

# The `catchfile` package

Heiko Oberdiek  
<oberdiek@uni-freiburg.de>

2007/11/11 v1.2

## Abstract

This package catches the contents of a file and puts it in a macro. It requires  $\varepsilon$ -TeX. Both L<sup>A</sup>T<sub>E</sub>X and plain-T<sub>E</sub>X are supported.

## Contents

<b>1</b>	<b>Documentation</b>	<b>1</b>
<b>2</b>	<b>Implementation</b>	<b>2</b>
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
<b>3</b>	<b>Test</b>	<b>6</b>
3.1	Catcode checks for loading . . . . .	6
3.2	L <sup>A</sup> T <sub>E</sub> X . . . . .	7
3.3	plain-T <sub>E</sub> X . . . . .	8
<b>4</b>	<b>Installation</b>	<b>8</b>
4.1	Download . . . . .	8
4.2	Bundle installation . . . . .	9
4.3	Package installation . . . . .	9
4.4	Refresh file name databases . . . . .	9
4.5	Some details for the interested . . . . .	9
<b>5</b>	<b>History</b>	<b>10</b>
	[2007/05/30 v1.0] . . . . .	10
	[2007/09/09 v1.1] . . . . .	10
	[2007/11/11 v1.2] . . . . .	10
<b>6</b>	<b>Index</b>	<b>10</b>

## 1 Documentation

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

<code>\CatchFileDef</code> $\langle cmd \rangle$ $\langle file name \rangle$ $\langle setup \rangle$ <code>\CatchFileEdef</code> $\langle cmd \rangle$ $\langle file name \rangle$ $\langle setup \rangle$
---

Macro  $\langle cmd \rangle$  is defined with the contents of file  $\langle file name \rangle$ . `\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<sup>A</sup>T<sub>E</sub>X.

```
2 \begingroup
3 \catcode44 12 % ,
4 \catcode45 12 % -
5 \catcode46 12 % .
6 \catcode58 12 % :
7 \catcode64 11 % @
8 \expandafter\let\expandafter\x\csname ver@catchfile.sty\endcsname
9 \ifcase 0%
10 \ifx\x\relax % plain
11 \else
12 \ifx\x\empty % LaTeX
13 \else
14 1%
15 \fi
16 \fi
17 \else
18 \catcode35 6 % #
19 \catcode123 1 % {
20 \catcode125 2 % }
21 \expandafter\ifx\csname PackageInfo\endcsname\relax
22 \def\x#1#2{%
23 \immediate\write-1{Package #1 Info: #2.}%
24 }%
25 \else
26 \def\x#1#2{\PackageInfo{#1}{#2, stopped}}%
27 \fi
28 \x{catchfile}{The package is already loaded}%
29 \endgroup
30 \expandafter\endinput
31 \fi
32 \endgroup
```

Package identification:

```
33 \begingroup
34 \catcode35 6 % #
35 \catcode40 12 % (
36 \catcode41 12 % )
37 \catcode44 12 % ,
38 \catcode45 12 % -
39 \catcode46 12 % .
40 \catcode47 12 % /
41 \catcode58 12 % :
42 \catcode64 11 % @
43 \catcode123 1 % {
44 \catcode125 2 % }
45 \expandafter\ifx\csname ProvidesPackage\endcsname\relax
46 \def\x#1#2#3[#4]{\endgroup
47 \immediate\write-1{Package: #3 #4}%
48 \xdef#1[#4}%
49 }%
50 \else
51 \def\x#1#2[#3]{\endgroup
```

```

52      #2[#{#3}]%
53      \ifx#1\@undefined
54      \xdef#1{#3}%
55      \fi
56      \ifx#1\relax
57      \xdef#1{#3}%
58      \fi
59    }%
60  \fi
61 \expandafter\x\csname ver@catchfile.sty\endcsname
62 \ProvidesPackage{catchfile}%
63 [2007/11/11 v1.2 Catches the contents of a file (HO)]

