

# The ltxcmds package

Heiko Oberdiek  
<heiko.oberdiek at gmail.com>

2011/04/14 v1.19

## Abstract

The package ltxcmds exports some utility macros from the L<sup>A</sup>T<sub>E</sub>X kernel into a separate namespace and also provides them for other formats such as plain-T<sub>E</sub>X.

## Contents

|          |   |          |
|----------|---|----------|
| <b>1</b> | <b>Documentation</b>                    | <b>3</b> |
| 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 . . . . .                  | 4        |
| 1.7      | Tail recursion . . . . .                | 5        |
| 1.8      | Empty macro . . . . .                   | 5        |
| 1.9      | Characters . . . . .                    | 5        |
| 1.10     | Boolean switch . . . . .                | 5        |
| 1.11     | Command definitions . . . . .           | 5        |
| 1.12     | Stripping . . . . .                     | 6        |
| 1.13     | File management . . . . .               | 6        |
| 1.13.1   | File extensions . . . . .               | 6        |
| 1.13.2   | Load check . . . . .                    | 6        |
| 1.13.3   | Version date check . . . . .            | 7        |
| 1.14     | Macro additions . . . . .               | 7        |
| 1.15     | Next character detection . . . . .      | 7        |
| 1.16     | \ltx@leavevmode, \ltx@mbox . . . . .    | 8        |
| 1.17     | Expandable test for emptiness . . . . . | 8        |
| 1.18     | Stripping spaces . . . . .              | 8        |
| 1.19     | Check for emptiness of boxes . . . . .  | 8        |
| <b>2</b> | <b>Implementation</b>                   | <b>9</b> |
| 2.1      | Identification . . . . .                | 9        |
| 2.2      | Numbers . . . . .                       | 11       |
| 2.3      | Scratch registers . . . . .             | 11       |
| 2.4      | Argument killers . . . . .              | 13       |
| 2.5      | Argument grabbers . . . . .             | 13       |
| 2.6      | List helpers . . . . .                  | 14       |
| 2.7      | Tail recursion . . . . .                | 15       |
| 2.8      | Empty macro . . . . .                   | 15       |
| 2.9      | Characters . . . . .                    | 15       |
| 2.10     | Boolean switch . . . . .                | 16       |
| 2.11     | Command definitions . . . . .           | 17       |
| 2.12     | Stripping . . . . .                     | 18       |

|          |   |           |
|----------|---|-----------|
| 2.13     | File management                                       | 18        |
| 2.13.1   | File extensions                                       | 18        |
| 2.13.2   | Load check  | 18        |
| 2.13.3   | Version date check                                    | 19        |
| 2.14     | Macro additions                                       | 20        |
| 2.15     | Next character detection                              | 20        |
| 2.16     | <code>\ltx@leavevmode</code> , <code>\ltx@mbox</code> | 21        |
| 2.17     | Help macros   | 22        |
| 2.18     | Expandable test for emptiness                         | 22        |
| 2.18.1   | Vanilla <code>T<sub>E</sub>X</code>                   | 22        |
| 2.18.2   | With <code>\detokenize</code>                         | 23        |
| 2.18.3   | <code>\ltx@ifblank</code>                             | 23        |
| 2.19     | <code>\ltx@zapspace</code>                            | 24        |
| 2.20     | <code>\ltx@ifBoxEmpty</code>                          | 24        |
| <b>3</b> | <b>Test</b>   | <b>25</b> |
| 3.1      | Catcode checks for loading                            | 25        |
| 3.2      | Test <code>\ltx@GobbleNum</code>                      | 27        |
| 3.3      | Test <code>\ltx@ifempty</code>                        | 30        |
| 3.4      | Test <code>\ltx@zap@space</code>                      | 32        |
| 3.5      | Test <code>\ltx@ifBoxEmpty</code>                     | 33        |
| 3.6      | Test for next character detection                     | 34        |
| <b>4</b> | <b>Installation</b>                                   | <b>36</b> |
| 4.1      | Download  | 36        |
| 4.2      | Bundle installation                                   | 37        |
| 4.3      | Package installation                                  | 37        |
| 4.4      | Refresh file name databases                           | 37        |
| 4.5      | Some details for the interested                       | 37        |
| <b>5</b> | <b>References</b>                                     | <b>38</b> |
| <b>6</b> | <b>History</b>  | <b>38</b> |
|          | [2009/08/05 v1.0]                                     | 38        |
|          | [2009/12/12 v1.1]                                     | 39        |
|          | [2010/01/28 v1.2]                                     | 39        |
|          | [2010/03/01 v1.3]                                     | 39        |
|          | [2010/03/09 v1.4]                                     | 39        |
|          | [2010/04/08 v1.5]                                     | 39        |
|          | [2010/04/16 v1.6]                                     | 39        |
|          | [2010/04/26 v1.7]                                     | 39        |
|          | [2010/09/11 v1.8]                                     | 39        |
|          | [2010/10/25 v1.9]                                     | 39        |
|          | [2010/10/31 v1.10]                                    | 39        |
|          | [2010/11/12 v1.11]                                    | 40        |
|          | [2010/12/02 v1.12]                                    | 40        |
|          | [2010/12/04 v1.13]                                    | 40        |
|          | [2010/12/07 v1.14]                                    | 40        |
|          | [2010/12/12 v1.15]                                    | 40        |
|          | [2011/02/04 v1.16]                                    | 40        |
|          | [2011/02/05 v1.17]                                    | 40        |
|          | [2011/03/16 v1.18]                                    | 40        |
|          | [2011/04/14 v1.19]                                    | 40        |
| <b>7</b> | <b>Index</b>  | <b>40</b> |

# 1 Documentation

## 1.1 Introduction

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

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

## 1.2 Numbers

|                        |               |     |
|------------------------|---------------|-----|
| $\text{\ltx@zero}$     | $\rightarrow$ | 0   |
| $\text{\ltx@one}$      | $\rightarrow$ | 1   |
| $\text{\ltx@two}$      | $\rightarrow$ | 2   |
| $\text{\ltx@ccclv}$    | $\rightarrow$ | 255 |
| $\text{\ltx@minusone}$ | $\rightarrow$ | -1  |

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

## 1.3 Scratch registers

Following the conventions of plain  $\text{\TeX}$  and  $\text{\LaTeX}$  the first ten registers are free to use. Even numbered registers are for local, odd numbered for global use.

|  |
|--|
| $\text{\ltx@}(\text{Loc},\text{Glob})(\text{Toks},\text{Dimen},\text{Skip})(\text{A},\text{B},\text{C},\text{D},\text{E})$ |
|--|

The name consists of the prefix  $\text{\ltx@}$ , 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:  $\text{\ltx@LocToksA}$ .

Since 2011/04/14 v1.19.

## 1.4 Argument killers

|  |               |
|--|---------------|
| $\text{\ltx@gobble} \{\langle 1 \rangle\}$   | $\rightarrow$ |
| $\text{\ltx@gobbletwo} \{\langle 1 \rangle\} \{\langle 2 \rangle\}$  | $\rightarrow$ |
| $\text{\ltx@gobblethree} \{\langle 1 \rangle\} \{\langle 2 \rangle\} \{\langle 3 \rangle\}$                      | $\rightarrow$ |
| $\text{\ltx@gobblefour} \{\langle 1 \rangle\} \{\langle 2 \rangle\} \{\langle 3 \rangle\} \{\langle 4 \rangle\}$ | $\rightarrow$ |

|   |               |
|---|---------------|
| $\text{\ltx@GobbleNum} \{\langle num \rangle\} \{\langle 1 \rangle\} \{\langle 2 \rangle\} \dots \{\langle \langle num \rangle \rangle\}$ | $\rightarrow$ |
|---|---------------|

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

## 1.5 Argument grabbers

|  |               |                     |
|--|---------------|---------------------|
| <code>\ltx@firstofone {⟨1⟩}</code>                     | $\rightarrow$ | $\langle 1 \rangle$ |
| <code>\ltx@firstoftwo {⟨1⟩} {⟨2⟩}</code>               | $\rightarrow$ | $\langle 1 \rangle$ |
| <code>\ltx@secondoftwo {⟨1⟩} {⟨2⟩}</code>              | $\rightarrow$ | $\langle 2 \rangle$ |
| <code>\ltx@firstofthree {⟨1⟩} {⟨2⟩} {⟨3⟩}</code>       | $\rightarrow$ | $\langle 1 \rangle$ |
| <code>\ltx@secondofthree {⟨1⟩} {⟨2⟩} {⟨3⟩}</code>      | $\rightarrow$ | $\langle 2 \rangle$ |
| <code>\ltx@thirdofthree {⟨1⟩} {⟨2⟩} {⟨3⟩}</code>       | $\rightarrow$ | $\langle 3 \rangle$ |
| <code>\ltx@firstoffour {⟨1⟩} {⟨2⟩} {⟨3⟩} {⟨4⟩}</code>  | $\rightarrow$ | $\langle 1 \rangle$ |
| <code>\ltx@secondoffour {⟨1⟩} {⟨2⟩} {⟨3⟩} {⟨4⟩}</code> | $\rightarrow$ | $\langle 2 \rangle$ |
| <code>\ltx@thirdoffour {⟨1⟩} {⟨2⟩} {⟨3⟩} {⟨4⟩}</code>  | $\rightarrow$ | $\langle 3 \rangle$ |
| <code>\ltx@fourthoffour {⟨1⟩} {⟨2⟩} {⟨3⟩} {⟨4⟩}</code> | $\rightarrow$ | $\langle 4 \rangle$ |

Macros `\ltx@firstofthree`, `\ltx@secondofthree` and `\ltx@thirdofthree` were added in version 2010/11/12 v1.11. Macros `\ltx@firstoffour`, ..., `\ltx@fourthoffour` were added in version 2011/02/04 v1.16.

## 1.6 List helpers

|                                     |               |     |
|-------------------------------------|---------------|-----|
| <code>\ltx@carzero ... \@nil</code> | $\rightarrow$ |     |
| <code>\ltx@cdrzero ... \@nil</code> | $\rightarrow$ | ... |

|                                       |               |                     |
|---------------------------------------|---------------|---------------------|
| <code>\ltx@car {⟨1⟩} ... \@nil</code> | $\rightarrow$ | $\langle 1 \rangle$ |
| <code>\ltx@cdr {⟨1⟩} ... \@nil</code> | $\rightarrow$ | ...                 |

|  |               |                                       |
|--|---------------|---------------------------------------|
| <code>\ltx@cartwo {⟨1⟩} {⟨2⟩} ... \@nil</code> | $\rightarrow$ | $\langle 1 \rangle \langle 2 \rangle$ |
| <code>\ltx@cdrtwo {⟨1⟩} {⟨2⟩} ... \@nil</code> | $\rightarrow$ | ...                                   |

|  |               |   |
|--|---------------|---|
| <code>\ltx@carthree {⟨1⟩} {⟨2⟩} {⟨3⟩} ... \@nil</code> | $\rightarrow$ | $\langle 1 \rangle \langle 2 \rangle \langle 3 \rangle$ |
| <code>\ltx@cdrthree {⟨1⟩} {⟨2⟩} {⟨3⟩} ... \@nil</code> | $\rightarrow$ | ...   |

|   |               |   |
|---|---------------|---|
| <code>\ltx@carfour {⟨1⟩} {⟨2⟩} {⟨3⟩} {⟨4⟩} ... \@nil</code> | $\rightarrow$ | $\langle 1 \rangle \langle 2 \rangle \langle 3 \rangle \langle 4 \rangle$ |
| <code>\ltx@cdrfour {⟨1⟩} {⟨2⟩} {⟨3⟩} {⟨4⟩} ... \@nil</code> | $\rightarrow$ | ...   |

|  |               |   |
|--|---------------|---|
| <code>\ltx@CarNum {⟨num⟩} {⟨1⟩} ... {⟨⟨num⟩⟩} {⟨⟨num⟩+1⟩} ... \@nil</code> | $\rightarrow$ | $\{ \langle 1 \rangle \} \dots \{ \langle \langle num \rangle \rangle \} \dots$ |
| <code>\ltx@CdrNum {⟨num⟩} {⟨1⟩} ... {⟨⟨num⟩⟩} {⟨⟨num⟩+1⟩} ... \@nil</code> | $\rightarrow$ | $\{ \langle \langle num \rangle + 1 \rangle \} \dots$                           |

Macros `\ltx@CarNum` and `\ltx@CdrNum` are expandable in exact two expansion steps.

## 1.7 Tail recursion

|   |               |                      |
|---|---------------|----------------------|
| <code>\ltx@ReturnAfterFi {⟨1⟩} \fi</code>                 | $\rightarrow$ | <code>\fi ⟨1⟩</code> |
| <code>\ltx@ReturnAfterElseFi {⟨1⟩} \else {⟨2⟩} \fi</code> | $\rightarrow$ | <code>\fi ⟨1⟩</code> |

## 1.8 Empty macro

|                         |               |
|-------------------------|---------------|
| <code>\ltx@empty</code> | $\rightarrow$ |
|-------------------------|---------------|

## 1.9 Characters

|                                  |               |                             |
|----------------------------------|---------------|-----------------------------|
| <code>\ltx@space</code>          | $\rightarrow$ | <code>\space</code>         |
| <code>\ltx@percentchar</code>    | $\rightarrow$ | <code>%</code>              |
| <code>\ltx@backslashchar</code>  | $\rightarrow$ | <code>\</code>              |
| <code>\ltx@hashchar</code>       | $\rightarrow$ | <code>#</code> (since v1.7) |
| <code>\ltx@leftbracechar</code>  | $\rightarrow$ | <code>{</code> (since v1.8) |
| <code>\ltx@rightbracechar</code> | $\rightarrow$ | <code>}</code> (since v1.8) |

### 1.10 Boolean switch

|                                 |
|---------------------------------|
| <code>\ltx@newif {⟨cmd⟩}</code> |
|---------------------------------|

`\ltx@newif` defines a new boolean switch `⟨cmd⟩` like `\newif`. Unlike plain  $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ 's `\newif`, `\ltx@newif` is not `\outer`. The command `⟨cmd⟩` must start with the two characters `if`.

|                                       |
|---------------------------------------|
| <code>\ltx@newglobalif {⟨cmd⟩}</code> |
|---------------------------------------|

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

### 1.11 Command definitions

|  |
|--|
| <code>\ltx@ifundefined {⟨cmd⟩} {⟨yes⟩} {⟨no⟩}</code> |
|--|

If  $\varepsilon\text{-T}_{\mathrm{E}}\mathrm{X}$  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”.

|  |
|--|
| <code>\ltx@ifUndefined {⟨cmd⟩} {⟨yes⟩} {⟨no⟩}</code> |
|--|

If  $\varepsilon\text{-T}_{\mathrm{E}}\mathrm{X}$  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  $\varepsilon\text{-T}_{\mathrm{E}}\mathrm{X}$ .

