

# The `templatetools` package

Matthias Pospiech  
[matthias@pospiech.eu](mailto:matthias@pospiech.eu)

v0.1 from 2014/06/27

## Abstract

Collection of tools, which are helpful for the creation of a L<sup>A</sup>T<sub>E</sub>X template if conditional paths for code execution are required.

## 1 Usage

### 1.1 Commands

The following commands check if a command sequence is defined or not.

`\IfDefined`  $\{\langle command \rangle\}\{\langle code\ defined \rangle\}$

Executes the code if the command is defined.

`\IfUndefined`  $\{\langle command \rangle\}\{\langle code\ undefined \rangle\}$

Executes the code if the command is not defined.

`\IfElseDefined`  $\{\langle command \rangle\}\{\langle defined \rangle\}\{\langle undefined \rangle\}$

Executes either the code in the *defined* bracket if the command is defined or in the *undefined* bracket if the code is undefined.

`\IfElseUndefined`  $\{\langle command \rangle\}\{\langle undefined \rangle\}\{\langle defined \rangle\}$

As `\IfElseDefined`, but with switched brackets for defined and undefined.

### Example

The `\usepackage` code is only executed if the required `\upmu` command is defined.

```
% Requires: Command \upmu
\IfDefined{upmu}{\usepackage[upmu]{gensymb}}
```

`\IfMultDefined`  $\{\langle list\ of\ commands\rangle\}\{\langle defined\rangle\}\{\langle undefined\rangle\}$

Checks a comma separated list of commands before it executes the defined code path if all commands were defined or the undefined code path else.

### Example

```
% Requires: Command longtable and rowcolors
\IfMultDefined{longtable,rowcolors}
{... longtable with rowcolors ...}
{Error: Neither longtable nor rowcolors are defined}
```

## 1.2 Draft mode

The following commands check if draft mode is active or not.

`\IfDraft`  $\{\langle draft\ mode\ active\rangle\}$

`\IfNotDraft`  $\{\langle draft\ mode\ disabled\rangle\}$

`\IfNotDraftElse`  $\{\langle draft\ mode\ disabled\rangle\}\{\langle draft\ mode\ active\rangle\}$

### Example

The bookmark is not loaded in draft mode:

```
\IfNotDraft{\usepackage[] {bookmark}}
```

## 1.3 Packages

These commands check if a package was loaded or not. This can be achieved in different ways with commands from other packages. The key point of these commands here is that they work not only in the preamble and include no @-char.

`\IfPackageLoaded`  $\{\langle package\rangle\}\{\langle is\ loaded\rangle\}$

`\IfPackageNotLoaded`  $\{\langle package\rangle\}\{\langle is\ not\ loaded\rangle\}$

`\IfPackagesLoaded`  $\{\langle list\ of\ packages\rangle\}\{\langle all\ are\ loaded\rangle\}$

`\IfPackagesNotLoaded`  $\{\langle list\ of\ packages\rangle\}\{\langle none\ is\ loaded\rangle\}$

## Example

```
% Load epstopdf only if graphicx was loaded
\IfPackageLoaded{graphicx}{%
  \usepackage{epstopdf}
}
% Do not load subcaption if subfig was loaded (incompatible)
\IfPackageNotLoaded{subfig}{
  \usepackage{subcaption}[2011/08/17]
}
```

## 1.4 Package Loading order

In L<sup>A</sup>T<sub>E</sub>X documents it is quite often essential to load packages in the right order to ensure that everything works. However this makes it impossible to group similar packages together.

The following commands allow to execute code after or before a specified package and thus also allows to load packages in a specified order using `\usepackage` commands.

If the reference package was not loaded in the preamble the code will nevertheless be executed before `\begin{document}`

`\ExecuteAfterPackage` `{\langle after this package \rangle}{\langle execute this code \rangle}`

`\ExecuteBeforePackage` `{\langle before this package \rangle}{\langle execute this code \rangle}`

## Example

`cleveref` package must be loaded after package `hyperref`.

```
% loading: must be loaded after hyperref and after varioref
\ExecuteAfterPackage{hyperref}{
% caption and cleveref incompatible in Versions before 2011/12/24
  \usepackage{cleveref}[2011/12/24]
}
```

## 1.5 Tikz Library

Checks if a tikz library was loaded.

