NAME

pfm − Personal File Manager for Linux/Unix

SYNOPSIS

pfm [−l, −−layout *number*] [*directory*] [−s, −−swap *directory*]

pfm { −v, −−version | −h, −−help }

DESCRIPTION

pfm is a terminal-based file manager, based on PFM.COM for MS-DOS.

All pfm commands are accessible through one or two keystrokes, and a few are accessible with the mouse. Most command keys are case-insensitive. pfm can operate in single-file mode or multiple-file mode. In single-file mode, the command corresponding to the keystroke will be performed on the current (highlighted) file only. In multiple-file mode, the command will apply to a selection of files.

Note that throughout this manual page, *file* can mean any type of file, not just plain regular files. These will be referred to as *regular files*.

OPTIONS

Most of pfm's configuration is read from a config file. The default location for this file is *$HOME/.pfm/.pfmrc*, but an alternative location may be specified using the environment variable PFMRC. If there is no config file present at startup, one will be created. The file contains many comments on the available options, and is therefore supposed to be self-explanatory. pfm will issue a warning if the config file version is older than the version of pfm you are running. In this case, please let pfm create a new default config file and compare the changes with your own settings, so that you do not miss any new config options or format changes. See also the **C**onfig command under MORE COMMANDS below, and DIAGNOSIS.

There are two commandline options that specify starting directories. The CDPATH environment variable is taken into account when pfm tries to find these directories.

directory
: The directory that pfm should initially use as its main directory. If unspecified, the current directory is used.

−h, −−help
: Print usage information, then exit.

−l, −−layout *number*
: Start pfm using the specified column layout (as defined in the *.pfmrc*).

−s, −−swap *directory*
: The directory that pfm should initially use as swap directory. (See also the **F7** command below).

  There would be no point in setting the swap directory and subsequently returning to the main directory if 'persistentswap' is turned off in your config file. Therefore, pfm will swap back to the main directory *only* if 'persistentswap' is turned on.

−v, −−version
: Print current version, then exit.

NAVIGATION

Navigation through directories is essentially done using the arrow keys and the *vi* (1) cursor keys (**hjkl**). The following additional navigation keys are available:

Movement inside a directory:

up arrow, *down arrow*	move the cursor by one line
k, **j**	move the cursor by one line
-, **+**	move the cursor by ten lines
CTRL-E, **CTRL-Y**	scroll the screen by one line
CTRL-U, **CTRL-D**	move the cursor by half a page

CTRL-B, **CTRL-F**	move the cursor by a full page
PgUp, **PgDn**	move the cursor by a full page
HOME, **END**	move the cursor to the top or bottom line
SPACE	mark the current file, then move the cursor one line down

Movement between directories:

right arrow, **l**	*chdir()* to a subdirectory
left arrow, **h**	*chdir()* to the parent directory
ENTER	*chdir()* to a subdirectory
ESC, **BS**	*chdir()* to the parent directory

If the option 'chdirautocmd' has been specified in the *.pfmrc* file, pfm will execute that command after every *chdir()*.

Note 1: the **l** and **ENTER** keys function differently when the cursor is on a non-directory file (see below under **L**ink and LAUNCHING FILES respectively).

Note 2: see below under BUGS on the functioning of **ESC**.

COMMANDS

Attrib

Changes the mode of the file if you are the owner. The mode may be specified either symbolically or numerically, see *chmod* (1) for more details.

Note 1: the mode on a symbolic link cannot be set. See *chmod* (1) for more details.

Note 2: the name **Attrib** for this command is a reminiscence of the DOS version.

Copy

Copy current file. You will be prompted for the destination filename. Directories will be copied recursively with all underlying files.

In multiple-file mode, it is not allowed to specify a single non-directory filename as a destination. Instead, the destination name must be a directory or a name containing a **=1**, **=2** or **=7** escape (see below under c**O**mmand).

If clobber mode is off (see below under the **!** command), existing files will not be overwritten unless the action is confirmed by the user.

Delete

Delete a file or directory. You must confirm this command with **Y** to actually delete the file. If the current file is a directory which contains files, and you want to delete it recursively, you must respond with **A**ffirmative to the additional prompt. Lost files (files on the screen but not actually present on disk) can be deleted from the screen listing without confirmation. Whiteouts cannot be deleted; use un**W**hiteout for this purpose.

