

The ltxcmds package

Heiko Oberdiek
<heiko.oberdiek at gmail.com>

2010/04/26 v1.7

Abstract

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

Contents

1	Documentation	2
1.1	Introduction	2
1.2	Numbers	2
1.3	Argument killers	3
1.4	Argument grabbers	3
1.5	List helpers	3
1.6	Tail recursion	4
1.7	Empty macro	4
1.8	Characters	4
1.9	Boolean switch	4
1.10	Command definitions	4
1.11	Stripping	5
1.12	File management	5
	1.12.1 File extensions	5
	1.12.2 Load check	5
	1.12.3 Version date check	5
1.13	Macro additions	6
1.14	Macro \ltx@ifnextchar	6
1.15	\ltx@leavevmode, \ltx@mbbox	6
2	Implementation	6
2.1	Identification	6
2.2	Numbers	8
2.3	Argument killers	8
2.4	Argument grabbers	9
2.5	List helpers	9
2.6	Tail recursion	10
2.7	Empty macro	10
2.8	Characters	11
2.9	Boolean switch	11
2.10	Command definitions	11
2.11	Stripping	12
2.12	File management	13
	2.12.1 File extensions	13
	2.12.2 Load check	13
	2.12.3 Version date check	13
2.13	Macro additions	14

2.14 Macro <code>\ltx@ifnextchar</code>	15
2.15 <code>\ltx@leavevmode</code> , <code>\ltx@mbox</code>	16
2.16 Help macros	16
3 Test	16
3.1 Catcode checks for loading	16
4 Test <code>\ltx@GobbleNum</code>	18
5 Installation	21
5.1 Download	21
5.2 Bundle installation	21
5.3 Package installation	21
5.4 Refresh file name databases	21
5.5 Some details for the interested	22
6 History	22
[2009/08/05 v1.0]	22
[2009/12/12 v1.1]	22
[2010/01/28 v1.2]	22
[2010/03/01 v1.3]	22
[2010/03/09 v1.4]	23
[2010/04/08 v1.5]	23
[2010/04/16 v1.6]	23
[2010/04/26 v1.7]	23
7 Index	23

1 Documentation

1.1 Introduction

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

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

1.2 Numbers

<code>\ltx@zero</code>	→ 0
<code>\ltx@one</code>	→ 1
<code>\ltx@two</code>	→ 2
<code>\ltx@ccclv</code>	→ 255

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

1.3 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>	→

<code>\ltx@GobbleNum {⟨num⟩} {⟨1⟩} {⟨2⟩} ... {⟨⟨num⟩⟩}</code>	→
---	---

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

1.4 Argument grabbers

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

1.5 List helpers

<code>\ltx@carzero ... \@nil</code>	→
<code>\ltx@cdrzero ... \@nil</code>	→ ...

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

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

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

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

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

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

1.6 Tail recursion

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

1.7 Empty macro

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

1.8 Characters

<code>\ltx@space</code>	→	<code>␣</code>
<code>\ltx@percentchar</code>	→	<code>%</code>
<code>\ltx@backslashchar</code>	→	<code>\</code>
<code>\ltx@hashchar</code>	→	<code>#</code>

1.9 Boolean switch

<code>\ltx@newif {⟨cmd⟩}</code>

`\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

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

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”.

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

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

<code>\ltx@LocalExpandAfter</code>

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 L^AT_EX’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 L^AT_EX kernel. Also they can be used after the preamble.

1.12.1 File extensions

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

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

1.12.2 Load check

```
\ltx@ifclassloaded {<class>} {<yes>} {<no>}  
\ltx@ifpackageloaded {<package>} {<yes>} {<no>}
```

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

```
\ltx@iffileloaded {<file>} {<yes>} {<no>}
```

If L^AT_EX’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^AT_EX 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_EX’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

```
\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 L^AT_EX 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 L^AT_EX 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_EX'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^AT_EX 2_ε in 1994.

1.13 Macro additions

```
\ltx@GlobalAppendToMacro {<cmd>} {<addition>}
\ltx@LocalAppendToMacro {<cmd>} {<addition>}
```

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

1.14 Macro `\ltx@ifnextchar`

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

If next character is `<char>` then `<yes>` is called, otherwise `<no>`. The character is not removed.

1.15 `\ltx@leavevmode`, `\ltx@mbox`

```
\ltx@leavevmode
```

Macro `\ltx@leavevmode` calls pdfT_EX'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^AT_EX.