`\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.

## 1.12 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 L<sup>A</sup>T<sub>E</sub>X’s `\strip@prefix`, whereas macro `\ltx@StripPrefix` expands the next token once before stripping the prefix.

`\ltx@onelevel@sanitize {⟨macro⟩}`

Macro `\ltx@onelevel@sanitize` provides L<sup>A</sup>T<sub>E</sub>X’s `\@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 L<sup>A</sup>T<sub>E</sub>X kernel. Also they can be used after the preamble.

### 1.13.1 File extensions

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

Macros `\ltx@clsextension` and `\ltx@styextension` stores the strings `cls` and `sty`. In opposite to L<sup>A</sup>T<sub>E</sub>X’s `\@clsextension` and `\@styextension` they can also be used after `\begin{document}`.

### 1.13.2 Load check

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

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

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

If L<sup>A</sup>T<sub>E</sub>X’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 L<sup>A</sup>T<sub>E</sub>X 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  $\text{\TeX}$ 's default extension `.tex` was given in the first command, then it must also be specified in the latter command and vice versa.

### 1.13.3 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  $\text{\LaTeX}$  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  $\text{\LaTeX}$  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  $\text{\pdfTeX}$ '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  $\text{\LaTeX}$  2<sub>ε</sub> in 1994.

## 1.14 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. Due to a bug `<addition>` must not contain `\par` before version 2010/10/25 v1.9.

## 1.15 Next character detection

```
\ltx@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  $\text{\LaTeX}$ '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  $\text{\TeX}$  error. `\ltx@ifnextchar` silently removes this token as optional space.

Since 2010/03/01 v1.3.

```
\ltx@ifnextchar@nospace {<char>} {<yes>} {<no>}
```

Macro `\ltx@ifnextchar@nospace` behaves like macro `\ltx@ifnextchar` with the exception that optional spaces are not supported before `<char>`. Since 2011/04/14 v1.19.

## 1.16 `\ltx@leavevmode`, `\ltx@mbox`

`\ltx@leavevmode`

Macro `\ltx@leavevmode` calls pdfTeX's `\quitvmode`. Otherwise `\leavevmode` is used and defined if it is necessary.

`\ltx@mbox`

Macro `\ltx@mbox` reimplements `\mbox` with two changes. Instead of `\leavevmode` it uses `\ltx@leavevmode` and stops right after `\hbox`. Especially it does not grab the argument and allows the extended syntax of `\hbox`.

## 1.17 Expandable test for emptiness

`\ltx@ifempty`  $\{\langle stuff \rangle\} \{\langle yes \rangle\} \{\langle no \rangle\}$

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

`\ltx@ifblank`  $\{\langle stuff \rangle\} \{\langle yes \rangle\} \{\langle no \rangle\}$

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

## 1.18 Stripping spaces

`\ltx@zapspace`  $\{\langle stuff \rangle\}$

Macro `\ltx@zapspace` strips spaces from  $\langle stuff \rangle$  that are not hidden inside curly braces. Like L<sup>A</sup>T<sub>E</sub>X's `\zap@space` it is expandable. Differences:

- Syntax: `\zap@space` also expects a space token and `\@empty` after  $\langle stuff \rangle$ .
- Macro `\ltx@zapspace` is expandable in exact two expansion steps.
- Macro `\ltx@zapspace` always retains curly braces.
- Macro `\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 `\ltx@zapspace` also works with `\par` and conditionals (`\if`, `\else`, `\fi`, ...).

Macro `\ltx@zapspace` is available since version 2010/12/07 v1.14.

## 1.19 Check for emptiness of boxes

`\ltx@ifBoxEmpty`  $\{\langle box\ register\ number \rangle\} \{\langle yes \rangle\} \{\langle no \rangle\}$

Macro `\ltx@ifBoxEmpty` calls  $\langle yes \rangle$  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  $\langle no \rangle$  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 0.0pt, but the box does not contain anything. Macro `\ltx@ifboxempty` is available since 2010/02/04 v1.16.

`\ltx@ifboxvoidoreempty {⟨box register number⟩} {⟨yes⟩} {⟨no⟩}`

Macro `\ltx@ifboxvoidoreempty` calls `⟨yes⟩` if the box is either void or does not contain any content. Otherwise `⟨no⟩` is executed. Macro `\ltx@ifboxvoidoreempty` 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 L<sup>A</sup>T<sub>E</sub>X.

```
2 \begingroup\catcode61\catcode48\catcode32=10\relax%
3 \catcode13=5 % ^^M
4 \endlinechar=13 %
5 \catcode35=6 % #
6 \catcode39=12 % '
7 \catcode44=12 % ,
8 \catcode45=12 % -
9 \catcode46=12 % .
10 \catcode58=12 % :
11 \catcode64=11 % @
12 \catcode123=1 % {
13 \catcode125=2 % }
14 \expandafter\let\expandafter\x\csname ver@ltxcmds.sty\endcsname
15 \ifx\x\relax % plain-TeX, first loading
16 \else
17 \def\empty{}%
18 \ifx\x\empty % LaTeX, first loading,
19 % variable is initialized, but \ProvidesPackage not yet seen
20 \else
21 \expandafter\ifx\csname PackageInfo\endcsname\relax
22 \def\x#1#2{%
23 \immediate\write-1{Package #1 Info: #2.}%
24 }%
25 \else
26 \def\x#1#2{\PackageInfo{#1}{#2, stopped}}%
27 \fi
28 \x{ltxcmds}{The package is already loaded}%
29 \aftergroup\endinput
30 \fi
31 \fi
32 \endgroup%
```

Package identification:

```
33 \begingroup\catcode61\catcode48\catcode32=10\relax%
34 \catcode13=5 % ^^M
35 \endlinechar=13 %
36 \catcode35=6 % #
37 \catcode39=12 % '
38 \catcode40=12 % (
39 \catcode41=12 % )
40 \catcode44=12 % ,
41 \catcode45=12 % -
42 \catcode46=12 % .
43 \catcode47=12 % /
44 \catcode58=12 % :
```

```

45 \catcode64=11 % @
46 \catcode91=12 % [
47 \catcode93=12 % ]
48 \catcode123=1 % {
49 \catcode125=2 % }
50 \expandafter\ifx\csname ProvidesPackage\endcsname\relax
51   \def\x#1#2#3[#4]{\endgroup
52     \immediate\write-1{Package: #3 #4}%
53     \xdef#1{#4}%
54   }%
55 \else
56   \def\x#1#2[#3]{\endgroup
57     #2[{#3}]%
58     \ifx#1\@undefined
59       \xdef#1{#3}%
60     \fi
61     \ifx#1\relax
62       \xdef#1{#3}%
63     \fi
64   }%
65 \fi
66 \expandafter\x\csname ver@ltxcmds.sty\endcsname
67 \ProvidesPackage{ltxcmds}%
68 [2011/04/14 v1.19 LaTeX kernel commands for general use (H0)]%
69 \begingroup\catcode61\catcode48\catcode32=10\relax%
70 \catcode13=5 % ^^M
71 \endlinechar=13 %
72 \catcode123=1 % {
73 \catcode125=2 % }
74 \catcode64=11 % @
75 \def\x{\endgroup
76   \expandafter\edef\csname LTXcmds@AtEnd\endcsname{%
77     \endlinechar=\the\endlinechar\relax
78     \catcode13=\the\catcode13\relax
79     \catcode32=\the\catcode32\relax
80     \catcode35=\the\catcode35\relax
81     \catcode61=\the\catcode61\relax
82     \catcode64=\the\catcode64\relax
83     \catcode123=\the\catcode123\relax
84     \catcode125=\the\catcode125\relax
85   }%
86 }%
87 \x\catcode61\catcode48\catcode32=10\relax%
88 \catcode13=5 % ^^M
89 \endlinechar=13 %
90 \catcode35=6 % #
91 \catcode64=11 % @
92 \catcode123=1 % {
93 \catcode125=2 % }
94 \def\TMP@EnsureCode#1#2{%
95   \edef\LTXcmds@AtEnd{%
96     \LTXcmds@AtEnd
97     \catcode#1=\the\catcode#1\relax
98   }%
99   \catcode#1=#2\relax
100 }
101 \TMP@EnsureCode{36}{3}% $
102 \TMP@EnsureCode{38}{4}% &
103 \TMP@EnsureCode{40}{12}% (
104 \TMP@EnsureCode{41}{12}% )
105 \TMP@EnsureCode{45}{12}% -
106 \TMP@EnsureCode{46}{12}% .

```

```

107 \TMP@EnsureCode{47}{12}% /
108 \TMP@EnsureCode{60}{12}% <
109 \TMP@EnsureCode{62}{12}% >
110 \TMP@EnsureCode{91}{12}% [
111 \TMP@EnsureCode{96}{12}% ‘
112 \TMP@EnsureCode{93}{12}% ]
113 \TMP@EnsureCode{94}{12}% ^ (superscript) (!)
114 \TMP@EnsureCode{124}{12}% |
115 \edef\LTXcmds@AtEnd{\LTXcmds@AtEnd\noexpand\endinput}

```

## 2.2 Numbers

```

\ltx@zero

116 \chardef\ltx@zero=0 %

\ltx@one

117 \chardef\ltx@one=1 %

\ltx@two

118 \chardef\ltx@two=2 %

\ltx@active

119 \chardef\ltx@active=13 %

\ltx@ccclv

120 \chardef\ltx@ccclv=255 %

\ltx@minusone

121 \def\ltx@minusone{%
122   -\ltx@one
123 }

```

## 2.3 Scratch registers

```

\ltx@LocToksA

124 \toksdef\ltx@LocToksA=0 %

\ltx@LocToksB

125 \toksdef\ltx@LocToksB=2 %

\ltx@LocToksC

126 \toksdef\ltx@LocToksC=4 %

\ltx@LocToksD

127 \toksdef\ltx@LocToksD=6 %

\ltx@LocToksE

128 \toksdef\ltx@LocToksE=8 %

\ltx@GlobToksA

129 \toksdef\ltx@GlobToksA=1 %

\ltx@GlobToksB

130 \toksdef\ltx@GlobToksB=3 %

\ltx@GlobToksC

131 \toksdef\ltx@GlobToksC=5 %

```

```

\ltx@GlobToksD
132 \toksdef\ltx@GlobToksD=7 %

\ltx@GlobToksE
133 \toksdef\ltx@GlobToksE=9 %

\ltx@LocDimenA
134 \dimendef\ltx@LocDimenA=0 %

\ltx@LocDimenB
135 \dimendef\ltx@LocDimenB=2 %

\ltx@LocDimenC
136 \dimendef\ltx@LocDimenC=4 %

\ltx@LocDimenD
137 \dimendef\ltx@LocDimenD=6 %

\ltx@LocDimenE
138 \dimendef\ltx@LocDimenE=8 %

\ltx@GlobDimenA
139 \dimendef\ltx@GlobDimenA=1 %

\ltx@GlobDimenB
140 \dimendef\ltx@GlobDimenB=3 %

\ltx@GlobDimenC
141 \dimendef\ltx@GlobDimenC=5 %

\ltx@GlobDimenD
142 \dimendef\ltx@GlobDimenD=7 %

\ltx@GlobDimenE
143 \dimendef\ltx@GlobDimenE=9 %

\ltx@LocSkipA
144 \skipdef\ltx@LocSkipA=0 %

\ltx@LocSkipB
145 \skipdef\ltx@LocSkipB=2 %

\ltx@LocSkipC
146 \skipdef\ltx@LocSkipC=4 %

\ltx@LocSkipD
147 \skipdef\ltx@LocSkipD=6 %

\ltx@LocSkipE
148 \skipdef\ltx@LocSkipE=8 %

\ltx@GlobSkipA
149 \skipdef\ltx@GlobSkipA=1 %

\ltx@GlobSkipB
150 \skipdef\ltx@GlobSkipB=3 %

```

```

\ltx@GlobSkipC
151 \skipdef\ltx@GlobSkipC=5 %

\ltx@GlobSkipD
152 \skipdef\ltx@GlobSkipD=7 %

\ltx@GlobSkipE
153 \skipdef\ltx@GlobSkipE=9 %

```

## 2.4 Argument killers

```

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

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

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

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

\ltx@GobbleNum
158 \def\ltx@GobbleNum#1{%
159   \romannumeral
160   \csname ltx@zero%
161   \expandafter\LTxcmds@GobbleNum
162   \romannumeral\LTxcmds@num{#1}000{m\endcsname}%
163 }

\LTxcmds@GobbleNum
164 \def\LTxcmds@GobbleNum#1{%
165   \csname LTxcmds@G#1\LTxcmds@GobbleNum
166 }

