

The *currfile* Package

Martin Scharrer

`martin@scharrer-online.de`

<http://www.ctan.org/pkg/currfile/>

Version v0.2 – 2010/12/08

Abstract

This small package provides the file name and path information of the current input file as \LaTeX macros.

1 Usage

<code>\currfiledir</code>	The directory, base (name without extension), extension (without dot), name
<code>\currfilebase</code>	(=base+'.'+ext) and path (=dir+name) of the current file are provided by these
<code>\currfileext</code>	macros. This means that the macros returns the file information of the file they
<code>\currfilename</code>	are used in. All macros are fully expanded, i.e. only hold text and not further
<code>\currfilepath</code>	macros. Special care is taken to keep the file information of <code>\included</code> files till
	the final <code>\clearpage</code> command, so that page header and footer of the last page
	will hold the correct data.

Since v0.2 all files are taken into account, i.e. files read using `\input`, `\include`, `\InputIfFileExists`, `\usepackage`, `\RequirePackage` and even `\LoadClass` and similar macros. Before v0.2 only `\input` or `\include` and the main file were taken into account.

This package uses the `filehook` package written by the same author. See there for possible incompatibilities with classes or other packages.

More detailed information can be found in the implementation section 3 if required.

Package Options

The package provides two options `mainext` and `maindir` which can be used to provide the extension and directory of the main file. This is required if the above macros should be used for the main file itself and if this does has a file extension other than `'\tex'` (e.g. a `.dtx` file) or is not located in the current directory. To provide support for the macros defined by the `fink` package (see section 2) a `fink` option exists.

Table 1: Conversion from `fink` package to `currfile`.

<code>fink</code>	<code>currfile</code>	Example Result
<code>\finkdir</code>	<code>\currfiledir</code>	
<code>\finkbase</code>	<code>\currfilebase</code>	<code>currfile</code>
<code>\finkext</code>	<code>\currfileext</code>	<code>dtx</code>
<code>\finkfile</code>	<code>\currfilename</code>	<code>currfile.dtx</code>
<code>\finkpath</code>	<code>\currfilepath</code>	<code>currfile.dtx</code>

2 Compatibility with the `fink` package

The `fink` package (*file name keeper*) provides a similar functionality. It has inspired this package in several points (e.g. package options). However, it does not exclude package and other preamble files and does not take care to change the filename *after* the `\clearpage` of `\include`. The author of `fink` is now discontinuing it in favour of this package. Existing documents which use `fink` should either rename the related macros as shown by Table 1 or use the `fink` option of `currfile` which defines the `fink` macros to use the `currfile` ones.

Because both packages do basically the same thing, especially patch the same macros, there are incompatible and should not be loaded both. In consent with the `fink` package author this package will undo most of the `fink` code if it was already loaded or prevent it from being loaded afterwards.

3 Implementation

3.1 Options

```

1 \RequirePackage{kvoptions}
2 \SetupKeyvalOptions{family=currfile,prefix=currfile@}
3
4 \@ifl@aded{\@pkgextension}{fink}{%
5   \DeclareStringOption[\fnk@mainext]{mainext}%
6   \DeclareStringOption[\fnk@maindir]{maindir}%
7   \DeclareBoolOption[true]{fink}%
8   \PackageWarning{currfile}{Package 'fink' detected. %
9     The 'fink' option will default to 'true'. %
10    If set to 'false' no 'fink' macros will be changed but they will stop
11    working correctly!}{-}{-}}%
12 }{%
13   \DeclareStringOption[tex]{mainext}%
14   \DeclareStringOption[\@currdir]{maindir}%
15   \DeclareBoolOption[false]{fink}%
16 }%
17 \ProcessKeyvalOptions*\relax
18
19 \begingroup
```

```

20 \xdef\currfile@mainext{\currfile@mainext}%
21 \xdef\currfile@maindir{\currfile@maindir}%
22 \def\@tempa{./}%
23 \ifx\@tempa\currfile@maindir
24     \global\let\currfile@maindir\empty
25 \fi
26 \endgroup

```

3.2 File Hooks

The `filehook` package is used to execute the macros at the correct places. The internal interface, not the user-interface, is used to make sure that the file names are valid for all other hooks.

```

27 \RequirePackage{filehook}
28 \filehook@prefixwarg\filehook@include@atbegin{%
29     \currfile@push
30     \currfile@set{#1}%
31 }
32 \filehook@appendwarg\filehook@include@after{%
33     \currfile@pop
34 }
35 \filehook@prefixwarg\filehook@atbegin{%
36     \currfile@push
37     \currfile@set{#1}%
38 }
39 \filehook@appendwarg\filehook@atend{%
40     \currfile@pop
41 }

