

**NAME**

`mc` – Visual shell for Unix-like systems.

**USAGE**

`mc` [-abcCdfhPstuUVx] [-l log] [dir1 [dir2]] [-e [file]] [-v file]

**DESCRIPTION**

GNU Midnight Commander is a directory browser/file manager for Unix-like operating systems.

**OPTIONS**

- `-a, --stickchars`  
Disable usage of graphic characters for line drawing.
- `-b, --nocolor`  
Force black and white display.
- `-c, --color`  
Force color mode, please check the section Colors for more information.
- `-C arg, --colors=arg`  
Specify a different color set in the command line. The format of `arg` is documented in the Colors section.
- `-d, --nomouse`  
Disable mouse support.
- `-e [file], --edit[=file]`  
Start the internal editor. If the file is specified, open it on startup. See also **mcedit (1)**.
- `-f, --datadir`  
Display the compiled-in search paths for Midnight Commander files.
- `-k, --resetsoft`  
Reset softkeys to their default from the termcap/terminfo database. Only useful on HP terminals when the function keys don't work.
- `-l file, --ftplib=file`  
Save the ftpfs dialog with the server in file.
- `-P file, --printwd=file`  
Print the last working directory to the specified file. This option is not meant to be used directly. Instead, it's used from a special shell script that automatically changes the current directory of the shell to the last directory the Midnight Commander was in. Source the file `/usr/share/mc/bin/mc.sh` (bash and zsh users) or `/usr/share/mc/bin/mc.csh` (tcsh users) respectively to define `mc` as an alias to the appropriate shell script.
- `-s, --slow`  
Turn on the slow terminal mode, in this mode the program will not draw expensive line drawing characters and will toggle verbose mode off.
- `-t, --termcap`  
Used only if the code was compiled with Slang and terminfo: it makes the Midnight Commander use the value of the **TERMCAP** variable for the terminal information instead of the information on the system wide terminal database
- `-u, --nosubshell`  
Disable use of the concurrent shell (only makes sense if the Midnight Commander has been built with concurrent shell support).
- `-U, --subshell`  
Enable use of the concurrent shell support (only makes sense if the Midnight Commander was built with the subshell support set as an optional feature).

- v file, --view=file*  
Start the internal viewer to view the specified file. See also **mcview (1)**.
- V, --version*  
Display the version of the program.
- x, --xterm*  
Force xterm mode. Used when running on xterm-capable terminals (two screen modes, and able to send mouse escape sequences).

If specified, the first path name is the directory to show in the selected panel; the second path name is the directory to be shown in the other panel.

## Overview

The screen of the Midnight Commander is divided into four parts. Almost all of the screen space is taken up by two directory panels. By default, the second line from the bottom of the screen is the shell command line, and the bottom line shows the function key labels. The topmost line is the menu bar line. The menu bar line may not be visible, but appears if you click the topmost line with the mouse or press the F9 key.

The Midnight Commander provides a view of two directories at the same time. One of the panels is the current panel (a selection bar is in the current panel). Almost all operations take place on the current panel. Some file operations like Rename and Copy by default use the directory of the unselected panel as a destination (don't worry, they always ask you for confirmation first). For more information, see the sections on the Directory Panels, the Left and Right Menus and the File Menu.

You can execute system commands from the Midnight Commander by simply typing them. Everything you type will appear on the shell command line, and when you press Enter the Midnight Commander will execute the command line you typed; read the Shell Command Line and Input Line Keys sections to learn more about the command line.

## Mouse Support

The Midnight Commander comes with mouse support. It is activated whenever you are running on an **xterm(1)** terminal (it even works if you take a telnet, ssh or rlogin connection to another machine from the xterm) or if you are running on a Linux console and have the **gpm** mouse server running.

When you left click on a file in the directory panels, that file is selected; if you click with the right button, the file is marked (or unmarked, depending on the previous state).

Double-clicking on a file will try to execute the command if it is an executable program; and if the extension file has a program specified for the file's extension, the specified program is executed.

Also, it is possible to execute the commands assigned to the function key labels by clicking on them.

If a mouse button is clicked on the top frame line of the directory panel, it is scrolled one page up. Likewise, a click on the bottom frame line will cause scrolling one page down. This frame line method works also in the Help Viewer and the Directory Tree.

The default auto repeat rate for the mouse buttons is 400 milliseconds. This may be changed to other values by editing the `~/.mc/ini` file and changing the `mouse_repeat_rate` parameter.

If you are running the Midnight Commander with the mouse support, you can get the default mouse behavior (cutting and pasting text) by holding down the Shift key.

## Keys

Some commands in the Midnight Commander involve the use of the *Control* (sometimes labeled CTRL or CTL) and the *Meta* (sometimes labeled ALT or even Compose) keys. In this manual we will use the following abbreviations:

**C-`<chr>`**

means hold the Control key while typing the character `<chr>`. Thus C-f would be: hold the Control key and type f.

**M-`<chr>`**

means hold the Meta or Alt key down while typing `<chr>`. If there is no Meta or Alt key, type *ESC*, release it, then type the character `<chr>`.

**S-`<chr>`**

means hold the Shift key down while typing `<chr>`.

All input lines in the Midnight Commander use an approximation to the GNU Emacs editor's key bindings.

There are many sections which tell about the keys. The following are the most important.

The File Menu section documents the keyboard shortcuts for the commands appearing in the File menu. This section includes the function keys. Most of these commands perform some action, usually on the selected file or the tagged files.

The Directory Panels section documents the keys which select a file or tag files as a target for a later action (the action is usually one from the file menu).

The Shell Command Line section list the keys which are used for entering and editing command lines. Most of these copy file names and such from the directory panels to the command line (to avoid excessive typing) or access the command line history.

Input Line Keys are used for editing input lines. This means both the command line and the input lines in the query dialogs.

**Miscellaneous Keys**

Here are some keys which don't fall into any of the other categories:

**Enter** if there is some text in the command line (the one at the bottom of the panels), then that command is executed. If there is no text in the command line then if the selection bar is over a directory the Midnight Commander does a **chdir(2)** to the selected directory and reloads the information on the panel; if the selection is an executable file then it is executed. Finally, if the extension of the selected file name matches one of the extensions in the extensions file then the corresponding command is executed.

**C-l** repaint all the information in the Midnight Commander.

**C-x c** run the Chmod command on a file or on the tagged files.

**C-x o** run the Chown command on the current file or on the tagged files.

**C-x l** run the link command.

**C-x s** run the symbolic link command.

**C-x i** set the other panel display mode to information.

**C-x q** set the other panel display mode to quick view.

**C-x !** execute the External panelize command.

**C-x h** run the add directory to hotlist command.

**M-!** executes the Filtered view command, described in the view command.

**M-?** executes the Find file command.

**M-c** pops up the quick cd dialog.

**C-o** when the program is being run in the Linux or FreeBSD console or under an xterm, it will show you the output of the previous command. When ran on the Linux console, the Midnight Commander uses an external program (cons.saver) to handle saving and restoring of information on the screen.

When the subshell support is compiled in, you can type C-o at any time and you will be taken back to the Midnight Commander main screen, to return to your application just type C-o. If you have an application suspended by using this trick, you won't be able to execute other programs from the Midnight Commander until you terminate the suspended application.

### Directory Panels

This section lists the keys which operate on the directory panels. If you want to know how to change the appearance of the panels take a look at the section on Left and Right Menus.

#### Tab, C-i

change the current panel. The old other panel becomes the new current panel and the old current panel becomes the new other panel. The selection bar moves from the old current panel to the new current panel.

#### Insert, C-t

to tag files you may use the Insert key (the kich1 terminfo sequence) or the C-t (Control-t) sequence. To untag files, just retag a tagged file.

#### M-g, M-r, M-j

used to select the top file in a panel, the middle file and the bottom one, respectively.

#### C-s, M-s

start a filename search in the directory listing. When the search is active, the user input will be added to the search string instead of the command line. If the *Show mini-status* option is enabled the search string is shown on the mini-status line. When typing, the selection bar will move to the next file starting with the typed letters. The *backspace* or *DEL* keys can be used to correct typing mistakes. If C-s is pressed again, the next match is searched for.

**M-t** toggle the current display listing to show the next display listing mode. With this it is possible to quickly switch from long listing to regular listing and the user defined listing mode.

#### C-\ (control-backslash)

show the directory hotlist and change to the selected directory.

#### + (plus)

this is used to select (tag) a group of files. The Midnight Commander will prompt for a regular expression describing the group. When *Shell Patterns* are enabled, the regular expression is much like the regular expressions in the shell (\* standing for zero or more characters and ? standing for one character). If *Shell Patterns* is off, then the tagging of files is done with normal regular expressions (see ed (1)).

If the expression starts or ends with a slash (/), then it will select directories instead of files.

#### \ (backslash)

use the "\ " key to unselect a group of files. This is the opposite of the Plus key.

#### up-key, C-p

move the selection bar to the previous entry in the panel.

#### down-key, C-n

move the selection bar to the next entry in the panel.

#### home, a1, M-<

move the selection bar to the first entry in the panel.

#### end, c1, M->

move the selection bar to the last entry in the panel.

#### next-page, C-v

move the selection bar one page down.

**prev-page, M-v**

move the selection bar one page up.

**M-o** make the current directory of the current panel also the current directory of the other panel. Put the other panel to the listing mode if needed. If the current panel is panelized, the other panel doesn't become panelized.

**C-PageUp, C-PageDown**

only when supported by the terminal: change to ".." and to the currently selected directory respectively.

**M-y** moves to the previous directory in the history, equivalent to clicking the < with the mouse.

**M-u** moves to the next directory in the history, equivalent to clicking the > with the mouse.

**M-S-h, M-H**

displays the directory history, equivalent to depressing the 'v' with the mouse.

