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

`\newcommandtwoopt`
`\renewcommandtwoopt`
`\providecommandtwoopt`

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 LaTeX-pendants:

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

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

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

```
⟨*package⟩
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{twoopt}
  [2016/05/16 v1.6 Definitions with two optional arguments (HO)]%
```

\newcommandtwoopt

```
\newcommand{\newcommandtwoopt}{%
  \@ifstar{\@newcommandtwoopt*}{\@newcommandtwoopt{}}%
}
```

\@newcommandtwoopt

⟨#1⟩: star
⟨#2⟩: macro name to be defined

```
\newcommand{\@newcommandtwoopt}{}
\long\def\@newcommandtwoopt#1#2{%
  \expandafter\@@newcommandtwoopt
    \csname2\string#2\endcsname{#1}{#2}%
}
```

\@@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

```
\newcommand{\@@newcommandtwoopt}{}
\long\def\@@newcommandtwoopt#1#2#3[#4][#5][#6]{%
  \newcommand#2#3[1][{#5}]{%
    \to@ScanSecondOptArg#1{##1}{#6}%
  }%
  \newcommand#2#1[{#4}]%
}
```

\renewcommandtwoopt

```
\newcommand{\renewcommandtwoopt}{%
  \@ifstar{\@renewcommandtwoopt*}{\@renewcommandtwoopt{}}%
}
```

\@renewcommandtwoopt

⟨#1⟩: star
⟨#2⟩: command name to be defined

```
\newcommand{\@renewcommandtwoopt}{}
\long\def\@renewcommandtwoopt#1#2{%
  \begingroup
```

```
    \escapechar\m@ne
    \xdef\@gtempa{{\string#2}}%
  \endgroup
  \expandafter\@ifundefined\@gtempa{%
    \@latex@error{\noexpand#2undefined}\@ehc
  }{}%
  \let#2\@undefined
  \expandafter\let\csname2\string#2\endcsname\@undefined
  \expandafter\@@newcommandtwoopt
    \csname2\string#2\endcsname{#1}{#2}%
}
```

`\providecommandtwoopt`

```
\newcommand{\providecommandtwoopt}{%
  \@ifstar{\@providecommandtwoopt*}{\@providecommandtwoopt{}}%
}
```

`\@providecommandtwoopt` ⟨*#1*⟩: star
⟨*#2*⟩: command name to be defined

```
\newcommand{\@providecommandtwoopt}{}
\long\def\@providecommandtwoopt#1#2{%
  \begingroup
    \escapechar\m@ne
    \xdef\@gtempa{{\string#2}}%
  \endgroup
  \expandafter\@ifundefined\@gtempa{%
    \expandafter\@@newcommandtwoopt
      \csname2\string#2\endcsname{#1}{#2}%
  }{%
    \let\to@dummyA\@undefined
    \let\to@dummyB\@undefined
    \@@newcommandtwoopt\to@dummyA{#1}\to@dummyB
  }%
}
```

`\to@ScanSecondOptArg` ⟨*#1*⟩: help command to be defined (`\2\<name>`)
⟨*#2*⟩: first arg of command to be defined
⟨*#3*⟩: default for second opt. arg.

```
\newcommand{\to@ScanSecondOptArg}[3]{%
  \@ifnextchar[{%
    \expandafter#1\to@ArgOptToArgArg{#2}%
  }{%
    #1{#2}{#3}%
  }%
}
```

`\to@ArgOptToArgArg`

```
\newcommand{\to@ArgOptToArgArg}{}
\long\def\to@ArgOptToArgArg#1[#2]{{#1}{#2}}
```

⟨/package⟩

3 Installation

3.1 Download

Package. This package is available on CTAN[1]:

[1] CTAN:pkg/twoopt

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 TEX Files" (CTAN:pkg/tds). Directories with `texmf` in their name are usually organized this way.

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 pdfLaTeX:

```
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[2], 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 Ulrich[4] 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`.

[2] Rebecca and Rowland's email address: rebecca@astrid.u-net.com
[3] Url: https://groups.google.com/group/comp.text.tex/msg/0ab1afde7b172d37
[4] Stefan Ulrich's email address: ulrich@cis.uni-muenchen.de
[5] Url: https://groups.google.com/group/comp.text.tex/msg/b8d84d4336f302c4

[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.