

The catchfile package

Heiko Oberdiek
<heiko.oberdiek at gmail.com>

2010/04/28 v1.5

Abstract

This package catches the contents of a file and puts it in a macro. It requires ϵ -TeX. Both L^ATeX and plain TeX are supported.

Contents

1	Documentation	1
2	Implementation	2
2.1	Reload check and package identification	2
2.2	Catcodes	3
2.3	Preparations	3
2.4	Looking for primitive <code>\input</code>	4
2.5	Input file check	4
2.6	Catch file contents	5
3	Test	6
3.1	Catcode checks for loading	6
3.2	L ^A TeX	8
3.3	plain TeX	9
4	Installation	11
4.1	Download	11
4.2	Bundle installation	11
4.3	Package installation	11
4.4	Refresh file name databases	12
4.5	Some details for the interested	12
5	History	12
	[2007/05/30 v1.0]	12
	[2007/09/09 v1.1]	12
	[2007/11/11 v1.2]	12
	[2010/03/01 v1.3]	12
	[2010/04/08 v1.4]	13
	[2010/04/28 v1.5]	13
6	Index	13

1 Documentation

The package relies on ϵ -TeX's `\everyeof`. Otherwise it aborts with an error message.

<pre> \CatchFileDef {<cmd>} {<file name>} {<setup>} \CatchFileEdef {<cmd>} {<file name>} {<setup>} </pre>

Macro `<cmd>` is defined with the contents of file `<file name>`. `\CatchFileDef` uses `\def`, `\CatchFileEdef` `\edef` for the definition. Additional setup code for setting catcodes or treatment of line ends can be given in code `<setup>`. See the test files for an example.

2 Implementation

```
1 (*package)
```

2.1 Reload check and package identification

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@catchfile.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{catchfile}{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@catchfile.sty\endcsname
61 \ProvidesPackage{catchfile}%
62 [2010/04/28 v1.5 Catches the contents of a file (HO)]

```

2.2 Catcodes

```

63 \begingroup
64 \catcode123 1 % {
65 \catcode125 2 % }
66 \def\x{\endgroup
67   \expandafter\edef\csname CatchFile@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\CatchFile@AtEnd{%
81     \CatchFile@AtEnd
82     \catcode#1 \the\catcode#1\relax
83   }%
84   \catcode#1 #2\relax
85 }
86 \TMP@EnsureCode{39}{12}% '
87 \TMP@EnsureCode{44}{12}% ,
88 \TMP@EnsureCode{45}{12}% -
89 \TMP@EnsureCode{46}{12}% .
90 \TMP@EnsureCode{47}{12}% /
91 \TMP@EnsureCode{61}{12}% =
92 \TMP@EnsureCode{96}{12}% `

```

2.3 Preparations

```

93 \begingroup\expandafter\expandafter\expandafter\endgroup
94 \expandafter\ifx\csname RequirePackage\endcsname\relax
95   \input infwarerr.sty\relax
96   \input ltxcmds.sty\relax
97 \else
98   \RequirePackage{infwarerr}[2007/09/09]%
99   \RequirePackage{ltxcmds}[2010/03/09]%
100 \fi

```

Check for ε -TeX's `\everyeof`.

```

101 \begingroup
102 \escapechar=92\relax

```

```

103 \edef\TestString{\string\everyeof}%
104 \edef\TestMeaning{\meaning\everyeof}%
105 \ifx\TestString\TestMeaning
106 \else
107   \@PackageError{catchfile}{%
108     Cannot find e-TeX's \string\everyeof,\MessageBreak
109     package loading is aborted%
110   }\@ehd
111 \endgroup
112 \CatchFile@AtEnd
113 \expandafter\endinput
114 \fi
115 \endgroup

```

2.4 Looking for primitive `\input`

`\CatchFile@Input` The package needs the expandable primitive `\input`. However there are formats that redefine it. For example, L^AT_EX's `\input` is not expandable, but it stores the primitive in `\@@input`. The third possibility is `\pdfprimitive`, introduced in pdfT_EX 1.40.0.

Thus we try to find the primitive and store it in `\CatchFile@Input`. If it is used, it must be expanded twice (because of the solution with `\pdfprimitive`).

