

# The ltxcmds package

Heiko Oberdiek  
<heiko.oberdiek at gmail.com>

2010/04/16 v1.6

## 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	Numbers . . . . .	2
1.3	Argument killers . . . . .	2
1.4	Argument grabbers . . . . .	3
1.5	List helpers . . . . .	3
1.6	Tail recursion . . . . .	3
1.7	Empty macro . . . . .	3
1.8	Characters . . . . .	3
1.9	Boolean switch . . . . .	4
1.10	Command definitions . . . . .	4
1.11	Stripping . . . . .	4
1.12	File management . . . . .	4
	1.12.1 File extensions . . . . .	4
	1.12.2 Load check . . . . .	5
	1.12.3 Version date check . . . . .	5
1.13	Macro additions . . . . .	5
1.14	Macro \ltx@ifnextchar . . . . .	5
1.15	\ltx@leavevmode, \ltx@mbbox . . . . .	6
<b>2</b>	<b>Implementation</b>	<b>6</b>
2.1	Identification . . . . .	6
2.2	Numbers . . . . .	7
2.3	Argument killers . . . . .	8
2.4	Argument grabbers . . . . .	8
2.5	List helpers . . . . .	8
2.6	Tail recursion . . . . .	9
2.7	Empty macro . . . . .	9
2.8	Characters . . . . .	9
2.9	Boolean switch . . . . .	9
2.10	Command definitions . . . . .	10
2.11	Stripping . . . . .	10
2.12	File management . . . . .	11
	2.12.1 File extensions . . . . .	11
	2.12.2 Load check . . . . .	11
	2.12.3 Version date check . . . . .	11
2.13	Macro additions . . . . .	12

2.14 Macro <code>\ltx@ifnextchar</code> . . . . .	13
2.15 <code>\ltx@leavevmode</code> , <code>\ltx@mbox</code> . . . . .	14
<b>3 Test</b> . . . . .	<b>14</b>
3.1 Catcode checks for loading . . . . .	14
<b>4 Installation</b> . . . . .	<b>16</b>
4.1 Download . . . . .	16
4.2 Bundle installation . . . . .	16
4.3 Package installation . . . . .	16
4.4 Refresh file name databases . . . . .	17
4.5 Some details for the interested . . . . .	17
<b>5 History</b> . . . . .	<b>17</b>
[2009/08/05 v1.0] . . . . .	17
[2009/12/12 v1.1] . . . . .	17
[2010/01/28 v1.2] . . . . .	17
[2010/03/01 v1.3] . . . . .	18
[2010/03/09 v1.4] . . . . .	18
[2010/04/08 v1.5] . . . . .	18
[2010/04/16 v1.6] . . . . .	18
<b>6 Index</b> . . . . .	<b>18</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. `\@gobble`, `\@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 `\ltx@`. This avoids also faulty redefinitions. I remember an example where a package redefined `\@firstoftwo` with forgetting `\long`.

## 1.2 Numbers

```
\ltx@zero
\ltx@one
\ltx@two
\ltx@ccclv
```

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

## 1.3 Argument killers

```
\ltx@gobble {\langle 1 \rangle} →
\ltx@gobbletwo {\langle 1 \rangle} {\langle 2 \rangle} →
\ltx@gobblethree {\langle 1 \rangle} {\langle 2 \rangle} {\langle 3 \rangle} →
\ltx@gobblefour {\langle 1 \rangle} {\langle 2 \rangle} {\langle 3 \rangle} {\langle 4 \rangle} →
```

## 1.4 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>	$\rightarrow$	$\langle 2 \rangle$

## 1.5 List helpers

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

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

<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>	$\rightarrow$	$\dots$

<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>	$\rightarrow$	$\dots$

## 1.6 Tail recursion

<code>\ltx@ReturnAfterFi {⟨1⟩} \fi</code>	$\rightarrow$	$\backslash\mathrm{fi} \langle 1 \rangle$
<code>\ltx@ReturnAfterElseFi {⟨1⟩} \else {⟨2⟩} \fi</code>	$\rightarrow$	$\backslash\mathrm{fi} \langle 1 \rangle$

