

The letltxmacro package

Heiko Oberdiek
<oberdiek@uni-freiburg.de>

2008/06/24 v1.3

Abstract

\TeX 's `\let` assignment does not work for \LaTeX macros with optional arguments or for macros that are defined as robust macros by `\DeclareRobustCommand`. This package defines `\LetLtxMacro` that also takes care of the involved internal macros.

Contents

| | | |
|----------|---|-----------|
| 1 | Documentation | 1 |
| 1.1 | Supported macro definition commands | 2 |
| 2 | Implementation | 2 |
| 2.1 | Show cases | 2 |
| 2.1.1 | letltxmacro-showcases.tex | 2 |
| 2.1.2 | Result | 3 |
| 2.2 | Package | 4 |
| 2.2.1 | Catcodes and identification | 4 |
| 2.2.2 | Main macros | 5 |
| 3 | Test | 7 |
| 3.1 | Catcode checks for loading | 7 |
| 3.2 | Package tests | 8 |
| 4 | Installation | 11 |
| 4.1 | Download | 11 |
| 4.2 | Bundle installation | 11 |
| 4.3 | Package installation | 11 |
| 4.4 | Refresh file name databases | 12 |
| 4.5 | Some details for the interested | 12 |
| 5 | History | 12 |
| | [2008/06/09 v1.0] | 12 |
| | [2008/06/12 v1.1] | 13 |
| | [2008/06/13 v1.2] | 13 |
| | [2008/06/24 v1.3] | 13 |
| 6 | Index | 13 |

1 Documentation

If someone wants to redefine a macro with using the old meaning, then one method is \TeX 's command `\let`:

```

\newcommand{\Macro}{\typeout{Test Macro}}
\let\SavedMacro=\Macro
\renewcommand{\Macro}{%
  \typeout{Begin}%
  \SavedMacro
  \typeout{End}%
}

```

However, this method fails, if `\Macro` is defined by `\DeclareRobustCommand` and/or has an optional argument. In both cases L^AT_EX defines an additional internal macro that is forgotten in the simple `\let` assignment of the example above.

`\LetLtxMacro {⟨new macro⟩} {⟨old macro⟩}`

Macro `\LetLtxMacro` behaves similar to T_EX's `\let` assignment, but it takes care of macros that are defined by `\DeclareRobustCommand` and/or have optional arguments. Example:

```

\DeclareRobustCommand{\Macro}[1][default]{...}
\LetLtxMacro{\SavedMacro}{\Macro}

```

Then macro `\SavedMacro` only uses internal macro names that are derived from `\SavedMacro`'s macro name. Macro `\Macro` can now be redefined without affecting `\SavedMacro`.

1.1 Supported macro definition commands

| | |
|---|--------------------------|
| <code>\newcommand</code> , <code>\renewcommand</code> | latex/base |
| <code>\newenvironment</code> , <code>\renewenvironment</code> | latex/base |
| <code>\DeclareRobustCommand</code> | latex/base |
| <code>\newrobustcmd</code> , <code>\renewrobustcmd</code> | etoolbox |
| <code>\robustify</code> | etoolbox 2008/06/22 v1.6 |

2 Implementation

2.1 Show cases

2.1.1 letltxmacro-showcases.tex

```

1 (*showcases)
2 \NeedsTeXFormat{LaTeX2e}
3 \makeatletter

```

`\Line` The result is displayed by macro `\Line`. The percent symbol at line start allows easy grepping and inserting into the DTX file.

```

4 \newcommand*{\Line}[1]{%
5   \typeout{\@percentchar#1}%
6 }

7 \newcommand*{\ShowCmdName}[1]{%
8   \@ifundefined{#1}{}{%
9     \Line{%
10       \space\space(\expandafter\string\csname#1\endcsname) = %
11       (\expandafter\meaning\csname#1\endcsname)%
12     }%
13   }%
14 }

15 \newcommand*{\ShowCmds}[1]{%
16   \ShowCmdName{#1}%
17   \ShowCmdName{#1 }%
18   \ShowCmdName{\#1}%

