pure::variants User's Guide Supplement for Model Server

Parametric Technology GmbH

Version 6.0.10.685 for pure::variants 6.0

2025

Table of Contents

1. Introduction

This pure::variants plug-in enables the use of a remote server for a centralized management of variant projects and models.

A printable version of this document is available.

1.1. Differences between local and remote pure::variants projects

There are some differences between local pure::variants projects and pure::variants projects on a remote model server.

- Remote pure::varaints projects can only contain pure::variants models and folders. Meaning, any transformation output, input, or script files cannot be saved on the remote project.

- The access to remote pure::variants projects is restricted based on the permissions the different user. The permissions can be defined for the whole project, folders, models and on element level. For local pure::variants projects there is no user management.

- Changes on remote projects are shared instantly with other clients. It is not necessary to save models after changing them. So all users are always working on the latest version of the model. Sharing of local projects involve a transport of the models to the user, which can be done using a version control system.

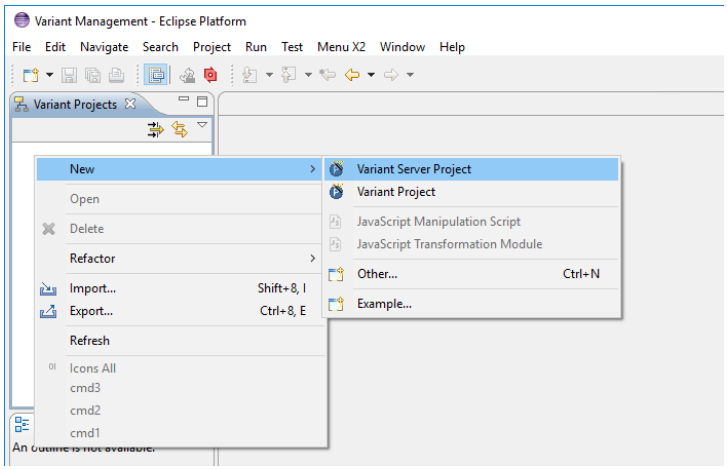
- Remote projects support branching and tagging of pure::variants projects. See Section 2.10, “Working with Branches and Versions” for more information about this.

2. Working with the pure::variants Server

2.1. Create a new Variant Server Project

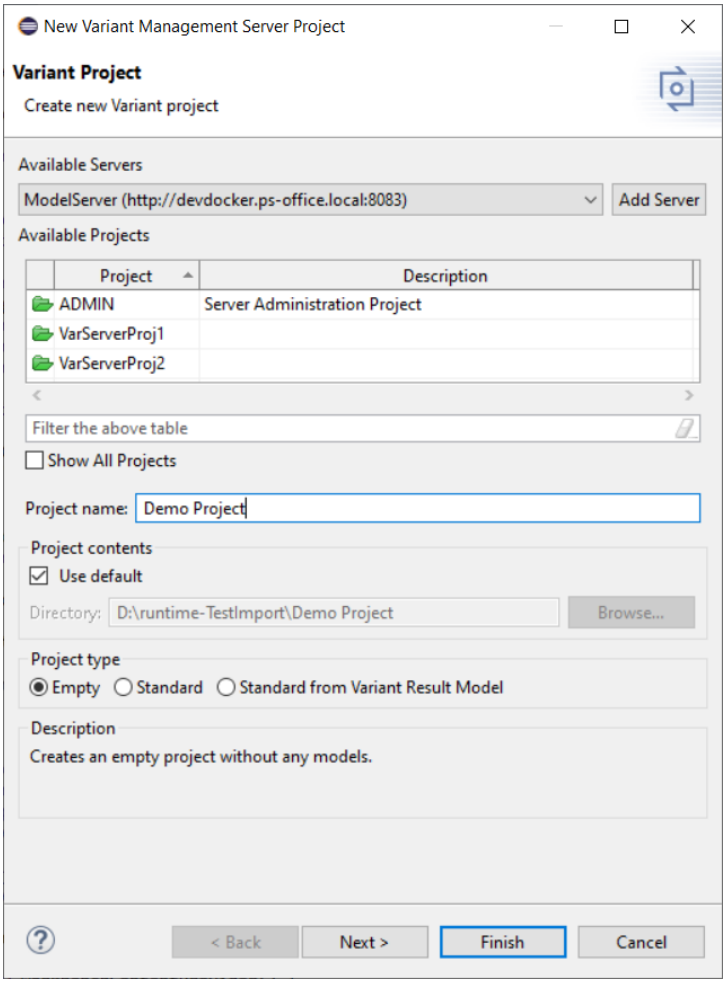
To create a new variant server project select “New” -> “Variant Server Project” in the context menu of the “Variant Projects” view. The “New Variant Management Server Project” wizard appears.

Figure 1. Create new Server Project



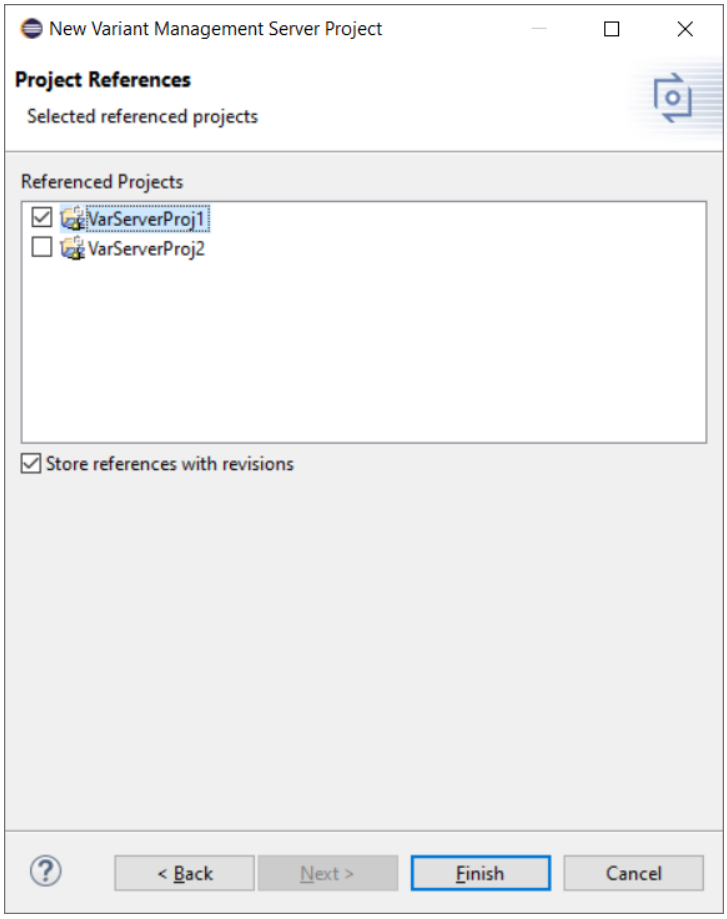
From the “Available Servers” list select the server on which the project should be created. If the requested server is not listed, press the “Add Server” button and enter the URL of the server to add. After selecting a server the “Available Projects” list is filled with all known projects of this server. If you are not logged in to the server yet, a login dialog comes up and asks for your credentials. Now enter the name of the new project and choose the project type.

