

The ltxcmds package

Heiko Oberdiek
<oberdiek@uni-freiburg.de>

2009/08/05 v1.0

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

1 Documentation

1.1 Introduction

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

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

1.2 Argument killers

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

1.3 Argument grabbers

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

1.4 List helpers

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

1.5 Tail recursion

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

1.6 Empty macro

<code>\ltx@empty</code>

1.7 Characters

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

1.8 Command definitions

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

If ε -TeX is available, `\ifcsname` is used that does not have the side effect of defining undefined commands with meaning of `\relax`.

`\ltx@LocalExpandAfter`

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

2 Implementation

2.1 Identification

```
1 (*package)
```

Reload check, especially if the package is not used with L^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 [2009/08/05 v1.0 Some LaTeX kernel commands for general use (H0)]
63 \begingroup
64 \catcode123 1 % {
65 \catcode125 2 % }
66 \def\x{\endgroup
67   \expandafter\edef\csname ltx@AtEnd\endcsname{%
68     \catcode35 \the\catcode35\relax
69     \catcode64 \the\catcode64\relax
70     \catcode123 \the\catcode123\relax
71     \catcode125 \the\catcode125\relax
72   }%
73 }%
74 \x
75 \catcode35 6 % #
76 \catcode64 11 % @
77 \catcode123 1 % {
78 \catcode125 2 % }
79 \def\TMP@EnsureCode#1#2{%
80   \edef\ltx@AtEnd{%
81     \ltx@AtEnd
82     \catcode#1 \the\catcode#1\relax
83   }%
84   \catcode#1 #2\relax
85 }
86 \TMP@EnsureCode{61}{12}% =
87 \TMP@EnsureCode{96}{12}% ‘

```

2.2 Argument killers

\ltx@gobble

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

\ltx@gobbletwo

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

\ltx@gobblethree

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

\ltx@gobblefour

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

2.3 Argument grabbers

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

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

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

2.4 List helpers

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

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

2.5 Tail recursion

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

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

2.6 Empty macro

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

2.7 Characters

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

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

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

2.8 Command definitions

`\ltx@ifundefined`

```
111 \begingroup\expandafter\expandafter\expandafter\endgroup
112 \expandafter\ifx\csname ifcsname\endcsname\relax
113   \def\ltx@ifundefined#1{%
114     \expandafter\ifx\csname #1\endcsname\relax
115       \expandafter\ltx@firstoftwo
116     \else
117       \expandafter\ltx@secondoftwo
118     \fi
119   }%
120   \expandafter\ltx@gobble
121 \else
122   \expandafter\ltx@firstofone
123 \fi
124 {%
125   \def\ltx@ifundefined#1{%
126     \ifcsname #1\endcsname
127       \expandafter\ltx@secondoftwo
128     \else
129       \expandafter\ltx@firstoftwo
130     \fi
131   }%
132 }
```

`\ltx@LocalExpandAfter`

```
133 \def\ltx@LocalExpandAfter{%
134   \begingroup
135     \expandafter\expandafter\expandafter
136   \endgroup
137   \expandafter
138 }

139 \ltx@AtEnd
140 </package>
```

3 Test

3.1 Catcode checks for loading

```
141 <test1>
142 \catcode'\{=1 %
143 \catcode'\}=2 %
144 \catcode'\#=6 %
145 \catcode'\@=11 %
146 \expandafter\ifx\csname count@\endcsname\relax
147   \countdef\count@=255 %
148 \fi
149 \expandafter\ifx\csname @gobble\endcsname\relax
150   \long\def\@gobble#1{}%
151 \fi
152 \expandafter\ifx\csname @firstofone\endcsname\relax
153   \long\def\@firstofone#1{#1}%
154 \fi
155 \expandafter\ifx\csname loop\endcsname\relax
156   \expandafter\@firstofone
157 \else
158   \expandafter\@gobble
159 \fi
160 {%
```

```

161 \def\loop#1\repeat{%
162   \def\body{#1}%
163   \iterate
164 }%
165 \def\iterate{%
166   \body
167   \let\next\iterate
168   \else
169     \let\next\relax
170   \fi
171   \next
172 }%
173 \let\repeat=\fi
174 }%
175 \def\RestoreCatcodes{}
176 \count@=0 %
177 \loop
178   \edef\RestoreCatcodes{%
179     \RestoreCatcodes
180     \catcode\the\count@=\the\catcode\count@\relax
181   }%
182   \ifnum\count@<255 %
183     \advance\count@ 1 %
184   \repeat
185
186 \def\RangeCatcodeInvalid#1#2{%
187   \count@=#1\relax
188   \loop
189     \catcode\count@=15 %
190     \ifnum\count@<#2\relax
191       \advance\count@ 1 %
192     \repeat
193 }
194 \expandafter\ifx\csname LoadCommand\endcsname\relax
195   \def\LoadCommand{\input ltxcmds.sty\relax}%
196 \fi
197 \def\Test{%
198   \RangeCatcodeInvalid{0}{47}%
199   \RangeCatcodeInvalid{58}{64}%
200   \RangeCatcodeInvalid{91}{96}%
201   \RangeCatcodeInvalid{123}{255}%
202   \catcode'\@=12 %
203   \catcode'\=0 %
204   \catcode'\{=1 %
205   \catcode'\}=2 %
206   \catcode'\#=6 %
207   \catcode'\[=12 %
208   \catcode'\]=12 %
209   \catcode'\%=14 %
210   \catcode'\ =10 %
211   \catcode\l3=5 %
212   \LoadCommand
213   \RestoreCatcodes
214 }
215 \Test
216 \csname @@end\endcsname
217 \end
218 </test1>

```

4 Installation

4.1 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 T_EX 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_EX:

```
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>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`.

4.4 Refresh file name databases

If your T_EX distribution (teT_EX, miK_TE_X, ...) relies on file name databases, you must refresh these. For example, teT_EX users run `texhash` or `mktextlsr`.

¹<http://ftp.ctan.org/tex-archive/>

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.

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	
<code>\#</code>	144, 206
<code>\%</code>	102, 209
<code>\@</code>	145, 202
<code>\@firstofone</code>	153, 156
<code>\@gobble</code>	150, 158
<code>\@nil</code>	95, 96
<code>\@undefined</code>	52
<code>\[</code>	207
<code>\]</code>	107, 203
<code>\{</code>	142, 204
<code>\}</code>	143, 205
<code>\]</code>	208
<code>_</code>	210
A	
<code>\advance</code>	183, 191
<code>\aftergroup</code>	26
B	
<code>\body</code>	162, 166
C	
<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, 142, 143, 144, 145, 180, 189, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211	\ltx@firstoftwo <u>93</u> , <u>115</u> , <u>129</u>
\count@ 147, 176, 180, 182, 183, 187, 189, 190, 191	\ltx@gobble <u>2</u> , <u>88</u> , <u>120</u>
\countdef 147	\ltx@gobblefour <u>91</u>
\csname 10, 18, 44, 60, 67, 112, 114, 146, 149, 152, 155, 194, 216	\ltx@gobblethree <u>90</u>
	\ltx@gobbletwo <u>89</u>
	\ltx@ifundefined <u>3</u> , <u>111</u>
	\ltx@LocalExpandAfter <u>3</u> , <u>133</u>
	\ltx@percentchar <u>101</u>
	\ltx@ReturnAfterElseFi <u>98</u>
	\ltx@ReturnAfterFi <u>2</u> , <u>97</u> , <u>98</u>
	\ltx@secondoftwo <u>94</u> , <u>117</u> , <u>127</u>
	\ltx@space <u>2</u> , <u>100</u>