

# The ltxcmds package

Heiko Oberdiek  
<oberdiek@uni-freiburg.de>

2009/12/12 v1.1

## 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>2</b>
1.1	Introduction . . . . .	2
1.2	Argument killers . . . . .	2
1.3	Argument grabbers . . . . .	2
1.4	List helpers . . . . .	2
1.5	Tail recursion . . . . .	2
1.6	Empty macro . . . . .	2
1.7	Characters . . . . .	2
1.8	Command definitions . . . . .	3
<b>2</b>	<b>Implementation</b>	<b>3</b>
2.1	Identification . . . . .	3
2.2	Argument killers . . . . .	4
2.3	Argument grabbers . . . . .	5
2.4	List helpers . . . . .	5
2.5	Tail recursion . . . . .	5
2.6	Empty macro . . . . .	5
2.7	Characters . . . . .	5
2.8	Command definitions . . . . .	6
<b>3</b>	<b>Test</b>	<b>7</b>
3.1	Catcode checks for loading . . . . .	7
<b>4</b>	<b>Installation</b>	<b>8</b>
4.1	Download . . . . .	8
4.2	Bundle installation . . . . .	8
4.3	Package installation . . . . .	8
4.4	Refresh file name databases . . . . .	9
4.5	Some details for the interested . . . . .	9
<b>5</b>	<b>History</b>	<b>9</b>
	[2009/08/05 v1.0] . . . . .	9
	[2009/12/12 v1.1] . . . . .	9
<b>6</b>	<b>Index</b>	<b>10</b>

# 1 Documentation

## 1.1 Introduction

Many of my packages also support other formats such as plain-TeX. Because I am rather familiar with the utility macros from L<sup>A</sup>T<sub>E</sub>X's kernel (e.g. `\@gobble`, `\@firstoftwo`), I found myself rewriting them again and again, because they are lacking in plain-TeX.

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

## 1.2 Argument killers

<code>\ltx@gobble {⟨1⟩}</code>	→
<code>\ltx@gobbletwo {⟨1⟩} {⟨2⟩}</code>	→
<code>\ltx@gobblethree {⟨1⟩} {⟨2⟩} {⟨3⟩}</code>	→
<code>\ltx@gobblefour {⟨1⟩} {⟨2⟩} {⟨3⟩} {⟨4⟩}</code>	→

## 1.3 Argument grabbers

<code>\ltx@firstofone {⟨1⟩}</code>	→	<code>⟨1⟩</code>
<code>\ltx@firstoftwo {⟨1⟩} {⟨2⟩}</code>	→	<code>⟨1⟩</code>
<code>\ltx@secondoftwo {⟨1⟩} {⟨2⟩}</code>	→	<code>⟨2⟩</code>

## 1.4 List helpers

<code>\ltx@car {⟨1⟩} ... \@nil</code>	→	<code>⟨1⟩</code>
<code>\ltx@cdr {⟨1⟩} ... \@nil</code>	→	<code>...</code>

## 1.5 Tail recursion

<code>\ltx@ReturnAfterFi {⟨1⟩} \fi</code>	→	<code>\fi ⟨1⟩</code>
<code>\ltx@ReturnAfterElseFi {⟨1⟩} \else {⟨2⟩} \fi</code>	→	<code>\fi ⟨1⟩</code>

## 1.6 Empty macro

<code>\ltx@empty</code>	→
-------------------------	---

## 1.7 Characters

<code>\ltx@space</code>
<code>\ltx@percentchar</code>
<code>\ltx@backslashchar</code>

