The **protecteddef** package

Heiko Oberdiek[*]

2016/05/16 v1.1

Abstract

This packages provides `\ProtectedDef` for defining robust macros for both plain TeX and LaTeX. First ε-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 . . . 1
1.2 The ε-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 . . . 3
2.3 Resources . . . 4

3 Installation **6**
3.1 Download . . . 6
3.2 Bundle installation . . . 6
3.3 Package installation . . . 7
3.4 Refresh file name databases . . . 7
3.5 Some details for the interested . . . 7

4 History **7**
[2011/01/31 v1.0] . . . 7
[2016/05/16 v1.1] . . . 7

5 Index **8**

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 latters 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 ε-TeX's way

The need for robust macros is addressed in `\eTeX`. 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 explicite 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

```
⟨*package⟩
```

2.1 Reload check and package identification

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

```
\begingroup\catcode61\catcode48\catcode32=10\relax%
  \catcode13=5 % ^^M
  \endlinechar=13 %
  \catcode35=6 % #
  \catcode39=12 % '
  \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
    \def\empty{}%
    \ifx\x\empty % LaTeX, first loading,
      % variable is initialized, but \ProvidesPackage not yet seen
    \else
      \expandafter\ifx\csname PackageInfo\endcsname\relax
        \def\x#1#2{%
          \immediate\write-1{Package #1 Info: #2.}%
        }%
      \else
        \def\x#1#2{\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#1\@undefined
        \xdef#1{#3}%
      \fi
      \ifx#1\relax
        \xdef#1{#3}%
      \fi
    }%
  \fi
\expandafter\x\csname ver@protecteddef.sty\endcsname
\ProvidesPackage{protecteddef}%
```

```
  [2016/05/16 v1.1 Define protected commands (HO)]%
```

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 %
\catcode35=6 % #
\catcode64=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
}
\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
}
\expandafter\def\csname ProDef@param\endcsname{}
\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##7}
\ProDef@temp8{##1##2##3##4##5##7##8}
\ProDef@temp9{##1##2##3##4##5##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
      \edef\expandafter\ProDef@temp\expandafter{\string#1 }%
      \@PackageError{protecteddef}{%
        Command \ltx@backslashchar\ProDef@temp already defined%
      }\@ehc
      \ltx@gobbletwo
    }%
  }%
}{%
  \long\def\ProDef@IfDefinable#1{%
    \let\ProDef@next\ltx@gobbletwo
    \@ifdefinable{#1}{%
      \let\ProDef@next\ltx@firstofone
    }%
    \ProDef@next
  }%
}
```

```
\begingroup\expandafter\expandafter\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{}%
      }%
    }%
    \long\def\ProDef@ProtectedDef#1#2#3#{%
      \ProDef@IfDefinable{#2}{%
        \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\noexpand\csname\string#2 \endcsname
          }%
          \x#3%
        }{%
          \DeclareRobustCommand#1{#2}#3%
        }%
      }%
    }%
    \expandafter\expandafter\expandafter\ProDef@AtEnd
  \fi
\else
  \catcode`\&=9 % ignore
\fi%
\ProDef@IfDefinable\ProtectedDef{%
&  \protected
  \def\ProtectedDef%
}{%
  \ltx@ifnextchar*{%
    \let\ProDef@long\ltx@empty
    \expandafter\ProDef@ProtectedDef\ltx@gobble
  }{%
    \let\ProDef@long\long
    \ProDef@ProtectedDef
  }%
}
\long\def\ProDef@ProtectedDef#1#2#{%
  \ProDef@IfDefinable{#1}{%
    \ltx@IfUndefined{ProDef@param#2}{%
&     \protected
      \ProDef@long
      \def#1#2%
    }{%
&     \protected
      \ProDef@long
      \expandafter\expandafter\expandafter\def
      \expandafter\expandafter\expandafter#1%
      \csname ProDef@param#2\endcsname
    }%
  }%
}

\ProDef@AtEnd%

⟨/package⟩
```

3 Installation

3.1 Download

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

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

CTAN:macros/latex/contrib/oberdiek/protecteddef.pdf Documentation.

---

[1] CTAN:pkg/protecteddef

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 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
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.

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{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
```

4 History

[2011/01/31 v1.0]

- First public version.

[2016/05/16 v1.1]

- 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.

Symbols

\& 169, 201
\@PackageError 150
\@ehc 152
\@ifdefinable 159
\@undefined 58

A

\aftergroup 29

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
\csname 14, 21, 50, 66, 76, 113, 116, 126, 128, 166, 168, 190, 226

D

\DeclareRobustCommand 181, 194

E

\empty 17, 18
\endcsname . 14, 21, 50, 66, 76, 113, 116, 126, 128, 166, 168, 190, 226
\endinput 29, 111
\endlinechar 4, 35, 71, 77, 89
\escapechar 142, 183

I

\ifx 15, 18, 21, 50, 58, 61, 113, 116, 166, 168, 186
\immediate 23, 52
\input 117

L

\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@ifundefined 143

N

\newcommand 171

P

\PackageInfo 26
\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
\protected 204, 218, 222
\ProtectedDef *2*, 171, 203, 205
\ProvidesPackage 19, 67

R

\RequirePackage 121

T

\the . . . 77, 78, 79, 80, 81, 82, 83, 84, 97
\TMP@EnsureCode . 94, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110
\TMP@RequirePackage 114, 121, 123, 124

W

\write 23, 52

X

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