

# The catchfile package

Heiko Oberdiek  
<oberdiek@uni-freiburg.de>

2007/09/09 v1.1

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

## 1 Documentation

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

<code>\CatchFileDef &lt;cmd&gt; &lt;file name&gt; &lt;setup&gt;</code>
<code>\CatchFileEdef &lt;cmd&gt; &lt;file name&gt; &lt;setup&gt;</code>

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
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 \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 \endgroup
27 \expandafter\endinput
28 \fi
29 \endgroup
```

Package identification:

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

```

52 \expandafter\x\csname ver@catchfile.sty\endcsname
53 \ProvidesPackage{catchfile}%
54 [2007/09/09 v1.1 Catches the contents of a file (H0)]

```

## 2.2 Catcodes

```

55 \expandafter\edef\csname CatchFile@AtEnd\endcsname{%
56   \catcode64 \the\catcode64\relax
57 }
58 \catcode64 11 % @
59 \def\TMP@EnsureCode#1#2{%
60   \edef\CatchFile@AtEnd{%
61     \CatchFile@AtEnd
62     \catcode#1 \the\catcode#1\relax
63   }%
64   \catcode#1 #2\relax
65 }
66 \TMP@EnsureCode{39}{12}% '
67 \TMP@EnsureCode{44}{12}% ,
68 \TMP@EnsureCode{45}{12}% -
69 \TMP@EnsureCode{46}{12}% .
70 \TMP@EnsureCode{47}{12}% /
71 \TMP@EnsureCode{61}{12}% =
72 \TMP@EnsureCode{96}{12}% `

```

## 2.3 Preparations

```

73 \begingroup\expandafter\expandafter\expandafter\endgroup
74 \expandafter\ifx\csname RequirePackage\endcsname\relax
75   \input infwarerr.sty\relax
76 \else
77   \RequirePackage{infwarerr}[2007/09/09]%
78 \fi

```

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

```

79 \begingroup
80   \escapechar=92\relax
81   \edef\TestString{\string\everyeof}%
82   \edef\TestMeaning{\meaning\everyeof}%
83   \ifx\TestString\TestMeaning
84   \else
85     \@PackageErrorNoLine{catchfile}{%
86       Cannot find  $\varepsilon$ -TeX's \string\everyeof, \MessageBreak
87       package loading is aborted%
88     }\@ehd
89   \endgroup
90   \CatchFile@AtEnd
91   \expandafter\endinput
92 \fi
93 \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`).

```

94 \begingroup
95   \def\Check#1#2#3#4\endgroup{%
96     \edef\TestString{\string#1}%
97     \edef\TestMeaning{\meaning#2}%
98     \ifx\TestString\TestMeaning

```

```

99      \endgroup
100      \let\CatchFile@Primitive#2%
101      \def\CatchFile@Input{\CatchFile@Primitive#3}%
102      \else
103      #4\endgroup
104      \fi
105  }%
106  \Check\input\input{%
107  \Check\input\@input{%
108  \Check\pdfprimitive\pdfprimitive\input
109  \@PackageErrorNoLine{%
110    Cannot find primitive \string\input,\MessageBreak
111    package loading is aborted%
112  }\@ehd
113  \csname endgroup\endcsname
114  \CatchFile@AtEnd
115  \endinput
116 \endgroup