## 1.8 Command definitions

`\ltx@ifundefined {<cmd>} {<yes>} {<no>}`

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

`\ltx@ifundefined {<cmd>} {<yes>} {<no>}`

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

`\ltx@LocalExpandAfter`

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

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

Package identification:

```
30 \begingroup
```

```

31 \catcode35 6 % #
32 \catcode40 12 % (
33 \catcode41 12 % )
34 \catcode44 12 % ,
35 \catcode45 12 % -
36 \catcode46 12 % .
37 \catcode47 12 % /
38 \catcode58 12 % :
39 \catcode64 11 % @
40 \catcode91 12 % [
41 \catcode93 12 % ]
42 \catcode123 1 % {
43 \catcode125 2 % }
44 \expandafter\ifx\csname ProvidesPackage\endcsname\relax
45   \def\x#1#2#3[#4]{\endgroup
46     \immediate\write-1{Package: #3 #4}%
47     \xdef#1{#4}%
48   }%
49 \else
50   \def\x#1#2[#3]{\endgroup
51     #2[#{#3}]%
52     \ifx#1\@undefined
53       \xdef#1{#3}%
54     \fi
55     \ifx#1\relax
56       \xdef#1{#3}%
57     \fi
58   }%
59 \fi
60 \expandafter\x\csname ver@ltxcmds.sty\endcsname
61 \ProvidesPackage{ltxcmds}%
62 [2009/12/12 v1.1 LaTeX kernel commands for general use (H0)]
63 \begingroup
64 \catcode123 1 % {
65 \catcode125 2 % }
66 \def\x{\endgroup
67   \expandafter\edef\csname ltx@AtEnd\endcsname{%
68     \catcode35 \the\catcode35\relax
69     \catcode64 \the\catcode64\relax
70     \catcode123 \the\catcode123\relax
71     \catcode125 \the\catcode125\relax
72   }%
73 }%
74 \x
75 \catcode35 6 % #
76 \catcode64 11 % @
77 \catcode123 1 % {
78 \catcode125 2 % }
79 \def\TMP@EnsureCode#1#2{%
80   \edef\ltx@AtEnd{%
81     \ltx@AtEnd
82     \catcode#1 \the\catcode#1\relax
83   }%
84   \catcode#1 #2\relax
85 }
86 \TMP@EnsureCode{61}{12}% =
87 \TMP@EnsureCode{96}{12}% ‘

```

## 2.2 Argument killers

\ltx@gobble

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

```

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

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

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

```

## 2.3 Argument grabbers

```

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

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

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

```

## 2.4 List helpers

```

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

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

```

## 2.5 Tail recursion

```

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

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

```

## 2.6 Empty macro

```

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

```

## 2.7 Characters

```

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

\ltx@percentchar
101 \begingroup
102 \lccode'0='%\relax
103 \lowercase{\endgroup
104 \def\ltx@percentchar{0}%
105 }

\ltx@backslashchar
106 \begingroup
107 \lccode'0='\\relax
108 \lowercase{\endgroup
109 \def\ltx@backslashchar{0}%
110 }

```

## 2.8 Command definitions

```

\ltx@LocalExpandAfter

111 \def\ltx@LocalExpandAfter{%
112   \begingroup
113     \expandafter\expandafter\expandafter
114   \endgroup
115   \expandafter
116 }

117 \ltx@LocalExpandAfter
118 \ifx\csname ifcsname\endcsname\relax

\ltx@ifundefined

119   \def\ltx@ifundefined#1{%
120     \expandafter\ifx\csname #1\endcsname\relax
121       \expandafter\ltx@firstoftwo
122     \else
123       \expandafter\ltx@secondoftwo
124     \fi
125   }%

\ltx@ifUndefined

126   \def\ltx@ifUndefined#1{%
127     \begingroup\expandafter\expandafter\expandafter\endgroup
128     \expandafter\ifx\csname #1\endcsname\relax
129       \expandafter\ltx@firstoftwo
130     \else
131       \expandafter\ltx@secondoftwo
132     \fi
133   }%

134   \expandafter\ltx@gobble
135 \else
136   \expandafter\ltx@firstofone
137 \fi
138 {%

\ltx@ifundefined

139   \def\ltx@ifundefined#1{%
140     \ifcsname #1\endcsname
141       \expandafter\ifx\csname #1\endcsname\relax
142         \expandafter\expandafter\expandafter\ltx@firstoftwo
143       \else
144         \expandafter\expandafter\expandafter\ltx@secondoftwo
145       \fi
146     \else
147       \expandafter\ltx@firstoftwo
148     \fi
149   }%

\ltx@ifUndefined

150   \let\ltx@ifUndefined\ltx@ifundefined

151 }

152 \ltx@AtEnd
153 \endpackage

