
pure::variants Connector for Capella Integration Manual

pure-systems GmbH

Version 5.0.8.685 for pure::variants 5.0

Copyright © 2003-2021 pure-systems GmbH

2021

Table of Contents

1. Introduction	1
1.1. What is pure::variants Connector for Capella Integration?	1
1.2. Software Requirements	2
1.3. Installation	2
1.4. About this manual	2
2. Using pure::variants Connector for Capella Integration	2
2.1. Starting pure::variants	2
2.2. Setup preferences for using Connector for Capella Integration	3
2.3. How pure::variants Connector for Capella Integration Works	3
2.4. Adding Capella Projects to pure::variants Family Models	4

1. Introduction

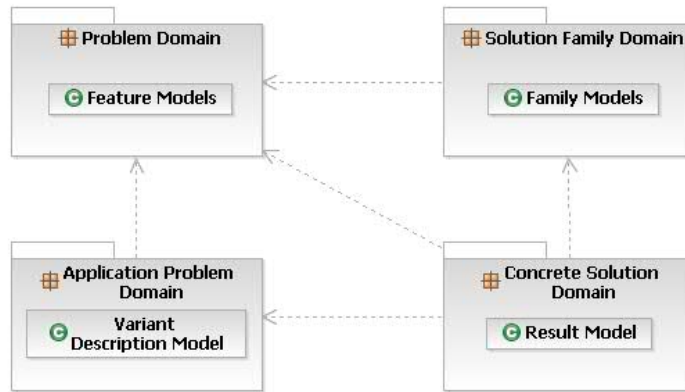
1.1. What is pure::variants Connector for Capella Integration?

pure::variants Connector for Capella Integration enables use of product line variability concepts in Capella projects. It allows to maintain one master project from which different project variants are created automatically by selecting features from Feature Models in pure::variants. So instead of having to merge changes in slight variations of the base Capella models, the change is applied once to the master project and then all relevant variants are automatically generated by pure::variants.

[Figure 1, “Overview of family-based software development with pure::variants”](#) shows the four cornerstone activities of software product line development and the models used in pure::variants as the basis for these activities.

When building the infrastructure for your Product Line, the problem domain is represented using hierarchical Feature Models. The solution domain, i.e. the concrete design and implementation of the software family, is implemented as Family Models.

The two model types used for Application Engineering, i.e. the creation of product variants, are complementary to the models described above. The Variant Description Model (VDM), containing the selected feature set and associated values, represents a single problem from the problem domain. The Variant Result Model describes a single concrete solution drawn from the solution family.

Figure 1. Overview of family-based software development with pure::variants

pure::variants manages the knowledge captured in these models and provides tool support for co-operation between the different roles within a family-based software development process:

- The *domain analyst* uses the pure::variants Feature Model editor and models in Capella to build and maintain the problem domain model containing the commonalities and variabilities in the given domain.
- The *domain designer* uses Capella models to describe the variable family architecture and to connect it via appropriate rules to the Feature Models.
- The *application analyst* uses a Variant Description Model to explore the problem domain and to express the problems to be solved in terms of selected features and additional configuration information. This information is used to derive a Variant Result Model from the Capella model(s).
- The *application developer* generates a member of the solution (feature selections and variant-specific Capella models) from the Variant Result Model by using the transformation engine.

1.2. Software Requirements

The following software has to be present on the user's machine in order to support the pure::variants Connector for Capella Integration:

Capella: Capella or Capella Studio of version 1.1.0 - 5.1.0 is required. Compatibility with other Capella releases is not guaranteed. In Capella, the pure::variants Integration for Capella needs to be installed.