```

```

19 \ShowCmdName{\#1 }%
20 }
21 \let\\ \@backslashchar

\ShowDef

22 \newcommand*{\ShowDef}[2]{%
23   \begingroup
24   \Line{}%
25   \newcommand*{\DefString}{#2}%
26   \@onelevel@sanitize\DefString
27   \Line{\DefString}%
28   #2%
29   \ShowCmds{#1}%
30   \endgroup
31 }

32 \typeout{}
33 \Line{* LaTeX definitions:}
34 \ShowDef{cmd}{%
35   \newcommand{\cmd}[2][default]{}%
36 }
37 \ShowDef{cmd}{%
38   \DeclareRobustCommand{\cmd}{}%
39 }
40 \ShowDef{cmd}{%
41   \DeclareRobustCommand{\cmd}[2][default]{}%
42 }
43 \typeout{}

The minimal version of package etoolbox is 2008/06/12 v1.6a because it fixes
\robustify.

44 \RequirePackage{etoolbox}[2008/06/12]%
45 \Line{}
46 \Line{* etoolbox's robust definitions:}
47 \ShowDef{cmd}{%
48   \newrobustcmd{\cmd}{}%
49 }
50 \ShowDef{cmd}{%
51   \newrobustcmd{\cmd}[2][default]{}%
52 }
53 \Line{}
54 \Line{* etoolbox's \string\robustify:}
55 \ShowDef{cmd}{%
56   \newcommand{\cmd}[2][default]{ } %
57   \robustify{\cmd}%
58 }
59 \ShowDef{cmd}{%
60   \DeclareRobustCommand{\cmd}{ } %
61   \robustify{\cmd}%
62 }
63 \ShowDef{cmd}{%
64   \DeclareRobustCommand{\cmd}[2][default]{ } %
65   \robustify{\cmd}%
66 }
67 \typeout{}
68 \@@end
69 \showcases

```

2.1.2 Result

* LaTeX definitions:

```
\newcommand {\cmd }[2][default]{}
```

```

(\cmd) = (macro:->\@protected@testopt \cmd \cmd {default})
(\cmd) = (\long macro:[#1]#2->)

\DeclareRobustCommand {\cmd }{}
(\cmd) = (macro:->\protect \cmd_ )
(\cmd_) = (\long macro:->)

\DeclareRobustCommand {\cmd }[2][default]{}
(\cmd) = (macro:->\protect \cmd_ )
(\cmd_) = (macro:->\@protected@testopt \cmd_ \cmd_ {default})
(\cmd_) = (\long macro:[#1]#2->)

* etoolbox's robust definitions:

\newrobustcmd {\cmd }{}
(\cmd) = (\protected\long macro:->)

\newrobustcmd {\cmd }[2][default]{}
(\cmd) = (\protected macro:->\@testopt \cmd {default})
(\cmd) = (\long macro:[#1]#2->)

* etoolbox's \robustify:

\newcommand {\cmd }[2][default]{} \robustify {\cmd }
(\cmd) = (\protected macro:->\@protected@testopt \cmd \cmd {default})
(\cmd) = (\long macro:[#1]#2->)

\DeclareRobustCommand {\cmd }{} \robustify {\cmd }
(\cmd) = (\protected macro:->)

\DeclareRobustCommand {\cmd }[2][default]{} \robustify {\cmd }
(\cmd) = (\protected macro:->\@protected@testopt \cmd_ \cmd_ {default})
(\cmd_) = (macro:->\@protected@testopt \cmd_ \cmd_ {default})
(\cmd_) = (\long macro:[#1]#2->)

```

2.2 Package

70 `\package`

2.2.1 Catcodes and identification

```

71 \begingroup
72 \catcode123 1 % {
73 \catcode125 2 % }
74 \def\x{\endgroup
75 \expandafter\edef\csname llm@AtEnd\endcsname{%
76 \catcode35 \the\catcode35\relax
77 \catcode64 \the\catcode64\relax
78 \catcode123 \the\catcode123\relax
79 \catcode125 \the\catcode125\relax
80 }%
81 }%
82 \x
83 \catcode35 6 % #
84 \catcode64 11 % @
85 \catcode123 1 % {
86 \catcode125 2 % }
87 \def\TMP@EnsureCode#1#2{%
88 \edef\llm@AtEnd{%
89 \llm@AtEnd
90 \catcode#1 \the\catcode#1\relax
91 }%
92 \catcode#1 #2\relax