## 1.7 Empty macro

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

## 1.8 Characters

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

## 1.9 Boolean switch

`\ltx@newif {<cmd>}`

`\ltx@newif` defines a new boolean switch `<cmd>` like `\newif`. Unlike plain  $\TeX$ 's `\newif`, `\ltx@newif` is not `\outer`. The command `<cmd>` must start with the two characters `if`.

## 1.10 Command definitions

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

If  $\varepsilon\text{-}\TeX$  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  $\varepsilon\text{-}\TeX$  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  $\varepsilon\text{-}\TeX$ .

`\ltx@LocalExpandAfter`

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

## 1.11 Stripping

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

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

## 1.12 File management

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

### 1.12.1 File extensions

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

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

### 1.12.2 Load check

<code>\ltx@ifclassloaded {&lt;class&gt;} {&lt;yes&gt;} {&lt;no&gt;}</code> <code>\ltx@ifpackageloaded {&lt;package&gt;} {&lt;yes&gt;} {&lt;no&gt;}</code>
--

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

<code>\ltx@iffileloaded {&lt;file&gt;} {&lt;yes&gt;} {&lt;no&gt;}</code>
--

If L<sup>A</sup>T<sub>E</sub>X's `\ProvidesFile` macro was called before using  $\langle file \rangle$  as argument, then `\ltx@iffileloaded` calls  $\langle yes \rangle$ , otherwise  $\langle no \rangle$ . Therefore it is possible that the  $\langle file \rangle$  is loaded, but  $\langle no \rangle$  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 T<sub>E</sub>X's default extension `.tex` was given in the first command, then it must also be specified in the latter command and vice versa.

### 1.12.3 Version date check

<code>\ltx@ifclasslater {&lt;class&gt;} {&lt;date&gt;} {&lt;yes&gt;} {&lt;no&gt;}</code> <code>\ltx@ifpackagelater {&lt;package&gt;} {&lt;date&gt;} {&lt;yes&gt;} {&lt;no&gt;}</code> <code>\ltx@iffilelater {&lt;file&gt;} {&lt;date&gt;} {&lt;yes&gt;} {&lt;no&gt;}</code>
--

If a `\ProvidesClass`/`\ProvidesPackage`/`\ProvidesFile` command with exact the same class/package/file was executed before with an optional argument that starts with a L<sup>A</sup>T<sub>E</sub>X version date, then this version date is compared with the argument  $\langle date \rangle$ . If they are equal or if the version date is the later date, then  $\langle yes \rangle$  is called. In all other cases  $\langle no \rangle$  is executed.

A L<sup>A</sup>T<sub>E</sub>X 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 pdfT<sub>E</sub>X'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 L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> in 1994.

## 1.13 Macro additions

<code>\ltx@GlobalAppendToMacro {&lt;cmd&gt;} {&lt;addition&gt;}</code> <code>\ltx@LocalAppendToMacro {&lt;cmd&gt;} {&lt;addition&gt;}</code>
---

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

### 1.14 Macro `\ltx@ifnextchar`

<code>\ltx@ifnextchar {&lt;char&gt;} {&lt;yes&gt;} {&lt;no&gt;}</code>
--

If next character is  $\langle char \rangle$  then  $\langle yes \rangle$  is called, otherwise  $\langle no \rangle$ . The character is not removed.

## 1.15 \ltx@leavevmode, \ltx@mbox

`\ltx@leavevmode`

Macro `\ltx@leavevmode` calls pdfTeX's `\quitvmode`. Otherwise it uses `\leavevmode` and defines it if necessary.