`\IfTikzLibraryLoaded` `{\langle library \rangle}{\langle if loaded \rangle}`

## Example

Executes the code only if the tikz library was loaded.

```
\IfTikzLibraryLoaded{lindenmayersystems}{%
% code origin: pgf/tikz manual
\begin{tikzpicture}
\pgfdeclarelindenmayersystem{Koch curve}{
  \rule{F -> F-F++F-F}
}
\shadedraw [top color=white, bottom color=blue!50, draw=blue!50!black]
  [l-system={Koch curve, step=2pt, angle=60, axiom=F++F++F, order
=3}]
  lindenmayer system -- cycle;
\end{tikzpicture}
}%
```

## 1.6 Column types

L<sup>A</sup>T<sub>E</sub>X provides no tool to check for the existence of a column type. This is provided by the following commands:

```
\IfColumnTypeDefined {<column type character>}{<is defined>}{<is undefined>}
```

```
\IfColumnTypesDefined {<column type character list>}{<is defined>}{<is undefined>}
```

## Example

Executes the code only if the X column type is defined and the tabularx package was loaded by checking that `\tabularx` is defined.

```
\IfColumnTypeDefined{X}{%
\IfDefined{tabularx}{%
%
\begin{tabularx}{0.9\textwidth}{lXX}
\hline
l & l & X & X \\ \hline
%
left column & left column &
text which is considerably longer than the width of the column &
text which is considerably longer than the width of the column \\
\hline
\end{tabularx}
%
}%
```

## 1.7 Color definitions

Color definitions are saved in L<sup>A</sup>T<sub>E</sub>X as names. The following commands provide a convenient way to check the existence of these color definitions.

`\IfColorDefined`  $\{\langle color\ name\rangle\}\{\langle is\ defined\rangle\}\{\langle is\ undefined\rangle\}$

`\IfColorsDefined`  $\{\langle list\ of\ color\ name\rangle\}\{\langle is\ defined\rangle\}\{\langle is\ undefined\rangle\}$

## 1.8 Math font version

`\IfMathVersionDefined`  $\{\langle font\ version\rangle\}\{\langle is\ defined\rangle\}\{\langle is\ undefined\rangle\}$

## 1.9 Glossaries styles

`\IfGlossariesStyleDefined`  $\{\langle style\ name\rangle\}\{\langle is\ defined\rangle\}$

## 1.10 Template Definitions

The following commands in principle define only macros, but in contrast to normal methods these are saved using two keys named *group* and *property*. With a matching command for the execution this allows to generate macros in an object like naming structure, which can be used to toggle settings.

`\SetTemplateDefinition`  $\{\langle Group\rangle\}\{\langle Property\rangle\}\{\langle Code\rangle\}$

Defines a collection of commands (a macro) with a *group* and *property*.

`\UseDefinition`  $\{\langle Group\rangle\}\{\langle Property\rangle\}$

Execute macro save with the *group* and *property*.

### Example

The following code allows to switch the colors anywhere in the document:

```
\SetTemplateDefinition{Target}{Web}{%
  \definecolor{pdfurlcolor}{rgb}{0,0,0.6}
}%
\SetTemplateDefinition{Target}{Print}{%
  \definecolor{pdfurlcolor}{rgb}{0,0,0}
}%
% Apply colors for web
\UseDefinition{Target}{Web}
```

## 2 Implementation

```
9 \NeedsTeXFormat{LaTeX2e}[1994/12/01]
10 \ProvidesPackage{templatetools}
11 [2014/06/27 v0.1 Collection of conditional commands useful inside
    templates]
12 %

13 %%% --- Necessary Packages
14 %%% -----
15 \RequirePackage{ifpdf}
16 \RequirePackage{etoolbox}
17 \RequirePackage{ltxcmds}
18 \RequirePackage{array} % for column types
19 \RequirePackage{ifdraft} % check draft
20 \RequirePackage{scrfile}
21 %
```