```

```

93 }
94 \TMP@EnsureCode{40}{12}% (
95 \TMP@EnsureCode{41}{12}% )
96 \TMP@EnsureCode{42}{12}% *
97 \TMP@EnsureCode{45}{12}% -
98 \TMP@EnsureCode{46}{12}% .
99 \TMP@EnsureCode{47}{12}% /
100 \TMP@EnsureCode{58}{12}% :
101 \TMP@EnsureCode{61}{12}% =
102 \TMP@EnsureCode{62}{12}% >
103 \edef\llm@AtEnd{%
104   \llm@AtEnd
105   \escapechar\the\escapechar\relax
106 }
107 \escapechar=92 % '\

Package identification.
108 \NeedsTeXFormat{LaTeX2e}
109 \ProvidesPackage{letltxmacro}%
110 [2008/06/24 v1.3 Let assignment for LaTeX macros (HO)]

```

2.2.2 Main macros

\LetLtxMacro

```

111 \newcommand*{\LetLtxMacro}[2]{%
112   \edef\reserved@a{%
113     \noexpand\protect
114     \expandafter\noexpand
115     \csname\expandafter@gobble\string#2 \endcsname
116   }%
117   \ifx\reserved@a#2\relax
118     \edef#1{%
119       \noexpand\protect
120       \expandafter\noexpand
121       \csname\expandafter@gobble\string#1 \endcsname
122     }%
123     \expandafter\let
124     \csname\expandafter@gobble\string#1 \expandafter\endcsname
125     \csname\expandafter@gobble\string#2 \endcsname
126     \expandafter\llm@LetLtxMacro
127     \csname\expandafter@gobble\string#1 \expandafter\endcsname
128     \csname\expandafter@gobble\string#2 \endcsname
129   \else
130     \llm@LetLtxMacro{#1}{#2}%
131   \fi
132 }

```

\llm@LetLtxMacro

```

133 \def\llm@LetLtxMacro#1#2{%
134   \expandafter\llm@CheckParams\meaning#2:->\@nil{%
135     \begingroup
136     \def\@protected@testopt{%
137       \expandafter\@testopt@gobble
138     }%
139     \def\@testopt##1##2{%
140       \toks@={##2}%
141     }%
142     \let\llm@testopt\@empty
143     \edef\x{%
144       \noexpand\@protected@testopt
145       \noexpand#2%
146       \expandafter\noexpand\csname\string#2\endcsname
147     }%
148     \expandafter\expandafter\expandafter\def

```

```

149 \expandafter\expandafter\expandafter\y
150 \expandafter\expandafter\expandafter{%
151 \expandafter\llm@CarThree#2{}}{\}\llm@nil
152 }%
153 \ifx\x\y
154 #2%
155 \def\llm@testopt{%
156 \noexpand\@protected@testopt
157 \noexpand#1%
158 }%
159 \else
160 \edef\x{%
161 \noexpand\@testopt
162 \expandafter\noexpand\csname\string#2\endcsname
163 }%
164 \expandafter\expandafter\expandafter\def
165 \expandafter\expandafter\expandafter\y
166 \expandafter\expandafter\expandafter{%
167 \expandafter\llm@CarTwo#2{}}{\}\llm@nil
168 }%
169 \ifx\x\y
170 #2%
171 \def\llm@testopt{%
172 \noexpand\@testopt
173 }%
174 \fi
175 \fi
176 \ifx\llm@testopt\@empty
177 \else
178 \llm@protected\xdef\llm@GlobalTemp{%
179 \llm@testopt
180 \expandafter\noexpand\csname\string#1\endcsname
181 {\the\toks0}%
182 }%
183 \fi
184 \expandafter\endgroup\ifx\llm@testopt\@empty
185 \let#1=#2\relax
186 \else
187 \let#1=\llm@GlobalTemp
188 \expandafter\let\csname\string#1\expandafter\endcsname
189 \csname\string#2\endcsname
190 \fi
191 }{%
192 \let#1=#2\relax
193 }%
194 }