Figure 2. New Variant Management Server Project Wizard



If the workspace already contains the variant server projects then the "Next" button will be enabled, by clicking on "Next" you can navigate to "Project References" page. In this page, all the available variant server projects will be listed, which can be used as references for the new project that is being created. If "Store references with revision" is checked, the revision information is also stored along with the project name.

Figure 3. Project References Page



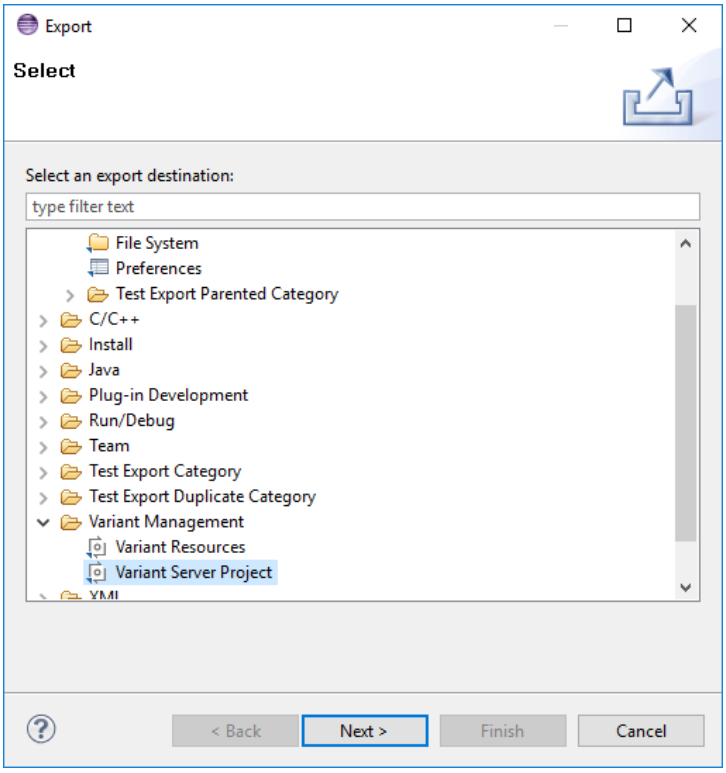
The created project can now be used similar to a local project. There is one big difference. All changes performed in a variant server project are immediately communicated to the pure::variants server. So all other users can see the changes at once. It is not necessary to save changes.

2.2. Export an existing Local Project to a pure::variants Server

Instead of creating a new variant server project an existing local pure::variants project can be exported to a pure::variants server as well.

To export an existing project select "Export..." from the context menu of that project. The standard Eclipse export wizard comes up. Select "Variant Server Project" in the "Variant Management" category and proceed to the next page. The next page allows the user to select one local pure::variants project. It is not possible to export multiple projects at once.

Figure 4. Exporting A Local Project (1)



Proceeding to the next page allows the user to specify the target server and the target project name. From the “Available Servers” list select the server to which the project should be exported. If the requested server is not listed, press the “Add Server” button and enter the URL of the server to add. After selecting a server the “Available Projects” list is filled with all known projects of this server. In most cases the rest of the page can be left as it is.

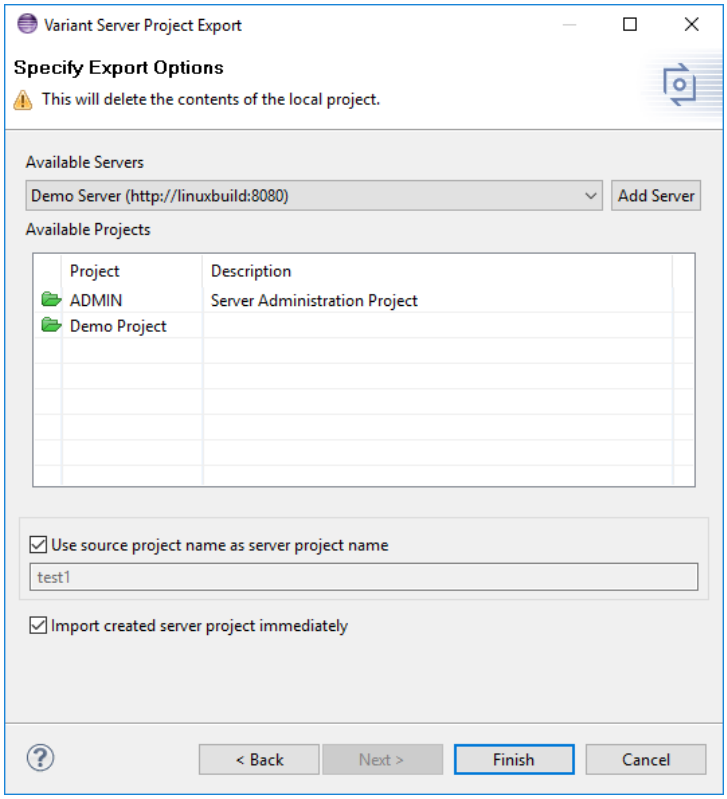
If it is necessary to rename the project while uploading, deactivate the checkbox "Use source project name as server project name" and specify a new name.

