

## NAME

mc – Visual shell for Unix-like systems.

## USAGE

**mc** [-abcCdfhPstuUVx?] [-l log] [dir1 [dir2]] [-v file]

## DESCRIPTION

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

## OPTIONS

- a Disables the usage of graphic characters for line drawing.
- b Forces black and white display.
- c Force color mode, please check the section Colors for more information.
- C *arg* Used to specify a different color set in the command line. The format of *arg* is documented in the Colors section.
- d Disables mouse support.
- f Displays the compiled-in search paths for Midnight Commander files.
- k Reset softkeys to their default from the termcap/terminfo database. Only useful on HP terminals when the function keys don't work.
- l *file* Save the ftpfs dialog with the server in file.
- P At program end, the Midnight Commander will print the last working directory; this, along with the shell function below, will allow you to browse through your directories and automatically move to the last directory you were in (thanks to Torben Fjordingstad and Sergey for contributing this function and the code which implements this option).

bash and zsh users:

```
mc ()
{
    MC=/tmp/mc$$-"$RANDOM"
    @prefix@/bin/mc -P "$@" > "$MC"
    cd "'cat $MC'"
    rm "$MC"
    unset MC;
}
```

tosh users:

```
alias mc 'setenv MC '@prefix@/bin/mc -P *'; cd $MC; unsetenv MC'
```

I know the bash function could be shorter for zsh and bash but the backquotes on bash won't accept your suspension the program with C-z.

- s Turns on the slow terminal mode, in this mode the program will not draw expensive line drawing characters and will toggle verbose mode off.
- t 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 Disables the use of a concurrent shell (only makes sense if the Midnight Commander has been built with concurrent shell support).
- U Enables the 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* Enters the internal viewer to view the file specified.
- V Displays the version of the program.
- x Forces 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 bottommost line 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 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 pageful backward. Correspondingly, a click on the bottom frame line will cause a scroll of one pageful forward. 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 Commander with the mouse support, you can bypass the Commander and 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>.

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 SCO 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-h (or 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 keypresses 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,** If the other panel is a listing panel and you are standing on a directory in the current panel, then the other panel contents are set to the contents of the currently selected directory (like Emacs' dired C-o key) otherwise the other panel contents are set to the parent dir of the current dir.

**C-PageUp, C-PageDown** Only when ran on the Linux console: does a chdir to "." and to the currently selected directory respectively.

## 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, this one only works on the Linux console.

**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 pageful backward.

**Next Page, Page Down, C-v.** Moves one pageful forward.

**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 to the ones mentioned above:

**b, C-b, C-h, Backspace, Delete.** Moves one pageful backward.

**Space bar.** Moves one pageful forward.

**u, d.** Moves one half of a page backward or forward.

**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.

### Left and Right Menus

The outlook of the directory panels can be changed from the **Left** and **Right** menus.

### 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. These are the available fields you may display:

**name**, displays the file name.

**size**, displays the file size.

**bsize**, is an alternative form of the `<bf/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 field type. This character is a superset of what is displayed by `ls` with the `-F` flag. An asterisk for executable files, a slash for directories, an at-sign for links, an equal sign for sockets, a hyphen for character devices, a plus sign for block devices, a pipe for fifos, a tilde for symbolic links to directories and an exclamation mark for stalled symlinks (links that point nowhere).

**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 may use these field names for arranging the display:

**space**, a space in the display format.

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

|, This character is used to add a vertical line to the display format.

To force one field to a fixed size (a size specifier), you just add a ':' and then the number of characters you want the field to have, if the number is followed by the symbol '+', then the size specifies the minimum field size, if the program finds out that there is more space on the screen, it will then expand this 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 Fkeys are terminfo capabilities kf1 trough 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 pre processing to the file.