```

116 \begingroup
117 \def\Check#1#2#3#4\endgroup{%
118   \edef\TestString{\string#1}%
119   \edef\TestMeaning{\meaning#2}%
120   \ifx\TestString\TestMeaning
121     \endgroup
122     \let\CatchFile@Primitive#2%
123     \def\CatchFile@Input{\CatchFile@Primitive#3}%
124   \else
125     #4\endgroup
126   \fi
127 }%
128 \Check\input\input{}%
129 \Check\input\@@input{}%
130 \Check\pdfprimitive\pdfprimitive\input
131 \@PackageError{%
132   Cannot find primitive \string\input,\MessageBreak
133   package loading is aborted%
134 }\@ehd
135 \csname endgroup\endcsname
136 \CatchFile@AtEnd
137 \endinput
138 \endgroup

```

2.5 Input file check

`\CatchFile@CheckFileExists`

```

139 \begingroup\expandafter\expandafter\expandafter\endgroup
140 \expandafter\ifx\csname IfFileExists\endcsname\relax
141   \input pdftexcmds.sty\relax
142 \begingroup\expandafter\expandafter\expandafter\endgroup
143 \expandafter\ifx\csname pdf@filesize\endcsname\relax
144   \def\CatchFile@CheckFileExists#1{%
145     \expandafter\ifx\csname @inputcheck\endcsname\relax
146       \csname newread\endcsname\@inputcheck
147     \fi
148     \openin\@inputcheck#1\relax
149     \expandafter\closein\expandafter\@inputcheck
150     \ifeof\@inputcheck
151       \let\CatchFile@File\relax

```

```

152     \else
153     \def\CatchFile@File{#1}%
154     \fi
155     }%
156 \else
157     \def\CatchFile@CheckFileExists#1{%
158     \expandafter\expandafter\expandafter\ifx
159     \expandafter\expandafter\expandafter\relax\pdf@filesize{#1}\relax
160     \let\CatchFile@File\relax
161     \else
162     \def\CatchFile@File{#1}%
163     \fi
164     }%
165 \fi
166 \else
167 \def\CatchFile@CheckFileExists#1{%
168 \IfFileExists{#1}{%
169     \expandafter\CatchFile@DefFile\@filef@und\@nil
170     \begingroup\expandafter\expandafter\expandafter\endgroup
171     \expandafter\ifx\csname @addtofilelist\endcsname\relax
172     \else
173     \@addtofilelist\CatchFile@File
174     \fi
175     }{%
176     \let\CatchFile@File\relax
177     }%
178 }%
179 \def\CatchFile@DefFile#1 \@nil{%
180 \def\CatchFile@File{#1}%
181 }%
182 \fi

```

\CatchFile@NotFound

```

183 \def\CatchFile@NotFound#1#2{%
184 \def#1{}}%
185 \@PackageError{catchfile}{%
186 File '#2' not found%
187 }\@ehc
188 }

```

2.6 Catch file contents

```

189 \ltx@ifundefined{RequirePackage}{%
190 \input etexcmds.sty\relax
191 }{%
192 \RequirePackage{etexcmds}[2010/01/28]%
193 }

```

\CatchFileEdef

```

194 \long\def\CatchFileEdef#1#2#3{%
195 \CatchFile@CheckFileExists{#2}%
196 \ifx\CatchFile@File\relax
197 \CatchFile@NotFound{#1}{#2}%
198 \else
199 \begingroup
200 \everyeof{\noexpand}%
201 #3%
202 \xdef\CatchFile@Contents{\CatchFile@Input\CatchFile@File\space}%
203 \endgroup
204 \let#1\CatchFile@Contents
205 \fi
206 }

```

\CatchFileDef

```
207 \long\def\CatchFileDef#1#2#3{%
208   \CatchFile@CheckFileExists{#2}%
209   \ifx\CatchFile@File\relax
210     \CatchFile@NotFound{#1}{#2}%
211   \else
212     \begingroup
213     \everyeof\expandafter{%
214       \CatchFile@EOF
215       \expandafter\CatchFile@Finish
216     \noexpand
217   }%
218   \ltx@ifundefined{etex@unexpanded}{%
219     \expandafter\long\expandafter\def\expandafter\CatchFile@Do
220       \expandafter##\expandafter1\CatchFile@EOF{%
221         \toks\ltx@zero{##1}%
222         \xdef\CatchFile@gtemp{\the\toks\ltx@zero}%
223         \def\CatchFile@Finish{%
224           \endgroup
225           \let#1\CatchFile@gtemp
226           \global\let\CatchFile@gtemp\ltx@undefined
227         }%
228       }%
229     }{%
230       \expandafter\long\expandafter\def\expandafter\CatchFile@Do
231         \expandafter##\expandafter1\CatchFile@EOF{%
232           \edef\CatchFile@Finish{%
233             \endgroup
234             \etex@unexpanded{%
235               \edef#1{\etex@unexpanded{##1}}%
236             }%
237           }%
238         }%
239       }%
240     #3%
241   \expandafter\expandafter\expandafter\CatchFile@Do
242   \CatchFile@Input\CatchFile@File\relax
243 \fi
244 }
```

\CatchFile@EOF If the file is read the catcode mappings are fixed. This means that the same character cannot occur inside the file with different catcodes. Thus we use as end of file marker the at sign twice with different catcodes.

