

---

# pure::variants - Connector for Microsoft Office Manual

pure-systems GmbH

Version 6.0.3.685 for pure::variants 6.0

Copyright © 2003-2023 pure-systems GmbH

2023

## Table of Contents

1. Introduction .....	1
1.1. What is pure::variants Connector for Microsoft Office? .....	1
1.2. Software Requirements .....	2
1.3. Installation .....	2
1.4. About this manual .....	3
2. Using pure::variants Connector for Microsoft Office .....	3
2.1. Starting pure::variants .....	3
2.2. How pure::variants Connector for Microsoft Office Works .....	3
2.3. Adding Variability using the pure::variants Integration for Microsoft Office .....	4
2.4. Using the pure::variants Integration for Microsoft Word .....	15
2.5. Using the pure::variants Integration for Microsoft Excel .....	23
2.6. Creating a pure::variants Project for Microsoft Word or Excel using the New Project Wizard .....	32
2.7. Adding an Microsoft Word or Excel Transformation to pure::variants Projects for Microsoft Word or Excel .....	33
2.8. Adding Office Documents to pure::variants Family Models .....	35
2.9. Using the Microsoft Office Document Variants .....	38
3. Known Issues .....	39
3.1. Forcefully Enable Buttons in pure::variants Ribbon Tab .....	39

## 1. Introduction

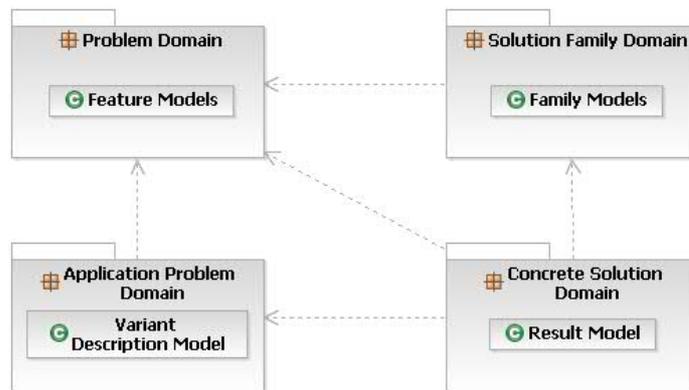
### 1.1. What is pure::variants Connector for Microsoft Office?

pure::variants Connector for Microsoft Office enables the use of product line variability concepts in Microsoft Office *Word* and *Excel* documents. It allows to maintain one master document from which different document 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 documents, the change is applied once to the master document 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 (VRM) 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 documents in Microsoft Office to build and maintain the problem domain model containing the commonalities and variabilities in the given domain.
- The *domain designer* uses Office documents 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 document(s) in Microsoft Office.
- The *application developer* generates a member of the solution (feature selections and variant specific Office documents) 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 Microsoft Office:

Microsoft Office: Office 2016 - 2021 or Office 365 is required (either 32bit or 64bit version). Compatibility with other Office releases is not guaranteed.

Note that the following features have to be installed with Microsoft Office ( Except for Office 365 and Office 2019. For details see section **pure::variants Integration for Microsoft Office** in the **pure::variants Setup Guide**):

- Microsoft Word / .NET Programmability Support (if using Microsoft Office Word Integration)
- Microsoft Excel / .NET Programmability Support (if using Microsoft Office Excel Integration)
- Office Tools / Actions .NET Programmability Support

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

## 1.3. Installation

For working with the pure::variants Connector for Microsoft Office two components have to be installed: The pure::variants Eclipse Plug-in and the pure::variants Integration for Microsoft Office for editing variability information.

## Installation of Eclipse Plug-in

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**).

## Installation of the pure::variants Integration for Microsoft Office

Please consult section **pure::variants Integrations** 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 Integrations**).

### 1.4. About this manual

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

## 2. Using pure::variants Connector for Microsoft Office

### 2.1. Starting pure::variants

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

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

### 2.2. How pure::variants Connector for Microsoft Office Works

The pure::variants Connector for Microsoft Office can be used to create variants of Microsoft Word and Excel documents. We now give a short overview of the steps necessary to create a variant for both Word and Excel.

Before an Office document is extended with variability information, a corresponding feature model project should be set up in the pure::variants Connector for Microsoft Office. In this pure::variants project, the features to control variability in Word and Excel documents are maintained. There are two different types of variability information: *Conditions* and *Calculations*. Conditions are used to mark up optional parts of the document, whereas calculations mark text fragments that are replaced with a calculated value during transformation. Both conditions and calculations are defined using the constraint language *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 text fragment optional and include it only when the feature `WindSpeed` is not selected in pure::variants, the corresponding pvSCL rule is: `not(WindSpeed)`.

To add variability information to documents, two different approaches are used: In Word, special comments are added to documents. These comments contain text that constrains the commented text fragment. To identify these comments, the comment author and the comment initial is set as shown in [Table 1, "Properties of pure::variants Comments"](#).

**Table 1. Properties of pure::variants Comments**

	<b>Comment Author</b>	<b>Comment Initial</b>
Condition	p::v restriction	p::vr
Calculation	p::v calculation	p::vc

In Excel, special pure::variants rows and columns can be defined, which can contain pure::variants conditions. Calculations can be added to all cells of the workbook, except pure::variants condition rows or columns. They are

marked by brackets. For Example, in *The maximum allowed speed is [Speed->Max] km/h*, the calculation *[Speed->Max]* will be replaced with the value of attribute "Max" on feature "Speed" in the transformed variant. See [the section called "Calculations"](#) for more details on calculations in Excel documents.

To create variants of the master document, Variant Description Models (VDMs) have to be created in the pure::variants Connector for Microsoft Office project. Each VDM contains the feature selection for one project variant. The transformation of a project variant will create a variant of the Word or Excel input document in a specified output location. All optional parts with failing condition have been removed from this document variant, and all fragments annotated with a calculation have been replaced with the referenced value.

Next, we describe how variability can be added to both Word and Excel documents.

## 2.3. Adding Variability using the pure::variants Integration for Microsoft Office

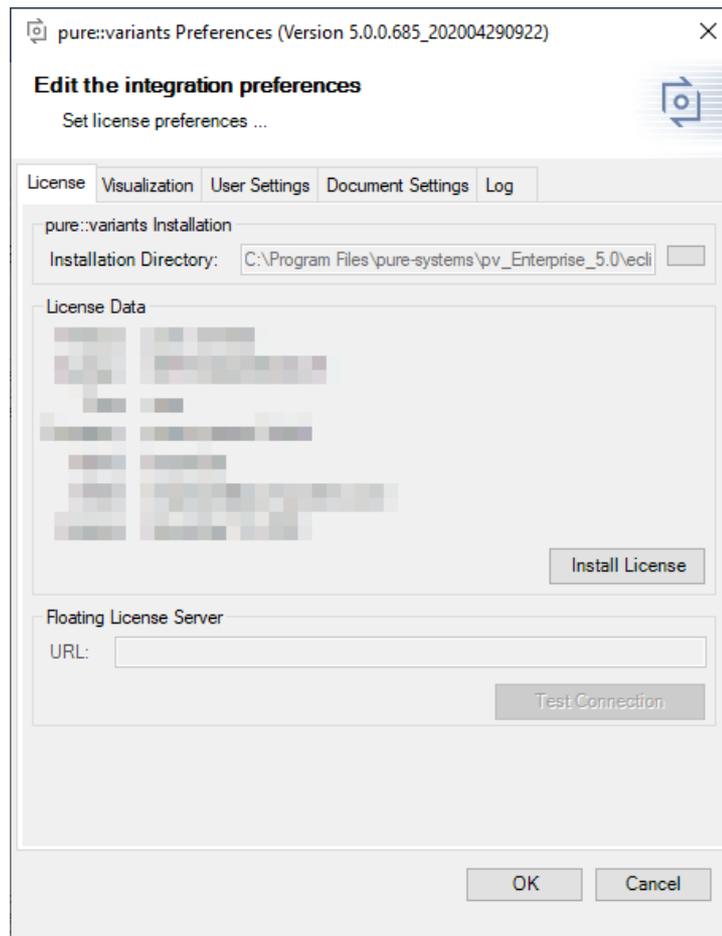
For adding variability information to both Word and Excel documents, you need the pure::variants Integrations for Microsoft Word and Excel, which are part of the pure::variants Integration for Microsoft Office installation. Both Integrations work basically the same. They provide a ribbon tab and a taskpane for loading and viewing pure::variants models. Using the ribbon tab you can also edit conditions and calculations and trigger visualizations for previewing variants and for finding errors in variability information more quickly. Next, we describe how to prepare both Integrations on first use and how to work with the pure::variants taskpane, which is the same in both Integrations.

### First Use

When you first use each Integration after installation, it is necessary to check whether the license preferences are correct. To this end, select the pure::variants tab on the ribbon and press the preferences button. A dialog opens that shows the path to your pure::variants installation and your license information (see [Figure 2, "Preferences Dialog"](#)). If any of the information is missing, you need to enter it. Use the ... button in the **pure::variants Installation** group to enter the installation directory, and the **Install License** button to specify your license.

If you are using a floating license and the URL in the **Floating License Server** group is not set already, you need to enter the URL. To test if the connection to the floating license server is established, press the button **Test Connection**.

Now you can use the Integration.

**Figure 2. Preferences Dialog**

## Using the pure::variants Taskpane

The pure::variants taskpane is the same in Word and Excel. You can show it by pressing the **Show** button on the pure::variants tab in the ribbon. Using the pure::variants taskpane, you can connect your document with pure::variants models, which are needed for editing variability information and viewing visualizations.

## Connecting with pure::variants Models

For editing variability information and viewing visualizations, it is necessary to connect your Microsoft Word or Excel project with one or more pure::variants models. The following types of pure::variants models can be loaded:

- Recommended: pure::variants configuration spaces, which enable selection of contained variant description models (.vdm)
- pure::variants variant result models (.vrm)
- pure::variants feature models (.xfm)
- pure::variants family models (.ccfm)

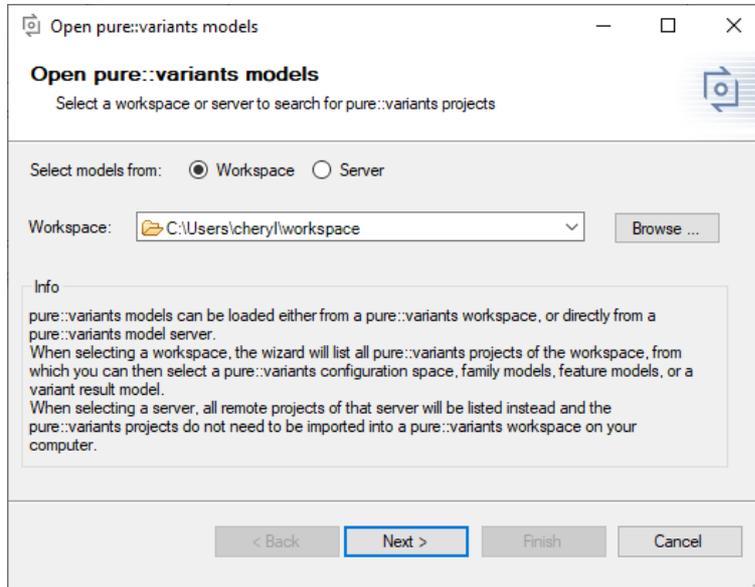
pure::variants models can be opened from two different sources: Either from a *pure::variants/Eclipse workspace* or from a *pure::variants model server*.

## Opening models from a workspace

To open a model or configuration space, press  on the pure::variants taskpane. This will open a wizard, which first allows choosing the source (workspace or server). Choose *workspace* and then browse to find your

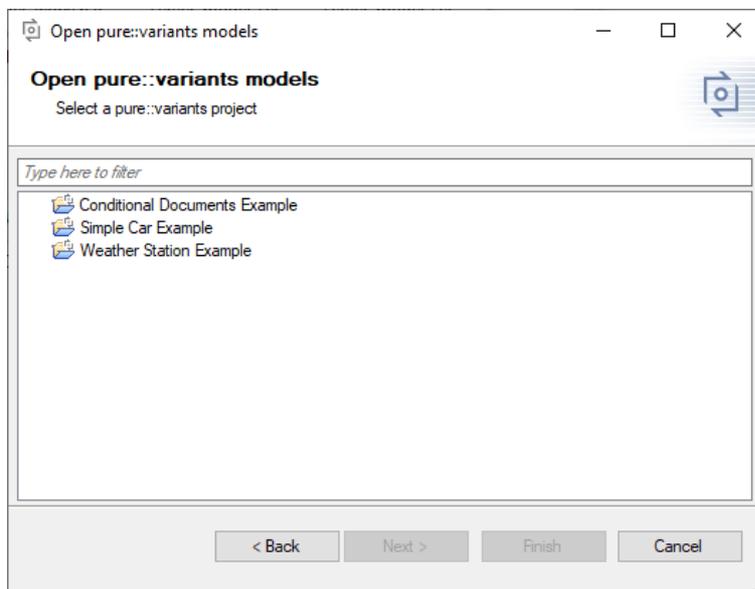
pure::variants workspace folder. Already known workspaces are listed in the workspace dropdown box. If you later need to add or remove a workspace from the list, you can go to tab *User Settings* of the Integration preferences (accessible via .

**Figure 3. Mode selection page**



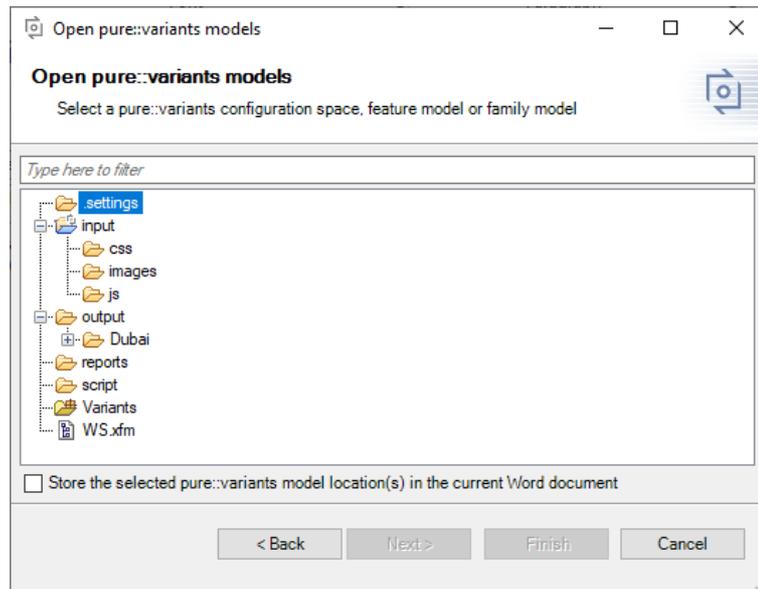
On the next page, all projects are listed that are located in the selected workspace folder or that are linked into the pure::variants/Eclipse workspace.

**Figure 4. Project selection page**



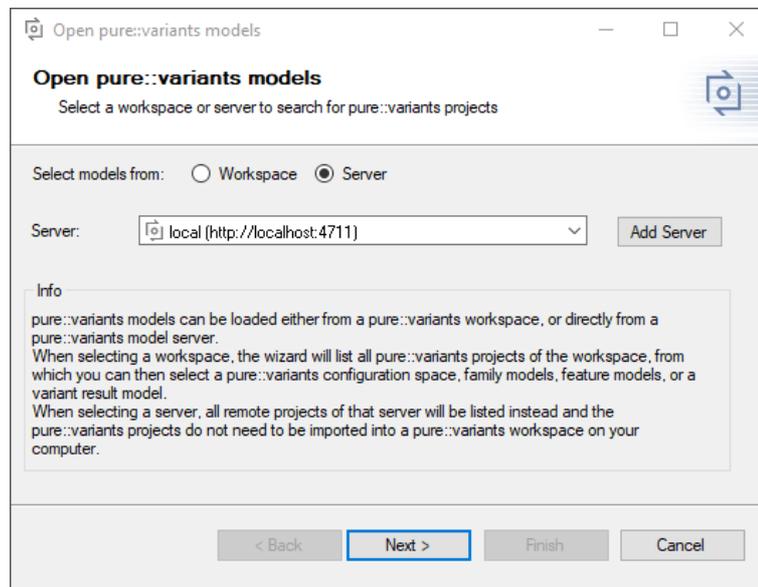
Select one and on the next page choose the model(s) or configuration space you want to open.