```
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{ltxcms}{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@ltxcms.sty\endcsname
61 \ProvidesPackage{ltxcms}%
62 [2010/04/26 v1.7 LaTeX kernel commands for general use (HO)]

63 \begingroup
64 \catcode123 1 % {
65 \catcode125 2 % }
66 \def\x{\endgroup
67 \expandafter\edef\csname LTXcms@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@active
102 \chardef\ltx@active=13 %

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

```

2.3 Argument killers

```

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

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

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

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

\ltx@GobbleNum
108 \def\ltx@GobbleNum#1{%
109   \romannumeral
110   \csname ltx@zero%
111   \expandafter\LTXcmds@GobbleNum
112   \romannumeral\expandafter\ltx@firstofone
113   \expandafter{\LTXcmds@num#1}000{m\endcsname}%
114 }

```



```

\LTxcmds@GobbleNum
115 \def\LTxcmds@GobbleNum#1{%
116 \csname LTxcmds@G#1\LTxcmds@GobbleNum
117 }

```

```

\LTxcmds@Gm
118 \long\def\LTxcmds@Gm#1{%
119 \endcsname
120 }

```

2.4 Argument grabbers

```

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

```

```

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

```

```

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

```

2.5 List helpers

```

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

```

```

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

```

```

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

```

```

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

```

```

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

```

```

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

```

```

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

```

```

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

```

```

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

```

```

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

```

`\ltx@CarNum`

```
134 \def\ltx@CarNum#1{%
135   \romannumeral
136   \csname LTXcmds@CarNumFinish%
137   \expandafter\LTXcmds@CarNum
138   \romannumeral\expandafter\ltx@firstofone
139   \expandafter{\LTXcmds@num#1}000{x\endcsname}%
140 }
```

`\LTXcmds@CarNum`

```
141 \def\LTXcmds@CarNum#1{%
142   \csname LTXcmds@C#1\LTXcmds@CarNum
143 }
```

`\LTXcmds@Cm`

```
144 \long\def\LTXcmds@Cm#1#2{%
145   \endcsname{#1#2}%
146 }
```

`\LTXcmds@Cx`

```
147 \def\LTXcmds@Cx#1{%
148   \endcsname{}%
149 }
```

`\LTXcmds@CarNumFinish`

```
150 \long\def\LTXcmds@CarNumFinish#1#2\@nil{%
151   \ltx@zero
152   #1%
153 }
```

`\ltx@CdrNum`

```
154 \def\ltx@CdrNum#1{%
155   \romannumeral0%
156   \expandafter\expandafter\expandafter\LTXcmds@CdrNum
157   \ltx@GobbleNum{#1}%
158 }
```

`\LTXcmds@CdrNum`

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

2.6 Tail recursion

`\ltx@ReturnAfterFi`

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

`\ltx@ReturnAfterElseFi`

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

2.7 Empty macro

`\ltx@empty`

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

2.8 Characters

`\ltx@space`

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

`\ltx@percentchar`

```
164 \begingroup
165 \lccode'0='\%\relax
166 \lowercase{\endgroup
167 \def\ltx@percentchar{0}%
168 }
```

`\ltx@backslashchar`

```
169 \begingroup
170 \lccode'0='\%\relax
171 \lowercase{\endgroup
172 \def\ltx@backslashchar{0}%
173 }
```

`\ltx@hashchar`

```
174 \begingroup
175 \lccode'0='\#\relax
176 \lowercase{\endgroup
177 \def\ltx@hashchar{0}%
178 }
```

2.9 Boolean switch

`\ltx@newif`

```
179 \def\ltx@newif#1{%
180 \begingroup
181 \escapechar=-1 %
182 \expandafter\endgroup
183 \expandafter\LTXcmds@newif\string#1\@nil
184 }
```

`\LTXcmds@newif`

```
185 \begingroup
186 \escapechar=-1 %
187 \expandafter\endgroup
188 \expandafter\def\expandafter\LTXcmds@newif\string\if#1\@nil{%
189 \expandafter\edef\csname#1true\endcsname{%
190 \let
191 \expandafter\noexpand\csname if#1\endcsname
192 \noexpand\iftrue
193 }%
194 \expandafter\edef\csname#1false\endcsname{%
195 \let
196 \expandafter\noexpand\csname if#1\endcsname
197 \noexpand\iffalse
198 }%
199 \csname#1false\endcsname
200 }
```

2.10 Command definitions

`\ltx@LocalExpandAfter`

```
201 \def\ltx@LocalExpandAfter{%
202 \begingroup
203 \expandafter\expandafter\expandafter
```

```

204 \endgroup
205 \expandafter
206 }

207 \ltx@LocalExpandAfter
208 \ifx\csname ifcsname\endcsname\relax

```

\ltx@ifundefined

```

209 \def\ltx@ifundefined#1{%
210   \expandafter\ifx\csname #1\endcsname\relax
211   \expandafter\ltx@firstoftwo
212   \else
213   \expandafter\ltx@secondoftwo
214   \fi
215 }%

```

\ltx@ifUndefined

```

216 \def\ltx@ifUndefined#1{%
217   \begingroup\expandafter\expandafter\expandafter\endgroup
218   \expandafter\ifx\csname #1\endcsname\relax
219   \expandafter\ltx@firstoftwo
220   \else
221   \expandafter\ltx@secondoftwo
222   \fi
223 }%

224 \expandafter\ltx@gobble
225 \else
226 \expandafter\ltx@firstofone
227 \fi
228 {%

```

\ltx@ifundefined

```

229 \def\ltx@ifundefined#1{%
230   \ifcsname #1\endcsname
231   \expandafter\ifx\csname #1\endcsname\relax
232   \expandafter\expandafter\expandafter\ltx@firstoftwo
233   \else
234   \expandafter\expandafter\expandafter\ltx@secondoftwo
235   \fi
236   \else
237   \expandafter\ltx@firstoftwo
238   \fi
239 }%

```

\ltx@ifUndefined

