The letltxmacro package

Heiko Oberdiek
<heiko.oberdiek at googlemail.com>

2010/09/02 v1.4

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 . . . 4
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 **12**
4.1 Download . . . 12
4.2 Bundle installation . . . 12
4.3 Package installation . . . 13
4.4 Refresh file name databases . . . 13
4.5 Some details for the interested . . . 13

5 History **14**
[2008/06/09 v1.0] . . . 14
[2008/06/12 v1.1] . . . 14
[2008/06/13 v1.2] . . . 14
[2008/06/24 v1.3] . . . 14
[2010/09/02 v1.4] . . . 14

6 Index **14**

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 LaTeX 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 TeX'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`.

`\GlobalLetLtxMacro` {⟨*new macro*⟩} {⟨*old macro*⟩}

Like `\LetLtxMacro`, but the ⟨*new macro*⟩ is defined globally. Since version 2010/09/02 v1.4.

1.1 Supported macro definition commands

`\newcommand`, `\renewcommand`	latex/base
`\newenvironment`, `\renewenvironment`	latex/base
`\DeclareRobustCommand`	latex/base
`\newrobustcmd`, `\renewrobustcmd`	etoolbox
`\robustify`	etoolbox 2008/06/22 v1.6

2 Implementation

2.1 Show cases

2.1.1 letltxmacro-showcases.tex

```
⟨*showcases⟩
\NeedsTeXFormat{LaTeX2e}
\makeatletter
```

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

```
\newcommand*{\Line}[1]{%
  \typeout{\@percentchar#1}%
}

\newcommand*{\ShowCmdName}[1]{%
  \@ifundefined{#1}{}{%
    \Line{%
      \space\space(\expandafter\string\csname#1\endcsname) = %
      (\expandafter\meaning\csname#1\endcsname)%
    }%
```

```
  }%
}
\newcommand*{\ShowCmds}[1]{%
  \ShowCmdName{#1}%
  \ShowCmdName{#1 }%
  \ShowCmdName{\\#1}%
  \ShowCmdName{\\#1 }%
}
\let\\\@backslashchar
```

\ShowDef

```
\newcommand*{\ShowDef}[2]{%
  \begingroup
    \Line{}%
    \newcommand*{\DefString}{#2}%
    \@onelevel@sanitize\DefString
    \Line{\DefString}%
    #2%
    \ShowCmds{#1}%
  \endgroup
}

\typeout{}
\Line{* LaTeX definitions:}
\ShowDef{cmd}{%
  \newcommand{\cmd}[2][default]{}%
}
\ShowDef{cmd}{%
  \DeclareRobustCommand{\cmd}{}%
}
\ShowDef{cmd}{%
  \DeclareRobustCommand{\cmd}[2][default]{}%
}
\typeout{}
```

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

```
\RequirePackage{etoolbox}[2008/06/12]%
\Line{}
\Line{* etoolbox's robust definitions:}
\ShowDef{cmd}{%
  \newrobustcmd{\cmd}{}%
}
\ShowDef{cmd}{%
  \newrobustcmd{\cmd}[2][default]{}%
}
\Line{}
\Line{* etoolbox's \string\robustify:}
\ShowDef{cmd}{%
  \newcommand{\cmd}[2][default]{} %
  \robustify{\cmd}%
}
\ShowDef{cmd}{%
  \DeclareRobustCommand{\cmd}{} %
  \robustify{\cmd}%
}
\ShowDef{cmd}{%
  \DeclareRobustCommand{\cmd}[2][default]{} %
  \robustify{\cmd}%
}
\typeout{}
\@@end
```

⟨/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\catcode61\catcode48\catcode32=10\relax%
72   \catcode13=5 % ^^M
73   \endlinechar=13 %
74   \catcode123=1 % {
75   \catcode125=2 % }
76   \catcode64=11 % @
77   \def\x{\endgroup
78     \expandafter\edef\csname llm@AtEnd\endcsname{%
79       \endlinechar=\the\endlinechar\relax
80       \catcode13=\the\catcode13\relax
81       \catcode32=\the\catcode32\relax
82       \catcode35=\the\catcode35\relax
83       \catcode61=\the\catcode61\relax
84       \catcode64=\the\catcode64\relax
85       \catcode123=\the\catcode123\relax
86       \catcode125=\the\catcode125\relax
87     }%
88   }%
```