### 2.1 Command sequences

**\IfDefined** Wrapper to \ifcsdef with only true path.

```
22 \newcommand{\IfDefined}[2]{\ifcsdef{#1}{#2}{}}%
23 %
```

**\IfUndefined** Wrapper to \ifcsdef with only false path.

```
24 \newcommand{\IfUndefined}[2]{\ifcsdef{#1}{}{#2}}%
25 %
```

**\IfElseDefined** Wrapper to \ifcsdef with true and false path.

```
26 \newcommand{\IfElseDefined}[3]{\ifcsdef{#1}{#2}{#3}}%
27 %
```

**\IfElseUndefined** Wrapper to \ifcsdef with true and false path in reverse order.

```
28 \newcommand{\IfElseUndefined}[3]{\ifcsdef{#1}{#3}{#2}}%
29 %
```

**\IfMultDefined** Checks if more than one command is defined

```
30 \newcommand{\IfMultDefined}[1]{%
31   \@tempswatru
32   \def\do##1{%
33   %% define \@tempa with trimmed index element.
34   \edef\@tempa{\zap@space##1 \@empty}}%
```

```

35 %% check if package of current index is loaded
36   \ifcsdef{\@tempa}{\@tempwafalse}%
37   }%
38 %% Process csv list with command \do (etoolbox)
39   \docsvlist{#1}%
40 %% makes sure that the conditional works with one or two (if, else)
parameters.
41   \if@tempwa\expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi%
42 }
43 %

```

Thanks to EGREG, ANDREY VIHROV, MARTIN SCHARRER on [tex.stackexchange.com](https://tex.stackexchange.com) for the help to implement a command that checks a comma separated list.

## 2.2 Draft mode

**\IfDraft** Tests if \@draft is undefined and executed false path in case draft string is defined.

```

44 \newcommand{\IfDraft}[1]{\ifx\@draft\@undefined \else #1 \fi}
45 %

```

**\IfNotDraft** Similar to \IfDraft but executes only path for draft mode undefined.

```

46 \newcommand{\IfNotDraft}[1]{\ifx\@draft\@undefined #1 \fi}
47 %

```

**\IfNotDraftElse** Similar to \IfDraft but executes true and false path.

```

48 \newcommand{\IfNotDraftElse}[2]{\ifx\@draft\@undefined #1 \else #2 \fi}
49 %

```

## 2.3 Packages

If a package is loaded can be checked in many ways, but here the `\ltx@ifpackageloaded` is used because it can be executed anywhere in the document.

**\IfPackageLoaded** Wrapper to `\ltx@ifpackageloaded` with only true path.

```

50 \newcommand{\IfPackageLoaded}[2]{\ltx@ifpackageloaded{#1}{#2}{}}
51 %

```

**\IfPackageNotLoaded** Wrapper to `\ltx@ifpackageloaded` with only false path.

```

52 \newcommand{\IfPackageNotLoaded}[2]{\ltx@ifpackageloaded{#1}{}{#2}}
53 %

```

`\IfElsePackageLoaded` Wrapper to `\ltx@ifpackageloaded`

```
54 \let\IfElsePackageLoaded\ltx@ifpackageloaded
55 %
```

`\IfPackagesLoaded` Checks a list of packages

```
56 \newcommand{\IfPackagesLoaded}[1]{%
57   \@tempswatrue
58   \def\do##1{%
59     \edef\@tempa{\zap@space##1 \empty}%
60     \ltx@ifpackageloaded{\@tempa}{\@tempswafalse}%
61   }%
62   \docsvlist{#1}%
63   \if@tempswa\expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi%
64 }
65 %
```

`\IfPackagesNotLoaded` Invers check if all packages in a list are not loaded

```
66 \newcommand{\IfPackagesNotLoaded}[1]{%
67   \@tempswatrue
68   \def\do##1{%
69     \edef\@tempa{\zap@space##1 \empty}%
70     \ltx@ifpackageloaded{\@tempa}{\@tempswafalse}{}%
71   }%
72   \docsvlist{#1}%
73   \if@tempswa\expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi%
74 }
75 %
```