```
245 \begingroup
246 \lccode65=64 % lowercase('A') = '@'
247 \lccode66=64 % lowercase('B') = '@'
248 \catcode65=8 % catcode('A') = subscript
249 \catcode66=3 % catcode('B') = math shift
250 \lowercase{\endgroup
251 \def\CatchFile@EOF{AB}%
252 }
```

```
253 \CatchFile@AtEnd
254 </package>
```

3 Test

3.1 Catcode checks for loading

```
255 <*test1>
256 \catcode'\{=1 %
```

```

257 \catcode'\}=2 %
258 \catcode'\#=6 %
259 \catcode'\@=11 %
260 \expandafter\ifx\csname count@\endcsname\relax
261 \countdef\count@=255 %
262 \fi
263 \expandafter\ifx\csname @gobble\endcsname\relax
264 \long\def@gobble#1{%
265 \fi
266 \expandafter\ifx\csname @firstofone\endcsname\relax
267 \long\def@firstofone#1{#1}%
268 \fi
269 \expandafter\ifx\csname loop\endcsname\relax
270 \expandafter@firstofone
271 \else
272 \expandafter@gobble
273 \fi
274 {%
275 \def\loop#1\repeat{%
276 \def\body{#1}%
277 \iterate
278 }%
279 \def\iterate{%
280 \body
281 \let\next\iterate
282 \else
283 \let\next\relax
284 \fi
285 \next
286 }%
287 \let\repeat=\fi
288 }%
289 \def\RestoreCatcodes{}
290 \count@=0 %
291 \loop
292 \edef\RestoreCatcodes{%
293 \RestoreCatcodes
294 \catcode\the\count@=\the\catcode\count@\relax
295 }%
296 \ifnum\count@<255 %
297 \advance\count@ 1 %
298 \repeat
299
300 \def\RangeCatcodeInvalid#1#2{%
301 \count@=#1\relax
302 \loop
303 \catcode\count@=15 %
304 \ifnum\count@<#2\relax
305 \advance\count@ 1 %
306 \repeat
307 }
308 \expandafter\ifx\csname LoadCommand\endcsname\relax
309 \def\LoadCommand{\input catchfile.sty\relax}%
310 \fi
311 \def\Test{%
312 \RangeCatcodeInvalid{0}{47}%
313 \RangeCatcodeInvalid{58}{64}%
314 \RangeCatcodeInvalid{91}{96}%
315 \RangeCatcodeInvalid{123}{255}%
316 \catcode'\@=12 %
317 \catcode'\}=0 %
318 \catcode'\{=1 %

```

```

319 \catcode'\}=2 %
320 \catcode'\#=6 %
321 \catcode'\[=12 %
322 \catcode'\]=12 %
323 \catcode'\%=14 %
324 \catcode'\ =10 %
325 \catcode13=5 %
326 \LoadCommand
327 \RestoreCatcodes
328 }
329 \Test
330 \csname @@end\endcsname
331 \end
332 </test1>

```

3.2 L^AT_EX

```

333 (*test2)
334 \NeedsTeXFormat{LaTeX2e}
335 \nofiles
336 \listfiles
337 \tracingnesting=2 %
338 \documentclass{minimal}
339 \usepackage{catchfile}[2010/04/28]
340 \makeatletter
341 \def\mysetup{%
342   \let\do\@makeother
343   \dospecials
344 }
345 \def\CheckContents{%
346   \begingroup
347     \expandafter\ifx\csname contents\endcsname\relax
348       \PackageError{TEST}{\string\contents\space is not defined}\@ehc
349     \fi
350   \endgroup
351 }
352 \def\StartDisableUnexpanded{%
353   \begingroup
354     \let\etex@unexpanded\@undefined
355 }
356 \def\StopDisableUnexpanded{%
357   \endgroup
358 }
359 \def\CheckCleanup{%
360   \begingroup
361     \edef\x{\the\toks0}%
362     \def\y{ABC}%
363     \ifx\x\y
364     \else
365       \PackageError{TEST}{\string\toks0 has changed}\@ehc
366     \fi
367     \ifx\CatchFile@gtemp\@undefined
368     \else
369       \PackageError{TEST}{Left over \string\CatchFile@gtemp}\@ehc
370     \fi
371   \endgroup
372 }
373 \def\Check{%
374   \CheckContents
375   \CheckCleanup
376 }
377 \makeatother
378 \begin{document}

```



