

# *F<sub>i</sub>NK* – the L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> File Name Keeper\*

Didier Verna

<mailto:didier@lrde.epita.fr>

<http://www.lrde.epita.fr/~didier/>

v2.2.1 (2011/10/19)

**Warning!** As of version 2.2.1, *F<sub>i</sub>NK* has been deprecated and is not maintained anymore. People interested in *F<sub>i</sub>NK*'s functionality are invited to use a package named `currfile` instead.

## 1 Description

This package is a real fink indeed: it looks over your shoulder and keeps track of files `\input`'ed (the L<sup>A</sup>T<sub>E</sub>X way) or `\include`'ed in your document. You then have a permanent access to the directory, name and extension of the file currently being processed through several macros. Dis packache fas orichinally a hack dat I used somefere elss, but since it might be off a cheneral interest, I'fe decided to make it a separate fink. . .

The *F<sub>i</sub>NK* package is Copyright © 1999, 2000, 2001, 2002, 2007, 2008, 2010, 2011 Didier Verna, and distributed under the terms of the LPPL license.

## 2 User Interface

To use the package, simply say `\usepackage[options]{fink}` in the preamble of your document. This will do everything for you. Available options will be described when appropriate.

### 2.1 Retrieving the current file's name components

`\finkdir` The file currently being processed is described by the macros `\finkdir`, `\finkbase`  
`\finkbase` and `\finkext` which expand (as you may have guessed) to the directory, base name  
`\finkext` (sans extension), and extension of the file.

`\finkfile` Additionally, the macro `\finkfile` is defined to be `\finkbase.\finkext` (as in  
`\finkpath` previous versions), and the macro `\finkpath` (new in version 2.0) is defined to be  
`\finkdir\finkfile`. Feel free to use these macros in your sources.

---

\**F<sub>i</sub>NK* homepage: <http://www.lrde.epita.fr/~didier/software/latex.php#fink>

## 2.2 Main file's name components

`maindir` Because there's no way  $\TeX$  can give you back information about the file being processed (apart from its base name), *F<sub>i</sub>NK* provides the options `maindir` (defaults to `./`) and `mainext` (defaults to `tex`) for changing the directory and the extension of the main source file. For instance, suppose your source file is in `src/foo.ltx` and you are compiling in `pdf/`. You can then use the package as follows:

```
\usepackage[maindir=../src,mainext=ltx]{fink}
```

## 3 AUC- $\TeX$ support

AUC- $\TeX$  is a powerful major mode for editing  $\TeX$  documents in Emacs or XEmacs. In particular, it provides automatic completion of macro names once they are known. *F<sub>i</sub>NK* supports AUC- $\TeX$  by providing a style file named `fink.el` which contains AUC- $\TeX$  definitions for the relevant macros. This file should be installed to a location where AUC- $\TeX$  can find it (usually in a subdirectory of your  $\LaTeX$  styles directory). Please refer to the AUC- $\TeX$  documentation for more information on this.

## 4 Caveat

*F<sub>i</sub>NK* cannot follow files included with the  $\TeX$  `\input` primitive. That's because  $\TeX$  has a very insensible way of defining primitives whose argument parsing syntax is not available for macros. As a consequence, it's almost impossible to redefine the `\input` primitive without breaking its syntax (one would have to parse the characters one by one, and I'm not ready to do so...). *F<sub>i</sub>NK* currently does not follow auxiliary files either.

## 5 Hints, Tricks, Tips

### 5.1 File names with special characters

Here, "special" is to be taken in the  $\LaTeX$  sense, for instance, a directory or file name containing an underscore. If this situation occurs, you're likely to face problems with *F<sub>i</sub>NK* macros because they don't try to properly escape those characters. So for instance, an underscore alone will make  $\LaTeX$  think that you forgot the math mode  $\$$  sign before it. There are actually two problems that you may encounter:

**Characters not displayed properly** Try to change your font encoding by putting this in your document's preamble: `\usepackage[T1]{fontenc}`.

**Compilation breakage** The `url` package might be of some help here. Put `\usepackage{url}` in your document's preamble first. Then (assuming that `\finkfile` is the culprit), instead of using `\finkfile` directly, use this instead: `\expandafter\url\expandafter{\finkfile}`. You might also want to play with `\urlstyle` to have your file name displayed in the font you prefer.

## 6 Changes

- v2.2.1 Declare package as obsolete.
  
- v2.2 Fix incompatibility with the `memoir` class, reported by Lars Madsen
  
- v2.1.1 Fix trailing whitespace in `\fink@restore`, reported by Maverick Woo  
Added some hints about filenames with special characters, suggested by David P. Goodall
  
- v2.1 Fix bug preventing expansion in math mode, reported by Alain Schremmer, fixed by Morten Hoegholm before I could even raise my little finger.
  
