The \texttt{ltxcmds} package

Heiko Oberdiek\textsuperscript{*}

2019/12/15 v1.24

Abstract

The package \texttt{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 3
  1.1 Introduction 3
  1.2 Numbers 3
  1.3 Scratch registers 3
  1.4 Argument killers 3
  1.5 Argument grabbers 4
  1.6 List helpers 5
  1.7 Tail recursion 5
  1.8 Empty macro 6
  1.9 Characters 6
  1.10 Boolean switch 6
  1.11 Command definitions 6
  1.12 Stripping 7
  1.13 File management 7
    1.13.1 File extensions 7
    1.13.2 Load check 7
    1.13.3 Version date check 8
  1.14 Macro additions 8
  1.15 Next character detection 8
  1.16 \texttt{\textbackslash ltx\@leavevmode, \textbackslash ltx\@mbox} 9
  1.17 Expandable test for emptiness 9
  1.18 Stripping spaces 9
  1.19 Check for emptiness of boxes 10

2 Implementation 10
  2.1 Identification 10
  2.2 Numbers 12
  2.3 Scratch registers 12
  2.4 Argument killers 14
  2.5 Argument grabbers 15
  2.6 List helpers 15
  2.7 Tail recursion 17

*Please report any issues at \url{https://github.com/ho-tex/ltxcmds/issues}
2.8 Empty macro ........................................... 17
2.9 Characters ............................................. 17
2.10 Boolean switch ........................................ 18
2.11 Command definitions .................................. 19
2.12 Stripping .............................................. 20
2.13 File management ....................................... 20
  2.13.1 File extensions ................................... 20
  2.13.2 Load check ....................................... 20
  2.13.3 Version date check ............................... 21
2.14 Macro additions ...................................... 22
2.15 Next character detection ............................. 23
2.16 \texttt{\textbackslash{}leavemode}, \texttt{\textbackslash{}mbox} ... 24
2.17 Help macros .......................................... 25
2.18 Expandable test for emptiness ....................... 25
  2.18.1 Vanilla \TeX ........................................ 25
  2.18.2 With \texttt{\textbackslash{}detokenize} ............ 26
  2.18.3 \texttt{\textbackslash{}leafflank} .................... 26
2.19 \texttt{\textbackslash{}zapspace} ......................... 27
2.20 \texttt{\textbackslash{}ffBoxEmpty} .................... 27

3 Installation ............................................. 28
  3.1 Download .............................................. 28
  3.2 Bundle installation .................................. 29
  3.3 Package installation ................................ 29
  3.4 Refresh file name databases ......................... 29
  3.5 Some details for the interested ................... 29

4 References ............................................. 30

5 History .................................................. 30
   [2009/08/05 v1.0] ....................................... 30
   [2009/12/12 v1.1] ....................................... 30
   [2010/01/28 v1.2] ....................................... 31
   [2010/03/01 v1.3] ....................................... 31
   [2010/03/09 v1.4] ....................................... 31
   [2010/04/08 v1.5] ....................................... 31
   [2010/04/16 v1.6] ....................................... 31
   [2010/04/26 v1.7] ....................................... 31
   [2010/09/11 v1.8] ....................................... 31
   [2010/10/25 v1.9] ....................................... 31
   [2010/10/31 v1.10] ...................................... 31
   [2010/11/12 v1.11] ...................................... 31
   [2010/12/02 v1.12] ...................................... 32
   [2010/12/04 v1.13] ...................................... 32
   [2010/12/07 v1.14] ...................................... 32
   [2010/12/12 v1.15] ...................................... 32
   [2011/02/04 v1.16] ...................................... 32
   [2011/02/05 v1.17] ...................................... 32
   [2011/03/16 v1.18] ...................................... 32
   [2011/04/14 v1.19] ...................................... 32
   [2011/04/18 v1.20] ...................................... 32
   [2011/08/22 v1.21] ...................................... 32
   [2011/11/09 v1.22] ...................................... 33
   [2016/05/16 v1.23] ...................................... 33
1 Documentation

1.1 Introduction

Many of my packages also support other formats such as plain-T\TeX. Because I am rather familiar with the utility macros from L\TeX’s kernel (e.g. \gobble, \firstoftwo), I found myself rewriting them again and again, because they are lacking in plain-T\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  | → 0 |
| \ltx@one   | → 1 |
| \ltx@two   | → 2 |
| \ltx@cclv  | → 255 |
| \ltx@minusone | → -1 |

These commands are numbers 0, 1, 2, 255 and -1. They are not digits and a space is not gobbled afterwards. Macro \ltx@minusone is available since version 2010/12/12 v1.15.

1.3 Scratch registers

Following the conventions of plain T\TeX and L\TeX the first ten registers are free to use. Even numbered registers are for local, odd numbered for global use.

\ltx@(Loc,Glob)(Toks,Dimen,Skip)(A,B,C,D,E)

The name consists of the prefix \ltx@, then Loc or Glob for local or global usage follows. The register type is given by Toks for token register, Dimen for dimen register and Skip for skip register. As last part the registers are numbered from A to E. Example: \ltx@LocToksA.

Since 2011/04/14 v1.19.
1.4 Argument killers

\[ \texttt{\textbackslash \text{gobble \{1\}}} \rightarrow \]
\[ \texttt{\textbackslash \text{gobbletwo \{1\} \{2\}}} \rightarrow \]
\[ \texttt{\textbackslash \text{gobblethree \{1\} \{2\} \{3\}}} \rightarrow \]
\[ \texttt{\textbackslash \text{gobblefour \{1\} \{2\} \{3\} \{4\}}} \rightarrow \]

\[ \texttt{\textbackslash \text{GobbleNum \{num\} \{1\} \{2\} \ldots \{\langle num\rangle\}}} \rightarrow \]

The first argument \( \langle num\rangle \) of macro \texttt{\textbackslash \text{GobbleNum}} specifies, how many following arguments are eaten. Macro \texttt{\textbackslash \text{GobbleNum}} is expandable in exact two expansion steps.