```

379 \toks0{ABC}
380
381 \CatchFileDef\contents{catchfile.sty}\mysetup
382 \typeout{\meaning\contents}
383 \Check
384 \typeout{*****}%
385 \CatchFileDef\contents{catchfile.sty}{}
386 \typeout{\meaning\contents}
387 \Check
388 \typeout{*****}%
389
390 \StartDisableUnexpanded
391 \CatchFileDef\contents{catchfile.sty}\mysetup
392 \typeout{\meaning\contents}
393 \Check
394 \typeout{*****}%
395 \CatchFileDef\contents{catchfile.sty}{}
396 \typeout{\meaning\contents}
397 \Check
398 \typeout{*****}%
399 \StopDisableUnexpanded
400
401 \CatchFileEdef\contents{catchfile.sty}{%
402   \mysetup
403   \def\par{^~J}%
404   \obeylines
405 }
406 \typeout{\contents}
407 \Check
408 \typeout{*****}%
409 \CatchFileEdef\contents{catchfile.sty}{%
410   \catcode'\=12 %
411   \catcode'\#=12 %
412   \def\par{^^J}%
413   \obeylines
414 }
415 \typeout{\contents}
416 \Check
417 \typeout{*****}%
418 \end{document}
419 </test2>

```

3.3 plain T_EX

```

420 (*test3)
421 \def\msg#{\immediate\write16}
422 \newlinechar=10 %
423 \tracingnesting=2 %
424 \input catchfile.sty\relax
425
426 \def\mysetup{%
427   \def\do##1{%
428     \catcode'##1=12\relax
429   }%
430   \dospecials
431 }
432 \def\CheckContents{%
433   \begingroup
434   \expandafter\ifx\csname contents\endcsname\relax
435   \def\space{ }%
436   \errmessage{\string\contents\space is not defined.}%
437   \fi
438 \endgroup

```

```

439 }
440 \catcode'\@=11 %
441 \def\CheckCleanup{%
442   \begingroup
443   \edef\x{\the\toks0}%
444   \def\y{ABC}%
445   \ifx\x\y
446   \else
447     \errmessage{\string\toks0 has changed}%
448   \fi
449   \ifx\CatchFile@gtemp\@undefined
450   \else
451     \errmessage{Left over \string\CatchFile@gtemp}%
452   \fi
453   \endgroup
454 }
455 \def\Check{%
456   \CheckContents
457   \CheckCleanup
458 }
459 \def\StartDisableUnexpanded{%
460   \begingroup
461   \let\etex@unexpanded\@undefined
462 }
463 \def\StopDisableUnexpanded{%
464   \endgroup
465 }
466 \catcode'\@=12 %
467
468 \toks0{ABC}
469
470 \CatchFileDef\contents{catchfile.sty}\mysetup
471 \msg{\meaning\contents}
472 \Check
473 \msg{*****}%
474 \CatchFileDef\contents{catchfile.sty}{ }
475 \msg{\meaning\contents}
476 \Check
477 \msg{*****}%
478
479 \StartDisableUnexpanded
480 \CatchFileDef\contents{catchfile.sty}\mysetup
481 \msg{\meaning\contents}
482 \Check
483 \msg{*****}%
484 \CatchFileDef\contents{catchfile.sty}{ }
485 \msg{\meaning\contents}
486 \Check
487 \msg{*****}%
488 \StopDisableUnexpanded
489
490 \CatchFileEdef\contents{catchfile.sty}{%
491   \mysetup
492   \def\par{^^J}%
493   \obeylines
494 }
495 \msg{\contents}
496 \Check
497 \msg{*****}%
498 \CatchFileEdef\contents{catchfile.sty}{%
499   \catcode'\=12 %
500   \catcode'\#=12 %

```

```

501 \def\par{^^J}%
502 \obeylines
503 }
504 \msg{\contents}
505 \Check
506 \msg{*****}%
507
508 \csname @@end\endcsname
509 \end
510 </test3>

```

4 Installation

4.1 Download

Package. This package is available on CTAN¹:

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

[CTAN:macros/latex/contrib/oberdiek/catchfile.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 catchfile.dtx
```

