The **protecteddef** package

Heiko Oberdiek

<heiko.oberdiek at googlemail.com>

2011/01/31 v1.0

Abstract

This package provides \texttt{\textbackslash protected\textbackslash def} for defining robust macros for both plain \TeX{} and \LaTeX{}. First \texttt{\textbackslash \textbackslash e\textbackslash TeX\textbackslash s \textbackslash protected} is tried, then \LaTeX{}'s \texttt{\textbackslash \textbackslash declare\textbackslash robust\textbackslash command} is used. Otherwise the macro is not made robust.

Contents

1 Documentation 1
   1.1 The \LaTeX{}'s way 2
   1.2 The \texttt{\textbackslash \textbackslash e\textbackslash TeX\textbackslash s \textbackslash way} 2
   1.3 The way of this package 2
   1.4 Usage 2

2 Implementation 2
   2.1 Reload check and package identification 2
   2.2 Catcodes 4
   2.3 Resources 4

3 Test 6
   3.1 Catcode checks for loading 6
   3.2 Test without \LaTeX{} and \texttt{\textbackslash \textbackslash protected} 8

4 Installation 11
   4.1 Download 11
   4.2 Bundle installation 11
   4.3 Package installation 11
   4.4 Refresh file name databases 11
   4.5 Some details for the interested 12

5 Catalogue 12

6 History 13
   [2011/01/31 v1.0] 13

7 Index 13

1 Documentation

Many of my packages work for both formats plain \TeX{} and \LaTeX{}, even init\TeX{} is often supported. It would be nice if fragile macros could be protected and made robust. However the different format worlds offer different solutions.
1.1 The \LaTeX\’s way

Usually \texttt{\newcommand} is used to define macros. It provides a check if the command to be defined is already defined or cannot be defined for other reasons.

For making robust macros \LaTeX\ provides \texttt{\DeclareRobustCommand}. It shares the syntax with \texttt{\newcommand}. However it does not provide latters check. Internally the check is available via \texttt{\@ifdefinable}.

Internally the robust macro is using \texttt{\protect} with a nested macro definition. The \texttt{\protect} infrastructure is a feature of \LaTeX\ and usually not available in other formats.

1.2 The -\TeX\’s way

The need for robust macros is addressed in \eTeX. It provides \texttt{\protected} that modifies the behaviour of \texttt{\def} in a similar way as \texttt{\long}. A protected macro does not expand in some expandable contexts like writing to a file or \texttt{\edef}.

1.3 The way of this package

The package tries to find the available protection mechanism. First it looks for \eTeX\’s \texttt{\protected}, then it uses \LaTeX\’s \texttt{\DeclareRobustCommand}. If both fails, then the macro remains unprotected.

Additionally, \LaTeX\’s check, if a macro is already defined is added in all cases. First \LaTeX\’s \texttt{\@ifdefinable} is tried to be compatible with \LaTeX. If \texttt{\@ifdefinable} is not available, then the test is implemented by asserting that the macro is undefined or has the meaning of \texttt{\relax}. If the test fails, then in all cases the macro is not defined and an error is thrown.

1.4 Usage

\begin{verbatim}
\ProtectedDef* {⟨cmd⟩} [⟨num⟩] {⟨definition text⟩}
\end{verbatim}

Macro \texttt{\ProtectedDef} follows the syntax of \LaTeX\’s \texttt{\newcommand} with the exception that an optional argument is not supported. Macro \langle cmd \rangle is to be defined as \texttt{\long} macro without star with \langle num \rangle arguments.