`\ltx@mbox`

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

## 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 [2010/04/16 v1.6 LaTeX kernel commands for general use (H0)]
63 \begingroup
64 \catcode123 1 % {
65 \catcode125 2 % }
66 \def\x{\endgroup
67 \expandafter\edef\csname LTXcmds@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\LTXcmds@AtEnd{%
81 \LTXcmds@AtEnd
82 \catcode#1 \the\catcode#1\relax
83 }%
84 \catcode#1 #2\relax
85 }
86 \TMP@EnsureCode{40}{12}% (
87 \TMP@EnsureCode{41}{12}% )
88 \TMP@EnsureCode{45}{12}% -
89 \TMP@EnsureCode{46}{12}% .
90 \TMP@EnsureCode{47}{12}% /
91 \TMP@EnsureCode{60}{12}% <
92 \TMP@EnsureCode{61}{12}% =
93 \TMP@EnsureCode{62}{12}% >
94 \TMP@EnsureCode{91}{12}% [
95 \TMP@EnsureCode{96}{12}% '
96 \TMP@EnsureCode{93}{12}% ]
97 \TMP@EnsureCode{94}{12}% ^ (superscript) (!)
98 \TMP@EnsureCode{124}{12}% |

```

## 2.2 Numbers

\ltx@zero

```

99 \chardef\ltx@zero=0 %

```

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

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

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

## 2.3 Argument killers

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

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

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

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

## 2.4 Argument grabbers

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

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

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

## 2.5 List helpers

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

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

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

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

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

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

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

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



## 2.6 Tail recursion

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

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

## 2.7 Empty macro

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

## 2.8 Characters

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

```
\ltx@percentchar
122 \begingroup
123 \lccode'0='%\relax
124 \lowercase\endgroup
125 \def\ltx@percentchar{0}%
126 }
```

```
\ltx@backslashchar
127 \begingroup
128 \lccode'0='\\relax
129 \lowercase\endgroup
130 \def\ltx@backslashchar{0}%
131 }
```

## 2.9 Boolean switch

```
\ltx@newif
132 \def\ltx@newif#1{%
133 \begingroup
134 \escapechar=-1 %
135 \expandafter\endgroup
136 \expandafter\LTxcmds@newif\string#1\@nil
137 }
```

```
\LTxcmds@newif
138 \begingroup
139 \escapechar=-1 %
140 \expandafter\endgroup
141 \expandafter\def\expandafter\LTxcmds@newif\string\if#1\@nil{%
142 \expandafter\edef\csname#1true\endcsname{%
143 \let
144 \expandafter\noexpand\csname if#1\endcsname
145 \noexpand\iftrue
146 }%
147 \expandafter\edef\csname#1false\endcsname{%
148 \let
149 \expandafter\noexpand\csname if#1\endcsname
150 \noexpand\iffalse
151 }%
152 \csname#1false\endcsname
153 }
```

## 2.10 Command definitions

```
\ltx@LocalExpandAfter

154 \def\ltx@LocalExpandAfter{%
155   \begingroup
156   \expandafter\expandafter\expandafter
157   \endgroup
158   \expandafter
159 }

160 \ltx@LocalExpandAfter
161 \ifx\csname ifcsname\endcsname\relax

\ltx@ifundefined

162 \def\ltx@ifundefined#1{%
163   \expandafter\ifx\csname #1\endcsname\relax
164   \expandafter\ltx@firstoftwo
165   \else
166   \expandafter\ltx@secondoftwo
167   \fi
168 }%

\ltx@ifUndefined

169 \def\ltx@ifUndefined#1{%
170   \begingroup\expandafter\expandafter\expandafter\endgroup
171   \expandafter\ifx\csname #1\endcsname\relax
172   \expandafter\ltx@firstoftwo
173   \else
174   \expandafter\ltx@secondoftwo
175   \fi
176 }%

177 \expandafter\ltx@gobble
178 \else
179 \expandafter\ltx@firstofone
180 \fi
181 {%