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

1.3. Installation

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

1.4. About this manual

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

2. Using pure::variants Connector for Capella Integration

2.1. Starting pure::variants

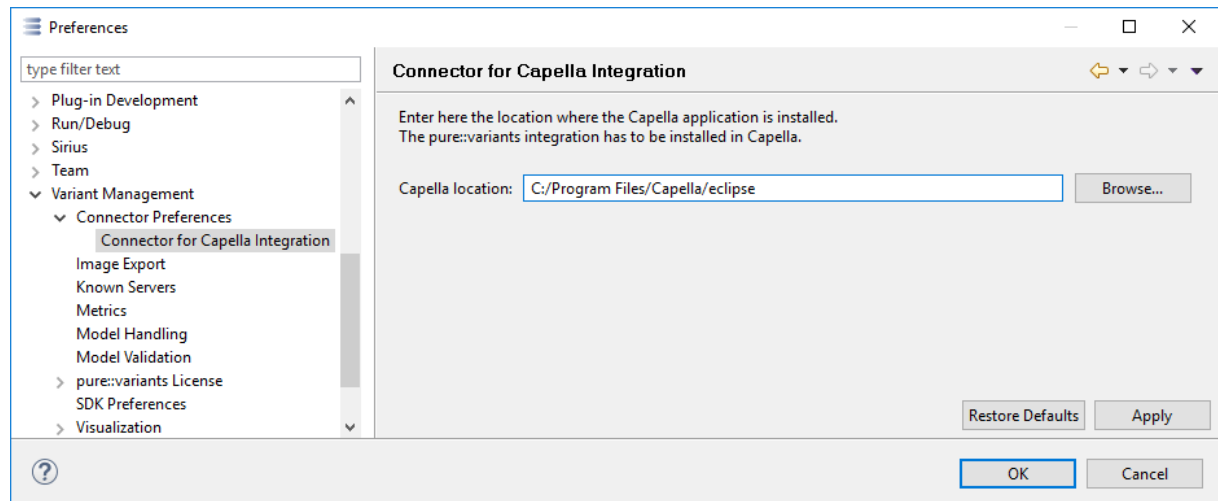
Depending on the installation method used either start the pure::variants-enabled Eclipse or under Windows select the **pure::variants** item from the **program** menu.

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

2.2. Setup preferences for using Connector for Capella Integration

Before the first transformation the path to the Capella installation needs to be set in the connector preferences. Open the Eclipse preferences **Window -> Preferences** and select the **Connector for Capella Integration** page below **Variant Management / Connector Preferences**. Enter the path to your Capella installation into the **Capella location** field.

Figure 2. Connector for Capella Integration Preferences



2.3. How pure::variants Connector for Capella Integration Works

Before the Capella project is extended with variability information, a corresponding feature model project should be set up in the pure::variants Connector for Capella Integration. In this pure::variants project the features to control variability in Capella projects are maintained. To add variability information to Capella models, a mapping between the variability information and the affected Capella elements is used. The mapping is stored in the Capella model under root node "Advanced Variability". Please see the pure::variants Integration for Capella manual for details on how to store variability information on different Capella model elements. The documentation is available in Capella at **Help->Help Contents**.

There are two types of variability information: *Conditions* and *calculations*. A condition controls the existence of the mapped Capella element(s), while a calculation is used to replace the value of an attribute with a calculated value.

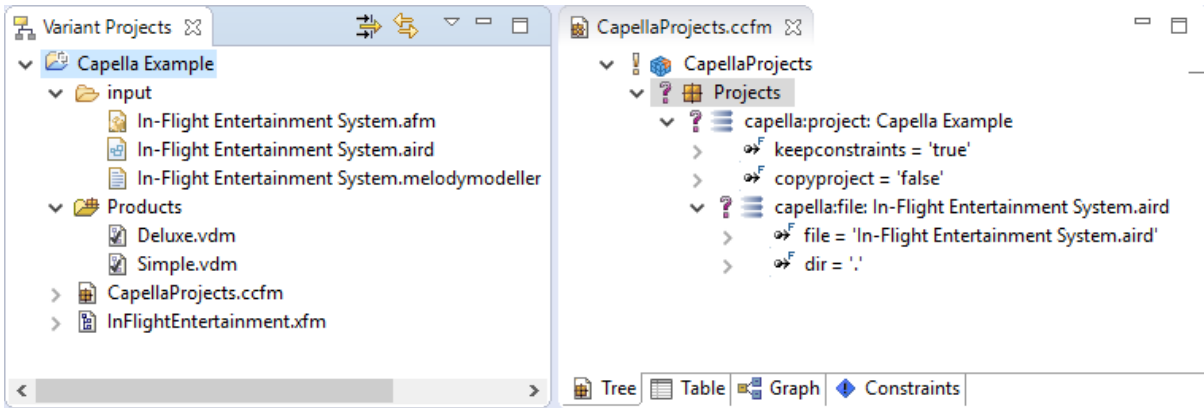
One element can be mapped to multiple conditions. If one of the element's conditions evaluates to 'false', the element is removed. Similarly one element attribute can be mapped to multiple calculations. To control which calculation applies to the element's attribute during transformation, each calculation has a special condition assigned. This condition decides whether the calculation is evaluated during transformation or not.