Checking the option "Import created server project immediately" causes pure::variants to import the created remote project immediately after the export is performed. This overwrites the local project, if the remote project maintains the same name as the source project.

Note

The pure::variants server supports model files, configuration spaces and folders only. All other files can not be uploaded to the pure::variants server.

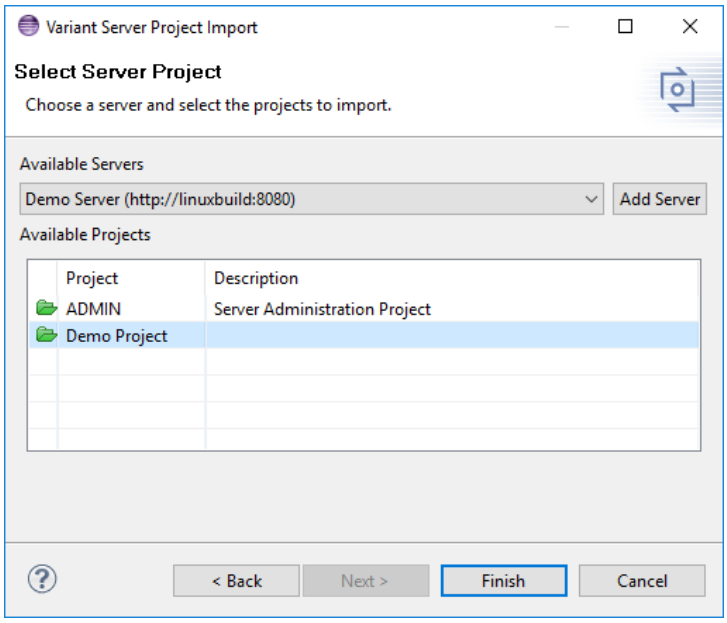
Figure 5. Exporting A Local Project (2)



2.3. Import an existing Variant Server Project

To import an already existing variant server project use the standard Eclipse import. For this purpose choose “Import...” from the context menu of the “Variant Projects” view. Select the “Variant Server Project” item in the “Variant Management” category and click “Next”.

Figure 6. Import Variant Server Project Dialog



From the “Available Servers” list select the server on which the project should be created. If the requested server is not listed, press the “Add Server” button and enter the URL of the server to add. After selecting a server the

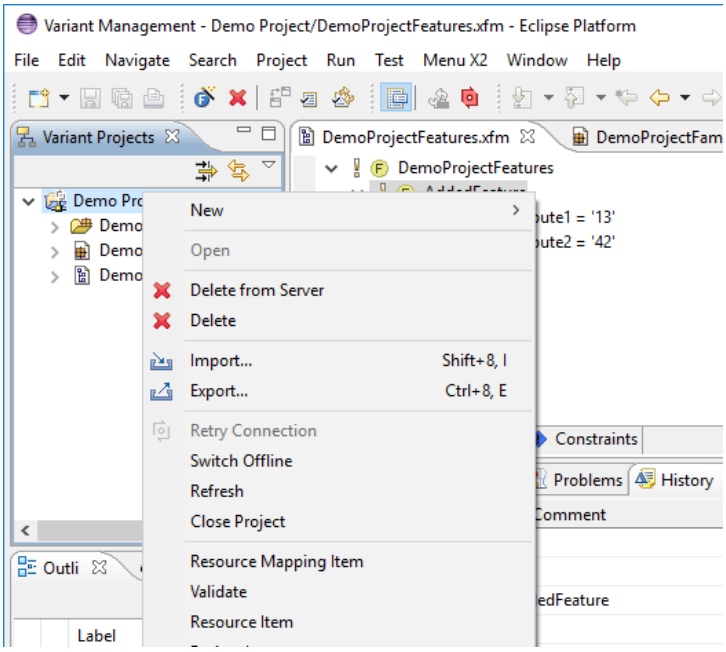
“Available Projects” list is filled with all known projects of this server. Select the project to import and press Finish. If the selected project has project references, you will be asked if those projects should also be imported. If the references got stored revisions, the referenced projects will be imported with that revision, otherwise in the same revision as the selected project.

2.4. Delete a Variant Server Project

In comparison to a local variant project, a variant server project can be deleted in two different ways. The "Delete" action in the context menu of the "Variant Projects" view just deletes the local representation of the server project. The project on the pure::variants server remains untouched.

The "Delete from Server" action instead removes the project from the pure::variants server and the local representation.

Figure 7. Deleting a Variant Server Project



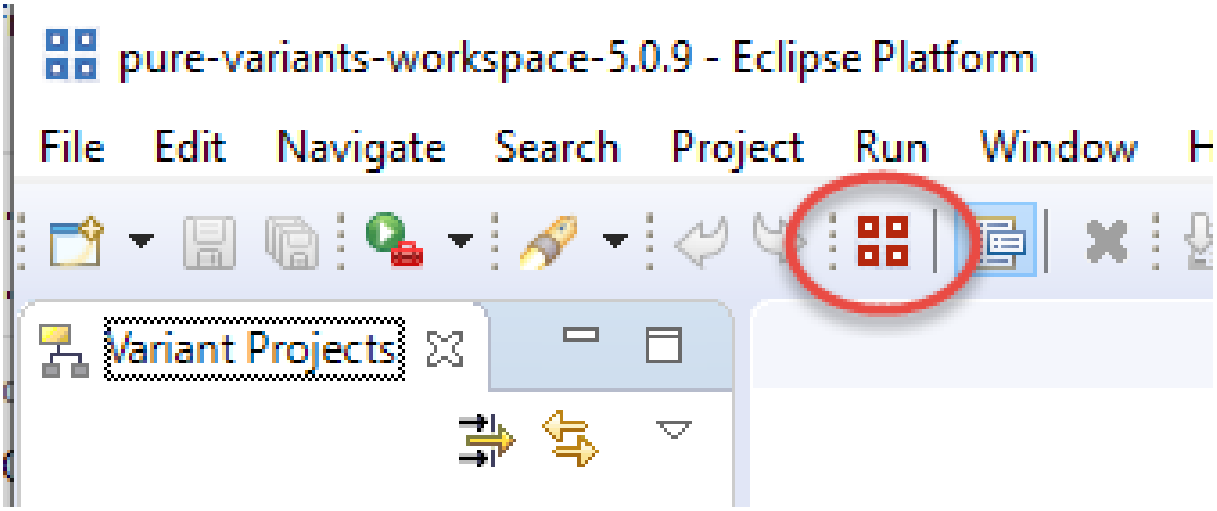
2.5. Disconnect from a pure::variants model server

Note

This action does disconnect the current user from all pure::variants model server and closes all pure::variants models including models, which do not belong to any remote model. To reopen the models in pure::variants server projects a reconnect to the servers is necessary.

