The catchfile package

Heiko Oberdiek*

2019/12/09 v1.8

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

*Please report any issues at https://github.com/ho-tex/catchfile/issues

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\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 % #
```

```
  \catcode39=12 % '
  \catcode40=12 % (
  \catcode41=12 % )
  \catcode44=12 % ,
  \catcode45=12 % -
  \catcode46=12 % .
  \catcode47=12 % /
  \catcode58=12 % :
  \catcode64=11 % @
  \catcode91=12 % [
  \catcode93=12 % ]
  \catcode123=1 % {
  \catcode125=2 % }
  \expandafter\ifx\csname ProvidesPackage\endcsname\relax
    \def\x#1#2#3[#4]{\endgroup
      \immediate\write-1{Package: #3 #4}%
      \xdef#1{#4}%
    }%
  \else
    \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}%
  [2019/12/09 v1.8 Catch the contents of a file (HO)]%
```

2.2 Catcodes

```
\begingroup\catcode61\catcode48\catcode32=10\relax%
  \catcode13=5 % ^^M
  \endlinechar=13 %
  \catcode123=1 % {
  \catcode125=2 % }
  \catcode64=11 % @
  \def\x{\endgroup
    \expandafter\edef\csname CatchFile@AtEnd\endcsname{%
      \endlinechar=\the\endlinechar\relax
      \catcode13=\the\catcode13\relax
      \catcode32=\the\catcode32\relax
      \catcode35=\the\catcode35\relax
      \catcode61=\the\catcode61\relax
      \catcode64=\the\catcode64\relax
      \catcode123=\the\catcode123\relax
      \catcode125=\the\catcode125\relax
    }%
  }%
\x\catcode61\catcode48\catcode32=10\relax%
\catcode13=5 % ^^M
\endlinechar=13 %
\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{91}{12}% [
\TMP@EnsureCode{93}{12}% ]
\TMP@EnsureCode{96}{12}% `
\edef\CatchFile@AtEnd{\CatchFile@AtEnd\noexpand\endinput}
```

2.3 Preparations

```
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname RequirePackage\endcsname\relax
  \input infwarerr.sty\relax
  \input ltxcmds.sty\relax
\else
  \RequirePackage{infwarerr}[2007/09/09]%
  \RequirePackage{ltxcmds}[2010/03/09]%
\fi
```

Check for ε-TEX's `\everyeof`.

```
\begingroup
  \escapechar=92\relax
  \edef\TestString{\string\everyeof}%
  \edef\TestMeaning{\meaning\everyeof}%
  \ifx\TestString\TestMeaning
  \else
    \@PackageError{catchfile}{%
      Cannot find e-TeX's \string\everyeof,\MessageBreak
      package loading is aborted%
    }\@ehd
    \endgroup
    \expandafter\CatchFile@AtEnd
  \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
  \@PackageError{%
    Cannot find primitive \string\input,\MessageBreak
    package loading is aborted%
  }\@ehd
  \csname endgroup\endcsname
  \CatchFile@AtEnd%
\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
      \expandafter\closein\expandafter\@inputcheck
      \ifeof\@inputcheck
        \let\CatchFile@File\relax
      \else
        \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
```

\CatchFile@NotFound

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

2.6 Catch file contents

```
\ltx@IfUndefined{RequirePackage}{%
  \input etexcmds.sty\relax
}{%
  \RequirePackage{etexcmds}[2010/01/28]%
}
```

\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
        \expandafter\CatchFile@Finish
        \noexpand
      }%
      \ltx@ifundefined{etex@unexpanded}{%
        \expandafter\long\expandafter\def\expandafter\CatchFile@Do
            \expandafter##\expandafter1\CatchFile@EOF{%
          \toks\ltx@zero{##1}%
          \xdef\CatchFile@gtemp{\the\toks\ltx@zero}%
          \def\CatchFile@Finish{%
            \endgroup
            \let#1\CatchFile@gtemp
```

```
            \global\let\CatchFile@gtemp\ltx@undefined
          }%
        }%
      }{%
        \expandafter\long\expandafter\def\expandafter\CatchFile@Do
            \expandafter##\expandafter1\CatchFile@EOF{%
          \edef\CatchFile@Finish{%
            \endgroup
            \etex@unexpanded{%
              \edef#1{\etex@unexpanded{##1}}%
            }%
          }%
        }%
      }%
      #3\relax
    \expandafter\expandafter\expandafter\CatchFile@Do
    \CatchFile@Input\CatchFile@File\relax
  \fi
}
```

`\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.

```
\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 Installation

3.1 Download

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

CTAN:macros/latex/contrib/catchfile/catchfile.dtx The source file.

CTAN:macros/latex/contrib/catchfile/catchfile.pdf Documentation.

Bundle. All the packages of the bundle 'catchfile' 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.

[1] CTAN:pkg/catchfile

CTAN:install/macros/latex/contrib/catchfile.tds.zip

TDS refers to the standard "A Directory Structure for TEX Files" (CTAN:pkg/tds). Directories with `texmf` in their name are usually organized this way.

3.2 Bundle installation

Unpacking. Unpack the `catchfile.tds.zip` in the TDS tree (also known as `texmf` tree) of your choice. Example (linux):

```
unzip catchfile.tds.zip -d ~/texmf
```

3.3 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/catchfile/catchfile.sty
catchfile.pdf → doc/latex/catchfile/catchfile.pdf
catchfile.dtx → source/latex/catchfile/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.

3.4 Refresh file name databases

If your TEX distribution (TEX Live, mikTEX, ...) relies on file name databases, you must refresh these. For example, TEX Live users run `texhash` or `mktexlsr`.

3.5 Some details for the interested

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
```

4 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.

[2019/12/09 v1.8]

- Documentation updates.

5 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.