

The *currfile* Package

Martin Scharrer

`martin@scharrer-online.de`

`http://www.ctan.org/pkg/standalone/`

Version v0.1 – 2010/04/08

Abstract

This small package provides the file name and path information of the current input file as L^AT_EX macros.

1 Usage

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

Only files loaded with the L^AT_EX macros `\input` or `\include` and the main file are taken into account. Files loaded using package macros like `\usepackage` or internal macros `\@input` and `\@@input` are ignored.

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.

2 Similar Packages

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`.

3 Implementation

3.1 Options

```
1 \RequirePackage{kvoptions}
2 \SetupKeyvalOptions{family=currfile,prefix=currfile@}
3 \DeclareStringOption[tex]{mainext}
4 \DeclareStringOption[] {maindir}
5 \ProcessKeyvalOptions*\relax
```

3.2 File Hooks

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

```
6 \RequirePackage{filehook}
7 \filehook@prefixwarg\filehook@include@atbegin{%
8   \currfile@push
9   \currfile@set{#1}%
10 }
11 \filehook@appendwarg\filehook@include@after{%
12   \currfile@pop
13 }
14 \filehook@prefixwarg\filehook@input@atbegin{%
15   \currfile@push
16   \currfile@set{#1}%
17 }
18 \filehook@appendwarg\filehook@input@atend{%
19   \currfile@pop
20 }
```

3.3 Set Current Values

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

```
21 \def\currfile@set#1{%
22   \begingroup
23     \filename@parse{#1}%
24     \global\let\currfiledir\filename@area
25     \global\let\currfilebase\filename@base
26     \xdef\currfileext{\ifx\filename@ext\relax tex\else\filename@ext\fi}%
27     \xdef\currfilename{\currfilebase\ifx\currfileext\empty\else.\currfileext\fi}%
28     \xdef\currfilepath{\currfiledir\currfilename}%
29   \endgroup
30 <debug> \expandafter\gdef\expandafter\dindent\expandafter{\dindent\space}%
31 <debug> \message{^^JDEBUG: \dindent\empty Entering file '\currfilename' ^^J }%
32 }
```

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`

```
33 \def\currfile@push{%
34   \xdef\currfile@stack{%
35     {\currfiledir}%
36     {\currfilebase}%
37     {\currfileext}%
38     \currfile@stack
39   }%
40 }
```

`\currfile@pop`

```
41 \def\currfile@pop{%
42 <debug> \message{^^JDEBUG: \dindent\empty Leaving file '\currfilename' ^^J }%
43   \ifx\currfile@stack\empty
44     \PackageWarning{currfile}{File stack underflow!}{-}{-}%
45     \global\let\currfile@stack\currfile@stackinit
46     \fi
47   \expandafter\currfile@pop@\currfile@stack\relax
48   \relax\relax\relax
49 <debug> \message{^^JDEBUG: \dindent\empty Restoring file '\currfilename' ^^J }%
50 }
```

`\currfile@pop@`

```
51 \def\currfile@pop@#1#2#3#4\relax{%
52   \gdef\currfiledir{#1}%
53   \gdef\currfilebase{#2}%
54   \gdef\currfileext{#3}%
55   \xdef\currfilename{\currfilebase\ifx\currfileext\empty\else.\currfileext\fi}%
56   \xdef\currfilepath{\currfiledir\currfilename}%
57   \gdef\currfile@stack{#4}%
58 <debug> \expandafter\expandafter\expandafter\gdef
59 <debug> \expandafter\expandafter\expandafter\dindent
60 <debug> \expandafter\expandafter\expandafter{\expandafter@gobble\dindent}%
61 }
```

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

`\currfile@stackinit`

```
62 \def\currfile@stack{}
63 \currfile@set{\currfile@maindir\jobname.\currfile@mainext}
64 \currfile@push
65 \let\currfile@stackinit\currfile@stack
```