The ltxcmds package

Heiko Oberdiek
<heiko.oberdiek at googlemail.com>

2010/03/09 v1.4

Abstract

The package ltxcmds exports some utility macros from the LaTeX kernel into a separate namespace and also provides them for other formats such as plain-TeX.

Contents

1 Documentation **2**
1.1 Introduction . . . 2
1.2 Numbers . . . 2
1.3 Argument killers . . . 2
1.4 Argument grabbers . . . 2
1.5 List helpers . . . 3
1.6 Tail recursion . . . 3
1.7 Empty macro . . . 3
1.8 Characters . . . 3
1.9 Boolean switch . . . 3
1.10 Command definitions . . . 3
1.11 Stripping . . . 4
1.12 File management . . . 4
1.12.1 File extensions . . . 4
1.12.2 Load check . . . 4
1.12.3 Version date check . . . 4
1.13 Macro additions . . . 5
1.14 Macro \ltx@ifnextchar . . . 5

2 Implementation **5**
2.1 Identification . . . 5
2.2 Numbers . . . 7
2.3 Argument killers . . . 7
2.4 Argument grabbers . . . 7
2.5 List helpers . . . 7
2.6 Tail recursion . . . 8
2.7 Empty macro . . . 8
2.8 Characters . . . 8
2.9 Boolean switch . . . 8
2.10 Command definitions . . . 9
2.11 Stripping . . . 9
2.12 File management . . . 10
2.12.1 File extensions . . . 10
2.12.2 Load check . . . 10
2.12.3 Version date check . . . 10
2.13 Macro additions . . . 11
2.14 Macro \ltx@ifnextchar . . . 12

3 Test **13**
3.1 Catcode checks for loading 13

4 Installation **14**
4.1 Download 14
4.2 Bundle installation 14
4.3 Package installation 15
4.4 Refresh file name databases 15
4.5 Some details for the interested 15

5 History **16**
[2009/08/05 v1.0] 16
[2009/12/12 v1.1] 16
[2010/01/28 v1.2] 16
[2010/03/01 v1.3] 16
[2010/03/09 v1.4] 16

6 Index **16**

1 Documentation

1.1 Introduction

Many of my packages also support other formats such as plain-TEX. Because I am rather familiar with the utility macros from LATEX's kernel (e.g. `\@gobble`, `\@firstoftwo`), I found myself rewriting them again and again, because they are lacking in plain-TEX.

Therefore this package provides often used macros and similar ones with the name prefix `\ltx@`. This avoids also faulty redefinitions. I remember an example where a package redefined `\@firstoftwo` with forgetting `\long`.

1.2 Numbers

```
\ltx@zero
\ltx@one
\ltx@two
\ltx@cclv
```

These commands are numbers 0, 1, 2 and 255. They are not digits and a space is not gobbled afterwards.

1.3 Argument killers

`\ltx@gobble` {⟨*1*⟩}	→
`\ltx@gobbletwo` {⟨*1*⟩} {⟨*2*⟩}	→
`\ltx@gobblethree` {⟨*1*⟩} {⟨*2*⟩} {⟨*3*⟩}	→
`\ltx@gobblefour` {⟨*1*⟩} {⟨*2*⟩} {⟨*3*⟩} {⟨*4*⟩}	→

1.4 Argument grabbers

`\ltx@firstofone` {⟨*1*⟩}	→	⟨*1*⟩
`\ltx@firstoftwo` {⟨*1*⟩} {⟨*2*⟩}	→	⟨*1*⟩
`\ltx@secondoftwo` {⟨*1*⟩} {⟨*2*⟩}	→	⟨*2*⟩

1.5 List helpers

`\ltx@car` {⟨*1*⟩} ... `\@nil`	→	⟨*1*⟩
`\ltx@cdr` {⟨*1*⟩} ... `\@nil`	→	...