Checkbox *Store the selected pure::variants model location(s) in the current document* allows you to save the selected model locations in the current document, so that these models will be opened again once any user opens this document. If you do not select the checkbox, the model locations will only be stored on your computer. For details, see [the section called “Saving and Loading pure::variants Models”](#).

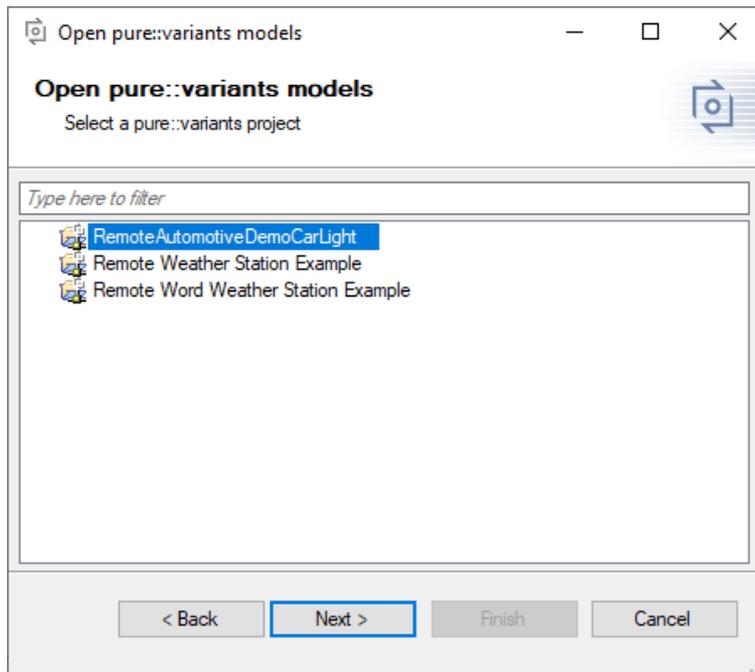
**Figure 5. Model selection page**

### Opening models from a model server

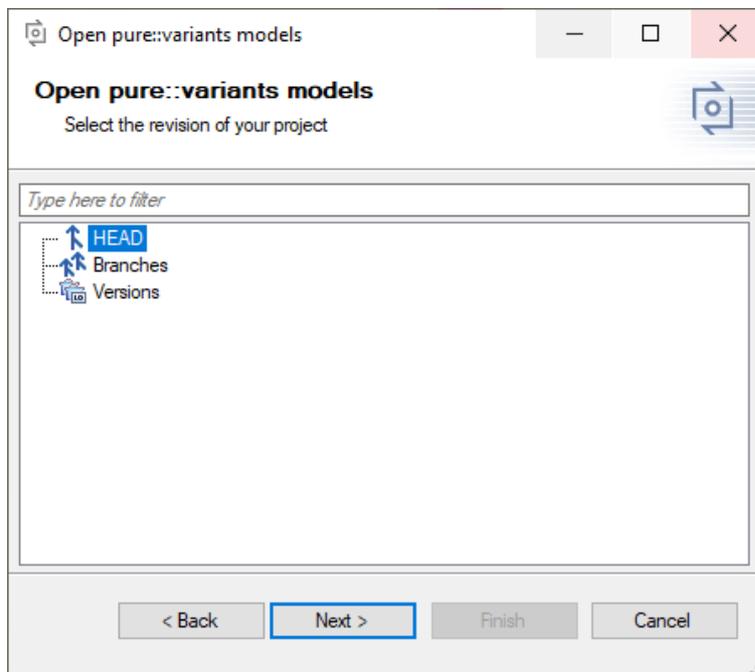
To open a model or configuration space directly from a pure::variants model server, also press . On the first page of the wizard, choose *server* and add the server address via button **Add Server**. Like in pure::variants, new servers need a name and the server address (e.g., <https://yourserveraddress:443>). Any known servers are listed in the server dropdown box. If you later need to add or remove a server from the list, open the Integration preferences by pressing . On tab *User Settings*, you can add or remove servers.

**Figure 6. Mode selection page**

On the next page of the wizard, all projects of the server that the current user has read access to are listed.

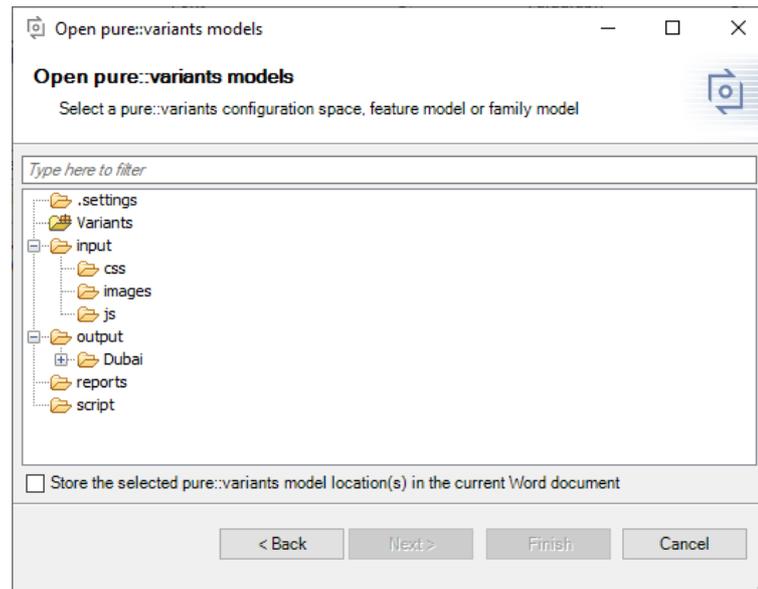
**Figure 7. Project selection page**

Select one and on the next page choose the project's revision (branch or tag) from which you want to load a model.

**Figure 8. Project selection page**

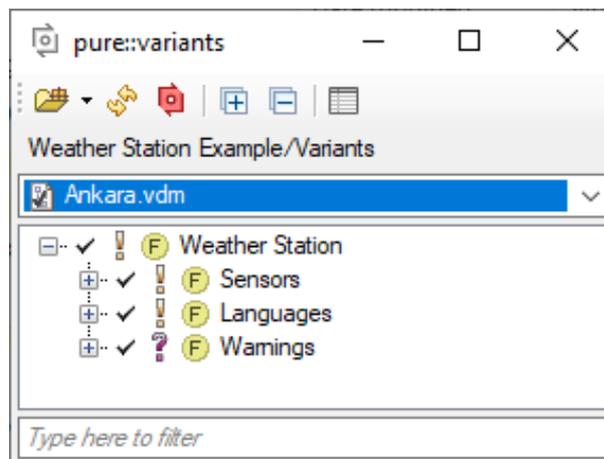
Finally, on the last page select the model(s) or configuration space you want to open.

Checkbox *Store the selected pure::variants model location(s) in the current document* allows you to save the selected model locations in the current document, so that these models will be opened again once any user opens this document. If you do not select the checkbox, the model locations will only be stored on your computer. For details, see [the section called "Saving and Loading pure::variants Models"](#).

**Figure 9. Model selection page**

### Opening pure::variants Configuration Spaces

To open a pure::variants configuration space, use the wizard as described above. On the last page, select a configuration space folder. Now the pure::variants taskpane should show all used models of your configuration space. Please note that family models (.ccfm) are not opened per default. You can enable loading family models in the Integration preferences on the **Visualization** tab. After selecting a variant from the dropdown list, selections should be shown in front of features. To ease usage of configuration spaces with many variants, the latest opened variants are shown at the top of the list.

**Figure 10. Configuration Space with Selected Variant**

### Opening Other pure::variants Models

Other pure::variants models, such as variant result models<sup>1</sup> (.vrm), feature models (.xfm), and family models (.ccfm) can also be opened via . Please note that family models (.ccfm) are not listed per default. You can enable loading family models in the Integration preferences on the **Visualization** tab.

<sup>1</sup>You can create a variant result model in pure::variants by clicking the **Save Result to File** button that is shown in the toolbar of a variant description model.

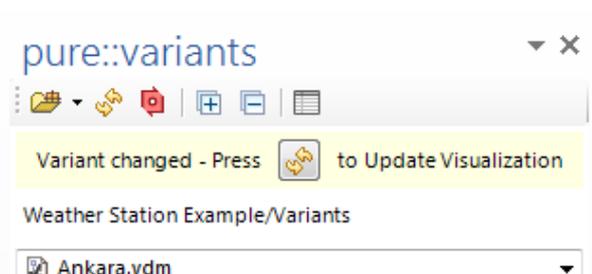
## Live Connection with pure::variants

Since pure::variants 4.x, changes of the loaded pure::variants models are propagated live to the Integration. For example, directly after editing the name or changing the selection of a feature the loaded models are updated in the pure::variants taskpane. To enable this live update, the following prerequisites need to be fulfilled:

- the opened model needs to be located in an Eclipse workspace
- the changes have to be done on the same Eclipse workspace using pure::variants 4.x or later
- either a configuration space, feature model or family model needs to be loaded (Variant result models can only be updated automatically when the .vrm file is saved)

If a visualization is active when a loaded model has changed, a pane is shown that informs you about a pending visualization update (see [Figure 11, “Information about Pending Visualization Update”](#)). When pressing the pane's refresh button, the visualization is updated.

**Figure 11. Information about Pending Visualization Update**



When the used models of a configuration space have changed or a new variant model was added to the configuration space, a live update of the currently loaded models is not possible. In this case, you can press  to manually reload all pure::variants models and refresh the current visualization. To unload all models and free the pure::variants license, press .

## Saving and Loading pure::variants Models

To ease the work with pure::variants Integrations, the last loaded model locations are saved, so that the model will be opened again automatically, next time you start the tool. Per default, these model locations are saved only for you on your local machine. If you want to save the last loaded model locations for all users, who are using a certain document, you can select checkbox *Store the selected pure::variants model location(s) in the current document* on the last page of the open model wizard. Then everytime a user opens that document, the pure::variants models stored in the document will be opened instead of the locally stored models (if they can be found on the user's machine).

Furthermore, a list of the latest loaded models can be accessed via the small arrow next to the  button.

Please note that models are saved relative to your current workspace, which is the workspace where your current model is located in or linked to. Therefore, you may be asked for your current workspace location when loading a model from a different workspace, or a model that is not located in a workspace (but may be linked into a workspace). Hint: If you want to know where exactly the loaded model is located, you can hover over the name of the model. A tooltip will show the full path of your currently loaded configuration space or model.

If you do not want to load a model again on startup, you can clear the stored model locations from the user and/or document settings. For details, see [the section called “Manage Settings”](#).

## Removing Old Model References

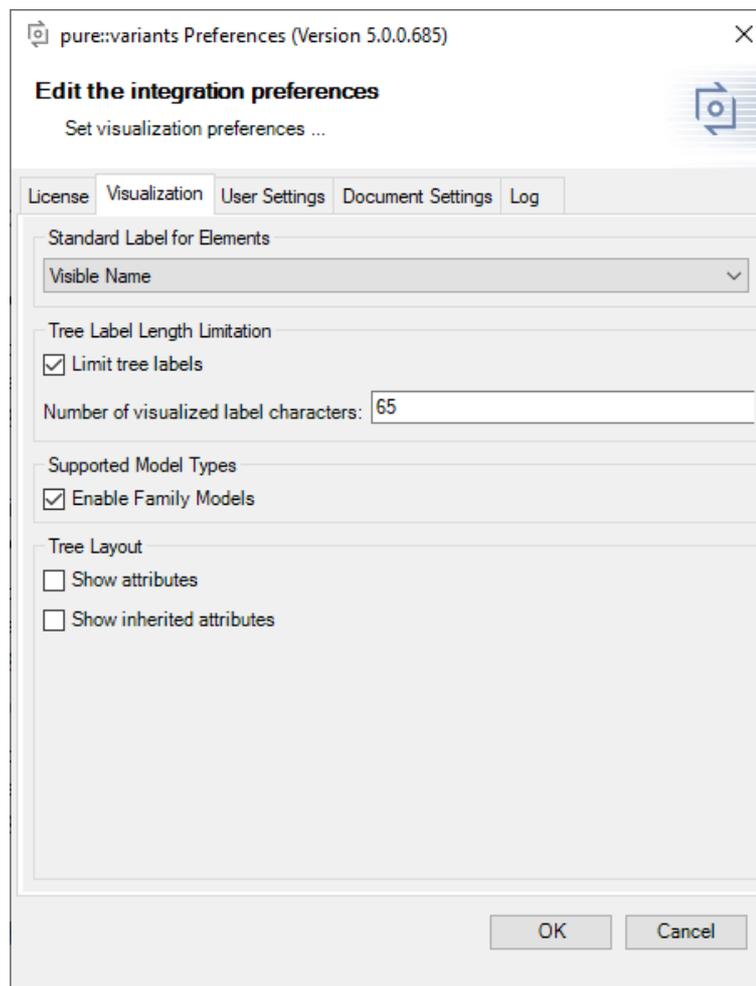
Since the loading of models has changed extensively in pure::variants Integrations of version 4.x, Integrations now save pure::variants models differently in the current document than they did in version 3.2.x. Therefore, references

to models loaded with version 3.2.x may still exist in your document, but are not used. This may be as intended if you still also need to open the document with an Integration of version 3.2.x. However, if this is not necessary you can remove these references from your document: Open the Integration preferences and press the **Remove References** button in the lower part of the **Log** tab (see [Figure 14, “Preferences Dialog Log Tab”](#)).

## Model Visualization Preferences

In the Integration preferences, you can set how pure::variants models will be displayed and which model types are supported. To do that, open the Integration preferences by pressing  and go to the **Visualization** tab (see [Figure 12, “Preferences Dialog Visualization Tab”](#)). The first dropdown box enables you to set how elements in the pure::variants model view are labeled. Furthermore, you can limit how many characters are shown for each element in the tree, enable or disable the loading of family models, and set whether attributes are shown in the model tree. To also show attributes inherited from parent elements, select **Show inherited attributes**.

**Figure 12. Preferences Dialog Visualization Tab**



## Manage Settings

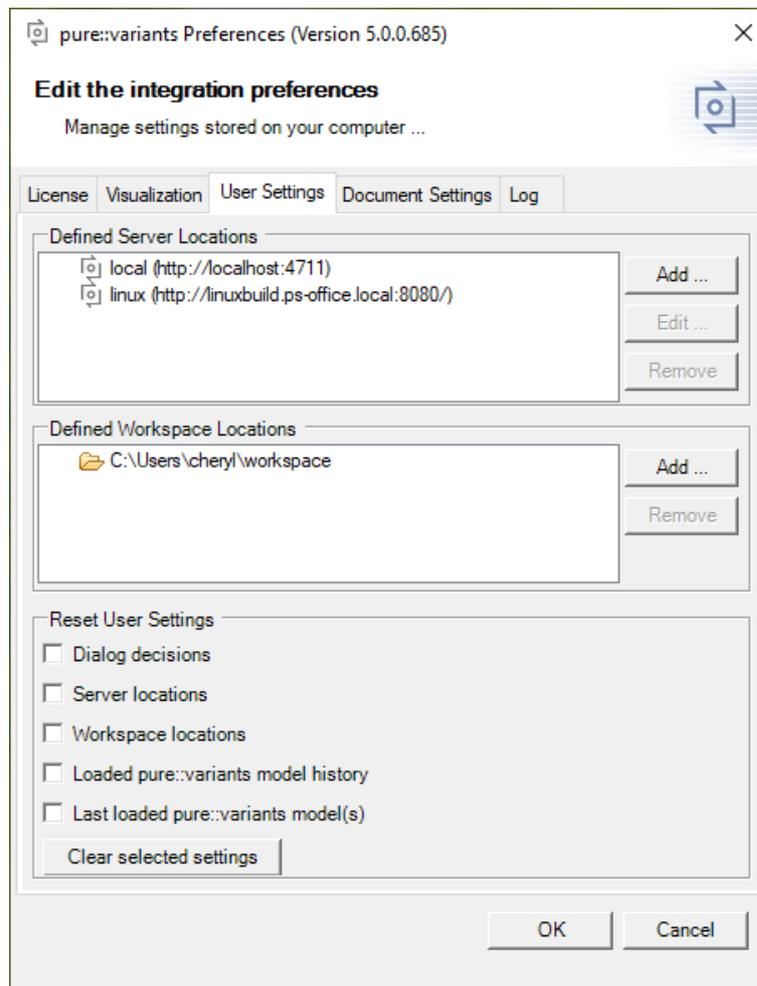
In the Integration preferences, you can also manage settings stored by the Integration. You can, for example, add or remove known workspace and server locations or clear certain settings. To do that, open the Integration preferences by pressing .