**Shell Command Line**

This section lists keys which are useful to avoid excessive typing when entering shell commands.

**M-Enter**

copy the currently selected file name to the command line.

**C-Enter**

same as M-Enter. May not work on remote systems and some terminals.

**C-S-Enter**

copy the full path name of the currently selected file to the command line. May not work on remote systems and some terminals.

**M-Tab**

does the filename, command, variable, username and hostname completion for you.

**C-x t, C-x C-t**

copy the tagged files (or if there are no tagged files, the selected file) of the current panel (C-x t) or of the other panel (C-x C-t) to the command line.

**C-x p, C-x C-p**

the first key sequence copies the current path name to the command line, and the second one copies the unselected panel's path name to the command line.

**C-q** the quote command can be used to insert characters that are otherwise interpreted by the Midnight Commander (like the '+' symbol)

**M-p, M-n**

use these keys to browse through the command history. M-p takes you to the last entry, M-n takes you to the next one.

**M-h** displays the history for the current input line.

**General Movement Keys**

The help viewer, the file viewer and the directory tree use common code to handle moving. Therefore they accept exactly the same keys. Each of them also accepts some keys of its own.

Other parts of the Midnight Commander use some of the same movement keys, so this section may be of use for those parts too.

**Up, C-p**

moves one line backward.

**Down, C-n**

moves one line forward.

**Prev Page, Page Up, M-v**  
moves one page up.

**Next Page, Page Down, C-v**  
moves one page down.

**Home, A1**  
moves to the beginning.

**End, C1**  
move to the end.

The help viewer and the file viewer accept the following keys in addition the to ones mentioned above:

**b, C-b, C-h, Backspace, Delete**  
moves one page up.

**Space bar**  
moves one page down.

**u, d** moves one half of a page up or down.

**g, G** moves to the beginning or to the end.

### Input Line Keys

The input lines (they are used for the command line and for the query dialogs in the program) accept these keys:

**C-a** puts the cursor at the beginning of line.

**C-e** puts the cursor at the end of the line.

**C-b, move-left**  
move the cursor one position left.

**C-f, move-right**  
move the cursor one position right.

**M-f** moves one word forward.

**M-b** moves one word backward.

**C-h, backspace**  
delete the previous character.

**C-d, Delete**  
delete the character in the point (over the cursor).

**C-@** sets the mark for cutting.

**C-w** copies the text between the cursor and the mark to a kill buffer and removes the text from the input line.

**M-w** copies the text between the cursor and the mark to a kill buffer.

**C-y** yanks back the contents of the kill buffer.

**C-k** kills the text from the cursor to the end of the line.

**M-p, M-n**  
Use these keys to browse through the command history. M-p takes you to the last entry, M-n takes you to the next one.

**M-C-h, M-Backspace**  
delete one word backward.

**M-Tab**  
does the filename, command, variable, username and hostname completion for you.

## Menu Bar

The menu bar pops up when you press F9 or click the mouse on the top row of the screen. The menu bar has five menus: "Left", "File", "Command", "Options" and "Right".

The Left and Right Menus allow you to modify the appearance of the left and right directory panels.

The File Menu lists the actions you can perform on the currently selected file or the tagged files.

The Command Menu lists the actions which are more general and bear no relation to the currently selected file or the tagged files.

The Options Menu lists the actions which allow you to customize the Midnight Commander.

## Left and Right (Above and Below) Menus

The outlook of the directory panels can be changed from the **Left** and **Right** menus (they are named **Above** and **Below** when the horizontal panel split is chosen from the Layout options dialog).

## Listing Mode...

The listing mode view is used to display a listing of files, there are four different listing modes available: **Full**, **Brief**, **Long** and **User**. The full directory view shows the file name, the size of the file and the modification time.

The brief view shows only the file name and it has two columns (therefore showing twice as many files as other views). The long view is similar to the output of `ls -l` command. The long view takes the whole screen width.

If you choose the "User" display format, then you have to specify the display format.

The user display format must start with a panel size specifier. This may be "half" or "full", and they specify a half screen panel and a full screen panel respectively.

After the panel size, you may specify the two columns mode on the panel, this is done by adding the number "2" to the user format string.

After this you add the name of the fields with an optional size specifier. This are the available fields you may display:

**name** displays the file name.

**size** displays the file size.

**bsize** is an alternative form of the **size** format. It displays the size of the files and for directories it just shows SUB-DIR or UP--DIR.

**type** displays a one character wide type field. This character is similar to what is displayed by `ls` with the `-F` flag - `*` for executable files, `/` for directories, `@` for links, `=` for sockets, `-` for character devices, `+` for block devices, `|` for pipes, `~` for symbolic links to directories and `!` for stale symlinks (links that point nowhere).

**mark** an asterisk if the file is tagged, a space if it's not.

**mtime** file's last modification time.

**atime** file's last access time.

**ctime** file's creation time.

**perm** a string representing the current permission bits of the file.

**mode** an octal value with the current permission bits of the file.

**nlink** the number of links to the file.

**ngid** the GID (numeric).  
**nuid** the UID (numeric).  
**owner** the owner of the file.  
**group** the group of the file.  
**inode** the inode of the file.

Also you can use following keywords to define the panel layout:

**space** a space in the display format.  
| add a vertical line to the display format.

To force one field to a fixed size (a size specifier), you just add **:** followed by the number of characters you want the field to have. If the number is followed by the symbol **+**, then the size specifies the minimal field size - if the program finds out that there is more space on the screen, it will then expand that field.

For example, the **Full** display corresponds to this format:

half type name | size | mtime

And the **Long** display corresponds to this format:

full perm space nlink space owner space group space size space mtime space name

This is a nice user display format:

half name | size:7 | type mode:3

Panels may also be set to the following modes:

**Info** The info view display information related to the currently selected file and if possible information about the current file system.

**Tree** The tree view is quite similar to the directory tree feature. See the section about it for more information.

#### Quick View

In this mode, the panel will switch to a reduced viewer that displays the contents of the currently selected file, if you select the panel (with the tab key or the mouse), you will have access to the usual viewer commands.

#### Sort Order...

The eight sort orders are by name, by extension, by modification time, by access time, and by inode information modification time, by size, by inode and unsorted. In the Sort order dialog box you can choose the sort order and you may also specify if you want to sort in reverse order by checking the reverse box.

By default directories are sorted before files but this can be changed from the Options menu (option **Mix all files**).

#### Filter...

The filter command allows you to specify a shell pattern (for example **\*.tar.gz**) which the files must match to be shown. Regardless of the filter pattern, the directories and the links to directories are always shown in the directory panel.

#### Reread

The reread command reload the list of files in the directory. It is useful if other processes have created or removed files. If you have panelized file names in a panel this will reload the directory contents and remove the panelized information (See the section External panelize for more information).

## File Menu

The Midnight Commander uses the F1 - F10 keys as keyboard shortcuts for commands appearing in the file menu. The escape sequences for the function keys are terminfo capabilities kf1 through kf10. On terminals without function key support, you can achieve the same functionality by pressing the ESC key and then a number in the range 1 through 9 and 0 (corresponding to F1 to F9 and F10 respectively).

The File menu has the following commands (keyboard shortcuts in parentheses):

### Help (F1)

Invokes the built-in hypertext help viewer. Inside the help viewer, you can use the Tab key to select the next link and the Enter key to follow that link. The keys Space and Backspace are used to move forward and backward in a help page. Press F1 again to get the full list of accepted keys.

### Menu (F2)

Invoke the user menu. The user menu provides an easy way to provide users with a menu and add extra features to the Midnight Commander.

### View (F3, Shift-F3)

View the currently selected file. By default this invokes the Internal File Viewer but if the option "Use internal view" is off, it invokes an external file viewer specified by the **PAGER** environment variable. If **PAGER** is undefined, the "view" command is invoked. If you use Shift-F3 instead, the viewer will be invoked without doing any formatting or preprocessing to the file.

### Filtered View (M-!)

This command prompts for a command and its arguments (the argument defaults to the currently selected file name), the output from such command is shown in the internal file viewer.

### Edit (F4)

Currently it invokes the **vi** editor, or the editor specified in the **EDITOR** environment variable, or the Internal File Editor if the use\_internal\_edit option is on.

### Copy (F5)

Pop up an input dialog with destination that defaults to the directory in the non-selected panel and copies the currently selected file (or the tagged files, if there is at least one file tagged) to the directory specified by the user in the input dialog. During this process, you can press C-c or ESC to abort the operation. For details about source mask (which will be usually either \* or ^\(.\*\)\$ depending on setting of Use shell patterns) and possible wildcards in the destination see Mask copy/rename.

On some systems, it is possible to do the copy in the background by clicking on the background button (or pressing M-b in the dialog box). The Background Jobs is used to control the background process.

### Link (C-x l)

Create a hard link to the current file.

### SymLink (C-x s)

Create a symbolic link to the current file. To those of you who don't know what links are: creating a link to a file is a bit like copying the file, but both the source filename and the destination filename represent the same file image. For example, if you edit one of these files, all changes you make will appear in both files. Some people call links aliases or shortcuts.

A hard link appears as a real file. After making it, there is no way of telling which one is the original and which is the link. If you delete either one of them the other one is still intact. It is very difficult to notice that the files represent the same image. Use hard links when you don't even want to know.

A symbolic link is a reference to the name of the original file. If the original file is deleted the

symbolic link is useless. It is quite easy to notice that the files represent the same image. The Midnight Commander shows an "@"-sign in front of the file name if it is a symbolic link to somewhere (except to directory, where it shows a tilde (~)). The original file which the link points to is shown on mini-status line if the *Show mini-status* option is enabled. Use symbolic links when you want to avoid the confusion that can be caused by hard links.