\ltx@ifundefined

182 \def\ltx@ifundefined#1{%
183   \ifcsname #1\endcsname
184   \expandafter\ifx\csname #1\endcsname\relax
185   \expandafter\expandafter\expandafter\ltx@firstoftwo
186   \else
187   \expandafter\expandafter\expandafter\ltx@secondoftwo
188   \fi
189   \else
190   \expandafter\ltx@firstoftwo
191   \fi
192 }%

\ltx@ifUndefined

193 \let\ltx@ifUndefined\ltx@ifundefined
194 }
```

## 2.11 Stripping

```
\ltx@RemovePrefix

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

```

\ltx@StripPrefix
196 \def\ltx@StripPrefix{%
197   \expandafter\ltx@RemovePrefix
198 }

```

## 2.12 File management

### 2.12.1 File extensions

```

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

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

```

### 2.12.2 Load check

```

\ltx@iffileloaded
201 \def\ltx@iffileloaded#1{%
202   \ltx@ifundefined{ver@#1}\ltx@secondoftwo\ltx@firstoftwo
203 }

\ltx@ifclassloaded
204 \def\ltx@ifclassloaded#1{%
205   \ltx@iffileloaded{#1.\ltx@clsextension}%
206 }

\ltx@ifpackageloaded
207 \def\ltx@ifpackageloaded#1{%
208   \ltx@iffileloaded{#1.\ltx@pkgextension}%
209 }

```

### 2.12.3 Version date check

```

\ltx@iffilelater
210 \def\ltx@iffilelater#1#2{%
211   \ltx@iffileloaded{#1}{%
212     \expandafter\LTXcmds@IfLater\expandafter{%
213       \number
214       \expandafter\expandafter\expandafter\LTXcmds@ParseVersion
215       \expandafter\expandafter\expandafter{%
216         \csname ver@#1\endcsname
217       }%
218     \expandafter}\expandafter{%
219       \number
220       \expandafter\LTXcmds@ParseVersion\expandafter{#2}%
221     }%
222   }\ltx@secondoftwo
223 }

\LTXcmds@IfLater
224 \def\LTXcmds@IfLater#1#2{%
225   \ifcase 0%
226     \ifnum#1<19940101 %
227   \else
228     \ifnum#2<19940101 %
229   \else
230     \ifnum#2>#1 %
231   \else
232     1%

```

```

233         \fi
234     \fi
235     \fi
236     \ltx@space
237     \expandafter\ltx@secondoftwo
238 \else
239     \expandafter\ltx@firstoftwo
240 \fi
241 }

\ltx@ifclasslater
242 \def\ltx@ifclasslater#1{%
243     \ltx@ifclasslater{#1.\ltx@clsextension}%
244 }

\ltx@ifpackagelater
245 \def\ltx@ifpackagelater#1{%
246     \ltx@iffilelater{#1.\ltx@pkgextension}%
247 }

248 \ltx@ifundefined{pdfmatch}{%

\LTxcmds@ParseVersion
249     \def\LTxcmds@ParseVersion#1{%
250         \LTxcmds@@ParseVersion#10000/00/00\@nil
251     }%

\LTxcmds@@ParseVersion
252     \def\LTxcmds@@ParseVersion#1#2#3#4/#5#6/#7#8#9\@nil{%
253         #1#2#3#4#5#6#7#8%
254     }%

255 }{%

\LTxcmds@ParseVersion
256     \def\LTxcmds@ParseVersion#1{%
257         \ifnum\pdfmatch{%
258             ~%
259             (199[4-9] | [2-9] [0-9] [0-9] [0-9])/%
260             (0[1-9] | 1[0-2])/%
261             (0[1-9] | [1-2] [0-9] | 3[0-1])%
262         }{#1}=1 %
263         \ltx@StripPrefix\pdfmatch1 %
264         \ltx@StripPrefix\pdfmatch2 %
265         \ltx@StripPrefix\pdfmatch3 %
266     \else
267         0%
268     \fi
269 }%
270 }