```

\llm@CheckParams

```

195 \def\llm@CheckParams#1:->#2\@nil{%
196 \begingroup
197 \def\x{#1}%
198 \ifx\x\llm@macro
199 \endgroup
200 \def\llm@protected{}%
201 \expandafter\@firstoftwo
202 \else
203 \ifx\x\llm@protectedmacro
204 \endgroup
205 \def\llm@protected{\protected}%
206 \expandafter\expandafter\expandafter\@firstoftwo
207 \else
208 \endgroup
209 \expandafter\expandafter\expandafter\@secondoftwo

```

```

210     \fi
211     \fi
212 }

\llm@macro

213 \def\llm@macro{macro}
214 \@onelevel@sanitize\llm@macro

\llm@protectedmacro

215 \def\llm@protectedmacro{\protected macro}
216 \@onelevel@sanitize\llm@protectedmacro

\llm@CarThree

217 \def\llm@CarThree#1#2#3#4\llm@nil{#1#2#3}%

\llm@CarTwo

218 \def\llm@CarTwo#1#2#3\llm@nil{#1#2}%

219 \llm@AtEnd
220 \</package>

```

3 Test

3.1 Catcode checks for loading

```

221 \<test1>
222 \NeedsTeXFormat{LaTeX2e}
223 \documentclass{minimal}
224 \makeatletter
225 \def\RestoreCatcodes{
226   \count@=0 %
227   \loop
228     \edef\RestoreCatcodes{%
229       \RestoreCatcodes
230       \catcode\the\count@=\the\catcode\count@\relax
231     }%
232   \ifnum\count@<255 %
233     \advance\count@\@ne
234   \repeat
235
236   \def\RangeCatcodeInvalid#1#2{%
237     \count@=#1\relax
238     \loop
239       \catcode\count@=15 %
240     \ifnum\count@<#2\relax
241       \advance\count@\@ne
242     \repeat
243   }
244   \def\Test{%
245     \RangeCatcodeInvalid{0}{47}%
246     \RangeCatcodeInvalid{58}{64}%
247     \RangeCatcodeInvalid{91}{96}%
248     \RangeCatcodeInvalid{123}{127}%
249     \catcode'\@=12 %
250     \catcode'\=0 %
251     \catcode'\{=1 %
252     \catcode'\}=2 %
253     \catcode'\#=6 %
254     \catcode'\[=12 %
255     \catcode'\]=12 %
256     \catcode'\%=14 %

```

```

257 \catcode'\ =10 %
258 \catcode13=5 %
259 \RequirePackage{letltxmacro}[2008/06/24]\relax
260 \RestoreCatcodes
261 }
262 \Test
263 \csname @@end\endcsname
264 \end
265 </test1>

```

3.2 Package tests

```

266 <*test2>
267 \NeedsTeXFormat{LaTeX2e}
268 \nofiles
269 \documentclass{minimal}
270
271 \usepackage{letltxmacro}[2008/06/24]
272
273 \usepackage{qstest}
274 \IncludeTests{*}
275 \LogTests{log}{*}{*}
276
277 \makeatletter
278
279 \def\TestDef#1{%
280   \begingroup
281     \@makeother\%
282     \@makeother\%
283     \expandafter\@TestDef\csname #1\endcsname
284 }
285 \def\@TestDef#1#2{%
286   \edef\a{\expandafter\strip@prefix\meaning#1}%
287   \edef\b{\detokenize{#2}}%
288   \ifx\a\b
289     \else
290       \typeout{1 [\a]}%
291       \typeout{2 [\b]}%
292     \fi
293   \endgroup
294   \Expect*{\expandafter\strip@prefix\meaning#1}*{\detokenize{#2}}%
295 }
296 \def\TestEquals#1#2{%
297   \Expect*{%
298     \expandafter\ifx\csname#1\expandafter\endcsname
299       \csname#2\endcsname
300       equals%
301     \else
302       wrong%
303     \fi
304   }{equals}%
305 }
306 \def\SaveA{%
307   \let\SavedA\TestA
308   \expandafter\let\csname\string\SavedA\expandafter\endcsname
309     \csname\string\TestA\endcsname
310   \expandafter\let\csname SavedA \expandafter\endcsname
311     \csname TestA \endcsname
312 }
313 \def\CheckA{%
314   \TestEquals{SavedA}{TestA}%
315   \TestEquals{\string\SavedA}{\string\TestA}%
316   \TestEquals{SavedA }{TestA }%