The language used for conditions and calculations is pvSCL (pure::variants Simple Constraint Language, see pure::variants User's Guide), which provides simple and intuitive syntax for expressing feature model conditions. For instance, to make an element optional and include it only when feature `WindSpeed` is not selected in pure::variants, the corresponding pvSCL rule is simply naming the feature inside the not operator: `not(WindSpeed)`. To create variants of the master Capella project, Variant Description Models (VDMs) have to be created in the pure::variants Connector for Capella Integration project. Each VDM contains the feature selection for one project variant. The transformation of a project variant will create a Capella project variant in a specified output location. All optional elements mapped to a failing condition are removed from this project variant and calculated values are applied to the mapped attributes.

2.4. Adding Capella Projects to pure::variants Family Models

Add Capella project information to family models to allow the generation of variant-specific Capella projects. The pure::variants Connector for Capella Integration recognizes the family model part `capella:project` as trigger for the transformation. Below this part must be at least one `capella:file` element pointing to a Capella project file.

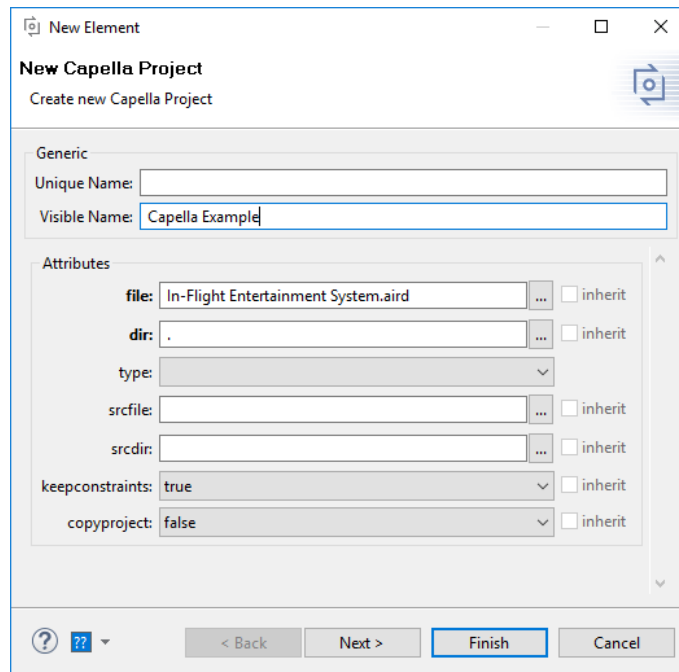
Figure 3. Family Model containing Capella Project Information



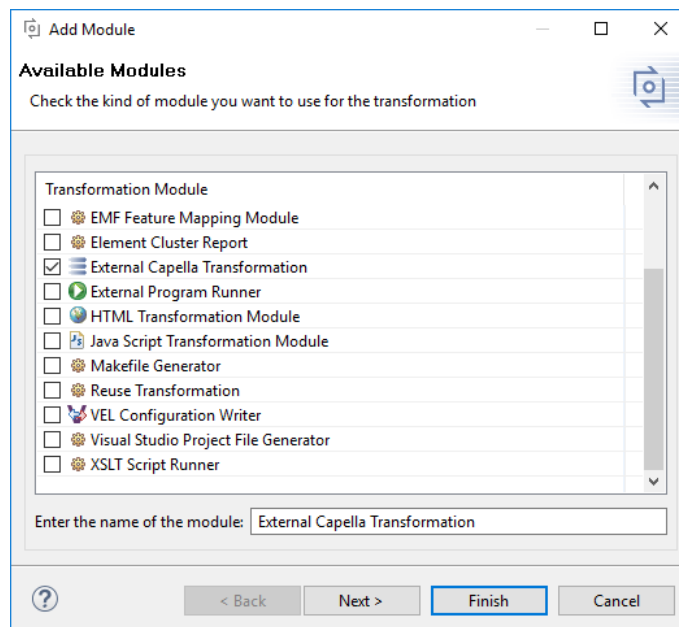
To influence the behavior of the transformation, the `capella:project` element can have the following attributes:

- `keepconstraints`: If this attribute is set to `true` the variability information stored below the "Advanced Variability" node will remain in the created project variant. Setting this to `false` will remove all conditions and calculations from the project during variant generation.
- `copyproject`: If this attribute is set to `true`, all files of the Capella project are copied to the output folder and the variant project name is set to the name of the output folder. Otherwise, only the files ending on `.aird`, `.melodymodeller`, and `.afm` are copied to the output folder.