`\ExecuteAfterPackage` Executes the code after the reference package has been loaded (using `\AfterAtEndOfPackage`) or finally at the end of the preamble if the reference package was not loaded until then.

```
76 \newcommand{\ExecuteAfterPackage}[2]{%
77   %% #1: after this package
78   %% #2: code to execute
79   \AfterAtEndOfPackage{#1}{
80     #2%
81   }
82   \AtEndPreamble{%
83     \IfPackageNotLoaded{#1}{%
84       #2%
85     }
86   }
87 }
88 %
```



`\ExecuteBeforePackage` Executes the code directly before the reference package is loaded (using `\BeforePackage`) or finally at the end of the preamble if the reference package was not loaded until then.

```
89 \newcommand{\ExecuteBeforePackage}[2]{%
90 %% #1: before this package
91 %% #2: code to execute
92 \BeforePackage{#1}{
93   #2%
94 }
95 \AtEndPreamble{%
96   \IfPackageNotLoaded{#1}{%
97     #2%
98   }
99 }
100 }
101 %
```

## 2.4 Tikz library

`\IfTikzLibraryLoaded` Checks if the tikz library is loaded

```
102 \def\IfTikzLibraryLoaded#1{%
103   \ifcsname tikz@library@#1@loaded\endcsname
104     \expandafter\@firstoftwo
105   \else
106     \expandafter\@secondoftwo
107   \fi
108 }
109 %
```

Thanks to EGREG and MARCO DANIEL on [tex.stackexchange.com](http://tex.stackexchange.com) for their help with this command.

## 2.5 Column types in tables

The code in this section was inspired by the discussion with EGREG on [tex.stackexchange.com](http://tex.stackexchange.com) on the detection of column definitions.

Creates a list of predefined columntypes

```
110 \expandafter\let\csname columntype@l\endcsname\@empty
111 \expandafter\let\csname columntype@c\endcsname\@empty
112 \expandafter\let\csname columntype@r\endcsname\@empty
113 \expandafter\let\csname columntype@p\endcsname\@empty
114 \expandafter\let\csname columntype@m\endcsname\@empty
115 \expandafter\let\csname columntype@b\endcsname\@empty
```

```

116 \expandafter\let\csname columntype@@\endcsname\@empty
117 \expandafter\let\csname columntype@!\endcsname\@empty
118 \expandafter\let\csname columntype@|\endcsname\@empty
119 \expandafter\let\csname columntype@<\endcsname\@empty
120 \expandafter\let\csname columntype@>\endcsname\@empty
121 \expandafter\let\csname columntype@=\endcsname\@empty
122 %

```

**\CheckIfColumntypeDefined** Creates a bool variable that saves the status of the column type.

```

123 \newcommand\CheckIfColumntypeDefined[1]{%
124 %% create the bool variable for column type
125 \providebool{tpl@coltype@#1}
126 %% check if new column type of this name was created
127 \ifcsdef{NC@find@\string#1}%
128 {\setbool{tpl@coltype@#1}{true}}%
129 %% if not check if it is a predefined column type
130 {\ifcsdef{columntype@\string#1}
131 {\setbool{tpl@coltype@#1}{true}}%
132 {\setbool{tpl@coltype@#1}{false}}%
133 }%
134 }
135 %

```

**\isColumntypeDefined** Returns the bool variable which can be interpreted by `\ifboolexpr`. This should only be used internally and fails for nonexistent bool variables.

```

136 \newcommand\isColumntypeDefined[1]{tpl@coltype@#1}
137 %

```

**\IfColumntypeDefined** Executes `\CheckIfColumntypeDefined` and uses the resulting bool variable with `\isColumntypeDefined` in a conditional sequence with `\ifboolexpr`.

```

138 \newcommand\IfColumntypeDefined[3]{%
139 %% Execute check which create bool variable
140 \CheckIfColumntypeDefined{#1}
141 %% use bool variable for if sequence
142 \ifboolexpr{ bool{\isColumntypeDefined{#1}} }{#2}{#3}%
143 }
144 %

```

**\IfColumntypesDefined** Checks a comma separated list instead of a single string.

```

145 \newcommand{\IfColumntypesDefined}[1]{%
146 \@tempswatrue
147 \def\do##1{%
148 \edef\@tempa{\zap@space##1 \@empty}%
149 %% check if column is defined.