```

## 3 Test

### 3.1 Catcode checks for loading

```
154 (*test1)
155 \catcode'\{=1 %
156 \catcode'\}=2 %
157 \catcode'\#=6 %
158 \catcode'\@=11 %
159 \expandafter\ifx\csname count@\endcsname\relax
160   \countdef\count@=255 %
161 \fi
162 \expandafter\ifx\csname @gobble\endcsname\relax
163   \long\def\@gobble#1{}%
164 \fi
165 \expandafter\ifx\csname @firstofone\endcsname\relax
166   \long\def\@firstofone#1{#1}%
167 \fi
168 \expandafter\ifx\csname loop\endcsname\relax
169   \expandafter\@firstofone
170 \else
171   \expandafter\@gobble
172 \fi
173 {%
174   \def\loop#1\repeat{%
175     \def\body{#1}%
176     \iterate
177   }%
178   \def\iterate{%
179     \body
180     \let\next\iterate
181   \else
182     \let\next\relax
183   \fi
184   \next
185 }%
186 \let\repeat=\fi
187 }%
188 \def\RestoreCatcodes{}
189 \count@=0 %
190 \loop
191   \edef\RestoreCatcodes{%
192     \RestoreCatcodes
193     \catcode\the\count@=\the\catcode\count@\relax
194   }%
195 \ifnum\count@<255 %
196   \advance\count@ 1 %
197 \repeat
198
199 \def\RangeCatcodeInvalid#1#2{%
200   \count@=#1\relax
201   \loop
202     \catcode\count@=15 %
203   \ifnum\count@<#2\relax
204     \advance\count@ 1 %
205   \repeat
206 }
207 \expandafter\ifx\csname LoadCommand\endcsname\relax
208   \def\LoadCommand{\input ltxcmds.sty\relax}%
209 \fi
210 \def\Test{%
211   \RangeCatcodeInvalid{0}{47}%
```

```

212 \RangeCatcodeInvalid{58}{64}%
213 \RangeCatcodeInvalid{91}{96}%
214 \RangeCatcodeInvalid{123}{255}%
215 \catcode'\@=12 %
216 \catcode'\=0 %
217 \catcode'\{=1 %
218 \catcode'\}=2 %
219 \catcode'\#=6 %
220 \catcode'\[=12 %
221 \catcode'\]=12 %
222 \catcode'\%=14 %
223 \catcode'\ =10 %
224 \catcode13=5 %
225 \LoadCommand
226 \RestoreCatcodes
227 }
228 \Test
229 \csname @@end\endcsname
230 \end
231 </test1>

```

## 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.

### 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-T<sub>E</sub>X:

```
tex ltxcmds.dtx
```

---

<sup>1</sup>[ftp://ftp.ctan.org/tex-archive/](http://ftp.ctan.org/tex-archive/)



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

```
ltxcmds.sty      → tex/generic/oberdiek/ltxcmds.sty
ltxcmds.pdf      → doc/latex/oberdiek/ltxcmds.pdf
test/ltxcmds-test1.tex → doc/latex/oberdiek/test/ltxcmds-test1.tex
ltxcmds.dtx      → source/latex/oberdiek/ltxcmds.dtx
```

If you have a `docstrip.cfg` that configures and enables `docstrip`'s TDS installing feature, then some files can already be in the right place, see the documentation of `docstrip`.

## 4.4 Refresh file name databases

If your  $\TeX$  distribution (`te $\TeX$` , `mik $\TeX$` , ...) relies on file name databases, you must refresh these. For example, `te $\TeX$`  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  $\LaTeX$ .** The `.dtx` chooses its action depending on the format:

**plain- $\TeX$ :** Run `docstrip` and extract the files.

**$\LaTeX$ :** Generate the documentation.

If you insist on using  $\LaTeX$  for `docstrip` (really, `docstrip` does not need  $\LaTeX$ ), then inform the autodetect routine about your intention:

```
latex \let\install=y\input{ltxcmds.dtx}
```

Do not forget to quote the argument according to the demands of your shell.