\LTxcmds@Gm
167 \long\def\LTxcmds@Gm#1{%
168   \endcsname
169 }

```

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

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

\ltx@carzero
182 \long\def\ltx@carzero#1\@nil{}%

\ltx@cdrzero
183 \long\def\ltx@cdrzero#1\@nil{#1}%

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

\ltx@cdrtwo
185 \long\def\ltx@cdrtwo#1#2#3\@nil{#3}

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

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

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

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

\ltx@CarNum
190 \def\ltx@CarNum#1{%
191   \romannumeral
192   \csname LTXcmds@CarNumFinish%
193   \expandafter\LTXcmds@CarNum
194   \romannumeral\LTXcmds@num{#1}000{x\endcsname}%
195 }

\LTXcmds@CarNum
196 \def\LTXcmds@CarNum#1{%
197   \csname LTXcmds@C#1\LTXcmds@CarNum
198 }

```

\LTXcmds@Cm

```
199 \long\def\LTXcmds@Cm#1#2{%
200   \endcsname{#1#2}%
201 }
```

\LTXcmds@Cx

```
202 \def\LTXcmds@Cx#1{%
203   \endcsname{#1}%
204 }
```

\LTXcmds@CarNumFinish

```
205 \long\def\LTXcmds@CarNumFinish#1#2\@nil{%
206   \ltx@zero
207   #1%
208 }
```

\ltx@CdrNum

```
209 \def\ltx@CdrNum#1{%
210   \romannumeral0%
211   \expandafter\expandafter\expandafter\LTXcmds@CdrNum
212   \ltx@GobbleNum{#1}%
213 }
```

\LTXcmds@CdrNum

```
214 \long\def\LTXcmds@CdrNum#1\@nil{ #1}%
```

## 2.7 Tail recursion

\ltx@ReturnAfterFi

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

\ltx@ReturnAfterElseFi

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

## 2.8 Empty macro

\ltx@empty

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

## 2.9 Characters

\ltx@space

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

\ltx@percentchar

```
219 \begingroup
220   \lccode'0='%\relax
221 \lowercase{\endgroup
222   \def\ltx@percentchar{0}%
223 }
```

\ltx@backslashchar

```
224 \begingroup
225   \lccode'0='\\ \relax
226 \lowercase{\endgroup
227   \def\ltx@backslashchar{0}%
228 }
```

\ltx@hashchar

```
229 \begingroup
230   \lccode'0='#\relax
231 \lowercase{\endgroup
232   \def\ltx@hashchar{0}%
233 }
```

\ltx@leftbracechar

```
234 \begingroup
235   \lccode'0='{ \relax
236 \lowercase{\endgroup
237   \def\ltx@leftbracechar{0}%
238 }
```

\ltx@rightbracechar

```
239 \begingroup
240   \lccode'0='}\relax
241 \lowercase{\endgroup
242   \def\ltx@rightbracechar{0}%
243 }
```

## 2.10 Boolean switch

\ltx@newif

```
244 \def\ltx@newif#1{%
245   \begingroup
246     \escapechar=-1 %
247   \expandafter\endgroup
248   \expandafter\LTXcmds@newif\string#1\@nil
249 }
```

\LTXcmds@newif

```
250 \begingroup
251   \escapechar=-1 %
252 \expandafter\endgroup
253 \expandafter\def\expandafter\LTXcmds@newif\string\if#1\@nil{%
254   \expandafter\edef\csname#1true\endcsname{%
255     \let
256     \expandafter\noexpand\csname if#1\endcsname
257     \noexpand\iftrue
258   }%
259   \expandafter\edef\csname#1false\endcsname{%
260     \let
261     \expandafter\noexpand\csname if#1\endcsname
262     \noexpand\iffalse
263   }%
264   \csname#1false\endcsname
265 }
```

\ltx@newglobalif

```
266 \def\ltx@newglobalif#1{%
267   \begingroup
268     \escapechar=-1 %
269   \expandafter\endgroup
270   \expandafter\LTXcmds@newglobalif\string#1\@nil
271 }
```

\LTXcmds@newglobalif

```
272 \begingroup
273   \escapechar=-1 %
```



```

274 \expandafter\endgroup
275 \expandafter
276 \def\expandafter\LTXcmds@newglobalif\string\if#1\@nil{%
277   \expandafter\edef\csname#1true\endcsname{%
278     \global\let
279     \expandafter\noexpand\csname if#1\endcsname
280     \noexpand\iftrue
281   }%
282   \expandafter\edef\csname#1false\endcsname{%
283     \global\let
284     \expandafter\noexpand\csname if#1\endcsname
285     \noexpand\iffalse
286   }%
287   \csname#1false\endcsname
288 }

```

## 2.11 Command definitions

\ltx@LocalExpandAfter

```

289 \def\ltx@LocalExpandAfter{%
290   \begingroup
291     \expandafter\expandafter\expandafter
292   \endgroup
293   \expandafter
294 }

295 \ltx@LocalExpandAfter
296 \ifx\csname ifcsname\endcsname\relax

```

\ltx@ifundefined

```

297 \def\ltx@ifundefined#1{%
298   \expandafter\ifx\csname #1\endcsname\relax
299     \expandafter\ltx@firstoftwo
300   \else
301     \expandafter\ltx@secondoftwo
302   \fi
303 }%

```

\ltx@ifUndefined

```

304 \def\ltx@ifUndefined#1{%
305   \begingroup\expandafter\expandafter\expandafter\endgroup
306   \expandafter\ifx\csname #1\endcsname\relax
307     \expandafter\ltx@firstoftwo
308   \else
309     \expandafter\ltx@secondoftwo
310   \fi
311 }%

312 \expandafter\ltx@gobble
313 \else
314 \expandafter\ltx@firstofone
315 \fi
316 {%

```

\ltx@ifundefined

```

317 \def\ltx@ifundefined#1{%
318   \ifcsname #1\endcsname
319     \expandafter\ifx\csname #1\endcsname\relax
320       \expandafter\expandafter\expandafter\ltx@firstoftwo
321     \else
322       \expandafter\expandafter\expandafter\ltx@secondoftwo
323     \fi

```

```

324     \else
325         \expandafter\ltx@firstoftwo
326     \fi
327 }%

```

\ltx@ifundefined

```

328 \let\ltx@ifundefined\ltx@ifundefined
329 }

```

## 2.12 Stripping

\ltx@RemovePrefix

```

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

```

\ltx@StripPrefix

```

331 \def\ltx@StripPrefix{%
332     \expandafter\ltx@RemovePrefix
333 }

```

\ltx@onelevel@sanitize

```

334 \def\ltx@onelevel@sanitize#1{%
335     \edef#1{%
336         \expandafter
337         \ltx@RemovePrefix\meaning#1%
338     }%
339 }

```

## 2.13 File management

### 2.13.1 File extensions

\ltx@clsextension

```

340 \def\ltx@clsextension{cls}

```

\ltx@pkgextension

```

341 \def\ltx@pkgextension{sty}

```

### 2.13.2 Load check

\ltx@iffileloaded

```

342 \def\ltx@iffileloaded#1{%
343     \ltx@ifundefined{ver@#1}\ltx@secondoftwo\ltx@firstoftwo
344 }

```

\ltx@ifclassloaded

```

345 \def\ltx@ifclassloaded#1{%
346     \ltx@iffileloaded{#1.\ltx@clsextension}%
347 }

```

\ltx@ifpackageloaded

```

348 \def\ltx@ifpackageloaded#1{%
349     \ltx@iffileloaded{#1.\ltx@pkgextension}%
350 }

```

### 2.13.3 Version date check

\ltx@iffilelater

```
351 \def\ltx@iffilelater#1#2{%
352   \ltx@iffileloaded{#1}{%
353     \expandafter\LTxcmds@iflater\expandafter{%
354       \number
355       \expandafter\expandafter\expandafter\LTxcmds@ParseVersion
356       \expandafter\expandafter\expandafter{%
357         \csname ver@#1\endcsname
358       }%
359     \expandafter}\expandafter{%
360       \number
361       \expandafter\LTxcmds@ParseVersion\expandafter{#2}%
362     }%
363   }\ltx@secondoftwo
364 }
```

\LTxcmds@iflater

```
365 \def\LTxcmds@iflater#1#2{%
366   \ifcase 0%
367     \ifnum#1<19940101 %
368     \else
369       \ifnum#2<19940101 %
370       \else
371         \ifnum#2>#1 %
372         \else
373           1%
374         \fi
375       \fi
376     \fi
377     \ltx@space
378     \expandafter\ltx@secondoftwo
379   \else
380     \expandafter\ltx@firstoftwo
381   \fi
382 }
```

\ltx@ifclasslater

```
383 \def\ltx@ifclasslater#1{%
384   \ltx@iffilelater{#1.\ltx@clsextension}%
385 }
```

\ltx@ifpackagelater

```
386 \def\ltx@ifpackagelater#1{%
387   \ltx@iffilelater{#1.\ltx@pkgextension}%
388 }

389 \ltx@ifundefined{pdfmatch}{%
```

\LTxcmds@ParseVersion

```
390 \def\LTxcmds@ParseVersion#1{%
391   \LTxcmds@@ParseVersion#10000/00/00\@nil
392   }%
```

\LTxcmds@@ParseVersion

```
393 \def\LTxcmds@@ParseVersion#1#2#3#4/#5#6/#7#8#9\@nil{%
394   #1#2#3#4#5#6#7#8%
395   }%

396 }{%
```

\LTXcmds@ParseVersion

```
397 \def\LTXcmds@ParseVersion#1{%
398   \ifnum\pdfmatch{%
399     ~%
400     (199[4-9] | [2-9] [0-9] [0-9] [0-9] )/%
401     (0[1-9] | 1[0-2] )/%
402     (0[1-9] | [1-2] [0-9] | 3[0-1] )%
403   }{#1}=1 %
404   \ltx@StripPrefix\pdflastmatch1 %
405   \ltx@StripPrefix\pdflastmatch2 %
406   \ltx@StripPrefix\pdflastmatch3 %
407   \else
408     0%
409   \fi
410 }%
411 }
```

## 2.14 Macro additions

\ltx@GlobalAppendToMacro

```
412 \long\def\ltx@GlobalAppendToMacro#1#2{%
413   \ifx\ltx@undefined#1%
414     \let#1\ltx@empty
415   \else
416     \ifx\relax#1%
417       \let#1\ltx@empty
418     \fi
419   \fi
420   \begingroup
421     \ltx@LocToksA\expandafter{#1#2}%
422     \xdef#1{\the\ltx@LocToksA}%
423   \endgroup
424 }
```

\ltx@LocalAppendToMacro

```
425 \long\def\ltx@LocalAppendToMacro#1#2{%
426   \global\let\LTXcmds@gtemp#1%
427   \ifx\ltx@undefined\LTXcmds@gtemp
428     \global\let\LTXcmds@gtemp\ltx@empty
429   \else
430     \ifx\relax\LTXcmds@gtemp
431       \global\let\LTXcmds@gtemp\ltx@empty
432     \fi
433   \fi
434   \begingroup
435     \ltx@LocToksA\expandafter{\LTXcmds@gtemp#2}%
436     \xdef\LTXcmds@gtemp{\the\ltx@LocToksA}%
437   \endgroup
438   \let#1\LTXcmds@gtemp
439 }
```

## 2.15 Next character detection

\ltx@ifnextchar

```
440 \long\def\ltx@ifnextchar#1#2#3{%
441   \begingroup
442   \let\LTXcmds@CharToken= #1\relax
443   \ltx@LocToksA{\endgroup#2}%
444   \ltx@LocToksB{\endgroup#3}%
445   \futurelet\LTXcmds@LetToken\LTXcmds@ifnextchar
446 }
```

`\LTXcmds@ifnextchar`

```

447 \def\LTXcmds@ifnextchar{%
448   \ifx\LTXcmds@LetToken\LTXcmds@CharToken
449     \the\expandafter\ltx@LocToksA
450   \else
451     \ifx\LTXcmds@LetToken\LTXcmds@SpaceToken
452       \expandafter\expandafter\expandafter\LTXcmds@@ifnextchar
453     \else
454       \the\expandafter\expandafter\expandafter\ltx@LocToksB
455     \fi
456   \fi
457 }

```

`\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.

```

458 \def\LTXcmds@@ifnextchar{%
459   \expandafter\futurelet
460   \expandafter\LTXcmds@LetToken
461   \expandafter\LTXcmds@ifnextchar
462   \romannumeral-'\.%
463 }

```

`\LTXcmds@SpaceToken`

```

464 \ltx@firstofone{\let\LTXcmds@SpaceToken= } %

```

`\ltx@ifnextchar@nospace`

```

465 \long\def\ltx@ifnextchar@nospace#1#2#3{%
466   \begingroup
467   \let\LTXcmds@CharToken= #1\relax
468   \ltx@LocToksA{\endgroup#2}%
469   \ltx@LocToksB{\endgroup#3}%
470   \futurelet\LTXcmds@LetToken\LTXcmds@ifnextchar@nospace
471 }

```

`\LTXcmds@ifnextchar@nospace`

```

472 \def\LTXcmds@ifnextchar@nospace{%
473   \the
474   \ifx\LTXcmds@LetToken\LTXcmds@CharToken
475     \expandafter\ltx@LocToksA
476   \else
477     \expandafter\ltx@LocToksB
478   \fi
479 }

```

## 2.16 `\ltx@leavevmode`, `\ltx@mbox`

`\ltx@leavevmode`

```

480 \ltx@ifundefined{quitvmode}{%
481   \ltx@ifundefined{leavevmode}{%
482     \ltx@ifundefined{voidb@x}{%
483       \ltx@ifundefined{newbox}{%
484         \def\ltx@leavevmode{%
485           \begingroup
486             \setbox\ltx@zero=\hbox{}}%
487           \begingroup
488             \setbox\ltx@zero=\hbox{\box\ltx@zero}%
489           \endgroup
490           \unhbox\ltx@zero
491         \endgroup