```

## 2.13 Macro additions

```

\ltx@GlobalAppendToMacro
271 \def\ltx@GlobalAppendToMacro#1#2{%
272     \ifx\ltx@undefined#1%
273         \let#1\ltx@empty
274     \else
275         \ifx\relax#1%
276             \let#1\ltx@empty
277         \fi

```

```

278 \fi
279 \begingroup
280 \toks0\expandafter{#1#2}%
281 \xdef#1{\the\toks0}%
282 \endgroup
283 }

```

`\ltx@LocalAppendToMacro`

```

284 \def\ltx@LocalAppendToMacro#1#2{%
285 \global\let\LTxcmds@gtemp#1%
286 \ifx\ltx@undefined\LTxcmds@gtemp
287 \global\let\LTxcmds@gtemp\ltx@empty
288 \else
289 \ifx\relax\LTxcmds@gtemp
290 \global\let\LTxcmds@gtemp\ltx@empty
291 \fi
292 \fi
293 \begingroup
294 \toks0\expandafter{\LTxcmds@gtemp#2}%
295 \xdef\LTxcmds@gtemp{\the\toks0}%
296 \endgroup
297 \let#1\LTxcmds@gtemp
298 }

```

## 2.14 Macro `\ltx@ifnextchar`

`\ltx@ifnextchar`

```

299 \long\def\ltx@ifnextchar#1#2#3{%
300 \begingroup
301 \let\LTxcmds@CharToken= #1\relax
302 \toks\ltx@zero{#2}%
303 \toks\ltx@two{#3}%
304 \futurelet\LTxcmds@LetToken\LTxcmds@ifnextchar
305 }

```

`\LTxcmds@ifnextchar`

```

306 \def\LTxcmds@ifnextchar{%
307 \ifx\LTxcmds@LetToken\LTxcmds@CharToken
308 \expandafter\endgroup\the\toks\expandafter\ltx@zero
309 \else
310 \ifx\LTxcmds@LetToken\LTxcmds@SpaceToken
311 \expandafter\expandafter\expandafter\LTxcmds@@ifnextchar
312 \else
313 \expandafter\endgroup\the\toks
314 \expandafter\expandafter\expandafter\ltx@two
315 \fi
316 \fi
317 }

```

`\LTxcmds@@ifnextchar`

```

318 \begingroup
319 \def\x#1{\endgroup
320 \def\LTxcmds@@ifnextchar#1{%
321 \futurelet\LTxcmds@LetToken\LTxcmds@ifnextchar
322 }%
323 }%
324 \x{ }

```

`\LTxcmds@SpaceToken`

```

325 \begingroup
326 \def\x#1{\endgroup

```

```

327 \let\LTxcmds@SpaceToken= #1%
328 }%
329 \x{ }

```

## 2.15 \ltx@leavevmode, \ltx@mbox

\ltx@leavevmode

```

330 \ltx@ifundefined{quitvmode}{%
331 \ltx@ifundefined{leavevmode}{%
332 \ltx@ifundefined{voidb@x}{%
333 \ltx@ifundefined{newbox}{%
334 \def\ltx@leavevmode{%
335 \begingroup
336 \setbox\ltx@zero=\hbox{}%
337 \begingroup
338 \setbox\ltx@zero=\hbox{\box\ltx@zero}%
339 \endgroup
340 \unhbox\ltx@zero
341 \endgroup
342 }%
343 }{%
344 \csname newbox\endcsname\LTxcmds@VoidBox
345 \ifvoid\LTxcmds@VoidBox
346 \else
347 \setbox\LTxcmds@VoidBox=\hbox{}%
348 \begingroup
349 \setbox\LTxcmds@VoidBox=\hbox{\box\LTxcmds@VoidBox}%
350 \endgroup
351 \fi
352 \def\ltx@leavevmode{\unhbox\LTxcmds@VoidBox}%
353 }%
354 }{%
355 \def\ltx@leavevmode{\unhbox\voidb@x}%
356 }%
357 }{%
358 \let\ltx@leavevmode\leavevmode
359 }%
360 }{%
361 \let\ltx@leavevmode\quitvmode
362 }

