

Ex Quick Reference

Entering ex

| | |
|---|---------------------------------|
| % ex <i>name</i> | edit <i>name</i> , start at end |
| % ex + <i>n</i> <i>name</i> | ... at line <i>n</i> |
| % ex -t <i>tag</i> | start at <i>tag</i> |
| % ex -r | list saved files |
| % ex -r <i>name</i> | recover file <i>name</i> |
| % ex <i>name</i> ... | edit first; rest via :n |

Ex states

| | |
|-------------|--|
| Command | Normal and initial state. Input prompted for by : . Your kill character cancels partial command. |
| Insert | Entered by a i and c . Arbitrary text then terminates with line having only . character on it or abnormally with interrupt. |
| Open/visual | Entered by o or v , terminates with q or two successive interrupts. |

Ex commands

| | | | | | |
|--------|-----------|----------|------------|-------------------|--------------|
| append | a | next | n | substitute | s |
| args | ar | number | nu | undo | u |
| change | c | open | o | version | ve |
| copy | co | preserve | pre | visual | vi |
| delete | d | print | p | write | w |
| edit | e | put | pu | yank | ya |
| file | f | quit | q | <i>window</i> | z |
| global | g | read | re | <i>escape</i> | ! |
| insert | i | recover | rec | <i>lshift</i> | < |
| join | j | rewind | rew | <i>print next</i> | CR |
| list | l | set | se | <i>resubst</i> | & |
| mark | ma | shell | sh | <i>rshift</i> | > |
| move | m | source | so | <i>scroll</i> | ^D |

Ex command addresses

| | | | |
|-------------------|------------------|-------------|---------------------------|
| <i>n</i> | line <i>n</i> | <i>lpat</i> | next with <i>pat</i> |
| . | current | <i>?pat</i> | previous with <i>pat</i> |
| \$ | last | <i>x-n</i> | <i>n</i> before <i>x</i> |
| + | next | <i>x,y</i> | <i>x</i> through <i>y</i> |
| - | previous | <i>'x</i> | marked with <i>x</i> |
| + <i>n</i> | <i>n</i> forward | <i>''</i> | previous context |

Specifying terminal type

% **setenv** **TERM** *type* *cs**h* and all version 6
\$ **TERM**=*type* *sh* in Version 7

Some terminal types

| | | | | |
|------|------|--------|-------|-------|
| 2621 | 43 | adm31 | dw1 | i100 |
| 2645 | 733 | adm3a | dw2 | mime |
| 300s | 745 | c100 | gt40 | owl |
| 33 | act4 | dm1520 | gt42 | t1061 |
| 37 | act5 | dm2500 | h1500 | vt52 |
| 4014 | adm3 | dm3025 | h1510 | |

Initializing options

| | |
|---------------------------------|--|
| .exrc | place set 's here in HOME directory |
| set <i>x</i> | enable option |
| set no <i>x</i> | disable option |
| set <i>x=</i> <i>val</i> | give value <i>val</i> |
| set | show changed options |
| set <i>x</i> ? | show value of option <i>x</i> |

Useful options

| | | |
|-------------------|------|-----------------------------|
| autoindent | ai | supply indent |
| beautify | bf | discard nonprinting chars |
| ignorecase | ic | in scanning |
| lisp | | () { } are s-exp's |
| list | | print ^I for tab, \$ at end |
| magic | | . [* special in patterns |
| number | nu | number lines |
| paragraphs | para | macro names which start ... |
| redraw | | simulate smart terminal |
| sections | sect | macro names ... |
| shiftwidth | sw | for < >, and input ^D ^T |
| showmatch | sm | to) and } as typed |
| slowopen | slow | choke updates during insert |
| wrapscan | ws | around end of buffer? |
| wrapmargin | wm | automatic line splitting |

Scanning pattern formation

| | |
|-------------------------|-----------------------------------|
| ↑ | beginning of line |
| \$ | end of line |
| . | any character |
| \< | beginning of word |
| \> | end of word |
| [<i>str</i>] | any char in <i>str</i> |
| [↑ <i>str</i>] | ... not in <i>str</i> |
| [<i>x</i> - <i>y</i>] | ... between <i>x</i> and <i>y</i> |
| * | any number of preceding |

Vi Quick Reference

Starting vi

| | |
|----------------------|--------------------------|
| % vi <i>name</i> | edit <i>name</i> at top |
| % vi + <i>n name</i> | ... at line <i>n</i> |
| % vi + <i>name</i> | ... at end |
| % vi -r | list saved files |
| % vi -r <i>name</i> | recover file <i>name</i> |
| % vi <i>name</i> ... | edit first; rest via :n |
| % vi -t <i>tag</i> | start at <i>tag</i> |

The display

| | |
|-----------|---|
| Last line | Error messages, echoing input to : / ? and !, feedback about i/o and large changes. |
| @ lines | On screen only, not in file. |
| ~ lines | Lines past end of file. |
| ^x | Control characters, ^? is delete. |
| tabs | Expand to spaces, cursor at last. |

Vi states

| | |
|-----------|--|
| Command | Normal and initial state. Others return here. ESC (escape) cancels partial command. |
| Insert | Entered by a i A I o O c C s S R. Arbitrary text then terminates with ESC character, or abnormally with interrupt. |
| Last line | Reading input for : / ? or !; terminate with ESC or CR to execute, interrupt to cancel. |

Counts before vi commands

| | |
|--------------------|-----------------------|
| new window size | : / ? [[]] ^F ^B ` ` |
| line/column number | z G |
| scroll amount | ^D ^U |
| replicate insert | a i A I |
| repeat effect | most rest |

