
pure::variants Connector for MagicDraw 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 MagicDraw?	1
1.2. Software Requirements	2
1.3. Installation	2
1.4. About this manual	2
2. Using pure::variants Connector for MagicDraw	3
2.1. Starting pure::variants	3
2.2. Setup preferences for using Connector for MagicDraw	3
2.3. How pure::variants Connector for MagicDraw Works	3
2.4. Adding MagicDraw Projects to pure::variants Family Models	4
3. Known Restrictions	6

1. Introduction

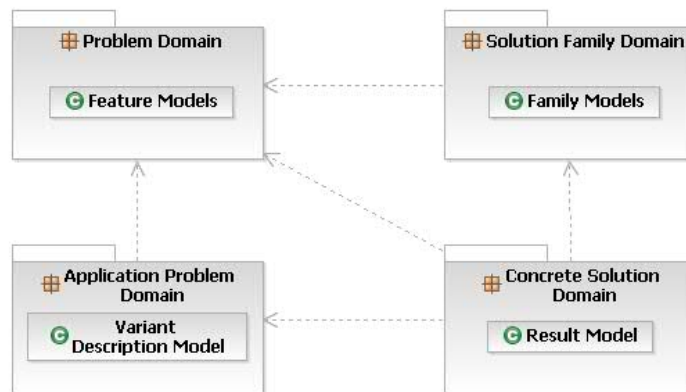
1.1. What is pure::variants Connector for MagicDraw?

pure::variants Connector for MagicDraw enables use of product line variability concepts in MagicDraw 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 UML 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 UML 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 UML/SysML models in MagicDraw to build and maintain the problem domain model containing the commonalities and variabilities in the given domain.
- The *domain designer* uses UML 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 UML model(s) in MagicDraw.
- The *application developer* generates a member of the solution (feature selections and variant specific MagicDraw 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 MagicDraw:

MagicDraw: MagicDraw 18.0 - 2021x is required. Compatibility with other Tool releases is not guaranteed. Inside MagicDraw the "Product Line Engineering" plugin as well as the "pure::variants Integration" needs to be installed.

The pure::variants Connector for MagicDraw 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 MagicDraw

2.1. Starting pure::variants

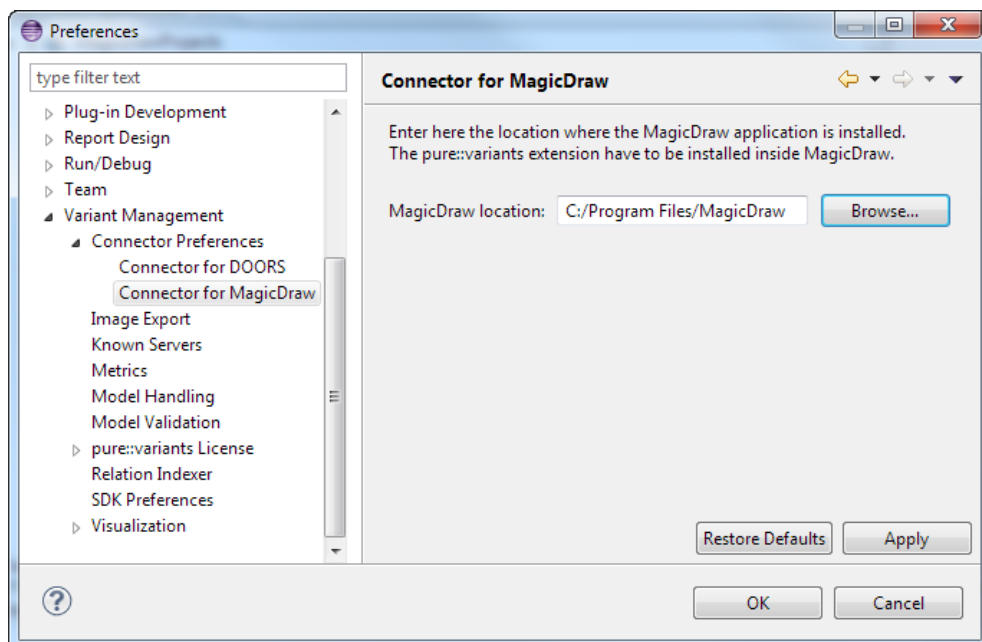
Depending on the installation method used either start the pure::variants-enabled Eclipse or under Windows select the **pure::variants** item from the **program** menu.