```
\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\llm@AtEnd{%
    \llm@AtEnd
    \catcode#1=\the\catcode#1\relax
  }%
  \catcode#1=#2\relax
}
\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{58}{12}% :
\TMP@EnsureCode{62}{12}% >
\TMP@EnsureCode{91}{12}% [
\TMP@EnsureCode{93}{12}% ]
\edef\llm@AtEnd{%
  \llm@AtEnd
  \escapechar\the\escapechar\relax
  \noexpand\endinput
}
\escapechar=92 % `\\
```

Package identification.

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{letltxmacro}%
  [2010/09/02 v1.4 Let assignment for LaTeX macros (HO)]
```

2.2.2 Main macros

`\LetLtxMacro`

```
\newcommand*{\LetLtxMacro}{%
  \llm@ModeLetLtxMacro{}%
}
```

`\GlobalLetLtxMacro`

```
\newcommand*{\GlobalLetLtxMacro}{%
  \llm@ModeLetLtxMacro\global
}
```

`\llm@ModeLetLtxMacro`

```
\newcommand*{\llm@ModeLetLtxMacro}[3]{%
  \edef\llm@escapechar{\the\escapechar}%
  \escapechar=-1 %
  \edef\reserved@a{%
    \noexpand\protect
    \expandafter\noexpand
    \csname\string#3 \endcsname
  }%
  \ifx\reserved@a#3\relax
    #1\edef#2{%
      \noexpand\protect
      \expandafter\noexpand
      \csname\string#2 \endcsname
    }%
```

```
    #1\expandafter\let
    \csname\string#2 \expandafter\endcsname
    \csname\string#3 \endcsname
    \expandafter\llm@LetLtxMacro
        \csname\string#2 \expandafter\endcsname
        \csname\string#3 \endcsname{#1}%
  \else
    \llm@LetLtxMacro{#2}{#3}{#1}%
  \fi
  \escapechar=\llm@escapechar\relax
}
```

\llm@LetLtxMacro

```
\def\llm@LetLtxMacro#1#2#3{%
  \escapechar=92 %
  \expandafter\llm@CheckParams\meaning#2:->\@nil{%
    \begingroup
      \def\@protected@testopt{%
        \expandafter\@testopt\@gobble
      }%
      \def\@testopt##1##2{%
        \toks@={##2}%
      }%
      \let\llm@testopt\@empty
      \edef\x{%
        \noexpand\@protected@testopt
        \noexpand#2%
        \expandafter\noexpand\csname\string#2\endcsname
      }%
      \expandafter\expandafter\expandafter\def
      \expandafter\expandafter\expandafter\y
      \expandafter\expandafter\expandafter{%
        \expandafter\llm@CarThree#2{}{}{}\llm@nil
      }%
      \ifx\x\y
        #2%
        \def\llm@testopt{%
          \noexpand\@protected@testopt
          \noexpand#1%
        }%
      \else
        \edef\x{%
          \noexpand\@testopt
          \expandafter\noexpand
          \csname\string#2\endcsname
        }%
        \expandafter\expandafter\expandafter\def
        \expandafter\expandafter\expandafter\y
        \expandafter\expandafter\expandafter{%
          \expandafter\llm@CarTwo#2{}{}\llm@nil
        }%
        \ifx\x\y
          #2%
          \def\llm@testopt{%
            \noexpand\@testopt
          }%
        \fi
      \fi
      \ifx\llm@testopt\@empty
      \else
        \llm@protected\xdef\llm@GlobalTemp{%
          \llm@testopt
          \expandafter\noexpand
```

```
          \csname\string#1\endcsname
          {\the\toks@}%
        }%
      \fi
    \expandafter\endgroup\ifx\llm@testopt\@empty
      #3\let#1=#2\relax
    \else
      #3\let#1=\llm@GlobalTemp
      #3\expandafter\let
          \csname\string#1\expandafter\endcsname
          \csname\string#2\endcsname
    \fi
  }{%
    #3\let#1=#2\relax
  }%
}
```

\llm@CheckParams

```
\def\llm@CheckParams#1:->#2\@nil{%
  \begingroup
    \def\x{#1}%
  \ifx\x\llm@macro
    \endgroup
    \def\llm@protected{}%
    \expandafter\@firstoftwo
  \else
    \ifx\x\llm@protectedmacro
      \endgroup
      \def\llm@protected{\protected}%
      \expandafter\expandafter\expandafter\@firstoftwo
    \else
      \endgroup
      \expandafter\expandafter\expandafter\@secondoftwo
    \fi
  \fi
}
```

