

# The `protecteddef` package

Heiko Oberdiek\*

2016/05/16 v1.1

## Abstract

This package provides `\ProtectedDef` for defining robust macros for both plain  $\text{\TeX}$  and  $\text{\LaTeX}$ . First  $\varepsilon\text{-}\text{\TeX}$ 's `\protected` is tried, then  $\text{\LaTeX}$ 's `\DeclareRobustCommand` is used. Otherwise the macro is not made robust.

## Contents

<b>1</b>	<b>Documentation</b>	<b>1</b>
1.1	The $\text{\LaTeX}$ 's way	1
1.2	The $\varepsilon\text{-}\text{\TeX}$ 's way	2
1.3	The way of this package	2
1.4	Usage	2
<b>2</b>	<b>Implementation</b>	<b>2</b>
2.1	Reload check and package identification	2
2.2	Catcodes	3
2.3	Resources	4
<b>3</b>	<b>Installation</b>	<b>6</b>
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
<b>4</b>	<b>History</b>	<b>7</b>
	[2011/01/31 v1.0]	7
	[2016/05/16 v1.1]	7
<b>5</b>	<b>Index</b>	<b>8</b>

## 1 Documentation

Many of my packages work for both formats plain  $\text{\TeX}$  and  $\text{\LaTeX}$ , even  $\text{\initEX}$  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 L<sup>A</sup>T<sub>E</sub>X'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 L<sup>A</sup>T<sub>E</sub>X 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-T<sub>E</sub>X'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 L<sup>A</sup>T<sub>E</sub>X'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 L<sup>A</sup>T<sub>E</sub>X's `\@ifdefinable` is tried to be compatible with L<sup>A</sup>T<sub>E</sub>X. 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 L<sup>A</sup>T<sub>E</sub>X'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 T<sub>E</sub>X. 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 L<sup>A</sup>T<sub>E</sub>X.

```
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 (H0)]%

## 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
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         }%
183         \begingroup
184         \escapechar=-1 %
185         \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<sup>1</sup>:

[CTAN:macros/latex/contrib/oberdiek/protecteddef.dtx](https://ctan.org/ctan/packages/macros/latex/contrib/oberdiek/protecteddef.dtx) The source file.

[CTAN:macros/latex/contrib/oberdiek/protecteddef.pdf](https://ctan.org/ctan/packages/macros/latex/contrib/oberdiek/protecteddef.pdf) Documentation.

---

<sup>1</sup>[CTAN:pkg/protecteddef](https://ctan.org/ctan/packages/macros/latex/contrib/oberdiek/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](http://CTAN:install/macros/latex/contrib/oberdiek.tds.zip)

*TDS* refers to the standard “A Directory Structure for T<sub>E</sub>X Files” ([CTAN:pkg/tds](http://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 T<sub>E</sub>X:

```
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 T<sub>E</sub>X distribution (T<sub>E</sub>X Live, MiK<sub>T</sub><sub>E</sub>X, ...) relies on file name databases, you must refresh these. For example, T<sub>E</sub>X Live users run `texhash` or `mktextlsr`.

### 3.5 Some details for the interested

**Unpacking with L<sup>A</sup>T<sub>E</sub>X.** The `.dtx` chooses its action depending on the format:

**plain T<sub>E</sub>X:** Run `docstrip` and extract the files.

**L<sup>A</sup>T<sub>E</sub>X:** Generate the documentation.

If you insist on using L<sup>A</sup>T<sub>E</sub>X for `docstrip` (really, `docstrip` does not need L<sup>A</sup>T<sub>E</sub>X), 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 pdfL<sup>A</sup>T<sub>E</sub>X:

```
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	
<code>\&amp;</code> .....	169, 201
<code>\@PackageError</code> .....	150
<code>\@ehc</code> .....	152
<code>\@ifdefinable</code> .....	159
<code>\@undefined</code> .....	58
<code>\ltx@firstofone</code> .....	145, 160
<code>\ltx@gobble</code> .....	209
<code>\ltx@gobbletwo</code> .....	153, 158
<code>\ltx@ifnextchar</code> .....	172, 207
<code>\ltx@ifundefined</code> .....	139, 180, 217
<code>\ltx@ifundefined</code> .....	143
A	
<code>\aftergroup</code> .....	29
C	
<code>\catcode</code> .....	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
<code>\csname</code> .....	14, 21, 50, 66, 76, 113, 116, 126, 128, 166, 168, 190, 226
D	
<code>\DeclareRobustCommand</code> .....	181, 194
E	
<code>\empty</code> .....	17, 18
<code>\endcsname</code> .....	14, 21, 50, 66, 76, 113, 116, 126, 128, 166, 168, 190, 226
<code>\endinput</code> .....	29, 111
<code>\endlinechar</code> .....	4, 35, 71, 77, 89
<code>\escapechar</code> .....	142, 183
I	
<code>\ifx</code> .....	15, 18, 21, 50, 58, 61, 113, 116, 166, 168, 186
<code>\immediate</code> .....	23, 52
<code>\input</code> .....	117
L	
<code>\ltx@backslashchar</code> .....	151
<code>\ltx@empty</code> .....	186, 208
N	
<code>\newcommand</code> .....	171
P	
<code>\PackageInfo</code> .....	26
<code>\ProDef@AtEnd</code> .....	95, 96, 111, 198, 230
<code>\ProDef@ifDefinable</code> .....	139, 179, 203, 216
<code>\ProDef@long</code> .....	208, 211, 219, 223
<code>\ProDef@next</code> .....	158, 160, 162
<code>\ProDef@ProtectedDef</code> .....	173, 175, 178, 209, 212, 215
<code>\ProDef@temp</code> .....	125, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 149, 151, 184, 186
<code>\protected</code> .....	204, 218, 222
<code>\ProtectedDef</code> .....	2, 171, 203, 205
<code>\ProvidesPackage</code> .....	19, 67
R	
<code>\RequirePackage</code> .....	121
T	
<code>\the</code> .....	77, 78, 79, 80, 81, 82, 83, 84, 97
<code>\TMP@EnsureCode</code> .....	94, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110
<code>\TMP@RequirePackage</code> .....	114, 121, 123, 124
W	
<code>\write</code> .....	23, 52
X	
<code>\x</code> .....	14, 15, 18, 22, 26, 28, 51, 56, 66, 75, 87, 185, 192