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 MagicDraw

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

Figure 2. Connector for MagicDraw Preferences



2.3. How pure::variants Connector for MagicDraw Works

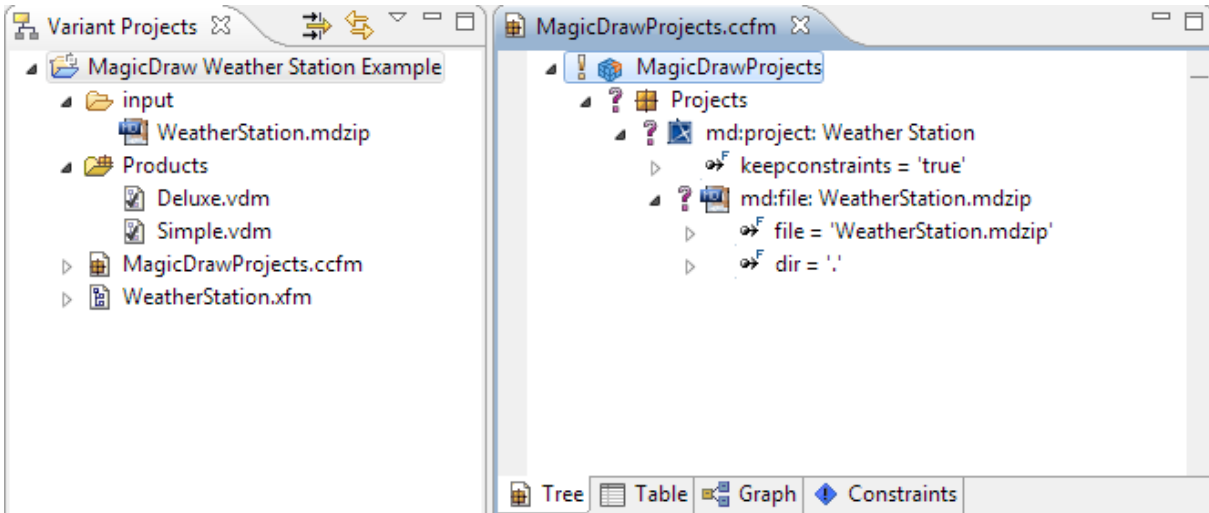
Before the MagicDraw project is extended with variability information a corresponding feature model project should be set up in the pure::variants Connector for MagicDraw. In this pure::variants project the features to control variability in the MagicDraw projects are maintained. To add variability information to UML models, the concept of UML constraints is being used. Special constraints are used to mark up optional elements and connections in an UML model such as classes, states, transition, class attributes and class members. Please see the MagicDraw documentation how to store variability information on the different MagicDraw model elements.

The constraint language 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 a class optional and include it only when the 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 MagicDraw, Variant Description Models (VDMs) have to be created in the pure::variants Connector for MagicDraw project. Each VDM contains the feature selection for one project variant. The transformation of a project variant will create an MagicDraw project variant in a specified output location. All optional elements with failing constraint have been removed from this project variant.

