

The ltxcmds package

Heiko Oberdiek
<heiko.oberdiek at gmail.com>

2010/01/28 v1.2

Abstract

The package `ltxcmds` exports some utility macros from the \LaTeX kernel into a separate namespace and also provides them for other formats such as `plain-TeX`.

Contents

1	Documentation	2
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	3
1.7	Characters	3
1.8	Command definitions	3
1.9	Stripping	3
1.10	File management	3
1.10.1	File extensions	4
1.10.2	Load check	4
1.10.3	Version date check	4
1.11	Macro additions	4
2	Implementation	5
2.1	Identification	5
2.2	Argument killers	6
2.3	Argument grabbers	6
2.4	List helpers	7
2.5	Tail recursion	7
2.6	Empty macro	7
2.7	Characters	7
2.8	Command definitions	7
2.9	Stripping	8
2.10	File management	8
2.10.1	File extensions	8
2.10.2	Load check	9
2.10.3	Version date check	9
2.11	Macro additions	10
3	Test	11
3.1	Catcode checks for loading	11

4 Installation	12
4.1 Download	12
4.2 Bundle installation	12
4.3 Package installation	13
4.4 Refresh file name databases	13
4.5 Some details for the interested	13
5 History	14
[2009/08/05 v1.0]	14
[2009/12/12 v1.1]	14
[2010/01/28 v1.2]	14
6 Index	14

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 Argument killers

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

1.3 Argument grabbers

$\text{\ltx@firstofone} \{ \langle 1 \rangle \}$	\rightarrow	$\langle 1 \rangle$
$\text{\ltx@firstoftwo} \{ \langle 1 \rangle \} \{ \langle 2 \rangle \}$	\rightarrow	$\langle 1 \rangle$
$\text{\ltx@secondoftwo} \{ \langle 1 \rangle \} \{ \langle 2 \rangle \}$	\rightarrow	$\langle 2 \rangle$

1.4 List helpers

$\text{\ltx@car} \{ \langle 1 \rangle \} \dots \text{\@nil}$	\rightarrow	$\langle 1 \rangle$
$\text{\ltx@cdr} \{ \langle 1 \rangle \} \dots \text{\@nil}$	\rightarrow	\dots

1.5 Tail recursion

$\text{\ltx@ReturnAfterFi} \{ \langle 1 \rangle \} \text{\fi}$	\rightarrow	$\text{\fi} \langle 1 \rangle$
$\text{\ltx@ReturnAfterElseFi} \{ \langle 1 \rangle \} \text{\else} \{ \langle 2 \rangle \} \text{\fi}$	\rightarrow	$\text{\fi} \langle 1 \rangle$

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

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

If ε - \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 ε - \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 ε - \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.

1.9 Stripping

<code>\ltx@RemovePrefix</code> <code>\ltx@StripPrefix</code>

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 `\pdfirstmatch`. 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.

1.10 File management

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

1.10.1 File extensions

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

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

1.10.2 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.

1.10.3 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.

1.11 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.

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{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 [2010/01/28 v1.2 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 Argument killers

\ltx@gobble

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

\ltx@gobbletwo

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

\ltx@gobblethree

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

\ltx@gobblefour

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

2.3 Argument grabbers

\ltx@firstofone

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

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

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

2.4 List helpers

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

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

2.5 Tail recursion

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

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

2.6 Empty macro

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

2.7 Characters

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

```
\ltx@percentchar
112 \begingroup
113 \lccode'0='%\relax
114 \lowercase{\endgroup
115 \def\ltx@percentchar{0}%
116 }
```

```
\ltx@backslashchar
117 \begingroup
118 \lccode'0='\\relax
119 \lowercase{\endgroup
120 \def\ltx@backslashchar{0}%
121 }
```

