The \texttt{protecteddef} package

Heiko Oberdiek\textsuperscript{*}

2016/05/16 v1.1

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

Contents

1 Documentation 1
  1.1 The \LaTeX{}'s way ........................................ 2
  1.2 The \eTeX{}'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 Installation 7
  3.1 Download .................................................. 7
  3.2 Bundle installation ....................................... 7
  3.3 Package installation ...................................... 7
  3.4 Refresh file name databases ............................... 7
  3.5 Some details for the interested .......................... 8

4 History 8
  [2011/01/31 v1.0] ............................................. 8
  [2016/05/16 v1.1] ............................................. 8

5 Index 8

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.

\textsuperscript{*}Please report any issues at \url{https://github.com/ho-tex/oberdiek/issues}
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 ε-\TeX{}'s way

The need for robust macros is addressed in \texttt{\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 \texttt{\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 \E\TeX{}'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 ⟨cmd⟩ is to be defined as \texttt{\long} macro without star with ⟨num⟩ arguments.

The number of arguments ⟨num⟩ must be given as expicite 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 \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.
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%
\edef\ProDef@AtEnd\noexpand\endinput

2.3 Resources
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname DeclareRobustCommand\endcsname\relax
\catcode'\&=14 \% comment
\else
\newcommand*{\ProtectedDef}{%
  \ltx@ifnextchar*{%
    \ProDef@ProtectedDef
  }{%}
  \ProDef@ProtectedDef%
}%
\long\def\ProDef@ProtectedDef#1#2#3#{%
  \ltx@ifUndefined{ProDef@param#3}{%
    \DeclareRobustCommand*{#2}{}%
    \begingroup
    \escapechar=-1 \%
    \def\ProDef@temp{#1}%
    \edef\x{\endgroup
    \ifx\ProDef@temp\ltx@empty
      \noexpand\long
      \fi
      \noexpand\def
      \expandafter\expandafter\expandafter\x#3%
    }%
    \else
    \catcode'\&=9 \% ignore
    \fi%
  }%
  \DeclareRobustCommand*{#2}{%}
  \long\def\ProDef@ProtectedDef#1#2#3#{%
    \ProDef@IfDefinable{#1}{%
      \let\ProDef@long\ltx@empty
      \expandafter\expandafter\expandafter\def
      \expandafter\expandafter\expandafter\ProDef@long\ltx@empty
      \expandafter\expandafter\expandafter\ProDef@ProtectedDef\ltx@gobble
    }{%}
    \let\ProDef@long\long
    \ProDef@ProtectedDef
  }%
}\long\def\ProDef@ProtectedDef#1#2#3#{%
  \ProDef@IfDefinable{#1}{%
    \ProDef@ProtectedDef
  }{%}
  \ProDef@ProtectedDef
}\long\def\ProDef@ProtectedDef#1#2#3#{%
  \ProDef@IfDefinable{#1}{%
    \ProDef@ProtectedDef
  }{%}
  \ProDef@ProtectedDef
}\long\def\ProDef@ProtectedDef#1#2#3#{%
  \ProDef@IfDefinable{#1}{%
    \ProDef@ProtectedDef
  }{%}
  \ProDef@ProtectedDef
}\long\def\ProDef@ProtectedDef#1#2#3#{%
  \ProDef@IfDefinable{#1}{%
    \ProDef@ProtectedDef
  }{%}
  \ProDef@ProtectedDef
}\endgroup
3 Installation

3.1 Download

Package. This package is available on CTAN:\footnote{CTAN:pkg/protecteddef}

\begin{itemize}
  \item \texttt{CTAN:macros/latex/contrib/oberdiek/protecteddef.dtx} The source file.
  \item \texttt{CTAN:macros/latex/contrib/oberdiek/protecteddef.pdf} Documentation.
\end{itemize}

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.

