The letltxmacro package

Heiko Oberdiek[*]
<heiko.oberdiek at googlemail.com>

2016/05/16 v1.5

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

[*] Please report any issues at https://github.com/ho-tex/oberdiek/issues

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 2016/05/16 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

```
⟨*package⟩
```

2.2.1 Catcodes and identification

```
\begingroup\catcode61\catcode48\catcode32=10\relax%
  \catcode13=5 % ^^M
  \endlinechar=13 %
  \catcode123=1 % {
  \catcode125=2 % }
  \catcode64=11 % @
  \def\x{\endgroup
    \expandafter\edef\csname llm@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\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}%
 [2016/05/16 v1.5 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}[2016/05/16]\relax
  \RestoreCatcodes
}
\Test
\csname @@end\endcsname
\end
```

⟨/test1⟩

3.2 Package tests

⟨*test2⟩

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

\usepackage{letltxmacro}[2016/05/16]

\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.

[1] http://ctan.org/pkg/letltxmacro

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 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

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 Catalogue

The following XML file can be used as source for the TeX Catalogue. The elements caption and description are imported from the original XML file from the Catalogue. The name of the XML file in the Catalogue is letltxmacro.xml.

```
⟨*catalogue⟩
<?xml version=’1.0’ encoding=’us-ascii’?>
<!DOCTYPE entry SYSTEM ’catalogue.dtd’>
<entry datestamp=’$Date$’ modifier=’$Author$’ id=’letltxmacro’>
  <name>letltxmacro</name>
  <caption>Let assignment for LaTeX macros.</caption>
  <authorref id=’auth:oberdiek’/>
  <copyright owner=’Heiko Oberdiek’ year=’2008,2010’/>
  <license type=’lppl1.3’/>
  <version number=’1.5’/>
  <description>
    TeX&#x2019;s <tt>\let</tt> assignment does not work for LaTeX macros
    with optional arguments or for macros that are defined
    as robust macros by <tt>\DeclareRobustCommand</tt>.  This package
    defines <tt>\LetLtxMacro</tt> that also takes care of the involved
    internal macros.
    <p/>
    The package is part of the <xref refid=’oberdiek’>oberdiek</xref>
    bundle.
  </description>
  <documentation details=’Package documentation’
      href=’ctan:/macros/latex/contrib/oberdiek/letltxmacro.pdf’/>
  <ctan file=’true’ path=’/macros/latex/contrib/oberdiek/letltxmacro.dtx’/>
  <miktex location=’oberdiek’/>
  <texlive location=’oberdiek’/>
  <install path=’/macros/latex/contrib/oberdiek/oberdiek.tds.zip’/>
</entry>
⟨/catalogue⟩
```

6 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.

[2016/05/16 v1.5]

- Documentation updates.

7 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.