### Rename/Move (F6)

Pop up an input dialog that defaults to the directory in the non-selected panel and moves the currently selected file (or the tagged files if there is at least one tagged file) to the directory specified by the user in the input dialog. During the process, you can press C-c or ESC to abort the operation. For more details look at Copy operation above, most of the things are quite similar.

On some systems, it is possible to do the copy in the background by clicking on the background button (or pressing M-b in the dialog box). The Background Jobs is used to control the background process.

### Mkdir (F7)

Pop up an input dialog and creates the directory specified.

### Delete (F8)

Delete the currently selected file or the tagged files in the currently selected panel. During the process, you can press C-c or ESC to abort the operation.

**Quick cd (M-c)** Use the quick cd command if you have full command line and want to cd somewhere.

### Select group (+)

This is used to select (tag) a group of files. The Midnight Commander will prompt for a regular expression describing the group. When *Shell Patterns* are enabled, the regular expression is much like the filename globbing in the shell (\* standing for zero or more characters and ? standing for one character). If *Shell Patterns* is off, then the tagging of files is done with normal regular expressions (see ed (1)).

To mark directories instead of files, the expression must start or end with a '/'.

### Unselect group (\)

Used to unselect a group of files. This is the opposite of the *Select group* command.

### Quit (F10, Shift-F10)

Terminate the Midnight Commander. Shift-F10 is used when you want to quit and you are using the shell wrapper. Shift-F10 will not take you to the last directory you visited with the Midnight Commander, instead it will stay at the directory where you started the Midnight Commander.

### Quick cd

This command is useful if you have a full command line and want to cd somewhere without having to yank and paste the command line. This command pops up a small dialog, where you enter everything you would enter after **cd** on the command line and then you press enter. This features all the things that are already in the internal cd command.

### Command Menu

The Directory tree command shows a tree figure of the directories.

The Find file command allows you to search for a specific file. The "Swap panels" command swaps the contents of the two directory panels.

The "Panels on/off" command shows the output of the last shell command. This works only on xterm and on Linux and FreeBSD console.

The Compare directories (C-x d) command compares the directory panels with each other. You can then use the Copy (F5) command to make the panels identical. There are three compare methods. The quick method compares only file size and file date. The thorough method makes a

full byte-by-byte compare. The thorough method is not available if the machine does not support the `mmap(2)` system call. The size-only compare method just compares the file sizes and does not check the contents or the date times, it just checks the file size.

The Command history command shows a list of typed commands. The selected command is copied to the command line. The command history can also be accessed by typing `M-p` or `M-n`.

The Directory hotlist (`C-\`) command makes changing of the current directory to often used directories faster.

The External panelize allows you to execute an external program, and make the output of that program the contents of the current panel.

Extension file edit command allows you to specify programs to executed when you try to execute, view, edit and do a bunch of other thing on files with certain extensions (filename endings). The Menu file edit command may be used for editing the user menu (which appears by pressing `F2`).

## Directory Tree

The Directory Tree command shows a tree figure of the directories. You can select a directory from the figure and the Midnight Commander will change to that directory.

There are two ways to invoke the tree. The real directory tree command is available from Commands menu. The other way is to select tree view from the Left or Right menu.

To get rid of long delays the Midnight Commander creates the tree figure by scanning only a small subset of all the directories. If the directory which you want to see is missing, move to its parent directory and press `C-r` (or `F2`).

You can use the following keys:

General movement keys are accepted.

**Enter.** In the directory tree, exits the directory tree and changes to this directory in the current panel. In the tree view, changes to this directory in the other panel and stays in tree view mode in the current panel.

**C-r, F2 (Rescan).** Rescan this directory. Use this when the tree figure is out of date: it is missing subdirectories or shows some subdirectories which don't exist any more.

**F3 (Forget).** Delete this directory from the tree figure. Use this to remove clutter from the figure. If you want the directory back to the tree figure press `F2` in its parent directory.

**F4 (Static/Dynamic).** Toggle between the dynamic navigation mode (default) and the static navigation mode.

In the static navigation mode you can use the `Up` and `Down` keys to select a directory. All known directories are shown.

In the dynamic navigation mode you can use the `Up` and `Down` keys to select a sibling directory, the `Left` key to move to the parent directory, and the `Right` key to move to a child directory. Only the parent, sibling and children directories are shown, others are left out. The tree figure changes dynamically as you traverse.

**F5 (Copy).** Copy the directory.

**F6 (RenMov).** Move the directory.

**F7 (Mkdir).** Make a new directory below this directory.

**F8 (Delete).** Delete this directory from the file system.

**C-s, M-s.** Search the next directory matching the search string. If there is no such directory these keys will move one line down.

**C-h, Backspace.** Delete the last character of the search string.

**Any other character.** Add the character to the search string and move to the next directory which starts with these characters. In the tree view you must first activate the search mode by

pressing C-s. The search string is shown in the mini status line.

The following actions are available only in the directory tree. They aren't supported in the tree view.

**F1 (Help).** Invoke the help viewer and show this section.

**Esc, F10.** Exit the directory tree. Do not change the directory.

The mouse is supported. A double-click behaves like Enter. See also the section on mouse support.

### Find File

The Find File feature first asks for the start directory for the search and the filename to be searched for. By pressing the Tree button you can select the start directory from the directory tree figure.

The contents field accepts regular expressions similar to `egrep(1)`. That means you have to escape characters with a special meaning to `egrep` with `"\"`, e.g. if you search for `"strcmp ("` you will have to input `"strcmp \"` (without the double quotes).

You can start the search by pressing the OK button. During the search you can stop from the Stop button and continue from the Start button.

You can browse the filelist with the up and down arrow keys. The Chdir button will change to the directory of the currently selected file. The Again button will ask for the parameters for a new search. The Quit button quits the search operation. The Panelize button will place the found files to the current directory panel so that you can do additional operations on them (view, copy, move, delete and so on). After panelizing you can press C-r to return to the normal file listing.

It is possible to have a list of directories that the Find File command should skip during the search (for example, you may want to avoid searches on a CD-ROM or on a NFS directory that is mounted across a slow link).

Directories to be skipped should be set on the variable `find_ignore_dirs` in the **Misc** section of your `~/.mc/ini` file.

Directory components should be separated with a colon, here is an example:

```
[Misc]
find_ignore_dirs=/cdrom:/nfs/wuarchive:/afs
```

You may consider using the External panelize command for some operations. Find file command is for simple queries only, while using External panelize you can do as mysterious searches as you would like.

### External panelize

The External panelize allows you to execute an external program, and make the output of that program the contents of the current panel.

For example, if you want to manipulate in one of the panels all the symbolic links in the current directory, you can use external panelization to run the following command:

```
find . -type l -print
```

Upon command completion, the directory contents of the panel will no longer be the directory listing of the current directory, but all the files that are symbolic links.

If you want to panelize all of the files that have been downloaded from your FTP server, you can use this `awk` command to extract the file name from the transfer log files:

```
awk '$9 ~! /incoming/ { print $9 }' < /usr/adm/xferlog
```

You may want to save often used panelize commands under a descriptive name, so that you can recall them quickly. You do this by typing the command on the input line and pressing Add new button. Then you enter a name under which you want the command to be saved. Next time, you just choose that command from the list and do not have to type it again.

## Hotlist

The Directory hotlist command shows the labels of the directories in the directory hotlist. The Midnight Commander will change to the directory corresponding to the selected label. From the hotlist dialog, you can remove already created label/directory pairs and add new ones. To add new directories quickly, you can use the Add to hotlist command (C-x h), which adds the current directory into the directory hotlist, asking just for the label for the directory.

This makes cd to often used directories faster. You may consider using the CDPATH variable as described in internal cd command description.

## Extension File Edit

This will invoke your editor on the file `~/.mc/bindings`. The format of this file following:

All lines starting with # or empty lines are thrown away.

Lines starting in the first column should have following format:

*keyword*/*expr*, i.e. everything after the slash until new line is *expr*.

*keyword* can be:

*shell* — *expr* is an extension (no wildcards). File matches if its name ends with *expr*. Example: *shell/.tar* matches *\*.tar*.

*regex* — *expr* is a regular expression. File matches if its name matches the regular expression.

*directory*

— *expr* is a regular expression. File matches if it is a directory and its name matches the regular expression.

*type* — *expr* is a regular expression. File matches if the output of *file %f* without the initial "filename:" part matches regular expression *expr*.

*default* — matches any file. *expr* is ignored.

*include*

— denotes a common section. *expr* is the name of the section.

Other lines should start with a space or tab and should be of the format: *keyword=command* (with no spaces around =), where *keyword* should be: *Open* (invoked on Enter or double click), *View* (F3), *Edit* (F4) or *Include* (to add rules from the common section). *command* is any one-line shell command, with the simple macro substitution.

Rules are matched from top to bottom, thus the order is important. If the appropriate action is missing, search continues as if this rule didn't match (i.e. if a file matches the first and second entry and View action is missing in the first one, then on pressing F3 the View action from the second entry will be used). *default* should match all the actions.

## Background Jobs

This lets you control the state of any background Midnight Commander process (only copy and move files operations can be done in the background). You can stop, restart and kill a background job from here.

## Menu File Edit

The user menu is a menu of useful actions that can be customized by the user. When you access the user menu, the file `.mc.menu` from the current directory is used if it exists, but only if it is owned by user or root and is not world-writable. If no such file found, `~/.mc/menu` is tried in the same way, and otherwise mc uses the default system-wide menu `/etc/mc/mc.menu`.

The format of the menu file is very simple. Lines that start with anything but space or tab are considered entries for the menu (in order to be able to use it like a hot key, the first character should be a letter). All the lines that start with a space or a tab are the commands that will be executed when the entry is selected.

When an option is selected all the command lines of the option are copied to a temporary file in

the temporary directory (usually /usr/tmp) and then that file is executed. This allows the user to put normal shell constructs in the menus. Also simple macro substitution takes place before executing the menu code. For more information, see macro substitution.