The number of arguments \langle num \rangle must be given as explicite digit 0 upto 9. Otherwise the part between the argument \langle cmd \rangle and the \langle definition text \rangle is taken as parameter text in the syntax of vanilla \TeX. Examples (with \texttt{\protected}):

\begin{verbatim}
\ProtectedDef*{\foo}[1]{\message{#1}}
⇒ \protected\def\foo#1{\message#1}
\ProtectedDef{\foo}{abc}
⇒ \protected\def\foo{abc}
\ProtectedDef*{\foo(#1)<#2>{#1/#2}}
⇒ \protected\def\foo(#1)<#2>{#1/#2}
\end{verbatim}

2 Implementation

2.1 Reload check and package identification

Reload check, especially if the package is not used with \LaTeX.

\begin{verbatim}
\begingroup\catcode61=10\relax%\endlinechar=13 %
\catcode35=6 % 
\catcode39=12 % 
\end{verbatim}
\catcode44=12 \ ,
\catcode45=12 \ -
\catcode46=12 \ .
\catcode58=12 \ :
\catcode64=11 \ @
\catcode123=1 \{
\catcode125=2 \ }
\expandafter\let\expandafter\x\csname ver@protecteddef.sty\endcsname
\ifx\x\relax \plain-TeX, first loading
\else
\expandafter\ifx\csname PackageInfo\endcsname\relax
\def\x#1#2{\immediate\write-1{Package #1 Info: #2.}}%
\else
\def\x#1#2\[#3] {
\PackageInfo{#1}{#2, stopped}
\fi
\x{protecteddef}{The package is already loaded}%
\aftergroup\endinput
\fi
\fi
\endgroup%
Package identification:
\begingroup\catcode61\catcode48\catcode32=10\relax%
\catcode13=5 \ ^\ ~\ M
\endlinechar=13 \ %
\catcode35=6 \ #
\catcode39=12 \ '
\catcode40=12 \ ( 
\catcode41=12 \ )
\catcode44=12 \ ,
\catcode45=12 \ -
\catcode46=12 \ .
\catcode47=12 \ /
\catcode58=12 \ : 
\catcode64=11 \ @
\catcode91=12 \ [ 
\catcode93=12 \ ]
\catcode123=1 \{
\catcode125=2 \ }
\expandafter\ifx\csname ProvidesPackage\endcsname\relax
\def\x#1#2#3[#4]{\endgroup
\immediate\write-1{Package: #3 #4}%
\xdef#1{#4}%
}%
\else
\def\x#1#2[#3]{\endgroup
\#2[{#3}]% 
\ifx\x\undefined
\xdef#1[{#3}]% 
\fi
\ifx\x\relax
\xdef#1{#3}%
\fi
}%
\fi
\expandafter\ifx\csname ver@protecteddef.sty\endcsname
\ProvidesPackage{protecteddef}%
\fi
2.2 Catcodes

```
\begingroup\catcode61\catcode48\catcode32=10\relax
\catcode13=5 \^^M
\endlinechar=13 \%
\catcode123=1 % {
\catcode125=2 % }
\def\x{\endgroup
\expandafter\edef\csname ProDef@AtEnd\endcsname{%
\endlinechar=\the\endlinechar\relax
\catcode13=\the\catcode13\relax
\catcode32=\the\catcode32\relax
\catcode35=\the\catcode35\relax
\catcode61=\the\catcode61\relax
\catcode64=\the\catcode64\relax
\catcode123=\the\catcode123\relax
\catcode125=\the\catcode125\relax
}%
\x\catcode61\catcode48\catcode32=10\relax
\catcode13=5 \^^M
\endlinechar=13 \%
\catcode32=6 % 
\catcode64=11 % @
\catcode35=6 % #
\catcode123=1 % {
\catcode125=2 % }
\def\TMP@EnsureCode#1#2{%
\edef\ProDef@AtEnd{\ProDef@AtEnd
\catcode#1=\the\catcode#1\relax
\catcode#1=#2\relax
}/%\edef\ProDef@AtEnd{\ProDef@AtEnd
\catcode#1=#2\relax
}
\TMP@EnsureCode{38}{4}% &
\TMP@EnsureCode{40}{12}% ( 
\TMP@EnsureCode{41}{12}% )
\TMP@EnsureCode{42}{12}% *
\TMP@EnsureCode{45}{12}% -
\TMP@EnsureCode{46}{12}% .
\TMP@EnsureCode{47}{12}% /
\TMP@EnsureCode{91}{12}% [
\TMP@EnsureCode{93}{12}% ]
\TMP@EnsureCode{96}{12}% `%
\edef\ProDef@AtEnd{\ProDef@AtEnd\noexpand\endinput}
```