Edit

Edit a file with your external editor. You can specify an editor with the environment variable VISUAL or EDITOR or with the 'editor' option in the *.pfmrc* file. Otherwise *vi* (1) is used.

Find

If the current sort mode is by filename, you are prompted for a (partial) filename. While you type, the cursor is positioned on the best match. Type ENTER to end typing.

If the current sort mode is not by filename, then you are prompted for a filename. The cursor is then positioned on that file.

tarGet

Allows you to change the target that a symbolic link points to. You must have permission to remove the current symbolic link.

Include

Allows you to mark a group of files which meet a certain criterion:

After / **B**efore
: files newer/older than a specified date and time

Every file
: all files, including dotfiles, except for the **.** and **..** entries

Files only
: regular files of which the filenames match a specified regular expression (not a glob pattern!)

Oldmarks
: files which were previously marked and are now denoted with an *oldmark* (**.**).

User
: files owned by the current user

Oldmarks may be used to perform more than one command on a group of files.

Link

Prompts to create either:

an **A**bsolute symlink
: This will create a symlink containing an absolute path to the target, irrespective of whether you enter a relative or an absolute symlink name.

  Example: when the cursor is on the file */home/rene/incoming/.plan*, and you request an absolute symlink to be made with either the name *../.plan* or */home/rene/.plan*, the actual symlink will become:

```
/home/rene/.plan -> /home/rene/incoming/.plan
```

a **H**ard link
: This will create an additional hard link to the current file with the specified name, which must be on the same filesystem.

a **R**elative symlink
: This will create a symlink containing a relative path to the target, irrespective of whether you enter a relative or an absolute symlink name.

  Example: when the cursor is on the file */home/rene/incoming/.plan*, and you request a relative symlink to be made with either the name *../.plan* or */home/rene/.plan*, the actual symlink will become:

```
/home/rene/.plan -> incoming/.plan
```

If a directory is specified, `pfm` will follow the behavior of *ln* (1), which is to create the new link inside that directory.

In multiple-file mode, it is not allowed to specify a single non-directory filename as a new name. Instead, the new name must be a directory or a name containing a **=1**, **=2** or **=7** escape (see below under c**O**mmand).

If clobber mode is off (see below under the **!** command), existing files will not be overwritten.

Note that if the current file is a directory, the **l** key, being one of the *vi* (1) cursor keys, will *chdir()* you into the directory. The capital **L** command will *always* try to make a link.

More

Presents you with a choice of operations not related to the current files. Use this to configure `pfm`, edit a new file, make a new directory, show a different directory, kill all child processes, or write the history files to disk. See below under MORE COMMANDS. Pressing **ESC** will take you back to the main menu.

Name

Shows the complete long filename. For a symbolic link, this command will also show the target of the symbolic link. This is useful in case the terminal is not wide enough to display the entire name, or if the name contains non-printable characters. Non-ASCII characters and control characters will be displayed as their octal or hexadecimal equivalents like the examples in the following table. Spaces will be converted as well, if the 'translatespace' option is turned on in the *.pfmrc* file. When the name is shown in its converted form, pressing ***** will change the radix. The 'defaultradix' option specifies the initial radix that will be used.

Examples:

character	representation in radix	
	octal	hexadecimal
CTRL-A	\001	\0x01
space	\040	\0x20
c cedilla (**ç**)	\347	\0xe7
backslash (\)	\\	\\

cOmmand

Allows execution of a shell command. After the command completes, `pfm` will resume.

On the commandline, you may use several special abbreviations, which will be replaced by `pfm` with the current filename, directoryname etc. (see below). These abbreviations start with an escape character. This escape character is defined with the option 'escapechar' in your *.pfmrc* file. The default is **=**. Previous versions of `pfm` used \, but this was deemed too confusing because backslashes are parsed by the shell as well. This manual page (and the default config file) will assume you are using **=** as 'escapechar'.

The following abbreviations are available:

=1 the current filename without extension (see below)

=2 the current filename, complete

=3 the full current directory path

=4 the mountpoint of the current filesystem

=5 the full swap directory path (see **F7** command)

=6 the basename of the current directory

=7 the extension of the current filename (see below)

=8 a space-separated list of all selected filenames

== a single literal =

=e the editor specified with the 'editor' option in the config file

=p the pager specified with the 'pager' option in the config file

=v the image viewer specified with the 'viewer' option in the config file

The *extension* of the filename is defined as follows:

If the filename does not contain a period at all, then the file has no extension (**=7** is empty) and its whole name is regarded as **=1**.

