

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 ε -TeX. Both L^AT_EX and plain-T_EX 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 T _E X	7
3.3	plain-T _E X	8
4	Installation	8
4.1	Download	8
4.2	Bundle installation	8
4.3	Package installation	9
4.4	Refresh file name databases	9
4.5	Some details for the interested	9
5	History	10
	[2007/05/30 v1.0]	10
	[2007/09/09 v1.1]	10
	[2007/11/11 v1.2]	10
6	Index	10

1 Documentation

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

<code>\CatchFileDef {<cmd>} {<file name>} {<setup>}</code>
<code>\CatchFileEdef {<cmd>} {<file name>} {<setup>}</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^AT_EX.

```
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\relax
54      \xdef#1{#3}%
55      \fi
56    }%
57  \fi
58 \expandafter\x\csname ver@catchfile.sty\endcsname
59 \ProvidesPackage{catchfile}%
60 [2007/11/11 v1.2 Catches the contents of a file (H0)]

```

2.2 Catcodes

```

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

```

2.3 Preparations

```

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

```

Check for ε -TeX's `\everyeof`.

```

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

```

```

108 \CatchFile@AtEnd
109 \expandafter\endinput
110 \fi
111 \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`).

```

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

```

2.5 Input file check

`\CatchFile@CheckFileExists`

```

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

```

```

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

```

\CatchFileNotFound

```

179 \def\CatchFile@NotFound#1#2{%
180 \def#1{%
181 \@PackageError{catchfile}{%
182 File ‘#2’ not found%
183 }\@ehc
184 }

```

2.6 Catch file contents

\CatchFileEdef

```

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

```

\CatchFileDef

```

198 \long\def\CatchFileDef#1#2#3{%
199 \CatchFile@CheckFileExists{#2}%
200 \ifx\CatchFile@File\relax
201 \CatchFile@NotFound{#1}{#2}%
202 \else
203 \begingroup
204 \everyeof\expandafter{%
205 \CatchFile@EOF
206 \noexpand
207 }%
208 \expandafter\long\expandafter\def\expandafter\CatchFile@Do
209 \expandafter##\expandafter1\CatchFile@EOF{%
210 \endgroup

```

```

211      \def#1{##1}%
212    }%
213    #3%
214    \expandafter\expandafter\expandafter\CatchFile@Do
215    \CatchFile@Input\CatchFile@File\relax
216  \fi
217 }

```

\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.

```

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

226 \CatchFile@AtEnd
227 \endpackage

```

3 Test

3.1 Catcode checks for loading

```

228 (*test1)

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

```

```

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

```

3.2 L^AT_EX

```

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

```

```

322
323 \CatchFileEdef\contents{catchfile.sty}{%
324   \mysetup
325   \def\par{^^J}%
326   \obeylines
327 }
328 \typeout{\contents}
329 \end{document}
330 \test2)

```

3.3 plain-TeX

```

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

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

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 (`teTEX`, `mikTEX`, ...) relies on file name databases, you must refresh these. For example, `teTEX` 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 \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 pdfL^AT_EX:

```
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 pdf_{tex}cmds for L^AT_EX 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	C
\# 231, 293	\CatchFile@AtEnd 78, 79, 108, 132, 226
\% 296	\CatchFile@CheckFileExists 135, 186, 199
\@ 232, 289	\CatchFile@Contents 193, 195
\@input 125	\CatchFile@DefFile 165, 175
\@PackageError 181	\CatchFile@Do 208, 214
\@PackageErrorNoLine 103, 127	\CatchFile@EOF 205, 209, 218
\@addtofilelist 169	\CatchFile@File 146, 149, 156, 158, 169, 172, 176, 187, 193, 200, 215
\@ehc 183	\CatchFile@Input 112, 193, 215
\@ehd 106, 130	\CatchFile@NotFound ... 179, 188, 201
\@filef@und 165	\CatchFile@Primitive 118, 119
\@firstofone 240, 243	\CatchFileDef 1, 198, 320, 343
\@gobble 237, 245	\CatchFileEdef 185, 323, 346
\@inputcheck 142, 144, 145, 148	\CatchFileNotFound 179
\@makeother 314	\catcode 3, 4, 5, 6, 7, 18, 19, 20, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 62, 63, 66, 67, 68, 69, 73, 74, 75, 76, 80, 82, 221, 222, 229, 230, 231, 232, 267, 276, 289, 290, 291, 292, 293, 294, 295, 296, 297, 298, 338
\@nil 165, 175	\Check 113, 124, 125, 126
\[..... 294	\closein 148
\\ 290	\contents 320, 321, 323, 328, 343, 344, 346, 351
\{ 229, 291	\count@ 234, 263, 267, 269, 270, 274, 276, 277, 278
\} 230, 292	\countdef 234
\] 295	\csname 8, 21, 45, 58, 65, 92, 131, 136, 139, 141, 142, 167,
_ 297	
A	B
\advance 270, 278	\begin 318
	\body 249, 253

233, 236, 239, 242, 281, 303, 353	N	
D	\NeedsTeXFormat	307
\do	\newlinechar	333
\documentclass	\next	254, 256, 258
\dospecials	\nofiles	308
E	O	
\empty	\obeylines	326, 349
\end	\openin	144
\endcsname	P	
8, 21, 45, 58, 65,	\PackageInfo	26
92, 131, 136, 139, 141, 142, 167,	\par	325, 348
233, 236, 239, 242, 281, 303, 353	\pdf@filesize	155
\endinput	\pdfprimitive	126
\escapechar	\ProvidesPackage	59
\everyeof	99, 100, 104, 191, 204	
I	R	
\ifcase	\RangeCatcodeInvalid	
\ifeof	273, 285, 286, 287, 288	
\IfFileExists	\repeat	248, 260, 271, 279
\ifnum	\RequirePackage	95
\ifx	\RestoreCatcodes	262, 265, 266, 300
10, 12, 21, 45, 53, 92, 101,	S	
116, 136, 139, 141, 154, 167,	\space	193
187, 200, 233, 236, 239, 242, 281	T	
\immediate	\Test	284, 302
\input	\TestMeaning	100, 101, 115, 116
124, 125, 126, 128, 137, 282, 334	\TestString	99, 101, 114, 116
\iterate	\the	66, 67, 68, 69, 80, 267
250, 252, 254	\TMP@EnsureCode	
L	77, 84, 85, 86, 87, 88, 89, 90	
\lccode	\typeout	321, 328
\listfiles	U	
\LoadCommand	\usepackage	311
\loop	W	
\lowercase	\write	23, 47, 332
M	X	
\makeatletter	\x	8, 10, 12, 22, 26, 28, 46, 51, 58, 64, 72
\makeatother		
\meaning		
\MessageBreak		
\msg		
\mysetup		