Here, you can find tabs *User Settings* and *Document Settings*.

On tab *User Settings*, you can manage settings that are stored only on your local machine, specifically for your user. (see [Figure 13, “Preferences Dialog User Settings Tab”](#)). This includes server and workspace locations,

dialog decisions, the history of previously loaded pure::variants models, and so on. The first section, **Defined Server Locations** enables you to add, edit and remove the server locations that are not locked. The second section, **Defined Workspace Locations** enables you to add and remove the workspace locations. The last section **Reset User Settings** enables you to clear the selected settings. For example, you can select checkbox *Last loaded pure::variants model(s)* and press the clear button, to make sure no pure::variants model is loaded from the user settings at startup of the Integration.

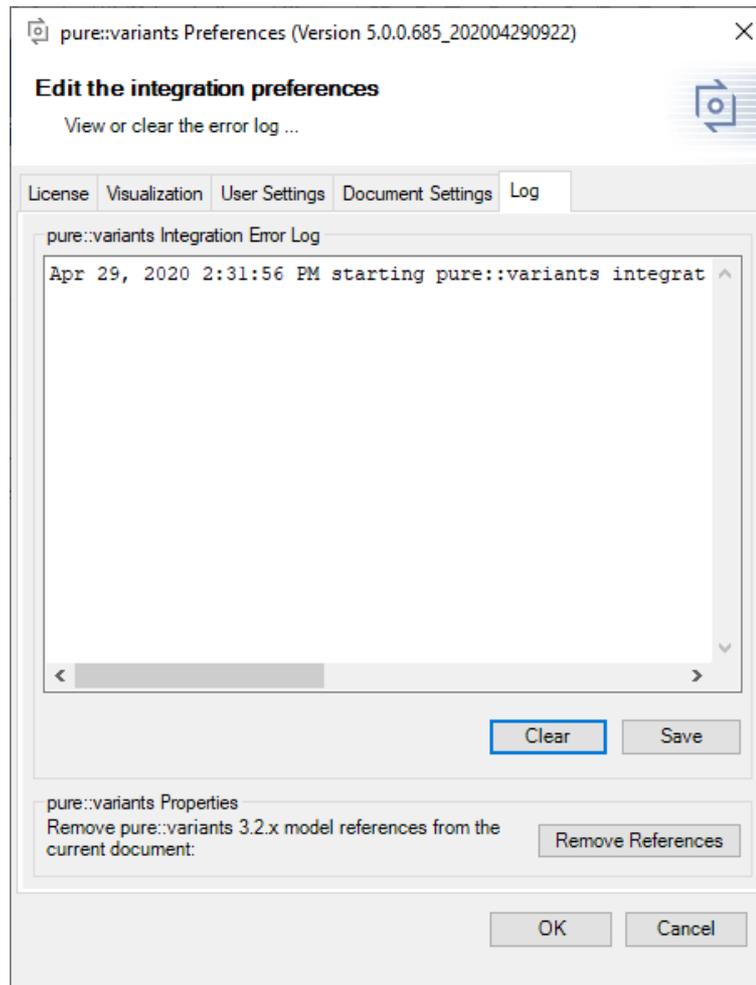
**Figure 13. Preferences Dialog User Settings Tab**



On tab *Document Settings* you can manage all settings that are stored in the current document. For example, you can clear the last loaded pure::variants model from the current document, so that, when you or other users next open the current document, the Integration will load no pure::variants model stored in the document.

## Troubleshooting

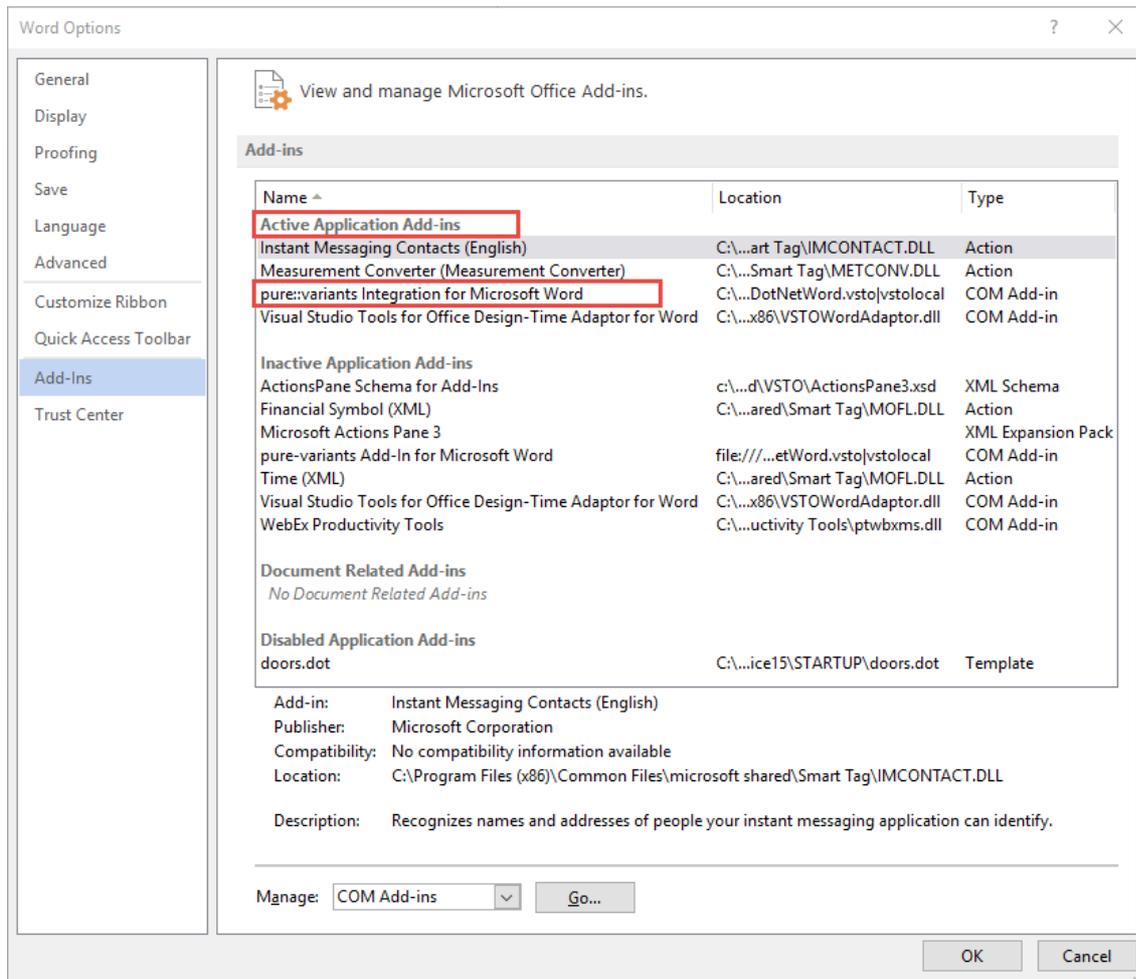
If the Integration does not behave as expected, it may be useful to check its log file. You can find it in the Integration preferences (📄) on the **Log** tab (see [Figure 14, “Preferences Dialog Log Tab”](#)). It also enables you to save it to your disk or clear the contents of the log file. Furthermore, the **Log** tab provides the option to remove references to pure::variants models that can only be read by Integrations of version 3.2.x (see [the section called “Removing Old Model References”](#)).

**Figure 14. Preferences Dialog Log Tab**

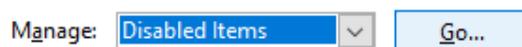
### What to do when the pure::variants Integration Tab is not Shown

When the Word or Excel integration tab is not shown, it may have been disabled by Word or Excel. Please follow these steps to troubleshoot:

In Word/Excel navigate to **File->Options->Add-Ins**. In the dialog, check whether pure::variants Integration for Microsoft Word/Excel is listed under "Active Application Add-Ins", like this:

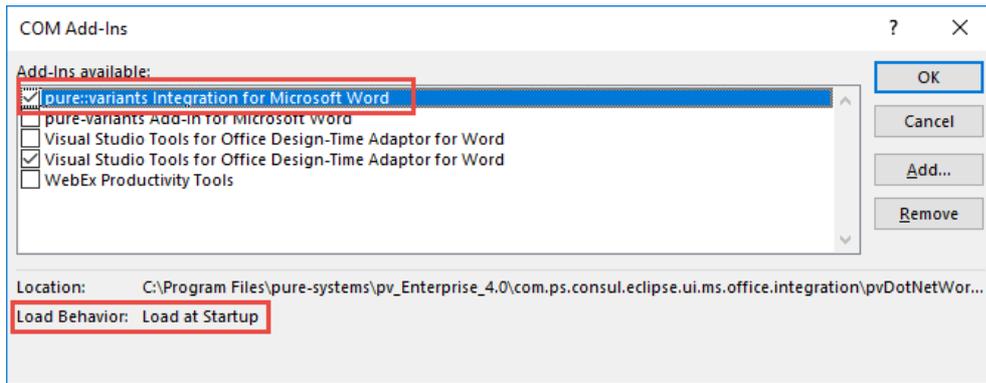
**Figure 15. View and Manage Add-Ins**

If not, use the bottom-most dropdown box, select "Disabled Items" and then click on "Go". If the integration is listed here, enable it. If the integration is not listed in the disabled item menu, continue to the next step.

**Figure 16. Disabled Items**

Use the same dropdown box to open the "COM Add-ins" dialog. In that dialog, make sure that the integration is selected and the load behavior says "Load at Startup". Then click on "OK".

**Figure 17. COM Add-Ins**



Now the pure::variants Integration Tab should be visible.

## Known Issues

The following known issue exist:

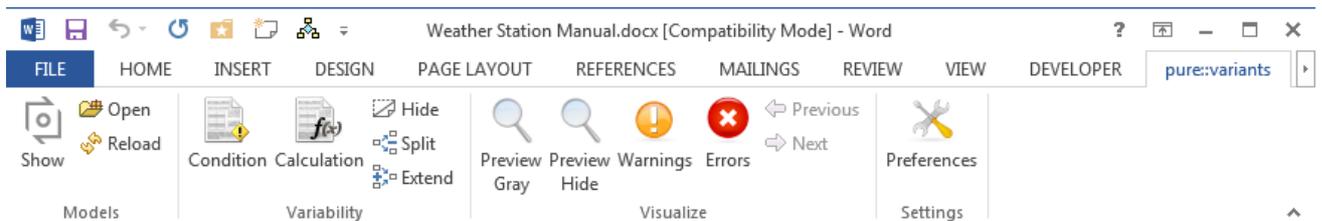
- If "Track Changes" is enabled, we cannot differentiate between already deleted conditions/calculations and normal ones. So we transform also the deleted conditions/calculations as usual. Thus, for example, calculation text is replaced inside the text that is marked as deleted.

## 2.4. Using the pure::variants Integration for Microsoft Word

### The pure::variants Ribbon Tab

To use the pure::variants Integration for Microsoft Word, select the **pure::variants** tab on the ribbon (see [Figure 18, "pure::variants Ribbon Tab"](#)). There are four groups of functions: *Models* for opening, reloading and showing pure::variants models; *Variability* for adding variability information to the Word document; *Visualize* for previewing variants of the document and finding errors in the variability information; and *Settings* for accessing the Integration preferences.

**Figure 18. pure::variants Ribbon Tab**



To show the pure::variants taskpane, press the button **Show** in the *Models* group. The following table gives an overview of the Integration's functionality:

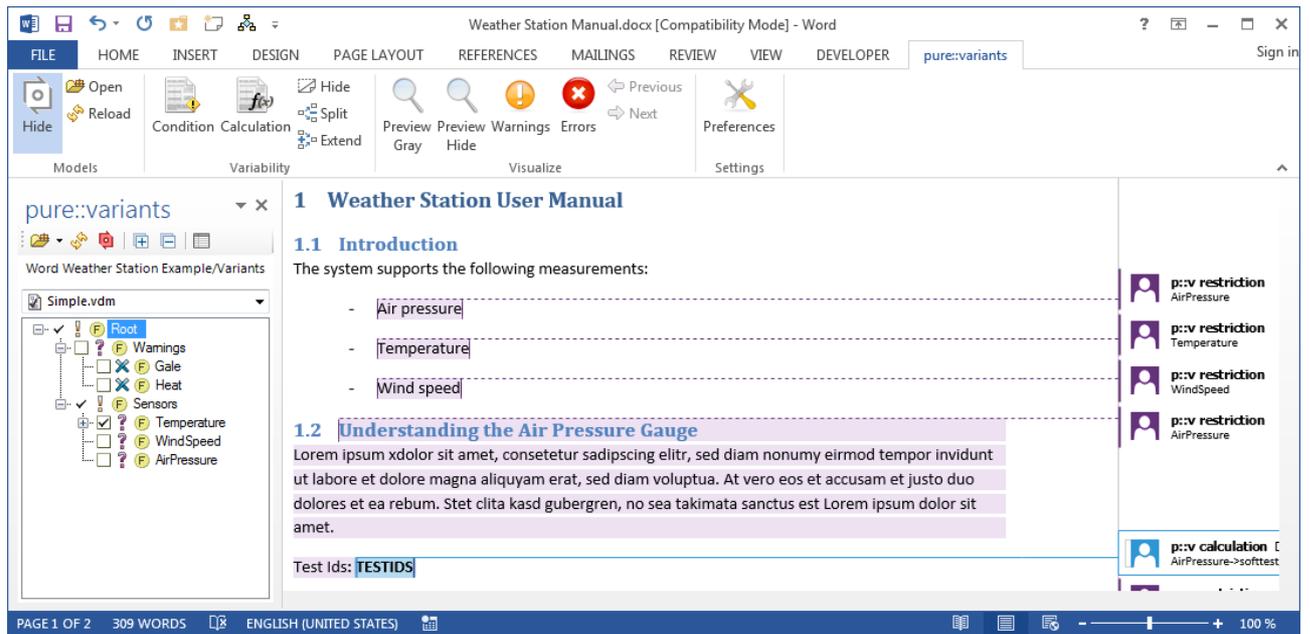
**Table 2. User Interface Elements**

Image	Description
	Shows or hides the pure::variants taskpane.
	Opens a configuration space or other pure::variants models (., .xfrm, or .ccfm). See <a href="#">the section called "Using the pure::variants Taskpane"</a> for details.

Image	Description
 Reload	Reloads all loaded pure::variants models and refreshes the visualization.
 Condition	Adds a condition to the current text selection (see <a href="#">the section called “Conditions”</a> ).
 Calculation	Adds a calculation to the current text selection (see <a href="#">the section called “Calculations”</a> ).
 Hide	Hides all variability information.
 Split	Splits the commented area of an existing condition or calculation at the current text cursor location. The text cursor must be inside exactly one commented area.
 Extend	Extends a commented area of an existing condition or calculation with the current selection. If the current selection overlaps with exactly one commented area, the overlapped area is extended with the selection.
 Preview Gray	Preview visualization, which grays out all elements that would not be included in a variant produced with the currently loaded variant model (see <a href="#">the section called “Preview Visualizations”</a> ).
 Preview Hide	Preview visualization, which hides all elements that would not be included in a variant produced with the currently loaded variant model (see <a href="#">the section called “Preview Visualizations”</a> ).
 Warnings	Visualization that highlights all conditions and calculations that contain warnings in the pvSCL expression, such as unknown names of features or attributes (see <a href="#">the section called “Error Visualizations”</a> ).
 Errors	Visualization that highlights all conditions and calculations that contain errors in the pvSCL expression (see <a href="#">the section called “Error Visualizations”</a> ).
 Previous	Jump to previous faulty condition or calculation.
 Next	Jump to next faulty condition or calculation.
 Preferences	Opens the preferences dialog (see <a href="#">the section called “First Use”</a> ).

## Editing Variability

To edit variability information of a Word document, the Integration provides an editor, which features auto-completion, syntax highlighting and checking for errors. There are two types of variability information that can be added to a Word document: *Conditions* and *Calculations*. In [Figure 19, “Variability”](#) both types are shown: Blue comments noted "p::vr" are conditions, whereas red comments noted "p::vc" are calculations (Note that the comment colors may change for each session, since they are generated automatically by Word). Both conditions and calculations are written in the pure::variants expression language pvSCL. For more information on pvSCL, consult the pure::variants User's Guide, or press the help button in the lower left corner of the condition or calculation editor.