If the filename does contain a period, the extension **=7** is defined as the final part of the filename, starting at the last period in the name. The filename **=1** is the part before the period.

In all cases, the concatenation of **=1** and **=7** is equal to **=2**.

Examples:

=2	**=1**	**=7**
track01.wav	track01	.wav
garden.jpg	garden	.jpg
end.	end	.
somename	somename	*empty*
.profile	*empty*	.profile
.profile.old	.profile	.old

See also below under QUOTING RULES.

Print

Will prompt for a print command (default `lpr -P$PRINTER =2`, or `lpr =2` if `PRINTER` is unset) and will run it. No formatting is done. You may specify a print command with the 'printcmd' option in the *.pfmrc* file.

Quit

Exit `pfm`. The option 'confirmquit' in the *.pfmrc* file specifies whether `pfm` should ask for confirmation. Note that by pressing a capital **Q** (quick quit), you will *never* be asked for confirmation.

Rename

Change the name of the file and/or move it into another directory. You will be prompted for the new filename. Depending on your Unix implementation, a pathname on another filesystem may or may not be allowed.

In multiple-file mode, it is not allowed to specify a single non-directory filename as a new name. Instead, the new name must be a directory or a name containing a **=1** or **=2** escape (see above under c**O**mmand).

If clobber mode is off (see below under the **!** command), existing files will not be overwritten unless the action is confirmed by the user.

Show

Displays the contents of the current file or directory on screen. You can choose which pager to use for file viewing with the environment variable `PAGER`, or with the 'pager' option in the *.pfmrc* file.

Time

Change mtime (modification date/time) of the file. The time may be entered either with or without clarifying interpunction (e.g. 2008−12−04 08:42.12) as the interpunction will be removed to obtain a format which *touch* (1) can use. Enter **.** to set the mtime to the current date and time.

Uid

Change ownership of a file. Note that many Unix variants do not allow normal (non−`root`) users to change ownership. Symbolic links will be followed.

sVn

Updates the current file with Subversion status information. The command to use can be configured in the *.pfmrc* with the 'rcscmd' option. See also **M**ore − s**V**n.

unWhiteout

(Only on platforms that support whiteout files). Provides the option to remove the whiteout entry in the top layer of a translucent (tfs), inheriting (ifs) or union (unionfs) filesystem, thereby restoring access to the corresponding file in the lower layer.

eXclude

Allows you to erase marks on a group of files which meet a certain criterion. See **I**nclude for details.

Your command

Like c**O**mmand (see above), except that it uses one-letter commands (case-sensitive) that have been preconfigured in the *.pfmrc* file. **Y**our commands may use **=1** up to **=8** and **=e**, **=p** and **=v** escapes just as in c**O**mmand, e.g.

```
your[c]:tar cvf - =2 | gzip > =2.tar.gz
your[t]:tar tvf =2 | =p
your[o]:svn commit =8
```

siZe

For directories, reports the grand total (in bytes) of the directory and its contents.

For other file types, reports the total number of bytes in allocated data blocks. For regular files, this is often more than the reported file size. For special files and *fast symbolic links*, the number is zero, as no data blocks are allocated for these file types.

If the screen layout (selected with **F9**) contains a 'grand total' column, that column will be used. Otherwise, the 'filesize' column will temporarily be (mis)used. A 'grand total' column in the layout will never be filled in when entering the directory.

Note: since *du* (1) commands are not portable, `pfm` guesses how it can calculate the size according to the Unix variant that it runs on. If `pfm` guesses this incorrectly, you might have to specify the `du` command (or `du | awk` combination) applicable for your Unix version in the *.pfmrc* file. Examples are provided. Please notify the author if you know any corrections that should be made.

MORE COMMANDS

These commands are accessible through the main screen **M**ore command.

Bookmark

This command will push the current directory onto the path history. With the **M**ore − **S**how command, it can be recalled using the up-arrow key.

Config pfm

This command will open the *.pfmrc* config file with your preferred editor. The file will be re-read by `pfm` after you exit your editor. Options that are only modifiable through the config file (like 'columnlayouts') will be reinitialized immediately, options that affect settings modifiable by key commands (like 'defaultsortmode') will not.

