

# The `catchfile` package

Heiko Oberdiek\*

<heiko.oberdiek at gmail.com>

2016/05/16 v1.7

## Abstract

This package catches the contents of a file and puts it in a macro. It requires  $\epsilon$ -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>2</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 . . . . .	4
2.4	Looking for primitive <code>\input</code> . . . . .	4
2.5	Input file check . . . . .	5
2.6	Catch file contents . . . . .	6
<b>3</b>	<b>Test</b>	<b>7</b>
3.1	Catcode checks for loading . . . . .	7
3.2	L <sup>A</sup> T <sub>E</sub> X . . . . .	9
3.3	plain T <sub>E</sub> X . . . . .	10
<b>4</b>	<b>Installation</b>	<b>12</b>
4.1	Download . . . . .	12
4.2	Bundle installation . . . . .	12
4.3	Package installation . . . . .	12
4.4	Refresh file name databases . . . . .	12
4.5	Some details for the interested . . . . .	13
<b>5</b>	<b>Catalogue</b>	<b>13</b>
<b>6</b>	<b>History</b>	<b>14</b>
	[2007/05/30 v1.0] . . . . .	14
	[2007/09/09 v1.1] . . . . .	14
	[2007/11/11 v1.2] . . . . .	14
	[2010/03/01 v1.3] . . . . .	14
	[2010/04/08 v1.4] . . . . .	14
	[2010/04/28 v1.5] . . . . .	14
	[2011/03/01 v1.6] . . . . .	14
	[2016/05/16 v1.7] . . . . .	14

---

\*Please report any issues at <https://github.com/ho-tex/oberdiek/issues>

# 1 Documentation

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

```
\CatchFileDef {\cmd} {\file name} {\setup}  
\CatchFileEdef {\cmd} {\file name} {\setup}
```

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<sup>A</sup>T<sub>E</sub>X.

```
2 \begingroup\catcode61\catcode48\catcode32=10\relax%  
3 \catcode13=5 % ^^M  
4 \endlinechar=13 %  
5 \catcode35=6 % #  
6 \catcode39=12 % '  
7 \catcode44=12 % ,  
8 \catcode45=12 % -  
9 \catcode46=12 % .  
10 \catcode58=12 % :  
11 \catcode64=11 % @  
12 \catcode123=1 % {  
13 \catcode125=2 % }  
14 \expandafter\let\expandafter\x\csname ver@catchfile.sty\endcsname  
15 \ifx\x\relax % plain-TeX, first loading  
16 \else  
17 \def\empty{}%  
18 \ifx\x\empty % LaTeX, first loading,  
19 % variable is initialized, but \ProvidesPackage not yet seen  
20 \else  
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 \aftergroup\endinput  
30 \fi  
31 \fi  
32 \endgroup%
```

Package identification:

```
33 \begingroup\catcode61\catcode48\catcode32=10\relax%  
34 \catcode13=5 % ^^M  
35 \endlinechar=13 %  
36 \catcode35=6 % #  
37 \catcode39=12 % '  
38 \catcode40=12 % (  
39 \catcode41=12 % )  
40 \catcode44=12 % ,
```

```

41 \catcode45=12 % -
42 \catcode46=12 % .
43 \catcode47=12 % /
44 \catcode58=12 % :
45 \catcode64=11 % @
46 \catcode91=12 % [
47 \catcode93=12 % ]
48 \catcode123=1 % {
49 \catcode125=2 % }
50 \expandafter\ifx\csname ProvidesPackage\endcsname\relax
51 \def\x#1#2#3[#4]{\endgroup
52 \immediate\write-1{Package: #3 #4}%
53 \xdef#1{#4}%
54 }%
55 \else
56 \def\x#1#2[#3]{\endgroup
57 #2[#{#3}]%
58 \ifx#1\@undefined
59 \xdef#1{#3}%
60 \fi
61 \ifx#1\relax
62 \xdef#1{#3}%
63 \fi
64 }%
65 \fi
66 \expandafter\x\csname ver@catchfile.sty\endcsname
67 \ProvidesPackage{catchfile}%
68 [2016/05/16 v1.7 Catch the contents of a file (HO)]%