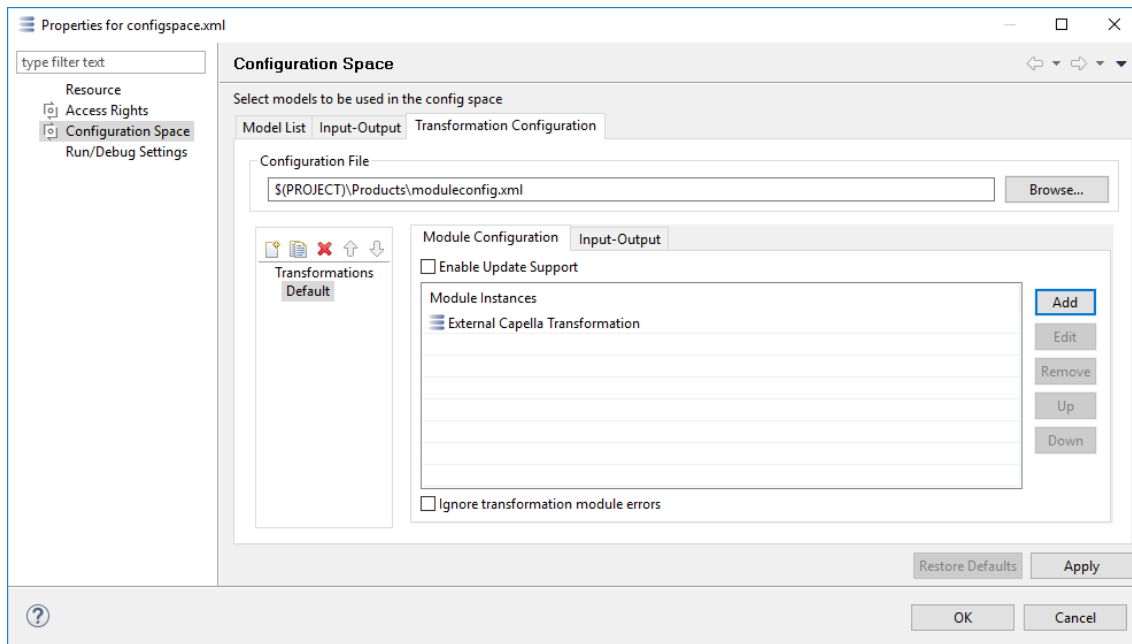
The family model part `capella:project` is added to a family model by right-clicking on a component element and then choosing **New->Capella Project** from the context menu. This opens the wizard for a new Capella project file part as shown in Figure 4, "New Capella Project wizard". Enter the name and path to the project's `.aird` file, or navigate to an existing file by clicking on button `...` to the right of field **file**. After finishing the wizard the part and the file element is created as shown in Figure 3, "Family Model containing Capella Project Information".

Figure 4. New Capella Project wizard

It is necessary to add the Capella transformation to the transformation module configuration. To add the transformation module, right-click on the configuration space, i.e. folder Products in [Figure 3, “Family Model containing Capella Project Information”](#), and choose **Properties** from the context menu. In the Properties dialog switch to page **Configuration Space** and there to tab **Transformation Configuration**. Click on button **Add**. This opens the transformation module selection dialog as shown in [Figure 5, “Transformation Module Selection Dialog”](#).

Figure 5. Transformation Module Selection Dialog

Select the *External Capella Transformation* module and click on **Finish**. The transformation configuration should then look as shown in [Figure 6, “Transformation Configuration with Capella Transformation”](#).

Figure 6. Transformation Configuration with Capella Transformation

Note

Before starting the first Capella transformation it is necessary to validate the path to the Capella installation in the preferences. See [Section 2.2, “Setup preferences for using Connector for Capella Integration”](#) for details.