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

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

1 ⟨∗package⟩

2.1 Reload check and package identification

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

```
 2 \begingroup\catcode61\catcode48\catcode32=10\relax%
 3   \catcode13=5 % ^^M
 4   \endlinechar=13 %
 5   \catcode35=6 % #
 6   \catcode39=12 % '
 7   \catcode44=12 % ,
 8   \catcode45=12 % -
 9   \catcode46=12 % .
10   \catcode58=12 % :
11   \catcode64=11 % @
12   \catcode123=1 % {
13   \catcode125=2 % }
14   \expandafter\let\expandafter\x\csname ver@protecteddef.sty\endcsname
15   \ifx\x\relax % plain-TeX, first loading
16   \else
17     \def\empty{}%
18     \ifx\x\empty % LaTeX, first loading,
19       % variable is initialized, but \ProvidesPackage not yet seen
20     \else
21       \expandafter\ifx\csname PackageInfo\endcsname\relax
22         \def\x#1#2{%
23           \immediate\write-1{Package #1 Info: #2.}%
24         }%
25       \else
26         \def\x#1#2{\PackageInfo{#1}{#2, stopped}}%
27       \fi
28       \x{protecteddef}{The package is already loaded}%
29       \aftergroup\endinput
30     \fi
31   \fi
32 \endgroup%
```

Package identification:

```
33 \begingroup\catcode61\catcode48\catcode32=10\relax%
34   \catcode13=5 % ^^M
35   \endlinechar=13 %
36   \catcode35=6 % #
37   \catcode39=12 % '
38   \catcode40=12 % (
39   \catcode41=12 % )
40   \catcode44=12 % ,
41   \catcode45=12 % -
42   \catcode46=12 % .
43   \catcode47=12 % /
44   \catcode58=12 % :
45   \catcode64=11 % @
46   \catcode91=12 % [
47   \catcode93=12 % ]
48   \catcode123=1 % {
49   \catcode125=2 % }
50   \expandafter\ifx\csname ProvidesPackage\endcsname\relax
51     \def\x#1#2#3[#4]{\endgroup
52       \immediate\write-1{Package: #3 #4}%
53       \xdef#1{#4}%
54     }%
55   \else
56     \def\x#1#2[#3]{\endgroup
57       #2[{#3}]%
58       \ifx#1\@undefined
```

```
59        \xdef#1{#3}%
60      \fi
61      \ifx#1\relax
62        \xdef#1{#3}%
63      \fi
64    }%
65  \fi
66 \expandafter\x\csname ver@protecteddef.sty\endcsname
67 \ProvidesPackage{protecteddef}%
68   [2016/05/16 v1.1 Define protected commands (HO)]%
```

2.2   Catcodes

```
69 \begingroup\catcode61\catcode48\catcode32=10\relax%
70   \catcode13=5 % ^^M
71   \endlinechar=13 %
72   \catcode123=1 % {
73   \catcode125=2 % }
74   \catcode64=11 % @
75   \def\x{\endgroup
76     \expandafter\edef\csname ProDef@AtEnd\endcsname{%
77       \endlinechar=\the\endlinechar\relax
78       \catcode13=\the\catcode13\relax
79       \catcode32=\the\catcode32\relax
80       \catcode35=\the\catcode35\relax
81       \catcode61=\the\catcode61\relax
82       \catcode64=\the\catcode64\relax
83       \catcode123=\the\catcode123\relax
84       \catcode125=\the\catcode125\relax
85     }%
86   }%
87 \x\catcode61\catcode48\catcode32=10\relax%
88 \catcode13=5 % ^^M
89 \endlinechar=13 %
90 \catcode35=6 % #
91 \catcode64=11 % @
92 \catcode123=1 % {
93 \catcode125=2 % }
94 \def\TMP@EnsureCode#1#2{%
95   \edef\ProDef@AtEnd{%
96     \ProDef@AtEnd
97     \catcode#1=\the\catcode#1\relax
98   }%
99   \catcode#1=#2\relax
100 }
101 \TMP@EnsureCode{38}{4}% &
102 \TMP@EnsureCode{40}{12}% (
103 \TMP@EnsureCode{41}{12}% )
104 \TMP@EnsureCode{42}{12}% *
105 \TMP@EnsureCode{45}{12}% -
106 \TMP@EnsureCode{46}{12}% .
107 \TMP@EnsureCode{47}{12}% /
108 \TMP@EnsureCode{91}{12}% [
109 \TMP@EnsureCode{93}{12}% ]
110 \TMP@EnsureCode{96}{12}% `
111 \edef\ProDef@AtEnd{\ProDef@AtEnd\noexpand\endinput}
```