```

```

317 }
318
319 \begin{qstest}{robust}{robust}
320   \DeclareRobustCommand*\TestA{\Test}%
321   \SaveA
322   \TestDef{TestA}{\protect \TestA }%
323   \CheckA
324   \DeclareRobustCommand{\TestA}{Test}%
325   \SaveA
326   \TestDef{TestA}{\protect \TestA }%
327   \LetLtxMacro\TestB\TestA
328   \TestDef{TestB}{\protect \TestB }%
329   \TestEquals{TestB }{TestA }%
330   \CheckA
331 \end{qstest}
332
333 \begin{qstest}{default}{default}
334   \newcommand{\TestA}[1][\relax default \empty]{TestA #1}%
335   \SaveA
336   \TestDef
337   {\TestA}{\@protected@testopt \TestA \TestA {\relax default \empty }}%
338   \LetLtxMacro\TestB\TestA
339   \TestDef
340   {\TestB}{\@protected@testopt \TestB \TestB {\relax default \empty }}%
341   \CheckA
342 \end{qstest}
343
344 \begin{qstest}{robustdefault}{robustdefault}
345   \DeclareRobustCommand{\TestA}[1][\default]{TestA}%
346   \SaveA
347   \LetLtxMacro\TestB\TestA
348   \TestDef{TestB}{\protect \TestB }%
349   \TestDef{TestB }{\@protected@testopt \TestB \TestB {\default }}%
350   \CheckA
351 \end{qstest}
352
353 \begin{qstest}{plain}{plain}
354   \LetLtxMacro\NewRelax\relax
355   \Expect*{\meaning\relax}*{\string\relax}%
356   \Expect*{\meaning\NewRelax}*{\string\relax}%
357   \LetLtxMacro\NewHbox\hbox
358   \Expect*{\meaning\hbox}*{\string\hbox}%
359   \Expect*{\meaning\NewHbox}*{\string\hbox}%
360   \LetLtxMacro\NewEmpty\empty
361   \Expect*{\meaning\empty}{macro:->}%
362   \Expect*{\meaning\NewEmpty}{macro:->}%
363   \def\TestA{\iffalse}%
364   \LetLtxMacro\TestB\TestA
365   \TestDef{TestB}{\iffalse }%
366   \TestEquals{TestA}{TestB}%
367 \end{qstest}
368 \usepackage{etoolbox}[2008/06/22]
369
370 \begin{qstest}{oldnewrobustcmd}{oldnewrobustcmd}
371   \protected\edef\TestA{%
372     \noexpand\@protected@testopt
373     \noexpand\TestA
374     \expandafter\noexpand\csname\string\TestA\endcsname
375     {\noexpand\default}%
376   }%
377   \expandafter\protected\expandafter
378   \def\csname\string\TestA\endcsname{TestA}%

```

```

379 \SaveA
380 \LetLtxMacro\TestB\TestA
381 \TestDef{TestB}{\@protected@testopt \TestB \TestB {\default }}%
382 \protected\edef\TestC{%
383   \noexpand\@protected@testopt
384   \noexpand\TestB
385   \expandafter\noexpand\csname\string\TestB\endcsname
386   {\noexpand\default}}%
387 }%
388 \TestEquals{TestB}{TestC}%
389 \CheckA
390 \end{qstest}
391
392 \begin{qstest}{newrobustcmd}{newrobustcmd}
393   \newrobustcmd{\TestA}[1][\default]{TestA}%
394   \SaveA
395   \LetLtxMacro\TestB\TestA
396   \TestDef{TestB}{\@testopt \TestB {\default }}%
397   \protected\edef\TestC{%
398     \noexpand\@testopt
399     \expandafter\noexpand\csname\string\TestB\endcsname
400     {\noexpand\default}}%
401   }%
402   \TestEquals{TestB}{TestC}%
403   \CheckA
404 \end{qstest}
405
406 \begin{qstest}{robustifyopt}{robustifyopt}
407   \newcommand{\TestA}[2][\default]{}%
408   \TestDef{TestA}{\@protected@testopt \TestA \TestA {\default }}%
409   \robustify\TestA
410   \TestDef{TestA}{\@protected@testopt \TestA \TestA {\default }}%
411   \protected\edef\TestC{%
412     \noexpand\@protected@testopt
413     \noexpand\TestA
414     \expandafter\noexpand\csname\string\TestA\endcsname
415     {\noexpand\default}}%
416   }%
417   \TestEquals{TestA}{TestC}%
418   \SaveA
419   \LetLtxMacro\TestB\TestA
420   \TestDef{TestB}{\@protected@testopt \TestB \TestB {\default }}%
421   \protected\edef\TestC{%
422     \noexpand\@protected@testopt
423     \noexpand\TestB
424     \expandafter\noexpand\csname\string\TestB\endcsname
425     {\noexpand\default}}%
426   }%
427   \TestEquals{TestB}{TestC}%
428   \CheckA
429 \end{qstest}
430
431 \begin{qstest}{robustifydeclare}{robustifydeclare}
432   \DeclareRobustCommand\TestA{\iffalse}%
433   \robustify\TestA
434   \SaveA
435   \LetLtxMacro\TestB\TestA
436   \TestDef{TestB}{\iffalse }%
437   % before etoolbox 2008/06/22: \protected\long\def
438   % since etoolbox 2008/06/22: \protected\def
439   \protected\def\TestC{\iffalse}%
440   \TestEquals{TestB}{TestC}%