**Generating the documentation.** You can use both the `.dtx` or the `.drv` to generate the documentation. The process can be configured by the configuration file `ltxdoc.cfg`. For instance, put this line into this file, if you want to have A4 as paper format:

```
\PassOptionsToClass{a4paper}{article}
```

An example follows how to generate the documentation with `pdf $\LaTeX$` :

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

# 5 History

[2009/08/05 v1.0]

- First version.

[2009/12/12 v1.1]

- Short title shortened.

## 6 Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in *roman* refer to the code lines where the entry is used.

Symbols		
\#	157, 219	\input . . . . . 208
\%	102, 222	\iterate . . . . . 176, 178, 180
\@	158, 215	
\@firstofone	166, 169	<b>L</b>
\@gobble	163, 171	\lccode . . . . . 102, 107
\@nil	95, 96	\LoadCommand . . . . . 208, 225
\@undefined	52	\loop . . . . . 174, 190, 201
\[	220	\lowercase . . . . . 103, 108
\\	107, 216	\ltx@AtEnd . . . . . 80, 81, 152
\{	155, 217	\ltx@backslashchar . . . . . 106
\}	156, 218	\ltx@car . . . . . 2, 95
\]	221	\ltx@cdr . . . . . 96
		\ltx@empty . . . . . 2, 99
\_	223	\ltx@firstofone . . . . . 2, 92, 136
		\ltx@firstoftwo . . . . . 93, 121, 129, 142, 147
<b>A</b>		\ltx@gobble . . . . . 2, 88, 134
\advance	196, 204	\ltx@gobblefour . . . . . 91
\aftergroup	26	\ltx@gobblethree . . . . . 90
		\ltx@gobbletwo . . . . . 89
<b>B</b>		\ltx@ifundefined . . . . . 3, 126, 150
\body	175, 179	\ltx@ifundefined . . . . . 3, 119, 139, 150
		\ltx@LocalExpandAfter . . . . . 3, 111, 117
<b>C</b>		\ltx@percentchar . . . . . 101
\catcode	3, 4, 5, 6, 7, 8, 9, 17, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 64, 65, 68, 69, 70, 71, 75, 76, 77, 78, 82, 84, 155, 156, 157, 158, 193, 202, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224	\ltx@ReturnAfterElseFi . . . . . 98
\count@	160, 189, 193, 195, 196, 200, 202, 203, 204	\ltx@ReturnAfterFi . . . . . 2, 97, 98
\countdef	160	\ltx@secondoftwo . . . . . 94, 123, 131, 144
\csname	10, 18, 44, 60, 67, 118, 120, 128, 141, 159, 162, 165, 168, 207, 229	\ltx@space . . . . . 2, 100
<b>E</b>		<b>N</b>
\empty	13, 14	\next . . . . . 180, 182, 184
\end	230	
\endcsname	10, 18, 44, 60, 67, 118, 120, 128, 140, 141, 159, 162, 165, 168, 207, 229	<b>P</b>
\endinput	26	\PackageInfo . . . . . 23
		\ProvidesPackage . . . . . 15, 61
<b>I</b>		
\ifcsname	140	<b>R</b>
\ifnum	195, 203	\RangeCatcodeInvalid . . . . .
\ifx	11, 14, 18, 44, 52, 55, 118, 120, 128, 141, 159, 162, 165, 168, 207	. . . . . 199, 211, 212, 213, 214
\immediate	20, 46	\repeat . . . . . 174, 186, 197, 205
		\RestoreCatcodes . . . . . 188, 191, 192, 226
		<b>T</b>
		\Test . . . . . 210, 228
		\the . . . . . 68, 69, 70, 71, 82, 193
		\TMP@EnsureCode . . . . . 79, 86, 87
		<b>W</b>
		\write . . . . . 20, 46
		<b>X</b>
		\x 10, 11, 14, 19, 23, 25, 45, 50, 60, 66, 74