```

240 \let\ltx@ifUndefined\ltx@ifundefined
241 }

```

2.11 Stripping

\ltx@RemovePrefix

```

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

```

\ltx@StripPrefix

```

243 \def\ltx@StripPrefix{%
244   \expandafter\ltx@RemovePrefix
245 }

```

2.12 File management

2.12.1 File extensions

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

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

2.12.2 Load check

```
\ltx@iffileloaded
248 \def\ltx@iffileloaded#1{%
249 \ltx@ifundefined{ver#1}\ltx@secondoftwo\ltx@firstoftwo
250 }
```

```
\ltx@ifclassloaded
251 \def\ltx@ifclassloaded#1{%
252 \ltx@iffileloaded{#1.\ltx@clsextension}%
253 }
```

```
\ltx@ifpackageloaded
254 \def\ltx@ifpackageloaded#1{%
255 \ltx@iffileloaded{#1.\ltx@pkgextension}%
256 }
```

2.12.3 Version date check

```
\ltx@iffilelater
257 \def\ltx@iffilelater#1#2{%
258 \ltx@iffileloaded{#1}{%
259 \expandafter\LTXcmds@IfLater\expandafter{%
260 \number
261 \expandafter\expandafter\expandafter\LTXcmds@ParseVersion
262 \expandafter\expandafter\expandafter{%
263 \csname ver@#1\endcsname
264 }%
265 \expandafter}\expandafter{%
266 \number
267 \expandafter\LTXcmds@ParseVersion\expandafter{#2}%
268 }%
269 }\ltx@secondoftwo
270 }
```

```
\LTXcmds@IfLater
271 \def\LTXcmds@IfLater#1#2{%
272 \ifcase 0%
273 \ifnum#1<19940101 %
274 \else
275 \ifnum#2<19940101 %
276 \else
277 \ifnum#2>#1 %
278 \else
279 1%
280 \fi
281 \fi
282 \fi
283 \ltx@space
284 \expandafter\ltx@secondoftwo
285 \else
```

```

286   \expandafter\ltx@firstoftwo
287   \fi
288 }

```

\ltx@ifclasslater

```

289 \def\ltx@ifclasslater#1{%
290   \ltx@ifclasslater{#1.\ltx@clsextension}%
291 }

```

\ltx@ifpackagelater

```

292 \def\ltx@ifpackagelater#1{%
293   \ltx@iffilelater{#1.\ltx@pkgextension}%
294 }

295 \ltx@ifUndefined{pdfmatch}{%

```

\LTXcmds@ParseVersion

```

296 \def\LTXcmds@ParseVersion#1{%
297   \LTXcmds@@ParseVersion#1000/00/00\@nil
298 }%

```

\LTXcmds@@ParseVersion

```

299 \def\LTXcmds@@ParseVersion#1#2#3#4/#5#6/#7#8#9\@nil{%
300   #1#2#3#4#5#6#7#8%
301 }%

302 }{%

```

\LTXcmds@ParseVersion

```

303 \def\LTXcmds@ParseVersion#1{%
304   \ifnum\pdfmatch{%
305     ~%
306     (199[4-9] | [2-9] [0-9] [0-9] [0-9])/%
307     (0[1-9] | 1[0-2])/%
308     (0[1-9] | [1-2] [0-9] | 3[0-1])%
309   }{#1}=1 %
310   \ltx@StripPrefix\pdfastmatch1 %
311   \ltx@StripPrefix\pdfastmatch2 %
312   \ltx@StripPrefix\pdfastmatch3 %
313   \else
314     0%
315   \fi
316 }%
317 }

```

2.13 Macro additions

\ltx@GlobalAppendToMacro

```

318 \def\ltx@GlobalAppendToMacro#1#2{%
319   \ifx\ltx@undefined#1%
320     \let#1\ltx@empty
321   \else
322     \ifx\relax#1%
323       \let#1\ltx@empty
324     \fi
325   \fi
326   \begingroup
327     \toks0\expandafter{#1#2}%
328     \xdef#1{\the\toks0}%
329   \endgroup
330 }

```

`\ltx@LocalAppendToMacro`

```
331 \def\ltx@LocalAppendToMacro#1#2{%
332   \global\let\LTXcmds@gtemp#1%
333   \ifx\ltx@undefined\LTXcmds@gtemp
334     \global\let\LTXcmds@gtemp\ltx@empty
335   \else
336     \ifx\relax\LTXcmds@gtemp
337       \global\let\LTXcmds@gtemp\ltx@empty
338     \fi
339   \fi
340   \begingroup
341     \toks0\expandafter{\LTXcmds@gtemp#2}%
342     \xdef\LTXcmds@gtemp{\the\toks0}%
343   \endgroup
344   \let#1\LTXcmds@gtemp
345 }
```

2.14 Macro `\ltx@ifnextchar`

`\ltx@ifnextchar`