### Filtered View (M-!)

this command prompts for a command and it's 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 file-name 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 for unselecting 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 SCO 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 sub-directories 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 CDROM 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 one. For adding you may want to use

a standalone Add to hotlist command (C-x h), which adds the current directory into the directory hotlist, as well. The user is prompted for a 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/ext`. The format of this file is as follows (the format has changed with version 3.0):

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

Lines starting in the first column should have following format:

*keyword/descNL*, i.e. everything after *keyword/* until new line is *desc*

keyword can be:

*shell*

(desc is then any extension (no wildcards), i.e. matches all the files `*desc` . Example: `.tar` matches `*.tar`)

*regex*

(desc is a regular expression)

*type*

(file matches this if 'file %f' matches regular expression desc (the filename: part from 'file %f' is removed))

*default*

(matches any file no matter what desc is)

Other lines should start with a space or tab and should be of the format:

*keyword=commandNL* (with no spaces around =), where *keyword* should be:

*Open* (if the user presses Enter or doubleclicks it), *View* (F3), *Edit* (F4), *Drop* (user drops some files on it) or any other user defined name (those will be listed in the extension dependent pop-up menu). *Icon* name is reserved for future use by mc.

*command* is any one-line shell command, with the simple macro substitution.

Target are evaluated from top to bottom (order is thus important). If some actions are missing, search continues as if this target 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 catch 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 `@prefix@/lib/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
      vi /tmp/mail.$$
      mail -s "Midnight Commander bug" root < /tmp/mail.$$

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:

```
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?
! <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 directory
r      regular file
d      directory
l      link
c      char special
```

b     block special  
 f     fifo  
 s     socket  
 x     executable  
 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' 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 Configuration command pops up a dialog from which you can change most of settings of the Midnight Commander.

The Display bits command pops up a dialog from which you may select which characters is your terminal able to display.

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

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

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 program has some options that may be toggled on and off from the Configuration dialog. Options are enabled if they have an asterisk or "x" in front of them. These options are divided into three groups: Screen Colors, Panel Options and Other Options.

### Screen Colors

You can select whether your display supports color or not. Normally this information is in the terminfo database. If you want to know how to change individual colors see the section on Colors.

### Panel Options

*Show Backup Files.* By default the Midnight Commander doesn't show files ending in '~' (like GNU's ls option -B).

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

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

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

*Mix all files.* When 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 afterwards.

*Fast directory reload.* This option is off by default. If you activate the fast reload, 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).

## 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.

*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 SCO 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.

*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.

*Confirm Delete.* This option is toggled on by default, and will cause the Midnight Commander to ask for confirmation when deleting a single file.

*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.

## 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.

## Confirmation

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

## Learn keys

This dialog lets you test if your keys F1-F20, Home, End, etc. work properly on your terminal. They often don't, since many terminal databases are broken.

You can move around with the Tab key, with the vi moving keys ('h' left, 'j' down, 'k' up and 'l' right) and after you press any arrow key once (this will mark it OK), then you can use that key as well.

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

If some keys do not work properly, then you won't see OK after the key name after you have pressed that key. You may then want to fix it. You do it by pressing the button of that key (either by mouse or using Tab and Enter). Then a red message will appear and you will be asked to type that key. If you want to abort this, press just Esc and wait until the message disappears. Otherwise type the key you're asked to type and also wait until the dialog disappears.

When you finish with all the keys, you may want either to Save your key fixes into your ~/.mc/ini file into the [terminal:TERM] section (where TERM is the name of your current terminal) or to discard them. If all your keys were working properly and you had not to fix any key, then (of course) no saving will occur.

## Virtual FS

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

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. Since the information that must be kept may be large (for example, compressed tar files may be kept in RAM for faster access), you may want to tune the parameters of the cached information to decrease your memory usage or to maximize the speed of access to frequently used file systems.

The Tar file system is quite clever about how it handles tar files: it just loads the directory entries and when it needs to use the information contained in the tar file, it goes and grab it.

In the wild, tar files are usually kept compressed (plain tar files are species in extinction), and because of the nature of those files (the directory entries for the tar files is not there waiting for us to be loaded), the tar file system has two choices: load the complete, uncompressed tar file into memory or uncompress the file in the disk in a temporary location and then access the uncompressed file as a regular tar file.

In this dialog box you tell the Midnight Commander which sizes for compressed tar files you will tolerate to load into your precious memory. The default setting is set to one megabyte, this means that compressed tar files whose size is at most one megabyte will be loaded into core, otherwise a temporary uncompressed tar file will be created to access the contents (all of this is transparent to the user).

The program will let you add a suffix to specify the units of the number you typed in, use 'k' for kilobyte and 'm' for megabyte. Our routine does not accept floating point numbers, so you can't use ".5 m" to specify 512 kilobytes, you will have to use "512 k" instead.