```

```

441 \CheckA
442 \end{qstest}
443
444 \begin{qstest}{robustifydeclarelong}{robustifydeclarelong}
445 \DeclareRobustCommand\TestA[1]{\iffalse}%
446 \robustify\TestA
447 \SaveA
448 \LetLtxMacro\TestB\TestA
449 \TestDef{TestB}{\iffalse}%
450 % before etoolbox 2008/06/22: \protected\long\def
451 % since etoolbox 2008/06/22: \protected\def
452 \protected\long\def\TestC#1{\iffalse}%
453 \TestEquals{TestB}{TestC}%
454 \CheckA
455 \end{qstest}
456 \csname @@end\endcsname
457 </test2>

```

4 Installation

4.1 Download

Package. This package is available on CTAN¹:

[CTAN:macros/latex/contrib/oberdiek/letltxmacro.dtx](#) The source file.

[CTAN:macros/latex/contrib/oberdiek/letltxmacro.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 T_EX 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-T_EX:

```
tex letltxmacro.dtx
```

¹<ftp://ftp.ctan.org/tex-archive/>

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

```
letltxmacro.sty      → tex/latex/oberdiek/letltxmacro.sty
letltxmacro.pdf      → doc/latex/oberdiek/letltxmacro.pdf
letltxmacro-showcases.tex → doc/latex/oberdiek/letltxmacro-showcases.tex
test/letltxmacro-test1.tex → doc/latex/oberdiek/test/letltxmacro-test1.tex
test/letltxmacro-test2.tex → doc/latex/oberdiek/test/letltxmacro-test2.tex
letltxmacro.dtx      → source/latex/oberdiek/letltxmacro.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 (`te \TeX` , `mik \TeX` , ...) relies on file name databases, you must refresh these. For example, `te \TeX` users run `texhash` or `mktextlsr`.

4.5 Some details for the interested

Attached source. The PDF documentation on CTAN also includes the `.dtx` source file. It can be extracted by AcrobatReader 6 or higher. Another option is `pdftk`, e.g. unpack the file into the current directory:

```
pdftk letltxmacro.pdf unpack_files output .
```

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{letltxmacro.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 `pdf \LaTeX` :