```

## 2.2 Catcodes

```

69 \begingroup\catcode61\catcode48\catcode32=10\relax%
70 \catcode13=5 % ^^M
71 \endlinechar=13 %
72 \catcode123=1 % {
73 \catcode125=2 % }
74 \catcode64=11 % @
75 \def\x{\endgroup
76 \expandafter\edef\csname CatchFile@AtEnd\endcsname{%
77 \endlinechar=\the\endlinechar\relax
78 \catcode13=\the\catcode13\relax
79 \catcode32=\the\catcode32\relax
80 \catcode35=\the\catcode35\relax
81 \catcode61=\the\catcode61\relax
82 \catcode64=\the\catcode64\relax
83 \catcode123=\the\catcode123\relax
84 \catcode125=\the\catcode125\relax
85 }%
86 }%
87 \x\catcode61\catcode48\catcode32=10\relax%
88 \catcode13=5 % ^^M
89 \endlinechar=13 %
90 \catcode35=6 % #
91 \catcode64=11 % @
92 \catcode123=1 % {
93 \catcode125=2 % }
94 \def\TMP@EnsureCode#1#2{%
95 \edef\CatchFile@AtEnd{%
96 \CatchFile@AtEnd
97 \catcode#1=\the\catcode#1\relax
98 }%
99 \catcode#1=#2\relax

```

```

100 }
101 \TMP@EnsureCode{39}{12}% '
102 \TMP@EnsureCode{44}{12}% ,
103 \TMP@EnsureCode{45}{12}% -
104 \TMP@EnsureCode{46}{12}% .
105 \TMP@EnsureCode{47}{12}% /
106 \TMP@EnsureCode{91}{12}% [
107 \TMP@EnsureCode{93}{12}% ]
108 \TMP@EnsureCode{96}{12}% `
109 \edef\CatchFile@AtEnd{\CatchFile@AtEnd\noexpand\endinput}

```

## 2.3 Preparations

```

110 \begingroup\expandafter\expandafter\expandafter\endgroup
111 \expandafter\ifx\csname RequirePackage\endcsname\relax
112   \input infwarerr.sty\relax
113   \input ltxcmds.sty\relax
114 \else
115   \RequirePackage{infwarerr}[2007/09/09]%
116   \RequirePackage{ltxcmds}[2010/03/09]%
117 \fi

```

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

```

118 \begingroup
119   \escapechar=92\relax
120   \edef\TestString{\string\everyeof}%
121   \edef\TestMeaning{\meaning\everyeof}%
122   \ifx\TestString\TestMeaning
123   \else
124     \@PackageError{catchfile}{%
125       Cannot find  $\varepsilon$ -TeX's \string\everyeof,\MessageBreak
126       package loading is aborted%
127     }\@ehd
128   \endgroup
129   \expandafter\CatchFile@AtEnd
130 \fi%
131 \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`).

```

132 \begingroup
133   \def\Check#1#2#3#4\endgroup{%
134     \edef\TestString{\string#1}%
135     \edef\TestMeaning{\meaning#2}%
136     \ifx\TestString\TestMeaning
137     \endgroup
138     \let\CatchFile@Primitive#2%
139     \def\CatchFile@Input{\CatchFile@Primitive#3}%
140   \else
141     #4\endgroup
142   \fi
143 }%
144 \Check\input\input{%
145 \Check\input\@@input{%
146 \Check\pdfprimitive\pdfprimitive\input
147 \@PackageError{%
148   Cannot find primitive \string\input,\MessageBreak
149   package loading is aborted%

```

```

150 } \@ehd
151 \csname endgroup\endcsname
152 \CatchFile@AtEnd%
153 \endgroup

```

## 2.5 Input file check

\CatchFile@CheckFileExists

