

# The `twoopt` package

Heiko Oberdiek  
<heiko.oberdiek at gmail.com>

2008/08/11 v1.5

## Abstract

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

## Contents

<b>1 Usage</b>	<b>1</b>
<b>2 Implementation</b>	<b>2</b>
<b>3 Installation</b>	<b>3</b>
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
<b>4 History</b>	<b>5</b>
[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
<b>5 Index</b>	<b>6</b>

## 1 Usage

`\newcommandtwoopt` Similar to `\newcommand`, `\renewcommand` and `\providecommand` this package provides commands to define macros with two optional arguments. The names of the commands are built by appending the package name to the L<sup>A</sup>T<sub>E</sub>X-pendants:

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

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:

```

\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 [2008/08/11 v1.5 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\to@dummyA\@undefined
51 \let\to@dummyB\@undefined
52 \@newcommandtwoopt\to@dummyA{#1}\to@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 CTAN<sup>1</sup>:

[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<sub>E</sub>X Files” ([CTAN:tds/tds.pdf](#)). Directories with `texmf` in their name are usually organized this way.

<sup>1</sup>[ftp://ftp.ctan.org/tex-archive/](http://ftp.ctan.org/tex-archive/)

### 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
```

**Script installation.** Check the directory `TDS:scripts/oberdiek/` for scripts that need further installation steps. Package `attachfile2` comes with the Perl script `pdfatfi.pl` that should be installed in such a way that it can be called as `pdfatfi`. Example (linux):

```
chmod +x scripts/oberdiek/pdfatfi.pl
cp scripts/oberdiek/pdfatfi.pl /usr/local/bin/
```

### 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 (`teTEX`, `mikTEX`, ...) relies on file name databases, you must refresh these. For example, `teTEX` users run `texhash` or `mktextlsr`.

### 3.5 Some details for the interested

**Attached source.** The PDF documentation on CTAN also includes the `.dtx` source file. It can be extracted by AcrobatReader 6 or higher. Another option is `pdftk`, e.g. unpack the file into the current directory:

```
pdftk twoopt.pdf unpack_files output .
```

**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 pdfL<sup>A</sup>T<sub>E</sub>X:

```
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 Rowland<sup>2</sup>, published in the newsgroup [comp.text.tex](#):  
“Re: [Q] LaTeX command with two optional arguments?”<sup>3</sup>

### [1998/10/30 v1.1]

- Improvements added in response to Stefan Ulrich<sup>4</sup> in the same thread:  
“Re: [Q] LaTeX command with two optional arguments?”<sup>5</sup>

### [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](#).

---

<sup>2</sup>Rebecca and Rowland’s email address: [rebecca@astrid.u-net.com](mailto:rebecca@astrid.u-net.com)

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

<sup>4</sup>Stefan Ulrich’s email address: [ulrich@cis.uni-muenchen.de](mailto:ulrich@cis.uni-muenchen.de)

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

## 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; numbers in roman refer to the code lines where the entry is used.

Symbols		M	
<code>\@newcommandtwopt</code>	10, <u>13</u> , 34, 47, 52	<code>\m@ne</code>	26, 43
<code>\@ehc</code>	30	N	
<code>\@gtempa</code>	27, 29, 44, 46	<code>\NeedsTeXFormat</code>	2
<code>\@ifnextchar</code>	56	<code>\newcommand</code>	5, 8, 13, 15, 18, 20, 23, 37, 40, 55, 62
<code>\@ifstar</code>	6, 21, 38	<code>\newcommandtwopt</code>	1, <u>5</u>
<code>\@ifundefined</code>	29, 46	P	
<code>\@latex@error</code>	30	<code>\providecommandtwopt</code>	1, <u>37</u>
<code>\@newcommandtwopt</code>	6, <u>8</u>	<code>\ProvidesPackage</code>	3
<code>\@providecommandtwopt</code>	38, <u>40</u>	R	
<code>\@renewcommandtwopt</code>	21, <u>23</u>	<code>\renewcommandtwopt</code>	1, <u>20</u>
<code>\@undefined</code>	32, 33, 50, 51	T	
C		<code>\to@ArgOptToArgArg</code>	57, <u>62</u>
<code>\csname</code>	11, 33, 35, 48	<code>\to@dummyA</code>	50, 52
E		<code>\to@dummyB</code>	51, 52
<code>\endcsname</code>	11, 33, 35, 48	<code>\to@ScanSecondOptArg</code>	16, <u>55</u>
<code>\escapechar</code>	26, 43		