```
pdflatex letltxmacro.dtx
makeindex -s gind.ist letltxmacro.idx
pdflatex letltxmacro.dtx
makeindex -s gind.ist letltxmacro.idx
pdflatex letltxmacro.dtx
```

5 History

[2008/06/09 v1.0]

- First version.

[2008/06/12 v1.1]

- Support for etoolbox's `\newrobustcmd` added.

[2008/06/13 v1.2]

- Support for etoolbox's `\robustify` added.

[2008/06/24 v1.3]

- Test file adapted for etoolbox 2008/06/22 v1.6.

6 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; numbers in roman refer to the code lines where the entry is used.

| Symbols | | 90, 92, 230, 239, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258 |
|--|--|---|
| <code>\#</code> | 253 | <code>\CheckA</code> 313, 323, 330, 341, 350, 389, 403, 428, 441, 454 |
| <code>\%</code> | 256 | <code>\cmd</code> 35, 38, 41, 48, 51, 56, 57, 60, 61, 64, 65 |
| <code>\@</code> | 249 | <code>\count@</code> 226, 230, 232, 233, 237, 239, 240, 241 |
| <code>\@@end</code> | 68 | <code>\csname</code> 10, 11, 75, 115, 121, 124, 125, 127, 128, 146, 162, 180, 188, 189, 263, 283, 298, 299, 308, 309, 310, 311, 374, 378, 385, 399, 414, 424, 456 |
| <code>\@TestDef</code> | 283, 285 | |
| <code>\@backslashchar</code> | 21 | |
| <code>\@empty</code> | 142, 176, 184 | |
| <code>\@firstoftwo</code> | 201, 206 | |
| <code>\@gobble</code> <i>115, 121, 124, 125, 127, 128, 137</i> | | |
| <code>\@ifundefined</code> | 8 | |
| <code>\@makeoother</code> | 281, 282 | |
| <code>\@ne</code> | 233, 241 | |
| <code>\@nil</code> | 134, 195 | |
| <code>\@onelevel@sanitize</code> | 26, 214, 216 | |
| <code>\@percentchar</code> | 5 | |
| <code>\@protected@testopt</code> | 136, 144, 156, 337, 340, 349, 372, 381, 383, 408, 410, 412, 420, 422 | |
| <code>\@secondoftwo</code> | 209 | |
| <code>\@testopt</code> . | 137, 139, 161, 172, 396, 398 | |
| <code>\[</code> | 254 | |
| <code>\backslash</code> ... | 18, 19, 21, 107, 250, 281, 337, 340, 349, 381, 396, 408, 410, 420 | |
| <code>\{</code> | 251 | |
| <code>\}</code> | 252 | |
| <code>\]</code> | 255 | |
| <code>_</code> | 257, 282 | |
| A | | |
| <code>\a</code> | 286, 288, 290 | |
| <code>\advance</code> | 233, 241 | |
| B | | |
| <code>\b</code> | 287, 288, 291 | |
| <code>\begin</code> | 319, 333, 344, 353, 370, 392, 406, 431, 444 | |
| C | | |
| <code>\catcode</code> | 72, 73, 76, 77, 78, 79, 83, 84, 85, 86, | |
| D | | |
| <code>\DeclareRobustCommand</code> | 38, 41, 60, 64, 320, 324, 345, 432, 445 | |
| <code>\default</code> | 345, 349, 375, 381, 386, 393, 396, 400, 407, 408, 410, 415, 420, 425 | |
| <code>\DefString</code> | 25, 26, 27 | |
| <code>\detokenize</code> | 287, 294 | |
| <code>\documentclass</code> | 223, 269 | |
| E | | |
| <code>\empty</code> | 334, 337, 340, 360, 361 | |
| <code>\end</code> | 264, 331, 342, 351, 367, 390, 404, 429, 442, 455 | |
| <code>\endcsname</code> | 10, 11, 75, 115, 121, 124, 125, 127, 128, 146, 162, 180, 188, 189, 263, 283, 298, 299, 308, 309, 310, 311, 374, 378, 385, 399, 414, 424, 456 | |
| <code>\escapechar</code> | 105, 107 | |
| <code>\Expect</code> | 294, 297, 355, 356, 358, 359, 361, 362 | |
| H | | |
| <code>\hbox</code> | 357, 358, 359 | |
| I | | |
| <code>\iffalse</code> | 363, 365, 432, 436, 439, 445, 449, 452 | |