To disconnect from a pure::variants model server the following toolbar item is used. It is located in the toolbar. This closes all sessions on all pure::variants model servers and allows the user to log into the server using different credentials.

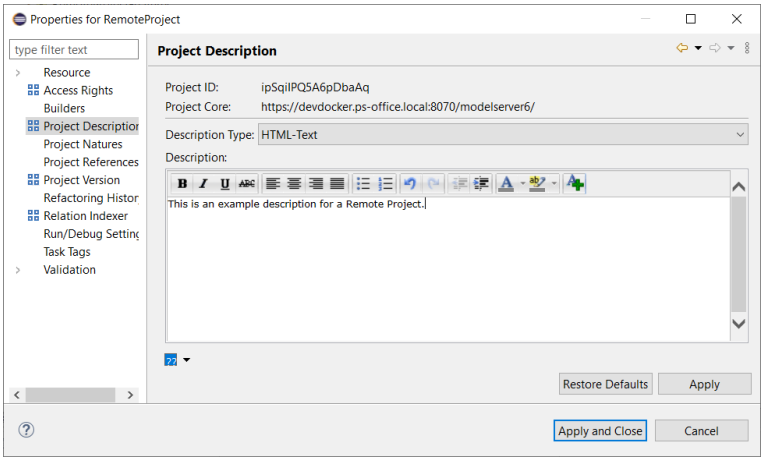
Figure 8. Disconnect from pure::variants model server



2.6. Project Description

It is possible to define a project description for collaboration purposes and a better overview of your projects. From this page you can also retrieve other relevant information, e.g. the project ID or project core URL. You can switch between plain text and HTML text by clicking on the combo box. When changing the MIME type of the description, the actual description gets converted into the new type. You can also define the description in different languages, therefore you can switch the language by clicking on the bottom left dropdown.

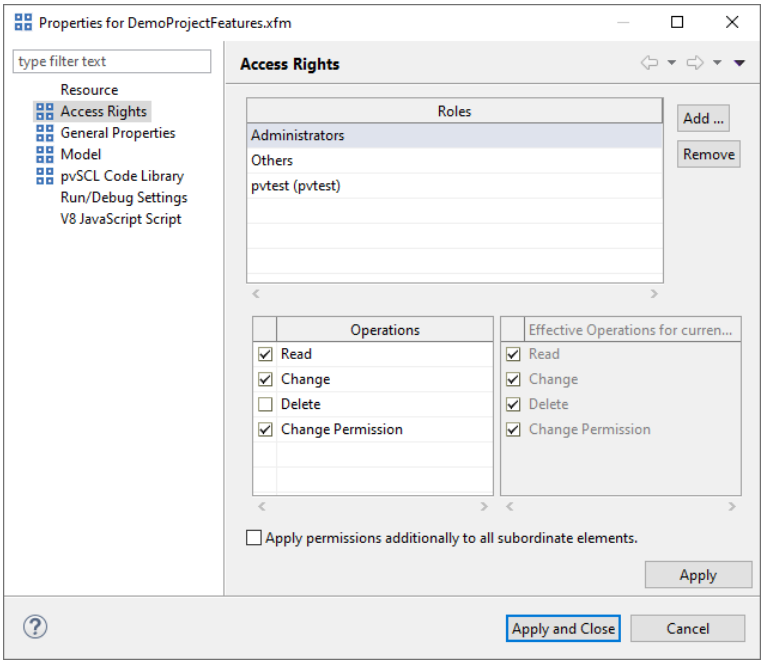
Figure 9. Project Description Property Page



2.7. Change Access Rights

Access rights are available for projects, folders, configuration spaces, models, and model elements. For changing the access rights for instance of a model, select the model in the “Variant Projects” view and choose “Properties” from the context menu. Select the “Access Rights” page in the opened dialog.

Figure 10. Access Rights Property Page



Select the role and change its rights. Press “Apply” or “Apply and Close” to set the new access rights. If the “Apply permissions additionally to all subordinate models and elements” box is checked, the new permissions are applied recursively to all subordinate elements resp. models, folders, and configuration spaces.

By pressing the “Add...” button you can add new roles and users to the roles list.

To remove a role or user use the “Remove” button. This removes the defined access rights for the selected role or user.

The lower right table does show the permissions active on the selected resource for the currently logged in user. They may be different from the permissions directly defined for the user on the selected resource since the effective permissions are an accumulation of the permissions from all roles the user belongs to and may be inherited from parent resources.