\llm@macro

```
\def\llm@macro{macro}
\@onelevel@sanitize\llm@macro
```

\llm@protectedmacro

```
\def\llm@protectedmacro{\protected macro}
\@onelevel@sanitize\llm@protectedmacro
```

\llm@CarThree

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

\llm@CarTwo

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

\llm@AtEnd%
⟨/package⟩
```

3 Test

3.1 Catcode checks for loading

```
⟨*test1⟩
\NeedsTeXFormat{LaTeX2e}
```

```
\documentclass{minimal}
\makeatletter
\def\RestoreCatcodes{}
\count@=0 %
\loop
  \edef\RestoreCatcodes{%
    \RestoreCatcodes
    \catcode\the\count@=\the\catcode\count@\relax
  }%
\ifnum\count@<255 %
  \advance\count@\@ne
\repeat

\def\RangeCatcodeInvalid#1#2{%
  \count@=#1\relax
  \loop
    \catcode\count@=15 %
  \ifnum\count@<#2\relax
    \advance\count@\@ne
  \repeat
}
\def\Test{%
  \RangeCatcodeInvalid{0}{47}%
  \RangeCatcodeInvalid{58}{64}%
  \RangeCatcodeInvalid{91}{96}%
  \RangeCatcodeInvalid{123}{127}%
  \catcode`\@=12 %
  \catcode`\\=0 %
  \catcode`\{=1 %
  \catcode`\}=2 %
  \catcode`\#=6 %
  \catcode`\[=12 %
  \catcode`\]=12 %
  \catcode`\%=14 %
  \catcode`\ =10 %
  \catcode13=5 %
  \RequirePackage{letltxmacro}[2010/09/02]\relax
  \RestoreCatcodes
}
\Test
\csname @@end\endcsname
\end
```

⟨/test1⟩

3.2 Package tests

⟨*test2⟩

```
\NeedsTeXFormat{LaTeX2e}
\nofiles
\documentclass{minimal}

\usepackage{letltxmacro}[2010/09/02]

\usepackage{qstest}
\IncludeTests{*}
\LogTests{log}{*}{*}

\makeatletter

\def\TestDef#1{%
  \begingroup
    \@makeother\\%
    \@makeother\ %
```

```
    \expandafter\@TestDef\csname #1\endcsname
}
\def\@TestDef#1#2{%
    \edef\a{\expandafter\strip@prefix\meaning#1}%
    \edef\b{\detokenize{#2}}%
    \ifx\a\b
    \else
      \typeout{1 [\a]}%
      \typeout{2 [\b]}%
    \fi
  \endgroup
  \Expect*{\expandafter\strip@prefix\meaning#1}*{\detokenize{#2}}%
}
\def\TestEquals#1#2{%
  \Expect*{%
    \expandafter\ifx\csname#1\expandafter\endcsname
                    \csname#2\endcsname
      equals%
    \else
      wrong%
    \fi
  }{equals}%
}
\def\SaveA{%
  \let\SavedA\TestA
  \expandafter\let\csname\string\SavedA\expandafter\endcsname
                  \csname\string\TestA\endcsname
  \expandafter\let\csname SavedA \expandafter\endcsname
                  \csname TestA \endcsname
}
\def\CheckA{%
  \TestEquals{SavedA}{TestA}%
  \TestEquals{\string\SavedA}{\string\TestA}%
  \TestEquals{SavedA }{TestA }%
}

\begin{qstest}{robust}{robust}
  \begingroup
    \DeclareRobustCommand*{\TestA}{Test}%
    \SaveA
    \TestDef{TestA}{\protect \TestA  }%
    \CheckA
    \DeclareRobustCommand{\TestA}{Test}%
    \SaveA
    \TestDef{TestA}{\protect \TestA  }%
    \LetLtxMacro\TestB\TestA
    \TestDef{TestB}{\protect \TestB  }%
    \TestEquals{TestB }{TestA }%
    \CheckA
    \GlobalLetLtxMacro\TestC\TestA
    \CheckA
  \endgroup
  \TestDef{TestC}{\protect \TestC  }%
  \DeclareRobustCommand{\TestA}{Test}%
  \TestEquals{TestC }{TestA }%
\end{qstest}

\begin{qstest}{default}{default}
  \begingroup
    \newcommand{\TestA}[1][\relax default \empty]{TestA #1}%
    \SaveA
    \TestDef
```

