

Beginner's Emacs



If you're reading this from the screen, you can pick a topic or just scroll through the text

[Starting](#)[Stopping](#)[Panic](#)[Moving](#)[Inserting](#)[Deleting](#)[Searching](#)[Saving](#)[Multiple files](#)[Customising](#)

What is Emacs?

Emacs is a text editing program. It's also a lot of other things, some of which you may discover and investigate later, like a mailer, newsreader, [WWW browser](#), [structured editor](#), directory manager and software development system. The name stands for **E**ditor **mac**ros (a macro is a miniature program which allows a complex function to be built up out of small, simple steps: like making a single keystroke do a lot of work). In addition to this document, there is [a lot more available online](#).

It's worth spending a little time learning how to use Emacs: because it is so common, so portable, and so powerful (even now outside the Unix world where it grew up), learning it will give you a skill which will stand you in good stead for the rest of your life.

Emacs is GNU software. The [GNU project](#) (it stands for `GNU's Not Unix') aims to provide a complete operating environment in a portable form, distributable free of charge.

There's an online tutorial you can use (once you have Emacs running) by typing `^h t` (the `^` prefix means you hold down the `Ctrl` key while pressing the letter that follows).

Getting running

[Emacs is available for many computer systems](#), including UNIX, VMS, Microsoft Windows and Apple Macintosh.

Emacs on a terminal system

First, make sure your terminal or emulator (eg ProComm, Windows Terminal, Zterm, VersaTerm, Telnet or whatever) is set correctly (conventionally, `9600/N/8/1 VT100`), and that you have told UNIX what you're using (type `set term=vt100` when you log in).

You get Emacs running by typing the command `emacs` followed by a filename (the name of an existing file, or make one up if this is to be a new file), for example:

Emacs in a windowing system

On a UNIX or VMS environment under the X Window system, Emacs can be made available as an icon or a menu entry, so you just click on it. It is also possible to start it from an xterm or shell window by typing `emacs&`

Emacs is also available for Microsoft Windows 3.11 and Windows 95. In these cases you double-click the icon and an Emacs window will start, in the same way

as any Windows program.

In either case you can open a file from the File menu in the normal way.

Because of Emacs' wide portability, it can work even on systems with no mouse and menus. It has therefore become conventional to use keyboard control characters for many functions, as this avoids you having to move your hand away from the keyboard to use the mouse for common functions, and then back again. In these examples, the notation \^x means 'Control-X', which you get by holding down the `Ctrl` or `Control` key and pressing the named letter key once, then releasing both. More details below.

Emacs will clear the screen and you are ready to type in your text. If this is an existing file, the first screenful will be displayed. You can also just type `emacs` on its own, and create a file, then give it a name when you save it later.

To open a file (Emacs uses the term 'visit'), type \^x\^f followed by the filename. It displays the current directory name as a prefix, but you can backspace over this if you want to load a file from another directory. If the file exists, it is displayed. If it does not, it is created.

Emacs lets you use a trick called 'completion' on all filenames or commands you type: if you type the first few characters and then press the `TAB` key, Emacs will complete it for you if what you have typed is unique. If not, it will let you press `TAB` again, and display a list of all valid completions for you to pick one of.

If you are editing automatically from within another program, for example mail (*Elm*) or news (*Tin*), then you don't have to type the `emacs` command, it's done for you: you just get a message file to edit.

Controlling things

If you are using Emacs in a native windowing version (eg X Windows, MS-Windows, Mac), then the normal window controls apply, and you use the mouse to operate the menus in the conventional way. The plain DOS version does not have its own menus, but you can use the mouse to mark text, and the function keys can be programmed.

If you are dialled in (or telnetted in) to a remote machine, you may have to use Emacs in its raw form (that is, without any bells and whistles of your own cooking). In this case, programs (including Emacs) *cannot* use the PC or Mac function keys because these keys can't always be trusted to work the same way from machine to machine, and some terminals don't even have any. This is particularly true if you log into a machine using a terminal program (an 'emulator' like ProComm, Kermit, Windows Terminal, Zterm, VersaTerm *etc*), each of which does different and incompatible things with the function keys.

Until you get better acquainted with your version of Emacs, stick with the standard key setups (called 'bindings') below. In these instructions:

- the \^ mark in front of a character means 'Control', that is, you hold down the `Ctrl` key while you press the given character, then release both;
- the symbol `RET` means you press the 'Return' or 'Enter' key;
- the symbol `SP` means 'space': you press the spacebar at the bottom of the keyboard;
- the symbol `ESC` means you press the 'Escape' key;
- *slanted* text means you must type meaningful values of your own.

If you are going to be typing more than a trivial amount of text, you probably want to set Emacs to *Text Mode* to start with: type `ESC x text-mode RET`.

Leaving Emacs

To exit from Emacs, press `^x^c`. If you have not saved your file(s), Emacs will ask if you want to save them: answer with `y` for yes or `n` for no to each one, as prompted.

See below for how to abort Emacs in an emergency.

Don't Panic

If something goes wrong (remember Emacs has many more commands than the ones shown here), don't worry:

- you can press `^g` to cancel whatever it's doing and go back to editing;
 - you can undo the last thing you did by typing `ESC x undo RET`
 - If the screen gets messed up with modem noise or other interruptions, pressing `^L` will refresh it.
 - For brief help, press `^h` at any time. You can get more specific, context-sensitive help by typing `ESC x describe-mode RET`, and the entire manual is online as well (type `ESC x info RET`).
 - In an emergency, you can abort Emacs pressing `^z`, but this will not save any files.
-

Moving around

Arrow keys will work if your [terminal type](#) is set correctly, otherwise:

- Next line (down): `^n`
- Previous line (up): `^p`
- Forward (right 1 character): `^f`
- Backward (left 1 character): `^b`
- New line: `RET`
- Forward one word: `ESC f`
- Backward one word: `ESC b`
- Next screenful (down a page): `^v`
- Previous screenful (up a page): `ESC v`
- Jump to start of line: `^a`
- Jump to end of line: `^e`

Marking blocks

The cursor shows your present location in the document. Like most editors, Emacs maintains its internal pointer (called ``point'`) *between* characters, not *on* a character, so on systems which place the cursor *on* a character, ``point'` is between that character and the one *before* it.