**Figure 19. Variability**

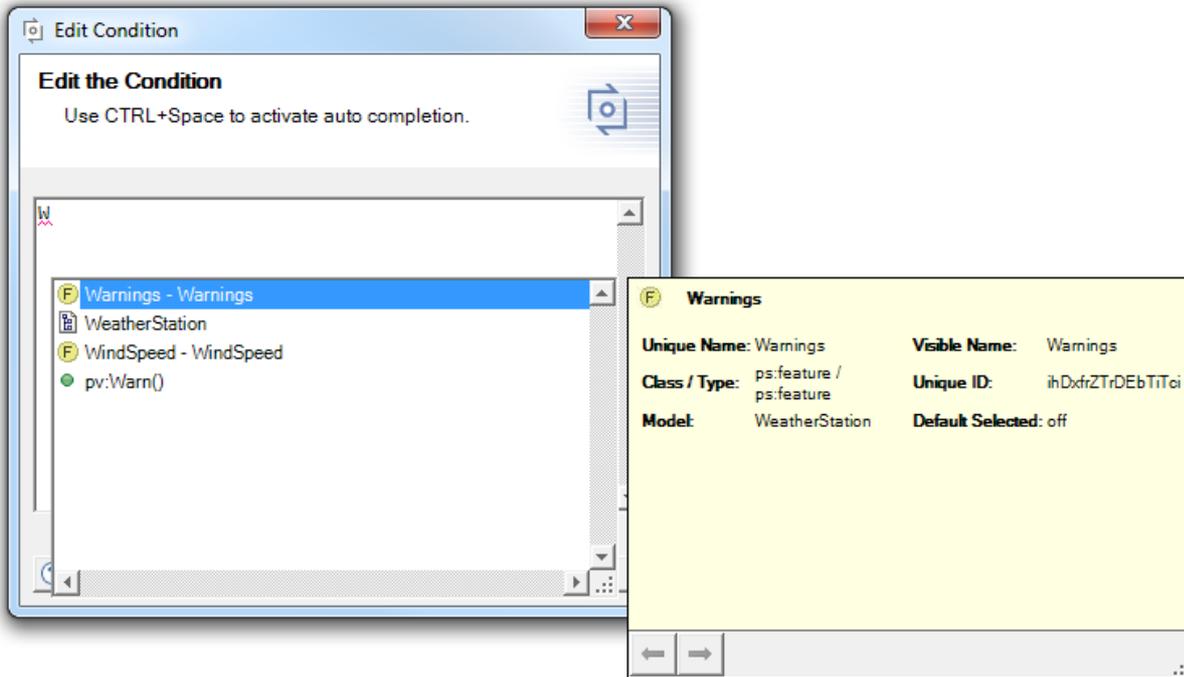
## Conditions

Conditions are pvSCL expressions that return a boolean value. This value is then used to decide if the annotated text fragment is included in the resulting variant or not.

To annotate a text fragment with a condition, select the desired text and press the **Condition** button. An editor dialog opens, which provides simple autocompletion and syntax highlighting. By pressing CTRL + space possible completions for the typed word are offered (see [Figure 20, “pure::variants Condition Editor”](#)).

When you are done editing and press OK, the entered expression is checked for errors. Errors in pvSCL expressions are reported if the expressions's syntax is not pvSCL compliant, or if a referenced element is not part of the loaded pure::variants models. Unknown elements are highlighted in red.

If you want to edit the condition later on, select its comment bubble and double-click it to show the editor again.

**Figure 20. pure::variants Condition Editor**

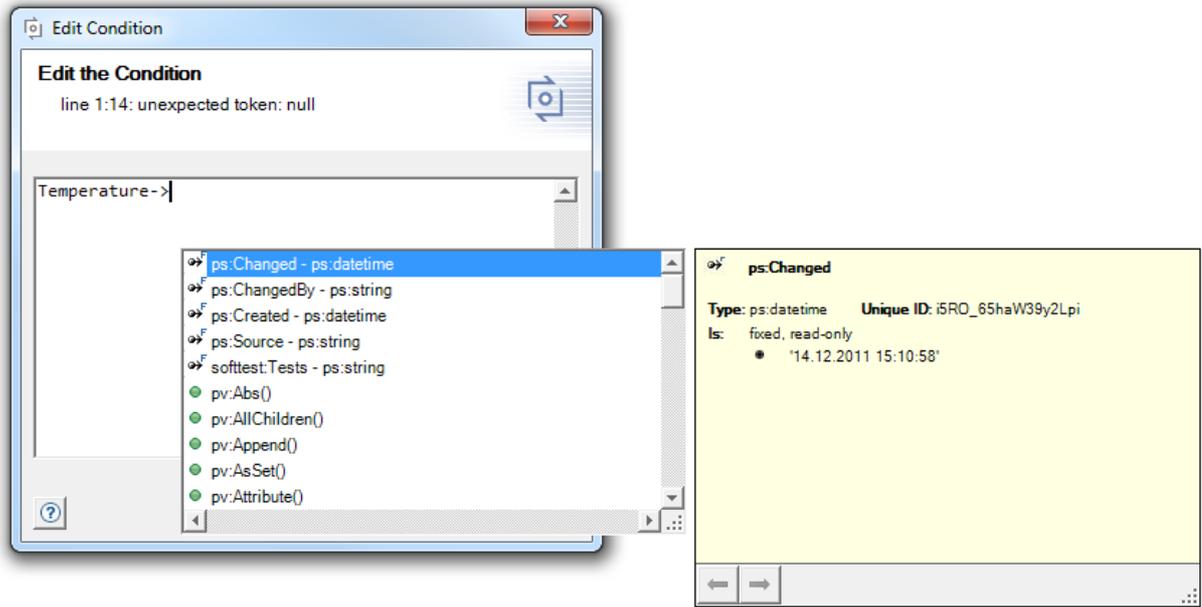
## Calculations

Calculations are pvSCL expressions that return a value, which replaces the commented text during transformation. For example, you can reference an attribute from the loaded model (e.g., `Temperature->softtest:Tests`), or you can compute a string value based on the selected features (e.g., `IF (Temperature AND NOT(WindSpeed) AND NOT(AirPressure)) THEN 'Thermometer' ELSE 'Weather Station' ENDIF`). For more examples of pvSCL expressions, consult the pure::variants User's Guide, or press the **Help** button in the lower left corner of the calculation editor.

To annotate a text fragment with a calculation, select the desired text and press the **Calculation** button. An editor dialog opens, similar to the editor for calculations. Here you can reference an attribute as you can see in [Figure 21, "pure::variants Calculation Editor"](#).

When you are done editing and press OK, the entered expression is checked for errors. If you want to edit the pvSCL rule later on, select its comment bubble and double-click it to show the editor again.

**Figure 21. pure::variants Calculation Editor**



**Note**

Use the **Split** and **Extend** buttons to modify which text fragment is related to an existing condition or calculation. The **Split** button splits the commented area of a condition or calculation at the current text cursor position. The **Extend** button extends the commented area of an existing condition or calculation with the current selection (if the selection overlaps with the commented area).

**Supported Word Document Elements**

Apart from text, Word documents can contain elements of different types, such as smart art, cross references, or images. In [Table 3, “Supported Word Document Elements”](#), we list to which elements you can add conditions and calculations. Except for the main text, you can only add conditions or calculations to the entire element. For instance, adding a condition to an entire smart art object is possible, but not to a text fragment within the smart art object.

**Table 3. Supported Word Document Elements**

Element Type	Add Condition or Calculation	Comments
Cross Reference	x	
Diagram	x	
Equation	x	
Excel Table	x	
Footer		
Footnote	x	in footnote itself no conditions or calculations possible
Header		
Hyperlink	x	
Image	x	
Reference Tables (e.g., Table of Contents, Index, etc.)	x	
Shape		

Element Type	Add Condition or Calculation	Comments
Smart Art	x	
Table	x	only single cells, the complete table, or a complete row can be commented
Text	x	only in the main part of the document
WordArt	x	

## Visualizations

After variability information has been added to the document, you can use several visualizations to check whether the variability information is correct: Either preview variants for loaded variant models or highlight errors in variability information. To ensure that during visualization no changes are made to your original document, the visualization is applied to a read-only copy of your document, which opens when you press a visualization button.

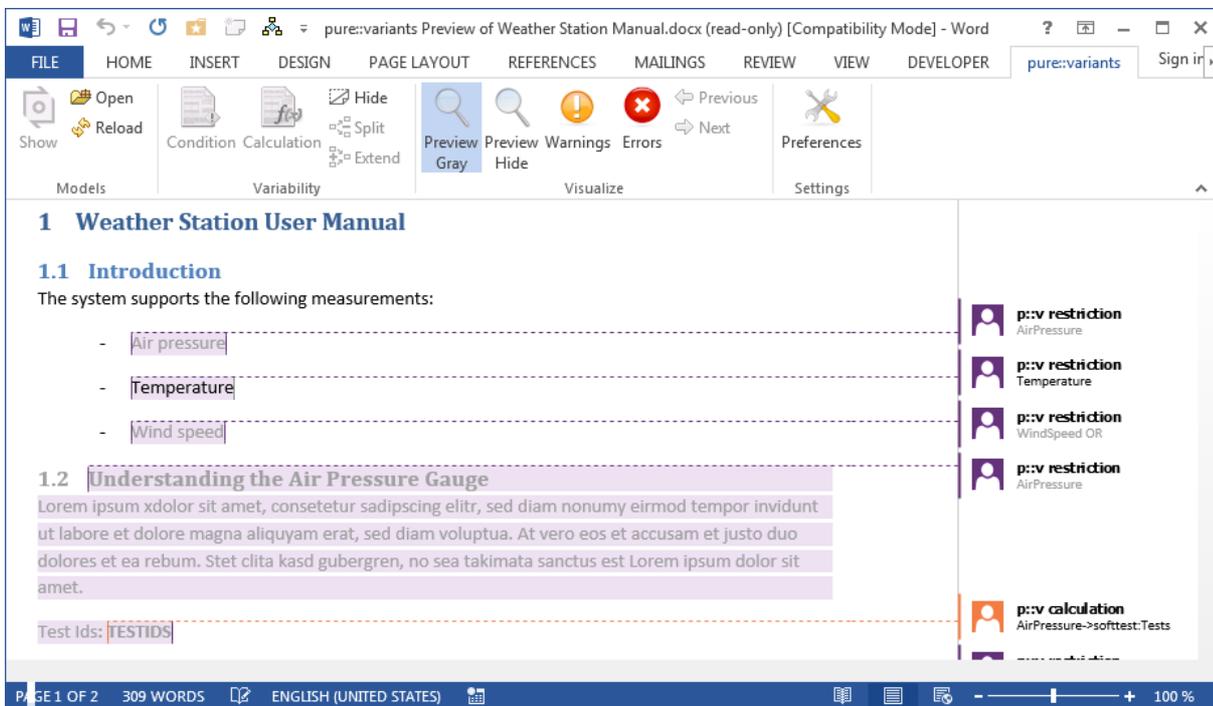
### Note

Visualizations are only supported for non-restricted documents. To remove possible restrictions from your document use the **Restrict Editing** button on the Word **Review** tab.

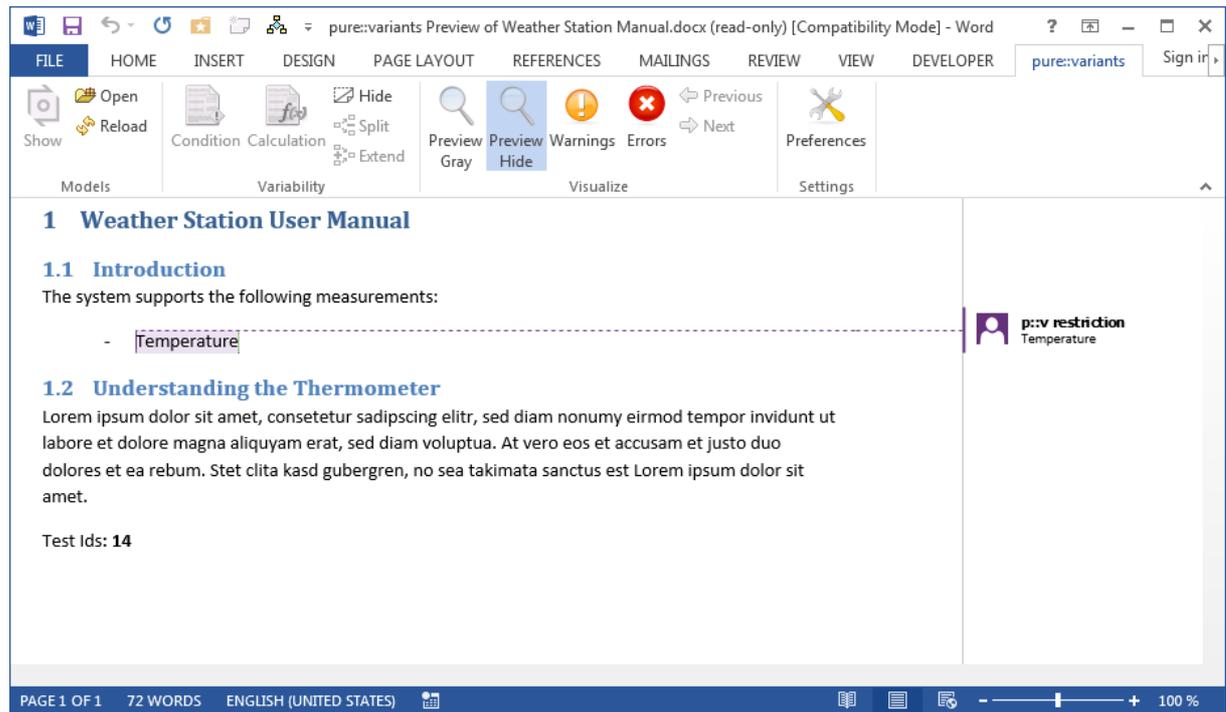
## Preview Visualizations

To preview variants, it is necessary to load a first (see the section called “Using the pure::variants Taskpane”). Figure 22, “Preview - Gray” shows the result of a gray preview: All annotated text fragments that would not appear in a variant are grayed out, and the calculation is replaced with the value of the referenced attribute.

Figure 22. Preview - Gray

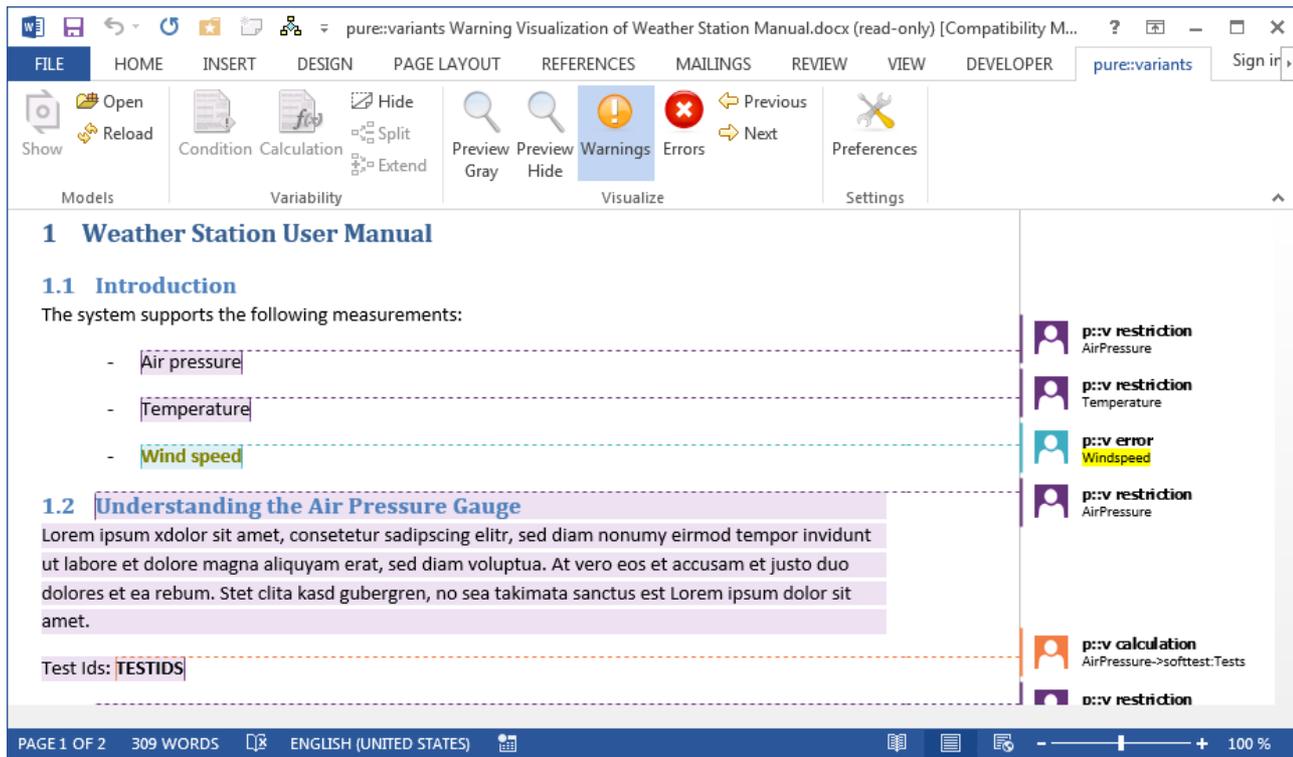


The preview - hide visualization works similar to the preview - gray visualization. The only difference is that it deletes all text fragments that would not be included in the final variant. Unlike during transformation, the pure::variants comments are not deleted.

