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 LATEX and plain-TEX are supported.

Contents

1 Documentation

The package relies on ε-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 LaTeX.

```
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
```

```
      #2[{#3}]%
      \ifx#1\@undefined
        \xdef#1{#3}%
      \fi
      \ifx#1\relax
        \xdef#1{#3}%
      \fi
    }%
  \fi
\expandafter\x\csname ver@catchfile.sty\endcsname
\ProvidesPackage{catchfile}%
  [2007/11/11 v1.2 Catches the contents of a file (HO)]
```

2.2 Catcodes

```
\begingroup
  \catcode123 1 % {
  \catcode125 2 % }
  \def\x{\endgroup
    \expandafter\edef\csname CatchFile@AtEnd\endcsname{%
      \catcode35 \the\catcode35\relax
      \catcode64 \the\catcode64\relax
      \catcode123 \the\catcode123\relax
      \catcode125 \the\catcode125\relax
    }%
  }%
\x
\catcode35 6 % #
\catcode64 11 % @
\catcode123 1 % {
\catcode125 2 % }
\def\TMP@EnsureCode#1#2{%
  \edef\CatchFile@AtEnd{%
    \CatchFile@AtEnd
    \catcode#1 \the\catcode#1\relax
  }%
  \catcode#1 #2\relax
}
\TMP@EnsureCode{39}{12}% '
\TMP@EnsureCode{44}{12}% ,
\TMP@EnsureCode{45}{12}% -
\TMP@EnsureCode{46}{12}% .
\TMP@EnsureCode{47}{12}% /
\TMP@EnsureCode{61}{12}% =
\TMP@EnsureCode{96}{12}% `
```

2.3 Preparations

```
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname RequirePackage\endcsname\relax
  \input infwarerr.sty\relax
\else
  \RequirePackage{infwarerr}[2007/09/09]%
\fi
```

Check for ε-TEX's `\everyeof`.

```
\begingroup
  \escapechar=92\relax
  \edef\TestString{\string\everyeof}%
  \edef\TestMeaning{\meaning\everyeof}%
  \ifx\TestString\TestMeaning
  \else
    \@PackageErrorNoLine{catchfile}{%
      Cannot find e-TeX's \string\everyeof,\MessageBreak
```

```
      package loading is aborted%
    }\@ehd
    \endgroup
    \CatchFile@AtEnd
    \expandafter\endinput
  \fi
\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, LaTeX's `\input` is not expandable, but it stores the primitive in `\@@input`. The third possibility is `\pdfprimitive`, introduced in pdfTeX 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`).

```
\begingroup
  \def\Check#1#2#3#4\endgroup{%
    \edef\TestString{\string#1}%
    \edef\TestMeaning{\meaning#2}%
    \ifx\TestString\TestMeaning
      \endgroup
      \let\CatchFile@Primitive#2%
      \def\CatchFile@Input{\CatchFile@Primitive#3}%
    \else
      #4\endgroup
    \fi
  }%
  \Check\input\input{}%
  \Check\input\@@input{}%
  \Check\pdfprimitive\pdfprimitive\input
  \@PackageErrorNoLine{%
    Cannot find primitive \string\input,\MessageBreak
    package loading is aborted%
  }\@ehd
  \csname endgroup\endcsname
  \CatchFile@AtEnd
  \endinput
\endgroup
```

2.5 Input file check

\CatchFile@CheckFileExists

```
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname IfFileExists\endcsname\relax
  \input pdftexcmds.sty\relax
  \begingroup\expandafter\expandafter\expandafter\endgroup
  \expandafter\ifx\csname pdf@filesize\endcsname\relax
    \def\CatchFile@CheckFileExists#1{%
      \expandafter\ifx\csname @inputcheck\endcsname\relax
        \csname newread\endcsname\@inputcheck
      \fi
      \openin\@inputcheck#1\relax
      \ifeof\@inputcheck
        \let\CatchFile@File\relax
      \else
        \closein\@inputcheck
        \def\CatchFile@File{#1}%
      \fi
    }%
  \else
    \def\CatchFile@CheckFileExists#1{%
```

```
      \expandafter\expandafter\expandafter\ifx
      \expandafter\expandafter\expandafter\relax\pdf@filesize{#1}\relax
        \let\CatchFile@File\relax
      \else
        \def\CatchFile@File{#1}%
      \fi
    }%
  \fi
\else
  \def\CatchFile@CheckFileExists#1{%
    \IfFileExists{#1}{%
      \expandafter\CatchFile@DefFile\@filef@und\@nil
      \begingroup\expandafter\expandafter\expandafter\endgroup
      \expandafter\ifx\csname @addtofilelist\endcsname\relax
      \else
        \@addtofilelist\CatchFile@File
      \fi
    }{%
      \let\CatchFile@File\relax
    }%
  }%
  \def\CatchFile@DefFile#1 \@nil{%
    \def\CatchFile@File{#1}%
  }%
\fi
```

\CatchFileNotFound

```
\def\CatchFile@NotFound#1#2{%
  \def#1{}%
  \@PackageError{catchfile}{%
    File `#2' not found%
  }\@ehc
}
```

2.6 Catch file contents

\CatchFileEdef

