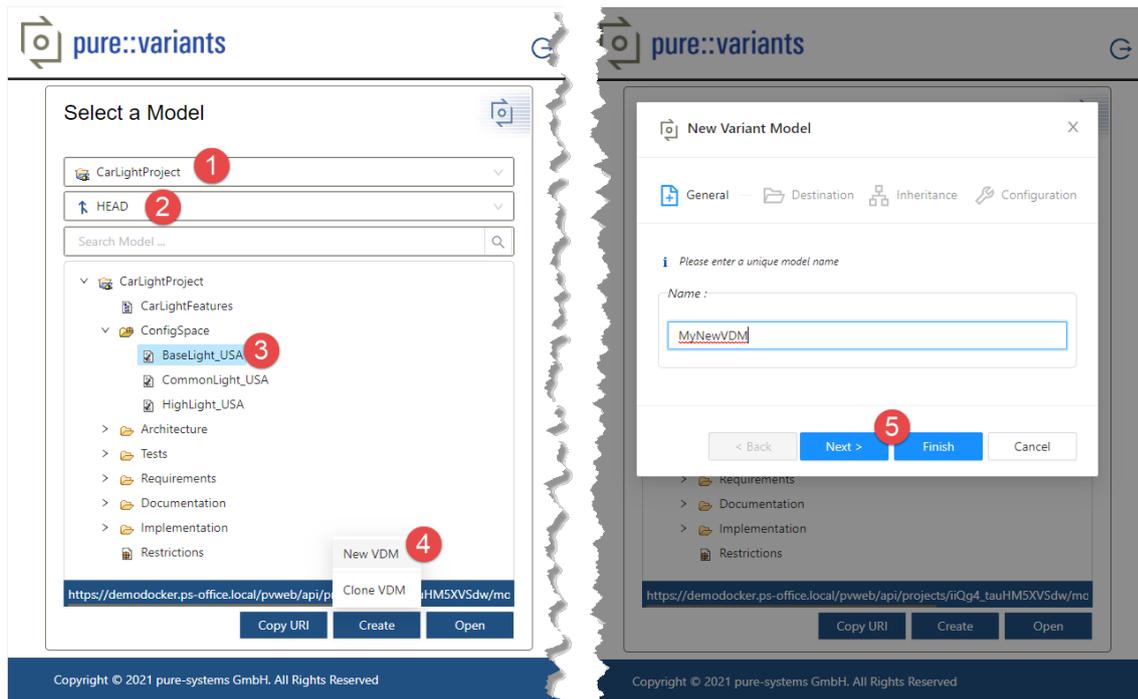
A login page might be shown if you are not already authenticated. Please enter your user credentials to continue. After successful login, the Model-Picker is shown (see [Figure 73, “Model Picker”](#)).

Figure 73. Model Picker



To choose a pure::variants model, you need to select a **Project** from the first drop-down menu, first. Then, you need to specify the project context, by selecting a **Revision** from the second drop-down menu. Thus the project explorer shows the whole project's content, including pure::variant models, which might be nested in a hierarchical directory structure. Finally, you have to select a **Model** and press **Open** to view the selected model in the *Model Page*.

For *Variant Description Models* and *Configuration Spaces* you can use the *Create* button, for creating a new VDM, cloning an existing VDM, or create a new Folder inside of a *Configuration Space*.

**Figure 74. Model Picker - Create new Variant Description Model**

The wizard will guide you through the steps of specify the Variant Description Model. You can choose a different destination folder, you can define other Variant Description Models to inherit from, or decide if this is considered as full or partial configuration. The *Create* toolbar button is context-sensitive to your selection. E.g. selecting a *Configspace* will allow you to create a new Variant Description Model or a new folder.

## 14. OSLC Services

pure::variants offers services to support the integration with other tools implementing the OSLC (Open Services for Lifecycle Collaboration) standard. The following chapters describe the details about these services.