2.8 Command definitions

```
\ltx@LocalExpandAfter
122 \def\ltx@LocalExpandAfter{%
123 \begingroup
124 \expandafter\expandafter\expandafter
125 \endgroup
126 \expandafter
127 }
128 \ltx@LocalExpandAfter
129 \ifx\csname ifcsname\endcsname\relax
```

```

\ltx@ifundefined
130 \def\ltx@ifundefined#1{%
131 \expandafter\ifx\csname #1\endcsname\relax
132 \expandafter\ltx@firstoftwo
133 \else
134 \expandafter\ltx@secondoftwo
135 \fi
136 }%

\ltx@ifUndefined
137 \def\ltx@ifUndefined#1{%
138 \beginngroup\expandafter\expandafter\expandafter\endgroup
139 \expandafter\ifx\csname #1\endcsname\relax
140 \expandafter\ltx@firstoftwo
141 \else
142 \expandafter\ltx@secondoftwo
143 \fi
144 }%

145 \expandafter\ltx@gobble
146 \else
147 \expandafter\ltx@firstofone
148 \fi
149 {%

\ltx@ifundefined
150 \def\ltx@ifundefined#1{%
151 \ifcsname #1\endcsname
152 \expandafter\ifx\csname #1\endcsname\relax
153 \expandafter\expandafter\expandafter\ltx@firstoftwo
154 \else
155 \expandafter\expandafter\expandafter\ltx@secondoftwo
156 \fi
157 \else
158 \expandafter\ltx@firstoftwo
159 \fi
160 }%

\ltx@ifUndefined
161 \let\ltx@ifUndefined\ltx@ifundefined
162 }

```

2.9 Stripping

```

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

\ltx@StripPrefix
164 \def\ltx@StripPrefix{%
165 \expandafter\ltx@RemovePrefix
166 }

```

2.10 File management

2.10.1 File extensions

```

\ltx@clsextension
167 \def\ltx@clsextension{%
168 \ltx@ifundefined{clsextension}{cls}\@clsextension
169 }

```


`\ltx@pkgextension`

```
170 \def\ltx@pkgextension{%
171   \ltx@ifundefined{@pkgextension}{sty}\@pkgextension
172 }
```

2.10.2 Load check

`\ltx@iffileloaded`

```
173 \def\ltx@iffileloaded#1{%
174   \ltx@ifundefined{ver@#1}\ltx@secondoftwo\ltx@firstoftwo
175 }
```

`\ltx@ifclassloaded`

```
176 \def\ltx@ifclassloaded#1{%
177   \ltx@iffileloaded{#1.\ltx@clsextension}%
178 }
```

`\ltx@ifpackageloaded`

```
179 \def\ltx@ifpackageloaded#1{%
180   \ltx@iffileloaded{#1.\ltx@pkgextension}%
181 }
```

2.10.3 Version date check

`\ltx@iffilelater`

```
182 \def\ltx@iffilelater#1#2{%
183   \ltx@iffileloaded{#1}{%
184     \expandafter\LTxcms@iflater\expandafter{%
185       \number
186       \expandafter\expandafter\expandafter\LTxcms@ParseVersion
187       \expandafter\expandafter\expandafter{%
188         \csname ver@#1\endcsname
189       }%
190     \expandafter}\expandafter{%
191       \number
192       \expandafter\LTxcms@ParseVersion\expandafter{#2}%
193     }%
194   }\ltx@secondoftwo
195 }
```

`\LTxcms@iflater`

```
196 \def\LTxcms@iflater#1#2{%
197   \ifcase 0%
198     \ifnum#1<19940101 %
199     \else
200     \ifnum#2<19940101 %
201     \else
202     \ifnum#2>#1 %
203     \else
204     1%
205     \fi
206     \fi
207     \fi
208     \ltx@space
209     \expandafter\ltx@secondoftwo
210   \else
211     \expandafter\ltx@firstoftwo
212   \fi
213 }
```

`\ltx@ifclasslater`

```
214 \def\ltx@ifclasslater#1{%
215   \ltx@ifclasslater{#1.\ltx@clsextension}%
216 }
```

`\ltx@ifpackagelater`

```
217 \def\ltx@ifpackagelater#1{%
218   \ltx@iffilelater{#1.\ltx@pkgextension}%
219 }

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

`\LTXcmds@ParseVersion`