```

```

150 %% Here with \expandafter because of the \string definition
151 %% in \CheckIfColumnTypeDefined.
152 \expandafter\IfColumnTypeDefined
153 \expandafter{\@tempa}{\@tempswafalse}
154 }%
155 \docsvlist{#1}%
156 \if@tempswa\expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi%
157 }
158 %

```

## 2.6 Color definitions

`\IfColorDefined` Tests if a color is defined

```

159 \newcommand{\IfColorDefined}[3]{%
160 \ifcsdef{\string\color @#1}
161 {#2} % color string is defined
162 {#3}} % color string is not defined
163 %

```

`\IfColorsDefined` Does the same for a list (comma separated) of color names.

```

164 \newcommand{\IfColorsDefined}[1]{%
165 \@tempswatrue
166 \def\do##1{%
167 \edef\@tempa{\zap@space##1 \@empty}%
168 \expandafter\IfColorDefined
169 \expandafter{\@tempa}{\@tempswafalse}
170 }%
171 \docsvlist{#1}%
172 \if@tempswa\expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi%
173 }
174 %

```

## 2.7 Math font version

`\IfMathVersionDefined` Checks if a mathversion font is defined.

```

175 \newcommand{\IfMathVersionDefined}[3]{\ifcsdef{mv@#1}{#2}{#3}}
176 %

```

Thanks to WERNER and ULRIKE FISCHER for pointing me to this name definition on [tex.stackexchange.com](http://tex.stackexchange.com).

## 2.8 Glossaries styles

`\IfGlossariesStyleDefined` Checks if a glossaries style is defined.

```
177 \newcommand{\IfGlossariesStyleDefined}[2]{\ifcsdef{@glsstyle@#1}{#2}{}}
178 %
```

Thanks to CGNIEDER on [tex.stackexchange.com](http://tex.stackexchange.com) for the help to implement the command.

## 2.9 Template definitions

`\SetTemplateDefinition` Defines a macro with the *group* and *property* parameter names.

```
179 \newcommand\SetTemplateDefinition[3]{% 1: group, 2: property, 3: code
180   \csdef{tpl@definition@#1@#2}{#3}
181 }
182 %
```

`\UseDefinition` Executes the macro using `\csuse` if it is defined. Otherwise a warning is thrown.

```
183 \newcommand\UseDefinition[2]{%
184   \ifcsdef{tpl@definition@#1@#2}
185     {\csuse{tpl@definition@#1@#2}}
186     {\PackageWarning{templatetools}%
187       {Definition #1->#2 is unknown\MessageBreak}{}}%
188 }%
189 %
```

## Index

Numbers written in *italics* refer to the page where the corresponding entry is described; numbers underlined refer to the definition; numbers in *roman* refer to the pages where the entry is used.

<b>C</b>	<b>I</b>	<code>\IfElseDefined</code> ... <a href="#">1, 6</a>
<code>\CheckIfColumnTypeDefined</code> .. <a href="#">5, 11</a>	<code>\IfColorDefined</code> .. <a href="#">5, 11</a>	<code>\IfElsePackageLoaded</code> \
..... <a href="#">10</a>	<code>\IfColorsDefined</code> .. <a href="#">5, 11</a>	..... <a href="#">8</a>
	<code>\IfColumnTypeDefined</code>	<code>\IfElseUndefined</code> .. <a href="#">1, 6</a>
<b>E</b>	..... <a href="#">4, 10</a>	<code>\IfGlossariesStyleDefined</code>
<code>\ExecuteAfterPackage</code>	<code>\IfColumnTypesDefined</code>	..... <a href="#">5, 12</a>
..... <a href="#">3, 8</a>	..... <a href="#">4, 10</a>	<code>\IfMathVersionDefined</code>
<code>\ExecuteBeforePackage</code>	<code>\IfDefined</code> .. <a href="#">1, 6</a>	..... <a href="#">5, 11</a>
..... <a href="#">3, 9</a>	<code>\IfDraft</code> .. <a href="#">2, 7</a>	<code>\IfMultDefined</code> ... <a href="#">2, 6</a>