2.3 Resources

```
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname RequirePackage\endcsname\relax
\def\TMP@RequirePackage#1[#2]{%
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname ver@#1.sty\endcsname\relax
\input #1.sty\relax
\fi
\else
\let\TMP@RequirePackage\RequirePackage
\fi
\TMP@RequirePackage{ltxcmds}[2010/12/12]%
\TMP@RequirePackage{infwarerr}[2010/04/08]%
```

\def\ProDef@temp#1{% 
\expandafter\def\csname ProDef@param[#1]\endcsname % hash-ok
}
\ProDef@temp0{}
\ProDef@temp1{##1}
\ProDef@temp2{##1##2}
\ProDef@temp3{##1##2##3}
\ProDef@temp4{##1##2##3##4}
\ProDef@temp5{##1##2##3##4##5}
\ProDef@temp6{##1##2##3##4##5##6}
\ProDef@temp7{##1##2##3##4##5##6##7}
\ProDef@temp8{##1##2##3##4##5##6##7##8}
\ProDef@temp9{##1##2##3##4##5##6##7##8##9}

\ProDef@IfDefinable
\ltx@ifUndefined{@ifdefinable}{% 
\long\def\ProDef@IfDefinable#1{% 
\begingroup 
\escapechar=-1 %
\ltx@ifUndefined{\string#1}{% 
\endgroup 
\ltx@firstofone 
\}%
\expandafter\endgroup
\expandafter\ifx\csname protected\endcsname\relax 
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname DeclareRobustCommand\endcsname\relax
\catcode`\&=14 % comment
\else
\newcommand*{\ProtectedDef}{% 
\ltx@ifnextchar*{\ProDef@ProtectedDef
}\{%
\ProDef@ProtectedDef{}
%
\ProDef@ProtectedDef{}}%
%
\long\def\ProDef@ProtectedDef#1#2#{% 
\ProDef@IfDefinable[#2]{% 
\ltx@ifUndefined{\ProDef@param#3}{% 
\DeclareRobustCommand*{\ProDef@param#3}{% 
\begingroup 
\escapechar=-1 %
\def\ProDef@temp{#1}%

3 Test

3.1 Catcode checks for loading

3 Test
3.2 Test without L\TeX{} and \texttt{\textbackslash protected}

\begin{verbatim}
(*test2*)
\errorcontextlines=10000 \%
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname RequirePackage\endcsname\relax
  \input protecteddef.sty\relax
\catcode`\{=1 \%
\catcode`\}=2 \%
\catcode`\#=6 \%
\else
  \RequirePackage{protecteddef}[2011/01/31]\%
\fi
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname protected\endcsname\relax
  \let\pdef\def
\else
  \def\pdef{\protected\def}\%
\fi
\def\msg#1{\immediate\write16{\the\errcount\!#1}}\%
\countdef\errcount=2 \%
\long\def\BeginCheck#1\ProtectedDef#2\EndCheck{\begingroup\toks0={\ProtectedDef#2}\msg{\the\toks0}\endgroup\ifdim\wd0=0pt\relax\else\errmessage{[Definition] Unwanted spaces?!}\fi}\%
\end{verbatim}
4 Installation

4.1 Download

Package. This package is available on CTAN:\footnote{ftp://ftp.ctan.org/tex-archive/}

\verb|CTAN:macros/latex/contrib/oberdiek/protecteddef.dtx| The source file.
\verb|CTAN:macros/latex/contrib/oberdiek/protecteddef.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.

\verb|CTAN:install/macros/latex/contrib/oberdiek.tds.zip|

\textit{TDS} refers to the standard “A Directory Structure for \LaTeX Files” (\texttt{CTAN:tds/tds.pdf}). Directories with \texttt{texmf} in their name are usually organized this way.