Here is a sample mc.menu file:

```
A      Dump the currently selected file
        od -c %f

B      Edit a bug report and send it to root
        I='mktmp ${MC_TMPDIR:-/tmp}/mail.XXXXXX' || exit 1
        vi $I
        mail -s "Midnight Commander bug" root < $I
        rm -f $I

M      Read mail
        emacs -f rmail

N      Read Usenet news
        emacs -f gnus

H      Call the info hypertext browser
        info

J      Copy current directory to other panel recursively
        tar cf - . | (cd %D && tar xvpf -)

K      Make a release of the current subdirectory
        echo -n "Name of distribution file: "
        read tar
        ln -s %d 'dirname %d'/$tar
        cd ..
        tar cvhf ${tar}.tar $tar

= f *.tar.gz | f *.tgz & t n
X      Extract the contents of a compressed tar file
        tar xzvf %f
```

### Default Conditions

Each menu entry may be preceded by a condition. The condition must start from the first column with a '=' character. If the condition is true, the menu entry will be the default entry.

```
Condition syntax:    = <sub-cond>
or:                  = <sub-cond> | <sub-cond> ...
or:                  = <sub-cond> & <sub-cond> ...
```

Sub-condition is one of following:

```
y <pattern>          syntax of current file matching pattern?
                      (for edit menu only)
f <pattern>          current file matching pattern?
F <pattern>          other file matching pattern?
d <pattern>          current directory matching pattern?
D <pattern>          other directory matching pattern?
t <type>             current file of type?
T <type>             other file of type?
x <filename>        is it executable filename?
```

! <sub-cond> negate the result of sub-condition

Pattern is a normal shell pattern or a regular expression, according to the shell patterns option. You can override the global value of the shell patterns option by writing "shell\_patterns=x" on the first line of the menu file (where "x" is either 0 or 1).

Type is one or more of the following characters:

```
n    not a directory
r    regular file
d    directory
l    link
c    character device
b    block device
f    FIFO (pipe)
s    socket
x    executable file
t    tagged
```

For example 'rlf' means either regular file, link or fifo. The 't' type is a little special because it acts on the panel instead of the file. The condition '=t t' is true if there are tagged files in the current panel and false if not.

If the condition starts with '=?' instead of '=' a debug trace will be shown whenever the value of the condition is calculated.

The conditions are calculated from left to right. This means

```
= f *.tar.gz | f *.tgz & t n
```

is calculated as

```
( ( f *.tar.gz ) | ( f *.tgz ) ) & ( t n )
```

Here is a sample of the use of conditions:

```
= f *.tar.gz | f *.tgz & t n
```

```
L    List the contents of a compressed tar-archive
      gzip -cd %f | tar xvf -
```

### Addition Conditions

If the condition begins with '+ ' (or '+?') instead of '=' (or '=?') it is an addition condition. If the condition is true the menu entry will be included in the menu. If the condition is false the menu entry will not be included in the menu.

You can combine default and addition conditions by starting condition with '+=' or '=+' (or '+=?' or '=+?' if you want debug trace). If you want to use two different conditions, one for adding and another for defaulting, you can precede a menu entry with two condition lines, one starting with '+ ' and another starting with '='.

Comments are started with '#'. The additional comment lines must start with '#', space or tab.

### Options Menu

The Midnight Commander has some options that may be toggled on and off in several dialogs which are accessible from this menu. Options are enabled if they have an asterisk or "x" in front of them.

The Configuration command pops up a dialog from which you can change most of settings of the Midnight Commander.

The Layout command pops up a dialog from which you specify a bunch of options how mc looks like on the screen.

The Confirmation command pops up a dialog from which you specify which actions you want to confirm.

The Display bits command pops up a dialog from which you may select which characters is your

terminal able to display.

The Learn keys command pops up a dialog from which you test some keys which are not working on some terminals and you may fix them.

The Virtual FS command pops up a dialog from which you specify some VFS related options.

The Save setup command saves the current settings of the Left, Right and Options menus. A small number of other settings is saved, too.

## Configuration

The options in this dialog are divided into three groups: Panel Options, Pause after run and Other Options.

### Panel Options

*Show Backup Files.* If enabled, the Midnight Commander will show files ending with a tilde. Otherwise, they won't be shown (like GNU's ls option -B).

*Show Hidden Files.* If enabled, the Midnight Commander will show all files that start with a dot (like ls -a).

*Mark moves down.* If enabled, the selection bar will move down when you mark a file (with either C-t or the Insert key).

*Drop down menus.* When this option is enabled, the pull down menus will be activated as soon as you press the **F9** key. Otherwise, you will only get the menu title, and you will have to activate the menu either with the arrow keys or with the hotkeys. It is recommended if you are using hotkeys.

*Mix all files.* If this option is enabled, all files and directories are shown mixed together. If the option is off, directories (and links to directories) are shown at the beginning of the listing, and other files below.

*Fast directory reload.* If this option is enabled, the Midnight Commander will use a trick to determine if the directory contents have changed. The trick is to reload the directory only if the i-node of the directory has changed; this means that reloads only happen when files are created or deleted. If what changes is the i-node for a file in the directory (file size changes, mode or owner changes, etc) the display is not updated. In these cases, if you have the option on, you have to rescan the directory manually (with C-r).

### Pause after run

After executing your commands, the Midnight Commander can pause, so that you can examine the output of the command. There are three possible settings for this variable:

*Never.* Means that you do not want to see the output of your command. If you are using the Linux or FreeBSD console or an xterm, you will be able to see the output of the command by typing C-o.

*On dumb terminals.* You will get the pause message on terminals that are not capable of showing the output of the last command executed (any terminal that is not an xterm or the Linux console).

*Always.* The program will pause after executing all of your commands.

### Other Options

*Verbose operation.* This toggles whether the file Copy, Rename and Delete operations are verbose (i.e., display a dialog box for each operation). If you have a slow terminal, you may wish to disable the verbose operation. It is automatically turned off if the speed of your terminal is less than 9600 bps.

*Compute totals.* If this option is enabled, the Midnight Commander computes total byte sizes and total number of files prior to any Copy, Rename and Delete operations. This will provide you with a more accurate progress bar at the expense of some speed. This option has no effect, if *Verbose*

*operation* is disabled.

*Shell Patterns.* By default the Select, Unselect and Filter commands will use shell-like regular expressions. The following conversions are performed to achieve this: the '\*' is replaced by '.\*' (zero or more characters); the '?' is replaced by '.' (exactly one character) and '.' by the literal dot. If the option is disabled, then the regular expressions are the ones described in ed(1).

*Auto Save Setup.* If this option is enabled, when you exit the Midnight Commander the configurable options of the Midnight Commander are saved in the ~/.mc/ini file.

*Auto menus.* If this option is enabled, the user menu will be invoked at startup. Useful for building menus for non-unixers.

*Use internal editor.* If this option is enabled, the built-in file editor is used to edit files. If the option is disabled, the editor specified in the **EDITOR** environment variable is used. If no editor is specified, vi is used. See the section on the internal file editor.

*Use internal viewer.* If this option is enabled, the built-in file viewer is used to view files. If the option is disabled, the pager specified in the **PAGER** environment variable is used. If no pager is specified, the **view** command is used. See the section on the internal file viewer.

*Complete: show all.* By default the Midnight Commander pops up all possible completions if the completion is ambiguous only when you press **M-Tab** for the second time. For the first time, it just completes as much as possible and beeps in the case of ambiguity. Enable this option if you want to see all possible completions even after pressing **M-Tab** the first time.

*Rotating dash.* If this option is enabled, the Midnight Commander shows a rotating dash in the upper right corner as a work in progress indicator.

*Lynx-like motion.* If this option is enabled, you may use the arrows keys to automatically chdir if the current selection is a subdirectory and the shell command line is empty. By default, this setting is off.

*Cd follows links.* This option, if set, causes the Midnight Commander to follow the logical chain of directories when changing current directory either in the panels, or using the cd command. This is the default behavior of bash. When unset, the Midnight Commander follows the real directory structure, so cd .. if you've entered that directory through a link will move you to the current directory's real parent and not to the directory where the link was present.

*Safe delete.* If this option is enabled, deleting files unintentionally becomes more difficult. The default selection in the confirmation dialogs for deletion changes from "Yes" to "No". This option is disabled by default.

## Layout

The layout dialog gives you a possibility to change the general layout of screen. You can specify whether the menubar, the command prompt, the hintbar and the function keybar are visible. On the Linux or FreeBSD console you can specify how many lines are shown in the output window.

The rest of the screen area is used for the two directory panels. You can specify whether the area is split to the panels in vertical or horizontal direction. The split can be equal or you can specify an unequal split.

You can specify whether *permissions* and *file types* should be highlighted with distinctive Colors. If the permission highlighting is enabled, the parts of the *perm* and *mode* display fields which apply to the user running Midnight Commander are highlighted with the color defined by the *selected* keyword. If the file type highlighting is enabled, files are colored according to their file type (e.g. directory, core file, executable, and so on).

If the *Show Mini-Status* option is enabled, one line of status information about the currently selected item is shown at the bottom of the panels.

When run in a terminal emulator for X11, Midnight Commander sets the terminal window title to the current working directory and updates it when necessary. If your terminal emulator is broken and you see some incorrect output on startup and directory change, turn off the *Xterm Window*

*Title* option.

### Confirmation

In this menu you configure the confirmation options for file deletion, overwriting, execution by pressing enter and quitting the program.

### Display bits

This is used to configure the range of visible characters on the screen. This setting may be 7-bits if your terminal/curses supports only seven output bits, ISO-8859-1 displays all the characters in the ISO-8859-1 map and full 8 bits is for those terminals that can display full 8 bit characters.