```

## 2.2 Catcodes

```

64 \begingroup
65   \catcode123 1 % {
66   \catcode125 2 % }
67   \def\x{\endgroup
68     \expandafter\edef\csname CatchFile@AtEnd\endcsname{%
69       \catcode35 \the\catcode35\relax
70       \catcode64 \the\catcode64\relax
71       \catcode123 \the\catcode123\relax
72       \catcode125 \the\catcode125\relax
73     }%
74   }%
75   \x
76   \catcode35 6 % #
77   \catcode64 11 % @
78   \catcode123 1 % {
79   \catcode125 2 % }
80   \def\TMP@EnsureCode#1#2{%
81     \edef\CatchFile@AtEnd{%
82       \CatchFile@AtEnd
83       \catcode#1 \the\catcode#1\relax
84     }%
85     \catcode#1 #2\relax
86   }
87   \TMP@EnsureCode{39}{12}% '
88   \TMP@EnsureCode{44}{12}% ,
89   \TMP@EnsureCode{45}{12}% -
90   \TMP@EnsureCode{46}{12}% .
91   \TMP@EnsureCode{47}{12}% /
92   \TMP@EnsureCode{61}{12}% =
93   \TMP@EnsureCode{96}{12}% `

```

## 2.3 Preparations

```

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

```

Check for  $\varepsilon$ -TeX's `\everyeof`.

```

100 \begingroup
101   \escapechar=92\relax
102   \edef\TestString{\string\everyeof}%
103   \edef\TestMeaning{\meaning\everyeof}%
104   \ifx\TestString\TestMeaning
105   \else
106     \@PackageErrorNoLine{catchfile}{%
107       Cannot find e-TeX's \string\everyeof,\MessageBreak

```

```

108     package loading is aborted%
109 } \@ehd
110 \endgroup
111 \CatchFile@AtEnd
112 \expandafter\endinput
113 \fi
114 \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<sup>A</sup>T<sub>E</sub>X's `\input` is not expandable, but it stores the primitive in `\@@input`. The third possibility is `\pdfprimitive`, introduced in pdf<sub>T</sub>E<sub>X</sub> 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`).

```

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

```

## 2.5 Input file check

`\CatchFile@CheckFileExists`

```

138 \begingroup\expandafter\expandafter\expandafter\endgroup
139 \expandafter\ifx\csname IfFileExists\endcsname\relax
140   \input pdftexcmds.sty\relax
141 \begingroup\expandafter\expandafter\expandafter\endgroup
142 \expandafter\ifx\csname pdf@filesize\endcsname\relax
143   \def\CatchFile@CheckFileExists#1{%
144     \expandafter\ifx\csname @inputcheck\endcsname\relax
145       \csname newread\endcsname\@inputcheck
146     \fi
147     \openin\@inputcheck#1\relax
148     \ifeof\@inputcheck
149       \let\CatchFile@File\relax
150     \else
151       \closein\@inputcheck
152       \def\CatchFile@File{#1}%
153     \fi
154   }%
155 \else
156   \def\CatchFile@CheckFileExists#1{%

```

```

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

```

\CatchFileNotFound

```

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

```

## 2.6 Catch file contents

\CatchFileEdef

```

188 \long\def\CatchFileEdef#1#2#3{%
189 \CatchFile@CheckFileExists{#2}%
190 \ifx\CatchFile@File\relax
191 \CatchFile@NotFound{#1}{#2}%
192 \else
193 \begingroup
194 \everyeof{\noexpand}%
195 #3%
196 \xdef\CatchFile@Contents{\CatchFile@Input\CatchFile@File\space}%
197 \endgroup
198 \let#1\CatchFile@Contents
199 \fi
200 }

```

\CatchFileDef

```

201 \long\def\CatchFileDef#1#2#3{%
202 \CatchFile@CheckFileExists{#2}%
203 \ifx\CatchFile@File\relax
204 \CatchFile@NotFound{#1}{#2}%
205 \else
206 \begingroup
207 \everyeof\expandafter{%
208 \CatchFile@EOF
209 \noexpand
210 }%

```

```

211 \expandafter\long\expandafter\def\expandafter\CatchFile@Do
212 \expandafter##\expandafter1\CatchFile@EOF{%
213 \endgroup
214 \def#1{##1}%
215 }%
216 #3%
217 \expandafter\expandafter\expandafter\CatchFile@Do
218 \CatchFile@Input\CatchFile@File\relax
219 \fi
220 }

```

\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.