- v2.0 New macros `\finkdir`, `\finkbase`, `\finkext` and `\finkpath` suggested by Alain Schremmer  
New options `mainext` and `maindir`, use `kvoptions` for options management
  
- v1.2 Fixed conflict with `\includegraphics`, reported by Jim Crumley
  
- v1.1 Fixed missing 3rd arg to `\PackageError` call from `\finkextension`

## 7 The Code

```
1 \fink\NeedsTeXFormat{LaTeX2e}
2 \*header
3 \ProvidesPackage{fink}[2011/10/19 v2.2.1 Keep track of the current filename]
4
5 \</header>
6 \*fink
7 \PackageWarning{FiNK}{as of version 2.2.1, FiNK has been deprecated and is
8 not\MessageBreak
9 maintained anymore. People interested in FiNK's\MessageBreak
10 functionality are invited to use a package named\MessageBreak
11 "currfile" instead,}
12
13 \RequirePackage{kvoptions}
14 \SetupKeyvalOptions{family=fnk,prefix=fnk@}
15
```

### 7.1 Main file initial settings

```
maindir
mainext
16 \DeclareStringOption[\@currdir]{maindir}
17 \DeclareStringOption[teX]{mainext}
18
```

The following is for backward compatibility only (not documented anymore). It provides support for the old `tex` and `ltx` options (still functional), and for the `\finkextension` macro. However, this macro is now defined to trigger an error, begging the user to use the new option instead.

```
19 \newcommand*\@fink@mainext[1]{\setkeys{fnk}{mainext={#1}}}
20 \newcommand*\fink@mainext{%
```

```

21 \expandafter\@fink@mainext\expandafter{\CurrentOption}}
22 \DeclareVoidOption{tex}{\fink@mainext}
23 \DeclareVoidOption{ltx}{\fink@mainext}
24
25 \newcommand*\finkextension[1]{%
26   \PackageError{FiNK}{%
27     \protect\finkextension\space shouldn't be used anymore.\MessageBreak
28     Please use the 'mainext' package option instead.}{%
29     No big deal right ?\MessageBreak
30     Type X to quit and modify your source.}}
31 \@onlypreamble\finkextension
32
33 \ProcessKeyvalOptions*
34

```

## 7.2 File's name components macros

```

\finkdir We declare the user-level macros here. \fink@file is used to compute file names,
\finkbase possibly with no extension.
\finkext 35 \newcommand*\finkdir{\fink@maindir}
\finkfile 36 \newcommand*\finkbase{\jobname}
\finkpath 37 \newcommand*\finkext{\fink@mainext}
38
39 \newcommand*\finkfile{}
40 \newcommand*\fink@file[2]{#1\ifx\\#2\\\else.#2\fi}
41 \xdef\finkfile{\fink@file{\jobname}{\fink@mainext}}
42
43 \newcommand*\finkpath{}
44 \xdef\finkpath{\finkdir\finkfile}
45
46 \PackageInfo{FiNK}{main file set to "\finkpath"}
47

```

## 7.3 Commands overriding

```

\fink@beginfile The memoir class redefines \InputIfFileExists as well, in order to provide its
\fink@endfile file hooks. Since we override its definition, we need to take care of those hooks
ourselves.
48 \@ifclassloaded{memoir}{
49   \let\fink@beginfile\m@matbeginf
50   \def\fink@endfile#1{\m@matendf{#1}\killm@matf{#1}}}%
51   \def\fink@beginfile#1{}
52   \def\fink@endfile#1{}
53

```

Note: as of version 1.2, every call to `\filename@parse` is done in a group of its own. This fixes a problem that appeared when using `\includegraphics` with a filename with an explicit extension. `\includegraphics` calls `\filename@parse` itself, so it is important that our call(s) only have a local effect.

```

\fink@input These macros are defined for a convenient use of \expandafter. They save and
\fink@restore restore the current filename. Remember that \@@input is LATEX's redefinition of
the TEX input primitive.

```

```

54 \newcommand*\fink@input[1]{%
55   \begingroup%
56     \filename@parse{#1}%
57   \xdef\finkdir{%
58     \ifx\filename@area\@empty%
59       \fnk@maindir%
60     \else%
61       \fnk@maindir\filename@area%
62     \fi}%
63   \xdef\finkbase{\filename@base}%
64   \xdef\finkext{\ifx\filename@ext\relax tex\else\filename@ext\fi}%
65   \xdef\finkfile{\fink@file{\finkbase}{\finkext}}%
66   \xdef\finkpath{\finkdir\finkfile}%
67 \endgroup%
68 \fink@beginfile{#1}%
69 \@@input\@filef@und%
70 \fink@endfile{#1}}
71
72 \newcommand*\fink@restore[1]{%
73   \begingroup%
74     \filename@parse{#1}%
75   \xdef\finkdir{\filename@area}%
76   \xdef\finkbase{\filename@base}%
77   \xdef\finkext{\filename@ext}%
78   \xdef\finkfile{\fink@file{\finkbase}{\finkext}}%
79   \xdef\finkpath{\finkdir\finkfile}%
80 \endgroup}
81

```

Note: in earlier versions, we redefined `\IfFileExists` to prepare the name of the next file, but this is bad because it can be used outside of *F<sub>i</sub>NK*'s scope. We also redefined `\@input`, but neither `\include` nor `\input` use it.

`\InputIfFileExists` L<sup>A</sup>T<sub>E</sub>X's `\input` and `\include` commands use `\InputIfFileExists`, so let's redefine it here:

```

82 \long\def\InputIfFileExists#1#2{%
83   \IfFileExists{#1}{%
84     #2\@addtofilelist{#1}%
85     \edef\fink@before{\noexpand\fink@input{#1}}%
86     \edef\fink@after{\noexpand\fink@restore{\finkpath}}%
87     \expandafter\fink@before\fink@after}}
88
89 </fink>

```

Well, I think that's it. Enjoy using *F<sub>i</sub>NK*!