```

```

492     }%
493   }{%
494     \csname newbox\endcsname\LTxcmds@VoidBox
495     \ifvoid\LTxcmds@VoidBox
496     \else
497       \setbox\LTxcmds@VoidBox=\hbox{}%
498       \begingroup
499         \setbox\LTxcmds@VoidBox=\hbox{\box\LTxcmds@VoidBox}%
500       \endgroup
501     \fi
502     \def\ltx@leavevmode{\unhbox\LTxcmds@VoidBox}%
503   }%
504   }{%
505     \def\ltx@leavevmode{\unhbox\voidb@x}%
506   }%
507 }{%
508   \let\ltx@leavevmode\leavevmode
509 }%
510 }{%
511   \let\ltx@leavevmode\quitvmode
512 }

```

`\ltx@mbox`

```

513 \def\ltx@mbox{%
514   \ltx@leavevmode
515   \hbox
516 }

```

## 2.17 Help macros

`\LTxcmds@num`

```

517 \ltx@ifundefined{numexpr}{%
518   \def\LTxcmds@num#1{%
519     \expandafter\ltx@firstofone\expandafter{%
520       \number#1%
521     }%
522   }%
523 }{%
524   \def\LTxcmds@num#1{%
525     \expandafter\ltx@firstofone\expandafter{%
526       \the\numexpr#1%
527     }%
528   }%
529 }

```

## 2.18 Expandable test for emptiness

```

530 \ltx@ifundefined{detokenize}{%

```

### 2.18.1 Vanilla T<sub>E</sub>X

`\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).

```

531 \def\LTxcmds@temp#1{%
532   \long\def\ltx@ifempty##1{%
533     \romannumeral0%
534     \iffalse{\fi
535       \expandafter\ltx@gobble\expandafter{%
536         \expandafter{\string##1}%

```

```

537         \expandafter\ltx@gobble\string
538     }%
539     \expandafter\ltx@firstofthree\expandafter
540     {\iffalse}\fi
541     \expandafter#1\ltx@secondoftwo
542 }%
543 \expandafter#1\ltx@firstoftwo
544 }%

\ltx@ifblank

545 \long\def\ltx@ifblank##1{%
546     \romannumeral0%
547     \iffalse{\fi
548         \expandafter\expandafter\expandafter\ltx@gobble
549         \expandafter\expandafter\expandafter{%
550             \expandafter\expandafter\expandafter{%
551                 \expandafter\string\ltx@gobble##1.%
552             }%
553             \expandafter\ltx@gobble\string
554         }%
555         \expandafter\ltx@firstofthree\expandafter
556         {\iffalse}\fi
557         \expandafter#1\ltx@secondoftwo
558     }%
559     \expandafter#1\ltx@firstoftwo
560 }%
561 }%
562 \LTXcmds@temp{ }%

563 }%

```

### 2.18.2 With \detokenize

Ahmed Musa provided `\ifstrempy` 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].

```

\ltx@ifempty

564 \long\def\ltx@ifempty#1{%
565     \romannumeral%
566     \csname
567         LTXcmds@ifempty%
568         \ifcat$\detokenize{#1}$%
569         @%
570     \fi
571     \endcsname
572 }%

\LTXcmds@ifempty@

573 \long\def\LTXcmds@ifempty@#1#2{0 #1}%

\LTXcmds@ifempty

574 \long\def\LTXcmds@ifempty#1#2{0 #2}%

```

### 2.18.3 \ltx@ifblank

```

\ltx@ifblank

575 \long\def\ltx@ifblank#1{%
576     \romannumeral%
577     \csname
578         LTXcmds@ifempty%

```

```

579      \ifcat$\detokenize\expandafter{\ltx@gobble#1.}$%
580      @%
581      \fi
582    \endcsname
583  }%

584 }

```

## 2.19 \ltx@zapspace

\ltx@zapspace

```

585 \long\def\ltx@zapspace#1{%
586   \romannumeral
587   \LTXcmds@zapspace\ltx@zero#1 \@nil
588 }

```

\LTXcmds@zapspace

```

589 \long\def\LTXcmds@zapspace#1 #2\@nil{%
590   \ltx@ifempty{#2}{%
591     #1%
592   }{%
593     \LTXcmds@zapspace#1#2\@nil
594   }%
595 }

```

## 2.20 \ltx@ifboxempty

In case of  $\varepsilon$ -TeX the test for an empty box is done via `\lastnodetype` as suggested by David Kastrup [9].

```

596 \ltx@ifundefined{lastnodetype}{%
597   \catcode'\$=9 %
598   \catcode'\&=14 %
599 }{%
600   \catcode'\$=14 %
601   \catcode'\&=9 %
602 }

```

\ltx@ifboxempty

```

603 \def\ltx@ifboxempty#1{%
604   \ifvoid#1\relax
605     \expandafter\ltx@secondoftwo
606   \else

```

Implementation using  $\varepsilon$ -TeX's `\lastnodetype`.

```

607 &   \begingroup
608 &     \setbox\ltx@zero=\ifhbox#1\hbox\else\vbox\fi{%
609 &       \ifhmode\unhcopy\else\unvcopy\fi#1\relax
610 &       \expandafter
611 &     }%
612 &   \expandafter\endgroup
613 &   \ifnum\lastnodetype<\ltx@zero
614 &     \expandafter\expandafter\expandafter\ltx@firstoftwo
615 &   \else
616 &     \expandafter\expandafter\expandafter\ltx@secondoftwo
617 &   \fi

```

Implementation without  $\varepsilon$ -TeX using a signature at the beginning of the test box.

```

618 $   \begingroup
619 $     \setbox\ltx@zero=\ifhbox#1\hbox\else\vbox\fi{%
620 $       \penalty\ltx@one
621 $       \ifhmode\unhcopy\else\unvcopy\fi#1\relax
622 $       \expandafter

```



```

623 $      }%
624 $      \ifnum\lastpenalty=\ltx@one
Box 0 has been changed and is restored by closing the group.
625 $      \endgroup
626 $      \begingroup
627 $      \setbox\ltx@zero=\ifhbox#1\hbox\else\vbox\fi{%
628 $      \penalty\ltx@two
629 $      \ifhmode\unhcopy\else\unvcopy\fi#1\relax
630 $      \expandafter
631 $      }%
632 $      \ifnum\lastpenalty=\ltx@two
633 $      \def\next{\endgroup\expandafter\ltx@firstoftwo}%
634 $      \else
635 $      \def\next{\endgroup\expandafter\ltx@secondoftwo}%
636 $      \fi
637 $      \else
638 $      \def\next{\endgroup\expandafter\ltx@secondoftwo}%
639 $      \fi
640 $      \next
641 \fi
642 }

```

\ltx@ifBoxVoidOrEmpty

```

643 \def\ltx@ifBoxVoidOrEmpty#1{%
644   \ifvoid#1\relax
645     \expandafter\ltx@thirdoffour
646   \fi
647   \ltx@ifBoxEmpty{#1}%
648 }

649 \LTXcmds@AtEnd%
650 </package>

```

## 3 Test

### 3.1 Catcode checks for loading

```

651 <*test1>

652 \catcode'\{=1 %
653 \catcode'\}=2 %
654 \catcode'\#=6 %
655 \catcode'\@=11 %
656 \expandafter\ifx\csname count@\endcsname\relax
657   \countdef\count@=255 %
658 \fi
659 \expandafter\ifx\csname @gobble\endcsname\relax
660   \long\def\@gobble#1{}%
661 \fi
662 \expandafter\ifx\csname @firstofone\endcsname\relax
663   \long\def\@firstofone#1{#1}%
664 \fi
665 \expandafter\ifx\csname loop\endcsname\relax
666   \expandafter\@firstofone
667 \else
668   \expandafter\@gobble
669 \fi
670 {%
671   \def\loop#1\repeat{%
672     \def\body{#1}%
673     \iterate
674   }%

```

```

675 \def\iterate{%
676   \body
677   \let\next\iterate
678   \else
679   \let\next\relax
680   \fi
681   \next
682 }%
683 \let\repeat=\fi
684 }%
685 \def\RestoreCatcodes{}
686 \count@=0 %
687 \loop
688   \edef\RestoreCatcodes{%
689     \RestoreCatcodes
690     \catcode\the\count@=\the\catcode\count@\relax
691   }%
692   \ifnum\count@<255 %
693     \advance\count@ 1 %
694   \repeat
695
696 \def\RangeCatcodeInvalid#1#2{%
697   \count@=#1\relax
698   \loop
699     \catcode\count@=15 %
700     \ifnum\count@<#2\relax
701       \advance\count@ 1 %
702     \repeat
703 }
704 \def\RangeCatcodeCheck#1#2#3{%
705   \count@=#1\relax
706   \loop
707     \ifnum#3=\catcode\count@
708     \else
709       \errmessage{%
710         Character \the\count@\space
711         with wrong catcode \the\catcode\count@\space
712         instead of \number#3%
713       }%
714     \fi
715     \ifnum\count@<#2\relax
716       \advance\count@ 1 %
717     \repeat
718 }
719 \def\space{ }
720 \expandafter\ifx\csname LoadCommand\endcsname\relax
721   \def\LoadCommand{\input ltxcmds.sty\relax}%
722 \fi
723 \def\Test{%
724   \RangeCatcodeInvalid{0}{47}%
725   \RangeCatcodeInvalid{58}{64}%
726   \RangeCatcodeInvalid{91}{96}%
727   \RangeCatcodeInvalid{123}{255}%
728   \catcode'\@=12 %
729   \catcode'\=0 %
730   \catcode'\%=14 %
731   \LoadCommand
732   \RangeCatcodeCheck{0}{36}{15}%
733   \RangeCatcodeCheck{37}{37}{14}%
734   \RangeCatcodeCheck{38}{47}{15}%
735   \RangeCatcodeCheck{48}{57}{12}%
736   \RangeCatcodeCheck{58}{63}{15}%

```

```

737 \RangeCatcodeCheck{64}{64}{12}%
738 \RangeCatcodeCheck{65}{90}{11}%
739 \RangeCatcodeCheck{91}{91}{15}%
740 \RangeCatcodeCheck{92}{92}{0}%
741 \RangeCatcodeCheck{93}{96}{15}%
742 \RangeCatcodeCheck{97}{122}{11}%
743 \RangeCatcodeCheck{123}{255}{15}%
744 \RestoreCatcodes
745 }
746 \Test
747 \csname @@end\endcsname
748 \end
749 </test1>

```

### 3.2 Test \ltx@GobbleNum

```

750 <*test-gobble>
751 \catcode'\{=1 %
752 \catcode'\}=2 %
753 \catcode'\#=6 %
754 \expandafter\ifx\csname RequirePackage\endcsname\relax
755 \input ltxcmds.sty\relax
756 \else
757 \RequirePackage{ltxcmds}[2011/04/14]%
758 \fi
759 \catcode'\@=11 %
760 \def\msg#\{\immediate\write16}%
761 \msg{[Test \string\ltx@GobbleNum]}%
762 \long\def\Test#1=#2\{\%
763 \edef\StrA{\ltx@GobbleNum#1}%
764 \expandafter\expandafter\expandafter\def
765 \expandafter\expandafter\expandafter\StrAA
766 \expandafter\expandafter\expandafter{\ltx@GobbleNum#1}%
767 \edef\StrB{#2}%
768 \ifx\StrA\StrB
769 \ifx\StrAA\StrB
770 \msg{* ok.}%
771 \else
772 \msg{StrAA: \StrAA}%
773 \msg{StrB: \StrB}%
774 \errhelp{Test: #1=#2}%
775 \errmessage{Test (two expansions) failed}%
776 \fi
777 \else
778 \msg{StrA: \StrA}%
779 \msg{StrB: \StrB}%
780 \errhelp{Test: #1=#2}%
781 \errmessage{Test (edef) failed!}%
782 \fi
783 }
784 \Test0abc=abc\\
785 \Test1abc=bc\\
786 \Test2abc=c\\
787 \Test3abcd=d\\
788 \Test4abcde=e\\
789 \Test5abcdef=f\\
790 \Test6abcdefg=g\\
791 \Test7abcdefgh=h\\
792 \Test8abcdefghi=i\\
793 \Test9abcdefghij=j\\
794 \Test{10}0123456789X=X\\
795 \Test{12}abcdefghijklm=m\\
796 \Test{700}%

```

```

797 012345678901234567890123456789012345678901234567890123456789%
798 012345678901234567890123456789012345678901234567890123456789%
799 012345678901234567890123456789012345678901234567890123456789%
800 012345678901234567890123456789012345678901234567890123456789%
801 012345678901234567890123456789012345678901234567890123456789%
802 012345678901234567890123456789012345678901234567890123456789%
803 012345678901234567890123456789012345678901234567890123456789%
804 012345678901234567890123456789012345678901234567890123456789%
805 012345678901234567890123456789012345678901234567890123456789%
806 012345678901234567890123456789012345678901234567890123456789%
807 X=X\\
808 \Test{-1}abc=abc\\
809 \Test2\par\par\relax=\relax\\
810
811 \begingroup
812   \count1=2 %
813   \Test{\count1}abc=c\\%
814 \endgroup
815
816 \ltx@ifundefined{numexpr}{%
817 }{%
818   \Test{1+1}abc=c\\%
819 }
820
821 \msg{[Test \string\ltx@CdrNum]}%
822 \long\def\Test#1=#2\\{%
823   \edef\StrA{\ltx@CdrNum#1\@nil}%
824   \expandafter\expandafter\expandafter\def
825   \expandafter\expandafter\expandafter\StrAA
826   \expandafter\expandafter\expandafter{\ltx@CdrNum#1\@nil}%
827   \edef\StrB{#2}%
828   \ifx\StrA\StrB
829     \ifx\StrAA\StrB
830       \msg{* ok.}%
831     \else
832       \msg{StrAA: \meaning\StrAA}%
833       \msg{StrB: \meaning\StrB}%
834       \errhelp{Test: #1=#2}%
835       \errmessage{Test (two expansions) failed}%
836     \fi
837   \else
838     \msg{StrA: \StrA}%
839     \msg{StrB: \StrB}%
840     \errhelp{Test: #1=#2}%
841     \errmessage{Test (edef) failed!}%
842   \fi
843 }
844 \Test0abc=abc\\
845 \Test1abc=bc\\
846 \Test2abc=c\\
847 \Test3abcd=d\\
848 \Test4abcde=e\\
849 \Test5abcdef=f\\
850 \Test6abcdefg=g\\
851 \Test7abcdefgh=h\\
852 \Test8abcdefghi=i\\
853 \Test9abcdefghij=j\\
854 \Test{10}0123456789X=X\\
855 \Test{12}abcdefghijkln=m\\
856 \Test{700}%
857 012345678901234567890123456789012345678901234567890123456789%
858 012345678901234567890123456789012345678901234567890123456789%