1.6 Tail recursion

`\ltx@ReturnAfterFi` {⟨*1*⟩} `\fi`	→	`\fi` ⟨*1*⟩
`\ltx@ReturnAfterElseFi` {⟨*1*⟩} `\else` {⟨*2*⟩} `\fi`	→	`\fi` ⟨*1*⟩

1.7 Empty macro

`\ltx@empty`	→

1.8 Characters

`\ltx@space`
`\ltx@percentchar`
`\ltx@backslashchar`

1.9 Boolean switch

`\ltx@newif` {⟨*cmd*⟩}

`\ltx@newif` defines a new boolean switch ⟨*cmd*⟩ like `\newif`. Unlike plain-TEX's `\newif`, `\ltx@newif` is not `\outer`. The command ⟨*cmd*⟩ must start with the two characters `if`.

1.10 Command definitions

`\ltx@ifundefined` {⟨*cmd*⟩} {⟨*yes*⟩} {⟨*no*⟩}

If ε-TEX is available, `\ifcsname` is used that does not have the side effect of defining undefined commands with meaning of `\relax`. This command is always expandable. Change in version 1.1: Also the meaning `\relax` is always considered "undefined".

`\ltx@IfUndefined` {⟨*cmd*⟩} {⟨*yes*⟩} {⟨*no*⟩}

If ε-TEX is available, `\ifcsname` is used that does not have the side effect of defining undefined commands with meaning of `\relax`. Also it always checks for the meaning of `\relax` and considers this as undefined. This macro is not expandable without ε-TEX.

`\ltx@LocalExpandAfter`

It expands the token after the next token but in a local context. That is the difference to `\expandafter`. The local context discards the side effect of `\csname`

and let the command undefined after the expansion step.

1.11 Stripping

`\ltx@RemovePrefix`
`\ltx@StripPrefix`

All tokens up to and including the next available character '>' are thrown away. Usually it is used to strip the first part of the output of the commands `\meaning` or `\pdflastmatch`. Macro `\ltx@RemovePrefix` has the same meaning as LaTeX's `\strip@prefix`, whereas macro `\ltx@StripPrefix` expands the next token once before stripping the prefix.

1.12 File management

All macros in this section are expandable like the counterparts of the LaTeX kernel. Also they can be used after the preamble.

1.12.1 File extensions

`\ltx@clsextension`
`\ltx@pkgextension`

If `\@clsextension`/`\@pkgextension` exists then `\ltx@clsextension`/`\ltx@pkgextension` returns this macro, otherwise the result is `cls`/`sty`.

1.12.2 Load check

`\ltx@ifclassloaded` {⟨*class*⟩} {⟨*yes*⟩} {⟨*no*⟩}
`\ltx@ifpackageloaded` {⟨*package*⟩} {⟨*yes*⟩} {⟨*no*⟩}

If the ⟨*class*⟩/⟨*package*⟩ are loaded the macros `\ltx@ifclassloaded`/`\ltx@ifpackageloaded` call the ⟨*yes*⟩ argument. Otherwise ⟨*no*⟩ is executed. Both ⟨*class*⟩ and ⟨*package*⟩ are specified without extension.

`\ltx@iffileloaded` {⟨*file*⟩} {⟨*yes*⟩} {⟨*no*⟩}

If LaTeX's `\ProvidesFile` macro was called before using ⟨*file*⟩ as argument, then `\ltx@iffileloaded` calls ⟨*yes*⟩, otherwise ⟨*no*⟩. Therefore it is possible that the ⟨*file*⟩ is loaded, but ⟨*no*⟩ is executed because of a missing `\ProvidesFile`. The LaTeX kernel does not have a counterpart of `\ltx@iffileloaded`.

Note that the file name used in `\ProvidesFile` and `\ltx@iffileloaded` must match. For example, if TeX's default extension `..tex` was given in the first command, then it must also specified in the latter command and vice versa.