```

154 \begingroup\expandafter\expandafter\expandafter\endgroup
155 \expandafter\ifx\csname IfFileExists\endcsname\relax
156 \input pdftexcmds.sty\relax
157 \begingroup\expandafter\expandafter\expandafter\endgroup
158 \expandafter\ifx\csname pdf@filesize\endcsname\relax
159 \def\CatchFile@CheckFileExists#1{%
160 \expandafter\ifx\csname @inputcheck\endcsname\relax
161 \csname newread\endcsname\@inputcheck
162 \fi
163 \openin\@inputcheck#1\relax
164 \expandafter\closein\expandafter\@inputcheck
165 \ifeof\@inputcheck
166 \let\CatchFile@File\relax
167 \else
168 \def\CatchFile@File{#1}%
169 \fi
170 }%
171 \else
172 \def\CatchFile@CheckFileExists#1{%
173 \expandafter\expandafter\expandafter\ifx
174 \expandafter\expandafter\expandafter\relax\pdf@filesize{#1}\relax
175 \let\CatchFile@File\relax
176 \else
177 \def\CatchFile@File{#1}%
178 \fi
179 }%
180 \fi
181 \else
182 \def\CatchFile@CheckFileExists#1{%
183 \IfFileExists{#1}{%
184 \expandafter\CatchFile@DefFile\@filef@und\@nil
185 \begingroup\expandafter\expandafter\expandafter\endgroup
186 \expandafter\ifx\csname @addtofilelist\endcsname\relax
187 \else
188 \@addtofilelist\CatchFile@File
189 \fi
190 }{%
191 \let\CatchFile@File\relax
192 }%
193 }%
194 \def\CatchFile@DefFile#1 \@nil{%
195 \def\CatchFile@File{#1}%
196 }%
197 \fi

```

\CatchFile@NotFound

```

198 \def\CatchFile@NotFound#1#2{%
199 \def#1{%
200 \@PackageError{catchfile}{%
201 File `#2' not found%
202 } \@ehc
203 }

```

## 2.6 Catch file contents

```

204 \ltx@ifundefined{RequirePackage}{%
205   \input etexcmds.sty\relax
206 }{%
207   \RequirePackage{etexcmds}[2010/01/28]%
208 }

\CatchFileEdef

209 \long\def\CatchFileEdef#1#2#3{%
210   \CatchFile@CheckFileExists{#2}%
211   \ifx\CatchFile@File\relax
212     \CatchFile@NotFound{#1}{#2}%
213   \else
214     \begingroup
215       \everyeof{\noexpand}%
216       #3%
217       \xdef\CatchFile@Contents{\CatchFile@Input\CatchFile@File\space}%
218     \endgroup
219     \let#1\CatchFile@Contents
220   \fi
221 }

\CatchFileDef

222 \long\def\CatchFileDef#1#2#3{%
223   \CatchFile@CheckFileExists{#2}%
224   \ifx\CatchFile@File\relax
225     \CatchFile@NotFound{#1}{#2}%
226   \else
227     \begingroup
228       \everyeof\expandafter{%
229         \CatchFile@EOF
230         \expandafter\CatchFile@Finish
231         \noexpand
232       }%
233     \ltx@ifundefined{etex@unexpanded}{%
234       \expandafter\long\expandafter\def\expandafter\CatchFile@Do
235         \expandafter##\expandafter1\CatchFile@EOF{%
236           \toks\ltx@zero{##1}%
237           \xdef\CatchFile@gtemp{\the\toks\ltx@zero}%
238           \def\CatchFile@Finish{%
239             \endgroup
240             \let#1\CatchFile@gtemp
241             \global\let\CatchFile@gtemp\ltx@undefined
242           }%
243         }%
244       }{%
245         \expandafter\long\expandafter\def\expandafter\CatchFile@Do
246           \expandafter##\expandafter1\CatchFile@EOF{%
247             \edef\CatchFile@Finish{%
248               \endgroup
249               \etex@unexpanded{%
250                 \edef#1{\etex@unexpanded{##1}}%
251               }%
252             }%
253           }%
254         }%
255       }%
256       \expandafter\expandafter\expandafter\CatchFile@Do
257       \CatchFile@Input\CatchFile@File\relax
258     \fi
259 }

```

`\relax` after `#3` was added to make it more robust in case the user uses something like

```
\CatchFileDef{\content}{\jobname.tt}{\endlinechar=-1}
```

that expands the following `\expandafter` after `#3` prematurely (contribution of Martin Scharrer).

`\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.