```

```

859 012345678901234567890123456789012345678901234567890123456789%
860 012345678901234567890123456789012345678901234567890123456789%
861 012345678901234567890123456789012345678901234567890123456789%
862 012345678901234567890123456789012345678901234567890123456789%
863 012345678901234567890123456789012345678901234567890123456789%
864 012345678901234567890123456789012345678901234567890123456789%
865 012345678901234567890123456789012345678901234567890123456789%
866 012345678901234567890123456789012345678901234567890123456789%
867 X=X\\
868 \Test{-1}abc=abc\\
869 \Test2\par\par\relax=\relax\\
870
871 \msg{[Test \string\ltx@CarNum]}%
872 \long\def\Test#1=#2\\{%
873   \edef\StrA{\ltx@CarNum#1\@nil}%
874   \expandafter\expandafter\expandafter\def
875   \expandafter\expandafter\expandafter\StrAA
876   \expandafter\expandafter\expandafter{\ltx@CarNum#1\@nil}%
877   \edef\StrB{#2}%
878   \ifx\StrA\StrB
879     \ifx\StrAA\StrB
880       \msg{* ok.}%
881     \else
882       \msg{StrAA: \meaning\StrAA}%
883       \msg{StrB: \meaning\StrB}%
884       \errhelp{Test: #1=#2}%
885       \errmessage{Test (two expansions) failed}%
886     \fi
887   \else
888     \msg{StrA: \StrA}%
889     \msg{StrB: \StrB}%
890     \errhelp{Test: #1=#2}%
891     \errmessage{Test (edef) failed!}%
892   \fi
893 }
894 \Test0abc=\\
895 \Test1abc=a\\
896 \Test2abc=ab\\
897 \Test3abc=abc\\
898 \Test3abcd=abc\\
899 \Test4abcde=abcd\\
900 \Test{10}0123456789X=0123456789\\
901 \Test{12}abcdefghijklm=abcdefghijkl\\
902 \Test{700}%
903 012345678901234567890123456789012345678901234567890123456789%
904 012345678901234567890123456789012345678901234567890123456789%
905 012345678901234567890123456789012345678901234567890123456789%
906 012345678901234567890123456789012345678901234567890123456789%
907 012345678901234567890123456789012345678901234567890123456789%
908 012345678901234567890123456789012345678901234567890123456789%
909 012345678901234567890123456789012345678901234567890123456789%
910 012345678901234567890123456789012345678901234567890123456789%
911 012345678901234567890123456789012345678901234567890123456789%
912 012345678901234567890123456789012345678901234567890123456789%
913 X=%
914 012345678901234567890123456789012345678901234567890123456789%
915 012345678901234567890123456789012345678901234567890123456789%
916 012345678901234567890123456789012345678901234567890123456789%
917 012345678901234567890123456789012345678901234567890123456789%
918 012345678901234567890123456789012345678901234567890123456789%
919 012345678901234567890123456789012345678901234567890123456789%
920 012345678901234567890123456789012345678901234567890123456789%

```

```

921 012345678901234567890123456789012345678901234567890123456789%
922 012345678901234567890123456789012345678901234567890123456789%
923 012345678901234567890123456789012345678901234567890123456789%
924 \\
925 \Test{-1}abc=\\
926 \Test2\par\par\relax=\par\par\\
927 \csname @@end\endcsname\end
928 </test-gobble>

```

### 3.3 Test \ltx@ifempty

```

929 <*test-ifempty>
930 \catcode'\{=1 %
931 \catcode'\}=2 %
932 \catcode'\#=6 %
933 \catcode'\@=11 %
934 \errorcontextlines=1000 %
935 \begingroup\expandafter\expandafter\expandafter\endgroup
936 \expandafter\ifx\csname RequirePackage\endcsname\relax
937 \input ltxcmds.sty\relax
938 \else
939 \RequirePackage{ltxcmds}[2011/04/14]%
940 \fi
941 \def\msg#{\immediate\write16}
942 \def\TestY{Y}
943 \def\TestN{N}
944 \msg{* \string\ltx@ifempty}
945 \long\def\test#1{%
946 \begingroup
947 % Calculate expected test result via macro definition
948 \def\Stuff{#1}%
949 \ifx\Stuff\ltx@empty
950 \def\StuffEmpty{Y}%
951 \else
952 \def\StuffEmpty{N}%
953 \fi
954 % Test \ltx@ifempty
955 \expandafter\expandafter\expandafter\def
956 \expandafter\expandafter\expandafter\TestEmpty
957 \expandafter\expandafter\expandafter{%
958 \ltx@ifempty{#1}{Y}{N}%
959 }%
960 \ifx\StuffEmpty\TestEmpty
961 \msg{* Test OK}%
962 \else
963 \ltx@ifundefined{detokenize}{}{%
964 \msg{Stuff: [\detokenize{\Stuff}]}%
965 }%
966 \errmessage{Test failed!}%
967 \fi
968 \endgroup
969 }
970 \test{}
971 \test{a}
972 \test{abc}
973 \test{\par}
974 \test{ }
975 \test{\if}
976 \test{{\if}}
977 \test{\else}
978 \test{{\else}}
979 \test{\fi}
980 \test{{}\fi}

```

```

981 \test{\or\ifcase}
982 \test{}}
983 \test{{a}}
984 \test{{}abc}
985 \test{{\par}}
986 \test{{}\par}

987 \def\SpaceTwo#1{%
988   \def\SpaceTwo{#1#1}%
989 }\SpaceTwo{ }
990 \msg{* \string\ltx@ifblank}
991 \long\def\test#1{%
992   \begingroup
993     % Calculate expected test result via macro definition
994     \def\Stuff{#1}%
995     \ifx\Stuff\ltx@empty
996       \def\StuffEmpty{\Y}%
997     \else
998       \ifx\Stuff\ltx@space
999         \def\StuffEmpty{\Y}%
1000     \else
1001       \ifx\Stuff\SpaceTwo
1002         \def\StuffEmpty{\Y}%
1003     \else
1004       \def\StuffEmpty{\N}%
1005     \fi
1006   \fi
1007   \fi
1008   % Test \ltx@ifblank
1009   \expandafter\expandafter\expandafter\def
1010   \expandafter\expandafter\expandafter\TestEmpty
1011   \expandafter\expandafter\expandafter{%
1012     \ltx@ifblank{#1}{\Y}{\N}%
1013   }%
1014   \ifx\StuffEmpty\TestEmpty
1015     \msg{* Test OK}%
1016   \else
1017     \ltx@ifundefined{detokenize}{}{%
1018       \msg{Stuff: [\detokenize{\Stuff}]}%
1019     }%
1020     \errmessage{Test failed!}%
1021   \fi
1022 \endgroup
1023 }

1024 \test{}
1025 \test{a}
1026 \test{\if}
1027 \test{\else}
1028 \test{\fi}
1029 \test{ \fi}
1030 \test{\par}
1031 \test{ \par}
1032 \test{}}
1033 \test{ {} }
1034 \def\x#1{%
1035   \test{#1#1}%
1036   \test{#1#1{}}%
1037   \test{#1#1\par}%
1038   \test{#1#1\else}%
1039 }\x{ }
1040 \csname @@end\endcsname\end
1041 </test-ifempty>

```

### 3.4 Test \ltx@zap@space

```

1042 <*test-zap@space>
1043 \catcode'\{=1 %
1044 \catcode'\}=2 %
1045 \catcode'\#=6 %
1046 \catcode'\@=11 %
1047 \errorcontextlines=1000 %
1048 \begingroup\expandafter\expandafter\expandafter\endgroup
1049 \expandafter\ifx\csname RequirePackage\endcsname\relax
1050 \input ltxcmds.sty\relax
1051 \else
1052 \RequirePackage{ltxcmds}[2011/04/14]%
1053 \fi
1054 \def\msg#{\immediate\write16}
1055 \def\space{ }
1056 \def\empty{}
1057 \msg{* \string\ltx@zap@space}
1058 \long\def\test#1#2{%
1059 \begingroup
1060 \def\TestInput{#1}%
1061 \def\TestExpected{#2}%
1062 % Test \ltx@zap@space
1063 \expandafter\expandafter\expandafter\def
1064 \expandafter\expandafter\expandafter\TestResult
1065 \expandafter\expandafter\expandafter{%
1066 \ltx@zap@space{#1}%
1067 }%
1068 \ifx\TestResult\TestExpected
1069 \msg{* Test OK}%
1070 \else
1071 \ltx@onelevel@sanitize\TestInput
1072 \ltx@onelevel@sanitize\TestExpected
1073 \ltx@onelevel@sanitize\TestResult
1074 \msg{* Input: \space\space\space[\TestInput]}%
1075 \msg{ \space Result: \space\space[\TestResult]}%
1076 \msg{ \space Expected: [\TestExpected]}%
1077 \errmessage{Test failed!}%
1078 \fi
1079 \endgroup
1080 }
1081 \long\def\etest#1#2{%
1082 \begingroup
1083 \edef\x{\endgroup
1084 \noexpand\test{#1}{#2}%
1085 }%
1086 \x
1087 }
1088 \catcode'\~=13 %
1089 \let~\noexpand

1090 \test{}{}
1091 \test{{}}{{}}
1092 \test{ }{}{}
1093 \test{{ }}{{ }}
1094 \test{{} }{{}}
1095 \test{ {} }{{}}
1096 \test{ { } }{{ }}
1097 \test{a {b} c}{a{b}c}
1098 \test{a bb ccc}{abbccc}
1099 \test{{a} {bb} {ccc}}{{a}{bb}{ccc}}
1100 \test{\par}{\par}
1101 \test{\if}{\if}
1102 \test{\space}{\space}

```



```

1103 \etest{\par\space\par}{\par\par}
1104 \etest{~\empty\space~\empty}{~\empty~\empty}
1105 \etest{~\fi\space~\else\space}{~\fi~\else}

1106 \csname @@end\endcsname\end
1107 </test-zapspace>

```

### 3.5 Test \ltx@ifBoxEmpty

```

1108 <*test-ifboxempty>
1109 \catcode'\{=1 %
1110 \catcode'\}=2 %
1111 \catcode'\#=6 %
1112 \catcode'\@=11 %
1113 \begingroup\expandafter\expandafter\expandafter\endgroup
1114 \expandafter\ifx\csname RequirePackage\endcsname\relax
1115 \input ltxcmds.sty\relax
1116 \else
1117 \RequirePackage{ltxcmds}[2011/04/14]%
1118 \fi
1119 \def\msg#\{\immediate\write16}
1120 % make box 0 void
1121 \begingroup
1122 \setbox0=\box0 %
1123 \endgroup
1124 \ifvoid0 %
1125 \else
1126 \errmessage{Voiding box 0 failed}%
1127 \fi
1128 \setbox2=\box0 %
1129 \def\test#1#2{%
1130 \@test{#1}{#2}%
1131 \@@test{#1}{#2}%
1132 \chardef\x=#1%
1133 \@test\x{#2}%
1134 \@@test\x{#2}%
1135 }
1136 \def\@test#1#2{%
1137 \begingroup
1138 \setbox9=\hbox{%
1139 \def\TestExpected{#2}%
1140 \ltx@ifBoxEmpty{#1}{%
1141 \def\TestResult{Y}%
1142 }{%
1143 \def\TestResult{N}%
1144 }%
1145 \ifx\TestExpected\TestResult
1146 \msg{* Test passed.}%
1147 \else
1148 \errmessage{Test failed!}%
1149 \fi
1150 }%
1151 \ifdim\wd9=0pt %
1152 \else
1153 \errmessage{Unwanted space?}%
1154 \fi
1155 \endgroup
1156 }
1157 \def\@@test#1#2{%
1158 \begingroup
1159 \setbox9=\hbox{%
1160 \def\TestExpected{#2}%
1161 \ifvoid#1\def\TestExpected{Y}\fi
1162 \ltx@ifBoxVoidOrEmpty{#1}{%

```

```

1163     \def\TestResult{Y}%
1164 }{%
1165     \def\TestResult{N}%
1166 }%
1167 \ifx\TestExpected\TestResult
1168     \msg{* Test passed.}%
1169 \else
1170     \errmessage{Test failed!}%
1171 \fi
1172 }%
1173 \ifdim\wd9=0pt %
1174 \else
1175     \errmessage{Unwanted space?}%
1176 \fi
1177 \endgroup
1178 }
1179 \test0N
1180 \test2N
1181 \setbox0=\hbox{}
1182 \test0Y
1183 \setbox2=\hbox{}
1184 \test2Y
1185 \setbox0=\vbox{}
1186 \test0Y
1187 \setbox2=\vbox{}
1188 \test0Y
1189 \setbox0=\hbox{ }%
1190 \test0N
1191 \setbox2=\hbox{ }%
1192 \test2N
1193 \setbox0=\hbox{\penalty1}%
1194 \test0N
1195 \setbox2=\hbox{\penalty1}%
1196 \test2N
1197 \csname @@end\endcsname\end
1198 </test-ifboxempty>