2.3   Resources

```
112 \begingroup\expandafter\expandafter\expandafter\endgroup
113 \expandafter\ifx\csname RequirePackage\endcsname\relax
114   \def\TMP@RequirePackage#1[#2]{%
115     \begingroup\expandafter\expandafter\expandafter\endgroup
116     \expandafter\ifx\csname ver@#1.sty\endcsname\relax
117       \input #1.sty\relax
118     \fi
119   }%
120 \else
121   \let\TMP@RequirePackage\RequirePackage
122 \fi
123 \TMP@RequirePackage{ltxcmds}[2010/12/12]%
124 \TMP@RequirePackage{infwarerr}[2010/04/08]%

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##7}
137 \ProDef@temp8{##1##2##3##4##5##7##8}
138 \ProDef@temp9{##1##2##3##4##5##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@firstofone
146     }{%
147       \expandafter\endgroup
148       \expandafter
149       \edef\expandafter\ProDef@temp\expandafter{\string#1 }%
150       \@PackageError{protecteddef}{%
151         Command \ltx@backslashchar\ProDef@temp already defined%
152       }\@ehc
153       \ltx@gobbletwo
154     }%
155   }%
156 }{%
157   \long\def\ProDef@IfDefinable#1{%
158     \let\ProDef@next\ltx@gobbletwo
159     \@ifdefinable{#1}{%
160       \let\ProDef@next\ltx@firstofone
161     }%
162     \ProDef@next
163   }%
164 }

165 \begingroup\expandafter\expandafter\expandafter\endgroup
166 \expandafter\ifx\csname protected\endcsname\relax
```

5

```
167   \begingroup\expandafter\expandafter\expandafter\endgroup
168   \expandafter\ifx\csname DeclareRobustCommand\endcsname\relax
169     \catcode`\&=14 % comment
170   \else
171     \newcommand*{\ProtectedDef}{%
172       \ltx@ifnextchar*{%
173         \ProDef@ProtectedDef
174       }{%
175         \ProDef@ProtectedDef{}%
176       }%
177     }%
178     \long\def\ProDef@ProtectedDef#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}%
185           \edef\x{\endgroup
186             \ifx\ProDef@temp\ltx@empty
187               \noexpand\long
188             \fi
189             \noexpand\def
190             \expandafter\noexpand\csname\string#2 \endcsname
191           }%
192           \x#3%
193         }{%
194           \DeclareRobustCommand#1{#2}#3%
195         }%
196       }%
197     }%
198     \expandafter\expandafter\expandafter\ProDef@AtEnd
199   \fi
200 \else
201   \catcode`\&=9 % ignore
202 \fi%
203 \ProDef@IfDefinable\ProtectedDef{%
204 &  \protected
205   \def\ProtectedDef%
206 }{%
207   \ltx@ifnextchar*{%
208     \let\ProDef@long\ltx@empty
209     \expandafter\ProDef@ProtectedDef\ltx@gobble
210   }{%
211     \let\ProDef@long\long
212     \ProDef@ProtectedDef
213   }%
214 }
215 \long\def\ProDef@ProtectedDef#1#2#{%
216   \ProDef@IfDefinable{#1}{%
217     \ltx@IfUndefined{ProDef@param#2}{%
218 &      \protected
219       \ProDef@long
220       \def#1#2%
221     }{%
222 &      \protected
223       \ProDef@long
224       \expandafter\expandafter\expandafter\def
```

```
225        \expandafter\expandafter\expandafter#1%
226        \csname ProDef@param#2\endcsname
227      }%
228    }%
229 }
230 \ProDef@AtEnd%
231 ⟨/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.

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`.

---

[1][CTAN:pkg/protecteddef](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 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.