1.12.3 Version date check

`\ltx@ifclasslater` {⟨*class*⟩} {⟨*date*⟩} {⟨*yes*⟩} {⟨*no*⟩}
`\ltx@ifpackagelater` {⟨*package*⟩} {⟨*date*⟩} {⟨*yes*⟩} {⟨*no*⟩}
`\ltx@iffilelater` {⟨*file*⟩} {⟨*date*⟩} {⟨*yes*⟩} {⟨*no*⟩}

If a `\ProvidesClass`/`\ProvidesPackage`/`\ProvidesFile` command with exact the same class/package/file was executed before with an optional argument that starts with a LaTeX version date, then this version date is compared with the

argument ⟨*date*⟩. If they are equal or if the version date is the later date, then ⟨*yes*⟩ is called. In all other cases ⟨*no*⟩ is executed.

A LaTeX date has the format `YYYY/MM/DD` with `YYYY` as year with four digits, `MM` as month with two digits and `DD` as day with two digits. If pdfTeX's `\pdfmatch` is available, then it is used to detect the version date, to reject invalid date formats and to reject some invalid dates. Dates before 1994/01/01 are always invalid, because version dates are introduced with LaTeX 2_ε in 1994.

1.13 Macro additions

`\ltx@GlobalAppendToMacro` {⟨*cmd*⟩} {⟨*addition*⟩}
`\ltx@LocalAppendToMacro` {⟨*cmd*⟩} {⟨*addition*⟩}

The ⟨*addition*⟩ is appended to the parameterless macro ⟨*cmd*⟩. If ⟨*cmd*⟩ is undefined or has the meaning `\relax`, then it will be initialized as empty macro before.

1.14 Macro `\ltx@ifnextchar`

`\ltx@ifnextchar` {⟨*char*⟩} {⟨*yes*⟩} {⟨*no*⟩}

If next character is ⟨*char*⟩ then ⟨*yes*⟩ is called, otherwise ⟨*no*⟩. The character is not removed.

2 Implementation

2.1 Identification

```
1 ⟨*package⟩
```

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

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

Package identification:

```
\begingroup
  \catcode35 6 % #
  \catcode40 12 % (
  \catcode41 12 % )
  \catcode44 12 % ,
  \catcode45 12 % -
  \catcode46 12 % .
  \catcode47 12 % /
  \catcode58 12 % :
  \catcode64 11 % @
  \catcode91 12 % [
  \catcode93 12 % ]
  \catcode123 1 % {
  \catcode125 2 % }
  \expandafter\ifx\csname ProvidesPackage\endcsname\relax
    \def\x#1#2#3[#4]{\endgroup
      \immediate\write-1{Package: #3 #4}%
      \xdef#1{#4}%
    }%
  \else
    \def\x#1#2[#3]{\endgroup
      #2[{#3}]%
      \ifx#1\@undefined
        \xdef#1{#3}%
      \fi
      \ifx#1\relax
        \xdef#1{#3}%
      \fi
    }%
  \fi
\expandafter\x\csname ver@ltxcmds.sty\endcsname
\ProvidesPackage{ltxcmds}%
  [2010/03/09 v1.4 LaTeX kernel commands for general use (HO)]
\begingroup
  \catcode123 1 % {
  \catcode125 2 % }
  \def\x{\endgroup
    \expandafter\edef\csname LTXcmds@AtEnd\endcsname{%
      \catcode35 \the\catcode35\relax
      \catcode64 \the\catcode64\relax
      \catcode123 \the\catcode123\relax
      \catcode125 \the\catcode125\relax
    }%
  }%
\x
\catcode35 6 % #
\catcode64 11 % @
\catcode123 1 % {
\catcode125 2 % }
\def\TMP@EnsureCode#1#2{%
  \edef\LTXcmds@AtEnd{%
    \LTXcmds@AtEnd
    \catcode#1 \the\catcode#1\relax
  }%
  \catcode#1 #2\relax
}
\TMP@EnsureCode{40}{12}% (
\TMP@EnsureCode{41}{12}% )
\TMP@EnsureCode{45}{12}% -
\TMP@EnsureCode{46}{12}% .
\TMP@EnsureCode{47}{12}% /
```

```
\TMP@EnsureCode{60}{12}% <
\TMP@EnsureCode{61}{12}% =
\TMP@EnsureCode{62}{12}% >
\TMP@EnsureCode{91}{12}% [
\TMP@EnsureCode{96}{12}% `
\TMP@EnsureCode{93}{12}% ]
\TMP@EnsureCode{94}{12}% ^ (superscript) (!)
\TMP@EnsureCode{124}{12}% |
```

