The protecteddef package

Heiko Oberdiek∗
<heiko.oberdiek at googlemail.com>
2016/05/16 v1.1

Abstract
This package provides \ProtectedDef for defining robust macros for both plain \TeX and \LaTeX. First \TeX's \protected is tried, then \LaTeX's \DeclareRobustCommand is used. Otherwise the macro is not made robust.

Contents
1 Documentation .......................... 1
1.1 The \LaTeX's way .......................... 2
1.2 The \TeX's 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 \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
[2016/05/16 v1.1] .......................... 13
7 Index ................................... 13

1 Documentation
Many of my packages work for both formats plain \TeX and \LaTeX, even \initex is often supported. It would be nice if fragile macros could be protected and made robust. However the different format worlds offer different solutions.

∗Please report any issues at https://github.com/ho-tex/oberdiek/issues
1.1 The \LaTeX’s way

Usually `\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 `\DeclareRobustCommand`. It shares the syntax with `\newcommand`. However it does not provide latter's check.

Internally the check is available via `\@ifdefinable`.

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

1.2 The $\varepsilon$-\TeX’s way

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

1.3 The way of this package

The package tries to find the available protection mechanism. First it looks for \eTeX’s `\protected`, then it uses \LaTeX’s `\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 `\@ifdefinable` is tried to be compatible with \LaTeX. If `\@ifdefinable` is not available, then the test is implemented by asserting that the macro is undefined or has the meaning of `\relax`. If the test fails, then in all cases the macro is not defined and an error is thrown.

1.4 Usage

```
\ProtectedDef* {⟨cmd⟩} [⟨num⟩] {⟨definition text⟩}
```

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

The number of arguments `⟨num⟩` must be given as explicit digit 0 upto 9. Otherwise the part between the argument `⟨cmd⟩` and the `⟨definition text⟩` is taken as parameter text in the syntax of vanilla \TeX. Examples (with `\protected`):

```
\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}
```

2 Implementation

2.1 Reload check and package identification

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

\begingroup\catcode61\catcode48\catcode32=10\relax\% 
\catcode13=5 \% ^^M 
\endlinechar=13 \%
\catcode123=1 \% {
\catcode125=2 \% }
\catcode64=11 \% @
\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 \% #
\catcode35=11 \% @
\catcode123=1 \% {
\catcode125=2 \% }
\def\TMP@EnsureCode#1#2{%
\edef\ProDef@AtEnd{%
\ProDef@AtEnd
\catcode#1=\the\catcode#1\relax
\catcode#1=#2\relax
}%
\catcode#1=#2}\relax
\}%
\x\catcode61\catcode48\catcode32=10\relax%
\def\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]%
\edef\ProDef@AtEnd{\ProDef@AtEnd\noexpand\endinput}
125 \def\ProDef@temp#1{%  
126 \expandafter\def\csname ProDef@param[#1]\endcsname % hash-ok  
127 }  
128 \expandafter\def\csname ProDef@param\endcsname{}  
129 \ProDef@temp0{}  
130 \ProDef@temp1{##1}  
131 \ProDef@temp2{##1##2}  
132 \ProDef@temp3{##1##2##3}  
133 \ProDef@temp4{##1##2##3##4}  
134 \ProDef@temp5{##1##2##3##4##5}  
135 \ProDef@temp6{##1##2##3##4##5##6}  
136 \ProDef@temp7{##1##2##3##4##5##6##7}  
137 \ProDef@temp8{##1##2##3##4##5##6##7##8}  
138 \ProDef@temp9{##1##2##3##4##5##6##7##8##9}  
ProDef@IfDefinable  
139 \ltx@IfUndefined{@ifdefinable}{%  
140 \long\def\ProDef@IfDefinable#1{%  
141 \begingroup  
142 \escapechar=-1 %  
143 \ltx@ifundefined{\string#1}{%  
144 \endgroup  
145 \ltx@gobbletwo  
146 }{%  
147 \expandafter\endgroup  
148 \expandafter\expandafter\expandafter\endgroup  
149 \expandafter\ifx\csname protected\endcsname\relax  
150 \begingroup\expandafter\expandafter\expandafter\endgroup  
151 \expandafter\ifx\csname DeclareRobustCommand\endcsname\relax  
152 \catcode`&=14 % comment  
153 \newcommand*{\ProtectedDef}{%  
154 \let\ProDef@next\ltx@IfUndefined\string\ProDef@temp\endgroup  
155 }{%  
156 }{%  
157 \long\def\ProDef@IfDefinable#1{%  
158 \let\ProDef@next\ltx@gobbletwo  
159 \@ifdefinable{#1}{%  
160 \let\ProDef@next\ltx@gobbletwo  
161 }{%  
162 \ProDef@IfDefinable  
163 }{%  
164 }{%  
165 \begingroup\expandafter\expandafter\expandafter\endgroup  
166 \expandafter\expandafter\expandafter\endgroup  
167 \begingroup\expandafter\expandafter\expandafter\endgroup  
168 \expandafter\expandafter\expandafter\endgroup  
169 \catcode`\&=14 % comment  
170 }{%  
171 \newcommand*{\ProtectedDef}{%  
172 \ltx@ifnextchar*{%  
173 \ProDef@ProtectedDef  
174 }{%  
175 \ProDef@ProtectedDef  
176 }{%  
177 }{%  
178 \long\def\ProDef@IfDefinable#1##2##3#{%  
179 \ProDef@IfDefinable[#2]{%  
180 \ltx@ifundefined{\ProDef@param#3}{%  
181 \DeclareRobustCommand*{#2}{%  
182 \begingroup  
183 \escapechar=-1 %  
184 \def\ProDef@temp{##1}
3 Test