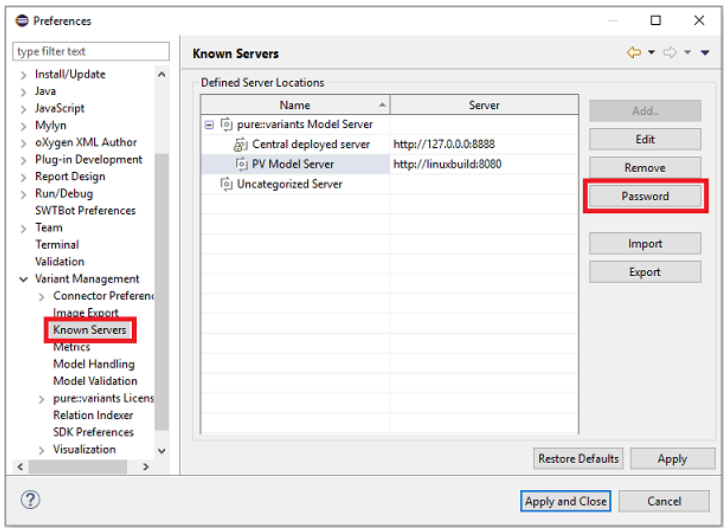
2.8. Change User Password

User can change password of a pure::variant server using the Preferences dialog. Open *"Variant Management -> Known Servers"* in *"Windows -> Preferences"*. Select the server entry from pure::variant Model Server category for which the user password shall be changed and then select the button *"Password"*. Enter the old password and twice the new password to change the password.

Note

The known server page contains servers of several categories which are being utilized by the corresponding connectors. User may only be able to change password of **pure::variants Model** category servers. For servers of other categories, *Password* button will remain disabled.

Figure 11. Change User Password



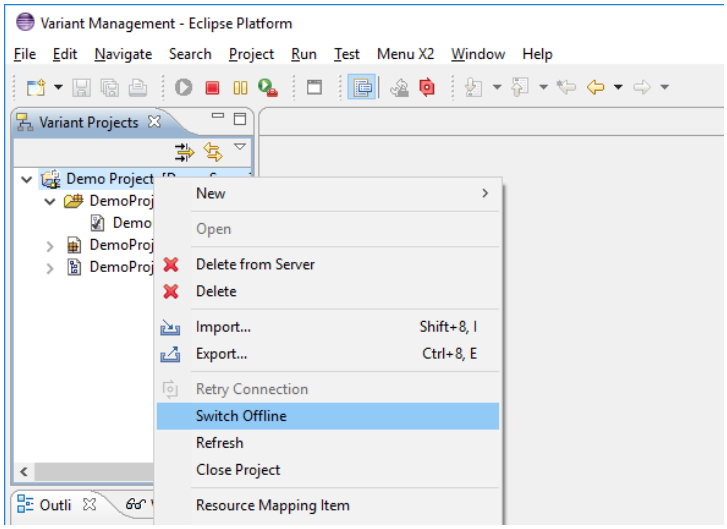
2.9. Switch a Variant Server Project Offline or Online

It may be necessary to switch a pure::variants server project offline, if the user needs access to the project independently from the central IT infrastructure.

Switch a Variant Server Project Offline

To switch offline a pure::variants server project use the "Switch Offline" entry in the context menu of the selected project. The project is saved to the users local disc and can then be used like any other local project. Changes made to the project on the pure::variants server are not applied to the local project and changes to the local project are not applied to the server, even if the server is available.

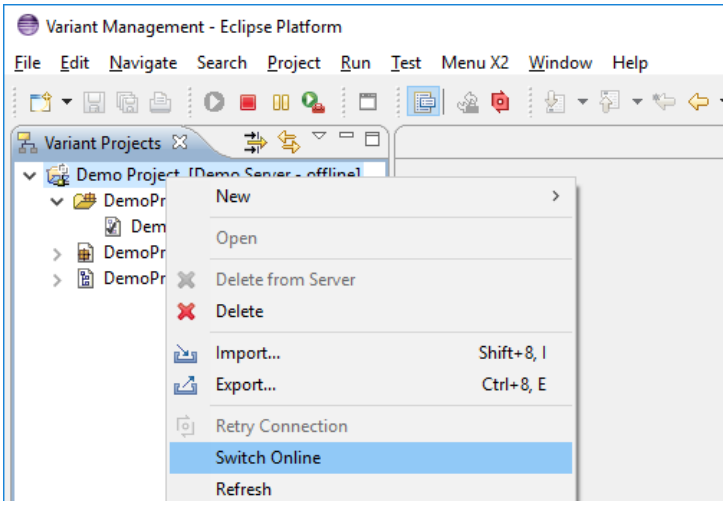
Figure 12. Switch Variant Server Project Offline



Switch a Variant Server Project Online

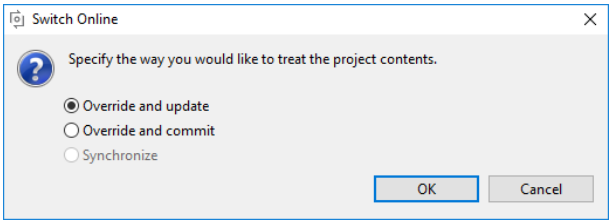
To switch a pure::variants server project back online use the "Switch Online" entry in the context menu of the selected project. Since the remote project can have been changed since the project was switched offline, the action opens a dialog which provides multiple options for handling differences between the local and the remote representation of the pure::variants server project.

Figure 13. Switch Variant Server Project Offline



The option "Override and update" retrieves the remote representation and overwrites all local changes. The result is the current state of the remote representation before the local project was switched online again. The second option "Override and commit" works in the other direction. It replaces the remote content with the state of the local representation of the project.

Figure 14. Switch Online Dialog



2.10. Working with Branches and Versions

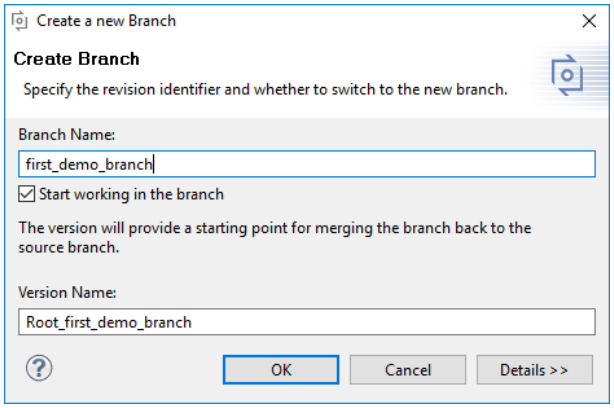
The branching and tagging versions on a pure::variants server works similar to the branching and tagging versions in CVS.

