

Emacs Reference

Emacs is invoked by the Unix command `emacs` followed by the filename you wish to edit. XEmacs is invoked by the Unix command `xemacs`. From a user's point of view, they are essentially the same. In this class, we will preferentially use XEmacs because Sun has done some customization to improve its handling of Fortran code on our machines. In an X-window session, either will normally generate a new window, so it is most useful to run it in the "background" using the `&` symbol at the end of the command, as in

```
xemacs programname.f95 &
```

If you get a "command not found" message from Unix, you will need to fix your path, but try `/opt/bin/emacs` instead or `/opt/bin/xemacs` instead of plain `xemacs` until you get your path fixed.

Control functions, such as `[control]-X` are obtained by holding down the `[control]` key while pressing the other indicated key. Meta functions are obtained by holding down a special "meta" key while pressing the other indicated key. Mac OS X computers running OS X with Apple's X11 implementation can use the "command" key—also known as the "apple" key—as a meta key *in XEmacs on Strauss*. In the Emacs version that runs directly on a Mac, the meta key is the "option" key. In Windows, the `[Alt]` key may work as meta. Dedicated X terminals use `[meta]` keys with `[⊞]` symbols found on each side of the space bar, but these are now rare. If none of these options works, then pressing *and releasing* the `[esc]` key before pressing the other indicated key will produce a meta effect. Because the meta key varies, we just use `[meta]` in this writeup, even though you will probably never see a keyboard (outside of a museum) with that key.

Moving the cursor

In Emacs the **text cursor** is different from the **mouse cursor**. The text cursor is actually always *between* two characters, never *on* a character, and having it appear as a vertical line is an option. Since a vertical line cursor is hard to find, it may be set to be a *block cursor* that covers an entire letter. The block cursor is actually to the left of the letter it covers.

Emacs dates back to terminals that did not have mice and arrow keys. Implementation of mouse clicks, arrow keys, and other special keyboard keys varies with the Emacs implementation. The control and meta functions described later are part of the original design of Emacs and do not vary.

Arrows, special keys, the mouse: system-dependent cursor moves

- Arrow keys do what you would expect. End-of-line markers are not visible in Emacs or XEmacs, but the cursor will not move past the end of a line unless there are blank space characters there, so right-arrow past the end of the line moves to the beginning of the next line.
- `page up` and `page down` keys usually work as expected. `[control]-home` and `[control]-end` will go to the beginning or end of a file on XEmacs through X11. `home` and `end` go to the beginning or end of a line without the applied `[control]` key. These work differently on Mac Emacs.
- Clicking the left mouse button anywhere will move the cursor to the position of the mouse pointer. The left mouse button can also be used to drag the scroll bar. Clicking the left mouse button anywhere on the scroll bar will move the file one screen of text towards the mouse cursor in the bar, and the center mouse button will move the file so that the scroll bar is at the mouse position. Scroll wheels on the mouse do not usually work on XEmacs through X11 on a Unix server.
- In an X window, a "scroll bar" will appear on the right side of the window, showing the size and location of your screen relative to the entire file.

Control and Meta keys: original and universal cursor moves

control-A Move to beginning of line (mnemonic: 1st letter = beginning).

control-E Move to end of line.

meta-**→** Move forward one word.

meta-**←** Move backward one word.

meta-A Move to beginning of sentence.

meta-E Move to end of sentence.

(Emacs assumes that sentences end with a period followed by two spaces.)

control-V Move down (forward) one screen view.

meta-V Move up (backward) one screen view.

control-L Refresh screen, center on line containing the cursor.

meta-< Go to top of file.

meta-> Go to end of file.

meta-G Go to a particular line number – a prompt to type a number will be in the mini-buffer at the bottom. (Most compiler error messages will refer to a line number.)

Destruction and Recovery

delete Left Delete. Delete character to the left (backspace and delete).

control-D Right Delete. Delete character to the right of the cursor. (When using a block cursor, this will be the character that appears to be *under* the cursor.)

control-K Delete (kill) everything on line to right of cursor.

meta-K Delete (kill) to end of sentence.

control-Y Recover (yank) everything in most recently killed block. Undelete. Alternately: click the center mouse button in the new location.

control-@ Mark beginning or end of block to kill. (“Mark set” should appear at bottom.) Alternately: drag the mouse with the left mouse button pressed.

control-W Delete everything (wipe) between current cursor position and where the mark was set.

meta-W Mark everything between the current cursor position and where the last mark was set as the most recently killed block of text, but do not actually kill it.

Note that “killing” a block of text with a series of **control**-K or a pair of **control**-@ – **control**-W commands, and then “yanking” or undeleting that text in another location, is the primary way to move a block of text from one location to another. Using **control**-@ – **meta**-W to mark the text and then yanking it elsewhere is the primary way to copy a block of text into multiple locations.

Getting out

control-X **control**-C Exit Emacs. You will be prompted on whether or not to save any files you have worked on.

control-X **control**-S Save file to disk without leaving Emacs. (Especially useful if Emacs is running on a background window on an X terminal, as you can save the current version of a program, and then test it in another window without leaving the editor.)