Edit new file

You will be prompted for the new filename, then your editor will be spawned.

make Fifo

Prompts for a name, then creates a FIFO file (named pipe) with that name. See also *fifo* (4) and *mkfifo* (1).

sHell

Spawns your default login shell. When you exit from it, `pfm` will resume.

Kill children

Lists available signals. After selection of a signal, sends this signal to all child processes of `pfm` (more accurately: all processes in the same process group).

Make new directory

Specify a new directory name and `pfm` will create it for you. Furthermore, if you don't have any files marked, your current directory will be set to the newly created directory.

Show directory

You will be asked for the directory you want to view. Note that this command is different from **F7** because this will not change your current swap directory status.

sVn

Updates the current directory with Subversion status information. If you set the 'autorcs' option in your *.pfmrc*, this will automatically be done every time `pfm` shows directory contents.

Write history

`pfm` uses the readline library for keeping track of the Unix commands, pathnames, regular expressions, modification times, and file modes entered. The history is read from individual files in *$HOME/.pfm/* every time `pfm` starts. The history is written only when this command is given, or when

pfm exits and the 'autowritehistory' option is set in *.pfmrc*.

MISCELLANEOUS and FUNCTION KEYS

ENTER

If the current file is a directory, pfm will *chdir()* to that directory. Otherwise, pfm will attempt to *launch* the file. See LAUNCHING FILES below.

DEL

Identical to the **D**elete command (see above).

! Toggle clobber mode. This controls whether a file should be overwritten when its name is reused in **C**opy, **L**ink or **R**ename.

" Toggle pathname handling. In **physical** mode, the current directory path will always be transformed to its canonical form (the simplest form, with symbolic names resolved). In **logical** mode, all symbolic link components in the current directory path will be preserved.

% Toggle show/hide whiteout files.

***** Toggle the radix used by the **N**ame command.

. Toggle show/hide dot files.

/ Identical to **F**ind (see above).

< Scroll the header and footer, in order to view all available commands.

= Cycle through displaying the username, the hostname, or username@hostname.

> Scroll the header and footer, in order to view all available commands.

? Display help. Identical to **F1**.

@ Allows the user to enter a perl command to be executed in the context of pfm. Primarily used for debugging.

F1 Display help, version number and license information.

F2 *chdir()* back to the previous directory.

F3 Fit the file list into the current window and refresh the display.

F4 Change the current colorset. Multiple colorsets may be defined, see the *.pfmrc* file itself for details.

F5 Current directory will be reread. Use this when the contents of the directory have changed. This command will erase all marks.

F6 Allows you to re-sort the directory listing. You will be presented a number of sort modes.

F7 Alternates the display between two directories. When switching for the first time, you are prompted for a directory path to show. When you switch back by pressing **F7** again, the contents of the alternate directory are displayed unchanged. Header text changes color when in swap screen. In shell commands, the directory path from the alternate screen may be referred to as **=5**. If the 'persistentswap' option has been set in the config file, then leaving the swap mode will store the main directory path as swap path again.

F8 Toggles the mark (include flag) on an individual file.

F9 Toggle the column layout. Layouts are defined in your *.pfmrc*, in the 'defaultlayout' and 'columnlayouts' options. See the config file itself for information on changing the column layout.

Note that a 'grand total' column in the layout will only be filled when the si**Z**e command is issued, not when reading the directory contents.

F10

Switch between single-file and multiple-file mode.

F11
Refresh (using *lstat* (2)) the displayed file data for the current file.

F12
Toggle mouse use. See below under MOUSE COMMANDS.

LAUNCHING FILES

The **ENTER** key, when used on a non-directory file, will attempt to launch the file.

The command used for launching a file is determined by the file type. File types are identified by a unique name, preferably MIME type names. Launch commands for every file type may be defined using the config file 'launch[*filetype*]' options.

Example:

```
launch[image/gif]      :=v =2 &
launch[application/pdf]:acroread =2 &
```

There are three methods for determining the file type. You may opt to use one, two, or all three of these methods, thereby using the second and third method as fallback.

The following methods are available:

extension
The filename extension will be translated to a file type using the 'extension[**.extension*]' options in the config file.

Example:

```
extension[*.gif]:image/gif
extension[*.pdf]:application/pdf
```

magic
The *file* (1) command will be run on the current file. Its output will be translated to a file type using the 'magic[*regular expression*]' options in the config file.