```
221 \def\LTXcmds@ParseVersion#1{%
222   \LTXcmds@@ParseVersion#10000/00/00\@nil
223 }
```

`\LTXcmds@@ParseVersion`

```
224 \def\LTXcmds@@ParseVersion#1#2#3#4/#5#6/#7#8#9\@nil{%
225   #1#2#3#4#5#6#7#8%
226   }%

227 }{%
```

`\LTXcmds@ParseVersion`

```
228 \def\LTXcmds@ParseVersion#1{%
229   \ifnum\pdfmatch{%
230     ~%
231     (199[4-9] | [2-9] [0-9] [0-9] [0-9])/%
232     (0[1-9] | 1[0-2])/%
233     (0[1-9] | [1-2] [0-9] | 3[0-1])%
234   }{#1}=1 %
235   \ltx@StripPrefix\pdfastmatch1 %
236   \ltx@StripPrefix\pdfastmatch2 %
237   \ltx@StripPrefix\pdfastmatch3 %
238   \else
239     0%
240   \fi
241 }%
242 }
```

2.11 Macro additions

`\ltx@GlobalAppendToMacro`

```
243 \def\ltx@GlobalAppendToMacro#1#2{%
244   \ifx\ltx@undefined#1%
245     \let#1\ltx@empty
246   \else
247     \ifx\relax#1%
248       \let#1\ltx@empty
249     \fi
250   \fi
251   \begingroup
252     \toks0\expandafter{#1#2}%
253     \xdef#1{\the\toks0}%
254   \endgroup
255 }
```

`\ltx@LocalAppendToMacro`

```
256 \def\ltx@LocalAppendToMacro#1#2{%
257   \global\let\LTXcmds@gtemp#1%
```

```

258 \ifx\ltx@undefined\LTXcmds@gtemp
259   \global\let\LTXcmds@gtemp\ltx@empty
260 \else
261   \ifx\relax\LTXcmds@gtemp
262     \global\let\LTXcmds@gtemp\ltx@empty
263   \fi
264 \fi
265 \begingroup
266   \toks0\expandafter{\LTXcmds@gtemp#2}%
267   \xdef\LTXcmds@gtemp{\the\toks0}%
268 \endgroup
269 \let#1\LTXcmds@gtemp
270 }

271 \LTXcmds@AtEnd
272 </package>

```

3 Test

3.1 Catcode checks for loading

```

273 (*test1)

274 \catcode'\{=1 %
275 \catcode'\}=2 %
276 \catcode'\#=6 %
277 \catcode'\@=11 %
278 \expandafter\ifx\csname count@\endcsname\relax
279   \countdef\count@=255 %
280 \fi
281 \expandafter\ifx\csname @gobble\endcsname\relax
282   \long\def@gobble#1{%
283 \fi
284 \expandafter\ifx\csname @firstofone\endcsname\relax
285   \long\def@firstofone#1{#1}%
286 \fi
287 \expandafter\ifx\csname loop\endcsname\relax
288   \expandafter@firstofone
289 \else
290   \expandafter@gobble
291 \fi
292 {%
293   \def\loop#1\repeat{%
294     \def\body{#1}%
295     \iterate
296   }%
297   \def\iterate{%
298     \body
299     \let\next\iterate
300   \else
301     \let\next\relax
302   \fi
303   \next
304 }%
305 \let\repeat=\fi
306 }%
307 \def\RestoreCatcodes{}
308 \count@=0 %
309 \loop
310   \edef\RestoreCatcodes{%
311     \RestoreCatcodes
312     \catcode\the\count@=\the\catcode\count@\relax
313   }%

```