1.5 Argument grabbers

\[ \texttt{\textbackslash \text{firstofone \{1\}}} \rightarrow \{1\} \]
\[ \texttt{\textbackslash \text{firstoftwo \{1\} \{2\}}} \rightarrow \{1\} \]
\[ \texttt{\textbackslash \text{secondoftwo \{1\} \{2\}}} \rightarrow \{2\} \]
\[ \texttt{\textbackslash \text{firstofthree \{1\} \{2\} \{3\}}} \rightarrow \{1\} \]
\[ \texttt{\textbackslash \text{secondofthree \{1\} \{2\} \{3\}}} \rightarrow \{2\} \]
\[ \texttt{\textbackslash \text{thirdofthree \{1\} \{2\} \{3\}}} \rightarrow \{3\} \]
\[ \texttt{\textbackslash \text{firstoffour \{1\} \{2\} \{3\} \{4\}}} \rightarrow \{1\} \]
\[ \texttt{\textbackslash \text{secondoffour \{1\} \{2\} \{3\} \{4\}}} \rightarrow \{2\} \]
\[ \texttt{\textbackslash \text{thirdoffour \{1\} \{2\} \{3\} \{4\}}} \rightarrow \{3\} \]
\[ \texttt{\textbackslash \text{fourthoffour \{1\} \{2\} \{3\} \{4\}}} \rightarrow \{4\} \]

Macros \texttt{\textbackslash \text{firstofthree}}, \texttt{\textbackslash \text{secondofthree}} and \texttt{\textbackslash \text{thirdofthree}} were added in version 2010/11/12 v1.11. Macros \texttt{\textbackslash \text{firstoffour}}, ..., \texttt{\textbackslash \text{fourthoffour}} were added in version 2011/02/04 v1.16.
1.6 List helpers

\begin{verbatim}
\ltx@carzero \ldots \@nil \rightarrow \ldots
\ltx@cdrzero \ldots \@nil \rightarrow \ldots
\ltx@car {⟨1⟩} \ldots \@nil \rightarrow ⟨1⟩
\ltx@caddr {⟨1⟩} \ldots \@nil \rightarrow \ldots
\ltx@cartwo {⟨1⟩} {⟨2⟩} \ldots \@nil \rightarrow ⟨1⟩⟨2⟩
\ltx@carsecond {⟨1⟩} {⟨2⟩} \ldots \@nil \rightarrow ⟨2⟩
\ltx@cdrtwo {⟨1⟩} {⟨2⟩} \ldots \@nil \rightarrow \ldots
\ltx@carthree {⟨1⟩} {⟨2⟩} {⟨3⟩} \ldots \@nil \rightarrow ⟨1⟩⟨2⟩⟨3⟩
\ltx@carthird {⟨1⟩} {⟨2⟩} {⟨3⟩} \ldots \@nil \rightarrow ⟨3⟩
\ltx@cdrthree {⟨1⟩} {⟨2⟩} {⟨3⟩} \ldots \@nil \rightarrow \ldots
\ltx@carfour {⟨1⟩} {⟨2⟩} {⟨3⟩} {⟨4⟩} \ldots \@nil \rightarrow ⟨1⟩⟨2⟩⟨3⟩⟨4⟩
\ltx@carfourth {⟨1⟩} {⟨2⟩} {⟨3⟩} {⟨4⟩} \ldots \@nil \rightarrow ⟨4⟩
\ltx@cdrfour {⟨1⟩} {⟨2⟩} {⟨3⟩} {⟨4⟩} \ldots \@nil \rightarrow \ldots
\ltx@CarNum {⟨num⟩} {⟨1⟩} \ldots {⟨⟨num⟩⟩} {⟨(num)+1⟩} \ldots \@nil \rightarrow ⟨1⟩\ldots ⟨⟨num⟩⟩\ldots
\ltx@CarNumth {⟨num⟩} {⟨1⟩} \ldots {⟨⟨num⟩⟩} {⟨(num)+1⟩} \ldots \@nil \rightarrow ⟨⟨num⟩⟩\ldots
\ltx@CdrNum {⟨num⟩} {⟨1⟩} \ldots {⟨⟨num⟩⟩} {⟨(num)+1⟩} \ldots \@nil \rightarrow ⟨⟨num⟩⟩+1\ldots
\end{verbatim}

Macros with uppercase letters are expandable in two expansion steps. Changes in version 2019/12/15 v1.24:

- Macros \ltx@carsecond, \ltx@carthird, \ltx@carfourth, \ltx@CarNumth added.
- Macros \ltx@cdr, \ltx@cdrtwo, \ltx@cdrthree, \ltx@cdrfour, \ltx@CdrNum are expandable in two expansion steps and retain spaces and braces after the first gobbled arguments.

1.7 Tail recursion

\begin{verbatim}
\ltx@ReturnAfterFi {⟨1⟩} \fi \rightarrow \fi ⟨1⟩
\ltx@ReturnAfterElseFi {⟨1⟩} \else {⟨2⟩} \fi \rightarrow \fi ⟨1⟩
\end{verbatim}
1.8 Empty macro

\ltxempty →

1.9 Characters

<table>
<thead>
<tr>
<th>Command</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>\ltxspace</td>
<td>→ \</td>
</tr>
<tr>
<td>\ltxpercentchar</td>
<td>→ %</td>
</tr>
<tr>
<td>\ltxbackslashchar</td>
<td>→ \</td>
</tr>
<tr>
<td>\ltxhashchar</td>
<td>→ #</td>
</tr>
<tr>
<td>\ltxleftbracechar</td>
<td>→ {</td>
</tr>
<tr>
<td>\ltxrightbracechar</td>
<td>→ }</td>
</tr>
</tbody>
</table>

(since v1.7)

(since v1.8)

1.10 Boolean switch

\ltxnewif {⟨cmd⟩}

\ltxnewif defines a new boolean switch ⟨cmd⟩ like \newif. Unlike plain \TeX’s \newif, \ltxnewif is not \outer. The command ⟨cmd⟩ must start with the two characters if.