```
\long\def\CatchFileEdef#1#2#3{%
  \CatchFile@CheckFileExists{#2}%
  \ifx\CatchFile@File\relax
    \CatchFile@NotFound{#1}{#2}%
  \else
    \begingroup
      \everyeof{\noexpand}%
      #3%
      \xdef\CatchFile@Contents{\CatchFile@Input\CatchFile@File\space}%
    \endgroup
    \let#1\CatchFile@Contents
  \fi
}
```

\CatchFileDef

```
\long\def\CatchFileDef#1#2#3{%
  \CatchFile@CheckFileExists{#2}%
  \ifx\CatchFile@File\relax
    \CatchFile@NotFound{#1}{#2}%
  \else
    \begingroup
      \everyeof\expandafter{%
        \CatchFile@EOF
        \noexpand
      }%
```

```
      \expandafter\long\expandafter\def\expandafter\CatchFile@Do
          \expandafter##\expandafter1\CatchFile@EOF{%
        \endgroup
        \def#1{##1}%
      }%
      #3%
    \expandafter\expandafter\expandafter\CatchFile@Do
    \CatchFile@Input\CatchFile@File\relax
  \fi
}
```

\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.

```
\begingroup
  \lccode65=64 % lowercase('A') = '@'
  \lccode66=64 % lowercase('B') = '@'
  \catcode65=8 % catcode('A') = subscript
  \catcode66=3 % catcode('B') = math shift
\lowercase{\endgroup
  \def\CatchFile@EOF{AB}%
}

\CatchFile@AtEnd
⟨/package⟩
```

3 Test

3.1 Catcode checks for loading

⟨*test1⟩

```
\catcode`\{=1 %
\catcode`\}=2 %
\catcode`\#=6 %
\catcode`\@=11 %
\expandafter\ifx\csname count@\endcsname\relax
  \countdef\count@=255 %
\fi
\expandafter\ifx\csname @gobble\endcsname\relax
  \long\def\@gobble#1{}%
\fi
\expandafter\ifx\csname @firstofone\endcsname\relax
  \long\def\@firstofone#1{#1}%
\fi
\expandafter\ifx\csname loop\endcsname\relax
  \expandafter\@firstofone
\else
  \expandafter\@gobble
\fi
{%
  \def\loop#1\repeat{%
    \def\body{#1}%
    \iterate
  }%
  \def\iterate{%
    \body
      \let\next\iterate
    \else
      \let\next\relax
    \fi
    \next
```

```
  }%
  \let\repeat=\fi
}%
\def\RestoreCatcodes{}
\count@=0 %
\loop
  \edef\RestoreCatcodes{%
    \RestoreCatcodes
    \catcode\the\count@=\the\catcode\count@\relax
  }%
\ifnum\count@<255 %
  \advance\count@ 1 %
\repeat

\def\RangeCatcodeInvalid#1#2{%
  \count@=#1\relax
  \loop
    \catcode\count@=15 %
  \ifnum\count@<#2\relax
    \advance\count@ 1 %
  \repeat
}
\expandafter\ifx\csname LoadCommand\endcsname\relax
  \def\LoadCommand{\input catchfile.sty\relax}%
\fi
\def\Test{%
  \RangeCatcodeInvalid{0}{47}%
  \RangeCatcodeInvalid{58}{64}%
  \RangeCatcodeInvalid{91}{96}%
  \RangeCatcodeInvalid{123}{255}%
  \catcode`\@=12 %
  \catcode`\\=0 %
  \catcode`\{=1 %
  \catcode`\}=2 %
  \catcode`\#=6 %
  \catcode`\[=12 %
  \catcode`\]=12 %
  \catcode`\%=14 %
  \catcode`\ =10 %
  \catcode13=5 %
  \LoadCommand
  \RestoreCatcodes
}
\Test
\csname @@end\endcsname
\end
```

⟨/test1⟩

3.2 LaTeX

```
⟨*test2⟩
\NeedsTeXFormat{LaTeX2e}
\nofiles
\listfiles
\documentclass{minimal}
\usepackage{catchfile}[2007/11/11]
\makeatletter
\def\mysetup{%
  \let\do\@makeother
  \dospecials
}
\makeatother
\begin{document}
```

```
\CatchFileDef\contents{catchfile.sty}\mysetup
\typeout{\meaning\contents}

\CatchFileEdef\contents{catchfile.sty}{%
  \mysetup
  \def\par{^^J}%
  \obeylines
}
\typeout{\contents}
\end{document}
⟨/test2⟩
```

3.3 plain-TEX

```
⟨*test3⟩
\def\msg#{\immediate\write16}
\newlinechar=10 %
\input catchfile.sty\relax

\def\mysetup{%
  \def\do##1{%
    \catcode`##1=12\relax
  }%
  \dospecials
}

\CatchFileDef\contents{catchfile.sty}\mysetup
\msg{\meaning\contents}

\CatchFileEdef\contents{catchfile.sty}{%
  \mysetup
  \def\par{^^J}%
  \obeylines
}
\msg{\contents}

\csname @@end\endcsname
\end
⟨/test3⟩
```

4 Installation

4.1 Download

Package. This package is available on CTAN[1]:

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.

[1] ftp://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 (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 pdfLaTeX:

```
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 LuaTeX 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.