```

### 3.6 Test for next character detection

```

1199 <*test-nextchar>
1200 \catcode'\{=1 %
1201 \catcode'\}=2 %
1202 \catcode'\#=6 %
1203 \catcode'\@=11 %
1204 \begingroup\expandafter\expandafter\expandafter\endgroup
1205 \expandafter\ifx\csname RequirePackage\endcsname\relax
1206     \input ltxcmds.sty\relax
1207     \input eolgrab.sty\relax
1208 \else
1209     \RequirePackage{ltxcmds}[2011/04/14]%
1210     \RequirePackage{eolgrab}[2011/01/12]%
1211 \fi
1212 \def\msg#{\immediate\write16}
1213 \begingroup
1214     \def\x#1{%
1215         \endgroup
1216         \let\TestSpaceToken= #1\relax
1217     }%
1218 \x{ }
1219 \def\TestSpace{ }
1220 \begingroup
1221     \lccode32=65 % space -> A
1222 \lowercase{%

```

```

1223 \endgroup
1224 \def\TestSpaceA{ }%
1225 }
1226 \def\TestCatch{%
1227 \eolgrab\@TestCatch
1228 }
1229 \def\@TestCatch#1{%
1230 \begingroup
1231 \def\x{#1}%
1232 \ifx\x\ltx@empty
1233 \else
1234 \ltx@onelevel@sanitize\x
1235 \errmessage{Unparsed stuff on line [\x]}%
1236 \fi
1237 \endgroup
1238 }
1239 \def\TestCmdM#1{%
1240 \TestCheckType{M}%
1241 \TestCatch
1242 }
1243 \def\TestCmdOM[#1]#2{%
1244 \TestCheckType{O}%
1245 \TestCatch
1246 }
1247 \def\TestCheckType#1{%
1248 \if\TestCmdType#1\relax
1249 \else
1250 \errmessage{Wrong type #1, expected: \TestCmdType}%
1251 \fi
1252 }
1253 \def\TestCmd#1{%
1254 \def\TestCmdType{#1}%
1255 \ltx@ifnextchar[\TestCmdOM\TestCmdM
1256 }
1257 \def\TestCmdExp#1{%
1258 \expandafter\TestCmd\expandafter#1%
1259 }
1260 \outer\def\TestOuter{
1261 \TestCmd O[o]{m}
1262 \TestCmd M{m}
1263 \TestCmd O [o]{m}
1264 \TestCmd M {m}
1265 \def\x#1{\def\x{#1#1}}\x{ }
1266 \TestCmdExp O\x[o]{m}
1267 \TestCmdExp M\x{m}
1268 \def\x#1{\def\x{#1#1#1}}\x{ }
1269 \TestCmdExp O\x[o]{m}
1270 \TestCmdExp M\x{m}
1271 \def\x{\TestSpaceToken}
1272 \TestCmdExp O\x[o]{m}
1273 \TestCmdExp M\x{m}
1274 \def\x{\TestSpaceToken\TestSpaceToken\TestSpaceToken}
1275 \TestCmdExp O\x[o]{m}
1276 \TestCmdExp M\x{m}
1277 \TestCmd M\TestSpace
1278 \TestOuter
1279 \TestCmd M \TestSpace
1280 \TestOuter
1281 %
1282 \def\TestCmd#1{%
1283 \def\TestCmdType{#1}%
1284 \ltx@ifnextchar@nospace[\TestCmdOM\TestCmdM

```

```

1285 }
1286 \TestCmd O[o]{m}
1287 \TestCmd M{m}
1288 \TestCmd M [
1289 \TestOuter
1290 \TestCmd M {m}
1291 \TestOuter
1292 %
1293 \def\TestCmd#1{%
1294   \def\TestCmdType{#1}%
1295   \ltx@ifnextchar(\TestCmdPM\TestCmdM
1296 }
1297 \def\TestCmdPM(#1)#2{%
1298   \TestCheckType{P}%
1299   \TestCatch
1300 }
1301 \TestCmd P(p){m}
1302 \TestCmd M{m}
1303 \TestCmd P (p){m}
1304 \TestCmd M {m}
1305 %
1306 \def\TestCmd#1{%
1307   \def\TestCmdType{#1}%
1308   \ltx@ifnextchar{ }\TestCmdSM\TestCmdM
1309 }
1310 \def\TestCmdSM#1{%
1311   \TestCheckType{S}%
1312   \begingroup
1313     \let\x= #1\relax
1314     \ifx\x\TestSpaceToken
1315     \else
1316       \errmessage{unexpected space token: \meaning#1}%
1317     \fi
1318   \endgroup
1319   \def\TestCmdType{M}%
1320   \TestCmdM
1321 }
1322 \TestCmd S {m}
1323 \TestCmd M{m}
1324 \tracingmacros=1
1325 \def\x#1{\def\x{#1}}\x{ }
1326 \TestCmdExp S\x{m}
1327 \csname @@end\endcsname\end
1328 \</test-nextchar>

```

## 4 Installation

### 4.1 Download

**Package.** This package is available on CTAN<sup>1</sup>:

[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 T<sub>E</sub>X Files” ([CTAN:tds/tds.pdf](#)). Directories with `texmf` in their name are usually organized this way.

<sup>1</sup><http://ftp.ctan.org/tex-archive/>

## 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  $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ :

```
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):

|   |  |
|---|--|
| <code>ltxcmds.sty</code>                      | $\rightarrow$ <code>tex/generic/oberdiek/ltxcmds.sty</code>                    |
| <code>ltxcmds.pdf</code>                      | $\rightarrow$ <code>doc/latex/oberdiek/ltxcmds.pdf</code>                      |
| <code>test/ltxcmds-test1.tex</code>           | $\rightarrow$ <code>doc/latex/oberdiek/test/ltxcmds-test1.tex</code>           |
| <code>test/ltxcmds-test-gobble.tex</code>     | $\rightarrow$ <code>doc/latex/oberdiek/test/ltxcmds-test-gobble.tex</code>     |
| <code>test/ltxcmds-test-ifempty.tex</code>    | $\rightarrow$ <code>doc/latex/oberdiek/test/ltxcmds-test-ifempty.tex</code>    |
| <code>test/ltxcmds-test-zapspace.tex</code>   | $\rightarrow$ <code>doc/latex/oberdiek/test/ltxcmds-test-zapspace.tex</code>   |
| <code>test/ltxcmds-test-ifboxempty.tex</code> | $\rightarrow$ <code>doc/latex/oberdiek/test/ltxcmds-test-ifboxempty.tex</code> |
| <code>test/ltxcmds-test-nextchar.tex</code>   | $\rightarrow$ <code>doc/latex/oberdiek/test/ltxcmds-test-nextchar.tex</code>   |
| <code>ltxcmds.dtx</code>                      | $\rightarrow$ <code>source/latex/oberdiek/ltxcmds.dtx</code>                   |

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  $\mathrm{T}_{\mathrm{E}}\mathrm{X}$  distribution (`te $\mathrm{T}_{\mathrm{E}}\mathrm{X}$` , `mik $\mathrm{T}_{\mathrm{E}}\mathrm{X}$` , ...) relies on file name databases, you must refresh these. For example, `te $\mathrm{T}_{\mathrm{E}}\mathrm{X}$`  users run `texhash` or `mktextlsr`.

## 4.5 Some details for the interested

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