4.2 Bundle installation

Unpacking. Unpack the \verb|oberdiek.tds.zip| in the TDS tree (also known as \texttt{texmf} tree) of your choice. Example (linux):

\begin{verbatim}
unzip oberdiek.tds.zip -d ~/texmf
\end{verbatim}

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

\begin{verbatim}
chmod +x scripts/oberdiek/pdfatfi.pl
cp scripts/oberdiek/pdfatfi.pl /usr/local/bin/
\end{verbatim}

4.3 Package installation

Unpacking. The \texttt{.dtx} file is a self-extracting \texttt{docstrip} archive. The files are extracted by running the \texttt{.dtx} through \LaTeX:

\begin{verbatim}
tex protecteddef.dtx
\end{verbatim}

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

\begin{verbatim}
protecteddef.sty → tex/generic/oberdiek/protecteddef.sty
protecteddef.pdf → doc/latex/oberdiek/protecteddef.pdf
test/protecteddef-test1.tex → doc/latex/oberdiek/test/protecteddef-test1.tex
test/protecteddef-test2.tex → doc/latex/oberdiek/test/protecteddef-test2.tex
protecteddef.dtx → source/latex/oberdiek/protecteddef.dtx
\end{verbatim}

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

4.4 Refresh file name databases

If your \LaTeX distribution (\teX, \mikTeX, ...) relies on file name databases, you must refresh these. For example, \teX users run \texttt{texhash} or \texttt{mktexlsr}.
4.5 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 protecteddef.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{protecteddef.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 protecteddef.dtx
makeindex -s gind.ist protecteddef.idx
pdflatex protecteddef.dtx
makeindex -s gind.ist protecteddef.idx
pdflatex protecteddef.dtx
```

5 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 `protecteddef.xml`.