```
260 \begingroup
261 \lccode65=64 % lowercase('A') = '@'
262 \lccode66=64 % lowercase('B') = '@'
263 \catcode65=8 % catcode('A') = subscript
264 \catcode66=3 % catcode('B') = math shift
265 \lowercase{\endgroup
266 \def\CatchFile@EOF{AB}%
267 }

268 \CatchFile@AtEnd%
269 \</package>
```

## 3 Test

### 3.1 Catcode checks for loading

```
270 \<test1>

271 \catcode`\{=1 %
272 \catcode`\}=2 %
273 \catcode`\#=6 %
274 \catcode`\@=11 %
275 \expandafter\ifx\csname count@\endcsname\relax
276 \countdef\count@=255 %
277 \fi
278 \expandafter\ifx\csname @gobble\endcsname\relax
279 \long\def\@gobble#1{}%
280 \fi
281 \expandafter\ifx\csname @firstofone\endcsname\relax
282 \long\def\@firstofone#1{#1}%
283 \fi
284 \expandafter\ifx\csname loop\endcsname\relax
285 \expandafter\@firstofone
286 \else
287 \expandafter\@gobble
288 \fi
289 {%
290 \def\loop#1\repeat{%
291 \def\body{#1}%
292 \iterate
293 }%
294 \def\iterate{%
295 \body
296 \let\next\iterate
297 \else
298 \let\next\relax
299 \fi
300 \next
301 }%
302 \let\repeat=\fi
303 }%
304 \def\RestoreCatcodes{}
```

```

305 \count@=0 %
306 \loop
307   \edef\RestoreCatcodes{%
308     \RestoreCatcodes
309     \catcode\the\count@=\the\catcode\count@\relax
310   }%
311 \ifnum\count@<255 %
312   \advance\count@ 1 %
313 \repeat
314
315 \def\RangeCatcodeInvalid#1#2{%
316   \count@=#1\relax
317   \loop
318     \catcode\count@=15 %
319     \ifnum\count@<#2\relax
320       \advance\count@ 1 %
321     \repeat
322 }
323 \def\RangeCatcodeCheck#1#2#3{%
324   \count@=#1\relax
325   \loop
326     \ifnum#3=\catcode\count@
327     \else
328       \errmessage{%
329         Character \the\count@\space
330         with wrong catcode \the\catcode\count@\space
331         instead of \number#3%
332       }%
333     \fi
334     \ifnum\count@<#2\relax
335       \advance\count@ 1 %
336     \repeat
337 }
338 \def\space{ }
339 \expandafter\ifx\csname LoadCommand\endcsname\relax
340 \def\LoadCommand{\input catchfile.sty\relax}%
341 \fi
342 \def\Test{%
343   \RangeCatcodeInvalid{0}{47}%
344   \RangeCatcodeInvalid{58}{64}%
345   \RangeCatcodeInvalid{91}{96}%
346   \RangeCatcodeInvalid{123}{255}%
347   \catcode`\@=12 %
348   \catcode`\|=0 %
349   \catcode`\%=14 %
350   \LoadCommand
351   \RangeCatcodeCheck{0}{36}{15}%
352   \RangeCatcodeCheck{37}{37}{14}%
353   \RangeCatcodeCheck{38}{47}{15}%
354   \RangeCatcodeCheck{48}{57}{12}%
355   \RangeCatcodeCheck{58}{63}{15}%
356   \RangeCatcodeCheck{64}{64}{12}%
357   \RangeCatcodeCheck{65}{90}{11}%
358   \RangeCatcodeCheck{91}{91}{15}%
359   \RangeCatcodeCheck{92}{92}{0}%
360   \RangeCatcodeCheck{93}{96}{15}%
361   \RangeCatcodeCheck{97}{122}{11}%
362   \RangeCatcodeCheck{123}{255}{15}%
363   \RestoreCatcodes
364 }
365 \Test
366 \csname @@end\endcsname

```