\ltxnewglobalif {⟨cmd⟩}

\ltxnewglobalif defines a new boolean switch ⟨cmd⟩ like \ltxnewif. However the switch setting commands, ⟨cmd⟩ without the prefix if and followed by true or false are acting globally.

1.11 Command definitions

\ltxisundefined {⟨cmd⟩} {⟨yes⟩} {⟨no⟩}

If ϵ-\TeX{} is available, \mbox{\ltxisundefined} 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”.

\ltxisundefined {⟨cmd⟩} {⟨yes⟩} {⟨no⟩}

If ϵ-\TeX{} is available, \mbox{\ltxisundefined} 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{}.

\ltxlocalexpandafter

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.
1.12 Stripping

\texttt{\LaTeX@RemovePrefix} \texttt{\LaTeX@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 \texttt{\meaning} or \texttt{\pdf@lastmatch}. Macro \texttt{\LaTeX@RemovePrefix} has the same meaning as \LaTeX’s \texttt{\strip@prefix}, whereas macro \texttt{\LaTeX@StripPrefix} expands the next token once before stripping the prefix.

\texttt{\LaTeX@onelevel@sanitize \{⟨macro⟩\}}

Macro \texttt{\LaTeX@onelevel@sanitize} provides \LaTeX’s \texttt{@onelevel@sanitize}. The macro is expanded once and the contents is converted to characters with catcode 12 (other) and space tokens with catcode 10 (space). Then then sanitized contents is stored into the macro again. Since version 1.12.

1.13 File management

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

1.13.1 File extensions

\texttt{\LaTeX@clsextension} \texttt{\LaTeX@pkgextension}

Macros \texttt{\LaTeX@clsextension} and \texttt{\LaTeX@pkgextension} stores the strings \texttt{cls} and \texttt{sty}. In opposite to \LaTeX’s \texttt{@clsextension} and \texttt{@pkgextension} they can also be used after \texttt{\begin{document}}.

1.13.2 Load check

\texttt{\LaTeX@ifclassloaded \{⟨class⟩\} \{⟨yes⟩\} \{⟨no⟩\}} \texttt{\LaTeX@ifpackageloaded \{⟨package⟩\} \{⟨yes⟩\} \{⟨no⟩\}}

Macros \texttt{\LaTeX@ifclassloaded/\LaTeX@ifpackageloaded} execute \texttt{⟨yes⟩}, if the \texttt{⟨class⟩} or \texttt{⟨package⟩} is loaded, otherwise \texttt{⟨no⟩} is called. Both \texttt{⟨class⟩} and \texttt{⟨package⟩} are specified without extension. The macros can also be used after \texttt{\begin{document}}.

\texttt{\LaTeX@iffileloaded \{⟨file⟩\} \{⟨yes⟩\} \{⟨no⟩\}}

If \LaTeX’s \texttt{\ProvidesFile} macro was called before using \texttt{⟨file⟩} as argument, then \texttt{\LaTeX@iffileloaded} calls \texttt{⟨yes⟩}, otherwise \texttt{⟨no⟩} is executed because of a missing \texttt{\ProvidesFile}. The \LaTeX kernel does not have a counterpart of \texttt{\LaTeX@iffileloaded}.

Note that the file name used in \texttt{\ProvidesFile} and \texttt{\LaTeX@iffileloaded} must match. For example, if \TeX’s default extension \texttt{.tex} was given in the first command, then it must also specified in the latter command and vice versa.
1.13.3 Version date check

\IfClass{⟨class⟩}{⟨date⟩}{⟨yes⟩}{⟨no⟩}
\IfPackage{⟨package⟩}{⟨date⟩}{⟨yes⟩}{⟨no⟩}
\IfFile{⟨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 pdf\TeX’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 2ε in 1994.

1.14 Macro additions

\GlobalAppendToMacro{⟨cmd⟩}{⟨addition⟩}
\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 beforehand. Due to a bug ⟨addition⟩ must not contain \par before version 2010/10/25 v1.9.

\GlobalPrependToMacro{⟨cmd⟩}{⟨addition⟩}
\LocalPrependToMacro{⟨cmd⟩}{⟨addition⟩}

The ⟨addition⟩ is prepended to the parameterless macro ⟨cmd⟩. If ⟨cmd⟩ is undefined or has the meaning \relax, then it will be initialized as empty macro beforehand. The macros were added in version 2011/08/22 v1.21.

1.15 Next character detection

\IfNextChar{⟨char⟩}{⟨yes⟩}{⟨no⟩}

If next character is ⟨char⟩ then ⟨yes⟩ is called, otherwise ⟨no⟩. The character is not removed. Spaces are silently removed when looking for ⟨char⟩ as \LaTeX’s version \kernel@ifnextchar does. But there are also small differences:

- The space can be used as ⟨char⟩. In this case optional spaces before ⟨char⟩ are not supported of course.

- If the optional space is a command that is a character (defined by \let or \futurelet), then \kernel@ifnextchar breaks with an \TeX error. \IfNextChar silently removes this token as optional space.

Since 2010/03/01 v1.3.
Macro \texttt{\textbackslash ifnextchar@nospace} behaves like macro \texttt{\textbackslash ifnextchar} with the exception that optional spaces are not supported before \texttt{\textbackslash char}. Since 2011/04/14 v1.19.

1.16 \texttt{\textbackslash leavevmode}, \texttt{\textbackslash mbox}

Macro \texttt{\textbackslash leavevmode} calls pdfTeX's \texttt{\textbackslash quitvmode}. Otherwise \texttt{\leavevmode} is used and defined if it is necessary.

Macro \texttt{\textbackslash mbox} reimplements \texttt{\mbox} with two changes. Instead of \texttt{\leavevmode} it uses \texttt{\textbackslash leavevmode} and stops right after \texttt{\hbox}. Especially it does not grab the argument and allows the extended syntax of \texttt{\hbox}.

1.17 Expandable test for emptiness

Macro \texttt{\textbackslash ifempty} checks in exact two expansion steps whether \texttt{\langle stuff\rangle} is empty or contains token. Depending on the result \texttt{\langle yes\rangle} or \texttt{\langle no\rangle} is executed. The token in \texttt{\langle stuff\rangle} may contain \texttt{\par} and unmatched conditionals (\texttt{\if, \else, \fi, …}). Since version 2010/11/12 v1.11.