### 14.1. Using the OSLC Configuration Management Service

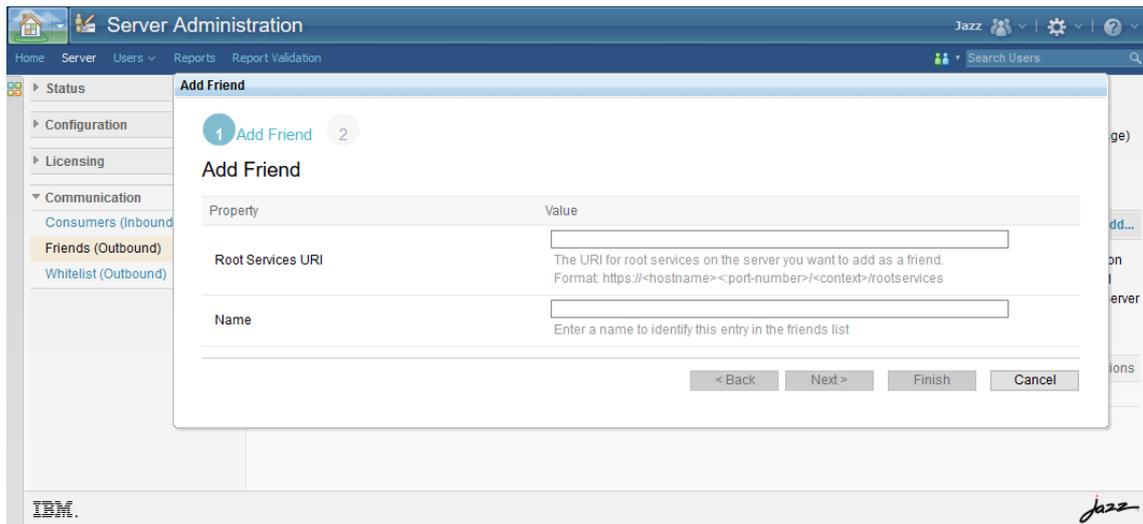
The *OSLC Configuration Management Service* adds the capability to integrate the pure::variants projects into a global context, defined by a GC's configuration, like a stream or baseline. The *OSLC Configuration Management Service* implements the OSLC Configuration Management provider capability. Therefore, the user can browse over all revisions of its pure::variants projects, residing on the pure::variants Model Server.

#### Note

The pure::variants Revision is like a Configuration in the terminology of IBM Jazz CLM. The Revision can be either a Branch or a Version. Appropriately to the terminology of IBM Jazz, the Branch is semantically equal to a Stream, and the Version is semantically equal to a Baseline.

#### Prerequisites

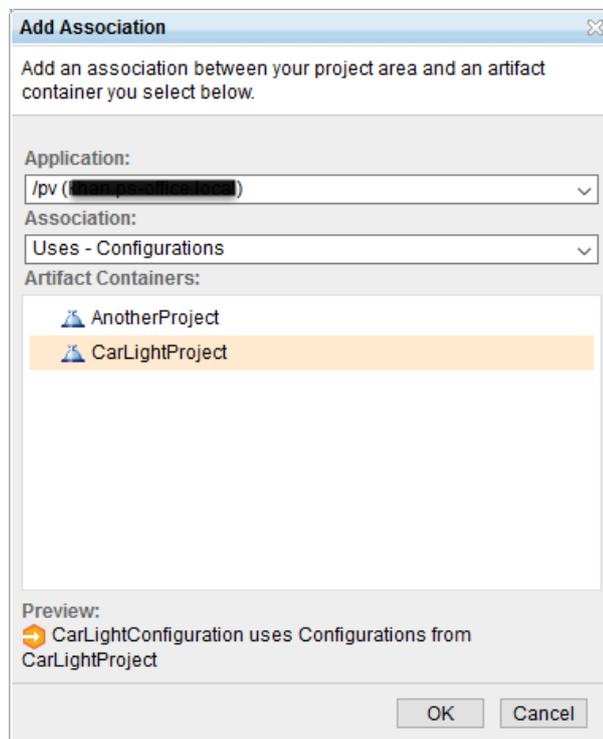
In order to use the pure::variants CM Provider with GC, it is required to have a Friends (Outbound) relationship from Jazz to pure::variants. For details, please see [Figure 75, “Jazz - Add a friend \(Outbound\)”](#).

**Figure 75. Jazz - Add a friend (Outbound)**

## Association of pure::variants project to GCM project

The project association between GCM and pure::variants is only required if your GCM application setup enforces it. For more information, please refer to [Limiting GCM contributions by enforcing project area associations](#).

To associate a pure::variants project with a GCM project, please go to the GCM application administration page at <https://hostname:port/gc/admin>, and click **Project Areas** > **Active Project Areas** on the banner and click on the particular project. Please navigate to *Associations* and press *Add...* Now, choose the pure::variants as an option and select one of your pure::variants projects.