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

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

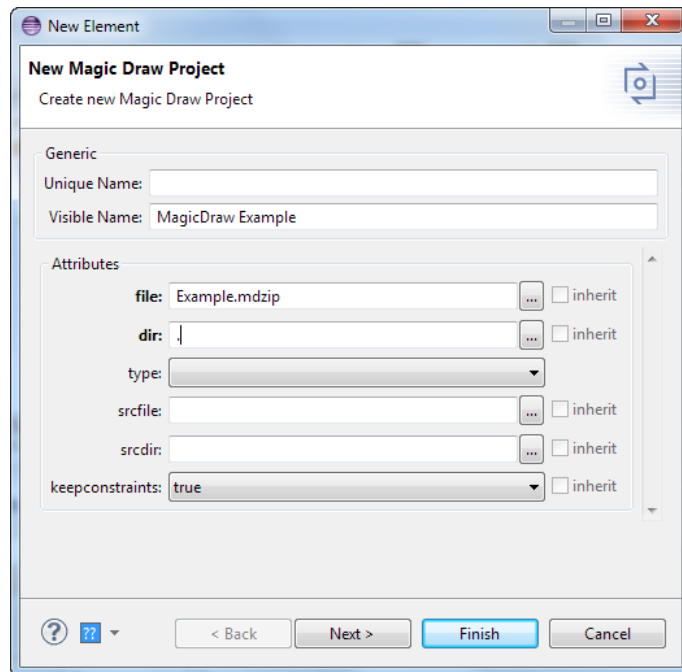
Figure 3. Family Model containing MagicDraw Project Information



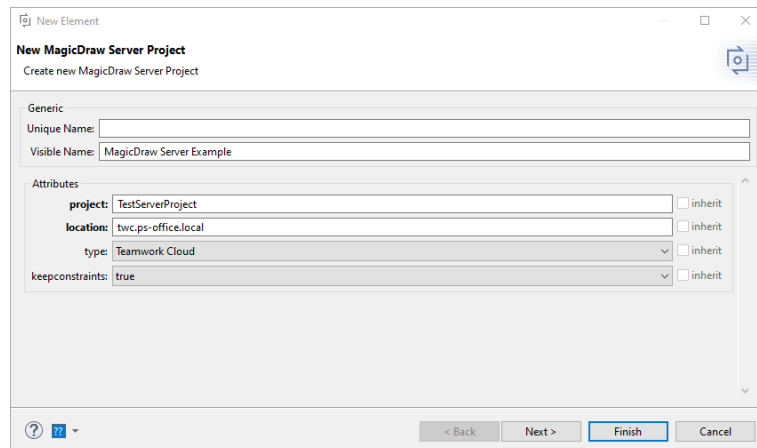
To influence the behavior of the transformation the `md:project` element can have the following two attributes:

- `keepconstraints`: If this attribute is set to `true` the constraints will be remain in the created project variant. Setting this to `false` will remove all constraints from the project during variant generation.
- `scope`: By setting the `scope` attribute the transformation can be limited to the specified model scopes.

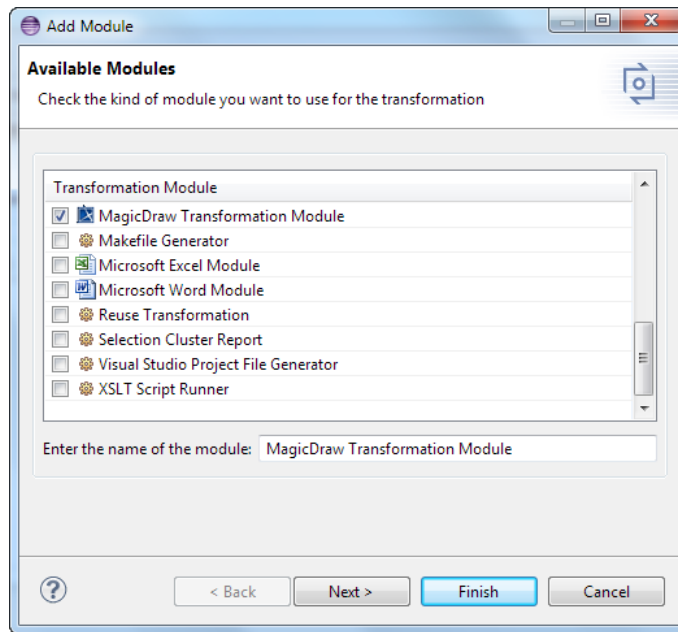
The family model part `md:project` is added to a family model by right-clicking on a component element and then choosing **New->MagicDraw Project** from the context menu. This opens the wizard for a new MagicDraw project file part as shown in Figure 4, “New MagicDraw Project wizard”. Enter the name and path to the project file, or navigate to an existing project 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 MagicDraw Project Information”.

