# FiXme – a LaTeX 2ε package for inserting fixme notes in your documents *

Didier Verna
mailto:didier@lrde.epita.fr
http://www.lrde.epita.fr/~didier/

April 14, 2006

**Abstract**

In the process of writing a long document, it is a common practice to leave some parts unwritten or uncomplete, and come back to them later. In such cases, you probably want to stick clues about which parts need to be "fixed", where they are located, and what needs to be done. This is what I call "fixme notes". The purpose of this package is to provide you with convenient ways to insert FiXme notes in your documents.



## 1 Description

With FiXme, you can insert different kinds of notes in your documents, ranging from simple not-so-important notices to critical stuff that must absolutely be fixed in the final version.

FiXme gives you full control on the layout of these notes: they can be displayed inline (directly in the text), as marginal paragraphs (the default), as footnotes and even as index entries. All these possibilities can be mixed together. Additionally, you can summarize all FiXme notes in a "list of fixme's".

FiXme notes are also recorded in the log file, and (depending on their importance level) some of them are displayed on the terminal during compilation. A final summary is also created at the end of the compilation process.

All these features are actually available when you're working in `draft` mode. In `final` mode, the behavior is slightly different: any remaining critical note generates an error (the compilation aborts), while non critical ones are just removed from the document's body (they're still recorded in the log file though).

## 2 Using FiXme

First of all, please note that the `ifthen` and `verbatim` packages are required. You don't have to load them explicitly though. As long as LaTeX 2ε can locate them, they will be used automatically.

*This document describes FiXme v3.2, release date 2006/04/12.

To use FiXme, simply say `\usepackage[`⟨*options*⟩`]{fixme}` in the preamble of your document.

## 2.1 Inserting notes

### 2.1.1 Macros

`\fixme` The main command for inserting a FiXme note in your document is the `\fixme` macro. It takes the note to insert as its mandatory argument. Notes inserted via this command are considered fatal to your document's final processing (see section 2.2).

`\fxnote` `\fxwarning` `\fxerror` As of version 2.0, FiXme provides three new macros that insert meta-comments about the document, rather than real fixmes. These comments have three different importance levels: note, warning and (non critical) error. The corresponding macros obey the same syntax as `\fixme`. However, none of the notes inserted via these macros are fatal.

### 2.1.2 Environments

`anfxnote` `anfxwarning` `anfxerror` `afixme` As of version 3.0, FiXme provides environments for inserting longer notes. These environments take one optional argument (should be short, perhaps a summary of the note) that will be used in the list of fixme's and in the index if required.

The FiXme environments behave exactly the same way as their macro counterpart, except for the layout (see section 2.3): as they are meant for longer notes, the layout is always `inline`. The `index` and `marginclue` layouts are still honored however. By default, FiXme environments are typeset in a `quotation` one.

## 2.2 Controlling the behavior of FiXme

`final` `draft` The global behavior of the package is controlled via the two standard options `final` and `draft`. These options are usually given to `\documentclass{}` which in turn passes them to all packages.

In `draft` mode, the notes are recorded in the log file, and appear in the text as specified by the layout settings (see section 2.3). Additionally, warnings, errors and fatal errors are also displayed on the terminal.

In final mode, non fatal notes (those generated by the `\fx*` commands) are still logged, but they're removed from the document's body. On the other hand, any remaining fatal note (generated by `\fixme`) will abort compilation with an informative message. This will help you track down forgotten important caveats in your document. Let me say it again: final documents can only have notes, warnings, and (non critical) errors left. Well, if, for some reason, you really want to compile in `final` mode with critical FiXme notes left behind, you always have the possibility to pass the `draft` option to FiXme directly...

The `final` mode has been chosen as the default because LaTeX 2ε itself behaves this way.

## 2.3 Controlling the notes layout

FiXme notes can appear in several forms (that can be combined) in your document.

### 2.3.1 Global layout

inline
margin
marginclue
footnote
index

The layout forms currently supported are: inline (directly in the text), marginal notes, marginal clues (see below), footnotes, and index entries. To activate a particular layout globally (that is, for the whole document), use the corresponding package option. By default, only the `margin` layout is active.

