

The `twoopt` package

Heiko Oberdiek*

2016/05/16 v1.6

Abstract

This package provides commands to define macros with two optional arguments.

Contents

1 Usage	1
2 Implementation	2
3 Installation	3
3.1 Download	3
3.2 Bundle installation	4
3.3 Package installation	4
3.4 Refresh file name databases	4
3.5 Some details for the interested	4
4 History	5
[1998/10/30 v1.0]	5
[1998/10/30 v1.1]	5
[1998/11/04 v1.2]	5
[1999/04/12 v1.3]	5
[2006/02/20 v1.4]	5
[2008/08/11 v1.5]	5
[2016/05/16 v1.6]	5
5 Index	5

1 Usage

`\newcommandtwoopt` Similar to `\newcommand`, `\renewcommand` and `\providecommand` this package provides commands to define macros with two optional arguments. The names of the `\renewcommandtwoopt` commands are built by appending the package name to the L^AT_EX-pendants:

```
\newcommandtwoopt   {\langle cmd \rangle} [\langle num \rangle] [\langle default1 \rangle] [\langle default2 \rangle] {\langle def. \rangle}
\renewcommandtwoopt {\langle cmd \rangle} [\langle num \rangle] [\langle default1 \rangle] [\langle default2 \rangle] {\langle def. \rangle}
\providecommandtwoopt {\langle cmd \rangle} [\langle num \rangle] [\langle default1 \rangle] [\langle default2 \rangle] {\langle def. \rangle}
```

Also the `*`-forms are supported. Indeed it is better to use this ones, unless it is intended to hold whole paragraphs in some of the arguments. If the macro is defined with the `*`-form, missing braces can be detected earlier.

Example:

*Please report any issues at <https://github.com/ho-tex/oberdiek/issues>

```

\newcommandtwoopt{\bsp}[3][AA][BB]{%
  \typeout{\string\bsp: #1,#2,#3}%
}
\bsp[aa][bb]{cc} → \bsp: aa,bb,cc
\bsp[aa]{cc}     → \bsp: aa,BB,cc
\bsp{cc}        → \bsp: AA,BB,cc

```

2 Implementation

```

1 (*package)
2 \NeedsTeXFormat{LaTeX2e}
3 \ProvidesPackage{twoopt}
4 [2016/05/16 v1.6 Definitions with two optional arguments (H0)]%

\newcommandtwoopt
5 \newcommand{\newcommandtwoopt}{%
6   \@ifstar{\@newcommandtwoopt*}{\@newcommandtwoopt}}%
7 }

\@newcommandtwoopt <#1>: star
                   <#2>: macro name to be defined
8 \newcommand{\@newcommandtwoopt}{%
9   \long\def\@newcommandtwoopt#1#2{%
10    \expandafter\@newcommandtwoopt
11     \csname2\string#2\endcsname{#1}{#2}%
12 }

\@@newcommandtwoopt <#1>: help command to be defined (\2<name>)
                   <#2>: star
                   <#3>: macro name to be defined
                   <#4>: number of total arguments
                   <#5>: default for optional argument one
                   <#6>: default for optional argument two
13 \newcommand{\@@newcommandtwoopt}{%
14   \long\def\@@newcommandtwoopt#1#2#3[#4][#5][#6]{%
15     \newcommand#2#3[1][#5]{%
16       \to@ScanSecondOptArg#1{##1}{#6}%
17     }%
18     \newcommand#2#1[#4]%
19 }

\renewcommandtwoopt
20 \newcommand{\renewcommandtwoopt}{%
21   \@ifstar{\@renewcommandtwoopt*}{\@renewcommandtwoopt}}%
22 }

\@renewcommandtwoopt <#1>: star
                    <#2>: command name to be defined
23 \newcommand{\@renewcommandtwoopt}{%
24   \long\def\@renewcommandtwoopt#1#2{%
25     \begingroup
26     \escapechar\m@ne
27     \xdef\@gtempa{\string#2}%
28   \endgroup
29   \expandafter\@ifundefined\@gtempa{%
30     \@latex@error{noexpand#2undefined}\@ehc
31   }-{}%
32   \let#2\@undefined
33   \expandafter\let\csname2\string#2\endcsname\@undefined
34   \expandafter\@newcommandtwoopt
35     \csname2\string#2\endcsname{#1}{#2}%
36 }

```

```

\providecommandtwoopt
37 \newcommand{\providecommandtwoopt}{%
38   \@ifstar{\providecommandtwoopt*}{\providecommandtwoopt}}%
39 }

\@providecommandtwoopt <#1>: star
<#2>: command name to be defined
40 \newcommand{\@providecommandtwoopt}{%
41   \long\def\@providecommandtwoopt#1#2{%
42     \begingroup
43       \escapechar\m@ne
44       \xdef\@gtempa{{\string#2}}%
45     \endgroup
46     \expandafter\@ifundefined\@gtempa{%
47       \expandafter\@newcommandtwoopt
48         \csname2\string#2\endcsname{#1}{#2}%
49     }{%
50       \let\t@dummyA\@undefined
51       \let\t@dummyB\@undefined
52       \@newcommandtwoopt\t@dummyA{#1}\t@dummyB
53     }%
54 }

\to@ScanSecondOptArg <#1>: help command to be defined (\2\<name>)
<#2>: first arg of command to be defined
<#3>: default for second opt. arg.
55 \newcommand{\to@ScanSecondOptArg}[3]{%
56   \@ifnextchar[{%
57     \expandafter#1\to@ArgOptToArgArg{#2}%
58   }{%
59     #1{#2}{#3}%
60   }%
61 }

\to@ArgOptToArgArg
62 \newcommand{\to@ArgOptToArgArg}{%
63   \long\def\to@ArgOptToArgArg#1[#2]{\{#1\}{#2}}

64 </package>

```

3 Installation

3.1 Download

Package. This package is available on CTAN¹:

[CTAN:macros/latex/contrib/oberdiek/twoopt.dtx](#) The source file.