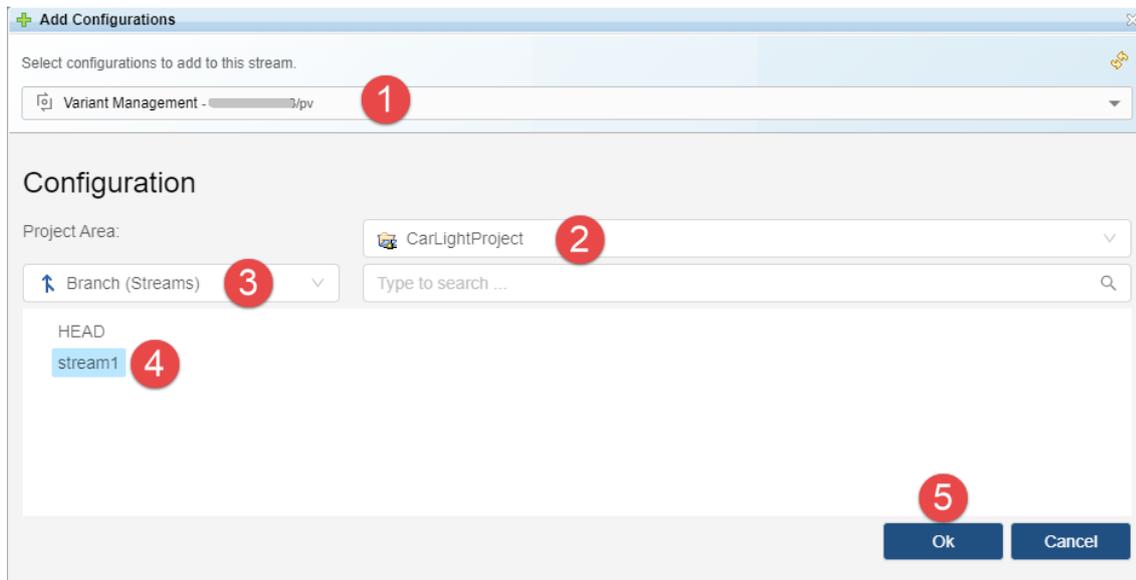
**Figure 76. Jazz - Associate a pure::variants project**

Please make sure to save your association, by pressing *Save* in the project area.

## Association of pure::variants configuration to GCM configuration

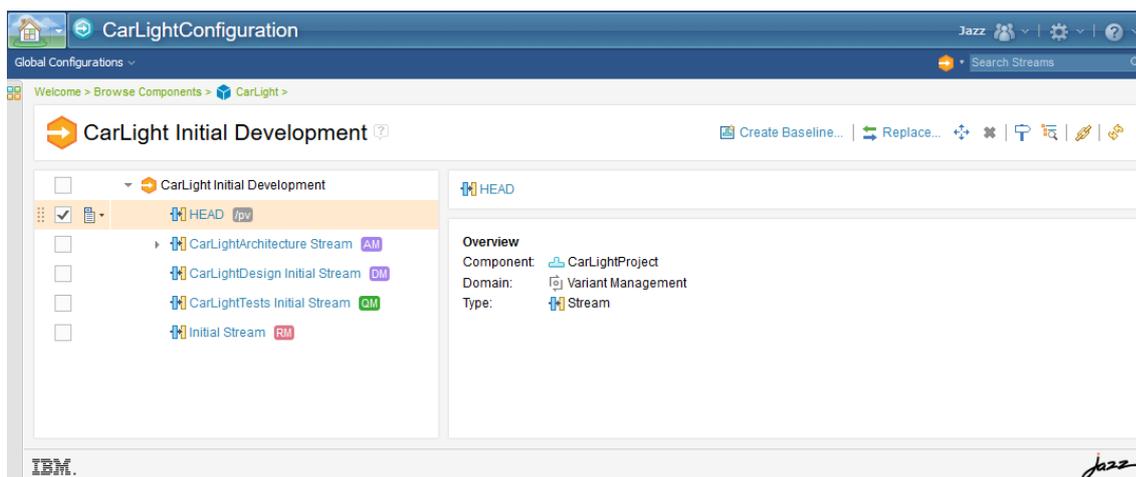
To associate a pure::variants revision, you need to navigate to a GCM configuration, first. Now, you can add the pure::variants revision as contribution to the GCM configuration, by pressing **Add Configuration** (+ Add Configurations...). The configuration selector dialog should show up, so you can choose **Variant Management** to get the selector dialog of pure::variants configurations (see [Figure 77, "GCM - Configuration Selection Dialog"](#)).