```

\ltx@mbox

```

363 \def\ltx@mbox{%
364 \ltx@leavevmode
365 \hbox
366 }

```

```

367 \LTxcmds@AtEnd
368 </package>

```

## 3 Test

### 3.1 Catcode checks for loading

```

369 (*test1)
370 \catcode'\{=1 %
371 \catcode'\}=2 %
372 \catcode'\#=6 %
373 \catcode'\@=11 %
374 \expandafter\ifx\csname count\endcsname\relax
375 \countdef\count@=255 %

```

```

376 \fi
377 \expandafter\ifx\csname @gobble\endcsname\relax
378   \long\def\@gobble#1{%
379 \fi
380 \expandafter\ifx\csname @firstofone\endcsname\relax
381   \long\def\@firstofone#1{#1}%
382 \fi
383 \expandafter\ifx\csname loop\endcsname\relax
384   \expandafter\@firstofone
385 \else
386   \expandafter\@gobble
387 \fi
388 {%
389   \def\loop#1\repeat{%
390     \def\body{#1}%
391     \iterate
392   }%
393   \def\iterate{%
394     \body
395     \let\next\iterate
396   \else
397     \let\next\relax
398   \fi
399   \next
400 }%
401 \let\repeat=\fi
402 }%
403 \def\RestoreCatcodes{}
404 \count@=0 %
405 \loop
406   \edef\RestoreCatcodes{%
407     \RestoreCatcodes
408     \catcode\the\count@=\the\catcode\count@\relax
409   }%
410 \ifnum\count@<255 %
411   \advance\count@ 1 %
412 \repeat
413
414 \def\RangeCatcodeInvalid#1#2{%
415   \count@=#1\relax
416   \loop
417     \catcode\count@=15 %
418     \ifnum\count@<#2\relax
419       \advance\count@ 1 %
420     \repeat
421 }
422 \expandafter\ifx\csname LoadCommand\endcsname\relax
423   \def\LoadCommand{\input ltxcmds.sty\relax}%
424 \fi
425 \def\Test{%
426   \RangeCatcodeInvalid{0}{47}%
427   \RangeCatcodeInvalid{58}{64}%
428   \RangeCatcodeInvalid{91}{96}%
429   \RangeCatcodeInvalid{123}{255}%
430   \catcode'\@=12 %
431   \catcode'\=0 %
432   \catcode'\{=1 %
433   \catcode'\}=2 %
434   \catcode'\#=6 %
435   \catcode'\[=12 %
436   \catcode'\]=12 %
437   \catcode'\%=14 %

```

```

438 \catcode'\ =10 %
439 \catcode13=5 %
440 \LoadCommand
441 \RestoreCatcodes
442 }
443 \Test
444 \csname @@end\endcsname
445 \end
446 </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
```

**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.

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



## 4.4 Refresh file name databases

If your  $\text{T}_{\text{E}}\text{X}$  distribution (te $\text{T}_{\text{E}}\text{X}$ , mik $\text{T}_{\text{E}}\text{X}$ , ...) relies on file name databases, you must refresh these. For example, te $\text{T}_{\text{E}}\text{X}$  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  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ .** The `.dtx` chooses its action depending on the format:

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

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

If you insist on using  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  for `docstrip` (really, `docstrip` does not need  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ ), 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 pdf $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ :

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

## 5 History

[2009/08/05 v1.0]

- First version.