Macro \texttt{\textbackslash ifblank} tests in exact two expansion steps if \texttt{\langle stuff\rangle} is empty or contain only blank spaces. In this case argument \texttt{\langle yes\rangle} is called. If \texttt{\langle stuff\rangle} contains other tokens than spaces then \texttt{\langle no\rangle} is executed. Since version 2010/12/04 v1.13.

1.18 Stripping spaces

Macro \texttt{\textbackslash zap@space} strips spaces from \texttt{\langle stuff\rangle} that are not hidden inside curly braces. Like \LaTeX{}'s \texttt{\zap@space} it is expandable. Differences:

- Syntax: \texttt{\zap@space} also expects a space token and \texttt{\textbackslash empty} after \texttt{\langle stuff\rangle}.
- Macro \texttt{\textbackslash zap@space} is expandable in exact two expansion steps.
- Macro \texttt{\textbackslash zap@space} always retains curly braces.
- Macro \texttt{\zap@space} has a bug. It stops stripping spaces after a token group in curly braces if the first two tokens inside the group are equal.
- Macro \texttt{\textbackslash zap@space} also works with \texttt{\par} and conditionals (\texttt{\if, \else, \fi, …}).

Macro \texttt{\textbackslash zap@space} is available since version 2010/12/07 v1.14.
1.19 Check for emptiness of boxes

```latex
\ltx@ifboxempty{(box register number)}{(yes)}{(no)}
```

Macro `\ltx@ifboxempty` calls `⟨yes⟩` if the box exists (`\ifvoid` returns false) and the box does not contain any content. Otherwise if the box is void or contains something, then `⟨no⟩` is executed. Thus being empty means that the box exists and is either an `\hbox` or a `\vbox` and may even have dimensions other than 0.0 pt, but the box does not contain anything. Macro `\ltx@ifboxempty` is available since 2010/02/04 v1.16.

```latex
\ltx@ifboxvoidorempty{(box register number)}{(yes)}{(no)}
```

Macro `\ltx@ifboxvoidorempty` calls `⟨yes⟩` if the box is either void or does not contain any content. Otherwise `⟨no⟩` is executed. Macro `\ltx@ifboxvoidorempty` is available since 2010/02/04 v1.16.

2 Implementation

2.1 Identification

1 (*package*)

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

```latex
\begingroup\catcode61\catcode48\catcode32=10\relax%
\catcode13=5 % ^^M
3 \catcode35=6 % #
\catcode39=12 %'
7 \catcode44=12 % ,
10 \catcode58=12 % :
12 \catcode123=1 % {
14 \expandafter\let\expandafter\x\csname ver@ltxcmds.sty\endcsname
16 \expandafter\ifx\csname PackageInfo\endcsname\relax
18 \def\empty{}%
20 \expandafter\immediate\write-1{Package #1: #2, stopped}%
22 \expandafter\immediate\write-1{Package #1 Info: #2.}%
24 \expandafter\PackageInfo{#1}{#2, stopped}%
26 \expandafter\PackageInfo{#1}{#2, stopped}%
28 \x{\ltcmds}{The package is already loaded}%
30 \aftergroup\endinput
32 \endgroup%
```
\begin{verbatim}
\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{36}{3}\%$
\TMP@EnsureCode{38}{4}\%&
\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{62}{12}\%>
\TMP@EnsureCode{91}{12}\%
\TMP@EnsureCode{96}{12}\%
\TMP@EnsureCode{93}{12}\%]
\TMP@EnsureCode{94}{12}\%{ (superscript) (!)
\TMP@EnsureCode{124}{12}\%|
\edef\LTXcmds@AtEnd{\LTXcmds@AtEnd\noexpand\endinput}
\end{verbatim}

\section*{2.2 Numbers}
\begin{verbatim}
\ltx@zero
\chardef\ltx@zero=0 \%
\ltx@one
\chardef\ltx@one=1 \%
\ltx@two
\chardef\ltx@two=2 \%
\ltx@active
\chardef\ltx@active=13 \%
\ltx@cclv
\chardef\ltx@cclv=255 \%
\ltx@minusone
\def\ltx@minusone{% 
  -\ltx@one
}
\end{verbatim}

\section*{2.3 Scratch registers}
\begin{verbatim}
\ltx@LocToksA
\toksdef\ltx@LocToksA=0 \%
\ltx@LocToksB
\toksdef\ltx@LocToksB=2 \%
\end{verbatim}
\let\LocToksC \toksdef\LocToksC=4 %
\let\LocToksD \toksdef\LocToksD=6 %
\let\LocToksE \toksdef\LocToksE=8 %
\let\GlobToksA \toksdef\GlobToksA=1 %
\let\GlobToksB \toksdef\GlobToksB=3 %
\let\GlobToksC \toksdef\GlobToksC=5 %
\let\GlobToksD \toksdef\GlobToksD=7 %
\let\GlobToksE \toksdef\GlobToksE=9 %
\let\LocDimenA \dimendef\LocDimenA=0 %
\let\LocDimenB \dimendef\LocDimenB=2 %
\let\LocDimenC \dimendef\LocDimenC=4 %
\let\LocDimenD \dimendef\LocDimenD=6 %
\let\LocDimenE \dimendef\LocDimenE=8 %
\let\GlobDimenA \dimendef\GlobDimenA=1 %
\let\GlobDimenB \dimendef\GlobDimenB=3 %
\let\GlobDimenC \dimendef\GlobDimenC=5 %
\let\GlobDimenD \dimendef\GlobDimenD=7 %
\let\GlobDimenE \dimendef\GlobDimenE=9 %
\let\LocSkipA \skipdef\LocSkipA=0 %
2.4 Argument killers

\long\def\gobble#1{}
\long\def\gobbletwo#1#2{}
\long\def\gobblethree#1#2#3{}
\long\def\gobblefour#1#2#3#4{}
\def\GobbleNum#1{\romannumeral\csname ltx@zero\endcsname}\LTXcmds@GobbleNum...
2.5 Argument grabbers