[CTAN:macros/latex/contrib/oberdiek/twoopt.pdf](#) Documentation.

Bundle. All the packages of the bundle ‘oberdiek’ are also available in a TDS compliant ZIP archive. There the packages are already unpacked and the documentation files are generated. The files and directories obey the TDS standard.

[CTAN:install/macros/latex/contrib/oberdiek.tds.zip](#)

TDS refers to the standard “A Directory Structure for T_EX Files” ([CTAN:pkg/tds](#)). Directories with `texmf` in their name are usually organized this way.

¹[CTAN:pkg/twoopt](#)

3.2 Bundle installation

Unpacking. Unpack the `oberdiek.tds.zip` in the TDS tree (also known as `texmf` tree) of your choice. Example (linux):

```
unzip oberdiek.tds.zip -d ~/texmf
```

3.3 Package installation

Unpacking. The `.dtx` file is a self-extracting `docstrip` archive. The files are extracted by running the `.dtx` through plain `TEX`:

```
tex twoopt.dtx
```

TDS. Now the different files must be moved into the different directories in your installation TDS tree (also known as `texmf` tree):

```
twoopt.sty → tex/latex/oberdiek/twoopt.sty
twoopt.pdf → doc/latex/oberdiek/twoopt.pdf
twoopt.dtx → source/latex/oberdiek/twoopt.dtx
```

If you have a `docstrip.cfg` that configures and enables `docstrip`'s TDS installing feature, then some files can already be in the right place, see the documentation of `docstrip`.

3.4 Refresh file name databases

If your `TEX` distribution (`TEX Live`, `MiKTEX`, ...) relies on file name databases, you must refresh these. For example, `TEX Live` users run `texhash` or `mktexlsr`.

3.5 Some details for the interested

Unpacking with \LaTeX . The `.dtx` chooses its action depending on the format:

plain `TEX`: Run `docstrip` and extract the files.

\LaTeX : Generate the documentation.

If you insist on using \LaTeX for `docstrip` (really, `docstrip` does not need \LaTeX), then inform the autodetect routine about your intention:

```
latex \let\install=y\input{twoopt.dtx}
```

Do not forget to quote the argument according to the demands of your shell.

Generating the documentation. You can use both the `.dtx` or the `.drv` to generate the documentation. The process can be configured by the configuration file `ltxdoc.cfg`. For instance, put this line into this file, if you want to have A4 as paper format:

```
\PassOptionsToClass{a4paper}{article}
```

An example follows how to generate the documentation with `pdf \LaTeX` :

```
pdflatex twoopt.dtx
makeindex -s gind.ist twoopt.idx
pdflatex twoopt.dtx
makeindex -s gind.ist twoopt.idx
pdflatex twoopt.dtx
```

4 History

[1998/10/30 v1.0]

- The first version was built as a response to a question of Rebecca and Rowland², published in the newsgroup `comp.text.tex`:
“Re: [Q] LaTeX command with two optional arguments?”³

[1998/10/30 v1.1]

- Improvements added in response to Stefan Ulrich⁴ in the same thread:
“Re: [Q] LaTeX command with two optional arguments?”⁵

[1998/11/04 v1.2]

- Fixes for LaTeX bugs 2896, 2901, 2902 added.

[1999/04/12 v1.3]

- Fixes removed because of LaTeX [1998/12/01].
- Documentation in dtx format.
- Copyright: LPPL (`CTAN:macros/latex/base/lppl.txt`)
- First CTAN release.

[2006/02/20 v1.4]

- Code is not changed.
- New DTX framework.
- LPPL 1.3

[2008/08/11 v1.5]

- Code is not changed.
- URLs updated from `www.dejanews.com` to `groups.google.com`.

[2016/05/16 v1.6]

- Documentation updates.

5 Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; plain numbers refer to the code lines where the entry is used.

Symbols	
<code>\@ifundefined</code>	29, 46
<code>\@newcommandtwoopt</code> . 10, <u>13</u> , 34, 47, 52	<code>\@latex@error</code> 30
<code>\@ehc</code> 30	<code>\@newcommandtwoopt</code> 6, <u>8</u>
<code>\@gtempa</code> 27, 29, 44, 46	<code>\@providecommandtwoopt</code> 38, <u>40</u>
<code>\@ifnextchar</code> 56	<code>\@renewcommandtwoopt</code> 21, <u>23</u>
<code>\@ifstar</code> 6, 21, 38	<code>\@undefined</code> 32, 33, 50, 51

²Rebecca and Rowland’s email address: `rebecca@astrid.u-net.com`

³Url: <https://groups.google.com/group/comp.text.tex/msg/0ab1afde7b172d37>

⁴Stefan Ulrich’s email address: `ulrich@cis.uni-muenchen.de`

⁵Url: <https://groups.google.com/group/comp.text.tex/msg/b8d84d4336f302c4>

	C				
\csname	11, 33, 35, 48		\newcommandtwoopt 1, <u>5</u>
	E				P
\endcsname	11, 33, 35, 48		\providecommandtwoopt 1, <u>37</u>
\escapechar	26, 43		\ProvidesPackage 3
	M				R
\m@ne	26, 43		\renewcommandtwoopt 1, <u>20</u>
	N				T
\NeedsTeXFormat	2		\to@ArgOptToArgArg 57, <u>62</u>
\newcommand	5, 8, 13, 15, 18, 20, 23, 37, 40, 55, 62		\to@dummyA 50, <u>52</u>
				\to@dummyB 51, <u>52</u>
				\to@ScanSecondOptArg 16, <u>55</u>