2.2 Numbers

\ltx@zero

```
\chardef\ltx@zero=0 %
```

\ltx@one

```
\chardef\ltx@one=1 %
```

\ltx@two

```
\chardef\ltx@two=2 %
```

\ltx@cclv

```
\chardef\ltx@cclv=255 %
```

2.3 Argument killers

\ltx@gobble

```
\long\def\ltx@gobble#1{}
```

\ltx@gobbletwo

```
\long\def\ltx@gobbletwo#1#2{}
```

\ltx@gobblethree

```
\long\def\ltx@gobblethree#1#2#3{}
```

\ltx@gobblefour

```
\long\def\ltx@gobblefour#1#2#3#4{}
```

2.4 Argument grabbers

\ltx@firstofone

```
\long\def\ltx@firstofone#1{#1}
```

\ltx@firstoftwo

```
\long\def\ltx@firstoftwo#1#2{#1}
```

\ltx@secondoftwo

```
\long\def\ltx@secondoftwo#1#2{#2}
```

2.5 List helpers

\ltx@car

```
\long\def\ltx@car#1#2\@nil{#1}
```

\ltx@cdr

```
\long\def\ltx@cdr#1#2\@nil{#2}
```

2.6 Tail recursion

\ltx@ReturnAfterFi

```
\long\def\ltx@ReturnAfterFi#1\fi{#1}
```

\ltx@ReturnAfterElseFi

```
\long\def\ltx@ReturnAfterFi#1\else#2\fi{#1}
```

2.7 Empty macro

\ltx@empty

```
\def\ltx@empty{}
```

2.8 Characters

\ltx@space

```
\def\ltx@space{ }
```

\ltx@percentchar

```
\begingroup
  \lccode`0=`\%\relax
\lowercase{\endgroup
  \def\ltx@percentchar{0}%
}
```

\ltx@backslashchar

```
\begingroup
  \lccode`0=`\\\relax
\lowercase{\endgroup
  \def\ltx@backslashchar{0}%
}
```

2.9 Boolean switch

\ltx@newif

```
\def\ltx@newif#1{%
  \begingroup
    \escapechar=-1 %
  \expandafter\endgroup
  \expandafter\LTXcmds@newif\string#1\@nil
}
```

\LTXcmds@newif

```
\begingroup
  \escapechar=-1 %
\expandafter\endgroup
\expandafter\def\expandafter\LTXcmds@newif\string\if#1\@nil{%
  \expandafter\edef\csname#1true\endcsname{%
    \let
    \expandafter\noexpand\csname if#1\endcsname
    \noexpand\iftrue
  }%
  \expandafter\edef\csname#1false\endcsname{%
    \let
    \expandafter\noexpand\csname if#1\endcsname
    \noexpand\iffalse
  }%
  \csname#1false\endcsname
}
```

2.10 Command definitions

\ltx@LocalExpandAfter

```
\def\ltx@LocalExpandAfter{%
  \begingroup
    \expandafter\expandafter\expandafter
  \endgroup
  \expandafter
}

\ltx@LocalExpandAfter
\ifx\csname ifcsname\endcsname\relax
```

\ltx@ifundefined

```
  \def\ltx@ifundefined#1{%
    \expandafter\ifx\csname #1\endcsname\relax
      \expandafter\ltx@firstoftwo
    \else
      \expandafter\ltx@secondoftwo
    \fi
  }%
```

\ltx@IfUndefined

```
  \def\ltx@IfUndefined#1{%
    \begingroup\expandafter\expandafter\expandafter\endgroup
    \expandafter\ifx\csname #1\endcsname\relax
      \expandafter\ltx@firstoftwo
    \else
      \expandafter\ltx@secondoftwo
    \fi
  }%

  \expandafter\ltx@gobble
\else
  \expandafter\ltx@firstofone
\fi
{%
```