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

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

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

\ltx@firstofthree
173 \long\def\ltx@firstofthree#1#2#3\{#1\}

\ltx@secondofthree
174 \long\def\ltx@secondofthree#1#2#3\{#2\}

\ltx@thirdofthree
175 \long\def\ltx@thirdofthree#1#2#3\{#3\}%

\ltx@firstoffour
176 \long\def\ltx@firstoffour#1#2#3#4\{#1\}

\ltx@secondoffour
177 \long\def\ltx@secondoffour#1#2#3#4\{#2\}

\ltx@thirdoffour
178 \long\def\ltx@thirdoffour#1#2#3#4\{#3\}%

\ltx@fourthoffour
179 \long\def\ltx@fourthoffour#1#2#3#4\{#4\}%

2.6 List helpers

\ltx@carzero
180 \long\def\ltx@carzero\{1\}@nil{}% 

\LTXcmds@cdrzero
181 \long\def\LTXcmds@cdrzero\{1\}@nil\{#1\}

\ltx@cdrzero
182 \def\ltx@cdrzero{% \roman numeral \LTXcmds@cdrzero\ltx@zero 184 }

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

\ltx@cdr
186 \long\def\ltx@cdr#1{% \roman numeral \LTXcmds@cdrzero\ltx@zero 188 }

\ltx@cartwo
189 \long\def\ltx@cartwo#1#2#3\@nil\{#1#2\}
\ltx@carsecond
  \long\def\ltx@carsecond#1#2#3\@nil{#2}

\ltx@cdrtwo
  \long\def\ltx@cdrtwo#1#2{%
  \romannumeral\LTXcmds@cdrzero\ltx@zero

\ltx@carthree
  \long\def\ltx@carthree#1#2#3#4\@nil{#1#2#3}

\ltx@carthird
  \long\def\ltx@carthird#1#2#3#4\@nil{#3}

\ltx@cdrthree
  \long\def\ltx@cdrthree#1#2#3{%
  \romannumeral\LTXcmds@cdrzero\ltx@zero

\ltx@carfour
  \long\def\ltx@carfour#1#2#3#4#5\@nil{#1#2#3#4}

\ltx@carfourth
  \long\def\ltx@carfourth#1#2#3#4#5\@nil{#4}

\ltx@cdrfour
  \long\def\ltx@cdrfour#1#2#3#4{%
  \romannumeral\LTXcmds@cdrzero\ltx@zero

\ltx@CarNum
  \def\ltx@CarNum#1{%
  \romannumeral\csname LTXcmds@CarNumFinish\endcsname{#1}000{x}\endcsname

\LTXcmds@CarNum
  \def\LTXcmds@CarNum#1{%
  \csname LTXcmds@C#1\endcsname{\LTXcmds@CarNum}

\LTXcmds@Cm
  \long\def\LTXcmds@Cm#1#2{%
  \endcsname{#1#2}

\LTXcmds@Cx
  \def\LTXcmds@Cx#1{%
  \endcsname{}
2.7 Tail recursion

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

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

2.8 Empty macro

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

2.9 Characters

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

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

\ltx@backslashchar
\begingroup
\lccode'0='\\\n\lowercase\endgroup
\def\ltx@backslashchar{0}
2.10 Boolean switch

2.10.1 Boolean switch

\ltx@hashchar
\begingroup
\lccode'0='\#{relax
\lowercase{\endgroup
\def\ltx@hashchar{0}%
\}
\endgroup
\lowercase{\endgroup
\def\ltx@leftbracechar{0}%
\}
\endgroup
\lowercase{\endgroup
\def\ltx@rightbracechar{0}%
\}
\endgroup
2.10 Boolean switch
\ltx@newif
\begingroup
\lccode'0='\#{relax
\lowercase{\endgroup
\expandafter\LTXcmds@newif\string#1\@nil
\}
\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
\}
\LTXcmds@newif
\begingroup
\lccode'0='-1 %
\expandafter\endgroup
\expandafter\LTXcmds@newif\string#1\@nil
\}
\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
\}
\ltx@newglobalif
\begingroup
\lccode'0='-1 %
\expandafter\endgroup
\expandafter\LTXcmds@newglobalif\string#1\@nil
\}
\endgroup
\expandafter\def\expandafter\LTXcmds@newglobalif\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
\}
\ltx@newglobalif
\begingroup
\lccode'0='-1 %
\expandafter\endgroup
\expandafter\LTXcmds@newglobalif\string#1\@nil
\}
\endgroup
\expandafter\def\expandafter\LTXcmds@newglobalif\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
\}
\LTXcmds@newglobalif

\begingroup
\expandafter\endgroup
\expandafter
\def\expandafter\LTXcmds@newglobalif\string\if#1\@nil{\
\expandafter\edef\csname#1true\endcsname{\
global\let\
\expandafter\noexpand\csname if#1\endcsname\
noexpand\iftrue\n\expandafter\edef\csname#1false\endcsname{\
global\let\
\expandafter\noexpand\csname if#1\endcsname\
noexpand\iffalse\}%
\csname#1false\endcsname}
}

2.11 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\n\}\
\expandafter\ltx@gobble\else\expandafter\ltx@firstofone\fi{\ltx@ifundefined}

\ltx@ifundefined
\def\ltx@ifundefined#1{\
\expandafter\ifx\csname #1\endcsname\relax\
\expandafter\ltx@firstoftwo\
\else\
\expandafter\ltx@secondoftwo\
\fi\n\}\
\expandafter\ltx@gobble\else\expandafter\ltx@firstofone\fi{\ltx@ifundefined}

\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\n\}\
\expandafter\ltx@gobble\else\expandafter\ltx@firstofone\fi{\ltx@IfUndefined}

\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\n\}\
\expandafter\ltx@gobble\else\expandafter\ltx@firstofone\fi{\ltx@IfUndefined}
2.12 Stripping