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 the re-enter it later. Since uncompression is slow, the Midnight Commander will cache the information in memory for a limited amount of time, after you hit the timeout, all of the memory resources associated with the file system will be freed. The default timeout is set to one minute.

## 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 SCO 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.

## 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 `@prefix@/lib/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 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 it's 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:

`%f`

The current file name.

`%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.

*%q*

Dropped files. In all places except in the Drop action of the mc.ext file, this will become a null string, in the Drop action it will be replaced with a space separated list of files that were dropped on the 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.

## 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.

### Controlling Midnight Commander

The Midnight Commander defines an environment variable MC\_CONTROL\_FILE. The commands executed by MC may give instructions to MC by writing to the file specified by this variable. This is only available if you compiled your copy of the Midnight Commander with the WANT\_PARSE option.

The following instructions are supported.

clear_tags	Clear all tags.
tag <filename>	Tag specified file.
untag <filename>	Untag specified file.
select <filename>	Move pointer to file.
change_panel	Switch between panels.
cd <path>	Change directory.

If the first letter of the instruction is in lower case it operates on the current panel. If the letter is in upper case the instruction operates on the other panel. The additional letters must be in lower case. Instructions must be separated by exactly one space, tab or newline. The instructions don't work in the Info, Tree and Quick views. The first error causes the rest to be ignored.

### 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 that selection (they are the highlight 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.

### File Operations

When you copy, move or delete files the Midnight Commander shows the file operations dialog. It shows the files currently being operated on and there are at most three progress bars. The file bar tells how big part of the current file has been copied so far. The count bar tells how many of tagged files

have been handled so far. The bytes bar tells how big part of total size of the tagged files has been handled so far. 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 lets 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 option which you can set:

Follow links tells whether make the symlinks and hardlinks in the source directory (recursively in sub-directories) new links in the target directory or whether would you like to copy their content.

Dive into subdirs tells what to do if in the target directory exists a directory with the same name as the file/directory being copied. The default action is to copy it's content into that directory, by enabling this you can copy the source directory into that directory. Perhaps an example will help:

You want to copy content of a directory foo to /bla/foo, which is an already existing directory. Normally (when Dive is not set), mc would copy it exactly into /bla/foo. By enabling this option you will copy the content into /bla/foo/foo, because the directory already exists.

Preserve attributes tells whether to preserve the original files' permissions, timestamps and if you are root whether to preserve the original files' UID and GID. 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".

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 "\*.c" 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 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

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 as well as hexadecimal constants.

You can mix quoted text with constants like this: `"String" 0xFE 0xBB "more text"`. Text between constants and quoted text is just ignored.

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 builtin 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.** 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.

**n.** 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.

**!** Spawn a shell in the currently working directory.

**[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 provides most of the features of common full screen editors. It is invoked using **F4** provided the `use_internal_edit` option is set in the initialization file. It has an extensible file size limit of sixteen megabytes and edits binary files flawlessly.

The features it presently supports are: Block copy, move, delete, cut, paste; *key for key undo* ; pull-down menus; file insertion; macro definition; regular expression search and replace (and our own `scanf` search and replace); shift-arrow MSW-MAC text highlighting (for the linux console only); insert-overwrite toggle; and an option to pipe text blocks through shell commands like `indent`.

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. The completion key also does a Return with an automatic indent. 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 `cedit/cooledit.macros` in your home directory. You can delete a macro by deleting the appropriate line in this file.

**F19 will format C code when it is highlighted. For this to work, make an executable file called `cedit/edit.indent.rc` in your home directory containing the following:**

```
#!/bin/sh
/usr/bin/indent -kr -pcs ~/cedit/cooledit.block >& /dev/null
cat /dev/null > ~/cedit/cooledit.error
```

You can use `scanf` search and replace to search and replace a C format string. First take a look at the **`scanf`** and **`printf` man pages to see what a format string** is and how it works. An example is as follows: Suppose you want to replace all occurrences of say, 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, I would fill in the Replace dialog box as follows:

```
Enter search string
(%d,%d,%d)
Enter replace 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. Scanf 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.

See also the file **README.edit** in the source tree for some more info.

## 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 builtin commands as well) in turn. If none of these produces a match, 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 show\_all\_if\_ambiguous variable in the Initialization file. If it is nonzero, 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 show\_all\_if\_ambiguous is set to zero, 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 five 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) and finally the mcfs (Midnight Commander file system), a network based file system.

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 ftpfs allows you to manipulate files on remote machines, to actually use it, you may try to use the panel command FTP 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:

```
ftp://[!][user[:pass]@]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 optional *pass* element, if present is the password used for the connection. This use is not recommended (nor keeping this in your hotlist, unless you set the appropriate permissions there, and then, it may not be entirely safe anyways).