**Figure 77. GCM - Configuration Selection Dialog**



As a result, the selected pure::variants configuration should show up in the GCM configuration.

**Figure 78. GCM - Configuration with the pure::variants configuration "HEAD"**



## Create a baseline of pure::variants configuration inside of GCM configuration

To create a baseline (or a version) of a pure::variants stream (or branch), you need to select the particular pure::variants stream and press **Create Baseline** (Create Baseline...). In the dialog you can define any suffix for the baseline naming, see [Figure 79, "GCM - Create Baseline Dialog"](#). Additionally, you can decide to replace the stream with the then created baseline.

**Figure 79. GCM - Create Baseline Dialog**

## 14.2. Add traceability links to OSLC-capable Tools

The pure::variants features and models can be linked to the OSLC-capable tools like Jazz's RM and QM.

### Prerequisites

In order to add pure::variant's traceability links in Jazz RM, the steps explained in [Section 14.1, "Using the OSLC Configuration Management Service"](#) needs to be completed first. The last step [the section called "Create a baseline of pure::variants configuration inside of GCM configuration"](#) is optional and can be skipped.

### Association of pure::variants project in RM project area

Navigate to the **Application Administration** of the Jazz RM application. Navigate further to the project area management page of a RM project area. In the **Associations** section, click the **Add** link to open the **Add Association** dialog as shown in the figure [Figure 80, "RM - Add Association"](#).

**Figure 80. RM - Add Association**

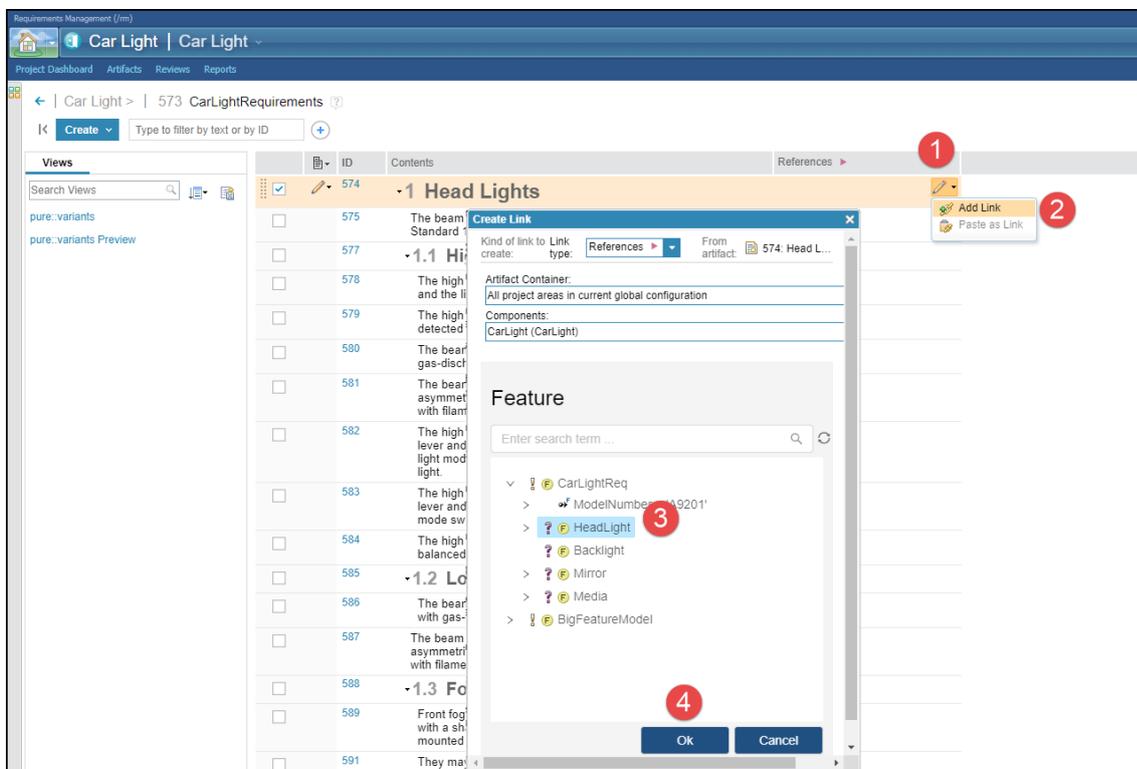
Select **PV** from the **Application** drop down list. In the **Association** drop down list, select **Provides - Related Requirements** if not already selected. Select a pure::variants project from the **Artifact Containers** list and press **OK**. Click **Save** button on the top right corner of the page to save the **Project Area** settings. Click the **Explore Project** to open the project area.