### Learn keys

This dialog allows you to test and redefine functional keys, cursor arrows and some other keys to make them work properly on your terminal. They often don't, since many terminal databases are incomplete or broken.

You can move around with the Tab key and with the vi moving keys ('h' left, 'j' down, 'k' up and 'l' right). Once you press any cursor movement key and it is recognized, you can use that key as well.

You can test keys just by pressing each of them. When you press a key and it is recognized properly, OK should appear next to the name of that key. Once a key is marked OK it starts working as usually, e.g. F1 pressed the first time will just check that the F1 key works, but after that it will show help. The same applies to the arrow keys. The Tab key should be working always.

If some keys do not work properly then you won't see OK appear after pressing one of these. Then you may want to redefine it. Do it by pressing the button with the name of that key (either by the mouse or by Enter or Space after selecting the button with Tab or arrows). Then a message box will appear asking you to press that key. Do it and wait until the message box disappears. If you want to abort, just press Escape once and wait.

When you finish with all the keys, you can Save them. The definitions for the keys you have redefined will be written into the [terminal:TERM] section of your ~/.mc/ini file (where TERM is the name of your current terminal). The definitions of the keys that were already working properly are not saved.

### Virtual FS

This option gives you control over the settings of the Virtual File System.

The Midnight Commander keeps in memory the information related to some of the virtual file systems to speed up the access to the files in the file system (for example, directory listings fetched from FTP servers).

Also, in order to access the contents of compressed files (for example, compressed tar files) the Midnight Commander needs to create temporary uncompressed files on your disk.

Since both the information in memory and the temporary files on disk take up resources, you may want to tune the parameters of the cached information to decrease your resource usage or to maximize the speed of access to frequently used file systems.

Because of the format of the tar archives, the *Tar filesystem* needs to read the whole file just to load the file entries. Since most tar files are usually kept compressed (plain tar files are species in extinction), the tar file system has to uncompress the file on the disk in a temporary location and then access the uncompressed file as a regular tar file.

Now, since we all love to browse files and tar files all over the disk, it's common that you will leave a tar file and re-enter it later. Since decompression is slow, the Midnight Commander will cache the information in memory for a limited time. When the timeout expires, all the resources associated with the file system are released. The default timeout is set to one minute.

The FTP File System (ftpf) allows you to browse directories on remote FTP servers. It has several options.

*ftp anonymous password* is the password used when you login as "anonymous". Some sites

require a valid e-mail address. On the other hand, you probably don't want to give your real e-mail address to untrusted sites, especially if you are not using spam filtering.

ftps keeps the directory listing it fetches from a FTP server in a cache. The cache expire time is configurable with the *ftps directory cache timeout* option. A low value for this option may slow down every operation on the ftps because every operation would require sending a request to the FTP server.

You can define an FTP proxy host for doing FTP. Note that most modern firewalls are fully transparent at least for passive FTP (see below), so FTP proxies are considered obsolete.

If *Always use ftp proxy* is not set, you can use the exclamation sign to enable proxy for certain hosts. See FTP File System for examples.

If this option is set, the program will do two things: consult the `/usr/lib/mc/mc.no_proxy` file for lines containing host names that are local (if the host name starts with a dot, it is assumed to be a domain) and to assume that any hostnames without dots in their names are directly accessible. All other hosts will be accessed through the specified FTP proxy.

You can enable using `~/.netrc` file, which keeps login names and passwords for ftp servers. See `netrc (5)` for the description of the `.netrc` format.

*Use passive mode* enables using FTP passive mode, when the connection for data transfer is initiated by the client, not by the server. This option is recommended and enabled by default. If this option is turned off, the data connection is initiated by the server. This may not work with some firewalls.

### Save Setup

At startup the Midnight Commander will try to load initialization information from the `~/.mc/ini` file. If this file doesn't exist, it will load the information from the system-wide configuration file, located in `/etc/mc/mc.ini`. If the system-wide configuration file doesn't exist, MC uses the default settings.

The *Save Setup* command creates the `~/.mc/ini` file by saving the current settings of the Left, Right and Options menus.

If you activate the *auto save setup* option, MC will always save the current settings when exiting.

There also exist settings which can't be changed from the menus. To change these settings you have to edit the setup file with your favorite editor. See the section on Special Settings for more information.

### Executing operating system commands

You may execute commands by typing them directly in the Midnight Commander's input line, or by selecting the program you want to execute with the selection bar in one of the panels and hitting Enter.

If you press Enter over a file that is not executable, the Midnight Commander checks the extension of the selected file against the extensions in the Extensions File. If a match is found then the code associated with that extension is executed. A very simple macro expansion takes place before executing the command.

### The `cd` internal command

The `cd` command is interpreted by the Midnight Commander, it is not passed to the command shell for execution. Thus it may not handle all of the nice macro expansion and substitution that your shell does, although it does some of them:

*Tilde substitution.* The (`~`) will be substituted with your home directory, if you append a username after the tilde, then it will be substituted with the login directory of the specified user.

For example, `~guest` is the home directory for the user `guest`, while `~/guest` is the directory `guest` in your home directory.

*Previous directory.* You can jump to the directory you were previously by using the special directory name '-' like this: **cd -**

*CDPATH directories.* If the directory specified to the **cd** command is not in the current directory, then The Midnight Commander uses the value in the environment variable **CDPATH** to search for the directory in any of the named directories.

For example you could set your **CDPATH** variable to `~/src:/usr/src`, allowing you to change your directory to any of the directories inside the `~/src` and `/usr/src` directories, from any place in the file system by using its relative name (for example `cd linux` could take you to `/usr/src/linux`).

## Macro Substitution

When accessing a user menu, or executing an extension dependent command, or running a command from the command line input, a simple macro substitution takes place.

The macros are:

*%i* The indent of blank space, equal the cursor column position. For edit menu only.

*%y* The syntax type of current file. For edit menu only.

*%k* The block file name.

*%e* The error file name.

*%m* The current menu name.

*%f* and *%p*  
The current file name.

*%x* The extension of current file name.

*%b* The current file name without extension.

*%d* The current directory name.

*%F* The current file in the unselected panel.

*%D* The directory name of the unselected panel.

*%t* The currently tagged files.

*%T* The tagged files in the unselected panel.

*%u* and *%U*  
Similar to the *%t* and *%T* macros, but in addition the files are untagged. You can use this macro only once per menu file entry or extension file entry, because next time there will be no tagged files.

*%s* and *%S*  
The selected files: The tagged files if there are any. Otherwise the current file.

*%cd* This is a special macro that is used to change the current directory to the directory specified in front of it. This is used primarily as an interface to the Virtual File System.

*%view* This macro is used to invoke the internal viewer. This macro can be used alone, or with arguments. If you pass any arguments to this macro, they should be enclosed in brackets.

The arguments are: *ascii* to force the viewer into ascii mode; *hex* to force the viewer into hex mode; *nroff* to tell the viewer that it should interpret the bold and underline sequences of nroff; *unformatted* to tell the viewer to not interpret nroff commands for making the text bold or underlined.

*%%* The % character

*%{some text}*  
Prompt for the substitution. An input box is shown and the text inside the braces is used as a prompt. The macro is substituted by the text typed by the user. The user can press ESC or F10 to cancel. This macro doesn't work on the command line yet.

*%var{ENV:default}*

If environment variable *ENV* is unset, the *default* is substituted. Otherwise, the value of *ENV* is substituted.

### The subshell support

The subshell support is a compile time option, that works with the shells: bash, tcsh and zsh.

When the subshell code is activated the Midnight Commander will spawn a concurrent copy of your shell (the one defined in the **SHELL** variable and if it is not defined, then the one in the `/etc/passwd` file) and run it in a pseudo terminal, instead of invoking a new shell each time you execute a command, the command will be passed to the subshell as if you had typed it. This also allows you to change the environment variables, use shell functions and define aliases that are valid until you quit the Midnight Commander.

If you are using **bash** you can specify startup commands for the subshell in your `~/.mc/bashrc` file and special keyboard maps in the `~/.mc/inputrc` file. **tcsh** users may specify startup commands in the `~/.mc/tcshrc` file.

When the subshell code is used, you can suspend applications at any time with the sequence C-o and jump back to the Midnight Commander, if you interrupt an application, you will not be able to run other external commands until you quit the application you interrupted.

An extra added feature of using the subshell is that the prompt displayed by the Midnight Commander is the same prompt that you are currently using in your shell.

The OPTIONS section has more information on how you can control the subshell code.

### Chmod

The Chmod window is used to change the attribute bits in a group of files and directories. It can be invoked with the C-x c key combination.

The Chmod window has two parts - *Permissions* and *File*.

In the File section are displayed the name of the file or directory and its permissions in octal form, as well as its owner and group.

In the Permissions section there is a set of check buttons which correspond to the file attribute bits. As you change the attribute bits, you can see the octal value change in the File section.

To move between the widgets (buttons and check buttons) use the *arrow keys* or the *Tab* key. To change the state of the check buttons or to select a button use *Space*. You can also use the hotkeys on the buttons to quickly activate them. Hotkeys are shown as highlighted letters on the buttons.

To set the attribute bits, use the Enter key.

When working with a group of files or directories, you just click on the bits you want to set or clear. Once you have selected the bits you want to change, you select one of the action buttons (Set marked or Clear marked).

Finally, to set the attributes exactly to those specified, you can use the **[Set all]** button, which will act on all the tagged files.

**[Marked all]** set only marked attributes to all selected files

**[Set marked]** set marked bits in attributes of all selected files

**[Clean marked]** clear marked bits in attributes of all selected files

**[Set]** set the attributes of one file

**[Cancel]** cancel the Chmod command

### Chown

The Chown command is used to change the owner/group of a file. The hot key for this command is C-x o.