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
```

To connect to sites behind a firewall, you will need to use the prefix `ftp://!` (ie, with a bang character after the double slash) to make the Midnight Commander use a proxy host for doing the ftp transfer. You can define the proxy host in the Virtual File System dialog box.

Another option is to set the `ftpfs_always_use_proxy` parameter in the initialization file. This will configure the program to always use the proxy host. If this variable is set, the program will do two things: consult the `@prefix@/lib/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.

If you are using the `ftpfs` code with a filtering packet router that does not allow you to use the regular mode of opening files, you may want to force the program to use the passive-open mode. To use this, set the `ftpfs_use_passive_connections` option.

The Midnight Commander keeps the directory listing in a cache. The cache expire time is configurable in the Virtual File System dialog box. This has the funny behavior that even if you make changes to a directory, they will not be reflected in the directory listing until you force a cache reload with the `C-r` key. This is a feature (when you think it's a bug, think about manipulating files on the other side of the Atlantic with `ftpfs`).

## 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:

```
tar:filename.tar[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:

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

The latter specifies the full path of the tar archive.

## 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 machine 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:/dev/sda2
```

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

## 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 menu, menusel, menuhot, menuhotsel, gauge; the dialog colors are: dnormal, dfocus, dhotnormal, dhotfocus; Help colors are: helpnormal, helpitalic, helpbold, helplink, helpslink; Viewer color is: viewunderline; Special highlighting mode: executable, directory, link, device, special.

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.

*gauge* determines color of filled part of the progress bar (gauge), which shows how many percent of files were copied etc. in a graphical way.

For file type highlighting mode *directory* specifies the color in which directories are shown; *executable* for executable files; *link* is used to represent links; *device* for character and block devices; *special* is for special files, such as FIFO and IPC sockets; *core* is for core files (see also the option **highlight\_mode** at the section on Special Settings).

The possible colors are: black, gray, red, brightred, green, brightgreen, brown, yellow, blue, brightblue, magenta, brightmagenta, cyan, brightcyan, lightgray and white.

## 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.

*drop\_menus.*

If this variable is set, when you press the F9 key, the pull down menus will be activated, else, you will only be presented with the menu title, and you will have to select the entry with the arrow keys or the first letter and from there select your option in the menu.

*ftpfs\_retry\_seconds.*

This value is the number of seconds the Midnight Commander will wait before attempting a reconnection to an ftp server that has denied the login. If the value is zero, the the program will not retry the login.

*ftpfs\_use\_passive\_connections.*

This option is by off default. This makes the ftpfs code use the passive open mode for transferring files. This is used by people that are behind a filtering packet router. This option just works if you are not using an ftp proxy.

*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 keypresses. 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.

*navigate\_with\_arrows.*

If this setting is turned on, then 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.

*nice\_rotating\_dash*

When on, this flag causes the commander to show a rotating dash as a work in progress indicator.

*old\_esc\_mode*

By default the Midnight Commander treats the ESC key as a key prefix (`old_esc_mode=0`), if you set this option (`old_esc_mode=1`), then 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*

set special treatment for '+', '-', '\*' in command line (select, unselect, reverse selection) only if command line is empty. No need to quote this characters in the middle of the command line. But we can not change selection when 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.