3.1 Catcode checks for loading

\catcode`\{=1 %
\catcode`\}=2 %
\catcode`#=6 %
\catcode`@=11 %
\expandafter\ifx\csname count@\endcsname\relax
\countdef\count@=255 %
\fi
\expandafter\ifx\csname @gobble\endcsname\relax
6
3.2 Test without \LaTeX\ and \verb\protected\n
\begin{verbatim}
\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}[2016/05/16]\%
\fi
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname protected\endcsname\relax
\let\pdef\def
\else
\def\pdef\verb\protected\verb\def\%
\fi
\def\msg#1{\immediate\write16{#1}}
\countdef\errcount=2 \%
\long\def\BeginCheck#1\ProtectedDef#2\EndCheck{%\%
\begingroup
\toks0={\ProtectedDef#2}\
\msg{<<\the\toks0>>}\
\endgroup
\setbox0=\hbox{#1\%
\ProtectedDef#2\%\check\foo}\
\ifdim\wd0=0pt\relax
\else
\errmessage{[Definition] Unwanted spaces?!}\
\fi
\def\msg#1{\immediate\write16{#1}}
\countdef\errcount=2 \%
\long\def\BeginCheck#1\ProtectedDef#2\EndCheck{%\%
\begingroup
\toks0={\ProtectedDef#2}\
\msg{<<\the\toks0>>}\
\endgroup
\setbox0=\hbox{#1\%
\ProtectedDef#2\%\check\foo}\
\ifdim\wd0=0pt\relax
\else
\errmessage{[Definition] Unwanted spaces?!}\
\fi
\end{verbatim}
\def\fooinitial{XYZ}%
\let\foo=\fooinitial
\errcount=0 %
\expandafter\def\csname PackageError\endcsname##1##2##3{%
  \advance\errcount by 1 %
}\expandafter\def\csname notdefinable\endcsname{%
  \advance\errcount by 1 %}
\ProtectedDef#2%
\ifnum\errcount=1 %
  \else
  \errmessage{1 error expected, but found: \the\errcount}%
\fi
\ifx\foo=\fooinitial
  \else
    \def\space{ }%
    \errmessage{\string\foo\space is overwritten}%
\fi
\ifdim\wd0=0pt\relax
  \else
    \errmessage{[Error] Unwanted spaces?!}%
\fi
\chardef\DeclareVersion=0 %
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname protected\endcsname\relax
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname DeclareRobustCommand\endcsname\relax
\else
\chardef\DeclareVersion=1 %
\fi
\fi
\ifnum\DeclareVersion=0 %
\def\check#1{%
  \ifx\cmp#1%
    \msg{* Test passed.}%
  \else
    \msg{}%
    \msg{\[
      \meaning#1\]}
    \msg{\[
      \meaning\cmp\]}
    \errmessage{Test failed!}%
  \fi
}\else
\def\check#1{%
  \begingroup
  \escapechar=-1 %
  \edef\x{\endgroup \def\noexpand\cs{\string#1} %
  \edef\CMP{\noexpand\protect \expandafter\noexpand\csname\cs/\endcsname %
  \ifx\CMP#1%
    \expandafter\ifx\csname\cs/\endcsname\cmp
    \msg{Test passed.}%
  \else
  \msg{Test failed!}%
}\fi
\if\exists#1%
\def\check#1{%
  \begingroup
  \edef\x{\endgroup %
  \edef\CMP{ %
  \noexpand\protect %
  \expandafter\noexpand\csname\cs/\endcsname %
  \expandafter\noexpand\csname\cs/\endcsname %
  \advance\errcount by 1 %
  \msg{Test passed.}%
  \else