## Advanced Chown

The Advanced Chown command is the Chmod and Chown command combined into one window. You can change the permissions and owner/group of files at once.

## File Operations

When you copy, move or delete files the Midnight Commander shows the file operations dialog. It shows the files currently being processed and uses up to three progress bars. The file bar indicates the percentage of the current file that has been processed so far. The count bar shows how many of the tagged files have been handled. The bytes bar indicates the percentage of the total size of the tagged files that has been handled. If the verbose option is off, the file and bytes bars are not shown.

There are two buttons at the bottom of the dialog. Pressing the Skip button will skip the rest of the current file. Pressing the Abort button will abort the whole operation, the rest of the files are skipped.

There are three other dialogs which you can run into during the file operations.

The error dialog informs about error conditions and has three choices. Normally you select either the Skip button to skip the file or the Abort button to abort the operation altogether. You can also select the Retry button if you fixed the problem from another terminal.

The replace dialog is shown when you attempt to copy or move a file on the top of an existing file. The dialog shows the dates and sizes of the both files. Press the Yes button to overwrite the file, the No button to skip the file, the All button to overwrite all the files, the None button to never overwrite and the Update button to overwrite if the source file is newer than the target file. You can abort the whole operation by pressing the Abort button.

The recursive delete dialog is shown when you try to delete a directory which is not empty. Press the Yes button to delete the directory recursively, the No button to skip the directory, the All button to delete all the directories and the None button to skip all the non-empty directories. You can abort the whole operation by pressing the Abort button. If you selected the Yes or All button you will be asked for a confirmation. Type "yes" only if you are really sure you want to do the recursive delete.

If you have tagged files and perform an operation on them only the files on which the operation succeeded are untagged. Failed and skipped files are left tagged.

## Mask Copy/Rename

The copy/move operations let you translate the names of files in an easy way. To do it, you have to specify the correct source mask and usually in the trailing part of the destination specify some wildcards. All the files matching the source mask are copied/renamed according to the target mask. If there are tagged files, only the tagged files matching the source mask are renamed.

There are other options which you can set:

### Follow links

determines whether make the symlinks and hardlinks in the source directory (recursively in subdirectories) new links in the target directory or whether would you like to copy their content.

### Dive into subdirs

determines the behavior when the source directory is about to be copied, but the target directory already exists. The default action is to copy the contents of the source directory into the target directory. Enabling this option causes copying the source directory itself into the target directory.

For example, you want to copy directory */foo* containing file *bar* to */bla/foo*, which is an already existing directory. Normally (when **Dive into subdirs** is not set), mc would copy file */foo/bar* into the file */bla/foo/bar*. By enabling this option the */bla/foo/foo* directory will be created, and */foo/bar* will be copied into */bla/foo/foo/bar*.

### Preserve attributes

determines whether to preserve the permissions, timestamps and (if you are root) the ownership of the original files. If this option is not set, the current value of the umask will be respected.

### Use shell patterns on

When the shell patterns option is on you can use the '\*' and '?' wildcards in the source mask. They work like they do in the shell. In the target mask only the '\*' and '\<digit>' wildcards are allowed. The first '\*' wildcard in the target mask corresponds to the first wildcard group in the source mask, the second '\*' corresponds to the second group and so on. The '\1' wildcard corresponds to the first wildcard group in the source mask, the '\2' wildcard corresponds to the second group and so on all the way up to '\9'. The '\0' wildcard is the whole filename of the source file.

Two examples:

If the source mask is "\*.tar.gz", the destination is "/bla/\*.tgz" and the file to be copied is "foo.tar.gz", the copy will be "foo.tgz" in "/bla".

Suppose you want to swap basename and extension so that "file.c" would become "c.file" and so on. The source mask for this is "/\*.\*" and the destination is "\2.\1".

### Use shell patterns off

When the shell patterns option is off the MC doesn't do automatic grouping anymore. You must use '\(...\)' expressions in the source mask to specify meaning for the wildcards in the target mask. This is more flexible but also requires more typing. Otherwise target masks are similar to the situation when the shell patterns option is on.

Two examples:

If the source mask is "^\(.\*\)\.tar\.gz\$", the destination is "/bla/\*.tgz" and the file to be copied is "foo.tar.gz", the copy will be "/bla/foo.tgz".

Let's suppose you want to swap basename and extension so that "file.c" will become "c.file" and so on. The source mask for this is "^\(.\*\)\. \(.\*\) \$" and the destination is "\2.\1".

### Case Conversions

You can also change the case of the filenames. If you use '\u' or '\l' in the target mask, the next character will be converted to uppercase or lowercase correspondingly.

If you use '\U' or '\L' in the target mask, the next characters will be converted to uppercase or lowercase correspondingly up to the next '\E' or next '\U', '\L' or the end of the file name.

The '\u' and '\l' are stronger than '\U' and '\L'.

For example, if the source mask is '\*' (shell patterns on) or '^\(.\*\) \$' (shell patterns off) and the target mask is '\L\u\*' the file names will be converted to have initial upper case and otherwise lower case.

You can also use '\ ' as a quote character. For example, '\ \' is a backslash and '\\*' is an asterisk.

## Internal File Viewer

The internal file viewer provides two display modes: ASCII and hex. To toggle between modes, use the F4 key. If you have the GNU gzip program installed, it will be used to automatically decompress the files on demand.

The viewer will try to use the best method provided by your system or the file type to display the information. The internal file viewer will interpret some string sequences to set the bold and underline attributes, thus making a pretty display of your files.

When in hex mode, the search function accepts text in quotes and constant numbers. Text in quotes is matched exactly after removing the quotes. Each number matches one byte. You can mix quoted text with constants like this:

```
"String" -1 0xBB 012 "more text"
```

Note that 012 is an octal number. -1 is converted to 0xFF.

Some internal details about the viewer: On systems that provide the `mmap(2)` system call, the program maps the file instead of loading it; if the system does not provide the `mmap(2)` system call or the file matches an action that requires a filter, then the viewer will use its growing buffers, thus loading only those parts of the file that you actually access (this includes compressed files).

Here is a listing of the actions associated with each key that the Midnight Commander handles in the internal file viewer.

**F1** Invoke the built-in hypertext help viewer.

**F2** Toggle the wrap mode.

**F4** Toggle the hex mode.

**F5** Goto line. This will prompt you for a line number and will display that line.

**F6**, **/**. Regular expression search.

**?**, **Reverse** regular expression search.

**F7** Normal search / hex mode search.

**C-s**, **F17**, **n**. Start normal search if there was no previous search expression else find next match.

**C-r**. Start reverse search if there was no previous search expression else find next match.

**F8** Toggle Raw/Parsed mode: This will show the file as found on disk or if a processing filter has been specified in the `mc.ext` file, then the output from the filter. Current mode is always the other than written on the button label, since on the button is the mode which you enter by that key.

**F9** Toggle the format/unformat mode: when format mode is on the viewer will interpret some string sequences to show bold and underline with different colors. Also, on button label is the other mode than current.

**F10**, **Esc**. Exit the internal file viewer.

**next-page**, **space**, **C-v**. Scroll one page forward.

**prev-page**, **M-v**, **C-b**, **backspace**. Scroll one page backward.

**down-key** Scroll one line forward.

**up-key** Scroll one line backward.

**C-l** Refresh the screen.

**C-o** Switch to the subshell and show the command screen.

**!** Like **C-o**, but run a new shell if the subshell is not running.

**[n] m** Set the mark `n`.

**[n] r** Jump to the mark `n`.

**C-f** Jump to the next file.

**C-b** Jump to the previous file.

**M-r** Toggle the ruler.

It's possible to instruct the file viewer how to display a file, look at the Extension File Edit section

### Internal File Editor

The internal file editor is a full-featured full screen editor. It can edit files up to 64 megabytes. It is possible to edit binary files. The internal file editor is invoked using **F4** if the `use_internal_edit` option is set in the initialization file.

The features it presently supports are: block copy, move, delete, cut, paste; key for key undo; pull-down menus; file insertion; macro commands; regular expression search and replace (and our own `scanf`/`printf` search and replace); shift-arrow text highlighting (if supported by the terminal);

insert-overwrite toggle; word wrap; autoindent; tunable tab size; syntax highlighting for various file types; and an option to pipe text blocks through shell commands like indent and ispell.

The editor is very easy to use and requires no tutoring. To see what keys do what, just consult the appropriate pull-down menu. Other keys are: Shift movement keys do text highlighting. **Ctrl-Ins** copies to the file **cooledit.clip** and **Shift-Ins** pastes from **cooledit.clip**. **Shift-Del** cuts to **cooledit.clip**, and **Ctrl-Del** deletes highlighted text. Mouse highlighting also works, and you can override the mouse as usual by holding down the shift key while dragging the mouse to let normal terminal mouse highlighting work.

To define a macro, press **Ctrl-R** and then type out the key strokes you want to be executed. Press **Ctrl-R** again when finished. You can then assign the macro to any key you like by pressing that key. The macro is executed when you press **Ctrl-A** and then the assigned key. The macro is also executed if you press Meta, Ctrl, or Esc and the assigned key, provided that the key is not used for any other function. Once defined, the macro commands go into the file **.mc/cedit/cooledit.macros** in your home directory. You can delete a macro by deleting the appropriate line in this file.

**F19** will format the currently highlighted block (plain text or **C** or **C++** code or another). This is controlled by the file **/usr/share/mc/edit.indent.rc** which is copied to **.mc/cedit/edit.indent.rc** in your home directory the first time you use it.

You can use scanf search and replace to search and replace a C format string. First take a look at the **sscanf** and **sprintf** man pages to see what a format string is and how it works. Consider following example. Suppose you want to replace all occurrences of an open bracket, three comma separated numbers, and a close bracket, with the word *apples*, the third number, the word *oranges* and then the second number. Then fill in the Replace dialog box as follows:

```
Enter search string:
(%d,%d,%d)
Enter replacement string:
apples %d oranges %d
Enter replacement argument order:
3,2
```

The last line specifies that the third and then the second number are to be used in place of the first and second.