[2009/12/12 v1.1]

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

[2010/01/28 v1.2]

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

## [2010/03/01 v1.3]

- `\ltx@newif` added.
- `\ltx@ifnextchar` added.
- Numbers `\ltx@zero`, `\ltx@one`, `\ltx@two`, `\ltx@cc1v` 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.

## 6 Index

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

Symbols		373, 408, 417, 430, 431, 432, 433, 434, 435, 436, 437, 438, 439
<code>\#</code> .....	372, 434	<code>\chardef</code> ..... 99, 100, 101, 102
<code>\%</code> .....	123, 437	<code>\count@</code> ..... 375, 404, 408, 410, 411, 415, 417, 418, 419
<code>\@</code> .....	373, 430	<code>\countdef</code> ..... 375
<code>\@firstofone</code> .....	381, 384	<code>\csname</code> ..... 10, 18, 44, 60, 67, 142, 144, 147, 149, 152, 161, 163, 171, 184, 216, 344, 374, 377, 380, 383, 422, 444
<code>\@gobble</code> .....	378, 386	
<code>\@nil</code> .....	110, 111, 112, 113, 114, 115, 116, 117, 136, 141, 250, 252	
<code>\@undefined</code> .....	52	
<code>\[</code> .....	435	
<code>\]</code> .....	128, 431	
<code>\{</code> .....	370, 432	
<code>\}</code> .....	371, 433	
<code>\]</code> .....	436	
<code>\_</code> .....	438	
A		<b>E</b>
<code>\advance</code> .....	411, 419	<code>\empty</code> ..... 13, 14
<code>\aftergroup</code> .....	26	<code>\end</code> ..... 445
B		<code>\endcsname</code> ..... 10, 18, 44, 60, 67, 142, 144, 147, 149, 152, 161, 163, 171, 183, 184, 216, 344, 374, 377, 380, 383, 422, 444
<code>\body</code> .....	390, 394	<code>\endinput</code> ..... 26
<code>\box</code> .....	338, 349	<code>\escapechar</code> ..... 134, 139
C		<b>F</b>
<code>\catcode</code> 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, 370, 371, 372,		<code>\futurelet</code> ..... 304, 321
I		<b>H</b>
<code>\if</code> .....	141	<code>\hbox</code> ..... 336, 338, 347, 349, 365
<code>\ifcase</code> .....	225	
<code>\ifcsname</code> .....	183	
<code>\iffalse</code> .....	150	