\message{Test failed!}i
\else
\message{\meaning#1}
\message{\meaning\CMP}
\errmessage{Test failed!}
\fi
\fi
\tracingmacros=1
\BeginCheck
\pdef\cmp{}%
\ProtectedDef*\foo{}%
\EndCheck
\BeginCheck
\pdef\cmp{}%
\ProtectedDef*\foo[0]{}%
\EndCheck
\BeginCheck
\pdef\cmp#1{<#1>}%
\ProtectedDef*\foo[1]{<#1>}%
\EndCheck
\BeginCheck
\long\pdef\cmp{}%
\ProtectedDef\foo{}%
\EndCheck
\BeginCheck
\long\pdef\cmp{}%
\ProtectedDef\foo[0]{}%
\EndCheck
\BeginCheck
\long\pdef\cmp#1{<#1>}%
\ProtectedDef\foo[1]{<#1>}%
\EndCheck
\BeginCheck
\long\pdef\cmp(#1){<#1>}%
\ProtectedDef\foo(#1){<#1>}%
\EndCheck
\csname @@end\endcsname\end⟨/test2⟩
4 Installation

4.1 Download

Package. This package is available on CTAN¹:


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.

4.2 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/

4.3 Package installation

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

tex protecteddef.dtx

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

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

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.

4.4 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 mktexlar.

¹http://ctan.org/pkg/protecteddef
4.5 Some details for the interested

Unpacking with \LaTeX. The \texttt{.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:

\texttt{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 \texttt{.dtx} or the \texttt{.drv} to generate the documentation. The process can be configured by the configuration file \texttt{ltxdoc.cfg}. For instance, put this line into this file, if you want to have A4 as paper format:

\texttt{\PassOptionsToClass{a4paper}{article}}

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

\begin{verbatim}
\input{protecteddef.dtx}
\makeindex -s gind.ist protecteddef.idx
\makeindex -s gind.ist protecteddef.idx
\makeindex -s gind.ist protecteddef.idx
\end{verbatim}

5 Catalogue

The following XML file can be used as source for the \TeX Catalogue. The elements \texttt{caption} and \texttt{description} are imported from the original XML file from the Catalogue. The name of the XML file in the Catalogue is \texttt{protecteddef.xml}.

\begin{verbatim}
<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='1.1'/>
<description>

The package defines a command \texttt{\ProtectedDef} that will create LaTeX 'robust' command or an e-\TeX 'protected' command as appropriate to its environment.

The package is part of the \texttt{xref refid='oberdiek'>oberdiek</xref> bundle.

</description>
<documentation details='Package documentation'

href='ctan:/macros/latex/contrib/oberdiek/protecteddef.pdf'/>
<ctan file='true' path='ctan:/macros/latex/contrib/oberdiek/protecteddef.dtx'/>
<miktex location='oberdiek'/>
<texlive location='oberdiek'/>
<install path='ctan:/macros/latex/contrib/oberdiek/oberdiek.tds.zip'/>

</entry>
</catalogue>
\end{verbatim}
6 History

[2011/01/31 v1.0]  
• First public version.

[2016/05/16 v1.1]  
• Documentation updates.

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
\@ehc ............................... 152
\@firstofone ....................... 244, 247
\@gobble ............................ 241, 249
\@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, 109, 201, 233, 234, 235, 236, 271, 280, 288, 292, 309, 310, 311, 336, 337, 338
\chardef ........................ 390, 396
\check .................. 358, 400, 411
\ChP .................. 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

D
\DeclareRobustCommand .... 181, 194
\DeclareVersion ........... 390, 396, 399

E
\empty .................. 17, 18
\end .................. 329, 481
\EndCheck .................. 350, 444, 449, 454, 459, 464, 469, 474, 479
\endsname .................. 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
\endinput .................. 29, 111
\endlinechar ............ 4, 35, 71, 77, 89
\errcount .................. 349, 367, 369, 372, 375, 377
\errmessage .................. 290, 362, 377, 382, 387, 407, 428, 434
\errorcontextlines ........... 332
\escapechar .............. 142, 183, 413

F
\foo .................. 358, 366, 379, 382, 443, 448, 453, 458, 463, 468, 473, 478
\fooinitial ............... 365, 366, 379

H
\bbox .................. 355, 364

I
\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