```
pdftk ltxcmds.pdf unpack_files output .
```

**Unpacking with  $\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$ .** The `.dtx` chooses its action depending on the format:

**plain  $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ :** Run `docstrip` and extract the files.

**$\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$ :** Generate the documentation.

If you insist on using  $\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$  for `docstrip` (really, `docstrip` does not need  $\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$ ), 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 pdfL<sup>A</sup>T<sub>E</sub>X:

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

## 5 References

- [1] Robert R. Schneck: *Re: \ifempty solution (was Macro puzzle: maximally general \ifempty)*; newsgroup `comp.text.tex`, `news:3eef1ada_6@corp.newsgroups.com`, 2003-06-17.  
<http://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$1@news.albasani.net`, 2010-11-12.  
<http://groups.google.com/group/comp.text.tex/msg/803bd57221a04996>
- [3] Ahmed Musa: *Re: TeX refuses to strip outer braces in argument*; newsgroup `comp.text.tex`, `news:f5496afe-40ed-42bd-b629-a2419ecf7c0d@o14g2000prn.googlegroups.com`, 2010-12-03.  
<http://groups.google.com/group/comp.text.tex/msg/fbf7d61a0c3a807d>
- [4] Ulrich Diez: *Re: TeX refuses to strip outer braces in argument*; newsgroup `comp.text.tex`, `news:idbo94$uka$1@four.albasani.net`, 2010-12-03.  
<http://groups.google.com/group/comp.text.tex/msg/0c230ee479487962>
- [5] Ulrich Diez: *Re: TeX refuses to strip outer braces in argument*; newsgroup `comp.text.tex`, `news:idbpu4$cg1$1@news.albasani.net`, 2010-12-03.  
<http://groups.google.com/group/comp.text.tex/msg/bbef4263390d647b>
- [6] Ulrich Diez: *Re: TeX refuses to strip outer braces in argument*; newsgroup `comp.text.tex`, `news:idd4ga$r83$1@four.albasani.net`, 2010-12-04.  
<http://groups.google.com/group/comp.text.tex/msg/00dfd1ec103cd272>
- [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.  
<http://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$3kj$1@news.eternal-september.org`, 2010-12-04.  
<http://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.  
<http://groups.google.com/group/comp.text.tex/msg/8d3cb89496a4d86d>

## 6 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`.

[2010/04/08 v1.5]

- `\ltx@cartwo`, `\ltx@cdrtwo`, `\ltx@carthree`, `\ltx@cdrthree`, `\ltx@carfour`, `\ltx@cdrfour` added.
- `\ltx@ReturnAfterFi` and `\ltx@ReturnAfterElseFi` fixed.

[2010/04/16 v1.6]

- `\ltx@leavevmode`, `\ltx@mbox` added.

[2010/04/26 v1.7]

- `\ltx@GobbleNum`, `\ltx@CdrNum`, `\ltx@CarNum` added.
- `\ltx@carzero`, `\ltx@cdrzero` added.
- `\ltx@hashchar` added.

[2010/09/11 v1.8]

- `\ltx@leftbracechar`, `\ltx@rightbracechar` added.

[2010/10/25 v1.9]

- `\ltx@LocalAppendToMacro` and `\ltx@GlobalAppendToMacro` are now `\long`.

[2010/10/31 v1.10]

- `\ltx@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.

## 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.

| Symbols                  |   |   |
|--------------------------|---|---|
| <code>\#</code>          | <i>230, 654, 753, 932, 1045, 1111, 1202</i> | <code>\@firstofone</code> ..... 663, 666            |
| <code>\\$</code>         | ..... 597, 600                              | <code>\@gobble</code> ..... 660, 668                |
| <code>\%</code>          | ..... 220, 730                              | <code>\@nil</code> .. 180, 181, 182, 183, 184, 185, |
| <code>\&amp;</code>      | ..... 598, 601                              | 186, 187, 188, 189, 205, 214,                       |
| <code>\.</code>          | ..... 462                                   | 248, 253, 270, 276, 391, 393,                       |
| <code>\@</code>          | <i>655, 728, 759, 933, 1046, 1112, 1203</i> | 587, 589, 593, 823, 826, 873, 876                   |
| <code>\@test</code>      | ..... 1131, 1134, 1157                      | <code>\@test</code> ..... 1130, 1133, 1136          |
| <code>\@TestCatch</code> | ..... 1227, 1229                            | <code>\@undefined</code> ..... 58                   |
|                          |   | <code>\</code> ..... 225, 729, 762, 784, 785,       |



|  |  |
|--|--|
| 786, 787, 788, 789, 790, 791,<br>792, 793, 794, 795, 807, 808,<br>809, 813, 818, 822, 844, 845,<br>846, 847, 848, 849, 850, 851,<br>852, 853, 854, 855, 867, 868,<br>869, 872, 894, 895, 896, 897,<br>898, 899, 900, 901, 924, 925, 926  | \endlinechar . . . . . 4, 35, 71, 77, 89   |
| \{ . 235, 652, 751, 930, 1043, 1109, 1200  | \eolgrab . . . . . 1227  |
| \} . 240, 653, 752, 931, 1044, 1110, 1201  | \errhelp . . 774, 780, 834, 840, 884, 890  |
| \~ . . . . . 1088  | \errmessage . . . . . 709,<br>775, 781, 835, 841, 885, 891,<br>966, 1020, 1077, 1126, 1148,<br>1153, 1170, 1175, 1235, 1250, 1316  |
| <b>A</b>   | \errorcontextlines . . . . . 934, 1047   |
| \advance . . . . . 693, 701, 716   | \escapechar . . . . . 246, 251, 268, 273   |
| \aftergroup . . . . . 29   | \etest . . . . . 1081, 1103, 1104, 1105  |
| <b>B</b>   | <b>F</b>   |
| \body . . . . . 672, 676   | \futurelet . . . . . 445, 459, 470   |
| \box . . . . . 488, 499, 1122, 1128  | <b>H</b>   |
| <b>C</b>   | \hbox . . . . . 486, 488, 497, 499,<br>515, 608, 619, 627, 1138, 1159,<br>1181, 1183, 1189, 1191, 1193, 1195   |
| \catcode . . . . . 2, 3, 5,<br>6, 7, 8, 9, 10, 11, 12, 13, 33, 34,<br>36, 37, 38, 39, 40, 41, 42, 43, 44,<br>45, 46, 47, 48, 49, 69, 70, 72, 73,<br>74, 78, 79, 80, 81, 82, 83, 84, 87,<br>88, 90, 91, 92, 93, 97, 99, 597,<br>598, 600, 601, 652, 653, 654,<br>655, 690, 699, 707, 711, 728,<br>729, 730, 751, 752, 753, 759,<br>930, 931, 932, 933, 1043, 1044,<br>1045, 1046, 1088, 1109, 1110,<br>1111, 1112, 1200, 1201, 1202, 1203 | <b>I</b>   |
| \chardef . . 116, 117, 118, 119, 120, 1132   | \if 253, 276, 975, 976, 1026, 1101, 1248   |
| \count . . . . . 812, 813  | \ifcase . . . . . 366, 981   |
| \count@ . . . . . 657, 686,<br>690, 692, 693, 697, 699, 700,<br>701, 705, 707, 710, 711, 715, 716  | \ifcat . . . . . 568, 579  |
| \countdef . . . . . 657  | \ifcsname . . . . . 318  |
| \csname 14, 21, 50, 66, 76, 160, 165,<br>192, 197, 254, 256, 259, 261,<br>264, 277, 279, 282, 284, 287,<br>296, 298, 306, 319, 357, 494,<br>566, 577, 656, 659, 662, 665,<br>720, 747, 754, 927, 936, 1040,<br>1049, 1106, 1114, 1197, 1205, 1327  | \ifdim . . . . . 1151, 1173  |
| <b>D</b>   | \iffalse . . 262, 285, 534, 540, 547, 556  |
| \detokenize . . . . . 568, 579, 964, 1018  | \ifhbox . . . . . 608, 619, 627  |
| \dimendef . . . . . 134, 135, 136,<br>137, 138, 139, 140, 141, 142, 143  | \ifhmode . . . . . 609, 621, 629   |
| <b>E</b>   | \ifnum . . . . . 367, 369, 371, 398,<br>613, 624, 632, 692, 700, 707, 715  |
| \empty . . . . . 17, 18, 1056, 1104  | \iftrue . . . . . 257, 280   |
| \end . . . 748, 927, 1040, 1106, 1197, 1327  | \ifvoid . . . . . 495, 604, 644, 1124, 1161  |
| \endcsname . . . . . 14, 21,<br>50, 66, 76, 162, 168, 194, 200,<br>203, 254, 256, 259, 261, 264,<br>277, 279, 282, 284, 287, 296,<br>298, 306, 318, 319, 357, 494,<br>571, 582, 656, 659, 662, 665,<br>720, 747, 754, 927, 936, 1040,<br>1049, 1106, 1114, 1197, 1205, 1327  | \ifx . . . . . 15,<br>18, 21, 50, 58, 61, 296, 298, 306,<br>319, 413, 416, 427, 430, 448,<br>451, 474, 656, 659, 662, 665,<br>720, 754, 768, 769, 828, 829,<br>878, 879, 936, 949, 960, 995,<br>998, 1001, 1014, 1049, 1068,<br>1114, 1145, 1167, 1205, 1232, 1314 |
| \endinput . . . . . 29, 115  | \immediate . . . . . 23, 52, 760, 941, 1054, 1119, 1212  |
|  | \input . . . . . 721,<br>755, 937, 1050, 1115, 1206, 1207  |
|  | \iterate . . . . . 673, 675, 677   |
|  | <b>L</b>   |
|  | \lastnodetype . . . . . 613  |
|  | \lastpenalty . . . . . 624, 632  |
|  | \lccode . . 220, 225, 230, 235, 240, 1221  |
|  | \leavevmode . . . . . 508  |
|  | \letLTXcmds@gtemp . . . . . 431  |
|  | \LoadCommand . . . . . 721, 731  |
|  | \loop . . . . . 671, 687, 698, 706   |
|  | \lowercase 221, 226, 231, 236, 241, 1222   |
|  | \ltx@(Loc, Glob) (Toks, Dimen, Skip) (A, B, C, D, E)<br>. . . . . 3  |
|  | \ltx@active . . . . . 119  |
|  | \ltx@backslashchar . . . . . 224   |
|  | \ltx@car . . . . . 4, 180  |
|  | \ltx@carfour . . . . . 4, 188  |
|  | \ltx@CarNum . . . . . 4, 190, 871, 873, 876  |
|  | \ltx@carthree . . . . . 4, 186   |

|                                       |  |  |  |
|---------------------------------------|--|--|--|
| <code>\ltx@cartwo</code>              | <a href="#">4</a> , <a href="#">184</a>  | <code>\ltx@LocalExpandAfter</code>       | <a href="#">6</a> , <a href="#">289</a> , <a href="#">295</a>  |
| <code>\ltx@carzero</code>             | <a href="#">4</a> , <a href="#">182</a>  | <code>\ltx@LocDimenA</code>              | <a href="#">134</a>  |
| <code>\ltx@cclv</code>                | <a href="#">120</a>  | <code>\ltx@LocDimenB</code>              | <a href="#">135</a>  |
| <code>\ltx@cdr</code>                 | <a href="#">181</a>  | <code>\ltx@LocDimenC</code>              | <a href="#">136</a>  |
| <code>\ltx@cdrfour</code>             | <a href="#">189</a>  | <code>\ltx@LocDimenD</code>              | <a href="#">137</a>  |
| <code>\ltx@CdrNum</code>              | <a href="#">209</a> , <a href="#">821</a> , <a href="#">823</a> , <a href="#">826</a>  | <code>\ltx@LocDimenE</code>              | <a href="#">138</a>  |
| <code>\ltx@cdrthree</code>            | <a href="#">187</a>  | <code>\ltx@LocSkipA</code>               | <a href="#">144</a>  |
| <code>\ltx@cdrtwo</code>              | <a href="#">185</a>  | <code>\ltx@LocSkipB</code>               | <a href="#">145</a>  |
| <code>\ltx@cdrzero</code>             | <a href="#">183</a>  | <code>\ltx@LocSkipC</code>               | <a href="#">146</a>  |
| <code>\ltx@clsextension</code>        | <a href="#">6</a> , <a href="#">340</a> , <a href="#">346</a> , <a href="#">384</a>  | <code>\ltx@LocSkipD</code>               | <a href="#">147</a>  |
| <code>\ltx@empty</code>               | <a href="#">5</a> , <a href="#">217</a> ,<br><a href="#">414</a> , <a href="#">417</a> , <a href="#">428</a> , <a href="#">431</a> , <a href="#">949</a> , <a href="#">995</a> , <a href="#">1232</a>  | <code>\ltx@LocSkipE</code>               | <a href="#">148</a>  |
| <code>\ltx@firstoffour</code>         | <a href="#">176</a>  | <code>\ltx@LocToksA</code>               | <a href="#">124</a> , <a href="#">421</a> ,<br><a href="#">422</a> , <a href="#">435</a> , <a href="#">436</a> , <a href="#">443</a> , <a href="#">449</a> , <a href="#">468</a> , <a href="#">475</a>   |
| <code>\ltx@firstofone</code>          | <a href="#">4</a> , <a href="#">170</a> , <a href="#">314</a> , <a href="#">464</a> , <a href="#">519</a> , <a href="#">525</a>  | <code>\ltx@LocToksB</code>               | <a href="#">125</a> , <a href="#">444</a> , <a href="#">454</a> , <a href="#">469</a> , <a href="#">477</a>  |
| <code>\ltx@firstofthree</code>        | <a href="#">173</a> , <a href="#">539</a> , <a href="#">555</a>  | <code>\ltx@LocToksC</code>               | <a href="#">126</a>  |
| <code>\ltx@firstoftwo</code>          | <a href="#">171</a> , <a href="#">299</a> , <a href="#">307</a> , <a href="#">320</a> ,<br><a href="#">325</a> , <a href="#">343</a> , <a href="#">380</a> , <a href="#">543</a> , <a href="#">559</a> , <a href="#">614</a> , <a href="#">633</a>   | <code>\ltx@LocToksD</code>               | <a href="#">127</a>  |
| <code>\ltx@fourthoffour</code>        | <a href="#">179</a>  | <code>\ltx@LocToksE</code>               | <a href="#">128</a>  |
| <code>\ltx@GlobalAppendToMacro</code> | <a href="#">7</a> , <a href="#">412</a>  | <code>\ltx@mbox</code>                   | <a href="#">8</a> , <a href="#">513</a>  |
| <code>\ltx@GlobDimenA</code>          | <a href="#">139</a>  | <code>\ltx@minusone</code>               | <a href="#">121</a>  |
| <code>\ltx@GlobDimenB</code>          | <a href="#">140</a>  | <code>\ltx@newglobalif</code>            | <a href="#">5</a> , <a href="#">266</a>  |
| <code>\ltx@GlobDimenC</code>          | <a href="#">141</a>  | <code>\ltx@newif</code>                  | <a href="#">5</a> , <a href="#">244</a>  |
| <code>\ltx@GlobDimenD</code>          | <a href="#">142</a>  | <code>\ltx@one</code>                    | <a href="#">117</a> , <a href="#">122</a> , <a href="#">620</a> , <a href="#">624</a>  |
| <code>\ltx@GlobDimenE</code>          | <a href="#">143</a>  | <code>\ltx@onelevel@sanitize</code>      | <a href="#">6</a> , <a href="#">334</a> , <a href="#">1071</a> , <a href="#">1072</a> , <a href="#">1073</a> , <a href="#">1234</a>  |
| <code>\ltx@GlobSkipA</code>           | <a href="#">149</a>  | <code>\ltx@percentchar</code>            | <a href="#">219</a>  |
| <code>\ltx@GlobSkipB</code>           | <a href="#">150</a>  | <code>\ltx@pkgextension</code>           | <a href="#">341</a> , <a href="#">349</a> , <a href="#">387</a>  |
| <code>\ltx@GlobSkipC</code>           | <a href="#">151</a>  | <code>\ltx@RemovePrefix</code>           | <a href="#">6</a> , <a href="#">330</a> , <a href="#">332</a> , <a href="#">337</a>  |
| <code>\ltx@GlobSkipD</code>           | <a href="#">152</a>  | <code>\ltx@ReturnAfterElseFi</code>      | <a href="#">216</a>  |
| <code>\ltx@GlobSkipE</code>           | <a href="#">153</a>  | <code>\ltx@ReturnAfterFi</code>          | <a href="#">5</a> , <a href="#">215</a>  |
| <code>\ltx@GlobToksA</code>           | <a href="#">129</a>  | <code>\ltx@rightbracechar</code>         | <a href="#">239</a>  |
| <code>\ltx@GlobToksB</code>           | <a href="#">130</a>  | <code>\ltx@secondoffour</code>           | <a href="#">177</a>  |
| <code>\ltx@GlobToksC</code>           | <a href="#">131</a>  | <code>\ltx@secondofthree</code>          | <a href="#">174</a>  |
| <code>\ltx@GlobToksD</code>           | <a href="#">132</a>  | <code>\ltx@secondoftwo</code>            | <a href="#">172</a> , <a href="#">301</a> , <a href="#">309</a> , <a href="#">322</a> , <a href="#">343</a> , <a href="#">363</a> ,<br><a href="#">378</a> , <a href="#">541</a> , <a href="#">557</a> , <a href="#">605</a> , <a href="#">616</a> , <a href="#">635</a> , <a href="#">638</a> |
| <code>\ltx@GlobToksE</code>           | <a href="#">133</a>  | <code>\ltx@space</code>                  | <a href="#">5</a> , <a href="#">218</a> , <a href="#">377</a> , <a href="#">998</a>  |