\ltx@ifundefined

```
  \def\ltx@ifundefined#1{%
    \ifcsname #1\endcsname
      \expandafter\ifx\csname #1\endcsname\relax
        \expandafter\expandafter\expandafter\ltx@firstoftwo
      \else
        \expandafter\expandafter\expandafter\ltx@secondoftwo
      \fi
    \else
      \expandafter\ltx@firstoftwo
    \fi
  }%
```

\ltx@IfUndefined

```
  \let\ltx@IfUndefined\ltx@ifundefined

}
```

2.11 Stripping

\ltx@RemovePrefix

```
\def\ltx@RemovePrefix#1>{}
```

\ltx@StripPrefix

```
\def\ltx@StripPrefix{%
  \expandafter\ltx@RemovePrefix
}
```

2.12 File management

2.12.1 File extensions

\ltx@clsextension

```
\def\ltx@clsextension{cls}
```

\ltx@pkgextension

```
\def\ltx@pkgextension{sty}
```

2.12.2 Load check

\ltx@iffileloaded

```
\def\ltx@iffileloaded#1{%
  \ltx@ifundefined{ver@#1}\ltx@secondoftwo\ltx@firstoftwo
}
```

\ltx@ifclassloaded

```
\def\ltx@ifclassloaded#1{%
  \ltx@iffileloaded{#1.\ltx@clsextension}%
}
```

\ltx@ifpackageloaded

```
\def\ltx@ifpackageloaded#1{%
  \ltx@iffileloaded{#1.\ltx@pkgextension}%
}
```

2.12.3 Version date check

\ltx@iffilelater

```
\def\ltx@iffilelater#1#2{%
  \ltx@iffileloaded{#1}{%
    \expandafter\LTXcmds@IfLater\expandafter{%
      \number
      \expandafter\expandafter\expandafter\LTXcmds@ParseVersion
      \expandafter\expandafter\expandafter{%
        \csname ver@#1\endcsname
      }%
    \expandafter}\expandafter{%
      \number
      \expandafter\LTXcmds@ParseVersion\expandafter{#2}%
    }%
  }\ltx@secondoftwo
}
```

\LTXcmds@IfLater

```
\def\LTXcmds@IfLater#1#2{%
  \ifcase 0%
      \ifnum#1<19940101 %
      \else
        \ifnum#2<19940101 %
        \else
          \ifnum#2>#1 %
          \else
            1%
```

```
          \fi
        \fi
      \fi
      \ltx@space
    \expandafter\ltx@secondoftwo
  \else
    \expandafter\ltx@firstoftwo
  \fi
}
```

\ltx@ifclasslater

```
\def\ltx@ifclasslater#1{%
  \ltx@ifclasslater{#1.\ltx@clsextension}%
}
```

\ltx@ifpackagelater

```
\def\ltx@ifpackagelater#1{%
  \ltx@iffilelater{#1.\ltx@pkgextension}%
}
\ltx@IfUndefined{pdfmatch}{%
```

\LTXcmds@ParseVersion

```
  \def\LTXcmds@ParseVersion#1{%
    \LTXcmds@@ParseVersion#10000/00/00\@nil
  }%
```

\LTXcmds@@ParseVersion

```
  \def\LTXcmds@@ParseVersion#1#2#3#4/#5#6/#7#8#9\@nil{%
    #1#2#3#4#5#6#7#8%
  }%
}{%
```

\LTXcmds@ParseVersion

```
  \def\LTXcmds@ParseVersion#1{%
    \ifnum\pdfmatch{%
      ^%
      (199[4-9]|[2-9][0-9][0-9][0-9])/%
      (0[1-9]|1[0-2])/%
      (0[1-9]|[1-2][0-9]|3[0-1])%
    }{#1}=1 %
      \ltx@StripPrefix\pdflastmatch1 %
      \ltx@StripPrefix\pdflastmatch2 %
      \ltx@StripPrefix\pdflastmatch3 %
    \else
      0%
    \fi
  }%
}
```