Branch a Variant Server Project

Branching means creating a new line of development on the pure::variants server. This may be useful in different cases, for example if different clients wish to get the same product but with some differences in functionality. Of course it's not convenient to create both products from the beginning to the end separately, so the developers create branches.

To create a new branch the user uses the "Team -> Branch..." menu item of the variant server project's context menu. The following dialog will appear to help the user to create a branch.

Figure 15. Branch Dialog



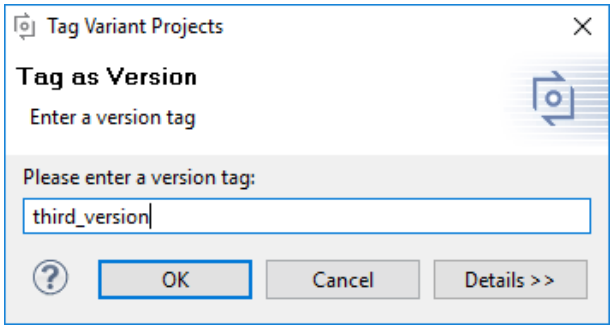
After specifying a branch name and a name for the branch's root version the dialog is closed with the "OK" button and pure::variants creates the specified branch and version. The remote project automatically switches to the newly created branch.

Create a Version of a Variant Server Project

To secure a specific state of a variant server project it can be tagged as a version. It is possible to go back to this version anytime. It is even possible to branch a version to start a new developing branch from the specific tagged state of the variant server project.

To create a new version the user uses the "Team -> Tag as Version..." menu item of the variant server project's context menu. The following dialog will appear to help the user to create a version.

Figure 16. Branch Dialog



After specifying the version name and closing the dialog with the "OK" button pure::variants creates the version.

Switch to a different Branch or Version

If it is necessary to work in a different branch or on a specific version the project can be switched to that branch or version.

This is done using the "Replace With -> Another Branch or Version" item from the remote project's context menu. A dialog comes up listing all existing branches and versions for the selected project. Choose one and close the dialog. pure::variants switches the project to the selected branch or version.

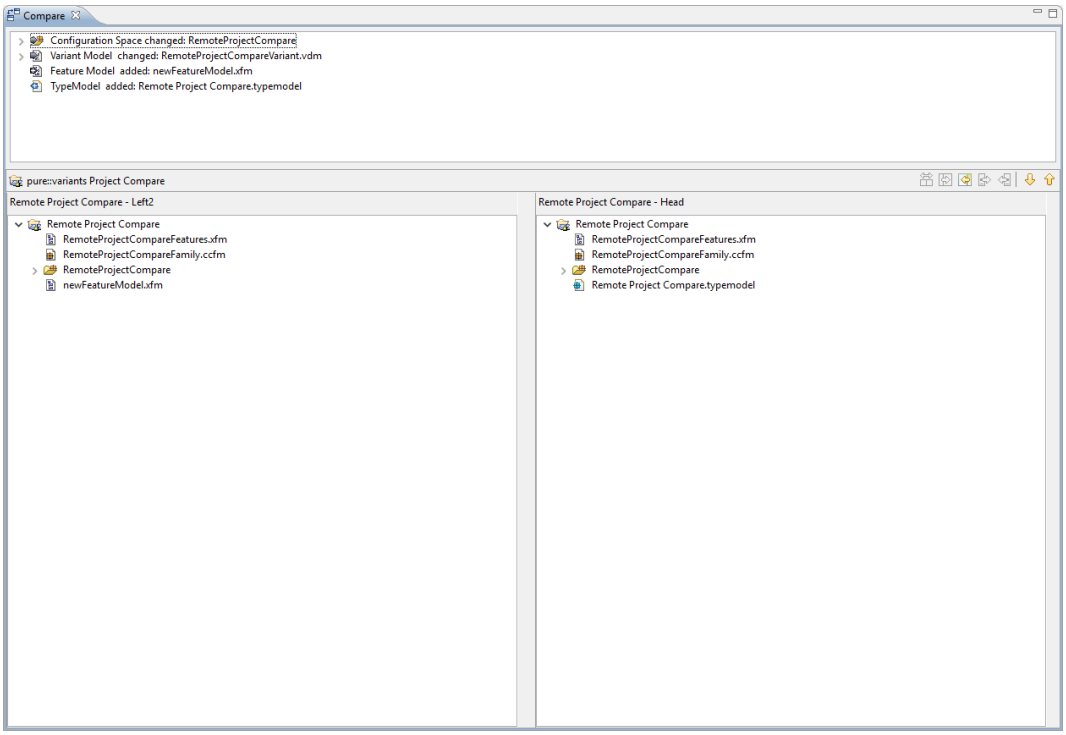
Compare a pure::variants project with another branch or version

If a remote project is used, different revisions can also be compared with the remote project compare. It is opened like any other compare via the context menu entry "Compare with ...". Choosing "Another Branch or Version"

will open a selection dialog for selecting the revision to compare the current one to. pure::variants then suggests the likely ancestor of both revisions used for the three-way-compare. The computed ancestor does not have to be the one the user wants to use though, which is why the common ancestor can be selected manually.