Each layout option has a counterpart that deactivates it. The counterpart option has the same name, prefixed with `no`. For instance, if you don't want marginal notes, use the `nomargin` option.

Finally, note that FiXme environments behave in a special way: they are always typeset `inline`, regardless of your layout settings (they respect your whish for index and marginal clues though).

### 2.3.2 Marginal Clues

Sometimes, marginal notes are too narrow for what you want to put in them, so you would switch to an inline or a footnote layout. But then, it is more difficult to keep track of the FiXme note's location on the page.

As of version 3.2, FiXme provides a special layout called "marginal clue" to help locating the notes: a marginal clue does not display the note's contents, but only an indication that there is a note at that place. So you have to use another layout form (typically inline or footnote) in order to get the actual contents.

Obvisouly, `margin` and `marginclue` are mutually exclusive. If you try to activate both, only the most recently activated form will be enabled (and you'll get a notice in the compilation log).

### 2.3.3 Local Layout

\fixme[]
\fxnote[]
\fxwarning[]
\fxerror[]

As of version 3.0, FiXme provides a way to change the selected layout(s) on a per-note basis: each note insertion command takes an optional first argument that overrides the global layout. This argument consists of one or more layout options (`inline`, `margin`, `marginclue`, `footnote` and `index`) separated by commas. Remember that local layouts *override* the global ones; they don't add to it.

On what occasion would one want to modify the layout for a particular note ? Here is a typical situation: suppose you have a document in which FiXme notes are typeset as margin paragraphs (this is the default). You would not be able to put a note in a figure, because floats can't be nested in LaTeX (margin paragraphs are floats). In such a case, you would rather inline the note, which can be done with something like `\fixme[inline]{blah}`.

## 2.4 Controlling the notes logging

silent
nosilent

As well as being displayed in the document itself, all FiXme notes are "logged" in different ways: by default, simple notes are recorded in the log file while the others are also displayed on the terminal output.

You have the ability to suppress all kind of logging by using the `silent` option. By default, the behavior is that of `nosilent`.

## 2.5 List of FiXme's

\listoffixmes

FiXme remembers where you put FiXme notes in a toc-like file whose extension

is `lox`. The `\listoffixmes` macro generates the list of all FiXme notes in a manner similar to that of the "list of figures" for instance. A standard layout is automatically used for the 'article', 'report', 'book' classes and their koma-script replacements. If another class is used, the 'article' layout is selected. Also, note that if no FiXme note remain in the document, this macro doesn't generate an empty list, but rather stays silent. It also stays silent in `final` mode, regardless of the presence of remaining notes.

# 3 Customizing FiXme

## 3.1 Customizing the notes layout

### 3.1.1 Macros

The `inline`, `margin` and `footnote` layouts have two parts: a "prefix" which depends on the note level, and the note itself. The prefix is one of "FiXme note:", "FiXme warning:", "FiXme error:" or simply "FiXme:", and appears in bold. The note itself appears emphasized. These layouts are implemented thanks to the corresponding `\FX*` macros that you can redefine if you whish. Each such macro takes two mandatory arguments: the prefix and the note, in that order.

`\FXInline` `\FXMargin` `\FXFootnote`

The special `marginclue` layout only outputs the prefix mentioned above. Its layout is controlled by the macro `\FXMarginClue` which takes the prefix as its only mandatory argument. The `\FXIndex` macro is used for the `index` layout. All FiXme index entries appear under the "FiXme" key in the symbols section. There are 4 subcategories under this key, as many as there are note levels. By default, only the first 3 of them are used though (fatal errors do not appear under a subkey, but directly under the FiXme key). The notes are numbered in the index.

`\FXMarginCLue` `\FXIndex`

### 3.1.2 Environments

The optional argument of FiXme environments is always typeset according to the `inline` layout described above. By default, FiXme uses a `quotation` for displaying the environments'contents. If you want to change that, you can redefine the macros `\FXEnvBegin` and `\FXEnvEnd` that open and close the environments.