```

367 \end
368 </test1>
3.2 LATEX
369 (*test2)
370 \NeedsTeXFormat{LaTeX2e}
371 \nofiles
372 \listfiles
373 \tracingnesting=2 %
374 \documentclass{minimal}
375 \usepackage{catchfile}[2016/05/16]
376 \makeatletter
377 \def\mysetup{%
378   \let\do\@makeoother
379   \dospecials
380 }
381 \def\CheckContents{%
382   \begingroup
383   \expandafter\ifx\cename contents\endcename\relax
384     \PackageError{TEST}{\string\contents\space is not defined}\@ehc
385   \fi
386   \endgroup
387 }
388 \def\StartDisableUnexpanded{%
389   \begingroup
390   \let\etex@unexpanded\@undefined
391 }
392 \def\StopDisableUnexpanded{%
393   \endgroup
394 }
395 \def\CheckCleanup{%
396   \begingroup
397   \edef\x{\the\toks0}%
398   \def\y{ABC}%
399   \ifx\x\y
400   \else
401     \PackageError{TEST}{\string\toks0 has changed}\@ehc
402   \fi
403   \ifx\CatchFile@gtemp\@undefined
404   \else
405     \PackageError{TEST}{Left over \string\CatchFile@gtemp}\@ehc
406   \fi
407   \endgroup
408 }
409 \def\Check{%
410   \CheckContents
411   \CheckCleanup
412 }
413 \makeatother
414 \begin{document}
415 \toks0{ABC}
416
417 \CatchFileDef\contents{catchfile.sty}\mysetup
418 \typeout{\meaning\contents}
419 \Check
420 \typeout{*****}%
421 \CatchFileDef\contents{catchfile.sty}{ }
422 \typeout{\meaning\contents}
423 \Check
424 \typeout{*****}%
425
426 \StartDisableUnexpanded

```

```

427 \CatchFileDef\contents{catchfile.sty}\mysetup
428 \typeout{\meaning\contents}
429 \Check
430 \typeout{*****}%
431 \CatchFileDef\contents{catchfile.sty}{%
432 \typeout{\meaning\contents}
433 \Check
434 \typeout{*****}%
435 \StopDisableUnexpanded
436
437 \CatchFileEdef\contents{catchfile.sty}{%
438 \mysetup
439 \def\par{^^J}%
440 \obeylines
441 }
442 \typeout{\contents}
443 \Check
444 \typeout{*****}%
445 \CatchFileEdef\contents{catchfile.sty}{%
446 \catcode`\:=12 %
447 \catcode`\#=12 %
448 \def\par{^^J}%
449 \obeylines
450 }
451 \typeout{\contents}
452 \Check
453 \typeout{*****}%
454 \end{document}
455 </test2>

```

### 3.3 plain T<sub>E</sub>X

```

456 (*test3)
457 \def\msg#{\immediate\write16}
458 \newlinechar=10 %
459 \tracingnesting=2 %
460 \input catchfile.sty\relax
461
462 \def\mysetup{%
463 \def\do##1{%
464 \catcode`##1=12\relax
465 }%
466 \dospecials
467 }
468 \def\CheckContents{%
469 \begingroup
470 \expandafter\ifx\csname contents\endcsname\relax
471 \def\space{ }%
472 \errmessage{\string\contents\space is not defined.}%
473 \fi
474 \endgroup
475 }
476 \catcode`\@=11 %
477 \def\CheckCleanup{%
478 \begingroup
479 \edef\x{\the\toks0}%
480 \def\y{ABC}%
481 \ifx\x\y
482 \else
483 \errmessage{\string\toks0 has changed}%
484 \fi
485 \ifx\CatchFile@gtemp\@undefined
486 \else

```

```

487 \errmessage{Left over \string\CatchFile@temp}%
488 \fi
489 \endgroup
490 }
491 \def\Check{%
492 \CheckContents
493 \CheckCleanup
494 }
495 \def\StartDisableUnexpanded{%
496 \begingroup
497 \let\etex@unexpanded\@undefined
498 }
499 \def\StopDisableUnexpanded{%
500 \endgroup
501 }
502 \catcode`\@=12 %
503
504 \toks0{ABC}
505
506 \CatchFileDef\contents{catchfile.sty}\mysetup
507 \msg{\meaning\contents}
508 \Check
509 \msg{*****}%
510 \CatchFileDef\contents{catchfile.sty}{ }
511 \msg{\meaning\contents}
512 \Check
513 \msg{*****}%
514
515 \StartDisableUnexpanded
516 \CatchFileDef\contents{catchfile.sty}\mysetup
517 \msg{\meaning\contents}
518 \Check
519 \msg{*****}%
520 \CatchFileDef\contents{catchfile.sty}{ }
521 \msg{\meaning\contents}
522 \Check
523 \msg{*****}%
524 \StopDisableUnexpanded
525
526 \CatchFileEdef\contents{catchfile.sty}{%
527 \mysetup
528 \def\par{^^J}%
529 \obeylines
530 }
531 \msg{\contents}
532 \Check
533 \msg{*****}%
534 \CatchFileEdef\contents{catchfile.sty}{%
535 \catcode`\=12 %
536 \catcode`\#=12 %
537 \def\par{^^J}%
538 \obeylines
539 }
540 \msg{\contents}
541 \Check
542 \msg{*****}%
543
544 \csname @@end\endcsname
545 \end
546 /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 T<sub>E</sub>X 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<sub>E</sub>X:

```
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 T<sub>E</sub>X distribution (teT<sub>E</sub>X, miK<sub>T</sub>E<sub>X</sub>, ...) relies on file name databases, you must refresh these. For example, teT<sub>E</sub>X users run `texhash` or `mktextlsr`.

---

<sup>1</sup><http://ctan.org/pkg/catchfile>

## 4.5 Some details for the interested

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

**plain T<sub>E</sub>X:** 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 Catalogue

The following XML file can be used as source for the [T<sub>E</sub>X Catalogue](#). The elements `caption` and `description` are imported from the original XML file from the Catalogue. The name of the XML file in the Catalogue is `catchfile.xml`.