Example:

```
magic[GIF image data]:image/gif
magic[PDF document]  :application/pdf
```

xbit
The executable bits in the file permissions will be checked (after symbolic links have been followed). If the current file is executable, `pfm` will attempt to start the file as an executable command.

To select which method or methods (*extension*, *magic*, and/or *xbit*) should be used for determining the file type, you should specify these using the 'launchby' option (separated by commas if more than one).

Example:

```
launchby:xbit,extension
```

If the file type cannot be determined, the current file will be displayed using your pager.

The **ENTER** key will always behave as if `pfm` runs in single-file mode. It will *not* launch multiple files. Use **Y**our or c**O**mmand to launch multiple files.

QUOTING RULES

`pfm` adds an extra layer of parsing to filenames and shell commands. It is important to take notice of the rules that `pfm` uses.

The following six types of input can be distinguished:

a regular expression (only the **I**nclude and e**X**clude commands)
The input is parsed as a regular expression.

a time (e.g. the **T**ime or **I**nclude − **B**efore commands)
Characters not in the set `[0-9.]` are removed from the input.

a literal pattern (only the **F**ind command)
The input is taken literally.

not a filename or shell command (e.g. in **A**ttribute or **U**id)
The input is taken literally.

a filename (e.g. in **C**opy or tar**G**et).
First of all, tilde expansion is performed.

Next, any `=[1-8evp]` character sequence is expanded to the corresponding value.

At the same time, any `=[^1-8evp]` character sequence is just replaced with the character itself.

Finally, if the filename is to be processed by `pfm`, it is taken literally; if it is to be handed over to a shell, all metacharacters are replaced *escaped*.

a shell command (e.g. in c**O**mmand or **P**rint)
First of all, tilde expansion is performed.

Next, any `=[1-8evp]` character sequence is expanded to the corresponding value, *with shell metacharacters escaped*.

At the same time, any `=[^1-8evp]` character sequence is just replaced with the character itself.

In short:

- `pfm` always escapes shell metacharacters in expanded **=2** *etc.* constructs.
- In filenames entered, shell metacharacters are taken literally.
- In shell commands entered, metacharacters that you want to be taken literally must be escaped one extra time.

Examples:

char(s) wanted in filename	char(s) to type in filename	char(s) to type in shell command
any non-metachar	*that char*	*that char*
\	\	\\ **or** '\'
"	"	\" **or** '"'
=	==	==
space	*space*	*space* **or** '*space*'
filename	=2	=2
\2	\2	\\2 **or** '\2'
=2	==2	==2

MOUSE COMMANDS

When `pfm` is run in an xterm or other terminal that supports the use of a mouse, turning on mouse mode (either initially with the 'defaultmousemode' option in the *.pfmrc* file, or while running using the **F12** key) will give mouse access to the following commands:

button	location clicked					
	pathline	title/ header	footer	fileline	filename	dirname
1	*chdir()*	CTRL-U	CTRL-D	F8	**S**how	**S**how
2	c**O**mmand	PgUp	PgDn	**S**how	ENTER	*new win*
3	c**O**mmand	PgUp	PgDn	**S**how	ENTER	*new win*
up	*three lines up*					
down	*three lines down*					

The cursor will *only* be moved when the title, header or footer is clicked, or when changing directory. The mouse wheel also works and moves the cursor three lines per notch, or one line if shift is pressed.

Clicking button 1 on the current directory path will *chdir()* up to the clicked ancestor directory. If the current directory was clicked, or the device name, it will act like a **M**ore − **S**how command.

Clicking button 2 on a directory name will open a new pfm terminal window.

Mouse use will be turned off during the execution of commands, unless 'mouseturnoff' is set to 'no' in *.pfmrc*. Note that setting this to 'no' means that your (external) commands (like your pager and editor) will receive escape codes when the mouse is clicked.

