
Using variables

To favor template reusability, some of the template elements support the use of variable. User who produce a document with a template that has variable defined will need to specify the value of the variable in runtime to generate the outcome he/she expect.

How does it work?

Let's take a look at the following example:

A template *List of Operations* has been written for element type 'Class' to output a list of operation names from a given class. Let's say we want to output only operations with a specific visibility to be decided by the person who produce the final document. This is the content of the template:

```
List of operations:
```

Pay attention to @value in . Instead of having the value of visibility hardcoded in the template, a variable `#{oper-visibility}` is used.

In [Doc. Composer](#), the person who want to output a document with the template will use the template as usual by dragging it onto the document.

After that, he/she has to specify the value of the variable, which is, in this case the visibility of operation expected. To do this, right click on the content block on the document and select **Configure Local Variables...** from the popup menu.

In the **Configure Local Variables** window, enter the value of the variable. In this case, *public* is entered. This means that the variable `#{oper-visibility}` will be replaced by the text 'public' when producing content for this block.

When finished, click **OK** to confirm.

Besides entering the value, there is an option **Follow Global** in the configuration window. The value of variable can be set locally and globally. Values set locally will be effective within a specific block of content, while values set globally will be effective to the entire document. To make a variable follows global setting, simply check **Follow Global**.

To enter the value of variables globally, click on the **Configure Global Variables...** button in the toolbar of Doc. Composer and enter the value in the **Configure Global Variables** window.

Why variable?

To use variable instead of hard-coding a value gives you the following benefits.

Unifying the outcome

Instead of writing and maintaining a set of similar templates, you just need to write a single template only. This ensures the consistency of similar content since they all come from a single source.

Let the 'user' of template decide what to output

The use of variable allows the person who produce the document decide what to output, instead of forcing the editor of template to make the decision when writing the template, which is sometimes impractical.

Elements that supports the use of variable

The following table lists the elements and the attributes that support the use of variable.

Element	Attribute
TableBlock	tableStyle
Text	style, href
Property	style
ForEachDiagram	diagramType
ForEachSubDiagram	diagramType
ForEachOwnerDiagram	diagramType
HasDiagramChecker	diagramType
HasSubDiagramChecker	diagramType
HasOwnerDiagramsChecker	diagramType
HasChildElementChecker	stereotypes
HasRelationshipChecker	modelTypes
HasParentModelChecker	modelType
HasValueChecker	stereotypes, modelType, name
IterationBlock	stereotypes, name, modelTypes
ValueChecker	value