```

314 \ifnum\count@<255 %
315   \advance\count@ 1 %
316 \repeat
317
318 \def\RangeCatcodeInvalid#1#2{%
319   \count@=#1\relax
320   \loop
321     \catcode\count@=15 %
322   \ifnum\count@<#2\relax
323     \advance\count@ 1 %
324   \repeat
325 }
326 \expandafter\ifx\csname LoadCommand\endcsname\relax
327   \def\LoadCommand{\input ltxcmds.sty\relax}%
328 \fi
329 \def\Test{%
330   \RangeCatcodeInvalid{0}{47}%
331   \RangeCatcodeInvalid{58}{64}%
332   \RangeCatcodeInvalid{91}{96}%
333   \RangeCatcodeInvalid{123}{255}%
334   \catcode'\@=12 %
335   \catcode'\=0 %
336   \catcode'\{=1 %
337   \catcode'\}=2 %
338   \catcode'\#=6 %
339   \catcode'\ [=12 %
340   \catcode'\]=12 %
341   \catcode'\%=14 %
342   \catcode'\ =10 %
343   \catcode13=5 %
344   \LoadCommand
345   \RestoreCatcodes
346 }
347 \Test
348 \csname @@end\endcsname
349 \end
350 </test1>

```

4 Installation

4.1 Download

Package. This package is available on CTAN¹:

[CTAN:macros/latex/contrib/oberdiek/ltxcmds.dtx](https://ctan.org/ctan/packages/macros/latex/contrib/oberdiek/ltxcmds.dtx) The source file.

[CTAN:macros/latex/contrib/oberdiek/ltxcmds.pdf](https://ctan.org/ctan/packages/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](https://ctan.org/ctan/packages/install/macros/latex/contrib/oberdiek.tds.zip)

TDS refers to the standard “A Directory Structure for \TeX Files” ([CTAN:tds/teds.pdf](https://ctan.org/ctan/packages/teds/teds.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
```

¹[ftp://ftp.ctan.org/tex-archive/](https://ftp.ctan.org/tex-archive/)

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-TeX`:

```
tex ltxcmds.dtx
```

