

Report Wizard custom language

On this page

- [#forrow directive](#)
- [#forpage directive](#)
- [#forpage directive for OpenDocument Presentation and Spreadsheet](#)
- [#forcol directive for OpenDocument Spreadsheet, Excel Workbook, Word Document, and PowerPoint Presentation](#)
- [#include and #parse directives](#)
- [#includeSection directive](#)
- [#include, #parse, and #includeSection directives: A comparison](#)

#forrow directive

The [Velocity template language](#) does not support loops inside a table structure. However, the Report Wizard engine introduces a new custom syntax that allows looping inside the table structure in order to clone the table rows.

The syntax is : `#forrow <query data> #endrow`

For example:

Name	Owner
<code>#forrow (\$uc in \$UseCase) \$uc.name</code>	<code>\$uc.owner.humanName #endrow</code>

#forrow (\$uc in \$UseCase) \$uc.name is used for printing a use case that one row will be printed for one use case. However, there are two use cases in this example, thus there are two rows printed out. The output will be as follows:

Name	Owner
UC1	Package Requirement
UC2	Package Requirement



Tip

- Statements preceding the **#forrow** directive, but in the same row or line will execute after the **#forrow** directive because **#forrow** is a first priority directive, while statements in the preceding rows will execute in normal sequence.

#forpage directive like importing a library, or creating an array variable, have to be declared in the rows preceding the **#forrow** directive.

The **#forpage** directive is used to provide a loop over the codes within a page. This directive provides implementation like **#forrow**, but it creates a loop over a page instead of a row. For example:

```
#forpage($uc in $UseCase)
$uc.name
#endpage
```

The report will contain a UseCase name for each document page.



Tip

- Statements preceding the **#forpage** directive, but in the same page will execute after the **#forpage** directive because **#forpage** is a first priority directive, while statements in the preceding pages will execute in normal sequence.

#forpage directive for OpenDocument Presentation and Spreadsheet like importing a library, or creating an array variable, have to be declared in the pages preceding the **#forpage** directive.

When this directive appears in the [OpenDocument Presentation template](#), it will create a loop over all directives that are present on the current page. All directives on this page will be included inside **#forpage** as shown in the figures below.

An example of a **#forpage** directive.

```
#forpage ($uc in $UseCase)
$uc.name
#endpage
```

In [OpenDocument Spreadsheet template](#), you can also use the directives as a template. The figure below displays an example of the **#forpage** directive in an OpenDocument Spreadsheet (ODS) file. When this directive is used in an ODS template, it will create a separate spreadsheet representing each name of the use cases in the model.

	A	B	C	D
1	#forpage(\$uc in \$UseCase)			
2	\$bookmark.create(\$uc.ID, \$uc.name)			
3	#endpage			
4				
5				
6				
7				

The sample of #forpage directive to create a template in an OpenDocument Spreadsheet file.

The figure below shows the output after including the directive code shown in the figure above. **Another Withdraw** is the name of one of the use cases in the model.

	A	B	C	D
1				
2	Another Withdraw			
3				
4				
5				
6				
7				

The output after including the #forpage directive in an OpenDocument Spreadsheet file.

#forcol directive for OpenDocument Spreadsheet, Excel Workbook, Word Document, and PowerPoint Presentation

This directive is designed only for the spreadsheet templates, which are OpenDocument Spreadsheet (ODS), Excel Workbook (XLSX), Word Document (DOCX), and PowerPoint Presentation (PPTX). This directive provides looping over the columns. In the figure below, the engine will generate a report with different columns for each Use Case name.

	A	B	C
1	#forcol(\$uc in \$UseCase)	\$uc.name	#endcol
2			

The sample of #forcol directive to create a template in an OpenDocument Spreadsheet file.

The output from this sample will be as shown in the figure below.

	A	B	C	D
1	Use Case A	Use Case B	Use Case C	Use Case D
2				

The output after including the #forcol directive in an OpenDocument Spreadsheet file.

You can combine both **#forrow** and **#forcol** directives and produce a more complex report output as shown below.

	A	B	C
1	#forrow(\$p in \$Package)	#forcol(\$e in \$p.ownedElement)\$e.name#endcol	#endrow

The sample of #forrow and #forcol directives to create a template in an OpenDocument Spreadsheet file.

