

UNIX EDITORS

vi EDITOR

Command-Line Syntax:

Starting the Section:

<i>vi file</i>	Open the file for editing
<i>vi +50 file</i>	Open the file at line 50 for editing
<i>vi +/pattern file</i>	Open file (where pattern appears the first time) for editing.

Ending the section:

:wq	Save the modifications and quit.
:q!	Quit without saving.

vi Modes:

It has 2 modes: command mode and insert mode (edit file).

Syntax of the Commands:

[n] operator [m] motion

The basic edit operators are:

d delete

y yank (copy)

If the current line is the object of operation, the motion is the same as the operator, then use dd and yy. Otherwise, it will work like d} for example, where you delete until the next paragraph. The n and m are the number of times to performe the operation, so 5yy means to copy 5 lines.

Status line Commands:

These commands are not echoed on the screen as you input them. The status line (in the bottom of the screen) is used for editing them.

/ Search forward for a pattern

? Search backward for a pattern

: Invoke an ex command

vi Commands:

<i>Command</i>	<i>Action</i>	<i>Description</i>
/pattern	search	Search forward for pattern. End with ENTER.
n		Repeat previous search.
%		Find match of current parenthesis, brace or bracket.
fx		Move cursor forward to letter x on current line.
Fx		Move cursor backward to letter x on current line.
:num	line number	Move to line number.
=%	align text	Position the cursor in the line of an opening statement curly bracket, then press ESC and =% to align the lines.
i	insert	Insert before cursor.
dd	delete	Delete current line.
d\$		Delete from current position to end-of-line.
d}		Delete up to next paragraph.
p	paste	Insert last deleted or yanked text after the cursor.
u	undo	Undo last change.
.	repeat	Repeat last change.
yy	copy	Copy current line.
y\$		Copy rest of line.

ex EDITOR

Syntax of ex commands:

: [address] command [options]

Address:

Especify the address using a range of lines: (x, y).

<i>Address</i>	<i>Meaning</i>
1,\$ (or %)	All lines in the file.

<i>Address</i>	<i>Meaning</i>
x,y	Lines x to y.
num	Absolute line number.
0	Top of line.
.	Current line.
\$	Last line.
/pattern/	Forward to line matching pattern.

Options:

<i>Option</i>	<i>Meaning</i>
!	Variant for of the command, must come immediately after the command.
count	The number of times the command must be repeated. Must come after, to not be interpreted as an address.
file	File affected by the command. % stands for current file and # is the previous file.

Commands:

<i>Command</i>	<i>Meaning</i>
[address] co destination	Copy the lines specified in the address to destination. Examples: :1,10 co 50 (copy the first 10 lines to just after line 50)
[address] d [register]	Delete the lines specified in the address. Examples: :/Part I/,/Part II/ -1d (delete to line above Part II) :/main/+d (delete line below main) :.,\$d x (delete from the actual line to last line, and store in register x)
[address] m destination	Move the lines specified in the address to destination. Examples: :1,10 m 50 (move the first 10 lines to just after line 50)
[address] pu [char]	Put the previous deleted from named register char to the line specified by address.

<i>Command</i>	<i>Meaning</i>
[address] s [/pattern/replacement/] [options] [count]	Substitute the first instance of pattern on each specified lines (address) with the replacement. Count specifies the number of line to execute starting to count from address. Options: c Prompt for confirmation before each substitution. g Substitute all instances of pattern on each line (global) Examples: :1,10s/yes/no/g (Substitute in first 10 lines) :%s/[Hh]ello/Hi/gc (confirm global substitutions to be executed in all lines)
[address] w[!] [[>>] file]	Write lines specified by address to file. If >> is used, then append to file. The oprtion ! means to force the writing process. :1,10w filename (copy first 10 lines to finename) :50w >> filename (append line 50 to filename)

sed EDITOR

The sed (stream editor) is a noninteractive editor. It interprets a script and performs the actions. It is a useful editor for editing many files automatically.

Command-Line Syntax:

```
sed 'command' file(s)
sed -f scriptfile file(s)
```

Syntax of sed Commands:

```
[address [,address]] [!]command [arguments]
```

<i>Address</i>	<i>Meaning</i>
/pattern/	Lines that match pattern. /BSD/d (delete lines conatining BSD)
N	Line number N.
\$	Last line.
no address	Each input line. s/xx/yy/g (substitute on all lines)

<i>Address</i>	<i>Meaning</i>
One address	Any line matching the address
Two comma-separated addresses	First matching line and all succeeding lines up to and including a line matching the second address. /^begin/,/^end/p (print between begin and end, inclusive)
An address followed by !	All lines that do NOT match the address.

Commands:

<i>Command</i>	
[address] a \text	Append text following each line that match the address. If text extends more than one line, use a backslash (\) at the end of the line to begin the new line. Last line does NOT have the \ (change line). \$a\ This is the first line of text\ second line\ last line.
[address [,address]] c \text	Change the lines selected by the addresses with text. Text follow the same rules as used for append (a\ 1,100c\ first line\ second line.
[address [,address]] d	Delete the lines specified by the address.
[address [,address]] s / pattern/replacement/ [flags]	Substitute pattern to replacement on each line addressed. The flags are: n replace the nth instance of pattern in each line instead of the 1st (default) g replace all instance of pattern on each line addressed (global).
[address [,address]] y / abc/xyz/	Translate characters. Change every instance of <i>a</i> to <i>x</i> , <i>b</i> to <i>y</i> , <i>c</i> to <i>z</i> , etc.