```
346 \long\def\ltx@ifnextchar#1#2#3{%
347   \begingroup
348   \let\LTXcmds@CharToken= #1\relax
349   \toks\ltx@zero{#2}%
350   \toks\ltx@two{#3}%
351   \futurelet\LTXcmds@LetToken\LTXcmds@ifnextchar
352 }
```

`\LTXcmds@ifnextchar`

```
353 \def\LTXcmds@ifnextchar{%
354   \ifx\LTXcmds@LetToken\LTXcmds@CharToken
355     \expandafter\endgroup\the\toks\expandafter\ltx@zero
356   \else
357     \ifx\LTXcmds@LetToken\LTXcmds@SpaceToken
358       \expandafter\expandafter\expandafter\LTXcmds@@ifnextchar
359     \else
360       \expandafter\endgroup\the\toks
361       \expandafter\expandafter\expandafter\ltx@two
362     \fi
363   \fi
364 }
```

`\LTXcmds@@ifnextchar`

```
365 \begingroup
366   \def\x#1{\endgroup
367     \def\LTXcmds@@ifnextchar#1{%
368       \futurelet\LTXcmds@LetToken\LTXcmds@ifnextchar
369     }%
370   }%
371   \x{ }
```

`\LTXcmds@SpaceToken`

```
372 \begingroup
373   \def\x#1{\endgroup
374     \let\LTXcmds@SpaceToken= #1%
375   }%
376   \x{ }
```

2.15 `\ltx@leavevmode`, `\ltx@mbox`

`\ltx@leavevmode`

```
377 \ltx@ifundefined{quitvmode}{%
378   \ltx@ifundefined{leavevmode}{%
379     \ltx@ifundefined{voidb@x}{%
380       \ltx@ifundefined{newbox}{%
381         \def\ltx@leavevmode{%
382           \begingroup
383             \setbox\ltx@zero=\hbox{}}%
384           \begingroup
385             \setbox\ltx@zero=\hbox{\box\ltx@zero}%
386           \endgroup
387           \unhbox\ltx@zero
388         \endgroup
389       }%
390     }{%
391       \csname newbox\endcsname\LTXcmds@VoidBox
392       \ifvoid\LTXcmds@VoidBox
393       \else
394         \setbox\LTXcmds@VoidBox=\hbox{}}%
395       \begingroup
396         \setbox\LTXcmds@VoidBox=\hbox{\box\LTXcmds@VoidBox}%
397       \endgroup
398     \fi
399     \def\ltx@leavevmode{\unhbox\LTXcmds@VoidBox}%
400   }%
401 }{%
402   \def\ltx@leavevmode{\unhbox\voidb@x}%
403 }%
404 }{%
405   \let\ltx@leavevmode\leavevmode
406 }%
407 }{%
408   \let\ltx@leavevmode\quitvmode
409 }
```

`\ltx@mbox`

```
410 \def\ltx@mbox{%
411   \ltx@leavevmode
412   \hbox
413 }
```

2.16 Help macros

`\LTXcmds@num`

```
414 \ltx@ifundefined{numexpr}{%
415   \let\LTXcmds@num\number
416 }{%
417   \let\LTXcmds@num\numexpr
418 }
```

```
419 \LTXcmds@AtEnd
420 </package>
```

3 Test

3.1 Catcode checks for loading

```
421 <*test1>
422 \catcode'\{=1 %
```



```

423 \catcode'\}=2 %
424 \catcode'\#=6 %
425 \catcode'\@=11 %
426 \expandafter\ifx\csname count@\endcsname\relax
427 \countdef\count@=255 %
428 \fi
429 \expandafter\ifx\csname @gobble\endcsname\relax
430 \long\def@gobble#1{}%
431 \fi
432 \expandafter\ifx\csname @firstofone\endcsname\relax
433 \long\def@firstofone#1{#1}%
434 \fi
435 \expandafter\ifx\csname loop\endcsname\relax
436 \expandafter@firstofone
437 \else
438 \expandafter@gobble
439 \fi
440 {%
441 \def\loop#1\repeat{%
442 \def\body{#1}%
443 \iterate
444 }%
445 \def\iterate{%
446 \body
447 \let\next\iterate
448 \else
449 \let\next\relax
450 \fi
451 \next
452 }%
453 \let\repeat=\fi
454 }%
455 \def\RestoreCatcodes{}
456 \count@=0 %
457 \loop
458 \edef\RestoreCatcodes{%
459 \RestoreCatcodes
460 \catcode\the\count@=\the\catcode\count@\relax
461 }%
462 \ifnum\count@<255 %
463 \advance\count@ 1 %
464 \repeat
465
466 \def\RangeCatcodeInvalid#1#2{%
467 \count@=#1\relax
468 \loop
469 \catcode\count@=15 %
470 \ifnum\count@<#2\relax
471 \advance\count@ 1 %
472 \repeat
473 }
474 \expandafter\ifx\csname LoadCommand\endcsname\relax
475 \def\LoadCommand{\input ltxcmds.sty\relax}%
476 \fi
477 \def\Test{%
478 \RangeCatcodeInvalid{0}{47}%
479 \RangeCatcodeInvalid{58}{64}%
480 \RangeCatcodeInvalid{91}{96}%
481 \RangeCatcodeInvalid{123}{255}%
482 \catcode'\@=12 %
483 \catcode'\}=0 %
484 \catcode'\{=1 %

```