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

```

catchfile.sty      → tex/generic/oberdiek/catchfile.sty
catchfile.pdf     → doc/latex/oberdiek/catchfile.pdf
test/catchfile-test1.tex → doc/latex/oberdiek/test/catchfile-test1.tex
test/catchfile-test2.tex → doc/latex/oberdiek/test/catchfile-test2.tex
test/catchfile-test3.tex → doc/latex/oberdiek/test/catchfile-test3.tex
catchfile.dtx     → source/latex/oberdiek/catchfile.dtx

```

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

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 `mktexlsr`.

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 catchfile.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{catchfile.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 catchfile.dtx
makeindex -s gind.ist catchfile.idx
pdflatex catchfile.dtx
makeindex -s gind.ist catchfile.idx
pdflatex catchfile.dtx
```

5 History

[2007/05/30 v1.0]

- First version.

[2007/09/09 v1.1]

- Catcode section rewritten.

[2007/11/11 v1.2]

- Use of package `pdftexcmds` for $\text{Lua}\TeX$ support.

[2010/03/01 v1.3]

- Fix for unknown `\@PackageErrorNoLine`.

[2010/04/08 v1.4]

- `\closein` also added if `\ifeof` is true.

[2010/04/28 v1.5]

- `\CatchFileDef`: Getting rid of warning ‘end of semi simple group entered at line ... of a different file (Florent Chervet).
- `\CatchFileDef`: Fix for error ‘Illegal parameter number in definition of ...’ (HO) including improvement that uses `\unexpanded` if available (Florent Chervet).

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	
<code>\#</code>	258, 320, 411, 500
<code>\%</code>	323
<code>\@</code>	259, 316, 440, 466
<code>\@input</code>	129
<code>\@PackageError</code>	107, 131, 185
<code>\@addtofilelist</code>	173
<code>\@ehc</code>	187, 348, 365, 369
<code>\@ehd</code>	110, 134
<code>\@fileofund</code>	169
<code>\@firstofone</code>	267, 270
<code>\@gobble</code>	264, 272
<code>\@inputcheck</code>	146, 148, 149, 150
<code>\@makeother</code>	342
<code>\@nil</code>	169, 179
<code>\@undefined</code>	52, 354, 367, 449, 461
<code>\[</code>	321
<code>\]</code>	317, 410, 499
<code>\{</code>	256, 318
<code>\}</code>	257, 319
<code>\]</code>	322
<code>_</code>	324
A	
<code>\advance</code>	297, 305
<code>\aftergroup</code>	26
B	
<code>\begin</code>	378
<code>\body</code>	276, 280
C	
<code>\CatchFile@AtEnd</code>	80, 81, 112, 136, 253
<code>\CatchFile@CheckFileExists</code>	139, 195, 208
<code>\CatchFile@Contents</code>	202, 204
<code>\CatchFile@DefFile</code>	169, 179
<code>\CatchFile@Do</code>	219, 230, 241
<code>\CatchFile@EOF</code>	214, 220, 231, 245
<code>\CatchFile@File</code>	151, 153, 160, 162, 173, 176, 180, 196, 202, 209, 242
<code>\CatchFile@Finish</code>	215, 223, 232
<code>\CatchFile@gtmp</code>	222, 225, 226, 367, 369, 449, 451
<code>\CatchFile@Input</code>	116, 202, 242
<code>\CatchFile@NotFound</code> ...	183, 197, 210
<code>\CatchFile@Primitive</code>	122, 123
<code>\CatchFileDef</code>	2, 207, 381, 385, 391, 395, 470, 474, 480, 484
<code>\CatchFileEdef</code>	194, 401, 409, 490, 498
<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, 248, 249, 256, 257, 258, 259, 294, 303, 316, 317, 318, 319, 320, 321, 322, 323, 324, 325, 410, 411, 428, 440, 466, 499, 500
<code>\Check</code>	117, 128, 129, 130, 373, 383, 387, 393, 397, 407, 416, 455, 472, 476, 482, 486, 496, 505
<code>\CheckCleanup</code>	359, 375, 441, 457