### *preserve\_uidgid*

If this option is set (the default), when logged in as root the default will be to preserve the UID and the GID of files. Some users prefer to disable this option, so that's why it's configurable.

### *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.

### *show\_all\_if\_ambiguous.*

By default the Midnight Commander pops up all possible completions if the completion is ambiguous if you press **M-Tab** for the second time, for the first time it just completes as much as possible and in the case of ambiguity beeps. If you want to see all the possible completions already after the first **M-Tab** pressing, set this option to 1.

### *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.

*highlight\_mode* By default all information on panels displayed with the same color. If this variable is set to 1, then *perm* or *mode* tokens in display format get ability to show access rights of the user to the shown file. Appropriate triplet of reading, writing and execution rights highlighted with the yellow ( *selected* ) color. In addition, if the variable is equal to 2, then all lines are displaying by the color according to their type (see Colors). Permissions highlighting also works in this mode.

### *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) or 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

The program will retrieve all of its information relative to the MCHOME environment variable, if this variable is not set, then it will fall back to the @prefix@ directory.

@prefix@/lib/mc.hlp

The help file for the program.

@prefix@/lib/mc/mc.ext

The default system-wide extensions file.

~/.mc/ext

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

@prefix@/lib/mc/mc.ini

The default system-wide setup for the Midnight Commander, used only if the user lacks his own ~/.mc/ini file.

@prefix@/lib/mc/mc.lib

Global settings for the Midnight Commander. Settings in this file are global to any Midnight Commander, it is useful to define site-global terminal settings.

~/.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.

@prefix@/lib/mc/mc.hint

This file contains the hints (cookies) displayed by the program.

@prefix@/lib/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. Each line is one entry. The lines starting with a slash are full directory names. The lines starting with a number have that many characters equal to the previous directory. If you want you may create this file by giving the command "find / -type d -print | sort > ~/.mc.tree". Normally there is no sense in doing it because the Midnight Commander automatically updates this file for you.

./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.nuclecu.unam.mx in the directory /linux/local and from Europe at sunsite.mff.cuni.cz in the directory /GNU/mc and at ftp.teuto.de in the directory /lmb/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.gnome.org/mc/>

## AUTHORS

Miguel de Icaza (miguel@roxanne.nuclecu.unam.mx), Janne Kukonlehto (jtklehto@paju.oulu.fi), Radek Doulik (rodo@earn.cvut.cz), Fred Leeflang (fredl@nebula.ow.org), Dugan Porter (dugan@b011.eunet.es), Jakub Jelinek (jj@sunsite.mff.cuni.cz), Ching Hui (mr854307@cs.nthu.edu.tw), Andrej Borsenkow (borsenkow.msk@sni.de), Norbert Warmuth (nwar-muth@privat.circular.de), Mauricio Plaza (mok@roxanne.nuclecu.unam.mx) and Paul Sheer (psheer@icon.co.za) are the developers of this package; Alessandro Rubini (rubini@ipvis.unipv.it) has been especially helpful debugging and enhancing the program's mouse support, John Davis (davis@space.mit.edu) also made his S-Lang library available to us under the GPL and answered my questions about it, and the following people have contributed code and many bug fixes (in alphabetical order):

Adam Tla/lka (atlka@sunrise.pg.gda.pl), alex@bcs.zp.ua (Alex I. Tkachenko), Antonio Palama, DOS port (palama@posso.dm.unipi.it), Erwin van Eijk (wabbit@corner.iaf.nl), Gerd Knorr (kraxel@cs.tu-berlin.de), Jean-Daniel Luiset (luiset@cih.hcuge.ch), Jon Stevens (root@dolphin.csudh.edu), Juan Francisco Grigera, Win32 port (j-grigera@usa.net), Juan Jose Ciarlante (jjciarla@raiz.uncu.edu.ar), Ilya Rybkin (rybkin@rouge.phys.lsu.edu), Marcelo Roccasalva (mfroccas@raiz.uncu.edu.ar), Massimo Fontanelli (MC8737@mclink.it), Pavel Roskin (pavel\_roskin@geocities.com), Sergey Ya. Korshunoff (root@seyko.msk.su), Thomas Pundt (pundtt@math.uni-muenster.de), Timur Bakeyev (timur@goff.comtat.kazan.su), Tomasz Cholewo (tjchol01@mecca.spd.louisville.edu), Torben Fjerd- ingstad (torben.fjerd- ingstad@uni-c.dk), Vadim Sinolitis (vvs@nsrd.npi.msu.su) and Wim Osterholt (wim@djo.wtm.tudelft.nl).

## 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-bugs@rox- anne.nuclecu.unam.mx.

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