```

485 \catcode'\}=2 %
486 \catcode'\#=6 %
487 \catcode'\ [=12 %
488 \catcode'\]=12 %
489 \catcode'\%=14 %
490 \catcode'\ =10 %
491 \catcode13=5 %
492 \LoadCommand
493 \RestoreCatcodes
494 }
495 \Test
496 \csname @@end\endcsname
497 \end
498 </test1>

```

4 Test \ltx@GobbleNum

```

499 <*test-gobble>
500 \catcode'\{=1 %
501 \catcode'\}=2 %
502 \catcode'\#=6 %
503 \expandafter\ifx\csname RequirePackage\endcsname\relax
504 \input ltxcmds.sty\relax
505 \else
506 \RequirePackage{ltxcmds}[2010/04/26]%
507 \fi
508 \catcode'\@=11 %
509 \def\msg#\{ \immediate\write16}%
510 \msg{[Test \string\ltx@GobbleNum]}%
511 \long\def\Test#1=#2\{\%
512 \edef\StrA{\ltx@GobbleNum#1}%
513 \expandafter\expandafter\expandafter\def
514 \expandafter\expandafter\expandafter\StrAA
515 \expandafter\expandafter\expandafter{\ltx@GobbleNum#1}%
516 \edef\StrB{#2}%
517 \ifx\StrA\StrB
518 \ifx\StrAA\StrB
519 \msg{* ok.}%
520 \else
521 \msg{StrAA: \StrAA}%
522 \msg{StrB: \StrB}%
523 \errhelp{Test: #1=#2}%
524 \errmessage{Test (two expansions) failed}%
525 \fi
526 \else
527 \msg{StrA: \StrA}%
528 \msg{StrB: \StrB}%
529 \errhelp{Test: #1=#2}%
530 \errmessage{Test (edef) failed!}%
531 \fi
532 }
533 \Test0abc=abc\\
534 \Test1abc=bc\\
535 \Test2abc=c\\
536 \Test3abcd=d\\
537 \Test4abcde=e\\
538 \Test5abcdef=f\\
539 \Test6abcdefg=g\\
540 \Test7abcdefgh=h\\
541 \Test8abcdefghi=i\\
542 \Test9abcdefghij=j\\
543 \Test{10}0123456789X=X\\

```

```

544 \Test{12}abcdefghijklm=m\\
545 \Test{700}%
546 0123456789012345678901234567890123456789012345678901234567890123456789%
547 0123456789012345678901234567890123456789012345678901234567890123456789%
548 0123456789012345678901234567890123456789012345678901234567890123456789%
549 0123456789012345678901234567890123456789012345678901234567890123456789%
550 0123456789012345678901234567890123456789012345678901234567890123456789%
551 0123456789012345678901234567890123456789012345678901234567890123456789%
552 0123456789012345678901234567890123456789012345678901234567890123456789%
553 0123456789012345678901234567890123456789012345678901234567890123456789%
554 0123456789012345678901234567890123456789012345678901234567890123456789%
555 0123456789012345678901234567890123456789012345678901234567890123456789%
556 X=X\\
557 \Test{-1}abc=abc\\
558 \Test2\par\par\relax=\relax\\
559
560 \msg{[Test \string\ltx@CdrNum]}%
561 \long\def\Test#1=#2\\{%
562   \edef\StrA{\ltx@CdrNum#1\@nil}%
563   \expandafter\expandafter\expandafter\def
564   \expandafter\expandafter\expandafter\StrAA
565   \expandafter\expandafter\expandafter{\ltx@CdrNum#1\@nil}%
566   \edef\StrB{#2}%
567   \ifx\StrA\StrB
568     \ifx\StrAA\StrB
569       \msg{* ok.}%
570     \else
571       \msg{StrAA: \meaning\StrAA}%
572       \msg{StrB: \meaning\StrB}%
573       \errhelp{Test: #1=#2}%
574       \errmessage{Test (two expansions) failed}%
575     \fi
576   \else
577     \msg{StrA: \StrA}%
578     \msg{StrB: \StrB}%
579     \errhelp{Test: #1=#2}%
580     \errmessage{Test (edef) failed!}%
581   \fi
582 }
583 \Test0abc=abc\\
584 \Test1abc=bc\\
585 \Test2abc=c\\
586 \Test3abcd=d\\
587 \Test4abcde=e\\
588 \Test5abcdef=f\\
589 \Test6abcdefg=g\\
590 \Test7abcdefgh=h\\
591 \Test8abcdefghi=i\\
592 \Test9abcdefghij=j\\
593 \Test{10}0123456789X=X\\
594 \Test{12}abcdefghijklm=m\\
595 \Test{700}%
596 0123456789012345678901234567890123456789012345678901234567890123456789%
597 0123456789012345678901234567890123456789012345678901234567890123456789%
598 0123456789012345678901234567890123456789012345678901234567890123456789%
599 0123456789012345678901234567890123456789012345678901234567890123456789%
600 0123456789012345678901234567890123456789012345678901234567890123456789%
601 0123456789012345678901234567890123456789012345678901234567890123456789%
602 0123456789012345678901234567890123456789012345678901234567890123456789%
603 0123456789012345678901234567890123456789012345678901234567890123456789%
604 0123456789012345678901234567890123456789012345678901234567890123456789%
605 0123456789012345678901234567890123456789012345678901234567890123456789%

```