<code>\CheckContents</code>	345, 374, 432, 456
<code>\closein</code>	149
<code>\contents</code>	348, 381, 382, 385, 386, 391, 392, 395, 396, 401, 406, 409, 415, 436, 470, 471, 474, 475, 480, 481, 484, 485, 490, 495, 498, 504
<code>\count@</code>	261, 290, 294, 296, 297, 301, 303, 304, 305
<code>\countdef</code>	261
<code>\csname</code>	10, 18, 44, 60, 67, 94, 135, 140, 143, 145, 146, 171, 260, 263, 266, 269, 308, 330, 347, 434, 508
D	
<code>\do</code>	342, 427
<code>\documentclass</code>	338
<code>\dospecials</code>	343, 430
E	
<code>\empty</code>	13, 14
<code>\end</code>	331, 418, 509

<code>\endcsname</code>	10, 18, 44, 60, 67, 94, 135, 140, 143, 145, 146, 171, 260, 263, 266, 269, 308, 330, 347, 434, 508		
<code>\endinginput</code>	26, 113, 137		
<code>\errmessage</code>	436, 447, 451		
<code>\escapechar</code>	102		
<code>\etex@unexpanded</code> ..	234, 235, 354, 461		
<code>\everyeof</code>	103, 104, 108, 200, 213		
I			
<code>\ifeof</code>	150		
<code>\IfFileExists</code>	168		
<code>\ifnum</code>	296, 304		
<code>\ifx</code> .. 11, 14, 18, 44, 52, 55, 94, 105, 120, 140, 143, 145, 158, 171, 196, 209, 260, 263, 266, 269, 308, 347, 363, 367, 434, 445, 449			
<code>\immediate</code>	20, 46, 421		
<code>\input</code>	95, 96, 128, 129, 130, 132, 141, 190, 309, 424		
<code>\iterate</code>	277, 279, 281		
L			
<code>\lccode</code>	246, 247		
<code>\listfiles</code>	336		
<code>\LoadCommand</code>	309, 326		
<code>\loop</code>	275, 291, 302		
<code>\lowercase</code>	250		
<code>\ltx@ifundefined</code>	189		
<code>\ltx@ifundefined</code>	218		
<code>\ltx@undefined</code>	226		
<code>\ltx@zero</code>	221, 222		
M			
<code>\makeatletter</code>	340		
<code>\makeatother</code>	377		
<code>\meaning</code>	104, 119, 382, 386, 392, 396, 471, 475, 481, 485		
<code>\MessageBreak</code>	108, 132		
<code>\msg</code> ...	421, 471, 473, 475, 477, 481, 483, 485, 487, 495, 497, 504, 506		
<code>\mysetup</code>	341, 381, 391, 402, 426, 470, 480, 491		
N			
<code>\NeedsTeXFormat</code>	334		
<code>\newlinechar</code>	422		
<code>\next</code>	281, 283, 285		
<code>\nofiles</code>	335		
O			
<code>\obeylines</code>	404, 413, 493, 502		
<code>\openin</code>	148		
P			
<code>\PackageError</code>	348, 365, 369		
<code>\PackageInfo</code>	23		
<code>\par</code>	403, 412, 492, 501		
<code>\pdf@filesize</code>	159		
<code>\pdfprimitive</code>	130		
<code>\ProvidesPackage</code>	15, 61		
R			
<code>\RangeCatcodeInvalid</code>	300, 312, 313, 314, 315		
<code>\repeat</code>	275, 287, 298, 306		
<code>\RequirePackage</code>	98, 99, 192		
<code>\RestoreCatcodes</code> ..	289, 292, 293, 327		
S			
<code>\space</code>	202, 348, 435, 436		
<code>\StartDisableUnexpanded</code>	352, 390, 459, 479		
<code>\StopDisableUnexpanded</code>	356, 399, 463, 488		
T			
<code>\Test</code>	311, 329		
<code>\TestMeaning</code>	104, 105, 119, 120		
<code>\TestString</code>	103, 105, 118, 120		
<code>\the</code> 68, 69, 70, 71, 82, 222, 294, 361, 443			
<code>\TMP@EnsureCode</code>	79, 86, 87, 88, 89, 90, 91, 92		
<code>\toks</code>	221, 222, 361, 365, 379, 443, 447, 468		
<code>\tracingnesting</code>	337, 423		
<code>\typeout</code> ..	382, 384, 386, 388, 392, 394, 396, 398, 406, 408, 415, 417		
U			
<code>\usepackage</code>	339		
W			
<code>\write</code>	20, 46, 421		
X			
<code>\x</code>	10, 11, 14, 19, 23, 25, 45, 50, 60, 66, 74, 361, 363, 443, 445		
Y			
<code>\y</code>	362, 363, 444, 445		