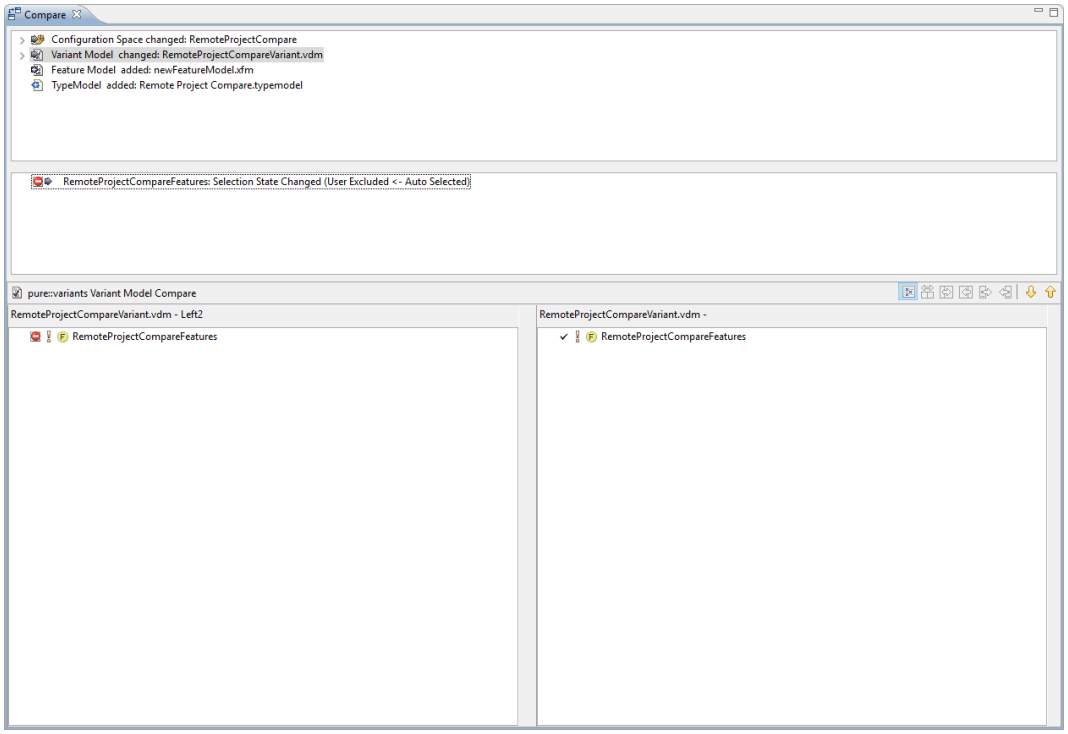
After all revisions are set, the remote project compare opens if there are differences found in those revisions. This check is done using the history of the models involved. If the history is disabled, the project compare will always open, but might be empty. The project compare itself is separated into two parts. On startup the project view of the compare will open, showing the projects on either side in the lower part of the compare with the differences listed in the upper part as seen in the picture below. Added and removed models can be merged here via the toolbar but only towards the current revision.

Figure 17. The pure::variants Project Compare



Clicking on a difference that represents an added or removed model will highlight this model in the lower part of the compare. If a model change was selected the lower view will switch to represent the models that changed. For performance reasons, it is only now that the actual differences of the changed models will be calculated. The result is shown in a new differences view as seen in the picture below. Changed models can be merged here via the toolbar but only towards the current revision.

Figure 18. The pure::variants Project Compare on changed Models



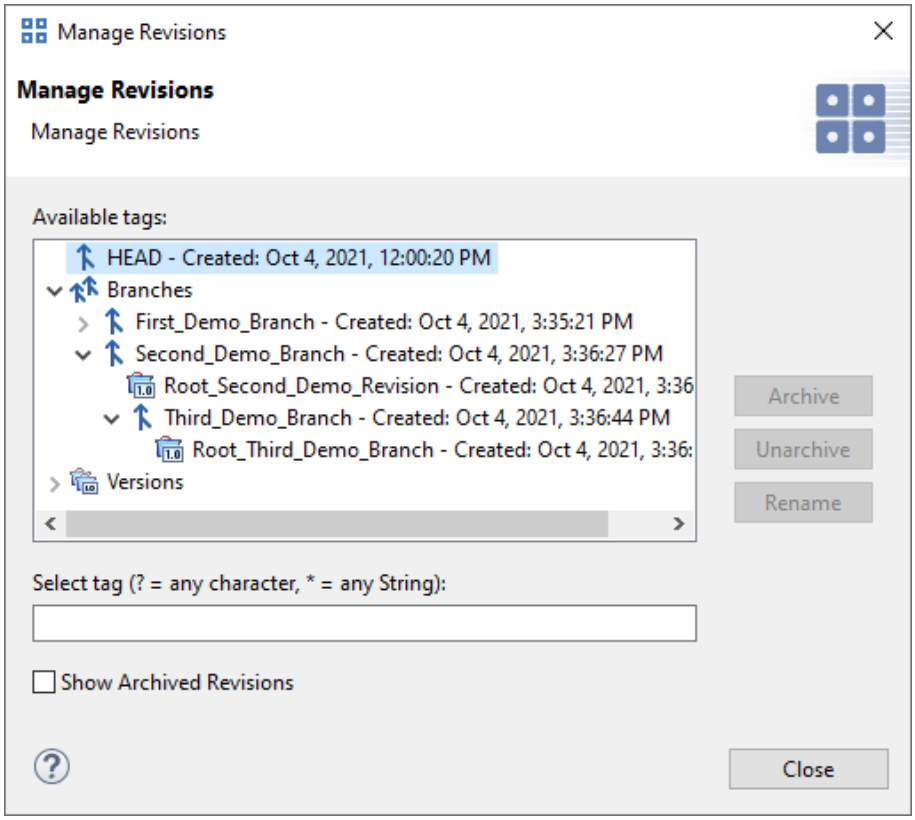
Compare a pure::variants model with another branch or version

To compare a model with the same model in a different branch or version choose the "Compare With -> Another Branch or Version" item from the model's context menu. A dialog comes up listing all existing branches and versions for the selected project. Choose one and close the dialog. A compare editor is opened showing the differences between the selected model and the model from the chosen branch or version. Differences can be merged to the local representation of the remote project.

Manage Revisions

To archive or rename a branch or a version choose "Team -> Manage Revisions" from the remote projects context menu. This opens a dialog listing all currently existing revisions.

Figure 19. Manage Revision Dialog



This dialog allows the user to archive, unarchive and rename revisions. The revision edited by any of the operations must not be used by any user of the remote project.

To archive a revision select the revision and click "Archive". If a revision shall be archived all child revisions have to be archived first. This is taken care of by the action. A question dialog pops up asking if all the child revisions shall be archived too. For the archive operation the user has to have delete permissions on the revision.

Archived revisions are not shown in any of the dialogs listing revision by default. They can be shown using the "Show Archived Revisions" checkbox, which exists in all relevant dialogs.