```

606 X=X\\
607 \Test{-1}abc=abc\\
608 \Test2\par\par\relax=\relax\\
609
610 \msg{[Test \string\ltx@CarNum]}%
611 \long\def\Test#1=#2\\{%
612   \edef\StrA{\ltx@CarNum#1\@nil}%
613   \expandafter\expandafter\expandafter\def
614   \expandafter\expandafter\expandafter\StrAA
615   \expandafter\expandafter\expandafter{\ltx@CarNum#1\@nil}%
616   \edef\StrB{#2}%
617   \ifx\StrA\StrB
618     \ifx\StrAA\StrB
619       \msg{* ok.}%
620     \else
621       \msg{StrAA: \meaning\StrAA}%
622       \msg{StrB: \meaning\StrB}%
623       \errhelp{Test: #1=#2}%
624       \errmessage{Test (two expansions) failed}%
625     \fi
626   \else
627     \msg{StrA: \StrA}%
628     \msg{StrB: \StrB}%
629     \errhelp{Test: #1=#2}%
630     \errmessage{Test (edef) failed!}%
631   \fi
632 }
633 \Test0abc=\\
634 \Test1abc=a\\
635 \Test2abc=ab\\
636 \Test3abc=abc\\
637 \Test3abcd=abc\\
638 \Test4abcde=abcd\\
639 \Test{10}0123456789X=0123456789\\
640 \Test{12}abcdefghijklm=abcdefghijkl\\
641 \Test{700}%
642 0123456789012345678901234567890123456789012345678901234567890123456789%
643 0123456789012345678901234567890123456789012345678901234567890123456789%
644 0123456789012345678901234567890123456789012345678901234567890123456789%
645 0123456789012345678901234567890123456789012345678901234567890123456789%
646 0123456789012345678901234567890123456789012345678901234567890123456789%
647 0123456789012345678901234567890123456789012345678901234567890123456789%
648 0123456789012345678901234567890123456789012345678901234567890123456789%
649 0123456789012345678901234567890123456789012345678901234567890123456789%
650 0123456789012345678901234567890123456789012345678901234567890123456789%
651 0123456789012345678901234567890123456789012345678901234567890123456789%
652 X=%
653 0123456789012345678901234567890123456789012345678901234567890123456789%
654 0123456789012345678901234567890123456789012345678901234567890123456789%
655 0123456789012345678901234567890123456789012345678901234567890123456789%
656 0123456789012345678901234567890123456789012345678901234567890123456789%
657 0123456789012345678901234567890123456789012345678901234567890123456789%
658 0123456789012345678901234567890123456789012345678901234567890123456789%
659 0123456789012345678901234567890123456789012345678901234567890123456789%
660 0123456789012345678901234567890123456789012345678901234567890123456789%
661 0123456789012345678901234567890123456789012345678901234567890123456789%
662 0123456789012345678901234567890123456789012345678901234567890123456789%
663 \\
664 \Test{-1}abc=\\
665 \Test2\par\par\relax=\par\par\\
666 \cscname @@end\endcsname\end
667 /</test-gobble)

```

5 Installation

5.1 Download

Package. This package is available on CTAN¹:

[CTAN:macros/latex/contrib/oberdiek/ltxcmds.dtx](#) The source file.

[CTAN:macros/latex/contrib/oberdiek/ltxcmds.pdf](#) Documentation.

Bundle. All the packages of the bundle ‘oberdiek’ are also available in a TDS compliant ZIP archive. There the packages are already unpacked and the documentation files are generated. The files and directories obey the TDS standard.

[CTAN:install/macros/latex/contrib/oberdiek.tds.zip](#)

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

5.2 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/
```

5.3 Package installation

Unpacking. The `.dtx` file is a self-extracting `docstrip` archive. The files are extracted by running the `.dtx` through plain \TeX :

```
tex ltxcmds.dtx
```

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

<code>ltxcmds.sty</code>	→ <code>tex/generic/oberdiek/ltxcmds.sty</code>
<code>ltxcmds.pdf</code>	→ <code>doc/latex/oberdiek/ltxcmds.pdf</code>
<code>test/ltxcmds-test1.tex</code>	→ <code>doc/latex/oberdiek/test/ltxcmds-test1.tex</code>
<code>test/ltxcmds-test-gobble.tex</code>	→ <code>doc/latex/oberdiek/test/ltxcmds-test-gobble.tex</code>
<code>ltxcmds.dtx</code>	→ <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`.

5.4 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`.