The figure below shows an output generated from the [Magic Library.mdzip](#) sample project.

	A	B	C	D	E	F
1	Domain Analysis	Diagram Loan Collaboration	Diagram Loan	Diagram Obj	Dependency	Dependency
2	conceptual perspective					
3	Implementation	Node Remote Web Client	Node Remote	Artifact Intern	Artifact Magi	Artifact Magi
4	High Level Domain Analysis	Diagram conceptualPerspective1	Diagram con	Class Penalt	Association	Association
5	collaboration	Collaboration Loan Library Item				
6	Analysis and Design	Diagram Analysis and Design	Package Hig	Package Dor	Package Dor	Package Tech
7	object diagram	Instance Specification	Instance Spe	Instance Spe	Instance Spe	Instance Spe
8	Technical Design	Diagram domain user	Diagram dom	Diagram dom	Diagram Ma	Package Magi
9	MagicLibrary	Instance Specification	Class Library	Class Books	Class Video	Class Audio
10	Domain Design	Diagram Package Dependencies	Diagram Imp	Dependency	Dependency	Package Imple
11	composite structure	Collaboration Loan	Class Partici	Collaboration	ReadingItem	Loan
12						

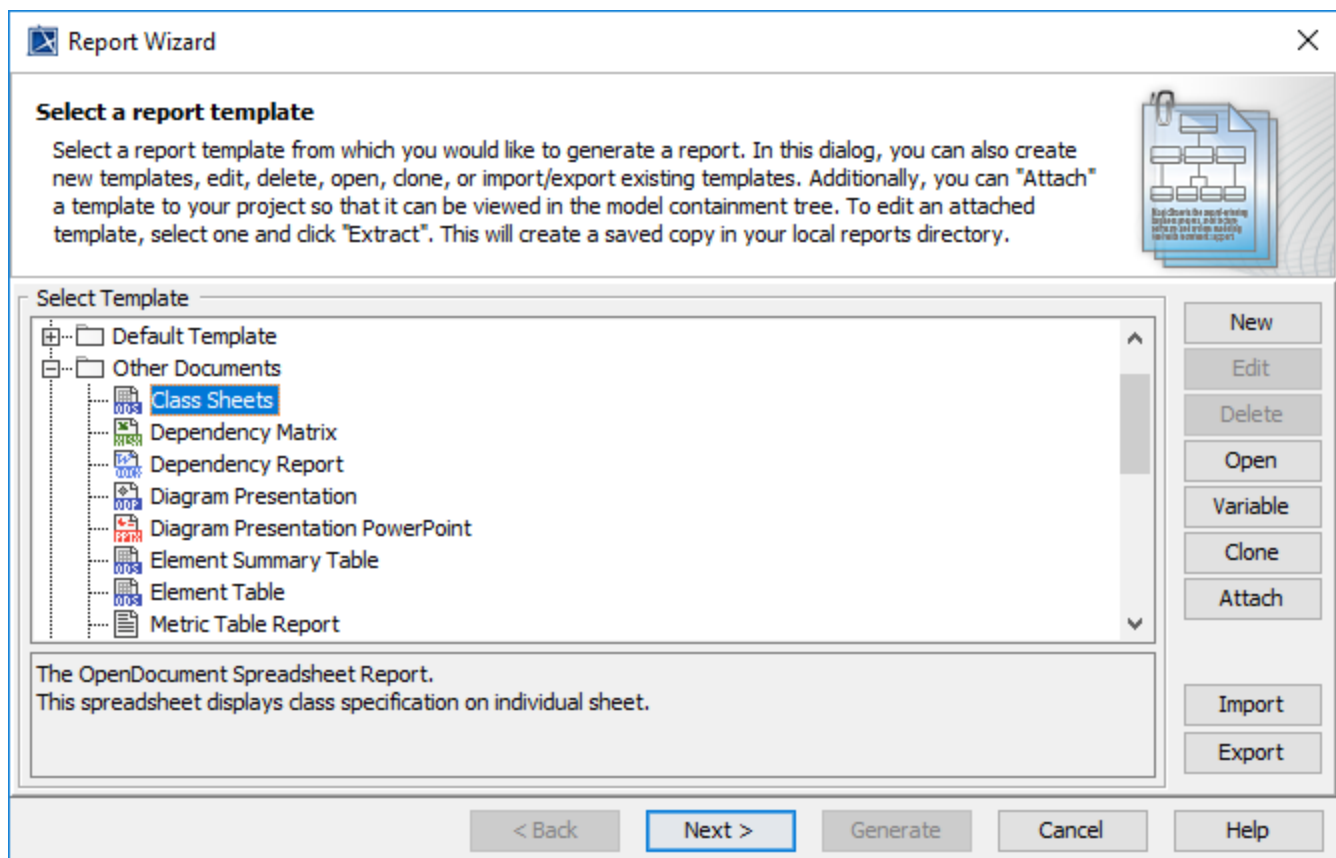
The output after including the `#forrow` and `#forcol` directives in an OpenDocument Spreadsheet file.