2.13 Macro additions

\ltx@GlobalAppendToMacro

```
\def\ltx@GlobalAppendToMacro#1#2{%
  \ifx\ltx@undefined#1%
    \let#1\ltx@empty
  \else
    \ifx\relax#1%
      \let#1\ltx@empty
    \fi
```

```
  \fi
  \begingroup
    \toks0\expandafter{#1#2}%
    \xdef#1{\the\toks0}%
  \endgroup
}
```

`\ltx@LocalAppendToMacro`

```
\def\ltx@LocalAppendToMacro#1#2{%
  \global\let\LTXcmds@gtemp#1%
  \ifx\ltx@undefined\LTXcmds@gtemp
    \global\let\LTXcmds@gtemp\ltx@empty
  \else
    \ifx\relax\LTXcmds@gtemp
      \global\letLTXcmds@gtemp\ltx@empty
    \fi
  \fi
  \begingroup
    \toks0\expandafter{\LTXcmds@gtemp#2}%
    \xdef\LTXcmds@gtemp{\the\toks0}%
  \endgroup
  \let#1\LTXcmds@gtemp
}
```

2.14 Macro `\ltx@ifnextchar`

`\ltx@ifnextchar`

```
\long\def\ltx@ifnextchar#1#2#3{%
  \begingroup
  \let\LTXcmds@CharToken= #1\relax
  \toks\ltx@zero{#2}%
  \toks\ltx@two{#3}%
  \futurelet\LTXcmds@LetToken\LTXcmds@ifnextchar
}
```

`\LTXcmds@ifnextchar`

```
\def\LTXcmds@ifnextchar{%
  \ifx\LTXcmds@LetToken\LTXcmds@CharToken
    \expandafter\endgroup\the\toks\expandafter\ltx@zero
  \else
    \ifx\LTXcmds@LetToken\LTXcmds@SpaceToken
      \expandafter\expandafter\expandafter\LTXcmds@@ifnextchar
    \else
      \expandafter\endgroup\the\toks
      \expandafter\expandafter\expandafter\ltx@two
    \fi
  \fi
}
```

`\LTXcmds@@ifnextchar`

```
\begingroup
  \def\x#1{\endgroup
    \def\LTXcmds@@ifnextchar#1{%
      \futurelet\LTXcmds@LetToken\LTXcmds@ifnextchar
    }%
  }%
\x{ }
```

`\LTXcmds@SpaceToken`

```
\begingroup
  \def\x#1{\endgroup
```

```
    \let\LTXcmds@SpaceToken= #1%
  }%
\x{ }

\LTXcmds@AtEnd
⟨/package⟩
```

3 Test

3.1 Catcode checks for loading

```
⟨*test1⟩

\catcode`\{=1 %
\catcode`\}=2 %
\catcode`\#=6 %
\catcode`\@=11 %
\expandafter\ifx\csname count@\endcsname\relax
  \countdef\count@=255 %
\fi
\expandafter\ifx\csname @gobble\endcsname\relax
  \long\def\@gobble#1{}%
\fi
\expandafter\ifx\csname @firstofone\endcsname\relax
  \long\def\@firstofone#1{#1}%
\fi
\expandafter\ifx\csname loop\endcsname\relax
  \expandafter\@firstofone
\else
  \expandafter\@gobble
\fi
{%
  \def\loop#1\repeat{%
    \def\body{#1}%
    \iterate
  }%
  \def\iterate{%
    \body
      \let\next\iterate
    \else
      \let\next\relax
    \fi
    \next
  }%
  \let\repeat=\fi
}%
\def\RestoreCatcodes{}
\count@=0 %
\loop
  \edef\RestoreCatcodes{%
    \RestoreCatcodes
    \catcode\the\count@=\the\catcode\count@\relax
  }%
\ifnum\count@<255 %
  \advance\count@ 1 %
\repeat

\def\RangeCatcodeInvalid#1#2{%
  \count@=#1\relax
  \loop
    \catcode\count@=15 %
  \ifnum\count@<#2\relax
    \advance\count@ 1 %
```

```
  \repeat
}
\expandafter\ifx\csname LoadCommand\endcsname\relax
  \def\LoadCommand{\input ltxcmds.sty\relax}%
\fi
\def\Test{%
  \RangeCatcodeInvalid{0}{47}%
  \RangeCatcodeInvalid{58}{64}%
  \RangeCatcodeInvalid{91}{96}%
  \RangeCatcodeInvalid{123}{255}%
  \catcode`\@=12 %
  \catcode`\\=0 %
  \catcode`\{=1 %
  \catcode`\}=2 %
  \catcode`\#=6 %
  \catcode`\[=12 %
  \catcode`\]=12 %
  \catcode`\%=14 %
  \catcode`\ =10 %
  \catcode13=5 %
  \LoadCommand
  \RestoreCatcodes
}
\Test
\csname @@end\endcsname
\end
```