```
547 (*catalogue)
548 <?xml version='1.0' encoding='us-ascii'?>
549 <!DOCTYPE entry SYSTEM 'catalogue.dtd'>
550 <entry datestamp='$Date$' modifier='$Author$' id='catchfile'>
551   <name>catchfile</name>
552   <caption>Catch an external file into a macro.</caption>
553   <authorref id='auth:oberdiek'>/>
554   <copyright owner='Heiko Oberdiek' year='2007,2010,2011'>/>
555   <license type='lppl1.3'>/>
556   <version number='1.7'>/>
557   <description>
558     This package catches the contents of a file and puts it in a macro.
559     It requires <xref refid='etex'>e-TeX</xref>. Both
560     <xref refid='latex'>LaTeX</xref> and
561     <xref refid='plain'>plain TeX</xref> are supported.
562   <p/>
563   The package is part of the <xref refid='oberdiek'>oberdiek</xref>
564   bundle.
565 </description>
566 <documentation details='Package documentation'
567   href='ctan:/macros/latex/contrib/oberdiek/catchfile.pdf'>/>
568 <ctan file='true' path='/macros/latex/contrib/oberdiek/catchfile.dtx'>/>
569 <miktex location='oberdiek'>/>
570 <texlive location='oberdiek'>/>
571 <install path='/macros/latex/contrib/oberdiek/oberdiek.tds.zip'>/>
572 </entry>
573 </catalogue>
```

## 6 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 LuaTeX 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).

[2011/03/01 v1.6]

- `\relax` added after the setup argument of `\CatchFileDef` to prevent premature file reading (Martin Scharrer).

[2016/05/16 v1.7]

- Documentation updates.

## 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>\#</code> .....	273, 447, 536	<code>\@nil</code> ..... 184, 194
<code>\%</code> .....	349	<code>\@undefined</code> ..... 58, 390, 403, 485, 497
<code>\@</code> .....	274, 347, 476, 502	<code>\%</code> ..... 348, 446, 535
<code>\@@input</code> .....	145	<code>\{</code> ..... 271
<code>\@PackageError</code> .....	124, 147, 200	<code>\}</code> ..... 272
<code>\@addtofilelist</code> .....	188	
<code>\@ehc</code> .....	202, 384, 401, 405	<b>A</b>
<code>\@ehd</code> .....	127, 150	<code>\advance</code> ..... 312, 320, 335
<code>\@filef@und</code> .....	184	<code>\aftergroup</code> ..... 29
<code>\@firstofone</code> .....	282, 285	
<code>\@gobble</code> .....	279, 287	<b>B</b>
<code>\@inputcheck</code> .....	161, 163, 164, 165	<code>\begin</code> ..... 414
<code>\@makeother</code> .....	378	<code>\body</code> ..... 291, 295

<b>C</b>	
\CatchFile@AtEnd .....	95, 96, 109, 129, 152, 268
\CatchFile@CheckFileExists	154, 210, 223
\CatchFile@Contents .....	217, 219
\CatchFile@DefFile .....	184, 194
\CatchFile@Do .....	234, 245, 256
\CatchFile@EOF ...	229, 235, 246, 260
\CatchFile@File .	166, 168, 175, 177, 188, 191, 195, 211, 217, 224, 257
\CatchFile@Finish .....	230, 238, 247
\CatchFile@gttemp .....	237, 240, 241, 403, 405, 485, 487
\CatchFile@Input .....	132, 217, 257