\ltx@RemovePrefix

\def\ltx@RemovePrefix#1>{}

\ltx@StripPrefix

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

\ltx@onelevel@sanitize

\def\ltx@onelevel@sanitize#1{\edef#1{\expandafter\ltx@RemovePrefix\meaning#1}}

2.13 File management

2.13.1 File extensions

\ltx@clsextension

\def\ltx@clsextension{cls}

\ltx@pkgextension

\def\ltx@pkgextension{sty}

2.13.2 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@secondoftwo\ltx@firstoftwo}
2.13.3 Version date check

\ltx@filelater

\LTXcmds@ifLater

\LTXcmds@ifclasslater

\ltx@ifpackagelater

\LTXcmds@ifpackageloaded

\ltx@ifdefined

\LTXcmds@ifpackageloaded
2.14 Macro additions
\longdef\ltx@GlobalPrependToMacro#1#2{%
  \ifx\ltx@undefined#1%
    \let#1\ltx@empty
  \else
    \ifx\relax#1%
      \let#1\ltx@empty
    \fi
  \fi
  \begingroup
    \ltx@LocToksA{#2}\
    \ltx@LocToksB\expandafter{#1}\
    \xdef#1{\the\ltx@LocToksA\the\ltx@LocToksB}\
  \endgroup
}\global
\longdef\ltx@LocalPrependToMacro#1#2{%
  \global\let\LTXcmds@gtemp#1\
  \ifx\ltx@undefined\LTXcmds@gtemp
    \global\let\LTXcmds@gtemp\ltx@empty
  \else
    \ifx\relax\LTXcmds@gtemp
      \global\let\LTXcmds@gtemp\ltx@empty
    \fi
  \fi
  \begingroup
    \ltx@LocToksA{#2}\
    \ltx@LocToksB\expandafter{\LTXcmds@gtemp}\
    \xdef\LTXcmds@gtemp{\the\ltx@LocToksA\the\ltx@LocToksB}\
  \endgroup
  \let#1\LTXcmds@gtemp
}\global
\longdef\ltx@ifnextchar#1#2#3{%
  \begingroup
    \let\LTXcmds@CharToken= #1\relax
    \ltx@LocToksA{\endgroup#2}\
    \ltx@LocToksB{\endgroup#3}\
    \futurelet\LTXcmds@LetToken\LTXcmds@ifnextchar
\endgroup
\def\LTXcmds@ifnextchar{\ifx\LTXcmds@LetToken\LTXcmds@CharToken\
  \the\expandafter\ltx@LocToksA\else\
    \expandafter\ifx\csname LTXcmds@LetToken\endcsname\LTXcmds@SpaceToken\
      \expandafter\expandafter\expandafter\LTXcmds@ifnextchar\else\
        \the\expandafter\expandafter\expandafter\ltx@LocToksB\fi\fi\fi\fi
\endgroup

2.15 Next character detection
\longdef\ltx@ifnextchar#1#2#3{%
  \begingroup
    \let\LTXcmds@CharToken= #1\relax
    \ltx@LocToksA{\endgroup#2}\
    \ltx@LocToksB{\endgroup#3}\
    \futurelet\LTXcmds@LetToken\LTXcmds@ifnextchar
\endgroup
\def\LTXcmds@ifnextchar{\ifx\LTXcmds@LetToken\LTXcmds@CharToken\
  \the\expandafter\ltx@LocToksA\else\
    \expandafter\ifx\csname LTXcmds@CharToken\endcsname\LTXcmds@SpaceToken\
      \expandafter\expandafter\expandafter\LTXcmds@ifnextchar\else\
        \the\expandafter\expandafter\expandafter\ltx@LocToksB\fi\fi\fi\fi
\endgroup
\longdef\ltx@ifnextchar#1#2{%
  \begingroup
    \let\LTXcmds@CharToken= #1\relax
    \ltx@LocToksA{\endgroup#2}\
    \ltx@LocToksB{\endgroup#3}\
  \endgroup
\def\ltx@ifnextchar{\ifx\LTXcmds@CharToken\LTXcmds@CharToken\
  \the\expandafter\ltx@LocToksA\
  \else\
    \expandafter\ifx\csname LTXcmds@CharToken\endcsname\LTXcmds@SpaceToken\
      \expandafter\expandafter\expandafter\LTXcmds@ifnextchar\else\
        \the\expandafter\expandafter\expandafter\ltx@LocToksB\fi\fi
\endgroup
23
\LTXcmds@ifnextchar \futurelet does not distinguish between a character and a command that is a character (defined by using \let or \futurelet). Therefore the space is caught by \romannumeral with negative character constant that gobbles one optional space.

\def\LTXcmds@ifnextchar{%  
\expandafter\futurelet\expandafter\LTXcmds@CharToken\expandafter\LTXcmds@ifnextchar\romannumeral-'.%}

\LTXcmds@SpaceToken
\ltx@firstofone{\let\LTXcmds@SpaceToken=} %
\ltx@ifnextchar@nospace
\long\def\ltx@ifnextchar@nospace#1#2#3{%  
\begingroup  
\let\LTXcmds@CharToken=#1\relax  
\ltx@LocToksA{\endgroup#2}%  
\ltx@LocToksB{\endgroup#3}%  
\futurelet\LTXcmds@CharToken\LTXcmds@ifnextchar@nospace
}

\long\def\ltx@ifnextchar@nospace#1#2#3{%  
\the\if\LTXcmds@CharToken\LTXcmds@CharToken\else\fi
\fi
}\ltx@leavevmode, \ltx@mbox
\ltx@leavevmode
\if\LTXcmds@CharToken\LTXcmds@CharToken\else\fi
\if\LTXcmds@CharToken\LTXcmds@CharToken\else\fi
\if\LTXcmds@CharToken\LTXcmds@CharToken\else\fi
\if\LTXcmds@CharToken\LTXcmds@CharToken\else\fi

2.16 \ltx@leavevmode, \ltx@mbox
\ltx@mbox
\def\ltx@mbox{\ltx@leavevmode\hbox}

2.17 Help macros
\LTXcmds@num
\ltx@ifundefined{numexpr}{\expandafter\ltx@firstofone\expandafter{\number#1}}{\expandafter\ltx@firstofone\expandafter{\the\numexpr#1}}}