```

3.3 Set Current Values

`\currfile@set` Sets the file information which are parsed by L^AT_EX's `\filename@parse`.

```

42 \def\currfile@set#1{%
43     \begingroup
44     \filename@parse{#1}%
45     \global\let\currfiledir\filename@area
46     \global\let\currfilebase\filename@base
47     \xdef\currfileext{\ifx\filename@ext\relax tex\else\filename@ext\fi}%
48     \xdef\currfilename{\currfilebase\ifx\currfileext\empty\else.\currfileext\fi}%
49     \xdef\currfilepath{\currfiledir\currfilename}%
50     \endgroup
51 <debug> \expandafter\gdef\expandafter\dindent\expandafter{\dindent\space}%
52 <debug> \message{^^JDEBUG: \dindent\empty Entering file '\currfilename' ^^J}%
53 }

```

3.4 File Stack

The file information are pushed and popped on a stack to save and restore them when entering and leaving a sub-file, respectively. This is quite similar to the way

L^AT_EX saves file base names and extension as well as the ‘@’ status (letter or other) for package and class files.

`\currfile@push`

```
54 \def\currfile@push{%
55   \xdef\currfile@stack{%
56     {\currfiledir}%
57     {\currfilebase}%
58     {\currfileext}%
59   \currfile@stack
60 }%
61 }
```

`\currfile@pop`

```
62 \def\currfile@pop{%
63 <debug> \message{^^JDEBUG: \dindent\empty Leaving file '\currfilename' ^^J }%
64   \ifx\currfile@stack\empty
65     \PackageWarning{currfile}{File stack underflow!}{-}{-}%
66     \global\let\currfile@stack\currfile@stackinit
67   \fi
68   \expandafter\currfile@pop@\currfile@stack\relax
69   \relax\relax\relax
70 <debug> \message{^^JDEBUG: \dindent\empty Restoring file '\currfilename' ^^J }%
71 }
```

`\currfile@pop@`

```
72 \def\currfile@pop@#1#2#3#4\relax{%
73   \gdef\currfiledir{#1}%
74   \gdef\currfilebase{#2}%
75   \gdef\currfileext{#3}%
76   \xdef\currfilename{\currfilebase\ifx\currfileext\empty\else.\currfileext\fi}%
77   \xdef\currfilepath{\currfiledir\currfilename}%
78   \gdef\currfile@stack{#4}%
79 <debug> \expandafter\expandafter\expandafter\gdef
80 <debug> \expandafter\expandafter\expandafter\dindent
81 <debug> \expandafter\expandafter\expandafter{\expandafter\@gobble\dindent}%
82 }
```

`\currfile@stack` Place `\jobname` values on stack and use this as init value.

`\currfile@stackinit`

```
83 \def\currfile@stack{}
84 \currfile@set{\currfile@maindir\jobname.\currfile@mainext}
85 \currfile@push
86 \let\currfile@stackinit\currfile@stack
```

3.5 Fink Macros

The `fink` option defines all `fink` package macros to use the ones provided by this package. If the `fink` package was loaded beforehand the restoration of these macros must be avoided at the end of this file (`finks \InputIfFileExists` was

```

87 \ifcurrfile@fink
88     \def\finkfile{\currfilename}%
89     \def\finkdir{\currfiledir}%
90     \def\finkpath{\currfilepath}%
91     \def\finkbase{\currfilebase}%
92     \def\finkext{\currfileext}%
93     \@ifl@aded{\@pkgextension}{fink}{%
94         \def\fink@restore#1{%
95             }{%
96                 \@namedef{ver@fink.sty}{2010/12/06}%
97                 \expandafter\edef\csname opt@fink.sty\endcsname{%
98                     maindir=\currfile@maindir,mainext=\currfile@mainext
99                 }%
100             }%
101 \else
102     \@ifl@aded{\@pkgextension}{fink}{}{%
103         \AtBeginOfFile{fink.sty}{%
104             \PackageError{currfile}{Package class with deprecated 'fink'. %
105                 Load 'currfile' with the 'fink' option or see the upgrade guide in the manual}{}{}
106         }%
107     }%
108 \fi

```

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.

5

```

maindir= \subitem maindir+, 6 \subitem **\jobname+, 84 \subitem **\relax+, 17, 47, 68, 69, 70
mainext= \subitem mainext+, 5 \subitem **\RequirePackage+, 24, 45, 46, 66, 86 \subitem **\RequirePackage+
let= \subitem **\let+, 24, 45, 46, 66, 86
gdef= \subitem **\gdef+, 51, 73--75, 78, 79 \subitem **\message+, 52, 63, 79 \subitem **\SetupKeyval
global= \subitem **\global+, 24, 45, 46, 66 \subitem **\SetupKeyvalOptions+, 211, 747--49, 55
PackageError= \subitem **\PackageError+, 104
PackageWarning= \subitem **\PackageWarning+, 8, 65
fink= \subitem fink+, 87
ifx= \subitem **\ifx+, 23, 45, 46, 64, 76 \subitem **\ifx+, 23, 45, 46, 64, 76

```