```

221 \begingroup
222 \lccode65=64 % lowercase('A') = '@'
223 \lccode66=64 % lowercase('B') = '@'
224 \catcode65=8 % catcode('A') = subscript
225 \catcode66=3 % catcode('B') = math shift
226 \lowercase{\endgroup
227 \def\CatchFile@EOF{AB}%
228 }

229 \CatchFile@AtEnd
230 </package>

```

## 3 Test

### 3.1 Catcode checks for loading

```

231 <*test1>
232 \catcode'\{=1 %
233 \catcode'\}=2 %
234 \catcode'\#=6 %
235 \catcode'\@=11 %
236 \expandafter\ifx\csname count@\endcsname\relax
237 \countdef\count@=255 %
238 \fi
239 \expandafter\ifx\csname @gobble\endcsname\relax
240 \long\def@gobble#1{}%
241 \fi
242 \expandafter\ifx\csname @firstofone\endcsname\relax
243 \long\def@firstofone#1{#1}%
244 \fi
245 \expandafter\ifx\csname loop\endcsname\relax
246 \expandafter@firstofone
247 \else
248 \expandafter@gobble
249 \fi
250 {%
251 \def\loop#1\repeat{%
252 \def\body{#1}%
253 \iterate
254 }%
255 \def\iterate{%
256 \body
257 \let\next\iterate
258 \else
259 \let\next\relax
260 \fi
261 \next

```

```

262 }%
263 \let\repeat=\fi
264 }%
265 \def\RestoreCatcodes{}
266 \count@=0 %
267 \loop
268 \edef\RestoreCatcodes{%
269 \RestoreCatcodes
270 \catcode\the\count@=\the\catcode\count@\relax
271 }%
272 \ifnum\count@<255 %
273 \advance\count@ 1 %
274 \repeat
275
276 \def\RangeCatcodeInvalid#1#2{%
277 \count@=#1\relax
278 \loop
279 \catcode\count@=15 %
280 \ifnum\count@<#2\relax
281 \advance\count@ 1 %
282 \repeat
283 }
284 \expandafter\ifx\csname LoadCommand\endcsname\relax
285 \def\LoadCommand{\input catchfile.sty\relax}%
286 \fi
287 \def\Test{%
288 \RangeCatcodeInvalid{0}{47}%
289 \RangeCatcodeInvalid{58}{64}%
290 \RangeCatcodeInvalid{91}{96}%
291 \RangeCatcodeInvalid{123}{255}%
292 \catcode'\@=12 %
293 \catcode'\=0 %
294 \catcode'\{=1 %
295 \catcode'\}=2 %
296 \catcode'\#=6 %
297 \catcode'\[=12 %
298 \catcode'\]=12 %
299 \catcode'\%=14 %
300 \catcode'\ =10 %
301 \catcode13=5 %
302 \LoadCommand
303 \RestoreCatcodes
304 }
305 \Test
306 \csname @@end\endcsname
307 \end
308 </test1>

```

### 3.2 L<sup>A</sup>T<sub>E</sub>X

```

309 <*test2>
310 \NeedsTeXFormat{LaTeX2e}
311 \nofiles
312 \listfiles
313 \documentclass{minimal}
314 \usepackage{catchfile}[2007/11/11]
315 \makeatletter
316 \def\mysetup{%
317 \let\do\@makeother
318 \dospecials
319 }
320 \makeatother
321 \begin{document}

```

