The templatetools package

Matthias Pospiech
matthias@pospiech.eu

v0.2 from 2023/03/26

Abstract

Collection of tools, which are helpful for the creation of a LaTeX 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` {⟨*command*⟩}{⟨*code defined*⟩}

Executes the code if the command is defined.

`\IfUndefined` {⟨*command*⟩}{⟨*code undefined*⟩}

Executes the code if the command is not defined.

`\IfElseDefined` {⟨*command*⟩}{⟨*defined*⟩}{⟨*undefined*⟩}

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

`\IfElseUndefined` {⟨*command*⟩}{⟨*undefined*⟩}{⟨*defined*⟩}

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` {⟨*list of commands*⟩}{⟨*defined*⟩}{⟨*undefined*⟩}

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` {⟨*draft mode active*⟩}

`\IfNotDraft` {⟨*draft mode disabled*⟩}

`\IfNotDraftElse` {⟨*draft mode disabled*⟩}{⟨*draft mode active*⟩}

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` {⟨*package*⟩}{⟨*is loaded*⟩}

`\IfPackageNotLoaded` {⟨*package*⟩}{⟨*is not loaded*⟩}

`\IfPackagesLoaded` {⟨*list of packages*⟩}{⟨*all are loaded*⟩}

`\IfPackagesNotLoaded` {⟨*list of packages*⟩}{⟨*none is loaded*⟩}

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 LaTeX 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` `{`⟨*after this package*⟩`}{`⟨*execute this code*⟩`}`

`\ExecuteBeforePackage` `{`⟨*before this package*⟩`}{`⟨*execute this code*⟩`}`

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` `{`⟨*library*⟩`}{`⟨*if loaded*⟩`}`

Example

Executes the code only of 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

LaTeX provides no tool to check for the existence of a column type. This is provided by the following commands:

`\IfColumntypeDefined` {⟨*columntype character*⟩}{⟨*is defined*⟩}{⟨*is undefined*⟩}

`\IfColumntypesDefined` {⟨*columntype character list*⟩}{⟨*is defined*⟩}{⟨*is undefined*⟩}

Example

Executes the code only of 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}{llXX}
\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 LaTeX as names. The following commands provide a convenient way to check the existence of these color definitions.

`\IfColorDefined` {⟨*color name*⟩}{⟨*is defined*⟩}{⟨*is undefined*⟩}

`\IfColorsDefined` {⟨*list of color name*⟩}{⟨*is defined*⟩}{⟨*is undefined*⟩}

1.8 Math font version

`\IfMathVersionDefined` {⟨*font version*⟩}{⟨*is defined*⟩}{⟨*is undefined*⟩}

1.9 Glossaries styles

`\IfGlossariesStyleDefined` {⟨*style name*⟩}{⟨*is defined*⟩}

1.10 Bib environments

`\IfBibEnvironmentDefined` {⟨*environment name*⟩}{⟨*is defined*⟩}

1.11 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` {⟨*Group*⟩}{⟨*Property*⟩}{⟨*Code*⟩}

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

`\UseDefinition` {⟨*Group*⟩}{⟨*Property*⟩}

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

```
\NeedsTeXFormat{LaTeX2e}[1994/12/01]
\ProvidesPackage{templatetools}
          [2023/03/26 v0.2 Collection of conditional commands useful inside templates]
%
```

```
%%% --- Necessary Packages
%%% ----------------------------------------------------------------------
\RequirePackage{iftex}
\RequirePackage{etoolbox}
\RequirePackage{ltxcmds}
\RequirePackage{array} % for column types
\RequirePackage{ifdraft} % check draft
\RequirePackage{scrlfile}
%
```

2.1 Command sequences

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

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

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

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

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

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

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

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

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

```
\newcommand{\IfMultDefined}[1]{%
  \@tempswatrue
  \def\do##1{%
%% define \@tempa with trimmed index element.
    \edef\@tempa{\zap@space##1 \@empty}%
%% check if package of current index is loaded
    \ifcsdef{\@tempa}{}{\@tempswafalse}%
  }%
%% Process csv list with command \do (etoolbox)
```

```
  \docsvlist{#1}%
%% makes sure that the conditional works with one or two (if, else) parameters.
  \if@tempswa\expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi%
}
%
```

Thanks to EGREG, ANDREY VIHROV, MARTIN SCHARRER on 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.

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

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

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

`\IfNotDraftElse` Similar to `\IfDraft` but executes true and fals path.

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

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.