```

## 2.5 Input file check

\CatchFile@CheckFileExists

```

117 \begingroup\expandafter\expandafter\expandafter\endgroup
118 \expandafter\ifx\csname IfFileExists\endcsname\relax
119 \begingroup\expandafter\expandafter\expandafter\endgroup
120 \expandafter\ifx\csname pdffilesize\endcsname\relax
121   \def\CatchFile@CheckFileExists#1{%
122     \expandafter\ifx\csname @inputcheck\endcsname\relax
123       \csname newread\endcsname\@inputcheck
124       \fi
125       \openin\@inputcheck#1\relax
126       \ifeof\@inputcheck
127       \let\CatchFile@File\relax
128       \else
129       \closein\@inputcheck
130       \def\CatchFile@File{#1}%
131       \fi
132   }%
133 \else
134   \def\CatchFile@CheckFileExists#1{%
135     \expandafter\ifx\expandafter\\pdffilesize{#1}\\%
136       \let\CatchFile@File\relax
137       \else
138       \def\CatchFile@File{#1}%
139       \fi
140   }%
141 \fi
142 \else
143   \def\CatchFile@CheckFileExists#1{%
144     \IfFileExists{#1}{%
145       \expandafter\CatchFile@DefFile\@filef@und\@nil
146       \begingroup\expandafter\expandafter\expandafter\endgroup
147       \expandafter\ifx\csname @addtofilelist\endcsname\relax
148       \else
149       \@addtofilelist\CatchFile@File
150       \fi
151     }{%
152       \let\CatchFile@File\relax
153     }%
154   }%
155   \def\CatchFile@DefFile#1 \@nil{%
156     \def\CatchFile@File{#1}%

```

```

157 }%
158 \fi

```

`\CatchFileNotFound`

```

159 \def\CatchFile@NotFound#1#2{%
160   \def#1{}%
161   \@PackageError{catchfile}{%
162     File ‘#2’ not found%
163   }\@ehc
164 }

```

## 2.6 Catch file contents

`\CatchFileEdef`

```

165 \long\def\CatchFileEdef#1#2#3{%
166   \CatchFile@CheckFileExists{#2}%
167   \ifx\CatchFile@File\relax
168     \CatchFile@NotFound{#1}{#2}%
169   \else
170     \begingroup
171       \everyeof{\noexpand}%
172       #3%
173       \xdef\CatchFile@Contents{\CatchFile@Input\CatchFile@File\space}%
174     \endgroup
175     \let#1\CatchFile@Contents
176   \fi
177 }

```

`\CatchFileDef`

```

178 \long\def\CatchFileDef#1#2#3{%
179   \CatchFile@CheckFileExists{#2}%
180   \ifx\CatchFile@File\relax
181     \CatchFile@NotFound{#1}{#2}%
182   \else
183     \begingroup
184       \everyeof{\expandafter{%
185         \CatchFile@EOF
186       }\noexpand
187     }%
188     \expandafter\long\expandafter\def\expandafter\CatchFile@Do
189     \expandafter##\expandafter1\CatchFile@EOF{%
190       \endgroup
191       \def#1{##1}%
192     }%
193     #3%
194     \expandafter\expandafter\expandafter\CatchFile@Do
195     \CatchFile@Input\CatchFile@File\relax
196   \fi
197 }

```

`\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.

```

198 \begingroup
199   \lccode65=64 % lowercase('A') = '@'
200   \lccode66=64 % lowercase('B') = '@'
201   \catcode65=8 % catcode('A') = subscript
202   \catcode66=3 % catcode('B') = math shift
203 \lowercase{\endgroup
204   \def\CatchFile@EOF{AB}%
205 }

```

```

206 \CatchFile@AtEnd
207 \</package>

```

## 3 Test

### 3.1 Catcode checks for loading

```

208 <*test1>

209 \catcode'\@=11 %
210 \def\RestoreCatcodes{
211 \count@=0 %
212 \loop
213   \edef\RestoreCatcodes{%
214     \RestoreCatcodes
215     \catcode\the\count@=\the\catcode\count@\relax
216   }%
217 \ifnum\count@<255 %
218   \advance\count@\@ne
219 \repeat
220
221 \def\RangeCatcodeInvalid#1#2{%
222   \count@=#1\relax
223   \loop
224     \catcode\count@=15 %
225     \ifnum\count@<#2\relax
226       \advance\count@\@ne
227     \repeat
228 }
229 \def\Test{%
230   \RangeCatcodeInvalid{0}{47}%
231   \RangeCatcodeInvalid{58}{64}%
232   \RangeCatcodeInvalid{91}{96}%
233   \RangeCatcodeInvalid{123}{255}%
234   \catcode'\@=12 %
235   \catcode'\=0 %
236   \catcode'\{=1 %
237   \catcode'\}=2 %
238   \catcode'\#=6 %
239   \catcode'\[=12 %
240   \catcode'\]=12 %
241   \catcode'\%=14 %
242   \catcode'\ =10 %
243   \catcode13=5 %
244   \input catchfile.sty\relax
245   \RestoreCatcodes
246 }
247 \Test
248 \csname @@end\endcsname
249 \end

250 </test1>