It is advisable to use this feature with Prompt on replace on, because a match is thought to be found whenever the number of arguments found matches the number given, which is not always a real match. Scnaf also treats whitespace as being elastic. Note that the scanf format **%[** is very useful for scanning strings, and whitespace.

The editor also displays non-us characters (160+). When editing binary files, you should set **display bits** to 7 bits in the options menu to keep the spacing clean.

## Completion

Let the Midnight Commander type for you.

Attempt to perform completion on the text before current position. MC attempts completion treating the text as variable (if the text begins with **\$**), username (if the text begins with **~**), hostname (if the text begins with **@**) or command (if you are on the command line in the position where you might type a command, possible completions then include shell reserved words and shell built-in commands as well) in turn. If none of these matches, filename completion is attempted.

Filename, username, variable and hostname completion works on all input lines, command completion is command line specific. If the completion is ambiguous (there are more different possibilities), MC beeps and the following action depends on the setting of the *Complete: show all* option in the Configuration dialog. If it is enabled, a list of all possibilities pops up next to the current position and you can select with the arrow keys and **Enter** the correct entry. You can also type

the first letters in which the possibilities differ to move to a subset of all possibilities and complete as much as possible. If you press **M-Tab** again, only the subset will be shown in the listbox, otherwise the first item which matches all the previous characters will be highlighted. As soon as there is no ambiguity, dialog disappears, but you can hide it by canceling keys **Esc**, **F10** and left and right arrow keys. If Complete: show all is disabled, the dialog pops up only if you press **M-Tab** for the second time, for the first time MC just beeps.

## Virtual File System

The Midnight Commander is provided with a code layer to access the file system; this code layer is known as the virtual file system switch. The virtual file system switch allows the Midnight Commander to manipulate files not located on the Unix file system.

Currently the Midnight Commander is packaged with some Virtual File Systems (VFS): the *local* file system, used for accessing the regular Unix file system; the *ftpfs*, used to manipulate files on remote systems with the FTP protocol; the *tarfs*, used to manipulate tar and compressed tar files; the *undelfs*, used to recover deleted files on ext2 file systems (the default file system for Linux systems), *fish* (for manipulating files over shell connections such as rsh and ssh) and finally the *mcfs* (Midnight Commander file system), a network based file system. If the code was compiled with *smbfs* support, you can manipulate files on remote systems with the SMB (CIFS) protocol.

A generic *extfs* (EXTernal virtual File System) is provided in order to easily expand VFS capabilities using scripts and external software.

The VFS switch code will interpret all of the path names used and will forward them to the correct file system, the formats used for each one of the file systems is described later in their own section.

## FTP File System

The FTP File System (*ftpfs*) allows you to manipulate files on remote machines. To actually use it, you can use the *FTP link* item in the menu or directly change your current directory using the *cd* command to a path name that looks like this:

```
/#ftp:[!][user[:pass]@]machine[:port][remote-dir]
```

The *user*, *port* and *remote-dir* elements are optional. If you specify the *user* element, the Midnight Commander will login to the remote machine as that user, otherwise it will use anonymous login or the login name from the *~/.netrc* file. The optional *pass* element is the password used for the connection. Using the password in the VFS directory name is not recommended, because it can appear on the screen in clear text and can be saved to the directory history.

To enable using FTP proxy, prepend **!** (an exclamation sign) to the hostname.

Examples:

```
/#ftp:ftp.nuclecu.unam.mx/linux/local
/#ftp:tsx-11.mit.edu/pub/linux/packages
/#ftp:!behind.firewall.edu/pub
/#ftp:guest@remote-host.com:40/pub
/#ftp:miguel:xxx@server/pub
```

Please check the Virtual File System dialog box for *ftpfs* options.

## Tar File System

The tar file system provides you with read-only access to your tar files and compressed tar files by using the *chdir* command. To change your directory to a tar file, you change your current directory to the tar file by using the following syntax:

```
/filename.tar#utar/[dir-inside-tar]
```

The *mc.ext* file already provides a shortcut for tar files, this means that usually you just point to a tar file and press return to enter into the tar file, see the Extension File Edit section for details on how this is done.

Examples:

```
mc-3.0.tar.gz#utar/mc-3.0/vfs
/ftp/GCC/gcc-2.7.0.tar#utar
```

The latter specifies the full path of the tar archive.

### File transfer over SHell filesystem

The fish file system is a network based file system that allows you to manipulate the files in a remote machine as if they were local. To use this, the other side has to either run fish server, or has to have bash-compatible shell.

To connect to a remote machine, you just need to chdir into a special directory which name is in the following format:

```
/#sh:[user@]machine[options]/[remote-dir]
```

The *user*, *options* and *remote-dir* elements are optional. If you specify the *user* element, the Midnight Commander will try to login on the remote machine as that user, otherwise it will use your login name.

The *options* are 'C' - use compression and 'rsh' use rsh instead of ssh. If the *remote-dir* element is present, your current directory on the remote machine will be set to this one.

Examples:

```
/#sh:onlyrsh.mx:r/linux/local
/#sh:joe@want.compression.edu:C/private
/#sh:joe@noncompressed.ssh.edu/private
```

### Network File System

The Midnight Commander file system is a network base file system that allows you to manipulate the files in a remote machine as if they were local. To use this, the remote machine must be running the mcserv(8) server program.

To connect to a remote machine, you just need to chdir into a special directory which name is in the following format:

```
/#mc:[user@]machine[:port]/[remote-dir]
```

The *user*, *port* and *remote-dir* elements are optional. If you specify the *user* element then the Midnight Commander will try to logon on the remote machine as that user, otherwise it will use your login name.

The *port* element is used when the remote server is running on a special port (see the mcserv(8) manual page for more information about ports); finally, if the *remote-dir* element is present, your current directory on the remote machine will be set to this one.

Examples:

```
/#mc:ftp.nuclecu.unam.mx/linux/local
/#mc:joe@foo.edu:11321/private
```

### Undelete File System

On Linux systems, if you asked configure to use the ext2fs undelete facilities, you will have the undelete file system available. Recovery of deleted files is only available on ext2 file systems. The undelete file system is just an interface to the ext2fs library to retrieve all of the deleted files names on an ext2fs and provides and to extract the selected files into a regular partition.

To use this file system, you have to chdir into the special file name formed by the "/#undel" prefix and the file name where the actual file system resides.

For example, to recover deleted files on the second partition of the first SCSI disk on Linux, you would use the following path name:

```
/#undel:sda2
```

It may take a while for the undelfs to load the required information before you start browsing files there.

## SMB File System

The `smbfs` allows you to manipulate files on remote machines with SMB (or CIFS) protocol. These include Windows for Workgroups, Windows 9x/ME/XP, Windows NT, Windows 2000 and Samba. To actually use it, you may try to use the panel command "SMB link..." (accessible from the menubar) or you may directly change your current directory to it using the `cd` command to a path name that looks like this:

```
/#smb:[user@]machine[/service][[/remote-dir]
```

The *user*, *service* and *remote-dir* elements are optional. The *user*, *domain* and *password* can be specified in an input dialog.

Examples:

```
/#smb:machine/Share
/#smb:other_machine
/#smb:guest@machine/Public/Irlex
```

## EXternal File System

`extfs` allows to integrate numerous features and file types into GNU Midnight Commander in an easy way, by writing scripts.

Extfs filesystems can be divided into two categories:

1. Stand-alone filesystems, which are not associated with any existing file. They represent certain system-wide data as a directory tree. You can invoke them by typing '`cd #fsname`' where *fsname* is an extfs short name (see below). Examples of such filesystems include `audio` (list audio tracks on the CD) or `apt` (list of all Debian packages in the system).

For example, to list CD-Audio tracks on your CD-ROM drive, type

```
cd #audio
```

2. 'Archive' filesystems (like `rpm`, `patchfs` and more), which represent contents of a file as a directory tree. It can consist of 'real' files compressed in an archive (`urarc`, `rpm`) or virtual files, like messages in a mailbox (`mailfs`) or parts of a patch (`patchfs`). To access such filesystems '*fsname*' should be appended to the archive name. Note that the archive itself can be on another `vfs`.

For example, to list contents of a zip archive `documents.zip` type

```
cd documents.zip#uzip
```

In many aspects, you could treat extfs like any other directory. For instance, you can add it to the hotlist or change to it from directory history. An important limitation is that you cannot invoke shell commands inside extfs, just like any other non-local VFS.

Common extfs scripts included with Midnight Commander are:

- a** access 'A:' DOS/Windows diskette (`cd #a`).
- apt** front end to Debian's APT package management system (`cd #apt`).
- audio** audio CD ripping and playing (`cd #audio` or `cd device#audio`).
- bpp** package of Bad Penguin GNU/Linux distribution (`cd file.bpp#bpp`).
- deb** package of Debian GNU/Linux distribution (`cd file.deb#deb`).
- dpkg** Debian GNU/Linux installed packages (`cd #deb`).
- hp48** view and copy files to/from a HP48 calculator (`cd #hp48`).
- lslR** browsing of lslR listings as found on many FTPs (`cd filename#lslR`).
- mailfs** mbox-style mailbox files support (`cd mailbox#mailfs`).
- patchfs**  
extfs to handle unified and context diffs (`cd filename#patchfs`).

**rpm** RPM package (*cd filename#rpm*).

**rpms** RPM database management (*cd #rpms*).

**ulha, urar, uzip, uzoo, uar, uha**

archivers (*cd archive#xxxx* where xxxx is one of: *ulha, urar, uzip, uzoo, uar, uha*).