```
    {TestA}{\@protected@testopt \TestA \\TestA {\relax default \empty }}%
    \LetLtxMacro\TestB\TestA
    \TestDef
    {TestB}{\@protected@testopt \TestB \\TestB {\relax default \empty }}%
    \CheckA
    \TestEquals{\string\TestB}{\string\TestA}%
    \GlobalLetLtxMacro\TestC\TestA
    \CheckA
  \endgroup
  \TestDef
  {TestC}{\@protected@testopt \TestC \\TestC {\relax default \empty }}%
  \newcommand{\TestA}[1][\relax default \empty]{TestA #1}%
  \TestEquals{\string\TestC}{\string\TestA}%
\end{qstest}

\begin{qstest}{robustdefault}{robustdefault}
  \begingroup
    \DeclareRobustCommand{\TestA}[1][\default]{TestA}%
    \SaveA
    \LetLtxMacro\TestB\TestA
    \TestDef{TestB}{\protect \TestB  }%
    \TestDef{TestB }{\@protected@testopt \TestB  \\TestB  {\default }}%
    \CheckA
    \GlobalLetLtxMacro\TestC\TestA
    \CheckA
  \endgroup
  \TestDef{TestC}{\protect \TestC  }%
  \TestDef{TestC }{\@protected@testopt \TestC  \\TestC  {\default }}%
\end{qstest}

\begin{qstest}{plain}{plain}
  \LetLtxMacro\NewRelax\relax
  \Expect*{\meaning\relax}*{\string\relax}%
  \Expect*{\meaning\NewRelax}*{\string\relax}%
  \LetLtxMacro\NewHbox\hbox
  \Expect*{\meaning\hbox}*{\string\hbox}%
  \Expect*{\meaning\NewHbox}*{\string\hbox}%
  \LetLtxMacro\NewEmpty\empty
  \Expect*{\meaning\empty}{macro:->}%
  \Expect*{\meaning\NewEmpty}{macro:->}%
  \def\TestA{\iffalse}%
  \LetLtxMacro\TestB\TestA
  \TestDef{TestB}{\iffalse }%
  \TestEquals{TestA}{TestB}%
\end{qstest}

\usepackage{etoolbox}[2008/06/22]

\begin{qstest}{oldnewrobustcmd}{oldnewrobustcmd}
  \begingroup
    \protected\edef\TestA{%
      \noexpand\@protected@testopt
      \noexpand\TestA
      \expandafter\noexpand\csname\string\TestA\endcsname
      {\noexpand\default}%
    }%
    \expandafter\protected\expandafter
    \def\csname\string\TestA\endcsname{TestA}%
    \SaveA
    \LetLtxMacro\TestB\TestA
    \TestDef{TestB}{\@protected@testopt \TestB \\TestB {\default }}%
    \protected\edef\TestC{%
      \noexpand\@protected@testopt
```

```
      \noexpand\TestB
      \expandafter\noexpand\csname\string\TestB\endcsname
      {\noexpand\default}%
    }%
    \TestEquals{TestB}{TestC}%
    \CheckA
    \GlobalLetLtxMacro\TestD\TestA
    \CheckA
  \endgroup
  \TestDef{TestD}{\@protected@testopt \TestD \\TestD {\default }}%
  \protected\edef\TestC{%
    \noexpand\@protected@testopt
    \noexpand\TestD
    \expandafter\noexpand\csname\string\TestD\endcsname
    {\noexpand\default}%
  }%
  \TestEquals{TestD}{TestC}%
\end{qstest}

\begin{qstest}{newrobustcmd}{newrobustcmd}
  \begingroup
    \newrobustcmd{\TestA}[1][\default]{TestA}%
    \SaveA
    \LetLtxMacro\TestB\TestA
    \TestDef{TestB}{\@testopt \\TestB {\default }}%
    \protected\edef\TestC{%
      \noexpand\@testopt
      \expandafter\noexpand\csname\string\TestB\endcsname
      {\noexpand\default}%
    }%
    \TestEquals{TestB}{TestC}%
    \CheckA
    \GlobalLetLtxMacro\TestD\TestA
    \CheckA
  \endgroup
  \TestDef{TestD}{\@testopt \\TestD {\default }}%
  \protected\edef\TestC{%
    \noexpand\@testopt
    \expandafter\noexpand\csname\string\TestD\endcsname
    {\noexpand\default}%
  }%
  \TestEquals{TestD}{TestC}%
\end{qstest}

\begin{qstest}{robustifyopt}{robustifyopt}
  \newcommand{\TestA}[2][\default]{}%
  \TestDef{TestA}{\@protected@testopt \TestA \\TestA {\default }}%
  \robustify\TestA
  \TestDef{TestA}{\@protected@testopt \TestA \\TestA {\default }}%
  \protected\edef\TestC{%
    \noexpand\@protected@testopt
    \noexpand\TestA
    \expandafter\noexpand\csname\string\TestA\endcsname
    {\noexpand\default}%
  }%
  \TestEquals{TestA}{TestC}%
  \SaveA
  \LetLtxMacro\TestB\TestA
  \TestDef{TestB}{\@protected@testopt \TestB \\TestB {\default }}%
  \protected\edef\TestC{%
    \noexpand\@protected@testopt
    \noexpand\TestB
```