Miscellaneous

- `control-T` Transpose letters (switch positions of the letters on each side of the cursor (before and under a block cursor)).
- `meta-T` Transpose words.
- `control-G` Quit whatever Emacs function you may be in the middle of.
- `control-S` Incremental search forward. `esc` to terminate search.)
- `control-R` Incremental search backward (reverse).
- `meta=` Count how many lines are in the current marked block.
- `meta-Q` Rejustify paragraph (*very* bad idea in programming language code or data entry—useful for text).

Multiple windows and files

If you are working in X-window sessions, you can always have multiple windows of Emacs sessions going. Switch between them with the mouse as you would between any other windows. Emacs has some prototypical windowing features that can be used either within a single X terminal window or on terminals that do not support windows. Nearly all of the following can be found either under the File menu or Buffers menu.xf

- `control-X 3` Split screen vertically into two buffers (nice for comparing files if you make the window very large).
- `control-X 2` Split screen horizontally into two buffers. In XEmacs, you can adjust the sizes of the two buffers by dragging the separator bar with the left mouse button.
- `control-X 1` Go to single screen mode. (Some error conditions will cause a second buffer to appear in which to print the error message. This happens very often at startup. You need this command to get back to the single buffer.)
- `control-X O` Go to **o**ther window (used only if in a split screen mode). (“Oh,” not “zero.”)
- `control-X control-F` Find a new file to edit, leaving the current file you are working on in another buffer. You can use `control-X - B` to switch back and forth between two active buffers, or with a large screen, you can have both files on the screen simultaneously in different Emacs windows.
- `control-X B` Go **b**ack to editing another file that has already been pulled into Emacs.
- `control-X I` Insert an existing file from disks into your working file at this point. (You will be prompted for a file name.)

Command-Line options

Typing `meta-X` moves the cursor to the bottom of the screen next to a “M-x” prompt. At that point, you can enter various commands. Emacs contains a huge list of commands, but here are a few that are useful right away. (You will find out that you don’t have to type an entire command most of the time—just type enough of each word that it becomes uniquely identifiable, and Emacs will fill in the rest.)

- `replace string` A global “search-and-replace” command. You will be prompted for a string to search for and a string that you wish to change into. Emacs will then change all occurrences of the first string into the second string, starting from the current cursor position and proceeding to the end of the file.
- `query replace` A “search-and-replace” command that offers you the choice of replacing or not replacing for each occurrence of the search string.

- kill rectangle** Similar to `control-W` except that it notes the *column* positions of the current mark and the current cursor position, and only deletes text between those two columns. Can be used to delete a column of data, for example.
- auto fill mode** Automatically starts new lines at spaces between words near the end of a line. Good for text, not for programming languages.
- overwrite-mode** Toggles between the default “insert mode,” where typing over text moves existing text to the right, and “overwrite mode,” where new text can be typed over the old as a replacement.
- recover file** Used to switch from the current file to an autosaved version.
- info** Puts you into the menu-driven help system. The complete Emacs manual and an introductory tutorial are available online via info. Like most Unix-based documentation, this is not friendly.

Fortran mode, major modes

If you start Emacs with a file whose names ends in `.f90` or `.f95` then Emacs assumes you are editing modern Fortran code with free-form source and sets a few additional keys you might find useful. The word “F90” will appear in the status bar at the bottom of the screen when this occurs, and an F90 pull-down menu will appear at the top. The following commands either work only in Fortran mode, or they work differently in Fortran mode than in other modes.

- `tab` This key works in all modes, but in Fortran mode it is set to indent an entire line either to the same level as the previous line, or further if the previous line defines a block (such as a `DO` or `IF` statement). The `tab` key will also reduce the indentation of an `END IF`, `END DO`, or `END` statement to match the indentation of corresponding block beginning.
- `Line Feed` In Fortran mode, this command does a `return` followed by a `tab` to where the next line of Fortran code should start. (Not all keyboards have this key.)
- `meta-control-Q` After typing `END` will cause the editor to look for a corresponding `IF`, `DO`, `PROGRAM`, or `SUBROUTINE` statement, complete the `END` statement appropriately, and reindent the entire block. Very useful and strongly recommended code checking tool.
 -) After typing a right parenthesis, `)`, the cursor will temporarily, for a about a second, jump to the matching left parenthesis—another very handy code checking tool, as Fortran requires parentheses to appear as matched sets in most circumstances.

It is possible to change the indenting, continuation character used, and various other aspects of Fortran mode by setting Emacs variables. This is an advanced topic.

When in F90 mode, the editor should automatically perform syntax coloring, in which different types of Fortran syntax (keywords, variable names, character strings, comments) appear in different colors. If this does not work for you, it may need to be turned on under the `Options` menu.

Emacs has editing modes for most programming languages, and it automatically enters them if it detects the standard filetype. Thus, if you start a file called `something.py` it will assume you are editing a Python program and set your syntax coloring and indenting rules accordingly. These modes are called “major modes” and will be indicated in the status line near the bottom of the buffer.