You could bind file type/extension to specified extfs as described in the Extension File Edit section. Here is an example entry for Debian packages:

```
regex/.deb$
  Open=%cd %p#deb
```

## Colors

The Midnight Commander will try to detect if your terminal supports color using the terminal database and your terminal name. Sometimes it gets confused, so you may force color mode or disable color mode using the `-c` and `-b` flag respectively.

If the program is compiled with the Slang screen manager instead of ncurses, it will also check the variable **COLORTERM**, if it is set, it has the same effect as the `-c` flag.

You may specify terminals that always force color mode by adding the *color\_terminals* variable to the Colors section of the initialization file. This will prevent the Midnight Commander from trying to detect if your terminal supports color. Example:

```
[Colors]
color_terminals=linux,xterm
color_terminals=terminal-name1,terminal-name2...
```

The program can be compiled with both ncurses and slang, ncurses does not provide a way to force color mode: ncurses uses just the information in the terminal database.

The Midnight Commander provides a way to change the default colors. Currently the colors are configured using the environment variable **MC\_COLOR\_TABLE** or the Colors section in the initialization file.

In the Colors section, the default color map is loaded from the *base\_color* variable. You can specify an alternate color map for a terminal by using the terminal name as the key in this section. Example:

```
[Colors]
base_color=
xterm=menu=magenta:marked=,magenta:markselect=,red
```

The format for the color definition is:

```
<keyword>=<foregroundcolor>,<backgroundcolor>:<keyword>= ...
```

The colors are optional, and the keywords are: normal, selected, marked, markselect, errors, input, reverse, gauge. Menu colors are: menu, menusel, menuhot, menuhotsel. Dialog colors are: dnormal, dfocus, dhotnormal, dhotfocus. Help colors are: helpnormal, helpitalic, helpbold, helplink, helpslink. Viewer color is: viewunderline. Special highlighting colors are: executable, directory, link, stalelink, device, special, core. Editor colors are: editnormal, editbold, editmarked.

*input* determines the color of input lines used in query dialogs.

*gauge* determines the color of the filled part of the progress bar (gauge), which is used to show the user the progress of file operations, such as copying.

The dialog boxes use the following colors: *dnormal* is used for the normal text, *dfocus* is the color used for the currently selected component, *dhotnormal* is the color used to differentiate the hotkey color in normal components, whereas the *dhotfocus* color is used for the highlighted color in the currently selected component.

Menus use the same scheme but uses the menu, menusel, menuhot and menuhotsel tags instead.

Help uses the following colors: *helpnormal* is used for normal text, *helpitalic* is used for text which

is emphasized in italic in the manual page, *helpbold* is used for text which is emphasized in bold in the manual page, *helplink* is used for not selected hyperlinks and *helpslink* is used for selected hyperlink.

Special highlight colors determine how files are displayed when file highlighting is enabled (see the section on Layout). *directory* is used for directories or symbolic links to directories; *executable* for executable files; *link* is used for symbolic links which are neither stale nor linked to a directory; *stalelink* is used for stale symbolic links; *device* - character and block devices; *special* is used for special files, such as pipes and sockets; *core* is for core files.

The possible colors are: black, gray, red, brightred, green, brightgreen, brown, yellow, blue, brightblue, magenta, brightmagenta, cyan, brightcyan, lightgray and white. And there is a special keyword for transparent background. It is 'default'. The 'default' can only be used for background color. Example:

```
[Colors]
base_color=normal=white,default:marked=magenta,default
```

## Special Settings

Most of the settings of the Midnight Commander can be changed from the menus. However, there are a small number of settings which can only be changed by editing the setup file.

These variables may be set in your `~/.mc/ini` file:

### *clear\_before\_exec*

By default the Midnight Commander clears the screen before executing a command. If you would prefer to see the output of the command at the bottom of the screen, edit your `~/.mc/ini` file and change the value of the field `clear_before_exec` to 0.

### *confirm\_view\_dir*

If you press F3 on a directory, normally MC enters that directory. If this flag is set to 1, then MC will ask for confirmation before changing the directory if you have files tagged.

### *ftps\_retry\_seconds*

This value is the number of seconds the Midnight Commander will wait before attempting to reconnect to an FTP server that has denied the login. If the value is zero, the login will no be retried.

### *max\_dirt\_limit*

Specifies how many screen updates can be skipped at most in the internal file viewer. Normally this value is not significant, because the code automatically adjusts the number of updates to skip according to the rate of incoming keystrokes. However, on very slow machines or terminals with a fast keyboard auto repeat, a big value can make screen updates too jumpy.

It seems that setting `max_dirt_limit` to 10 causes the best behavior, and that is the default value.

### *mouse\_move\_pages*

Controls whenever scrolling with the mouse is done by pages or line by line on the panels.

### *mouse\_move\_pages\_viewer*

Controls if scrolling with the mouse is done by pages or line by line on the internal file viewer.

### *old\_esc\_mode*

By default the Midnight Commander treats the ESC key as a key prefix (`old_esc_mode=0`). If this option is set (`old_esc_mode=1`), the ESC key will act as a prefix key for one second, and if no extra keys have arrived, then the ESC key is interpreted as a cancel key (ESC ESC).

*only\_leading\_plus\_minus*

Allow special treatment for '+', '-', '\*' in the command line (select, unselect, reverse selection) only if the command line is empty. You don't need to quote those characters in the middle of the command line. On the other hand, you cannot use them to change selection when the command line is not empty.

*panel\_scroll\_pages*

If set (the default), panel will scroll by half the display when the cursor reaches the end or the beginning of the panel, otherwise it will just scroll a file at a time.

*show\_output\_starts\_shell*

This variable only works if you are not using the subshell support. When you use the C-o keystroke to go back to the user screen, if this one is set, you will get a fresh shell. Otherwise, pressing any key will bring you back to the Midnight Commander.

*torben\_fj\_mode*

If this flag is set, then the home and end keys will work slightly different on the panels, instead of moving the selection to the first and last files in the panels, they will act as follows:

The home key will: Go up to the middle line, if below it; else go to the top line unless it is already on the top line, in this case it will go to the first file in the panel.

The end key has a similar behavior: Go down to the middle line, if over it; else go to the bottom line unless you already are at the bottom line, in such case it will move the selection to the last file name in the panel.

*use\_file\_to\_guess\_type*

If this variable is on (the default) it will spawn the file command to match the file types listed on the mc.ext file.

*xterm\_mode*

If this variable is on (default is off) when you browse the file system on a Tree panel, it will automatically reload the other panel with the contents of the selected directory.

**Terminal databases**

The Midnight Commander provides a way to fix your system terminal database without requiring root privileges. The Midnight Commander searches in the system initialization file (the mc.lib file located in the Midnight Commander library directory) and in the ~/.mc/ini file for the section "terminal:your-terminal-name" and then for the section "terminal:general", each line of the section contains a key symbol that you want to define, followed by an equal sign and the definition for the key. You can use the special \e form to represent the escape character and the ^x to represent the control-x character.

The possible key symbols are:

f0 to f20	Function keys f0-f20
bs	backspace
home	home key
end	end key
up	up arrow key
down	down arrow key
left	left arrow key
right	right arrow key
pgdn	page down key
pgup	page up key
insert	the insert character
delete	the delete character
complete	to do completion

For example, to define the key insert to be the Escape + [ + O + p, you set this in the ini file:

insert=\e[Op

The *complete* key symbol represents the escape sequences used to invoke the completion process, this is invoked with M-tab, but you can define other keys to do the same work (on those keyboard with tons of nice and unused keys everywhere).

## FILES

Full paths below may vary between installations. They are also affected by the MC\_DATADIR environment variable. If it's set, its value is used instead of /usr/share/mc in the paths below.

*/usr/share/mc/mc.hlp*

The help file for the program.

*/etc/mc/mc.ext*

The default system-wide extensions file.

*~/.mc/bindings*

User's own extension, view configuration and edit configuration file. They override the contents of the system wide files if present.

*/etc/mc/mc.ini*

The default system-wide setup for the Midnight Commander, used only if the user doesn't have his own *~/.mc/ini* file.

*/etc/mc/mc.lib*

Global settings for the Midnight Commander. Settings in this file affect all users, whether they have *~/.mc/ini* or not. Currently, only terminal settings are loaded from mc.lib.

*~/.mc/ini*

User's own setup. If this file is present then the setup is loaded from here instead of the system-wide startup file.

*/usr/share/mc/mc.hint*

This file contains the hints displayed by the program.

*/etc/mc/mc.menu*

This file contains the default system-wide applications menu.

*~/.mc/menu*

User's own application menu. If this file is present it is used instead of the system-wide applications menu.

*~/.mc/Tree*

The directory list for the directory tree and tree view features.

*./mc.menu*

Local user-defined menu. If this file is present, it is used instead of the home or system-wide applications menu.

## LICENSE

This program is distributed under the terms of the GNU General Public License as published by the Free Software Foundation. See the built-in help for details on the License and the lack of warranty.

## AVAILABILITY

The latest version of this program can be found at <ftp://ftp.ibiblio.org/pub/Linux/utils/file/managers/mc/>.

**SEE ALSO**

ed(1), gpm(1), mcserv(8), terminfo(1), view(1), sh(1), bash(1), tcsh(1), zsh(1).

The Midnight Commander page on the World Wide Web:

<http://www.ibiblio.org/mc/>

**AUTHORS**

Authors and contributors are listed in the AUTHORS file in the source distribution.

**BUGS**

See the file TODO in the distribution for information on what remains to be done.

If you want to report a problem with the program, please send mail to this address: mc-devel@gnome.org.

Provide a detailed description of the bug, the version of the program you are running (*mc -V* displays this information), the operating system you are running the program on. If the program crashes, we would appreciate a stack trace.