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

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

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

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

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

`\IfPackagesLoaded` Checks a list of packages

```
\newcommand{\IfPackagesLoaded}[1]{%
  \@tempswatrue
  \def\do##1{%
    \edef\@tempa{\zap@space##1 \@empty}%
    \ltx@ifpackageloaded{\@tempa}{}{\@tempswafalse}%
  }%
  \docsvlist{#1}%
  \if@tempswa\expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi%
}
%
```

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

```
\newcommand{\IfPackagesNotLoaded}[1]{%
  \@tempswatrue
  \def\do##1{%
    \edef\@tempa{\zap@space##1 \@empty}%
    \ltx@ifpackageloaded{\@tempa}{\@tempswafalse}{}%
  }%
  \docsvlist{#1}%
  \if@tempswa\expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi%
}
%
```

`\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.

```
\newcommand{\ExecuteAfterPackage}[2]{%
%% #1: after this package
%% #2: code to execute
  \AfterAtEndOfPackage{#1}{
    #2%
  }
  \AtEndPreamble{%
    \IfPackageNotLoaded{#1}{%
     #2%
    }
  }
}
%
```

`\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.

```
\newcommand{\ExecuteBeforePackage}[2]{%
%% #1: before this package
%% #2: code to execute
  \BeforePackage{#1}{
    #2%
```

```
  }
  \AtEndPreamble{%
    \IfPackageNotLoaded{#1}{%
     #2%
    }
  }
}
%
```

2.4 Tikz library

`\IfTikzLibraryLoaded` Checks if the tikz library is loaded

```
\def\IfTikzLibraryLoaded#1{%
  \ifcsname tikz@library@#1@loaded\endcsname
    \expandafter\@firstoftwo
  \else
    \expandafter\@secondoftwo
  \fi
}
%
```

Thanks to EGREG and MARCO DANIEL on 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 on the detection of column definitions.

Creates a list of predefined columntypes

```
\expandafter\let\csname columntype@l\endcsname\@empty
\expandafter\let\csname columntype@c\endcsname\@empty
\expandafter\let\csname columntype@r\endcsname\@empty
\expandafter\let\csname columntype@p\endcsname\@empty
\expandafter\let\csname columntype@m\endcsname\@empty
\expandafter\let\csname columntype@b\endcsname\@empty
\expandafter\let\csname columntype@@\endcsname\@empty
\expandafter\let\csname columntype@!\endcsname\@empty
\expandafter\let\csname columntype@|\endcsname\@empty
\expandafter\let\csname columntype@<\endcsname\@empty
\expandafter\let\csname columntype@>\endcsname\@empty
\expandafter\let\csname columntype@=\endcsname\@empty
%
```

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

```
\newcommand\CheckIfColumntypeDefined[1]{%
%% create the bool variable for column type
  \providebool{tpl@coltype@#1}
```

```
%% check if new column type of this name was created
  \ifcsdef{NC@find@\string#1}%
    {\setbool{tpl@coltype@#1}{true}}%
%% if not check if it is a predefined column type
    {\ifcsdef{columntype@\string#1}
      {\setbool{tpl@coltype@#1}{true}}%
      {\setbool{tpl@coltype@#1}{false}}%
    }%
}
%
```

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

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

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

```
\newcommand\IfColumntypeDefined[3]{%
%% Execute check which create bool variable
  \CheckIfColumntypeDefined{#1}
%% use bool variable for if sequence
  \ifboolexpr{ bool{\isColumntypeDefined{#1}} }{#2}{#3}%
}
%
```

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

```
\newcommand{\IfColumntypesDefined}[1]{%
  \@tempswatrue
  \def\do##1{%
    \edef\@tempa{\zap@space##1 \@empty}%
%% check if column is defined.
%%   Here with \expandafter because of the \string definition
%%   in \CheckIfColumntypeDefined.
    \expandafter\IfColumntypeDefined
        \expandafter{\@tempa}{}{\@tempswafalse}
  }%
  \docsvlist{#1}%
  \if@tempswa\expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi%
}
%
```

2.6 Color definitions

`\IfColorDefined` Tests if a color is defined

```
\newcommand{\IfColorDefined}[3]{%
  \ifcsdef{\string\color @#1}
```

```
    {#2} % color string is defined
    {#3}} % color string is not defined
%
```

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

```
\newcommand{\IfColorsDefined}[1]{%
  \@tempswatrue
  \def\do##1{%
    \edef\@tempa{\zap@space##1 \@empty}%
    \expandafter\IfColorDefined
        \expandafter{\@tempa}{}{\@tempswafalse}
  }%
  \docsvlist{#1}%
  \if@tempswa\expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi%
}
%
```

2.7 Math font version

\IfMathVersionDefined Checks if a mathversion font is defined.

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

Thanks to Werner and Ulrike Fischer for pointing me to this name definition on tex.stackexchange.com.

2.8 Glossaries styles

\IfGlossariesStyleDefined Checks if a glossaries style is defined.

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

Thanks to cgnieder on tex.stackexchange.com for the help to implement the command.

2.9 BibLaTeX Enivronemnts

\IfBibEnvironmentDefined Checks if a bib environment is defined.

```
\newcommand{\IfBibEnvironmentDefined}[1]{\ifcsdef{blx@env@#1}}
%
```

Thanks to egreg on tex.stackexchange.com for the help to implement the command.

2.10 Template definitions

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

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

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

```
\newcommand\UseDefinition[2]{%
  \ifcsdef{tpl@definition@#1@#2}
    {\csuse{tpl@definition@#1@#2}}
    {\PackageWarning{templatetools}%
      {Definition #1->#2 is unknown\MessageBreak}{}}%
}%
%
```

Index

Numbers written in italic 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.