¹<ftp://ftp.ctan.org/tex-archive/>

5.5 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
```

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@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.

[2010/04/26 v1.7]

- `\ltx@GobbleNum`, `\ltx@CdrNum`, `\ltx@CarNum` added.
- `\ltx@carzero`, `\ltx@cdrzero` added.
- `\ltx@hashchar` 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>\box</code> 385, 396
<code>\#</code>	175, 424, 486, 502	
<code>\%</code>	165, 489	
<code>\@</code>	425, 482, 508	
<code>\@firstofone</code>	433, 436	
<code>\@gobble</code>	430, 438	
<code>\@nil</code> <i>124, 125, 126, 127, 128, 129, 130,</i> <i>131, 132, 133, 150, 159, 183,</i> <i>188, 297, 299, 562, 565, 612, 615</i>		
<code>\@undefined</code>	52	
<code>\[</code>	487	
<code>\]</code>	170, 483, 511, 533, 534, 535, 536, 537, 538, 539, 540, 541, 542, 543, 544, 556, 557, 558, 561, 583, 584, 585, 586, 587, 588, 589, 590, 591, 592, 593, 594, 606, 607, 608, 611, 633, 634, 635, 636, 637, 638, 639, 640, 663, 664, 665	
<code>\{</code>	422, 484, 500	
<code>\}</code>	423, 485, 501	
<code>\]</code>	488	
<code>_</code>	490	
A		
<code>\advance</code>	463, 471	
<code>\aftergroup</code>	26	
B		
<code>\body</code>	442, 446	
		<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, 422, 423, 424, 425, 460, 469, 482, 483, 484, 485, 486, 487, 488, 489, 490, 491, 500, 501, 502, 508
		<code>\chardef</code> 99, 100, 101, 102, 103
		<code>\count@</code> 427, 456, 460, 462, 463, 467, 469, 470, 471
		<code>\countdef</code> 427
		<code>\csname</code> 10, 18, 44, 60, 67, 110, 116, 136, 142, 189, 191, 194, 196, 199, 208, 210, 218, 231, 263, 391, 426, 429, 432, 435, 474, 496, 503, 666
		E
		<code>\empty</code> 13, 14
		<code>\end</code> 497, 666
		<code>\endcsname</code> 10, 18, 44, 60, 67, 113, 119, 139, 145, 148, 189, 191, 194, 196, 199, 208, 210, 218, 230, 231, 263, 391, 426, 429, 432, 435, 474, 496, 503, 666
		<code>\endinput</code> 26
		<code>\errhelp</code> 523, 529, 573, 579, 623, 629
		<code>\errmessage</code> 524, 530, 574, 580, 624, 630
		<code>\escapechar</code> 181, 186

F	
<code>\futurelet</code>	351, 368
H	
<code>\hbox</code>	383, 385, 394, 396, 412
I	
<code>\if</code>	188
<code>\ifcase</code>	272
<code>\ifcsname</code>	230
<code>\iffalse</code>	197
<code>\ifnum</code>	273, 275, 277, 304, 462, 470
<code>\iftrue</code>	192
<code>\ifvoid</code>	392
<code>\ifx</code>	11, 14, 18, 44, 52, 55, 208, 210, 218, 231, 319, 322, 333, 336, 354, 357, 426, 429, 432, 435, 474, 503, 517, 518, 567, 568, 617, 618
<code>\immediate</code>	20, 46, 509
<code>\input</code>	475, 504
<code>\iterate</code>	443, 445, 447
L	
<code>\lccode</code>	165, 170, 175
<code>\leavevmode</code>	405
<code>\letLTXcmds@gttemp</code>	337
<code>\LoadCommand</code>	475, 492
<code>\loop</code>	441, 457, 468
<code>\lowercase</code>	166, 171, 176
<code>\ltx@active</code>	102
<code>\ltx@backslashchar</code>	169
<code>\ltx@car</code>	3, 124
<code>\ltx@carfour</code>	3, 132
<code>\ltx@CarNum</code>	3, 134, 610, 612, 615
<code>\ltx@carthree</code>	3, 130
<code>\ltx@cartwo</code>	3, 128
<code>\ltx@carzero</code>	3, 126
<code>\ltx@cclv</code>	103
<code>\ltx@cdr</code>	125
<code>\ltx@cdrfour</code>	133
<code>\ltx@CdrNum</code>	154, 560, 562, 565
<code>\ltx@cdrthree</code>	131
<code>\ltx@cdrtwo</code>	129
<code>\ltx@cdrzero</code>	127
<code>\ltx@clsextension</code> ...	5, 246, 252, 290
<code>\ltx@empty</code> ..	4, 162, 320, 323, 334, 337
<code>\ltx@firstofone</code> .	3, 112, 121, 138, 226
<code>\ltx@firstoftwo</code>	122, 211, 219, 232, 237, 249, 286
<code>\ltx@GlobalAppendToMacro</code>	6, 318