## Adding the pure::variants Links in RM

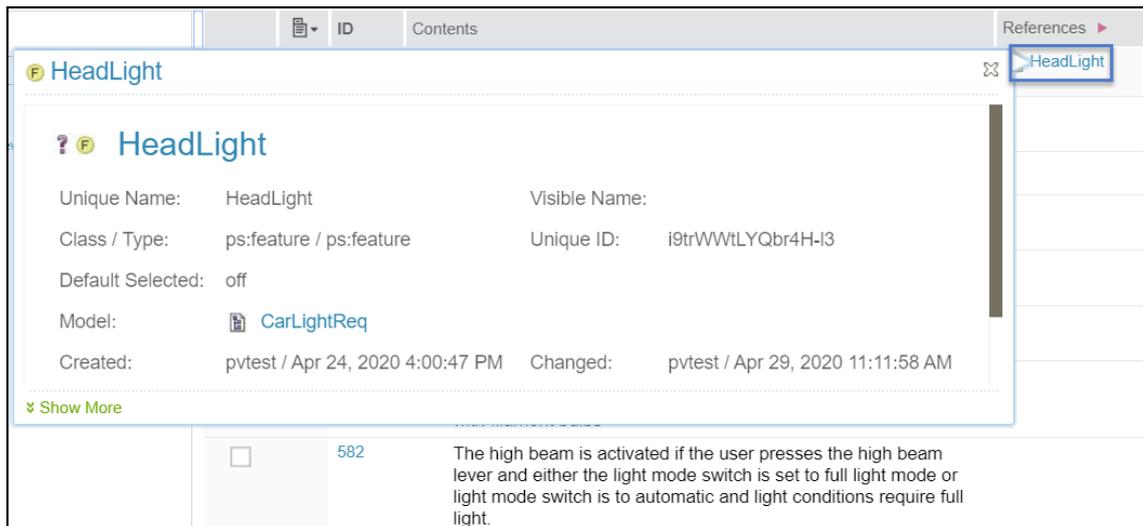
Navigate to the RM project area with which you have associated the pure::variants project. Navigate to the Artifacts page and then open an RM **Module**. From the top right corner of the RM page, in the **Configuration Context**, make sure that the correct GC Configuration is selected in which you have added the pure::variants configuration.

In the RM Module, configure the References column to be visible in the list view. In a Requirement, in the **References** column, click the **Edit** icon and click **Add Link** as shown in the figure [Figure 81, “RM - Add pure::variants Link”](#).

**Figure 81. RM - Add pure::variants Link**



A **Create Link** dialog appears, select a feature and click **OK**. The dialog will close and the pure::variants feature's name will appear in the **References** column. If the pure::variants feature's name does not appear automatically then refresh the whole RM page. Hover over the link title in the References column in order to see the pure::variants artifact's preview as shown in the figure [Figure 82, “RM, Preview of the Linked Artifact”](#).

**Figure 82. RM, Preview of the Linked Artifact**

## 15. Known Limitations of the pure::variants Web Client

1. Limitations when working in *Global Configuration* mode:
  - a. Component hierarchies in Global Configurations are not supported, i.e., it is not allowed to have the same project with different versions in the same Global Configuration at the same time, upon opening such model Web Client will report an error. The Global Configuration needs to be corrected.
2. Changed access rights and role assignments for users are only considered in a new Web Client session. User needs to log out and log in again to the Web Client so that changes in access rights or roles take effect.
3. When adding feature or family models to or removing them from a configuration space, the user needs to refresh the pages with opened VDM-s so that the changes take effect.
4. After creation of a new VDM outside of a specific Variant Model Page, the page must be reloaded to show the updated list of variant models which includes the new variant model in the Variant Selector.
5. Not supported actions (have to be performed in the pure::variants Desktop Client):
  - a. Creation of family models except those created by importing assets from supported connected tools, type models, variant configuration wizard models
  - b. Viewing and changing of type models
  - c. Viewing and editing of variant configuration wizard models
  - d. Editing properties of a configuration space not relevant to supported transformations
  - e. Changing role assignments for users and access rights for project and model items
  - f. Changing the inheritance relationship or configuration mode of a variant model

---