```
    \expandafter\noexpand\csname\string\TestB\endcsname
    {\noexpand\default}%
  }%
  \TestEquals{TestB}{TestC}%
  \CheckA
\end{qstest}

\begin{qstest}{robustifydeclare}{robustifydeclare}
  \DeclareRobustCommand\TestA{\iffalse}%
  \robustify\TestA
  \SaveA
  \LetLtxMacro\TestB\TestA
  \TestDef{TestB}{\iffalse }%
  % before etoolbox 2008/06/22: \protected\long\def
  % since etoolbox 2008/06/22: \protected\def
  \protected\def\TestC{\iffalse}%
  \TestEquals{TestB}{TestC}%
  \CheckA
\end{qstest}

\begin{qstest}{robustifydeclarelong}{robustifydeclarelong}
  \DeclareRobustCommand\TestA[1]{\iffalse}%
  \robustify\TestA
  \SaveA
  \LetLtxMacro\TestB\TestA
  \TestDef{TestB}{\iffalse }%
  % before etoolbox 2008/06/22: \protected\long\def
  % since etoolbox 2008/06/22: \protected\def
  \protected\long\def\TestC#1{\iffalse}%
  \TestEquals{TestB}{TestC}%
  \CheckA
\end{qstest}

\csname @@end\endcsname

⟨/test2⟩
```

4 Installation

4.1 Download

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

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 TEX 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
```

[1] ftp://ftp.ctan.org/tex-archive/

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 TEX:

```
tex letltxmacro.dtx
```

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 (teTEX, mikTEX, ...) relies on file name databases, you must refresh these. For example, teTEX users run `texhash` or `mktexlsr`.