An archived revision can be unarchived. To perform this action selected the revision, which shall be unarchived and click "Unarchive". If a revision shall be unarchived all parent revisions have to be unarchived too. This is taken care of by the action. A question dialog pops up asking if all parent revisions shall be unarchived as well.

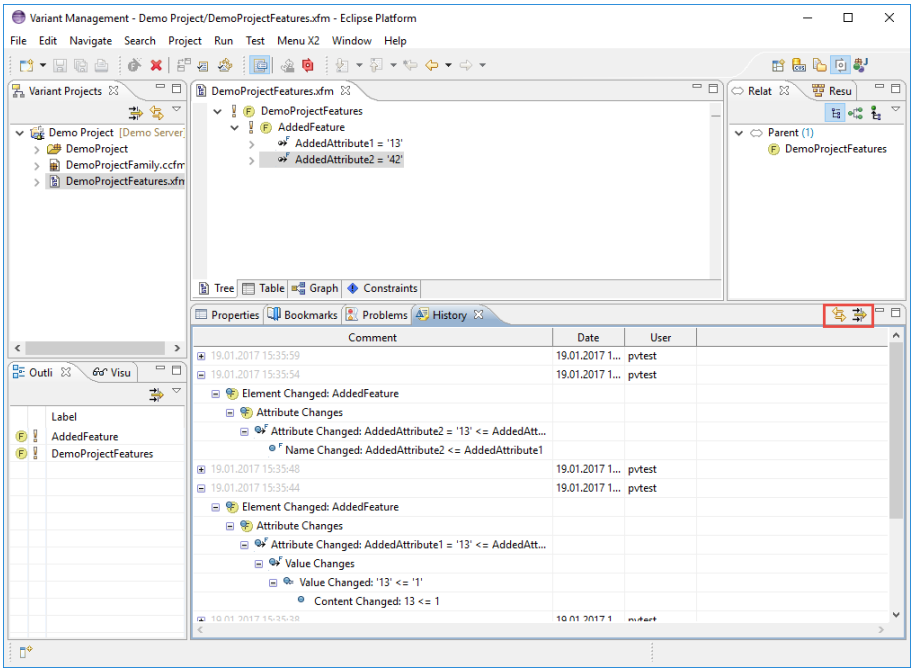
The HEAD revision can not be archived or renamed.

To rename a revisions select the revision to rename and click "Rename". Give the new name in the opening dialog and click ok. A revision can only be renamed, if it is not used by any user of the remote project.

2.11. Show History of Model Changes

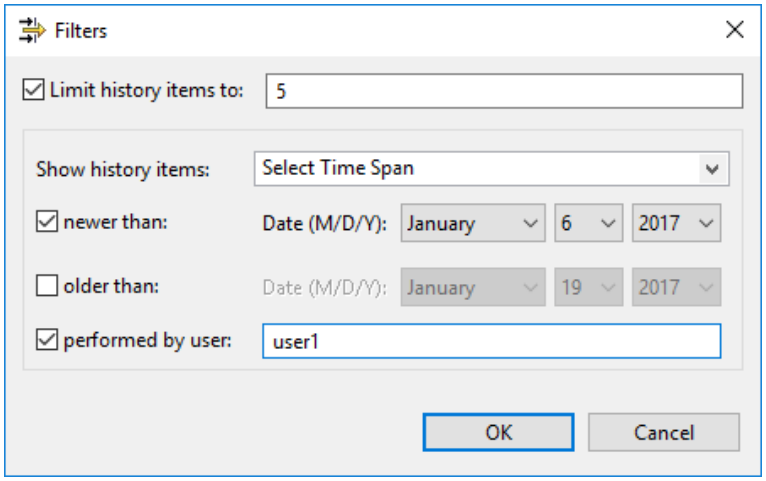
For remote servers a model change history view is available (see Figure 20, “History View”). This view shows each change on the elements of a model. The history entries can be commented and filtered. To open the view use the "Show View" -> "History" entries from the "Window" menu.

Figure 20. History View



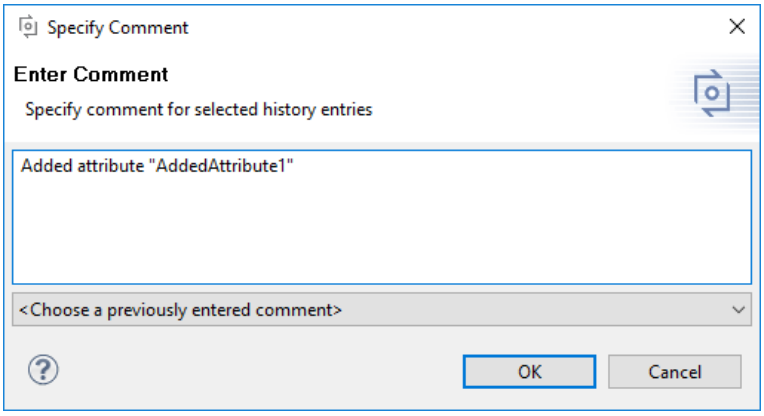
The toolbar of the history view has two buttons. With the left button it can be switched between showing all changes of the model or only the changes for the selected elements in the editor. With the right button a filter can be defined for the history view (see Figure 21, “History View Filter Dialog”). Beside limiting the number of items shown in the history a time period and the user that made the changes can be specified.

Figure 21. History View Filter Dialog



A comment can be added for history entries which do not yet have a comment. Right-click on a change and choose **Set Comment** from the context menu. This opens a dialog where a new comment can be entered (see Figure 22, “Specify Comment Dialog”). Alternatively a previously entered comment can be chosen from a list.

Figure 22. Specify Comment Dialog



To easily find a corresponding item of a history entry a context menu action is provided. Use "Goto ... " from the context menu to directly jump to the item in the model editor.

The context menu "Compare With Current …" will open the Compare Editor with the models content before the selected change was made. This allows to revert model changes by using the model history.

With the context menu entry "Save as XML ..." the history can be saved to an XML file, listing the shown history entries from the view.

3. Known Restrictions

- The pure::variants server does only support pure::variants models, config spaces and folders. Any other files, e.g. input data and output data, will not be uploaded to the server.