Simple commands

| | |
|----------|---------------------------|
| dw | delete a word |
| de | ... leaving punctuation |
| dd | delete a line |
| 3dd | ... 3 lines |
| itextESC | insert text <i>abc</i> |
| cwnewESC | change word to <i>new</i> |
| easESC | pluralize word |
| xp | transpose characters |

Interrupting, cancelling

| | |
|-----|-----------------------------------|
| ESC | end insert or incomplete cmd |
| ^? | (delete or rubout) interrupts |
| ^L | reprint screen if ^? scrambles it |

File manipulation

| | |
|------------------|-----------------------------------|
| :w | write back changes |
| :wq | write and quit |
| :q | quit |
| :q! | quit, discard changes |
| :e <i>name</i> | edit file <i>name</i> |
| :e! | reedit, discard changes |
| :e + <i>name</i> | edit, starting at end |
| :e + <i>n</i> | edit starting at line <i>n</i> |
| :e # | edit alternate file |
| ^↑ | synonym for :e # |
| :w <i>name</i> | write file <i>name</i> |
| :w! <i>name</i> | overwrite file <i>name</i> |
| :sh | run shell, then return |
| :! <i>cmd</i> | run <i>cmd</i> , then return |
| :n | edit next file in arglist |
| :n <i>args</i> | specify new arglist |
| :f | show current file and line |
| ^G | synonym for :f |
| :ta <i>tag</i> | to tag file entry <i>tag</i> |
| ^] | :ta, following word is <i>tag</i> |

Positioning within file

| | |
|--------------------------|-------------------------------|
| ^F | forward screenful |
| ^B | backward screenful |
| ^D | scroll down half screen |
| ^U | scroll up half screen |
| G | goto line (end default) |
| / <i>pat</i> | next line matching <i>pat</i> |
| ? <i>pat</i> | prev line matching <i>pat</i> |
| n | repeat last / or ? |
| N | reverse last / or ? |
| / <i>pat</i> ! <i>n</i> | n'th line after <i>pat</i> |
| ? <i>pat</i> ?- <i>n</i> | n'th line before <i>pat</i> |
|]] | next section/function |
| [[| previous section/function |
| % | find matching () { or } |

Adjusting the screen

| | |
|------------------|-------------------------------|
| ^L | clear and redraw |
| ^R | retype, eliminate @ lines |
| zCR | redraw, current at window top |
| z- | ... at bottom |
| z. | ... at center |
| / <i>pat</i> /z- | <i>pat</i> line at bottom |
| zn. | use <i>n</i> line window |

Marking and returning

| | |
|-----------------|------------------------------------|
| <code>``</code> | previous context |
| <code>''</code> | ... at first non-white in line |
| <code>mx</code> | mark position with letter <i>x</i> |
| <code>`x</code> | to mark <i>x</i> |
| <code>'x</code> | ... at first non-white in line |

Line positioning

| | |
|-----------|-----------------------------------|
| H | home window line |
| L | last window line |
| M | middle window line |
| + | next line, at first non-white |
| - | previous line, at first non-white |
| CR | return, same as + |
| ^N | next line, same column |
| ^P | previous line, same column |
| LF | linefeed, same as ^N |

Character positioning

| | |
|--------------|--------------------------------------|
| ↑ | first non white |
| 0 | beginning of line |
| \$ | end of line |
| space | forward |
| ^H | backwards |
| h | same as ^H |
| fx | find <i>x</i> forward |
| Fx | f backward |
| tx | upto <i>x</i> forward |
| Tx | back upto <i>x</i> |
| ; | repeat last f F t or T |
| , | inverse of ; |
| | to specified column |
| % | find matching ({) or } |

Words, sentences, paragraphs

| | |
|----------|----------------------|
| w | word forward |
| b | back word |
| e | end of word |
|) | to next sentence |
| } | to next paragraph |
| (| back sentence |
| { | back paragraph |
| W | blank delimited word |
| B | back W |
| E | to end of W |

Commands for LISP

| | |
|----------|-----------------------------|
|) | Forward s-expression |
| } | ... but don't stop at atoms |
| (| Back s-expression |
| { | ... but don't stop at atoms |

Corrections during insert

| | |
|------------|---|
| ^H | erase last character |
| ^W | erases last word |
| erase | your erase, same as ^H |
| kill | your kill, erase input this line |
| \ | escapes ^H , your erase and kill |
| ESC | ends insertion, back to command |
| ^? | interrupt, terminates insert |
| ^D | backtab over <i>autoindent</i> |
| ↑^D | kill <i>autoindent</i> , save for next |
| 0^D | ... but at margin next also |
| ^Q | quote non-printing character |
| ^T | <i>shiftwidth</i> forward tab |

Insert and replace

| | |
|-----------|-----------------------------------|
| a | append after cursor |
| i | insert before |
| A | append at end of line |
| I | insert before first non-blank |
| o | open line below |
| O | open above |
| rx | replace single char with <i>x</i> |
| R | replace characters |

Operators (double to affect lines)

| | |
|-------------|------------------------|
| d | delete |
| c | change |
| < | left shift |
| > | right shift |
| ! | filter through command |
| = | indent for LISP |
| y | yank lines to buffer |

Miscellaneous operations

| | |
|----------|---------------------|
| C | change rest of line |
| D | delete rest of line |
| s | substitute chars |
| S | substitute lines |
| J | join lines |
| x | delete characters |
| X | ... before cursor |
| Y | yank lines |

Yank and put

| | |
|------------|-----------------------------|
| p | put back lines |
| P | put before |
| "xp | put from buffer <i>x</i> |
| "xy | yank to buffer <i>x</i> |
| "xd | delete into buffer <i>x</i> |

Undo, redo, retrieve

| | |
|------------|-----------------------------------|
| u | undo last change |
| U | restore current line |
| . | repeat last change |
| "dp | retrieve <i>d</i> 'th last delete |