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 pdfLaTeX:

```
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.

[2010/09/02 v1.4]

- `\GlobalLetLtxMacro` added.

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; plain numbers refer to the code lines where the entry is used.

Symbols

\# 277
\% 280
\@ 273
\@@end 68
\@TestDef 307, 309
\@backslashchar 21
\@empty 163, 198, 207
\@firstoftwo 225, 230
\@gobble 158
\@ifundefined 8
\@makeother 305, 306
\@ne 257, 265
\@nil 155, 219
\@onelevel@sanitize 26, 238, 240
\@percentchar 5
\@protected@testopt 157, 165, 177, 369, 372, 379, 390, 396, 419, 428, 430, 440, 442, 477, 479, 481, 489, 491
\@secondoftwo 233
\@testopt 158, 160, 182, 194, 455, 457, 466, 468
\[. 278
\\ 18, 19, 21, 118, 274, 305, 369, 372, 379, 390, 396, 428, 440, 455, 466, 477, 479, 489
\{ 275
\} 276
\] 279
\␣ 281, 306

A

\a 310, 312, 314
\advance 257, 265

B

\b 311, 312, 315
\begin 343, 364, 384, 399, 416, 450, 475, 500, 513

C

\catcode 71, 72, 74, 75, 76, 80, 81, 82, 83, 84, 85, 86, 89, 90, 92, 93, 94, 95, 99, 101, 254, 263, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282

\CheckA 337, 348, 355, 357, 373, 376, 391, 393, 436, 438, 462, 464, 497, 510, 523
\cmd 35, 38, 41, 48, 51, 56, 57, 60, 61, 64, 65
\count@ 250, 254, 256, 257, 261, 263, 264, 265
\csname . . 10, 11, 78, 134, 140, 143, 144, 146, 147, 167, 184, 203, 212, 213, 287, 307, 322, 323, 332, 333, 334, 335, 421, 425, 432, 444, 458, 469, 483, 493, 525

D

\DeclareRobustCommand 38, 41, 60, 64, 345, 349, 360, 386, 501, 514
\default 386, 390, 396, 422, 428, 433, 440, 445, 452, 455, 459, 466, 470, 476, 477, 479, 484, 489, 494
\DefString 25, 26, 27
\detokenize 311, 318
\documentclass 247, 293

E

\empty 366, 369, 372, 379, 380, 406, 407
\end 288, 362, 382, 397, 413, 448, 473, 498, 511, 524
\endcsname 10, 11, 78, 134, 140, 143, 144, 146, 147, 167, 184, 203, 212, 213, 287, 307, 322, 323, 332, 333, 334, 335, 421, 425, 432, 444, 458, 469, 483, 493, 525
\endinput 116
\endlinechar 73, 79, 91
\escapechar 115, 118, 129, 130, 151, 154
\Expect 318, 321, 401, 402, 404, 405, 407, 408

G

\GlobalLetLtxMacro 2, 125, 356, 375, 392, 437, 463

H

\hbox 403, 404, 405

I

\iffalse 409, 411, 501, 505, 508, 514, 518, 521
\ifnum 256, 264
\ifx 136, 174, 191, 198, 207, 222, 227, 312, 322
\IncludeTests 298

L

\LetLtxMacro 2, 122, 352, 370, 388, 400, 403, 406, 410, 427, 454, 488, 504, 517
\Line 4, 9, 24, 27, 33, 45, 46, 53, 54
\llm@AtEnd 97, 98, 113, 114, 243
\llm@CarThree 172, 241
\llm@CarTwo 189, 242
\llm@CheckParams 155, 219
\llm@escapechar 129, 151
\llm@GlobalTemp 200, 210
\llm@LetLtxMacro 145, 149, 153
\llm@macro 222, 237
\llm@ModeLetLtxMacro . . 123, 126, 128
\llm@nil 172, 189, 241, 242
\llm@protected 200, 224, 229
\llm@protectedmacro 227, 239
\llm@testopt 163, 176, 193, 198, 201, 207
\LogTests 299
\loop 251, 262

M

\makeatletter 3, 248, 301
\meaning 11, 155, 310, 318, 401, 402, 404, 405, 407, 408

N

\NeedsTeXFormat 2, 119, 246, 291
\newcommand 4, 7, 15, 22, 25, 35, 56, 122, 125, 128, 366, 380, 476
\NewEmpty 406, 408
\NewHbox 403, 405
\NewRelax 400, 402
\newrobustcmd 48, 51, 452
\nofiles 292

P

\protect 132, 138, 347, 351, 353, 359, 389, 395
\protected 229, 239, 418, 424, 429, 441, 456, 467, 480, 490, 506, 507, 508, 519, 520, 521
\ProvidesPackage 120

R

\RangeCatcodeInvalid 260, 269, 270, 271, 272
\repeat 258, 266
\RequirePackage 44, 283
\reserved@a 131, 136
\RestoreCatcodes . . 249, 252, 253, 284
\robustify 54, 57, 61, 65, 478, 502, 515

S

\SaveA 330, 346, 350, 367, 387, 426, 453, 487, 503, 516
\SavedA 331, 332, 339
\ShowCmdName 7, 16, 17, 18, 19
\ShowCmds 15, 29
\ShowDef 22, 34, 37, 40, 47, 50, 55, 59, 63
\space 10
\strip@prefix 310, 318

T

\Test 268, 286
\TestA 331, 333, 339, 345, 347, 349, 351, 352, 356, 360, 366, 369, 370, 374, 375, 380, 381, 386, 388, 392, 409, 410, 418, 420, 421, 425, 427, 437, 452, 454, 463, 476, 477, 478, 479, 482, 483, 488, 501, 502, 504, 514, 515, 517
\TestB 352, 353, 370, 372, 374, 388, 389, 390,

410, 427, 428, 431, 432, 454, 458, 488, 489, 492, 493, 504, 517
\TestC 356, 359, 375, 379, 381, 392, 395, 396, 429, 441, 456, 467, 480, 490, 508, 521
\TestD 437, 440, 443, 444, 463, 469
\TestDef 303, 347, 351, 353, 359, 368, 371, 378, 389, 390, 395, 396, 411, 428, 440, 455, 466, 477, 479, 489, 505, 518
\TestEquals 320, 338, 339, 340, 354, 361, 374, 381, 412, 435, 447, 461, 472, 486, 496, 509, 522
\the 79, 80, 81, 82, 83, 84, 85, 86, 99, 115, 129, 204, 254
\TMP@EnsureCode . 96, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112
\toks@ 161, 204
\typeout 5, 32, 43, 67, 314, 315

U

\usepackage 295, 297, 414

X

\x 77, 89, 164, 174, 181, 191, 221, 222, 227

Y

\y 170, 174, 187, 191