**Figure 23. Preview - Hide**

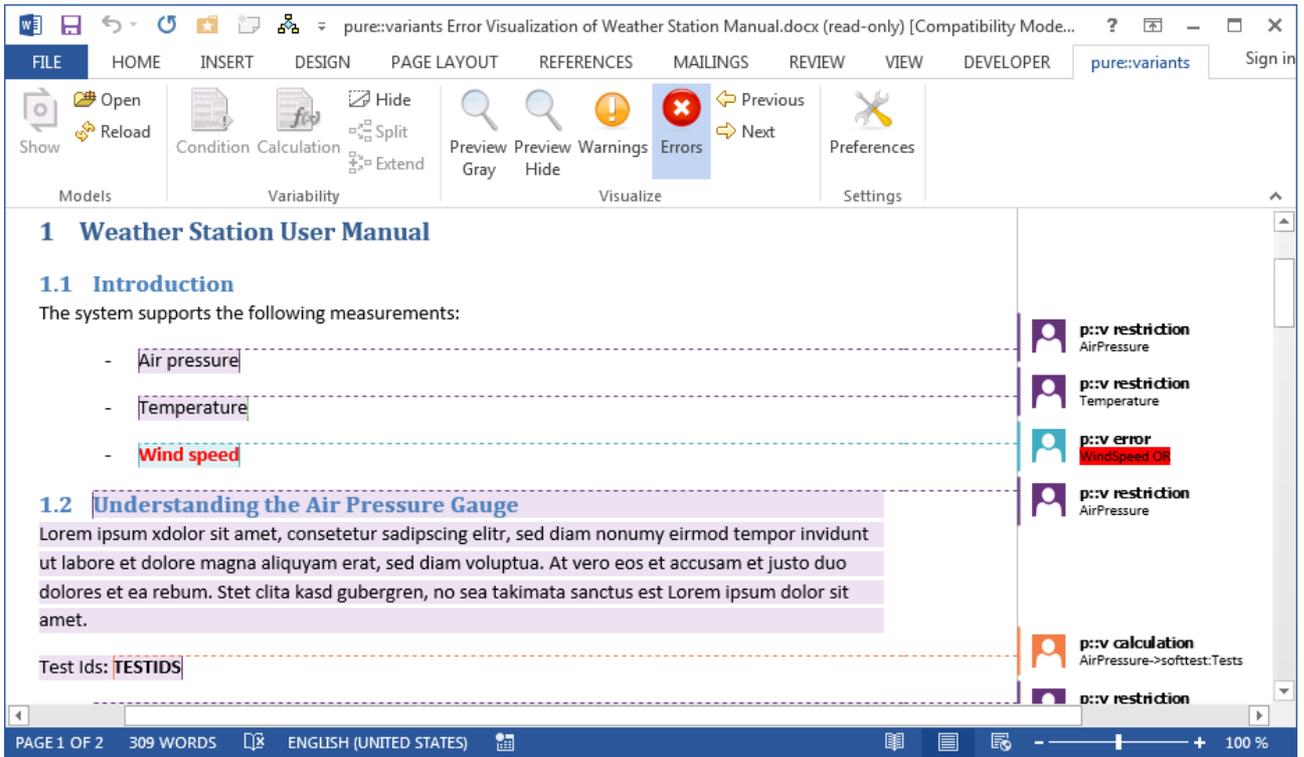
## Error Visualizations

Variability information in a Word document can contain errors. To find these errors, the Integration provides two visualizations: The *Warning Visualization* and the *Error Visualization*. Using the Warning Visualization, you can highlight problems that result from incorrectly typed or missing element names (e.g., feature or attribute names). To enable or disable it, press the button **Warnings**. For example in [Figure 24, "Warning Visualization"](#) the feature "Windspeed" does not exist, since it should be spelled "WindSpeed". Therefore, the condition is highlighted. The condition's text is highlighted in yellow in the comment bubble and the commented text is shown in ochre. Moreover the comment color changes compared to non-highlighted comments, but since it is generated automatically by Word the color is not necessarily the same in every session.

**Figure 24. Warning Visualization**

Using the Error Visualization, you can highlight all pvSCL expressions that contain pvSCL errors. To enable or disable it, press the button **Errors**. For example, in [Figure 25, “Error Visualization”](#) the expression "WindSpeed OR" is incorrect, because the OR keyword requires a second argument. Therefore, the condition is highlighted. Both the conditions that contain errors and the annotated text are highlighted in red. Moreover, the comment color changes compared to non-highlighted comments, but as it is generated automatically by Word the color is not necessarily the same in every session.

**Figure 25. Error Visualization**



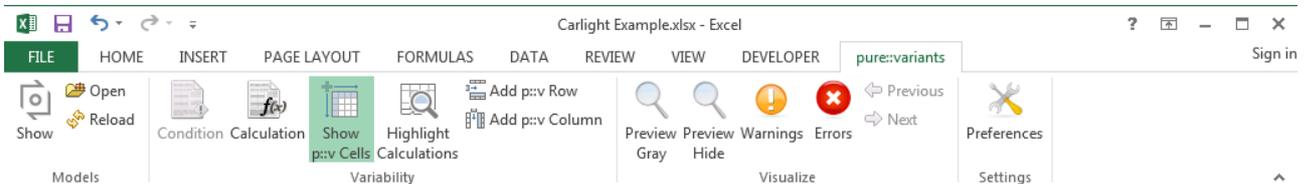
To better find errors, the buttons **Previous** and **Next** provide a way to jump to the last or next faulty condition or calculation.

## 2.5. Using the pure::variants Integration for Microsoft Excel

### The pure::variants Ribbon Tab

To use the Integration, select the **pure::variants** tab on the ribbon (see Figure 26, “pure::variants Ribbon Tab”). The basic structure is the same as in Word: There are four groups of functions: *Models* for opening, reloading and showing pure::variants models; *Variability* for adding variability information to the Word document; *Visualize* for previewing variants of the document and finding errors in the variability information; and *Settings* for accessing the Integration preferences.

**Figure 26. pure::variants Ribbon Tab**



To show the pure::variants taskpane, press the button **Show** in the *Models* group. The following table gives an overview of the Integration's functionality:

**Table 4. User Interface Elements**

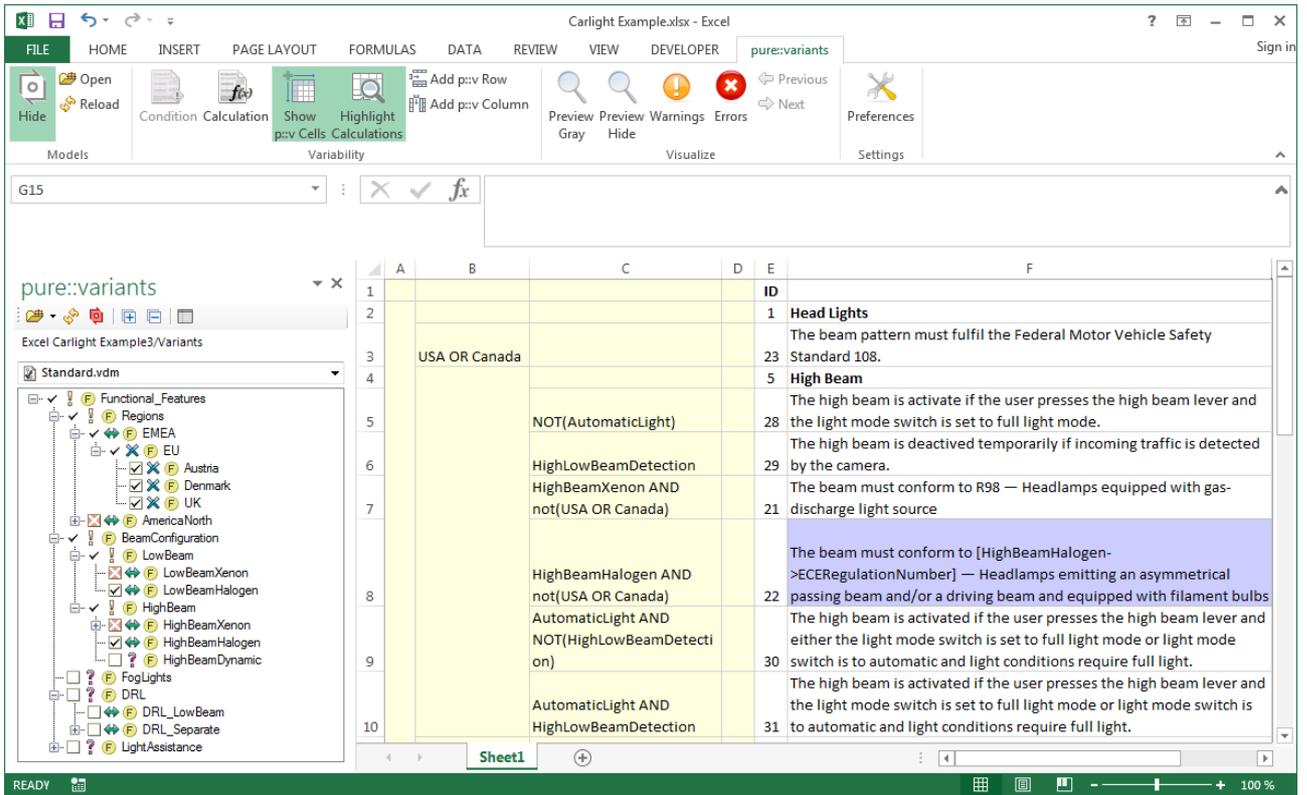
Image	Description
	Shows or hides the pure::variants taskpane.

Image	Description
 Open	Opens a configuration space or other pure::variants models (., .xfm, or .ccfm). See <a href="#">the section called “Using the pure::variants Taskpane”</a> for details.
 Reload	Reloads all loaded pure::variants models and refreshes the visualization.
 Condition	Adds a condition to the current cell selection (see <a href="#">the section called “Conditions”</a> ).
 Calculation	Adds a calculation to the current cell selection (see <a href="#">the section called “Calculations”</a> ).
 Show p::v Cells	Shows or hides all pure::variants condition rows and columns. This also resets the style of the pure::variants condition rows and columns to "p::v Condition Cells".
 Highlight Calculations	Highlights all cells containing calculations.
 Add p::v Row	Adds a pure::variants condition row.
 Add p::v Column	Adds a pure::variants condition column.
 Preview Gray	Preview visualization, which grays out all elements that would not be included in a variant produced with the currently loaded variant model (see <a href="#">the section called “Variant Visualizations”</a> ).
 Preview Hide	Preview visualization, which hides all elements that would not be included in a variant produced with the currently loaded variant model(see <a href="#">the section called “Variant Visualizations”</a> ).
 Warnings	Visualization that highlights all conditions and calculations that contain semantic errors in the pvSCL expression, such as unknown names of features or attributes (see <a href="#">the section called “Error Visualizations”</a> ).
 Errors	Visualization that highlights all conditions and calculations that contain syntactic errors in the pvSCL expression (see <a href="#">the section called “Error Visualizations”</a> ).
 Previous	Jump to previous faulty condition or calculation on the current worksheet.
 Next	Jump to next faulty condition or calculation on the current worksheet.
 Preferences	Opens the preferences dialog (see <a href="#">the section called “First Use”</a> ).

## Editing Variability

To edit variability information of an Excel workbook efficiently and less error-prone, the Integration provides an editor, which features autocompletion, syntax-highlighting and checking for errors. As in Word there are two types of variability information that can be added to a Word document: *Conditions* and *Calculations*. In [Figure 27, “Variability”](#) both types are shown: The columns with a yellow background contain pure::variants conditions, and the cells with a blue background contain calculations. Both conditions and calculations are written in the pure::variants expression language pvSCL. For more information on pvSCL, consult the pure::variants User's Guide, or press the help button in the lower left corner of the condition or calculation editor.

Figure 27. Variability



## Conditions

Conditions are pvSCL expressions that return a boolean value. This value is then used to decide if the annotated text fragment is included in the resulting variant or not. In Excel, conditions can be assigned only to entire rows and entire columns. To annotate a row with a condition, you first need to insert a pure::variants condition column. Use the **Add p::v Column** button to do this. The column is always inserted left of the current cell selection, or at the start of the worksheet, if the selection is invalid. For adding conditions to columns, you need to use the **Add p::v Row** button instead.

## Note

pure::variants columns and rows are always created with the *p::v Condition Cells* style set. If you want to change their color, you can edit the style.

You can define multiple columns or rows in a worksheet. Thus, multiple conditions can apply to one column or row. During transformation these conditions are evaluated hierarchically from left to right (for pure::variants columns), or from top to bottom (for pure::variants rows). See [Figure 28, “pure::variants Condition Hierarchy”](#) for an example: If the feature *LightAssistance* is unselected in a variant description model, all other pure::variants condition columns will not be evaluated. All of the shown cells will be deleted in the generated variant. If all features except *CorneringStaticLights* are selected in a variant description model, only the rows annotated with *CorneringStaticLights* will be deleted.

**Figure 28. pure::variants Condition Hierarchy**

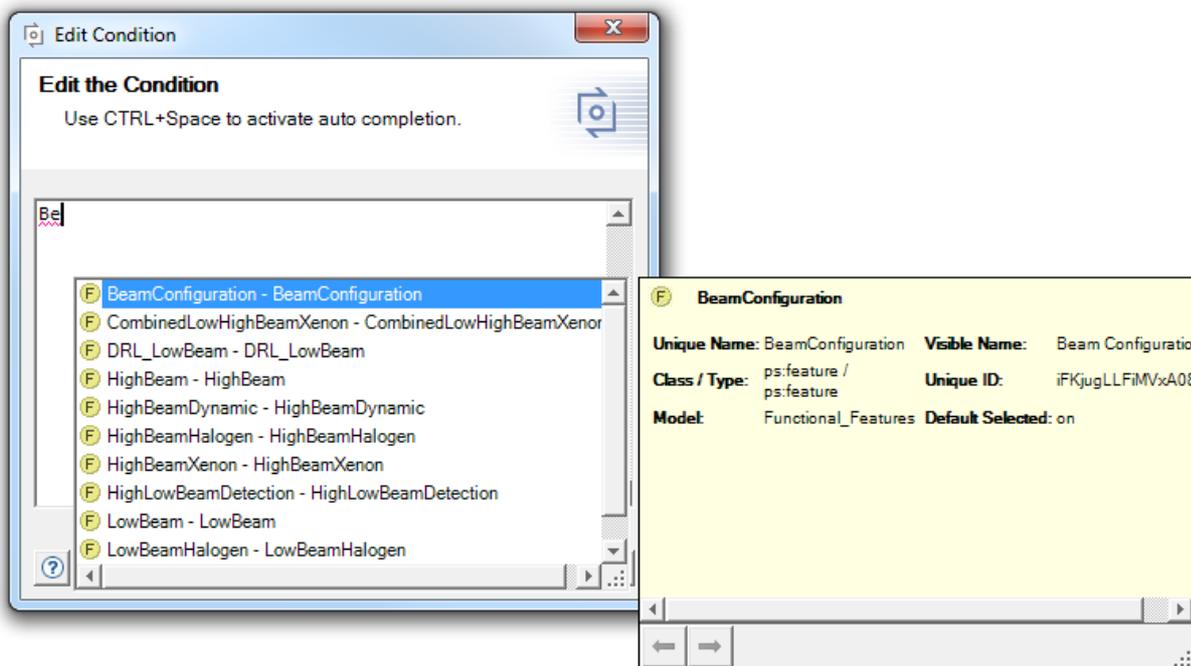
	A	B	C	D	F	G
17	LightAssistance				2	Assistance Systems
18		CorneringLights			2.1	Cornering Light
19	+		AdaptiveForwardLighting		2.1.1	Adaptive Forward Lighting
20					2.1.1.0-1	The adaptive forward lighting system is activated only when high or low beam is operating in full light mode.
21			CorneringStaticLights		2.1.2	Static Cornering Light
22				DRL	2.1.2.0-1	The day running light on the side of the car is activated when the steering angle is above $-/+15^\circ$ , the vehicle is moving, and the vehicle speed is at least 10m/s.
23					2.1.2.0-2	The fog light on the side of the car is activated when the steering angle is above $-/+15^\circ$ , the vehicle is moving, and the vehicle speed is at least 10m/s.