Marking with the mouse works as normal: hold down the left-hand button and drag the rodent to wherever you want to go. If you are doing keyboard operation, you can mark a location by pressing `^@` or `^SP` (control-space) and then use the arrow keys or a search to move the cursor elsewhere. A marked location is referred to as ``mark'`.

Block operations work on text between ``mark'` and ``point'`, that is, between the location you marked, and the current location of point.

- Block delete (cut): `^w` (``wipe'`)
 - Block copy: `ESC w`
 - Block paste: `^y` (``yank'`)
-

Inserting

Unlike old twin-mode editors like `vi`, you do not have to use an `insert` command, just start typing.

See the section on [Marking](#) for block insertion (``paste'`) operations.

Reformatting

To make text wrap automatically at the right margin, type `ESC x auto-fill RET`. You can use `ESC q` to reformat a single paragraph,

To set a new lefthand margin (for an indented block, assuming you are in Text mode), start a new paragraph (press `RET`) and type the number of spaces you want to indent, then press `^x .` (that's a period or fullstop). Then type your paragraph. To reset it, move to the lefthand edge and type `^x .` again.

Deleting

- Delete forwards: `^d` (deletes character following point)
- Delete backwards: `del` or `backsp` (deletes character before point)
- Delete to end of line: `^k` (deletes *text* to the end of line: to remove the line entirely, press `^k` twice. If the line was blank, once is enough.)
- Delete word forwards: `ESC d`
- Delete word backwards: `ESC del`
- Undelete last deletion: `^y` (``yank'`)

See the section on [Marking](#) for block copy and block deletion (``cut'`) operations.

Searching

- Incremental search forwards: `^s word`
- Incremental search backwards: `^r word`

- Repeat the search (‘find next’): `^s` or `^r` (as appropriate)
- Stop incremental search: `ESC` or `RET`

With an incremental search, as you type *each character* of the *word*, Emacs immediately starts searching, so you may find what you are looking for before completing the word. Press `RET` to stop searching, or `^g` to abort the search and return to where you started.

To do a non-incremental search, type `^s` or `^r` followed by `RET` and then the word or characters you are looking for.

Do not use `^s` over old dial-up lines running software flow-control (XON/XOFF), as `^s` gets swallowed by the communications software and interpreted as a ‘halt display’ instruction. If this happens, you can usually restart the display with `^q`. If you are in this position, either use the command `ESC x search-forward RET` or assign the search function to another key (see below) or assign a software key in your communications program to transmit this command (eg ProComm's `altG` or Windows Terminal's keypad).

Replacing

To do global search-and-replace, type `ESC x replace-string RET` then type the text to search for (and press `RET`) then the text to replace it with (and press `RET`). If you are going to do a lot of these, you can assign the command to a keystroke to make it faster to use. There are other more complex pattern-matching searches (using Unix-like ‘regular expressions’) that you can learn later.

Saving

Emacs automatically makes a backup (safety) copy of your file every 300 keystrokes (this frequency can be changed). To save your file explicitly (a good idea from time to time, especially if you're logged in over a phone line), type `^x^s`. **Do not use `^s` over old dial-up lines running software flow-control (XON/XOFF), as `^s` gets swallowed by the communications software and interpreted as a ‘halt display’ instruction. If this happens, you can usually restart the display with `^q`.**

If you *do* get disconnected, don't panic: when you log in again and try to edit the same file, Emacs will automatically tell you if there was a backup file (because of the autosave). To use it, type `ESC x recover RET`, and Emacs will ask you if want to carry on where you left off: answer `yes`.

If you were editing via Mail or News, your file will have been saved under a temporary name, so Emacs won't find it. If you are using Unix, it will probably be in your `~/tmp` subdirectory as a file called `snd.nnnnn` or something similar.

Multiple files

Emacs can edit many files simultaneously, each one in its own ‘buffer’ (memory area), and you can divide the edit window into subwindows, each displaying a different file. You can divide your display into subwindows, each showing a different file, even if you are connected using a single-screen system (for example, using a terminal emulator). On windowing systems you can also

create multiple frames, each of which can be subdivided.

- To open ('visit') another file: `^x^f` (type the filename)
 - To close (kill) a buffer: `^xk` (prompts for saving)
 - To switch buffers: `ESC x switch` (type the filename)
 - To display two windows: `^x2` (then switch buffers in one of them)
 - To move between windows: `^xo`
 - To remove a window from display: `^x0` (keeps the buffer active)
-

Customising

You can assign any function to any key. To identify the functions available, use the `ESC x apropos RET` command and type the keyword you are looking for. When you have identified the command name, type `ESC x global-set` (or `local-set` if you want it restricted to the current buffer) `RET` and then type the key to assign (use an otherwise unwanted control-key) followed by the function name.

You can save this kind of setup permanently by editing your `~/ .emacs` file (Unix and VMS) or `_emacs` file (DOS and others) to include such assignments. Details are in the manual.

Documentation

The full (450-page) manual is available from the Computer Centre, price £25, or as a PostScript file free of charge (bring a disk) to print on your own printer. You can download the print file from <ftp://www.ucc.ie/pub/net/emacs.ps>