\texttt{CTAN:install/macros/latex/contrib/oberdiek.tds.zip}

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

3.2 Bundle installation

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

```
unzip oberdiek.tds.zip -d ~/texmf
```

3.3 Package installation

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

```
protecteddef.sty → tex/generic/oberdiek/protecteddef.sty
protecteddef.pdf → doc/latex/oberdiek/protecteddef.pdf
protecteddef.dtx → source/latex/oberdiek/protecteddef.dtx
```

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}.

3.4 Refresh file name databases

If your \TeX\ distribution (\TeX\ Live, MiK\TeX, \ldots) relies on file name databases, you must refresh these. For example, \TeX\ Live users run \texttt{texhash} or \texttt{mktexlsr}.

\footnote{CTAN:pkg/protecteddef}
3.5 Some details for the interested

Unpacking with \LaTeX. The .dtx chooses its action depending on the format:

plain \TeX: Run \texttt{docstrip} and extract the files.

\LaTeX: Generate the documentation.

If you insist on using \LaTeX for \texttt{docstrip} (really, \texttt{docstrip} does not need \LaTeX), then inform the autodetect routine about your intention:

\begin{verbatim}
 latex \let\install=y\input{protecteddef.dtx}
\end{verbatim}

Do not forget to quote the argument according to the demands of your shell.

\textbf{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 \texttt{ltxdoc.cfg}. For instance, put this line into this file, if you want to have A4 as paper format:

\begin{verbatim}
\PassOptionsToClass{a4paper}{article}
\end{verbatim}

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

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

4 History

[2011/01/31 v1.0]

\begin{itemize}
\item First public version.
\end{itemize}

[2016/05/16 v1.1]

\begin{itemize}
\item Documentation updates.
\end{itemize}

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.

\begin{center}
\begin{tabular}{ll}
\textbf{Symbols} & 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 \\
\texttt{\&} & 169, 201 \\
\texttt{\@PackageError} & 150, 152, 74, 78, 79, 80, 81, 82, 83, 84, 87, 88, 90, 91, 92, 93, 97, 99, 169, 201 \\
\texttt{\@ehc} & 159, 88, 90, 91, 92, 93, 97, 99, 169, 201 \\
\texttt{\@ifdefinable} & 58, 113, 116, 126, 128, 166, 168, 190, 226 \\
\texttt{\@undefined} & \texttt{\csname} \texttt{\endcsname} \\
\texttt{\aftergroup} & 29 \\
\texttt{\catcode} & 2, 3, 5, 6 \\
\texttt{\DeclareRobustCommand} & 181, 194
\end{tabular}
\end{center}
E
\empty \hfill 17, 18
endcsname \hfill 14, 21, 50, 66, 76, 113,
116, 126, 128, 166, 168, 190, 226
endinput \hfill 29, 111
endlinechar \hfill 4, 35, 71, 77, 89
escapechar \hfill 142, 183
\endcsname . \hfill 14, 21, 50, 66, 76, 113,
116, 126, 128, 166, 168, 190, 226
\endinput . \hfill 29, 111
\endlinechar . \hfill 4, 35, 71, 77, 89
\escapechar . \hfill 142, 183
\ifx \hfill 15, 18, 21,
50, 58, 61, 113, 116, 166, 168, 186
\immediate \hfill 23, 52
\input \hfill 117
\ltx@backslashchar \hfill 151
\ltx@empty \hfill 186, 208
\ltx@firstofone \hfill 145, 160
\ltx@gobble \hfill 209
\ltx@gobbletwo \hfill 153, 158
\ltx@ifnextchar \hfill 172, 207
\ltx@ifundefined \hfill 139, 180, 217
\ltx@ifundefined \hfill 143
\newcommand \hfill 171
\PackageInfo \hfill 26
\RequirePackage \hfill 121
\the \hfill 77, 78, 79, 80, 81, 82, 83, 84, 97
\TMP@EnsureCode . \hfill 94, 101, 102, 103,
104, 105, 106, 107, 108, 109, 110
\TMP@RequirePackage \hfill 114, 121, 123, 124
\write \hfill 23, 52
x \hfill 14, 15, 18, 22
\x \hfill 14, 15, 18, 22
\PackageInfo \hfill 26, 28, 51, 56, 66, 75, 87, 185, 192