When you have defined a pure::variants column or row, a condition can be added to any part of it. Just select the cell(s) you want to add the conditions to and press the condition button on the ribbon. An editor dialog opens, which provides simple autocompletion and syntax highlighting. By pressing CTRL + space, possible completions for the typed word are offered (see Figure 29, “pure::variants Condition Editor”).

When you are done editing and press OK, the entered expression is checked for errors. Errors in pvSCL expressions are reported if the expressions's syntax is not pvSCL compliant, or if an element is unknown based on the loaded pure::variants models. Unknown elements are highlighted in red.

If you want to edit the condition later on, just select the containing cell and press the condition button again.

**Figure 29. pure::variants Condition Editor**



## Calculations

Texts in Excel cells may have variable parts, while the most part of the text will remain equal in all of your variants. In this case you can add a pvSCL statement to be evaluated by pure::variants - a *Calculation*. The statements will be replaced with actual values of your variant, during transformation of your variant.

A calculation is marked per default by square brackets ('[pvSCL expression]') which enclose the pvSCL expression. The pvSCL expression can be a simple reference to a pure::variants attribute (e.g., HighBeamHalogen->ECERegu\

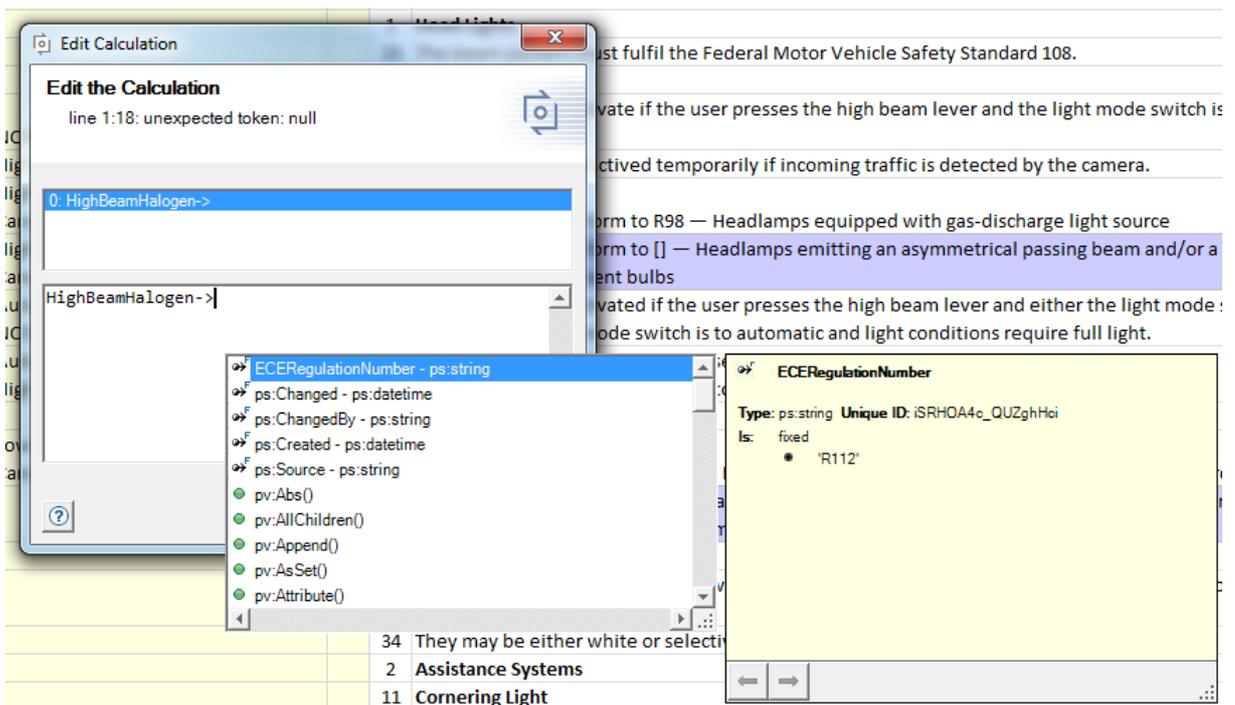
ationNumber), or a more complex rule in which the returned value is computed. For examples of pvSCL rules, either consult the pure::variants User's Guide, or press the help button in the lower left corner of the condition or calculation editor. To escape a statement the character '\$' is used per default. This will prevent pure::variants from evaluating and replacing the escaped statement.

Example: *The maximum allowed speed is [Speed->Max] km/h* in an Excel cell will be replaced with the value of attribute "Max" on Feature "Speed" in the transformed variant. The result could be: *The maximum allowed speed is 100 km/h*

Escaping the rule in the previous example, like *The maximum allowed speed is \$[Speed->Max] km/h*, forces pure::variants to ignore the rule. The result, would be *The maximum allowed speed is [Speed->Max] km/h* in that case. The rule is not changed, but the escape character is removed.

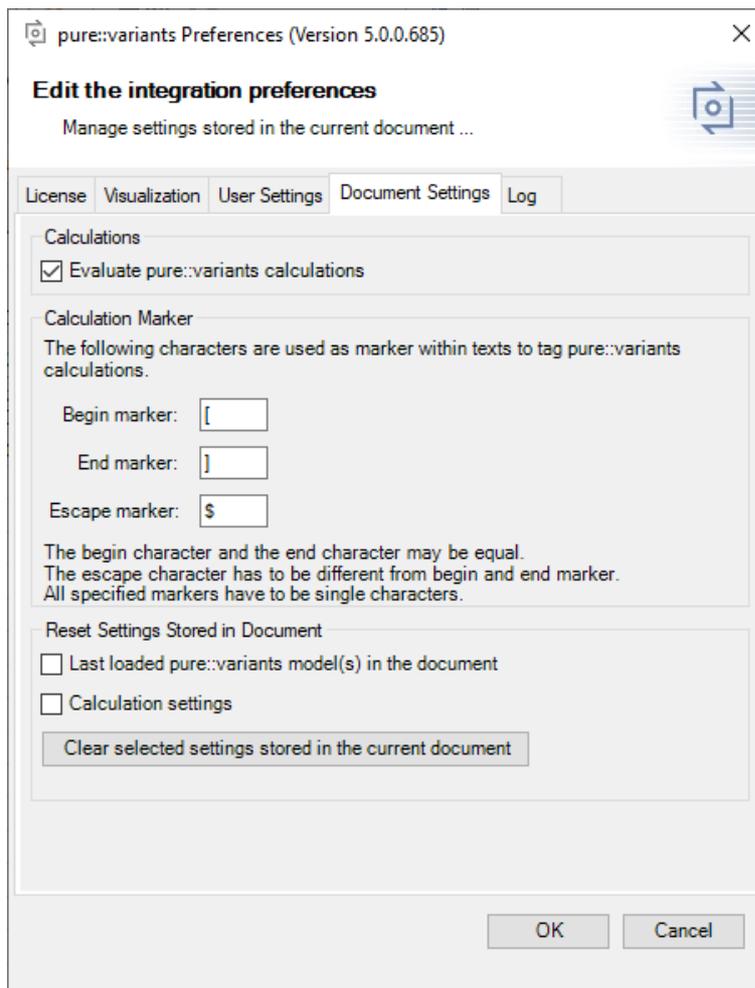
To add a calculation to a cell, enter "[]" in the cell's text, where you want to insert the calculation. Then select the cell and press the calculation button on the ribbon. The calculation editor opens (see Figure 30, "pure::variants Calculation Editor"). It works basically in the same way as the condition editor. However, it additionally provides a selection box that lists all calculations found in the selected cell. Use this selection box, to switch between calculations. Please note that editing calculations is not possible for cells that contain a formula, but only for cells that contain text.

**Figure 30. pure::variants Calculation Editor**



**Note**

If square brackets already have another meaning in your Excel project, you can customize the calculation markers in the Excel Integration preferences (see Figure 31, "pure::variants Calculation Preferences").

**Figure 31. pure::variants Calculation Preferences**

## Supported Excel Elements

Conditions can be applied to every Excel row or column. However, only elements that are affected by the deletion of a row or column will also be affected by the condition. This means that all standard cells can be annotated with conditions. Excel charts, smart art, or similar objects can only be annotated indirectly through the cells they are located on. For example, to achieve that an Excel chart is removed during transformation, you need to connect the Excel chart with the cells it is located on. To do that, open the **Format Chart** dialog of the Excel chart, go to the tab **Properties**, and select the **Object Positioning** option **Move and size with cells**. So if all rows or columns the chart is located on are removed during transformation, the chart will also not be part of the variant.

Calculations can be applied to all text cells that contain no formula.

## Visualizations

The Integration for Excel contains the same visualizations as in Word: Either preview variants for loaded *variant result models* or highlight errors in variability information. To ensure that during visualization no changes are made to your original workbook, the visualization is applied to a read-only copy of your workbook, which opens when you press a visualization button.

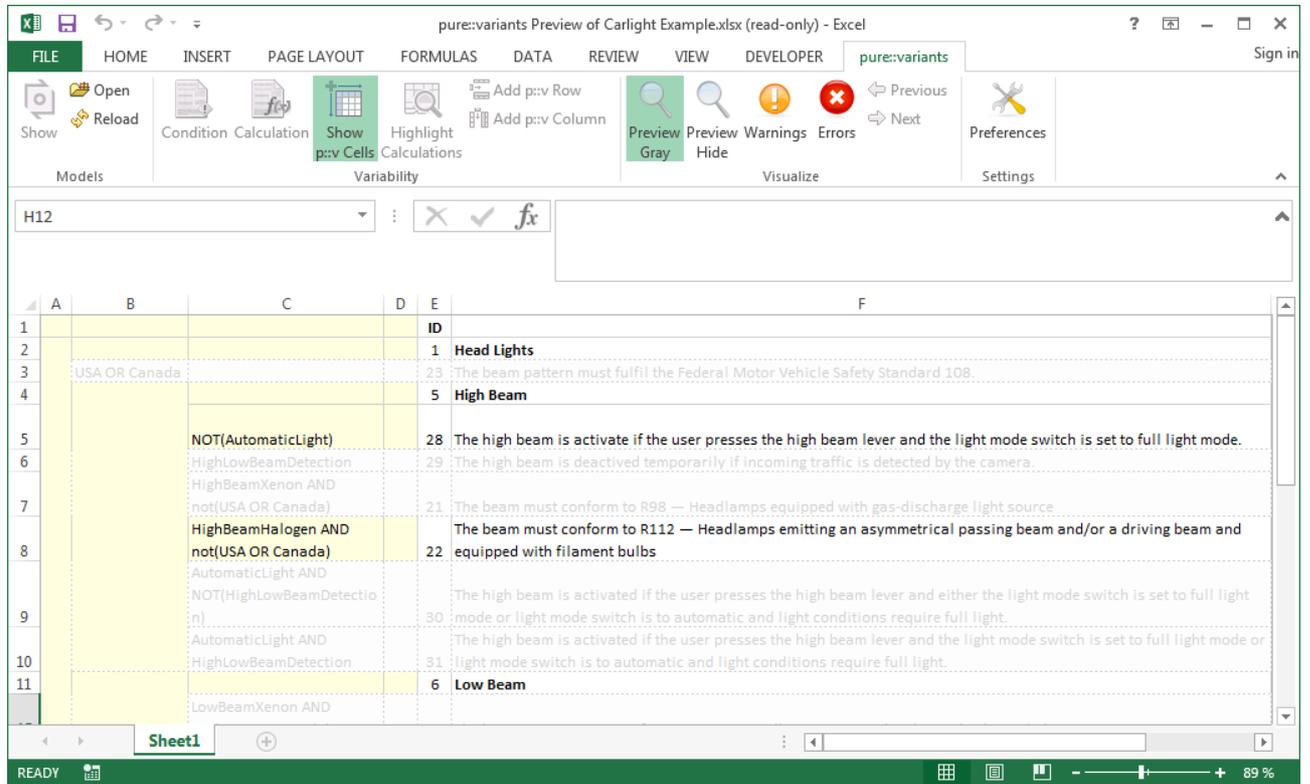
### Note

Visualizations are only supported for non-restricted and non-shared workbooks. To remove possible restrictions from your workbook use the **Protect Sheet** and **Protect Workbook** button on the Excel **Review** tab. To edit the sharing settings of your workbook, use the **Share Workbook** button on the same tab.

## Variants Visualizations

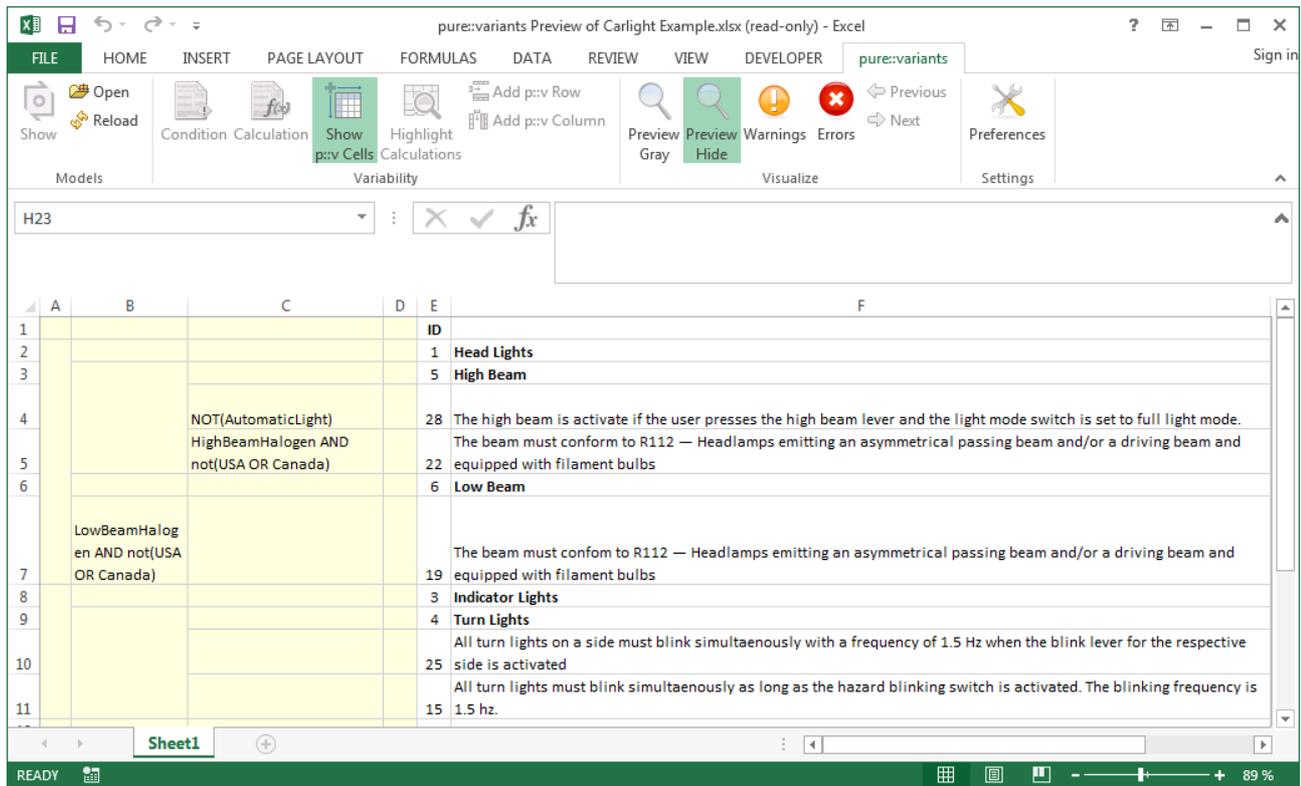
To preview variants, it is necessary to load a variant model first. Figure 32, “Preview - Gray” shows the result of a gray preview: All annotated rows or columns that would not appear in a variant are grayed out, and calculations are replaced with the value of the referenced attribute.