```

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

```

251 <*test2>
252 \NeedsTeXFormat{LaTeX2e}
253 \nofiles
254 \listfiles
255 \documentclass{minimal}
256 \usepackage{catchfile}[2007/09/09]
257 \makeatletter
258 \def\mysetup{%
259   \let\do\@makeother

```

```

260 \dospecials
261 }
262 \makeatother
263 \begin{document}
264
265 \CatchFileDef\contents{catchfile.sty}\mysetup
266 \typeout{\meaning\contents}
267
268 \CatchFileEdef\contents{catchfile.sty}{%
269   \mysetup
270   \def\par{^^J}%
271   \obeylines
272 }
273 \typeout{\contents}
274 \end{document}
275 </test2>

```

### 3.3 plain-TeX

```

276 <*test3>
277 \def\msg#{\immediate\write16}
278 \newlinechar=10 %
279 \input catchfile.sty\relax
280
281 \def\mysetup{%
282   \def\do##1{%
283     \catcode'##1=12\relax
284   }%
285   \dospecials
286 }
287
288 \CatchFileDef\contents{catchfile.sty}\mysetup
289 \msg{\meaning\contents}
290
291 \CatchFileEdef\contents{catchfile.sty}{%
292   \mysetup
293   \def\par{^^J}%
294   \obeylines
295 }
296 \msg{\contents}
297
298 \csname @@end\endcsname
299 \end
300 </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:macros/latex/contrib/oberdiek/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>[ftp://ftp.ctan.org/tex-archive/](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):