`\FXEnvBegin` `\FXEnvEnd`

## 3.2 Customizing the notes logging

If you want a finer control on logging, you can redefine the commands used to implement it. These commands (on the left) are the ones used by `\fxnote`, `\fxwarning`, `\fxerror` and `\fixme` respectively. They take the note itself as mandatory argument.

`\FXNote` `\FXWarning` `\FXError` `\FXFatal`

## 3.3 Fancy fruit salad layout

If, for some totally unjustified reason, you are not happy with the available layouts, you have the ability to define your own: pass the `user` option to the package (it also has its `nouser` counterpart), and define an `\FXUser` macro in the following manner:

`user` `\FXUser`

`\newcommand{\FXUser}[2]{`⟨*fancy fruit salad layout job*⟩`}`

The arguments are the prefix, and the note itself, in that order.

Note that the `user` option can also be used in the optional argument of the notes insertion commands.

### 3.4 Internationalization

`english` `french` `francais` `spanish` `italian` `german` `ngerman` `danish` `croatian`

FiXme currently supports English, French, Spanish, Italian, German, Danish and Croatian. You can select the language you want to use by passing the corresponding option (these options are usually given directly to `\documentclass{}` which in turn passes them to all packages). The `french` and `francais` options are synonyms. The `german` and `ngerman` options are currently equivalent.

If you want a finer grain on the language-dependent parts of FiXme, the following macros are provided and can be redefined.

`\fixmenoteprefix` `\fixmewarningprefix` `\fixmeerrorprefix` `\fixmefatalprefix` `\fixmelogo`

The `\fixme*prefix` macros define the prefix for the four different note levels. They make intensive use of the macro `\fixmelogo` ;-)

`\fixmeindexname` `\fixmenoteindexname` `\fixmewarningindexname` `\fixmeerrorindexname` `\fixmefatalindexname`

The macro `\fixmeindexname` defines the main FiXme index key. The other ones define the different index subkeys for each note level. Please note that an empty name for a subkey means that you don't actually want a subcategory (that's the case by default for fatal errors). The corresponding notes will then appear directly under the main FiXme key. For that reason, a non empty subkey must end with an exclamation mark.

`\listfixmename`

`\listfixmename` defines the title for the "list of fixmes" section.

## 4 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. FiXme supports AUC-TEX by providing a style file named `fixme.el` which contains AUC-TEX definitions for the relevant macros. This file should be installed in a place 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.

## 5 Changes

v3.2 Added the marginclue layout option which only signals a fixme in the margin, withtout the actual contents.
Fix incompatibility with `amsbook` reported by Claude Lacoursière: `\@starttoc` takes two arguments.
Support for Croatian thanks to Marcel Maretic `<marcel@fsb.hr>`.
Fix incompatibility with Beamer reported by Akim Demaille: protect contents of lox file.

v3.1 Fix bug reported by Arnold Beckmann: the environments were visible in final mode.

v3.0 Added environments corresponding to the note insertion commands.
Added an optional first argument to the note insertion commands to change the layout locally.
Fix bug reported by Akim Demaille: marginal notes could mess up the document's layout by flushing it right.

v2.2 New option `silent` to suppress notes logging.
Support for Danish thanks to Kim Rud Bille <krbi01@control.auc.dk>.

v2.1 Use `\nobreakspace` instead of the tilda character. This avoids conflicts with Babel in Spanish environments.
Fix bug reported by Knut Lickert: index entries were unconditionally built.

v2.0 New feature: note levels.
New feature: FiXme note counters and usage summary.
Suggestions from Kasper B. Graversen <kbg@dkik.dk>.
Support for Spanish thanks to Agustín Martín <agusmba@terra.es>

v1.5 New appearance option: `inline`.

v1.4 Support for the koma-script classes.
Fix bug reported by Ulf Jaenicke-Roessler: the `\listoffixmes` command didn't work when called before the first FiXme note.

v1.3 Support for Italian thanks to Riccardo Murri <murri@phc.unipi.it>.

v1.2 Support for German thanks to Harald Harders <h.harders@tu-bs.de>.

Well, I think that's it. Enjoy using FiXme!