\ifnum . . . . .	226, 228, 230, 257, 410, 418	\ltx@StripPrefix . .	196, 263, 264, 265
\iftrue . . . . .	145	\ltx@two . . . . .	101, 303, 314
\ifvoid . . . . .	345	\ltx@undefined . . . . .	272, 286
\ifx . 11, 14, 18, 44, 52, 55, 161, 163,		\ltx@zero 2, 99, 302, 308, 336, 338, 340	
171, 184, 272, 275, 286, 289,		\LTXcmds@ifnextchar . . . . .	311, 318
307, 310, 374, 377, 380, 383, 422		\LTXcmds@ParseVersion . . . . .	250, 252
\immediate . . . . .	20, 46	\LTXcmds@AtEnd . . . . .	80, 81, 367
\input . . . . .	423	\LTXcmds@CharToken . . . . .	301, 307
\iterate . . . . .	391, 393, 395	\LTXcmds@gtmp . . . . .	
		. 285, 286, 287, 289, 294, 295, 297	
<b>L</b>		\LTXcmds@iflater . . . . .	212, 224
\lccode . . . . .	123, 128	\LTXcmds@ifnextchar . . .	304, 306, 321
\leavevmode . . . . .	358	\LTXcmds@LetToken . 304, 307, 310, 321	
\letLTXcmds@gtmp . . . . .	290	\LTXcmds@newif . . . . .	136, 138
\LoadCommand . . . . .	423, 440	\LTXcmds@ParseVersion . . . . .	
\loop . . . . .	389, 405, 416	. . . . .	214, 220, 249, 256
\lowercase . . . . .	124, 129	\LTXcmds@SpaceToken . . . . .	310, 325
\ltx@backslashchar . . . . .	127	\LTXcmds@VoidBox 344, 345, 347, 349, 352	
\ltx@car . . . . .	3, 110	<b>N</b>	
\ltx@carfour . . . . .	3, 116	\next . . . . .	395, 397, 399
\ltx@carthree . . . . .	3, 114	\number . . . . .	213, 219
\ltx@cartwo . . . . .	3, 112	<b>P</b>	
\ltx@cclv . . . . .	102	\PackageInfo . . . . .	23
\ltx@cdr . . . . .	111	\pdflastmatch . . . . .	263, 264, 265
\ltx@cdrfour . . . . .	117	\pdfmatch . . . . .	257
\ltx@cdrthree . . . . .	115	\ProvidesPackage . . . . .	15, 61
\ltx@cdrtwo . . . . .	113	<b>Q</b>	
\ltx@clsextension . . . 4, 199, 205, 243		\quitvmode . . . . .	361
\ltx@empty . . . 3, 120, 273, 276, 287, 290		<b>R</b>	
\ltx@firstofone . . . . .	3, 107, 179	\RangeCatcodeInvalid . . . . .	
\ltx@firstoftwo . . . . .		. . . . .	414, 426, 427, 428, 429
. 108, 164, 172, 185, 190, 202, 239		\repeat . . . . .	389, 401, 412, 420
\ltx@GlobalAppendToMacro . . . . .	5, 271	\RestoreCatcodes . . 403, 406, 407, 441	
\ltx@gobble . . . . .	2, 103, 177	<b>S</b>	
\ltx@gobblefour . . . . .	106	\setbox . . . . .	336, 338, 347, 349
\ltx@gobblethree . . . . .	105	<b>T</b>	
\ltx@gobbletwo . . . . .	104	\Test . . . . .	425, 443
\ltx@ifclasslater . . . . .	5, 242	\the . . . . .	68, 69,
\ltx@ifclassloaded . . . . .	5, 204	70, 71, 82, 281, 295, 308, 313, 408	
\ltx@iffilelater . . . . .	210, 246	\TMP@EnsureCode . . . 79, 86, 87, 88,	
\ltx@iffileloaded 5, 201, 205, 208, 211		89, 90, 91, 92, 93, 94, 95, 96, 97, 98	
\ltx@ifnextchar . . . . .	5, 299	\toks . . . . .	280,
\ltx@ifpackagelater . . . . .	245	281, 294, 295, 302, 303, 308, 313	
\ltx@ifpackageloaded . . . . .	207	<b>U</b>	
\ltx@ifUndefined . . . . .	4,	\unhbox . . . . .	340, 352, 355
169, 193, 248, 330, 331, 332, 333		<b>V</b>	
\ltx@ifundefined 4, 162, 182, 193, 202		\voidb@x . . . . .	355
\ltx@leavevmode . . . . .	6, 330, 364	<b>W</b>	
\ltx@LocalAppendToMacro . . . . .	284	\write . . . . .	20, 46
\ltx@LocalExpandAfter . . . 4, 154, 160		<b>X</b>	
\ltx@mbox . . . . .	6, 363	\x . . . . .	10, 11, 14, 19, 23, 25,
\ltx@newif . . . . .	4, 132	45, 50, 60, 66, 74, 319, 324, 326, 329	
\ltx@one . . . . .	100		
\ltx@percentchar . . . . .	122		
\ltx@pkgextension . . . . 200, 208, 246			
\ltx@RemovePrefix . . . . .	4, 195, 197		
\ltx@ReturnAfterElseFi . . . . .	119		
\ltx@ReturnAfterFi . . . . .	3, 118		
\ltx@secondoftwo . . . . .			
. 109, 166, 174, 187, 202, 222, 237			
\ltx@space . . . . .	3, 121, 236		