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

```
ltxcmds.sty          → tex/generic/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 (`teTeX`, `mikTeX`, ...) relies on file name databases, you must refresh these. For example, `teTeX` 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.
- `\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@GlobalAppendMacro`, `\ltx@LocalAppendMacro` 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; numbers in roman refer to the code lines where the entry is used.

Symbols	C
<code>\#</code> 276, 338	<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, 274, 275, 276, 277, 312, 321, 334, 335, 336, 337, 338, 339, 340, 341, 342, 343
<code>\%</code> 113, 341	
<code>\@</code> 277, 334	
<code>\@clsextension</code> 168	
<code>\@firstofone</code> 285, 288	
<code>\@gobble</code> 282, 290	
<code>\@nil</code> 106, 107, 222, 224	<code>\count@</code> 279, 308, 312, 314, 315, 319, 321, 322, 323
<code>\@pkgextension</code> 171	<code>\countdef</code> 279
<code>\@undefined</code> 52	<code>\csname</code> 10, 18, 44, 60, 67, 129, 131, 139, 152, 188, 278, 281, 284, 287, 326, 348
<code>\[</code> 339	
<code>\]</code> 118, 335	
<code>\{</code> 274, 336	
<code>\}</code> 275, 337	
<code>\]</code> 340	
	E
<code>_</code> 342	<code>\empty</code> 13, 14
	<code>\end</code> 349
	<code>\endcsname</code> 10, 18, 44, 60, 67, 129, 131, 139, 151, 152, 188, 278, 281, 284, 287, 326, 348
	<code>\endinput</code> 26
A	
<code>\advance</code> 315, 323	
<code>\aftergroup</code> 26	
B	
<code>\body</code> 294, 298	
	I
	<code>\ifcase</code> 197
	<code>\ifcsname</code> 151
	<code>\ifnum</code> 198, 200, 202, 229, 314, 322

<code>\ifx</code>	11, 14, 18, 44, 52, 55, 129, 131, 139, 152, 244, 247, 258, 261, 278, 281, 284, 287, 326	<code>\ltx@ReturnAfterFi</code>	2, <u>108</u> , 109
<code>\immediate</code>	20, 46	<code>\ltx@secondoftwo</code> <u>105</u> , 134, 142, 155, 174, 194, 209
<code>\input</code>	327	<code>\ltx@space</code>	3, <u>111</u> , 208
<code>\iterate</code>	295, 297, 299	<code>\ltx@StripPrefix</code> ..	<u>164</u> , 235, 236, 237
L		<code>\ltx@undefined</code>	244, 258
<code>\lccode</code>	113, 118	<code>\LTXcmds@@ParseVersion</code>	222, <u>224</u>
<code>\letLTXcmds@gttemp</code>	262	<code>\LTXcmds@AtEnd</code>	80, 81, 271
<code>\LoadCommand</code>	327, 344	<code>\LTXcmds@gttemp</code> 257, 258, 259, 261, 266, 267, 269
<code>\loop</code>	293, 309, 320	<code>\LTXcmds@IfLater</code>	184, <u>196</u>
<code>\lowercase</code>	114, 119	<code>\LTXcmds@ParseVersion</code> 186, 192, <u>221</u> , <u>228</u>
<code>\ltx@backslashchar</code>	<u>117</u>	N	
<code>\ltx@car</code>	2, <u>106</u>	<code>\next</code>	299, 301, 303
<code>\ltx@cdr</code>	<u>107</u>	<code>\number</code>	185, 191
<code>\ltx@clsextension</code> ...	4, <u>167</u> , 177, 215	P	
<code>\ltx@empty</code> ..	3, <u>110</u> , 245, 248, 259, 262	<code>\PackageInfo</code>	23
<code>\ltx@firstofone</code>	2, <u>103</u> , 147	<code>\pdflastmatch</code>	235, 236, 237
<code>\ltx@firstoftwo</code> <u>104</u> , 132, 140, 153, 158, 174, 211	<code>\pdfmatch</code>	229
<code>\ltx@GlobalAppendToMacro</code>	4, <u>243</u>	<code>\ProvidesPackage</code>	15, 61
<code>\ltx@gobble</code>	2, 99, 145	R	
<code>\ltx@gobblefour</code>	<u>102</u>	<code>\RangeCatcodeInvalid</code> 318, 330, 331, 332, 333
<code>\ltx@gobblethree</code>	<u>101</u>	<code>\repeat</code>	293, 305, 316, 324
<code>\ltx@gobbletwo</code>	<u>100</u>	<code>\RestoreCatcodes</code> ..	307, 310, 311, 345
<code>\ltx@ifclasslater</code>	4, <u>214</u>	T	
<code>\ltx@ifclassloaded</code>	4, <u>176</u>	<code>\Test</code>	329, 347
<code>\ltx@iffilelater</code>	<u>182</u> , 218	<code>\the</code> ...	68, 69, 70, 71, 82, 253, 267, 312
<code>\ltx@iffileloaded</code>	4, <u>173</u> , 177, 180, 183	<code>\TMP@EnsureCode</code> ...	79, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98
<code>\ltx@ifpackagelater</code>	<u>217</u>	<code>\toks</code>	252, 253, 266, 267
<code>\ltx@ifpackageloaded</code>	<u>179</u>	W	
<code>\ltx@ifundefined</code> ...	3, <u>137</u> , <u>161</u> , 220	<code>\write</code>	20, 46
<code>\ltx@ifundefined</code> 3, <u>130</u> , <u>150</u> , 161, 168, 171, 174	X	
<code>\ltx@LocalAppendToMacro</code>	<u>256</u>	<code>\x</code>	10, 11, 14, 19, 23, 25, 45, 50, 60, 66, 74
<code>\ltx@LocalExpandAfter</code> ...	3, <u>122</u> , 128		
<code>\ltx@percentchar</code>	<u>112</u>		
<code>\ltx@pkgextension</code>	<u>170</u> , 180, 218		
<code>\ltx@RemovePrefix</code>	3, <u>163</u> , 165		
<code>\ltx@ReturnAfterElseFi</code>	<u>109</u>		