Figure 32. Preview - Gray



The preview - hide visualization works similar to the gray visualization. The only difference is that it deletes all rows or columns that would not be included in the final variant.

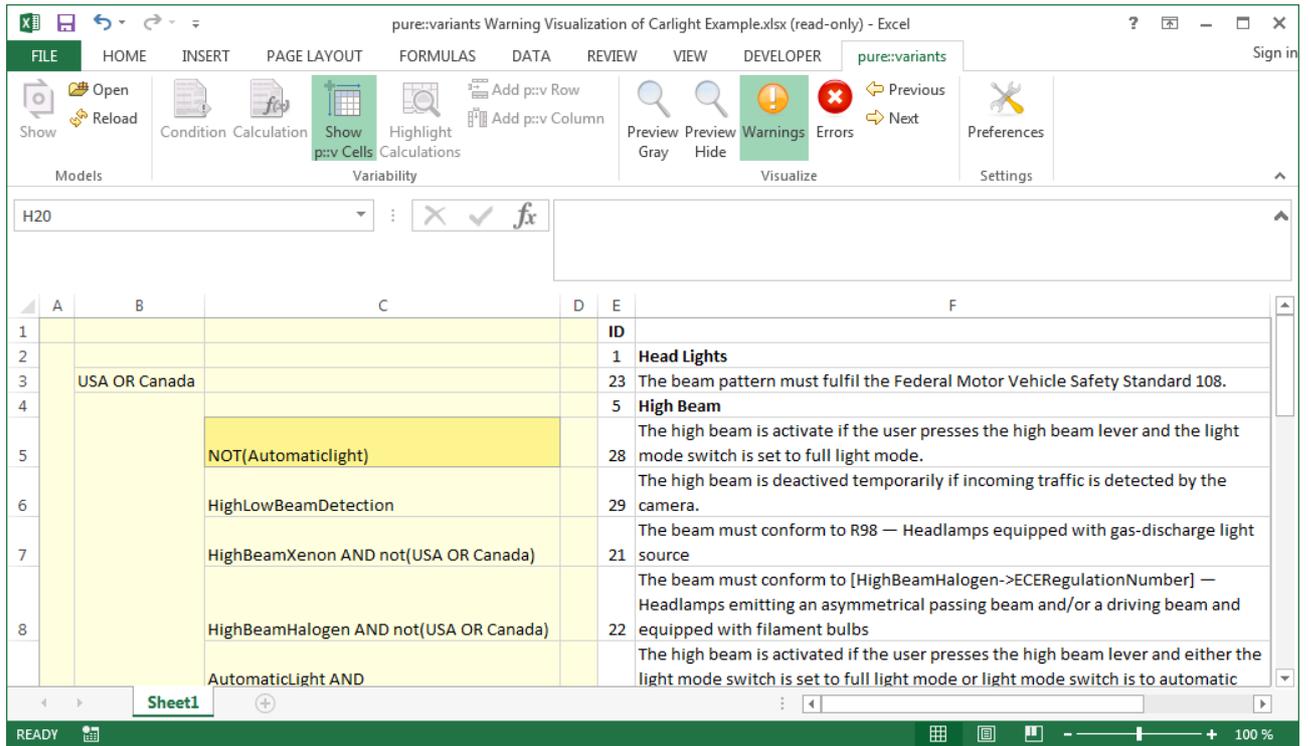
Figure 33. Preview - Hide



## Error Visualizations

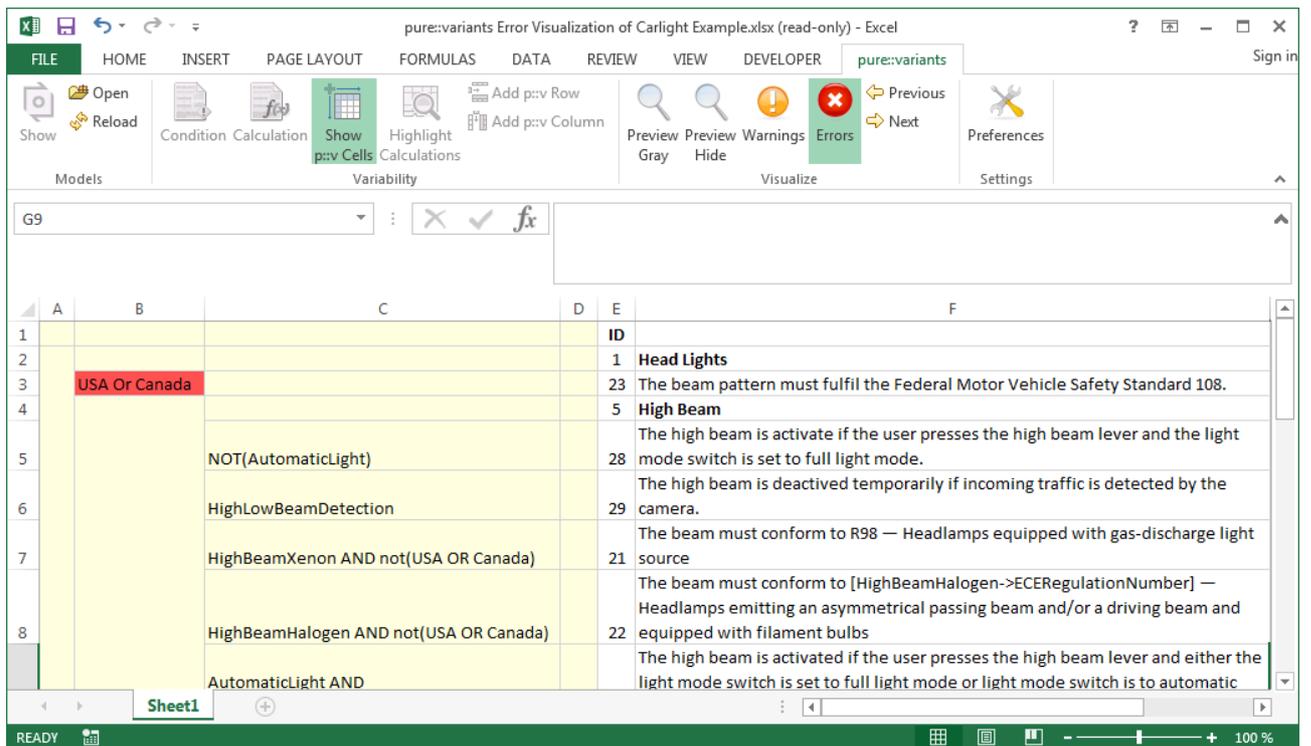
Variability information in Excel Workbooks can contain errors. To find these errors, the Integration provides two visualizations: The *Semantic Error Visualization* and the *Syntactic Error Visualization*. Using the Semantic Error Visualization, you can highlight errors that result from incorrectly typed or missing element names. To enable or disable it, press the button **Warnings**. For example in [Figure 34](#), “[Semantic Error Visualization](#)” the feature "Automaticlight" does not exist, since it should be spelled "AutomaticLight". Therefore the condition is highlighted in yellow.

**Figure 34. Semantic Error Visualization**



Using the Syntax Error Visualization, you can highlight all pvSCL expressions that are not compliant with pvSCL syntax. To enable or disable it, press the button **Errors**. For example in [Figure 35, "Syntax Error Visualization"](#) the expression "USA Or Canada" is incorrect, because the OR operation has to be spelled in upper case letters or lowercase letters. Therefore the condition is highlighted in red.

**Figure 35. Syntax Error Visualization**



To better find errors, the buttons **Previous** and **Next** provide a way to jump to the last or next faulty condition or calculation.

## Undo/Redo Behaviour Notes

After using any function of the pure::variants Integration for Microsoft Excel, the undo or redo function of Excel cannot be used any more.

## 2.6. Creating a pure::variants Project for Microsoft Word or Excel using the New Project Wizard

Creation of the corresponding pure::variants project is supported by a "New Project" wizard. It creates a standard project with most common settings, allowing to start quickly.

To start the wizard, select **File->New->Project->Variant Management->Variant Project**.

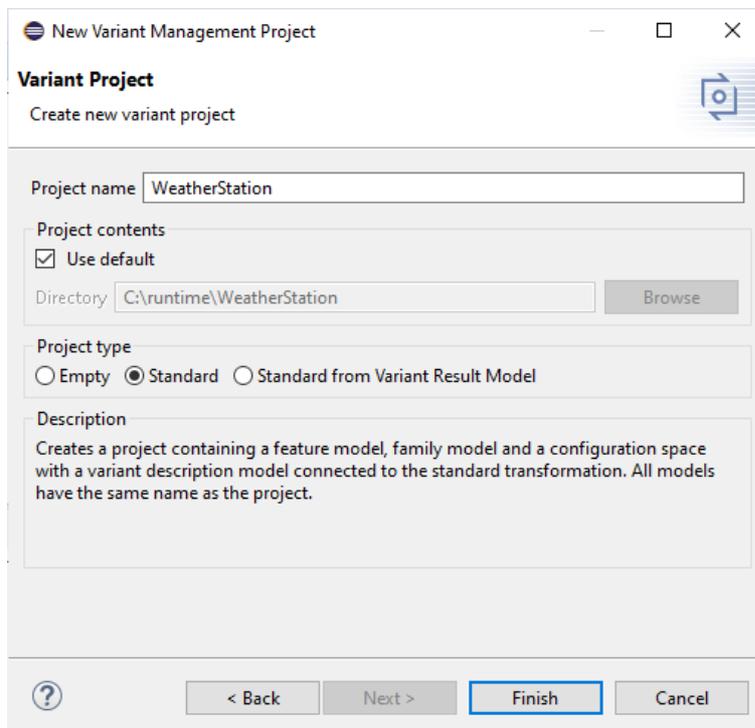
The wizard will ask for the name of the project on the first page (see [Figure 36, "Running the pure::variants Variant Project wizard"](#)). You should select the **Standard** project type, because this will create an initial pure::variants project with all needed models.

### Note

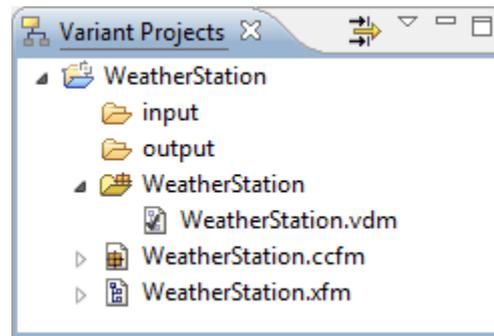
If "Empty" is selected, just an empty project is created, to which all information has to be added manually.

Only in special cases the next page is of relevance. Here other projects can be marked as referenced projects.

**Figure 36. Running the pure::variants Variant Project wizard**



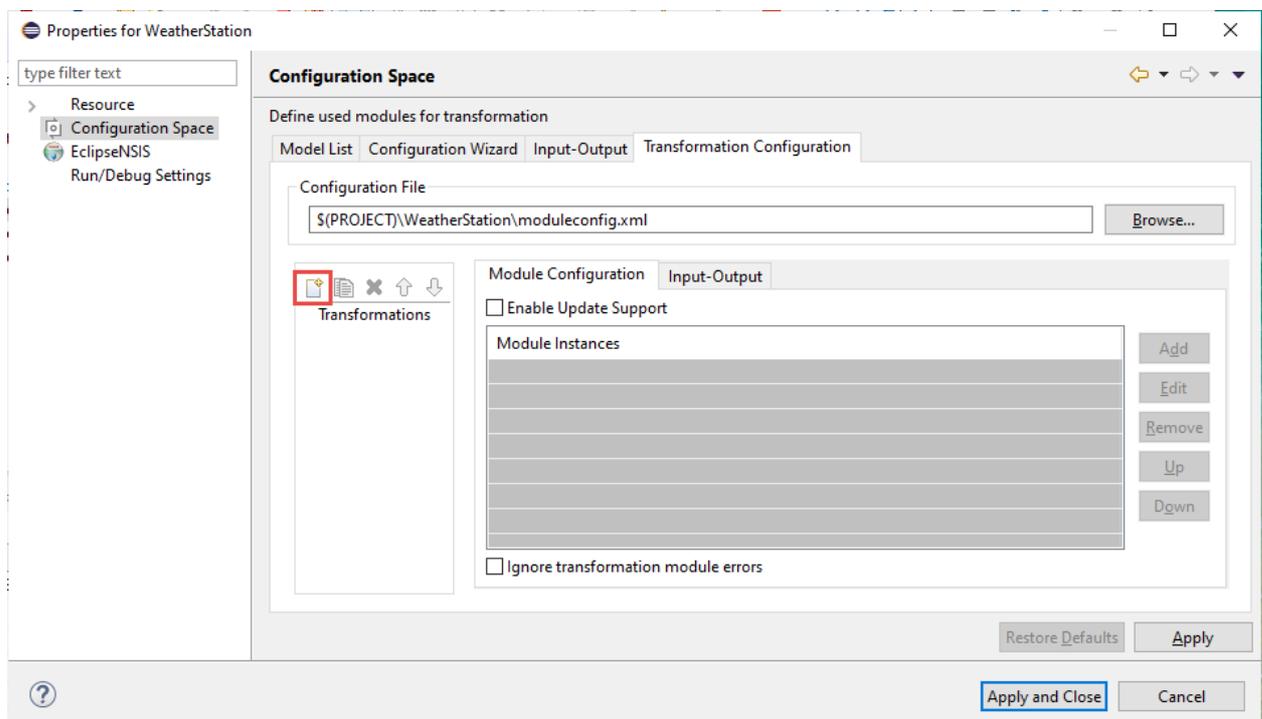
The wizard creates several models. All have the name of the project (*WeatherStation* in this case). It creates a feature model with just the root feature named like the project, a configuration space, an initial variant description model in this configuration space and two folders *input* and *output*. The *input* folder should contain the document document(s) to transform. The *output* folder will hold generated model variants. The layout of the project is shown in [Figure 37, "Layout of a Standard pure::variants project"](#).

**Figure 37. Layout of a Standard pure::variants project**

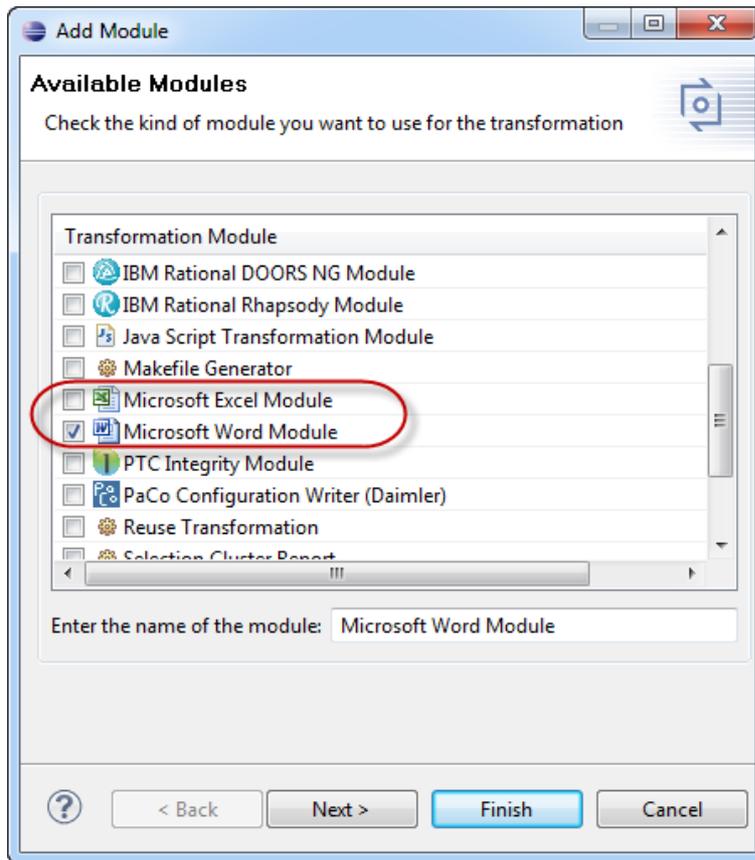
## 2.7. Adding an Microsoft Word or Excel Transformation to pure::variants Projects for Microsoft Word or Excel

For transforming document documents, a transformation module is needed. Depending on the type of the input files, either a Microsoft Word transformation module or a Microsoft Excel transformation module is required. However, pure::variants projects created with the Variant Project Wizard do not contain a transformation module per default.