\CatchFile@NotFound ...	198, 212, 225
\CatchFile@Primitive .....	138, 139
\CatchFileDef .....	2, 222, 417, 421, 427, 431, 506, 510, 516, 520
\CatchFileEdef .	209, 437, 445, 526, 534
\catcode .....	2, 3, 5, 6, 7, 8, 9, 10, 11, 12, 13, 33, 34, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 69, 70, 72, 73, 74, 78, 79, 80, 81, 82, 83, 84, 87, 88, 90, 91, 92, 93, 97, 99, 263, 264, 271, 272, 273, 274, 309, 318, 326, 330, 347, 348, 349, 446, 447, 464, 476, 502, 535, 536
\Check ....	133, 144, 145, 146, 409, 419, 423, 429, 433, 443, 452, 491, 508, 512, 518, 522, 532, 541
\CheckCleanup ....	395, 411, 477, 493
\CheckContents ....	381, 410, 468, 492
\closein .....	164
\contents .....	384, 417, 418, 421, 422, 427, 428, 431, 432, 437, 442, 445, 451, 472, 506, 507, 510, 511, 516, 517, 520, 521, 526, 531, 534, 540
\count@ .....	276, 305, 309, 311, 312, 316, 318, 319, 320, 324, 326, 329, 330, 334, 335
\countdef .....	276
\csname .....	14, 21, 50, 66, 76, 111, 151, 155, 158, 160, 161, 186, 275, 278, 281, 284, 339, 366, 383, 470, 544
<b>D</b>	
\do .....	378, 463
\documentclass .....	374
\dospecials .....	379, 466
<b>E</b>	
\empty .....	17, 18
\end .....	367, 454, 545
\endcsname .....	14, 21, 50, 66, 76, 111, 151, 155, 158, 160, 161, 186, 275, 278, 281, 284, 339, 366, 383, 470, 544
\endinput .....	29, 109
\endlinechar .....	4, 35, 71, 77, 89
\errmessage .....	328, 472, 483, 487
\escapechar .....	119
\etex@unexpanded ..	249, 250, 390, 497
\everyeof .....	120, 121, 125, 215, 228
<b>I</b>	
\ifeof .....	165
\IfFileExists .....	183
\ifnum .....	311, 319, 326, 334
\ifx .	15, 18, 21, 50, 58, 61, 111, 122, 136, 155, 158, 160, 173, 186, 211, 224, 275, 278, 281, 284, 339, 383, 399, 403, 470, 481, 485
\immediate .....	23, 52, 457
\input .....	112, 113, 144, 145, 146, 148, 156, 205, 340, 460
\iterate .....	292, 294, 296
<b>L</b>	
\lccode .....	261, 262
\listfiles .....	372
\LoadCommand .....	340, 350
\loop .....	290, 306, 317, 325
\lowercase .....	265
\ltx@ifUndefined .....	204
\ltx@ifundefined .....	233
\ltx@undefined .....	241
\ltx@zero .....	236, 237
<b>M</b>	
\makeatletter .....	376
\makeatother .....	413
\meaning .....	121, 135, 418, 422, 428, 432, 507, 511, 517, 521
\MessageBreak .....	125, 148
\msg ..	457, 507, 509, 511, 513, 517, 519, 521, 523, 531, 533, 540, 542
\mysetup .....	377, 417, 427, 438, 462, 506, 516, 527
<b>N</b>	
\NeedsTeXFormat .....	370
\newlinechar .....	458
\next .....	296, 298, 300
\nofiles .....	371
\number .....	331
<b>O</b>	
\obeylines .....	440, 449, 529, 538
\openin .....	163
<b>P</b>	
\PackageError .....	384, 401, 405
\PackageInfo .....	26
\par .....	439, 448, 528, 537
\pdf@filesize .....	174
\pdfprimitive .....	146
\ProvidesPackage .....	19, 67
<b>R</b>	
\RangeCatcodeCheck .....	323, 351, 352, 353, 354, 355, 356, 357, 358, 359, 360, 361, 362
\RangeCatcodeInvalid .....	315, 343, 344, 345, 346