WORKING DIRECTORY INHERITANCE

Upon exit, `pfm` will save its current working directory in the file *$HOME/.pfm/cwd*, and its swap directory, if any, in *$HOME/.pfm/swd*. This enables the user to have the calling process (shell) ''inherit'' `pfm`'s current working directory, and to reinstate the swap directory upon the next invocation. To achieve this, you may call `pfm` using a function or alias like the following:

Example for *ksh* (1), *bash* (1) and *zsh* (1):

```
pfm() {
    if [ -s ~/.pfm/swd ]; then
        swd=-s"`cat ~/.pfm/swd`"
    fi
    # providing $swd is optional
    env pfm $swd "$@"
    if [ -s ~/.pfm/cwd ]; then
        cd "`cat ~/.pfm/cwd`"
        rm -f ~/.pfm/cwd
    fi
}
```

Example for *csh* (1) and *tcsh* (1):

```
alias pfm ':                                \
if (-s ~/.pfm/swd) then                     \
    set swd=-s"`cat ~/.pfm/swd`"            \
endif                                       \
: providing $swd is optional                \
env pfm $swd \!*                            \
if (-s ~/.pfm/cwd) then                     \
    cd "`cat ~/.pfm/cwd`"                   \
    rm -f ~/.pfm/cwd                        \
endif'
```

ENVIRONMENT

ANSI_COLORS_DISABLED

Detected as an indication that ANSI coloring escape sequences should not be used.

CDPATH

A colon-separated list of directories specifying the search path when changing directories. There is always an implicit **.** entry at the start of this search path.

EDITOR

The editor to be used for the **E**dit command. Overridden by VISUAL.

LC_ALL
LC_COLLATE
LC_CTYPE
LC_MESSAGES

LC_NUMERIC
LC_TIME
LANG
Determine locale settings, most notably for collation sequence, messages and date/time format. See *locale* (7).

PAGER
Identifies the pager with which to view text files. Defaults to *less* (1) for Linux systems or *more* (1) for Unix systems.

PERL_RL
Indicate whether and how the readline prompts should be highlighted. See *Term::ReadLine* (3pm). If unset, a good guess is made based on your config file 'framecolors[]' setting.

PFMRC
Specify a location of an alternate *.pfmrc* file. If unset, the default location *$HOME/.pfm/.pfmrc* is used. The cwd− and history-files cannot be displaced in this manner, and will always be located in the directory *$HOME/.pfm/*.

PRINTER
May be used to specify a printer to print to using the **P**rint command.

SHELL
Your default login shell, spawned by **M**ore − s**H**ell.

VISUAL
The editor to be used for the **E**dit command. Overrides EDITOR.

FILES

The directory *$HOME/.pfm/* and files therein. A number of input histories and the current working directory on exit are saved to this directory.

The default location for the config file is *$HOME/.pfm/.pfmrc*.

DIAGNOSIS

If `pfm` reports that your config file might be outdated, you might be missing some of the newer configuration options (or default values for these). Try the following command and compare the new config file with your original one:

```
env PFMRC=~/.pfm/.pfmrc-new pfm
```

To prevent the warning from occurring again, update the '## Version' line.

BUGS and WARNINGS

When typed by itself, the **ESC** key needs to be pressed twice. This is due to the lack of a proper timeout in `Term::Screen`.

`Term::ReadLine::Gnu` does not allow a half-finished line to be aborted by pressing **ESC**. For most commands, you will need to clear the half-finished line. You may use the terminal kill character (usually **CTRL-U**) for this (see *stty* (1)).

The author once almost pressed **ENTER** when logged in as root and with the cursor on the file */sbin/reboot*. You have been warned.

The smallest terminal size supported is 80x24. The display will be messed up if you resize your terminal window to a smaller size.

VERSION

This manual pertains to `pfm` version 2.01.3.

AUTHOR and COPYRIGHT

This program was based on PFM.COM version 2.32, originally written for MS-DOS by Paul R. Culley and Henk de Heer. The name 'pfm' was adopted with kind permission of the original authors.

SEE ALSO

The documentation on PFM.COM. The manual pages for *chmod* (1), *file* (1), *less* (1), *locale* (7), *lpr* (1), *touch* (1), *vi* (1).

For programmers: *Term::Screen* (3pm), *Term::ScreenColor* (3pm), *Term::ReadLine* (3pm), *PFM::Abstract* (3pm), *PFM::Application* (3pm), *PFM::Browser* (3pm), *PFM::CommandHandler* (3pm), *PFM::Config* (3pm), *PFM::Directory* (3pm), *PFM::History* (3pm), *PFM::Job* (3pm), *PFM::Screen* (3pm), *PFM::State* (3pm) and *PFM::Util* (3pm).

The pfm project page: http://sourceforge.net/projects/p−f−m/