| <code>\ifnum</code> | 232, 240 | |

| | |
|---|--|
| <code>\ifx</code> | 117, 153, 169, 176, 184, 198, 203, 288, 298 |
| <code>\IncludeTests</code> | 274 |
| L | |
| <code>\LetLtxMacro</code> | 2, 111, 327, 338, 347, 354, 357, 360, 364, 380, 395, 419, 435, 448 |
| <code>\Line</code> | 4, 9, 24, 27, 33, 45, 46, 53, 54 |
| <code>\llm@AtEnd</code> | 88, 89, 103, 104, 219 |
| <code>\llm@CarThree</code> | 151, 217 |
| <code>\llm@CarTwo</code> | 167, 218 |
| <code>\llm@CheckParams</code> | 134, 195 |
| <code>\llm@GlobalTemp</code> | 178, 187 |
| <code>\llm@LetLtxMacro</code> | 126, 130, 133 |
| <code>\llm@macro</code> | 198, 213 |
| <code>\llm@nil</code> | 151, 167, 217, 218 |
| <code>\llm@protected</code> | 178, 200, 205 |
| <code>\llm@protectedmacro</code> | 203, 215 |
| <code>\llm@testopt</code> | 142, 155, 171, 176, 179, 184 |
| <code>\LogTests</code> | 275 |
| <code>\loop</code> | 227, 238 |
| M | |
| <code>\makeatletter</code> | 3, 224, 277 |
| <code>\meaning</code> | 11, 134, 286, 294, 355, 356, 358, 359, 361, 362 |
| N | |
| <code>\NeedsTeXFormat</code> | 2, 108, 222, 267 |
| <code>\newcommand</code> | 4, 7, 15, 22, 25, 35, 56, 111, 334, 407 |
| <code>\NewEmpty</code> | 360, 362 |
| <code>\NewHbox</code> | 357, 359 |
| <code>\NewRelax</code> | 354, 356 |
| <code>\newrobustcmd</code> | 48, 51, 393 |
| <code>\nofiles</code> | 268 |
| P | |
| <code>\protect</code> | 113, 119, 322, 326, 328, 348 |
| <code>\protected</code> | 205, 215, 371, 377, 382, 397, 411, 421, 437, 438, 439, 450, 451, 452 |
| <code>\ProvidesPackage</code> | 109 |
| R | |
| <code>\RangeCatcodeInvalid</code> | 236, 245, 246, 247, 248 |
| <code>\repeat</code> | 234, 242 |
| <code>\RequirePackage</code> | 44, 259 |
| <code>\reserved@a</code> | 112, 117 |
| <code>\RestoreCatcodes</code> | 225, 228, 229, 260 |
| <code>\robustify</code> | 54, 57, 61, 65, 409, 433, 446 |
| S | |
| <code>\SaveA</code> | 306, 321, 325, 335, 346, 379, 394, 418, 434, 447 |
| <code>\SavedA</code> | 307, 308, 315 |
| <code>\ShowCmdName</code> | 7, 16, 17, 18, 19 |
| <code>\ShowCmds</code> | 15, 29 |
| <code>\ShowDef</code> | 22, 34, 37, 40, 47, 50, 55, 59, 63 |
| <code>\space</code> | 10 |
| <code>\strip@prefix</code> | 286, 294 |
| T | |
| <code>\Test</code> | 244, 262 |
| <code>\TestA</code> | 307, 309, 315, 320, 322, 324, 326, 327, 334, 337, 338, 345, 347, 363, 364, 371, 373, 374, 378, 380, 393, 395, 407, 408, 409, 410, 413, 414, 419, 432, 433, 435, 445, 446, 448 |
| <code>\TestB</code> | 327, 328, 338, 340, 347, 348, 349, 364, 380, 381, 384, 385, 395, 399, 419, 420, 423, 424, 435, 448 |
| <code>\TestC</code> | 382, 397, 411, 421, 439, 452 |
| <code>\TestDef</code> | 279, 322, 326, 328, 336, 339, 348, 349, 365, 381, 396, 408, 410, 420, 436, 449 |
| <code>\TestEquals</code> | 296, 314, 315, 316, 329, 366, 388, 402, 417, 427, 440, 453 |
| <code>\the</code> | 76, 77, 78, 79, 90, 105, 181, 230 |
| <code>\TMP@EnsureCode</code> | 87, 94, 95, 96, 97, 98, 99, 100, 101, 102 |
| <code>\toks@</code> | 140, 181 |
| <code>\typeout</code> | 5, 32, 43, 67, 290, 291 |
| U | |
| <code>\usepackage</code> | 271, 273, 368 |
| X | |
| <code>\x</code> | 74, 82, 143, 153, 160, 169, 197, 198, 203 |
| Y | |
| <code>\y</code> | 149, 153, 165, 169 |