To add a Microsoft Word or Excel Module, open the configuration space properties of the pure::variants project, i.e. right-click on the configuration space in the Projects View and choose **Properties** from the context menu. In the properties dialog switch to page **Configuration Space** and there to tab **Transformation Configuration**.

**Figure 38. Transformation Configuration**

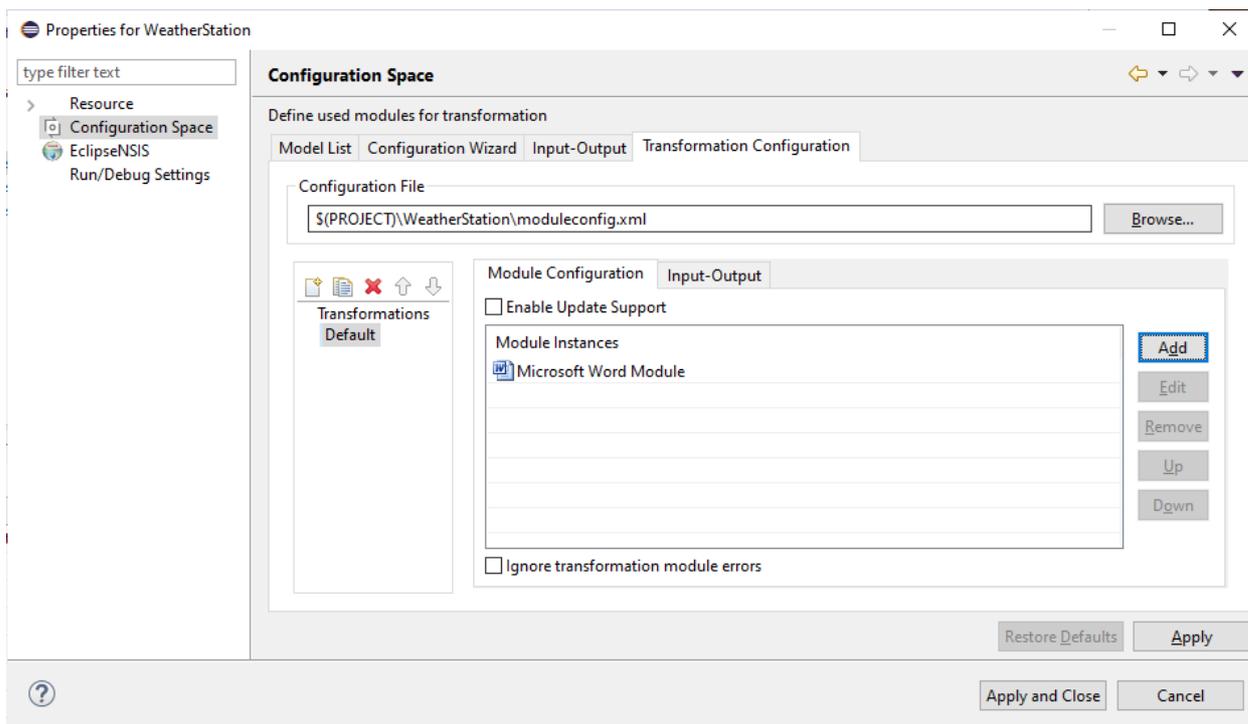
Create a new module configuration with the marked button in the image above. Click on button **Add** after creating the new module configuration. This opens the transformation module selection dialog as shown in [Figure 39](#), “Transformation Module Selection Dialog”.

**Figure 39. Transformation Module Selection Dialog**

Select Microsoft Word or Excel Module and enter a name for the new transformation. Then, click on **Finish**. The transformation configuration should then look as shown in [Figure 40, “Transformation Configuration with Microsoft Word or Excel Transformation”](#).

The transformation configuration should then look as shown in [Figure 40, “Transformation Configuration with Microsoft Word or Excel Transformation”](#).

**Figure 40. Transformation Configuration with Microsoft Word or Excel Transformation**



The new transformation is inserted at the end of the transformation module list. It is strongly recommended to leave this module on the last position in the module list.

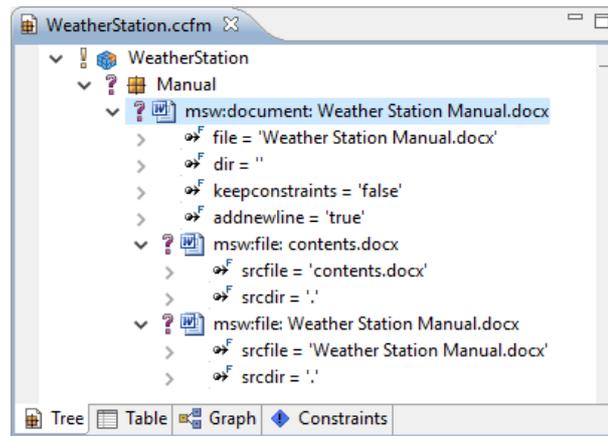
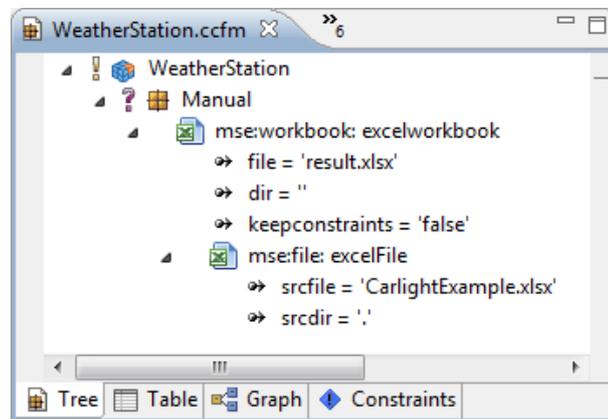
## Note

If you select the **Enable Update Support** checkbox above the list of modules, your variant output will be prepared for merging custom changes made in a variant with a newly transformed version of that variant. For details, see [Section 2.9, “Using the Microsoft Office Document Variants”](#) and section *Variant Update* in the pure::variants User's Guide.

## 2.8. Adding Office Documents to pure::variants Family Models

Now the input and output document(s) for the transformation have to be specified. This can be done using family models. The pure::variants Connector for Microsoft Office recognizes the family model part `msw:document` for Word and `mse:workbook` for Excel as trigger for a transformation. With these elements the output documents are modeled.

Below the output documents the input documents have to be modeled, as child elements of the output document. Each output document needs to have at least one input document. An `msw:file` part is used to model a Microsoft Word document as input document. The `mse:file` represents a Microsoft Excel input workbook. Input documents need to exist in file system and must not be restricted or shared in Word or Excel. Restriction or sharing settings can be changed for each input document on the **Review** tab of the Microsoft Office application.

**Figure 41. Family Model containing Word document information****Figure 42. Family Model containing Excel workbook information**

For Word transformations, the family model part `msw:document` is added to a family model by right-clicking on a component element and then choosing **New->Microsoft Word Document** from the context menu. This opens the wizard for a new Microsoft Word document part as shown in Figure 43, “New `msw:document` wizard”. Enter the name and path to the document that should be the output of the transformation, or navigate to an existing document by clicking on button ... to the right of field **file**. The option **keepconstraints** specifies whether all pure::variants comments that constrain commented text fragments should be deleted. The option **addnewline** is specific for Word transformations. It sets whether a new line should be added between documents if multiple documents are merged.

Additionally, an input document can be specified here, using the **srcfile** and **srcdir** attributes. The input document is specified in the same way as the output document. In this wizard only one input file can be specified.

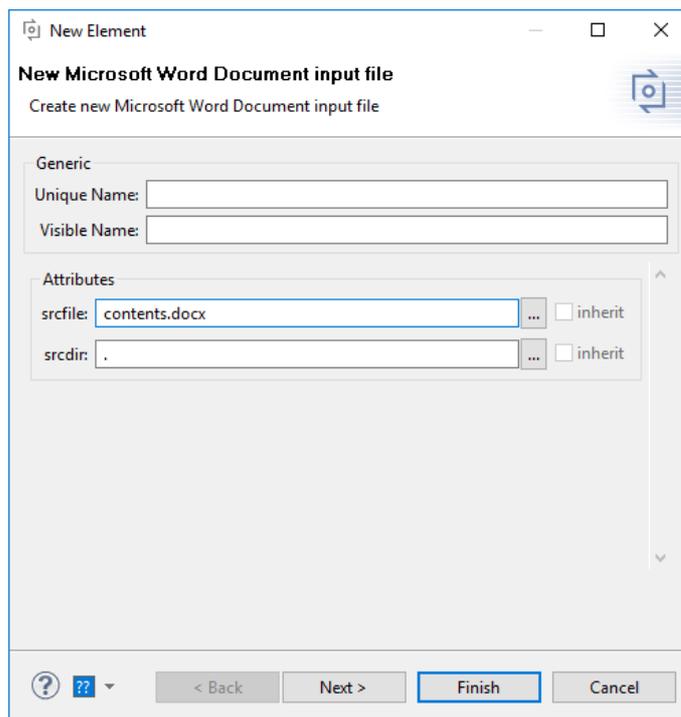
**Figure 43. New msw:document wizard**

To add more input documents for the output document the Microsoft Word File wizard can be used. This is done by right-clicking on a `msw:document` element and then choosing **New->Microsoft Word File** from the context menu. In the wizard, enter the name, the **srcfile** and **srcdir** of the input document. More input documents can be added with the same procedure. During transformation the input documents are transformed and merged into the output document specified in the `msw:document` part.

For Excel transformations, the procedure is basically the same. Instead of adding a **Microsoft Word Document** and a **Microsoft Word File**, add a **Microsoft Excel Workbook** and **Microsoft Excel Files**. Similar to Word transformations, multiple Excel input files can be added. During transformation all worksheets of the specified input files are merged into one workbook. Please note that hyperlinks between input workbooks are corrected during transformation, so that they point to a place in the output workbook.

## Note

When merging multiple input documents into one Word or Excel output, the styles and macros of the output document are always taken from the first input document.

**Figure 44. New msw:file wizard**

## 2.9. Using the Microsoft Office Document Variants

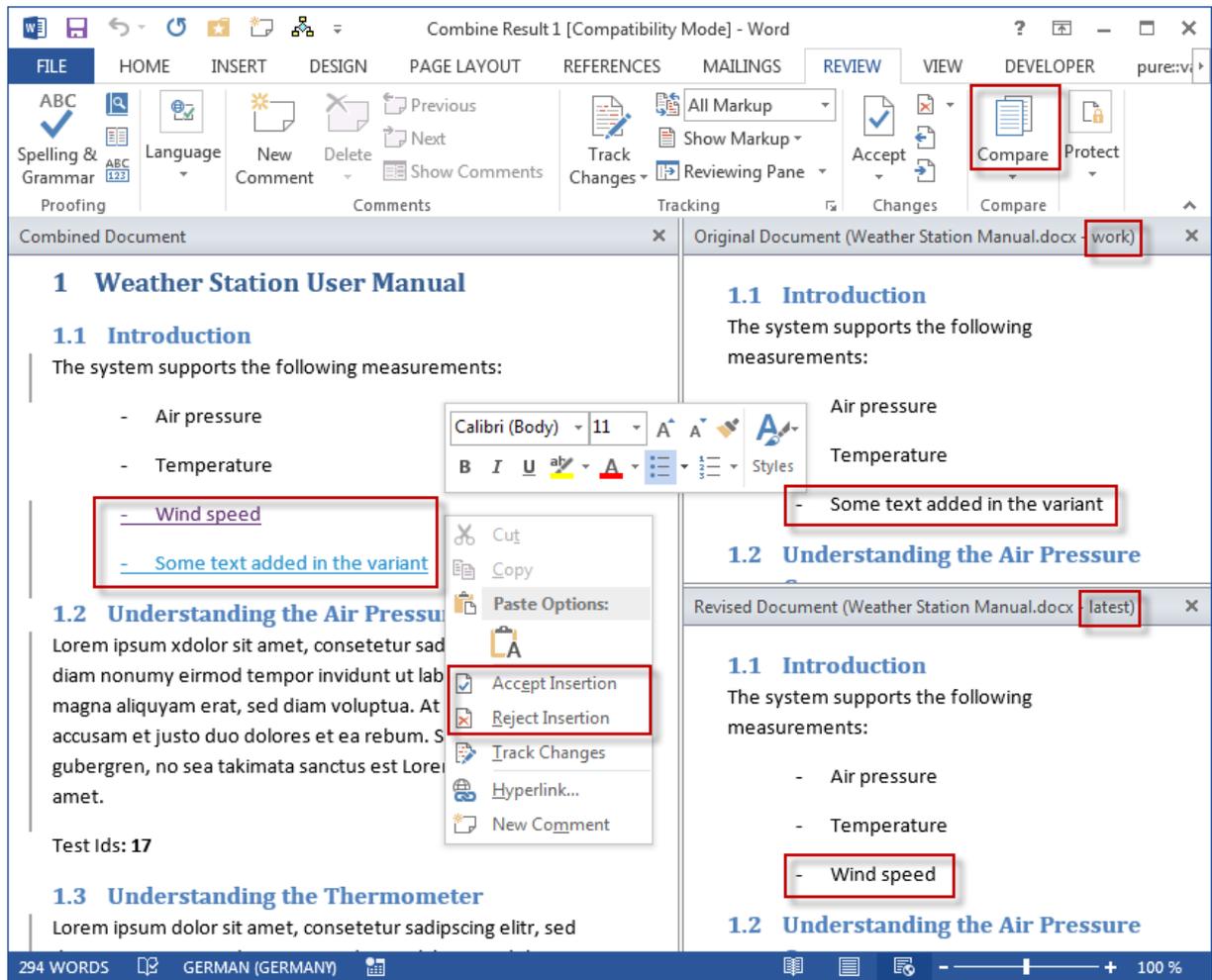
In the output folder (for standard projects the name is `output`) a sub folder with the name of the VDM is created. Inside this folder the output document is stored. A double-click will open the generated document in Microsoft Word or Excel.

The generated document can be used like the original master document. Since this document is technically a copy of the original document, all properties etc. are the same as in the master document, permitting easy compare options of generated variants with each other or with the master document.

If you selected the **Enable Update Support** checkbox in the transformation configuration of your configuration space (see [Figure 40, “Transformation Configuration with Microsoft Word or Excel Transformation”](#)), multiple sub folders will be created below the folder with the name of the VDM. You can use the documents in these folders to merge custom changes made in a variant with a newly transformed version of that variant. For details, see section *Variant Update* in the pure::variants User's Guide.

Since per default a three-way merge of Word or Excel documents is not supported in Eclipse, you may need to execute some merge steps manually or by using an external merge tool. For merging Word documents, you can utilize the **Combine...** function in Word. You can find it on the **Review** tab of the Word ribbon, in the dropdown area of the **Compare** button (see [Figure 45, “Merging two Word Documents”](#)). Please make sure that the combined document is always the document in the *work* folder. Use the context-menu of changed text fragments to accept or reject the change.

**Figure 45. Merging two Word Documents**



## Note

Currently, the Word transformation module does not update bibliography tables. Therefore, it is necessary to update them manually for each variant. All other reference tables, such as the table of contents or the table of figures, are updated correctly.

Furthermore, all filters on Excel worksheets are disabled during transformation. In case a filter is needed in the Excel variant workbook, you need to recreate it manually.

## 3. Known Issues

### 3.1. Forcefully Enable Buttons in pure::variants Ribbon Tab

We disable the pure::variants Integration at startup when Excel is not visible, to deal with the problem with event handling of Excel COM Addins and embedded Excel instances. This may cause issues when Word/Excel documents are opened through a different framework. In this case, if the buttons in pure::variants Ribbon Tab are disabled even after the document is opened, an environment variable "FORCE\_ENABLE\_PV\_OFFICE\_BUTTONS" can be set to true to override this behavior.

---