```

322
323 \CatchFileDef\contents{catchfile.sty}\mysetup
324 \typeout{\meaning\contents}
325
326 \CatchFileEdef\contents{catchfile.sty}{%
327   \mysetup
328   \def\par{^^J}%
329   \obeylines
330 }
331 \typeout{\contents}
332 \end{document}
333 </test2>

```

### 3.3 plain-TeX

```

334 <*test3>
335 \def\msg#\{\immediate\write16}
336 \newlinechar=10 %
337 \input catchfile.sty\relax
338
339 \def\mysetup{%
340   \def\do##1{%
341     \catcode'\##1=12\relax
342   }%
343   \dospecials
344 }
345
346 \CatchFileDef\contents{catchfile.sty}\mysetup
347 \msg{\meaning\contents}
348
349 \CatchFileEdef\contents{catchfile.sty}{%
350   \mysetup
351   \def\par{^^J}%
352   \obeylines
353 }
354 \msg{\contents}
355
356 \csname @@end\endcsname
357 \end
358 </test3>

```

## 4 Installation

### 4.1 Download

**Package.** This package is available on CTAN<sup>1</sup>:

[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 TeX Files” ([CTAN:tds/tds.pdf](#)). Directories with `texmf` in their name are usually organized this way.

---

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



## 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-`TEX`:

```
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):

<code>catchfile.sty</code>	→ <code>tex/generic/oberdiek/catchfile.sty</code>
<code>catchfile.pdf</code>	→ <code>doc/latex/oberdiek/catchfile.pdf</code>
<code>test/catchfile-test1.tex</code>	→ <code>doc/latex/oberdiek/test/catchfile-test1.tex</code>
<code>test/catchfile-test2.tex</code>	→ <code>doc/latex/oberdiek/test/catchfile-test2.tex</code>
<code>test/catchfile-test3.tex</code>	→ <code>doc/latex/oberdiek/test/catchfile-test3.tex</code>
<code>catchfile.dtx</code>	→ <code>source/latex/oberdiek/catchfile.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 `TEX` distribution (te`TEX`, mi`TEX`, ...) relies on file name databases, you must refresh these. For example, te`TEX` 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 catchfile.pdf unpack_files output .
```

**Unpacking with L<sup>A</sup>T<sub>E</sub>X.** The `.dtx` chooses its action depending on the format:

**plain-`TEX`:** Run `docstrip` and extract the files.

**L<sup>A</sup>T<sub>E</sub>X:** Generate the documentation.

If you insist on using L<sup>A</sup>T<sub>E</sub>X for `docstrip` (really, `docstrip` does not need L<sup>A</sup>T<sub>E</sub>X), 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 pdfL<sup>A</sup>T<sub>E</sub>X:

```
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 L<sup>A</sup>T<sub>E</sub>X support.

## 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	A
\# ..... 234, 296	\advance ..... 273, 281
\% ..... 299	
\@ ..... 235, 292	<b>B</b>
\@@input ..... 128	\begin ..... 321
\@PackageError ..... 184	\body ..... 252, 256
\@PackageErrorNoLine ..... 106, 130	
\@addtofilelist ..... 172	<b>C</b>
\@ehc ..... 186	\CatchFile@AtEnd 81, 82, 111, 135, 229
\@ehd ..... 109, 133	\CatchFile@CheckFileExists ..... ..... 138, 189, 202
\@filef@und ..... 168	\CatchFile@Contents ..... 196, 198
\@firstofone ..... 243, 246	\CatchFile@DefFile ..... 168, 178
\@gobble ..... 240, 248	\CatchFile@Do ..... 211, 217
\@inputcheck ..... 145, 147, 148, 151	\CatchFile@EOF ..... 208, 212, <u>221</u>
\@makeother ..... 317	\CatchFile@File 149, 152, 159, 161, 172, 175, 179, 190, 196, 203, 218
\@nil ..... 168, 178	\CatchFile@Input ..... <u>115</u> , 196, 218
\@undefined ..... 53	\CatchFile@NotFound ... 182, 191, 204
\[ ..... 297	\CatchFile@Primitive ..... 121, 122
\ \ ..... 293	\CatchFileDef ..... 1, <u>201</u> , 323, 346
\{ ..... 232, 294	\CatchFileEdef ..... <u>188</u> , 326, 349
\} ..... 233, 295	\CatchFileNotFound ..... <u>182</u>
\] ..... 298	\catcode ..... 3, 4, 5, 6, 7, 18, 19, 20, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 65, 66,
\_ ..... 300	

