The makerobust package

Heiko Oberdiek*
<heiko.oberdiek at googlemail.com>

2016/05/16 v1.1

Abstract

Package makerobust provides \MakeRobustCommand that converts an existing macro to a robust one.

Contents

1 User interface **1**
1.1 Example 2

2 Implementation **2**

3 Installation **3**
3.1 Download 3
3.2 Bundle installation 3
3.3 Package installation 3
3.4 Refresh file name databases 4
3.5 Some details for the interested 4

4 Catalogue **4**

5 History **5**
[2006/03/18 v1.0] 5
[2016/05/16 v1.1] 5

6 Index **5**

1 User interface

LaTeX offers \DeclareRobustCommand to define a robust macro that does not break if it is used in moving arguments. Sometimes a macro is already defined, but not robust. For example, \(and \) are not robust, inside \section the user must use \protect explicitly. This could be avoided by making \(and \) robust.

\MakeRobustCommand{⟨*cmd*⟩}

\MakeRobustCommand redefines the macro ⟨*cmd*⟩ by using \DeclareRobustCommand and the existing definition of the macro ⟨*cmd*⟩.

- It is an error if ⟨*cmd*⟩ is undefined. If you want to define a robust command, then you can use \DeclareRobustCommand directly.

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

- If the macro has previously been defined by \DeclareRobustCommand then the redefinition of \MakeRobustCommand is omitted, because the macro is already robust. Only an information entry is written to the .log file. Thus you do not get a warning or an error if the macro is already robust because of an updated LaTeX or package that defines the macro.
- Two macros are defined for a macro, defined by \DeclareRobustCommand. Example:

  ```
  \DeclareRobustCommand{\foobar}{definition text}
  ```

  Then the macro "\foobar" contains the protection code and, depending on the protection mode, calls the internal macro "\foobar ". Notice the space at the end of the macro name. This internal macro "\foobar " now contains the definition "definition text", given in \DeclareRobustCommand.

  Sometimes it can happen, that the internal macro already exists. This can be caused by a previous \DeclareRobustCommand followed by \renewcommand. Then the redefinition by \MakeRobustCommand would be safe.

  However, it can also be possible that the macro is already robust, using the internal macro, but with a different protection code. The redefinition by \MakeRobustCommand would then generate an infinite loop.

  Therefore \MakeRobustCommand raises an error message, if the internal macro (with space at the end) already exists.

1.1 Example

```
⟨*example⟩
\documentclass{article}
\usepackage{makerobust}
\MakeRobustCommand\(
\MakeRobustCommand\)
\pagestyle{headings}
\begin{document}
\tableofcontents
\section{Einstein: \(E=mc^2\)}
\newpage
Second page.
\end{document}
⟨/example⟩
```

2 Implementation

```
⟨*package⟩
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{makerobust}%
  [2016/05/16 v1.1 Make existing macro robust (HO)]%

\def\MakeRobustCommand#1{%
  \begingroup
  \@ifundefined{\expandafter\@gobble\string#1}{%
    \endgroup
    \PackageError{makerobust}{%
      Macro \string`\string#1\string' is not defined%
    }\@ehc
  }{%
    \global\let\MR@gtemp#1%
    \let#1\@undefined
    \expandafter\let\expandafter\MR@temp
       \csname\expandafter\@gobble\string#1 \endcsname
    \DeclareRobustCommand#1{}%
    \ifx#1\MR@gtemp
```

```
    \endgroup
    \PackageInfo{makerobust}{%
      \string`\string#1\string' is already robust%
    }%
  \else
    \@ifundefined{MR@temp}{%
      \global\let\MR@gtemp#1%
      \endgroup
      \expandafter\let\csname\expandafter\@gobble\string#1 \endcsname#1%
      \let#1\MR@gtemp
    }{%
      \endgroup
      \PackageError{makerobust}{%
        Internal macro \string`\string#1 \string' already exists%
      }\@ehc
    }%
  \fi
}%
}
```

⟨/package⟩

3 Installation

3.1 Download

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

CTAN:macros/latex/contrib/oberdiek/makerobust.dtx The source file.

CTAN:macros/latex/contrib/oberdiek/makerobust.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:tds/tds.pdf). 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
```

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 makerobust.dtx
```

[1] http://ctan.org/pkg/makerobust

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

```
makerobust.sty          → tex/latex/oberdiek/makerobust.sty
makerobust.pdf          → doc/latex/oberdiek/makerobust.pdf
makerobust-example.tex → doc/latex/oberdiek/makerobust-example.tex
makerobust.dtx          → source/latex/oberdiek/makerobust.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 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{makerobust.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 makerobust.dtx
makeindex -s gind.ist makerobust.idx
pdflatex makerobust.dtx
makeindex -s gind.ist makerobust.idx
pdflatex makerobust.dtx
```

4 Catalogue

The following XML file can be used as source for the TEX Catalogue. The elements caption and description are imported from the original XML file from the Catalogue. The name of the XML file in the Catalogue is makerobust.xml.

```
⟨*catalogue⟩
<?xml version='1.0' encoding='us-ascii'?>
<!DOCTYPE entry SYSTEM 'catalogue.dtd'>
<entry datestamp='$Date$' modifier='$Author$' id='makerobust'>
  <name>makerobust</name>
  <caption>Making a macro robust.</caption>
  <authorref id='auth:oberdiek'/>
  <copyright owner='Heiko Oberdiek' year='2006'/>
  <license type='lppl1.3'/>
  <version number='1.1'/>
```

```
<description>
  This package provides the command MakeRobustCommand
  that converts an existing macro to a robust one.
  <p/>
  The package is part of the <xref refid='oberdiek'>oberdiek</xref>
  bundle.
</description>
<documentation details='Package documentation'
    href='ctan:/macros/latex/contrib/oberdiek/makerobust.pdf'/>
<ctan file='true' path='/macros/latex/contrib/oberdiek/makerobust.dtx'/>
<miktex location='oberdiek'/>
<texlive location='oberdiek'/>
<install path='/macros/latex/contrib/oberdiek/oberdiek.tds.zip'/>
</entry>
⟨/catalogue⟩
```

5 History

[2006/03/18 v1.0]

- First version.

[2016/05/16 v1.1]

- Documentation updates.

6 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

\(. 4, 9
\) 5, 9
\@ehc 24, 46
\@gobble 20, 29, 40
\@ifundefined 20, 37
\@undefined 27

B

\begin 7

C

\csname 29, 40

D

\DeclareRobustCommand 30
\documentclass 2

E

\end 12
\endcsname 29, 40

I

\ifx 31

M

\MakeRobustCommand *1*, 4, 5, 18
\MR@gtemp 26, 31, 38, 41
\MR@temp 28

N

\NeedsTeXFormat 15
\newpage 10

P

\PackageError 22, 44
\PackageInfo 33
\pagestyle 6
\ProvidesPackage 16

S

\section 9

T

\tableofcontents 8

U

\usepackage 3