⟨/test1⟩

4 Installation

4.1 Download

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

CTAN:macros/latex/contrib/oberdiek/ltxcmds.dtx The source file.

CTAN:macros/latex/contrib/oberdiek/ltxcmds.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
```

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/
```

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

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 ltxcmds.dtx
```

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

```
ltxcmds.sty            → tex/generic/oberdiek/ltxcmds.sty
ltxcmds.pdf            → doc/latex/oberdiek/ltxcmds.pdf
test/ltxcmds-test1.tex → doc/latex/oberdiek/test/ltxcmds-test1.tex
ltxcmds.dtx            → source/latex/oberdiek/ltxcmds.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 ltxcmds.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{ltxcmds.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 ltxcmds.dtx
makeindex -s gind.ist ltxcmds.idx
pdflatex ltxcmds.dtx
makeindex -s gind.ist ltxcmds.idx
pdflatex ltxcmds.dtx
```

5 History

[2009/08/05 v1.0]

- First version.

[2009/12/12 v1.1]

- Short title shortened.
- `\ltx@IfUndefined` added.

[2010/01/28 v1.2]

- `\ltx@RemovePrefix` and `\ltx@StripPrefix` added.
- `\ltx@ifclassloaded`, `\ltx@ifpackageloaded`, `\ltx@iffileloaded`, `\ltx@ifclasslater`, `\ltx@ifpackagelater`, `\ltx@iffilelater`, `\ltx@clsextension`, `\ltx@pkgextension` added.
- `\ltx@GlobalAppendToMacro`, `\ltx@LocalAppendToMacro` added.

[2010/03/01 v1.3]

- `\ltx@newif` added.
- `\ltx@ifnextchar` added.
- Numbers `\ltx@zero`, `\ltx@one`, `\ltx@two`, `\ltx@cclv` added.

[2010/03/09 v1.4]

- `\ltx@pkgextension` and `\ltx@clsextension` are hardcoded to avoid trouble with `\@onlypreamble`.

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

\# 329, 391
\% 117, 394
\@ 330, 387
\@firstofone 338, 341
\@gobble 335, 343
\@nil 110, 111, 130, 135, 244, 246
\@undefined 52
\[. 392
\\ 122, 388
\{ 327, 389
\} 328, 390
\] 393
\␣ 395

A

\advance 368, 376
\aftergroup 26

B

\body 347, 351

C

\catcode 3, 4, 5, 6, 7, 8, 9, 17, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 64, 65, 68, 69, 70, 71, 75, 76, 77, 78, 82, 84, 327, 328, 329, 330, 365, 374, 387, 388, 389, 390, 391, 392, 393, 394, 395, 396
\chardef 99, 100, 101, 102
\count@ 332, 361, 365, 367, 368, 372, 374, 375, 376
\countdef 332
\csname 10, 18, 44, 60, 67, 136, 138, 141, 143, 146, 155, 157, 165, 178, 210, 331, 334, 337, 340, 379, 401

E

\empty 13, 14

\end 402