```
<catalogue>
  <xml version='1.0' encoding='us-ascii'/>
  <!DOCTYPE entry SYSTEM 'catalogue.dtd'>
  <entry datestamp='$Date$' modifier='$Author$' id='protecteddef'>
    <name>protecteddef</name>
    <caption>Define protected commands.</caption>
    <authorref id='auth:oberdiek'/>
    <copyright owner='Heiko Oberdiek' year='2011'/>
    <license type='lppl1.3'/>
    <version number='1.0'/>
    <description>
      The package defines a command \texttt{\ProtectedDef} that will
      create LaTeX 'robust' command or an e-TeX
      \texttt{\protected\&\#x2018;\texttt{\textbackslash}x2019;} command as appropriate
      to its environment.
    </description>
  </entry>
</catalogue>
```

The package is part of the `<xref refid='oberdiek'>oberdiek</xref>` bundle.
6 History

[2011/01/31 v1.0]

- First public version.

7 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       | \#               | 235, 338 |
|              | \%              | 311      |
|              | \&              | 169, 201 |
|              | @              | 236, 309 |
|              | PackageError    | 150      |
|              | @firstofone     | 244, 247 |
|              | @gobble         | 241, 249 |
|              | @ifdefinable    | 159      |
|              | @undefined      | 58       |
|              | \              | 310      |
|              | \{              | 233, 336 |
|              | \}              | 234, 337 |

A

\advance          | 274, 282, 297, 369, 372 |
\aftergroup       | 29       |

B

\BeginCheck       | 350, 441, 446, 451, 456, 461, 466, 471, 476 |
\body             | 253, 257 |

C

\catcode          | 2, 3, 5, 6, 7, 8, 9, 10, 11, 12, 13, 33, 34, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 69, 70, 72, 73, 74, 78, 79, 80, 81, 82, 83, 84, 87, 88, 90, 91, 92, 93, 97, 99, 169, 201, 233, 234, 235, 236, 271, 280, 288, 292, 309, 310, 311, 336, 337, 338 |
\chardef         | 390, 396 |
\check           | 358, 400, 411 |
\CMP             | 417, 421, 433 |
\cmp             | 401, 406, 422, 427, 442, 447, 452, 457, 462, 467, 472, 477 |
\count@          | 238, 267, 271, 273, 274, 278, 280, 281, 282, 286, 288, 291, 292, 296, 297 |
\countdef        | 238, 349 |
\cs              | 415, 419, 422, 426 |
\csname          | 14, 21, 50, 66, 76, 113, 116, 126, 128, 166, 168, 190, 226, 237, 240, 243, 246, 301, 328, 334, 343, 368, 371, 392, 394, 419, 422, 426, 481 |
\DeclareRobustCommand | 181, 194 |
\DeclareVersion  | 390, 396, 399 |
\empty           | 17, 18       |
\end             | 329, 481     |
\EndCheck        | 350, 444, 449, 454, 459, 464, 469, 474, 479 |
\endcsname       | 14, 21, 50, 66, 76, 113, 116, 126, 128, 166 |
\endinput        | 481, 446, 449, 454, 459, 464, 469, 474, 479 |
\errmessage      | 290, 362, 377, 382, 387, 407, 428, 434 |
\errorcontextlines | 332     |
\escapechar      | 142, 183, 413 |
\foo             | 358, 366, 379, 382, 443, 448, 453, 458, 463, 468, 473, 478 |
\foonewline       | 365, 366, 379 |
\hbox            | 355, 364     |
\ifdim           | 360, 385     |
\ifnum           | 273, 281, 288, 296, 375, 399 |
\ifx             | 15, 18, 21, 50, 58, 61, 113, 116, 166, 168, 186, 237, 240, 243, 246, 301, 334, 343, 379, 392, 394, 401, 421, 422 |
\immediate .................... 23, 52, 348
\input ...................... 117, 302, 335
\iterate ..................... 254, 256, 258

L
\LoadCommand .................. 302, 312
\loop ......................... 252, 268, 279, 287
\ltx@backslashchar .......... 151
\ltx@empty .................. 186, 208
\ltx@firstofone ............ 145, 160
\ltx@gobble ................ 209
\ltx@gobbletwo ............ 153, 158
\ltx@ifnextchar ........... 172, 207
\ltx@IfUndefined .......... 139, 180, 217
\ltx@undefined .......... 143

M
\meaning .................... 405, 406, 426, 427, 432, 433
\msg .................. 348, 353, 402, 404, 405, 406,
423, 425, 426, 427, 431, 432, 433

N
\newcommand ................ 171
\next ....................... 258, 260, 262
\number .................... 293

P
\PackageInfo ............... 26
\pdef ....................... 344, 346, 442,
447, 452, 457, 462, 467, 472, 477
\ProDef@AtEnd ............. 95, 96, 111, 198, 230
\ProDef@IfDefinable .... 139, 179, 203, 216
\ProDef@long ............. 208, 211, 219, 223
\ProDef@next ............ 158, 160, 162
\ProDef@ProtectedDef ....
........ 173, 175, 178, 209, 212, 215
\ProDef@temp ............ 125, 129,
130, 131, 132, 133, 134, 135,
136, 137, 138, 149, 151, 184, 186
\protect .................... 418
\protected ................ 204, 218, 222, 346
\ProtectedDef ........... 2, 171, 203, 205,
350, 352, 357, 374, 443, 448,
453, 458, 463, 468, 473, 478, 494
\ProvidesPackage .......... 19, 67

R
\RangeCatcodeCheck ......... . 285, 313, 314, 315, 316, 317,
318, 319, 320, 321, 322, 323, 324
\RangeCatcodeInvalid ...... 277, 305, 306, 307, 308
\repeat .................... 252, 264, 275, 283, 298
\RequirePackage ........ 121, 340
\RestoreCatcodes ........ 266, 269, 270, 325

S
\setbox .................... 355, 364
\space .................. 291, 292, 300, 381, 382

T
\Test .................... 304, 327
\the .................. 77, 78, 79, 80, 81, 82,
83, 84, 97, 271, 291, 292, 353, 377
\TMP@EnsureCode ........ 94, 101, 102, 103,
104, 105, 106, 107, 108, 109, 110
\TMP@RequirePackage .... 114, 121, 123, 124
\toks .................. 352, 353
\tracingmacros .......... 439

W
\wd ..................... 360, 385
\write .................. 23, 52, 348

X
\x .................. 14, 15, 18, 22, 26, 28,
51, 56, 66, 75, 87, 185, 192, 414, 416