| <code>\ltx@gobble</code>              | <a href="#">3</a> , <a href="#">154</a> ,<br><a href="#">312</a> , <a href="#">535</a> , <a href="#">537</a> , <a href="#">548</a> , <a href="#">551</a> , <a href="#">553</a> , <a href="#">579</a>   | <code>\ltx@StripPrefix</code>            | <a href="#">331</a> , <a href="#">404</a> , <a href="#">405</a> , <a href="#">406</a>  |
| <code>\ltx@gobblefour</code>          | <a href="#">157</a>  | <code>\ltx@thirdoffour</code>            | <a href="#">178</a> , <a href="#">645</a>  |
| <code>\ltx@GobbleNum</code>           | <a href="#">3</a> , <a href="#">158</a> , <a href="#">212</a> , <a href="#">761</a> , <a href="#">763</a> , <a href="#">766</a>  | <code>\ltx@thirdofthree</code>           | <a href="#">175</a>  |
| <code>\ltx@gobblethree</code>         | <a href="#">156</a>  | <code>\ltx@two</code>                    | <a href="#">118</a> , <a href="#">628</a> , <a href="#">632</a>  |
| <code>\ltx@gobbletwo</code>           | <a href="#">155</a>  | <code>\ltx@undefined</code>              | <a href="#">413</a> , <a href="#">427</a>  |
| <code>\ltx@hashchar</code>            | <a href="#">229</a>  | <code>\ltx@zapspace</code>               | <a href="#">8</a> , <a href="#">585</a> , <a href="#">1057</a> , <a href="#">1062</a> , <a href="#">1066</a>   |
| <code>\ltx@ifblank</code>             | <a href="#">8</a> , <a href="#">545</a> , <a href="#">575</a> , <a href="#">990</a> , <a href="#">1008</a> , <a href="#">1012</a>  | <code>\ltx@zero</code>                   | <a href="#">3</a> , <a href="#">116</a> , <a href="#">206</a> , <a href="#">486</a> ,<br><a href="#">488</a> , <a href="#">490</a> , <a href="#">587</a> , <a href="#">608</a> , <a href="#">613</a> , <a href="#">619</a> , <a href="#">627</a>   |
| <code>\ltx@ifBoxEmpty</code>          | <a href="#">8</a> , <a href="#">603</a> , <a href="#">647</a> , <a href="#">1140</a>   | <code>\LTXcmds@ifnextchar</code>         | <a href="#">452</a> , <a href="#">458</a>  |
| <code>\ltx@ifBoxVoidOrEmpty</code>    | <a href="#">9</a> , <a href="#">643</a> , <a href="#">1162</a>   | <code>\LTXcmds@ParseVersion</code>       | <a href="#">391</a> , <a href="#">393</a>  |
| <code>\ltx@ifclasslater</code>        | <a href="#">7</a> , <a href="#">383</a>  | <code>\LTXcmds@AtEnd</code>              | <a href="#">95</a> , <a href="#">96</a> , <a href="#">115</a> , <a href="#">649</a>  |
| <code>\ltx@ifclassloaded</code>       | <a href="#">6</a> , <a href="#">345</a>  | <code>\LTXcmds@CarNum</code>             | <a href="#">193</a> , <a href="#">196</a>  |
| <code>\ltx@ifempty</code>             | <a href="#">8</a> , <a href="#">531</a> , <a href="#">564</a> , <a href="#">590</a> , <a href="#">944</a> , <a href="#">954</a> , <a href="#">958</a>  | <code>\LTXcmds@CarNumFinish</code>       | <a href="#">205</a>  |
| <code>\ltx@iffilelater</code>         | <a href="#">351</a> , <a href="#">384</a> , <a href="#">387</a>  | <code>\LTXcmds@CdrNum</code>             | <a href="#">211</a> , <a href="#">214</a>  |
| <code>\ltx@iffileloaded</code>        | <a href="#">6</a> , <a href="#">342</a> , <a href="#">346</a> , <a href="#">349</a> , <a href="#">352</a>  | <code>\LTXcmds@CharToken</code>          | <a href="#">442</a> , <a href="#">448</a> , <a href="#">467</a> , <a href="#">474</a>  |
| <code>\ltx@ifnextchar</code>          | <a href="#">7</a> , <a href="#">440</a> , <a href="#">1255</a> , <a href="#">1295</a> , <a href="#">1308</a>   | <code>\LTXcmds@Cm</code>                 | <a href="#">199</a>  |
| <code>\ltx@ifnextchar@nospace</code>  | <a href="#">7</a> , <a href="#">465</a> , <a href="#">1284</a>   | <code>\LTXcmds@Cx</code>                 | <a href="#">202</a>  |
| <code>\ltx@ifpackagelater</code>      | <a href="#">386</a>  | <code>\LTXcmds@Gm</code>                 | <a href="#">167</a>  |
| <code>\ltx@ifpackageloaded</code>     | <a href="#">348</a>  | <code>\LTXcmds@GobbleNum</code>          | <a href="#">161</a> , <a href="#">164</a>  |
| <code>\ltx@ifUndefined</code>         | <a href="#">5</a> ,<br><a href="#">304</a> , <a href="#">328</a> , <a href="#">389</a> , <a href="#">480</a> , <a href="#">481</a> , <a href="#">482</a> ,<br><a href="#">483</a> , <a href="#">517</a> , <a href="#">530</a> , <a href="#">596</a> , <a href="#">816</a> , <a href="#">963</a> , <a href="#">1017</a> | <code>\LTXcmds@gtemp</code>              | <a href="#">426</a> , <a href="#">427</a> , <a href="#">428</a> , <a href="#">430</a> , <a href="#">435</a> , <a href="#">436</a> , <a href="#">438</a>  |
| <code>\ltx@ifundefined</code>         | <a href="#">5</a> , <a href="#">297</a> , <a href="#">317</a> , <a href="#">328</a> , <a href="#">343</a>  | <code>\LTXcmds@ifempty</code>            | <a href="#">574</a>  |
| <code>\ltx@leavevmode</code>          | <a href="#">8</a> , <a href="#">480</a> , <a href="#">514</a>  | <code>\LTXcmds@ifempty@</code>           | <a href="#">573</a>  |
| <code>\ltx@leftbracechar</code>       | <a href="#">234</a>  | <code>\LTXcmds@ifLater</code>            | <a href="#">353</a> , <a href="#">365</a>  |
| <code>\ltx@LocalAppendToMacro</code>  | <a href="#">425</a>  | <code>\LTXcmds@ifnextchar</code>         | <a href="#">445</a> , <a href="#">447</a> , <a href="#">461</a>  |
|                                       |  | <code>\LTXcmds@ifnextchar@nospace</code> | <a href="#">470</a> , <a href="#">472</a>  |

|   |   |
|---|---|
| \LTXcmds@LetToken .....                   | \SpaceTwo .....                           |
| ..... 445, 448, 451, 460, 470, 474        | 987, 988, 989, 1001                       |
| \LTXcmds@newglobalif .....                | \StrA .....                               |
| 270, 272                                  | 763, 768,                                 |
| \LTXcmds@newif .....                      | 778, 823, 828, 838, 873, 878, 888         |
| 248, 250                                  | \StrAA .....                              |
| \LTXcmds@num .....                        | 765, 769,                                 |
| 162, 194, 517                             | 772, 825, 829, 832, 875, 879, 882         |
| \LTXcmds@ParseVersion .....               | \StrB .....                               |
| ..... 355, 361, 390, 397                  | 767, 768,                                 |
| \LTXcmds@SpaceToken .....                 | 769, 773, 779, 827, 828, 829,             |
| 451, 464                                  | 833, 839, 877, 878, 879, 883, 889         |
| \LTXcmds@temp .....                       | \Stuff .....                              |
| 531, 562                                  | 948,                                      |
| \LTXcmds@VoidBox 494, 495, 497, 499, 502  | 949, 964, 994, 995, 998, 1001, 1018       |
| \LTXcmds@zapspace .....                   | \StuffEmpty .....                         |
| 587, 589                                  | 950,                                      |
|   | 952, 960, 996, 999, 1002, 1004, 1014      |
| <b>M</b>                                  |   |
| \meaning . 337, 832, 833, 882, 883, 1316  |   |
| \msg 760, 761, 770, 772, 773, 778, 779,   |   |
| 821, 830, 832, 833, 838, 839,             |   |
| 871, 880, 882, 883, 888, 889,             |   |
| 941, 944, 961, 964, 990, 1015,            |   |
| 1018, 1054, 1057, 1069, 1074,             |   |
| 1075, 1076, 1119, 1146, 1168, 1212        |   |
| <b>N</b>                                  |   |
| \N .....                                  |   |
| 943, 952, 958, 1004, 1012                 |   |
| \next . 633, 635, 638, 640, 677, 679, 681 |   |
| \number .....                             |   |
| 354, 360, 520, 712                        |   |
| \numexpr .....                            |   |
| 526                                       |   |
| <b>O</b>                                  |   |
| \outer .....                              |   |
| 1260                                      |   |
| <b>P</b>                                  |   |
| \PackageInfo .....                        |   |
| 26  |   |
| \par .....                                |   |
| 809, 869, 926, 973, 985,                  |   |
| 986, 1030, 1031, 1037, 1100, 1103         |   |
| \pdflastmatch .....                       |   |
| 404, 405, 406                             |   |
| \pdfmatch .....                           |   |
| 398                                       |   |
| \penalty .....                            |   |
| 620, 628, 1193, 1195                      |   |
| \ProvidesPackage .....                    |   |
| 19, 67                                    |   |
| <b>Q</b>                                  |   |
| \quitvmode .....                          |   |
| 511                                       |   |
| <b>R</b>                                  |   |
| \RangeCatcodeCheck .....                  |   |
| . 704, 732, 733, 734, 735, 736,           |   |
| 737, 738, 739, 740, 741, 742, 743         |   |
| \RangeCatcodeInvalid .....                |   |
| ..... 696, 724, 725, 726, 727             |   |
| \repeat .....                             |   |
| 671, 683, 694, 702, 717                   |   |
| \RequirePackage .....                     |   |
| . 757, 939, 1052, 1117, 1209, 1210        |   |
| \RestoreCatcodes . . 685, 688, 689, 744   |   |
| \romannumeral . . 159, 162, 191, 194,     |   |
| 210, 462, 533, 546, 565, 576, 586         |   |
| <b>S</b>                                  |   |
| \setbox .....                             |   |
| 486, 488,                                 |   |
| 497, 499, 608, 619, 627, 1122,            |   |
| 1128, 1138, 1159, 1181, 1183,             |   |
| 1185, 1187, 1189, 1191, 1193, 1195        |   |
| \skipdef .....                            |   |
| 144, 145, 146,                            |   |
| 147, 148, 149, 150, 151, 152, 153         |   |
| \space . . . 710, 711, 719, 1055, 1074,   |   |
| 1075, 1076, 1102, 1103, 1104, 1105        |   |
|   | <b>T</b>                                  |
|   | \Test .....                               |
|   | 723, 746, 762, 784, 785,                  |
|   | 786, 787, 788, 789, 790, 791,             |
|   | 792, 793, 794, 795, 796, 808,             |
|   | 809, 813, 818, 822, 844, 845,             |
|   | 846, 847, 848, 849, 850, 851,             |
|   | 852, 853, 854, 855, 856, 868,             |
|   | 869, 872, 894, 895, 896, 897,             |
|   | 898, 899, 900, 901, 902, 925, 926         |
|   | \test . . 945, 970, 971, 972, 973, 974,   |
|   | 975, 976, 977, 978, 979, 980,             |
|   | 981, 982, 983, 984, 985, 986,             |
|   | 991, 1024, 1025, 1026, 1027,              |
|   | 1028, 1029, 1030, 1031, 1032,             |
|   | 1033, 1035, 1036, 1037, 1038,             |
|   | 1058, 1084, 1090, 1091, 1092,             |
|   | 1093, 1094, 1095, 1096, 1097,             |
|   | 1098, 1099, 1100, 1101, 1102,             |
|   | 1129, 1179, 1180, 1182, 1184,             |
|   | 1186, 1188, 1190, 1192, 1194, 1196        |
|   | \TestCatch . . . . 1226, 1241, 1245, 1299 |
|   | \TestCheckType .....                      |
|   | ..... 1240, 1244, 1247, 1298, 1311        |
|   | \TestCmd 1253, 1258, 1261, 1262, 1263,    |
|   | 1264, 1277, 1279, 1282, 1286,             |
|   | 1287, 1288, 1290, 1293, 1301,             |
|   | 1302, 1303, 1304, 1306, 1322, 1323        |
|   | \TestCmdExp 1257, 1266, 1267, 1269,       |
|   | 1270, 1272, 1273, 1275, 1276, 1326        |
|   | \TestCmdM .....                           |
|   | 1239, 1255, 1284, 1295, 1308, 1320        |
|   | \TestCmdOM .....                          |
|   | 1243, 1255, 1284                          |
|   | \TestCmdPM .....                          |
|   | 1295, 1297                                |
|   | \TestCmdSM .....                          |
|   | 1308, 1310                                |
|   | \TestCmdType .....                        |
|   | 1248,                                     |
|   | 1250, 1254, 1283, 1294, 1307, 1319        |
|   | \TestEmpty .....                          |
|   | 956, 960, 1010, 1014                      |
|   | \TestExpected . . . 1061, 1068, 1072,     |
|   | 1076, 1139, 1145, 1160, 1161, 1167        |
|   | \TestInput .....                          |
|   | 1060, 1071, 1074                          |
|   | \TestN .....                              |
|   | 943                                       |
|   | \TestOuter 1260, 1278, 1280, 1289, 1291   |
|   | \TestResult 1064, 1068, 1073, 1075,       |
|   | 1141, 1143, 1145, 1163, 1165, 1167        |
|   | \TestSpace .....                          |
|   | 1219, 1277, 1279                          |
|   | \TestSpaceA .....                         |
|   | 1224                                      |
|   | \TestSpaceToken 1216, 1271, 1274, 1314    |
|   | \TestY .....                              |
|   | 942                                       |

|  |   |                                 |  |
|--|---|---------------------------------|--|
| <code>\the</code> . . . . .            | 77, 78, 79,<br>80, 81, 82, 83, 84, 97, 422, 436,<br>449, 454, 473, 526, 690, 710, 711 | <code>\voidb@x</code> . . . . . | 505  |
| <b>W</b>                               |   |                                 |  |
| <code>\TMP@EnsureCode</code> . . . . . | 94, 101,<br>102, 103, 104, 105, 106, 107,<br>108, 109, 110, 111, 112, 113, 114        | <code>\wd</code> . . . . .      | 1151, 1173   |
|  |   | <code>\write</code>             | 23, 52, 760, 941, 1054, 1119, 1212   |
| <b>X</b>                               |   |                                 |  |
| <code>\toksdef</code> . . . . .        | 124, 125, 126,<br>127, 128, 129, 130, 131, 132, 133                                   | <code>\x</code> . . . . .       | 14, 15, 18, 22, 26, 28, 51,<br>56, 66, 75, 87, 1034, 1039, 1083,<br>1086, 1132, 1133, 1134, 1214,<br>1218, 1231, 1232, 1234, 1235,<br>1265, 1266, 1267, 1268, 1269,<br>1270, 1271, 1272, 1273, 1274,<br>1275, 1276, 1313, 1314, 1325, 1326 |
| <code>\tracingmacros</code> . . . . .  | 1324  |                                 |  |
| <b>U</b>                               |   |                                 |  |
| <code>\unhbox</code> . . . . .         | 490, 502, 505   |                                 |  |
| <code>\unhcopy</code> . . . . .        | 609, 621, 629   |                                 |  |
| <code>\unvcopy</code> . . . . .        | 609, 621, 629   |                                 |  |
| <b>V</b>                               |   |                                 |  |
| <code>\vbox</code> . . . . .           | 608, 619, 627, 1185, 1187   | <b>Y</b>                        |  |
|  |   | <code>\Y</code> . .             | 942, 950, 958, 996, 999, 1002, 1012  |