\endcsname 10, 18, 44, 60, 67, 136, 138, 141, 143, 146, 155, 157, 165, 177, 178, 210, 331, 334, 337, 340, 379, 401
\endinput 26
\escapechar 128, 133

F

\futurelet 298, 315

I

\if 135
\ifcase 219
\ifcsname 177
\iffalse 144
\ifnum 220, 222, 224, 251, 367, 375
\iftrue 139
\ifx . 11, 14, 18, 44, 52, 55, 155, 157, 165, 178, 266, 269, 280, 283, 301, 304, 331, 334, 337, 340, 379
\immediate 20, 46
\input 380
\iterate 348, 350, 352

L

\lccode 117, 122
\letLTXcmds@gtemp 284
\LoadCommand 380, 397
\loop 346, 362, 373
\lowercase 118, 123
\ltx@backslashchar 121
\ltx@car *3*, 110
\ltx@cclv 102
\ltx@cdr 111
\ltx@clsextension . . . *4*, 193, 199, 237
\ltx@empty . . *3*, 114, 267, 270, 281, 284
\ltx@firstofone *2*, 107, 173
\ltx@firstoftwo 108, 158, 166, 179, 184, 196, 233
\ltx@GlobalAppendToMacro *5*, 265
\ltx@gobble *2*, 103, 171
\ltx@gobblefour 106
\ltx@gobblethree 105
\ltx@gobbletwo 104
\ltx@ifclasslater *4*, 236
\ltx@ifclassloaded *4*, 198
\ltx@iffilelater 204, 240
\ltx@iffileloaded *4*, 195, 199, 202, 205
\ltx@ifnextchar *5*, 293
\ltx@ifpackagelater 239
\ltx@ifpackageloaded 201
\ltx@IfUndefined *3*, 163, 187, 242
\ltx@ifundefined *3*, 156, 176, 187, 196
\ltx@LocalAppendToMacro 278
\ltx@LocalExpandAfter . . . *3*, 148, 154
\ltx@newif *3*, 126
\ltx@one 100
\ltx@percentchar 116
\ltx@pkgextension 194, 202, 240
\ltx@RemovePrefix *4*, 189, 191
\ltx@ReturnAfterElseFi 113
\ltx@ReturnAfterFi *3*, 112, 113
\ltx@secondoftwo 109, 160, 168, 181, 196, 216, 231
\ltx@space *3*, 115, 230
\ltx@StripPrefix . . 190, 257, 258, 259
\ltx@two 101, 297, 308
\ltx@undefined 266, 280
\ltx@zero *2*, 99, 296, 302
\LTXcmds@@ifnextchar 305, 312
\LTXcmds@@ParseVersion 244, 246
\LTXcmds@AtEnd 80, 81, 324
\LTXcmds@CharToken 295, 301
\LTXcmds@gtemp 279, 280, 281, 283, 288, 289, 291
\LTXcmds@IfLater 206, 218
\LTXcmds@ifnextchar . . . 298, 300, 315
\LTXcmds@LetToken . 298, 301, 304, 315
\LTXcmds@newif 130, 132
\LTXcmds@ParseVersion 208, 214, 243, 250
\LTXcmds@SpaceToken 304, 319

N

\next 352, 354, 356
\number 207, 213

P

\PackageInfo 23
\pdflastmatch 257, 258, 259
\pdfmatch 251
\ProvidesPackage 15, 61

R

\RangeCatcodeInvalid 371, 383, 384, 385, 386
\repeat 346, 358, 369, 377
\RestoreCatcodes . . 360, 363, 364, 398

T

\Test 382, 400
\the 68, 69, 70, 71, 82, 275, 289, 302, 307, 365
\TMP@EnsureCode . . . 79, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98
\toks 274, 275, 288, 289, 296, 297, 302, 307

W

\write 20, 46

X

\x 10, 11, 14, 19, 23, 25, 45, 50, 60, 66, 74, 313, 318, 320, 323