```
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
```

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.



\PassOptionsToClass{a4paper}{article}

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

[2007/05/30 v1.0]

- [2007/09/09 v1.1]

- ## 6 Index

Symbols	C
\# . . . . .	238
\% . . . . .	241
\@ . . . . .	209, 234
\@@input . . . . .	107
\@PackageError . . . . .	161
\@PackageErrorNoLine . . . . .	85, 109
\@addtofilelist . . . . .	149
\@ehc . . . . .	163
\@ehd . . . . .	88, 112
\@filef@und . . . . .	145
\@inputcheck . . . . .	123, 125, 126, 129
\@makeother . . . . .	259
\@ne . . . . .	218, 226
\@nil . . . . .	145, 155
\[ . . . . .	239
\\ . . . . .	135, 235
\{ . . . . .	236
\} . . . . .	237
\] . . . . .	240
\_ . . . . .	242
<b>A</b>	
\advance . . . . .	218, 226
<b>B</b>	
\begin . . . . .	263
\CatchFile@AtEnd . . . . .	60, 61, 90, 114, 206
\CatchFile@CheckFileExists . . . . .	117, 166, 179
\CatchFile@Contents . . . . .	173, 175
\CatchFile@DefFile . . . . .	145, 155
\CatchFile@Do . . . . .	188, 194
\CatchFile@EOF . . . . .	185, 189, 198
\CatchFile@File . . . . .	127, 130, 136, 138, 149, 152, 156, 167, 173, 180, 195
\CatchFile@Input . . . . .	94, 173, 195
\CatchFile@NotFound . . . . .	159, 168, 181
\CatchFile@Primitive . . . . .	100, 101
\CatchFileDef . . . . .	1, 178, 265, 288
\CatchFileEdef . . . . .	165, 268, 291
\CatchFileNotFound . . . . .	159
\catcode . . . . .	3, 4, 5, 6, 7, 31, 32, 33, 34, 35, 36, 37, 38, 56, 58, 62, 64, 201, 202, 209, 215, 224, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 283
\Check . . . . .	95, 106, 107, 108
\closein . . . . .	129
\contents . . . . .	265, 266, 268, 273, 288, 289, 291, 296
\count@ . . . . .	211, 215, 217, 218, 222, 224, 225, 226
\csname . . . . .	8, 18, 39, 52, 55, 74, 113, 118, 120, 122, 123, 147, 248, 298

<b>D</b>		<b>\nofiles</b> . . . . . 253
<b>\do</b> . . . . .	259, 282	<b>O</b>
<b>\documentclass</b> . . . . .	255	
<b>\dospecials</b> . . . . .	260, 285	<b>\obeylines</b> . . . . . 271, 294
<b>E</b>		<b>\openin</b> . . . . . 125
<b>\empty</b> . . . . .	12	<b>P</b>
<b>\end</b> . . . . .	249, 274, 299	
<b>\endcsname</b> 8, 18, 39, 52, 55, 74, 113,		
118, 120, 122, 123, 147, 248, 298		
<b>\endinginput</b> . . . . .	27, 91, 115	
<b>\escapechar</b> . . . . .	80	<b>\PackageInfo</b> . . . . . 23
<b>\everyeof</b> . . . . .	81, 82, 86, 171, 184	<b>\par</b> . . . . . 270, 293
<b>I</b>		<b>\pdffilesize</b> . . . . . 135
<b>\ifcase</b> . . . . .	9	<b>\pdfprimitive</b> . . . . . 108
<b>\ifeof</b> . . . . .	126	<b>\ProvidesPackage</b> . . . . . 53
<b>\IfFileExists</b> . . . . .	144	<b>R</b>
<b>\ifnum</b> . . . . .	217, 225	
<b>\ifx</b> . . . . .	10, 12, 18, 39, 47, 74, 83,	
98, 118, 120, 122, 135, 147, 167, 180		
<b>\immediate</b> . . . . .	20, 41, 277	
<b>\input</b> . . . . .	75, 106, 107, 108, 110, 244, 279	<b>\RangeCatcodeInvalid</b> . . . . .
<b>L</b>		. . . . . 221, 230, 231, 232, 233
<b>\lccode</b> . . . . .	199, 200	<b>\repeat</b> . . . . . 219, 227
<b>\listfiles</b> . . . . .	254	<b>\RequirePackage</b> . . . . . 77
<b>\loop</b> . . . . .	212, 223	<b>\RestoreCatcodes</b> . . . . . 210, 213, 214, 245
<b>\lowercase</b> . . . . .	203	<b>S</b>
<b>M</b>		
<b>\makeatletter</b> . . . . .	257	<b>T</b>
<b>\makeatother</b> . . . . .	262	
<b>\meaning</b> . . . . .	82, 97, 266, 289	
<b>\MessageBreak</b> . . . . .	86, 110	
<b>\msg</b> . . . . .	277, 289, 296	
<b>\mysetup</b> . . . . .	258, 265, 269, 281, 288, 292	<b>\Test</b> . . . . . 229, 247
<b>N</b>		<b>\TestMeaning</b> . . . . . 82, 83, 97, 98
<b>\NeedsTeXFormat</b> . . . . .	252	<b>\TestString</b> . . . . . 81, 83, 96, 98
<b>\newlinechar</b> . . . . .	278	<b>\the</b> . . . . . 56, 62, 215
<b>X</b>		<b>\TMP@EnsureCode</b> . . . . .
<b>\x</b> . . . . .	8, 10, 12, 19, 23, 25, 40, 45, 52	. . . . . 59, 66, 67, 68, 69, 70, 71, 72
<b>W</b>		<b>\typeout</b> . . . . . 266, 273
<b>X</b>		<b>U</b>
<b>\write</b> . . . . .	20, 41, 277	
<b>X</b>		<b>\usepackage</b> . . . . . 256
<b>W</b>		<b>W</b>
<b>X</b>		
<b>X</b>		<b>\write</b> . . . . . 20, 41, 277