<code>\ltx@gobble</code>	3, 104, 224
<code>\ltx@gobblefour</code>	107
<code>\ltx@GobbleNum</code> 3,	108, 157, 510, 512, 515
<code>\ltx@gobblethree</code>	106
<code>\ltx@gobbletwo</code>	105
<code>\ltx@hashchar</code>	174
<code>\ltx@ifclasslater</code>	5, 289
<code>\ltx@ifclassloaded</code>	5, 251
<code>\ltx@iffilelater</code>	257, 293
<code>\ltx@iffileloaded</code> 5,	248, 252, 255, 258
<code>\ltx@ifnextchar</code>	6, 346
<code>\ltx@ifpackagelater</code>	292
<code>\ltx@ifpackageloaded</code>	254
<code>\ltx@ifUndefined</code>	4, 216, 240, 295, 377, 378, 379, 380, 414
<code>\ltx@ifundefined</code> 4,	209, 229, 240, 249
<code>\ltx@leavevmode</code>	6, 377, 411
<code>\ltx@LocalAppendToMacro</code>	331
<code>\ltx@LocalExpandAfter</code> ...	4, 201, 207
<code>\ltx@mbox</code>	6, 410
<code>\ltx@newif</code>	4, 179
<code>\ltx@one</code>	100
<code>\ltx@percentchar</code>	164
<code>\ltx@pkgextension</code>	247, 255, 293
<code>\ltx@RemovePrefix</code>	5, 242, 244
<code>\ltx@ReturnAfterElseFi</code>	161
<code>\ltx@ReturnAfterFi</code>	4, 160
<code>\ltx@secondoftwo</code>	123, 213, 221, 234, 249, 269, 284
<code>\ltx@space</code>	4, 163, 283
<code>\ltx@StripPrefix</code> ..	243, 310, 311, 312
<code>\ltx@two</code>	101, 350, 361
<code>\ltx@undefined</code>	319, 333
<code>\ltx@zero</code>	2, 99, 151, 349, 355, 383, 385, 387
<code>\LTXcmds@ifnextchar</code>	358, 365
<code>\LTXcmds@ParseVersion</code>	297, 299
<code>\LTXcmds@AtEnd</code>	80, 81, 419
<code>\LTXcmds@CarNum</code>	137, 141
<code>\LTXcmds@CarNumFinish</code>	150
<code>\LTXcmds@CdrNum</code>	156, 159
<code>\LTXcmds@CharToken</code>	348, 354
<code>\LTXcmds@Cm</code>	144
<code>\LTXcmds@Cx</code>	147
<code>\LTXcmds@Gm</code>	118
<code>\LTXcmds@GobbleNum</code>	111, 115
<code>\LTXcmds@gttemp</code>	332, 333, 334, 336, 341, 342, 344
<code>\LTXcmds@ifLater</code>	259, 271
<code>\LTXcmds@ifnextchar</code> ...	351, 353, 368
<code>\LTXcmds@LetToken</code> .	351, 354, 357, 368
<code>\LTXcmds@newif</code>	183, 185
<code>\LTXcmds@num</code>	113, 139, 414
<code>\LTXcmds@ParseVersion</code>	261, 267, 296, 303
<code>\LTXcmds@SpaceToken</code>	357, 372
<code>\LTXcmds@VoidBox</code> 391,	392, 394, 396, 399
M	
<code>\meaning</code>	571, 572, 621, 622
<code>\msg</code> ...	509, 510, 519, 521, 522, 527, 528, 560, 569, 571, 572, 577, 578, 610, 619, 621, 622, 627, 628
N	
<code>\next</code>	447, 449, 451
<code>\number</code>	260, 266, 415
<code>\numexpr</code>	417
P	
<code>\PackageInfo</code>	23
<code>\par</code>	558, 608, 665
<code>\pdflastmatch</code>	310, 311, 312
<code>\pdfmatch</code>	304
<code>\ProvidesPackage</code>	15, 61

	Q	
\quitvmode	408	545, 557, 558, 561, 583, 584, 585, 586, 587, 588, 589, 590, 591, 592, 593, 594, 595, 607, 608, 611, 633, 634, 635, 636, 637, 638, 639, 640, 641, 664, 665
	R	
\RangeCatcodeInvalid	466, 478, 479, 480, 481	\the 68, 69, 70, 71, 82, 328, 342, 355, 360, 460
\repeat	441, 453, 464, 472	\TMP@EnsureCode 79, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98
\RequirePackage	506	\toks 327, 328, 341, 342, 349, 350, 355, 360
\RestoreCatcodes	455, 458, 459, 493	
\romannumeral	109, 112, 135, 138, 155	
	S	
\setbox	383, 385, 394, 396	
\StrA	512, 517, 527, 562, 567, 577, 612, 617, 627	U
\StrAA	514, 518, 521, 564, 568, 571, 614, 618, 621	\unhbox 387, 399, 402
\StrB	516, 517, 518, 522, 528, 566, 567, 568, 572, 578, 616, 617, 618, 622, 628	V
		\voidb@x 402
	T	W
\Test	477, 495, 511, 533, 534, 535, 536, 537, 538, 539, 540, 541, 542, 543, 544,	\write 20, 46, 509
		X
		\x 10, 11, 14, 19, 23, 25, 45, 50, 60, 66, 74, 366, 371, 373, 376