Since the `#forrow` syntax is similar to the `#foreach` syntax, the `#foreach` syntax can then be used in `#forrow`.

You can find more examples of using the `#forrow` and `#forcol` directives in the **Other Documents** templates in the [Report Wizard dialog](#).

To open an **Other Documents** template in the [Report Wizard dialog](#)

1. On the main menu, click **Tools > Report Wizard**. The **Report Wizard** dialog opens.
2. In the **Select Template** tree, expand **Other Documents**.
3. Select any template and click **Open**.



#include and #parse directives

The original [Velocity Engine](#) provides two include directives: `#include` and `#parse`.

`#include` allows you to import another template. The contents of the file will not be rendered through the template engine.

To include a filename using `#include($fileName)`, for example, type either:

```
#set($fileName = $attachedFile.file)
#include($fileName)
```

or

```
#include("document.docx")
```



Information

- **\$fileName** can be either an absolute or relative path to a file in the file system.
- **\$fileName** can be only static or dynamic String.

• If the file is not found in the file system, **\$fileName** would be searched as an Attached File Element in a MagicDraw project in the File property. **#parse** allows you to import another template. The contents of the file will be rendered through the template engine. However, the file being included will be inserted with all contents.

#includeSection directive

Report Wizard introduces a statement, which allows a template to include any section of a document from another template. This statement requires the template to define the beginning and the end of the section.

The logical concept of the **#includeSection** and **#parse** directives is similar. Both directives allow a template to include another template and render it through the template engine. However, **#includeSection** can be used to specify only the section that you would like to include.

To declare a section, for example, type:

```
#sectionBegin ('sectionA')
...
#sectionEnd
```

To include a section, for example, type:

```
#includeSection('document.docx', 'sectionA')
```



Information

- **fileName** and **sectionName** must be static String only, e.g., **#includeSection('document.docx', 'sectionA')**.

- If the file is not found in the file system, **fileName** would be searched as an Attached File Element in a MagicDraw project in the File property.

#include, #parse, and #includeSection directives: A comparison

The **#include** and **#parse** directives are built-in directives provided by Velocity. The **#includeSection** directive is a custom directive implemented by MagicDraw. The table below shows the differences among these three directives.

	#include	#parse	#includeSection
Execution Time	Executed at runtime.	Executed as a separate template at runtime.	Executed at translation time.
Variable Scope	Variables declared in the parent template are not accessible in the included page.	Variables declared in the parent template can be accessed in the included page.	Variables declared in the parent template can be accessed in the included page.
Rendering	The Include template is not rendered through the template engine.	The Include template is rendered through the template engine as a separate process.	The Include template is rendered through the template engine as a single process.
Include only required section	No	No	Yes
Size of parent template	The size of the parent template remains unchanged.	The size of the parent template remains unchanged.	The size of the parent template is increased by the included section.

Processing overhead	The #include directive increases the processing overhead with the necessity of an additional call to the template engine.	The #parse directive increases the processing overhead with the necessity of an additional call to the template engine.	The #includeSection directive does not increase the processing overhead.
Support RTF template	No	No	Yes
Support ODF template	No	No	Yes
Support DOCX template	No	No	Yes
Support XLSX template	No	No	No
Support PPTX template	No	No	No

Related pages

- [Creating a template for an OpenDocument Presentation \(ODP\) or Microsoft PowerPoint presentation \(PPTX\)](#)
- [Creating a template for an OpenDocument spreadsheet or Microsoft Excel spreadsheet](#)
- [Velocity template language](#)
- [Velocity engine: #include and #parse directives](#)