2.18 Expandable test for emptiness
2.18.1 Vanilla \TeX
\ltx@ifempty\The macro is based on \@ifempty of Robert R. Schneck [1] and \@ifnull of Ulrich Diez [2]. There are three cases to consider:
1. #1 is empty,
2. #1 is not empty and the first token is not a begingroup character,
3. #1 starts with a begingroup character (catcode 1).
\def\LTXcmds@temp#1{\long\def\ltx@ifempty##1{\romannumeral0\iffalse\fi \expandafter\ltx@gobble\expandafter{\string##1}\expandafter\ltx@gobble\string \expandafter\ltx@firstofthree\expandafter{\iffalse}\fi \expandafter\ltx@secondoftwo}
2.18.2 With \detokenize

Ahmed Musa provided \ifstrequal using \detokenize and \pdfstrcmp [3]. Ulrich Diez, GL, Heiko Oberdiek improved it further by removing \pdfstrcmp and taking three arguments [4, 5, 6, 7, 8].
In case of \( \varepsilon \)-\TeX{} the test for an empty box is done via \texttt{lastnodetype} as suggested by David Kastrup [9].

Implementation using \( \varepsilon \)-\TeX{}'s \texttt{lastnodetype}.

Implementation without \( \varepsilon \)-\TeX{} using a signature at the beginning of the test box.
Box 0 has been changed and is restored by closing the group.

\endgroup
\begingroup
\setbox\ltx@zero=\ifhbox#1\hbox\else\vbox\fi{%
\penalty\ltx@two
\ifhmode\unhcopy\else\unvcopy\fi#1\relax
\expandafter
\ifnum\lastpenalty=\ltx@two
\def\next{\endgroup\expandafter\ltx@firstoftwo}\
\else
\def\next{\endgroup\expandafter\ltx@secondoftwo}\
\fi
\else
\def\next{\endgroup\expandafter\ltx@secondoftwo}\
\fi
\next
\fi
\ltx@ifBoxVoidOrEmpty
\def\ltx@ifBoxVoidOrEmpty#1{\
\ifvoid#1\relax
\expandafter\ltx@thirdoffour
\fi
\ltx@ifBoxEmpty{#1}\
\}
\LTXcmds@AtEnd%
⟨/package⟩

3 Installation

3.1 Download

Package. This package is available on CTAN:\!


Bundle. All the packages of the bundle ‘ltxcmds’ 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/ltxcmds.tds.zip

TDS refers to the standard “A Directory Structure for \TeX\ Files” (CTAN:pkg/tds). Directories with texmf in their name are usually organized this way.

\footnote{CTAN:pkg/ltxcmds}
3.2 Bundle installation

**Unpacking.** Unpack the `ltxcmds.tds.zip` in the TDS tree (also known as `texmf` tree) of your choice. Example (Linux):

```
unzip ltxcmds.tds.zip -d `~/texmf`
```

3.3 Package installation

**Unpacking.** The `.dtx` file is a self-extracting `docstrip` archive. The files are extracted by running the `.dtx` through plain 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/ltxcmds/ltxcmds.sty`
- `ltxcmds.pdf` → `doc/latex/ltxcmds/ltxcmds.pdf`
- `ltxcmds.dtx` → `source/latex/ltxcmds/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`.

3.4 Refresh file name databases

If your TeX distribution (TeX Live, MiKTeX, ...) relies on file name databases, you must refresh these. For example, TeX Live users run `texhash` or `mktexlsr`.

3.5 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:

```
ltext \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 pdf\LaTeX:

```
pdflatex ltxcmds.dtx
makeindex -s gind.ist ltxcmds.idx
pdflatex ltxcmds.dtx
makeindex -s gind.ist ltxcmds.idx
pdflatex ltxcmds.dtx
```
4 References

[1] Robert R. Schneck: \ifempty solution (was Macro puzzle: maximally general \ifempty); newsgroup comp.text.tex, news:3eflada_6@corp.newsgroups.com, 2003-06-17. https://groups.google.com/group/comp.text.tex/msg/be03a159ec374895

[2] Ulrich Diez: Re: TeX refuses to strip outer braces in argument; newsgroup comp.text.tex, news:ibk3t8$ee7$1@news.albasani.net, 2010-11-12. https://groups.google.com/group/comp.text.tex/msg/803bd57221a04996

[3] Ahmed Musa: Re: TeX refuses to strip outer braces in argument; newsgroup comp.text.tex, news:f5496afe-40ed-42bd-b629-a2419ecf7c0d@o14g2000prn.googlegroups.com, 2010-12-03. https://groups.google.com/group/comp.text.tex/msg/fbf7d61a0c3a807d

[4] Ulrich Diez: Re: TeX refuses to strip outer braces in argument; newsgroup comp.text.tex, news:idbo94$uka$1@four.albasani.net, 2010-12-03. https://groups.google.com/group/comp.text.tex/msg/0c230ee479487962

[5] Ulrich Diez: Re: TeX refuses to strip outer braces in argument; newsgroup comp.text.tex, news:idbp4$cg1$1@news.albasani.net, 2010-12-03. https://groups.google.com/group/comp.text.tex/msg/bbef4263390d647b

[6] Ulrich Diez: Re: TeX refuses to strip outer braces in argument; newsgroup comp.text.tex, news:idd4ga$r83$1@four.albasani.net, 2010-12-04. https://groups.google.com/group/comp.text.tex/msg/00df1ec103cd272

[7] GL: Re: TeX refuses to strip outer braces in argument; newsgroup comp.text.tex, news:4cfa2e27$0$7389$426a74cc@news.free.fr, 2010-12-04. https://groups.google.com/group/comp.text.tex/msg/d3a75995c1cf267e

[8] Heiko Oberdiek: Re: TeX refuses to strip outer braces in argument; newsgroup comp.text.tex, news:iddhq1$skj$1@news.eternal-september.org, 2010-12-04. https://groups.google.com/group/comp.text.tex/msg/5f7a23e3ab70e347

[9] David Kastrup: How to detect if \vbox is empty; newsgroup comp.text.tex, 2011-02-04. https://groups.google.com/group/comp.text.tex/msg/8d3cb89496a4d86d

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]
- \texttt{\textbackslash removeprefix} and \texttt{\textbackslash stripprefix} added.
- \texttt{\textbackslash ifclassloaded}, \texttt{\textbackslash ifpackageloaded}, \texttt{\textbackslash iffileloaded}, \texttt{\textbackslash ifclasslater}, \texttt{\textbackslash ifpackagelater}, \texttt{\textbackslash iffilelater}, \texttt{\textbackslash clsextension}, \texttt{\textbackslash pkgextension} added.
- \texttt{\textbackslash globappento} and \texttt{\textbackslash localappento} added.

[2010/03/01 v1.3]
- \texttt{\textbackslash newif} added.
- \texttt{\textbackslash ifnextchar} added.
- Numbers \texttt{\textbackslash zero}, \texttt{\textbackslash one}, \texttt{\textbackslash two}, \texttt{\textbackslash cclv} added.

[2010/03/09 v1.4]
- \texttt{\textbackslash pkgextension} and \texttt{\textbackslash clsextension} are hardcoded to avoid trouble with \texttt{\onlypreamble}.

[2010/04/08 v1.5]
- \texttt{\textbackslash cartwo}, \texttt{\textbackslash cdrtwo}, \texttt{\textbackslash carthree}, \texttt{\textbackslash cdrthree}, \texttt{\textbackslash carfour}, \texttt{\textbackslash cdrfour} added.
- \texttt{\textbackslash returnafterfi} and \texttt{\textbackslash returnafterelsefi} fixed.

[2010/04/16 v1.6]
- \texttt{\textbackslash leavevmode}, \texttt{\textbackslash mbox} added.

[2010/04/26 v1.7]
- \texttt{\textbackslash gobblenum}, \texttt{\textbackslash cdrnum}, \texttt{\textbackslash carnum} added.
- \texttt{\textbackslash carzero}, \texttt{\textbackslash cdrzero} added.
- \texttt{\textbackslash hashchar} added.

[2010/09/11 v1.8]
- \texttt{\textbackslash leftbracechar}, \texttt{\textbackslash rightbracechar} added.

[2010/10/25 v1.9]
- \texttt{\textbackslash localappento} and \texttt{\textbackslash globappento} are now \texttt{\long}.

[2010/10/31 v1.10]
- \texttt{\textbackslash newglobalif} added.
[2010/11/12 v1.11]
- \ltx@ifempty added.
- \ltx@firstofthree, \ltx@secondofthree, \ltx@thirdofthree added.

[2010/12/02 v1.12]
- \ltx@onelevel@sanitize added.
- \LTXcmds@num fixed for the case with \numexpr (bug found by GL).

[2010/12/04 v1.13]
- \ltx@ifblank added.
- Optimization for \ltx@ifempty.

[2010/12/07 v1.14]
- \ltx@zapspace added.

[2010/12/12 v1.15]
- \ltx@minusone added.

[2011/02/04 v1.16]
- \ltx@IfBoxEmpty and \ltx@IfBoxVoidOrEmpty added.
- \ltx@firstoffour, ..., \ltx@fourthoffour added.

[2011/02/05 v1.17]
- \ltx@IfBoxEmpty: an empty box may have non-zero dimensions.

[2011/03/16 v1.18]
- \ltx@ifclasslater fixed.

[2011/04/14 v1.19]
- \ltx@ifnextchar: detection of optional spaces modified.
- \ltx(Loc,Glob)(Toks,Dimen, Skip)(A,B,C,D,E) added.

[2011/04/18 v1.20]
- \ltx@ifnextchar with conditional support (thanks GL for bug report).

[2011/08/22 v1.21]
- \ltx@GlobalPrependToMacro, \ltx@LocalPrependToMacro added (feature request of Martin Münch).
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

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Page(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>#</td>
<td>254</td>
</tr>
<tr>
<td>%</td>
<td>244</td>
</tr>
<tr>
<td>&amp;</td>
<td>653, 656</td>
</tr>
<tr>
<td>{</td>
<td>264</td>
</tr>
<tr>
<td>@nil</td>
<td>180, 181, 185, 189, 190, 194, 195, 199, 200, 219, 228, 272, 277, 294, 300, 415, 417, 642, 644, 648</td>
</tr>
<tr>
<td>@undefined</td>
<td>58</td>
</tr>
<tr>
<td>\</td>
<td>249</td>
</tr>
<tr>
<td>{</td>
<td>259</td>
</tr>
<tr>
<td>}</td>
<td>264</td>
</tr>
</tbody>
</table>

A

\aftergroup 29

B

\box 543, 554

C

\catcode 2, 3, 5, 6, 7, 8, 9, 10, 11, 12, 13, 33, 34, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 60, 69, 70, 72, 73, 74, 75, 76, 80, 81, 82, 83, 84, 87, 88, 90, 91, 92, 93, 97, 99, 652, 653, 655, 656 |

D

\detokenize 623, 634
\dimendef 134, 135, 136, 137, 138, 139, 140, 141, 142, 143

E

\empty 17, 18
\endcsname 14, 21, 50, 66, 76, 162, 168, 208, 214, 217, 278, 280, 283, 285, 288, 301, 303, 306, 308, 310, 320, 322, 330, 342, 343, 381, 506, 549, 626, 637
\endinput 29, 115
\endlinechar 4, 35, 71, 77, 89
\escapechar 270, 275, 292, 297

F

\futurelet 499, 514, 525

H

\bbox 541, 543, 552, 554, 570, 663, 674, 682

I

\if 277, 300
\ifcase 390
\ifcat 623, 634
\ifcsname 342
\iffalse 286, 309, 589, 595, 602, 611
\ifhbox 663, 674, 682
\ifhmode 664, 676, 684
\ifnum 391, 393, 395, 422, 608, 679, 687
\iftrue 281, 304
\ifdef 550, 659, 699