69, 70, 71, 72, 76, 77, 78, 79, 83, 85, 224, 225, 232, 233, 234, 235, 270, 279, 292, 293, 294, 295, 296, 297, 298, 299, 300, 301, 341	\makeatother . . . . . 320
\Check . . . . . 116, 127, 128, 129	\meaning . . . . . 103, 118, 324, 347
\closein . . . . . 151	\MessageBreak . . . . . 107, 131
\contents . . . . . 323, 324, 326, 331, 346, 347, 349, 354	\msg . . . . . 335, 347, 354
\count@ . . . . . 237, 266, 270, 272, 273, 277, 279, 280, 281	\mysetup . . 316, 323, 327, 339, 346, 350
\countdef . . . . . 237	
\csname . . . . . 8, 21, 45, 61, 68, 95, 134, 139, 142, 144, 145, 170, 236, 239, 242, 245, 284, 306, 356	
	<b>N</b>
<b>D</b>	\NeedsTeXFormat . . . . . 310
\do . . . . . 317, 340	\newlinechar . . . . . 336
\documentclass . . . . . 313	\next . . . . . 257, 259, 261
\dospecials . . . . . 318, 343	\nofiles . . . . . 311
	<b>O</b>
<b>E</b>	\obeylines . . . . . 329, 352
\empty . . . . . 12	\openin . . . . . 147
\end . . . . . 307, 332, 357	
\endcsname . . . . . 8, 21, 45, 61, 68, 95, 134, 139, 142, 144, 145, 170, 236, 239, 242, 245, 284, 306, 356	<b>P</b>
\endinput . . . . . 30, 112, 136	\PackageInfo . . . . . 26
\escapechar . . . . . 101	\par . . . . . 328, 351
\everyeof . . . . . 102, 103, 107, 194, 207	\pdf@filesize . . . . . 158
	\pdfprimitive . . . . . 129
<b>I</b>	\ProvidesPackage . . . . . 62
\ifcase . . . . . 9	
\ifeof . . . . . 148	<b>R</b>
\IfFileExists . . . . . 167	\RangeCatcodeInvalid . . . . . . . . . . 276, 288, 289, 290, 291
\ifnum . . . . . 272, 280	\repeat . . . . . 251, 263, 274, 282
\ifx . . 10, 12, 21, 45, 53, 56, 95, 104, 119, 139, 142, 144, 157, 170, 190, 203, 236, 239, 242, 245, 284	\RequirePackage . . . . . 98
\immediate . . . . . 23, 47, 335	\RestoreCatcodes . . 265, 268, 269, 303
\input . . . . . 96, 127, 128, 129, 131, 140, 285, 337	
\iterate . . . . . 253, 255, 257	<b>S</b>
	\space . . . . . 196
<b>L</b>	
\lccode . . . . . 222, 223	<b>T</b>
\listfiles . . . . . 312	\Test . . . . . 287, 305
\LoadCommand . . . . . 285, 302	\TestMeaning . . . . . 103, 104, 118, 119
\loop . . . . . 251, 267, 278	\TestString . . . . . 102, 104, 117, 119
\lowercase . . . . . 226	\the . . . . . 69, 70, 71, 72, 83, 270
	\TMP@EnsureCode . . . . . . . . . . 80, 87, 88, 89, 90, 91, 92, 93
<b>M</b>	\typeout . . . . . 324, 331
\makeatletter . . . . . 315	
	<b>U</b>
	\usepackage . . . . . 314
	<b>W</b>
	\write . . . . . 23, 47, 335
	<b>X</b>
	\x 8, 10, 12, 22, 26, 28, 46, 51, 61, 67, 75