Figure 4. New MagicDraw Project wizard

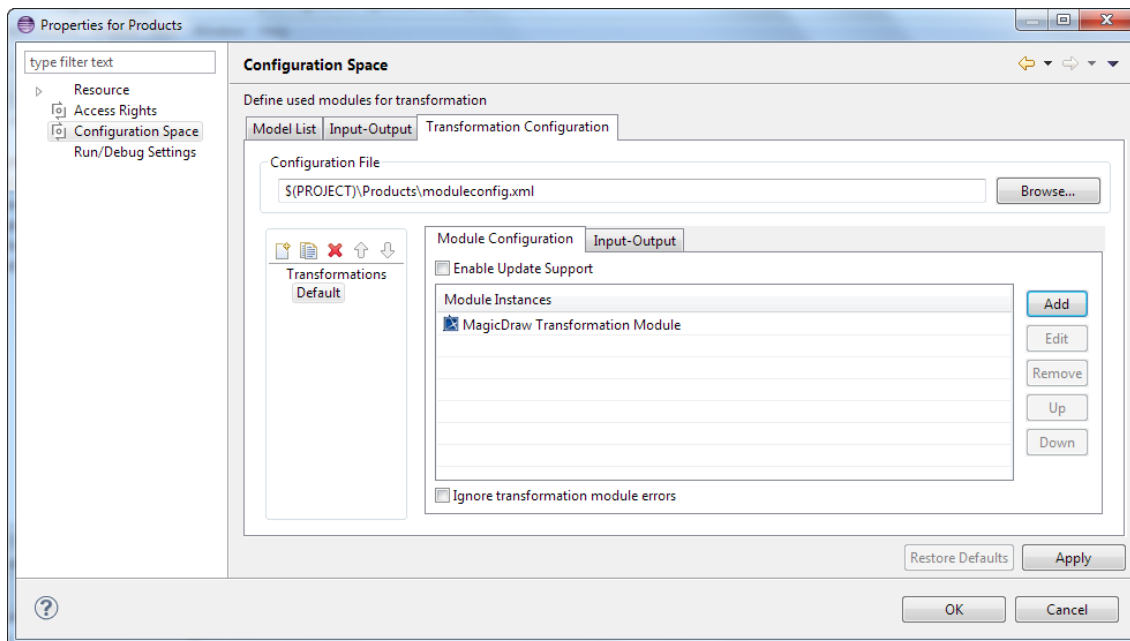
The family model part `md:project` can also be added to a family model by right-clicking on a component element and then choosing **New->MagicDraw Server Project** from the context menu. This opens the wizard for a new MagicDraw project file part as shown in [Figure 5, “New MagicDraw Server Project wizard”](#). Enter a name for the new model part. Enter the name of the existing server project in the field **project**, the server location in the field **location** and select the type in the field **type**. After finishing the wizard the part and the file element is created.

Figure 5. New MagicDraw Server Project wizard

It is necessary to add the MagicDraw 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 MagicDraw 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 6, “Transformation Module Selection Dialog”](#).

Figure 6. Transformation Module Selection Dialog

Select the MagicDraw module and enter a name, then click on **Finish**. The transformation configuration should then look as shown in [Figure 7, “Transformation Configuration with MagicDraw Transformation”](#).

Figure 7. Transformation Configuration with MagicDraw Transformation

Note

Before starting the first MagicDraw transformation it is necessary to validate the path to the MagicDraw installation in the preferences.

3. Known Restrictions

- If using the integration or connector together with variant instances the instance data is not calculated correctly.