NAME

ex, **vi**, **view** - text editors

SYNOPSIS

ex [**-FRrSsv**] [**-c** *cmd*] [**-t** *tag*] [**-w** *size*] [*file ...*]
vi [**-eFRrS**] [**-c** *cmd*] [**-t** *tag*] [**-w** *size*] [*file ...*]
view [**-eFrS**] [**-c** *cmd*] [**-t** *tag*] [**-w** *size*] [*file ...*]

DESCRIPTION

vi is a screen-oriented text editor. **ex** is a line-oriented text editor. **ex** and **vi** are different interfaces to the same program, and it is possible to switch back and forth during an edit session. **view** is the equivalent of using the **-R** (read-only) option of **vi**.

This manual page is the one provided with the **nex**/**nvi** versions of the **ex**/**vi** text editors. **nex**/**nvi** are intended as bug-for-bug compatible replacements for the original Fourth Berkeley Software Distribution (4BSD) **ex** and **vi** programs. For the rest of this manual page, **nex**/**nvi** is used only when it's necessary to distinguish it from the historic implementations of **ex**/**vi**.

This manual page is intended for users already familiar with **ex**/**vi**. Anyone else should almost certainly read a good tutorial on the editor before this manual page. If you're in an unfamiliar environment, and you absolutely have to get work done immediately, read the section after the options description, entitled *FAST STARTUP*. It's probably enough to get you going.

The following options are available:

-c *cmd* Execute *cmd* on the first file loaded. Particularly useful for initial positioning in the file, although *cmd* is not limited to positioning commands. This is the POSIX 1003.2 interface for the historic "+cmd" syntax. **nex**/**nvi** supports both the old and new syntax.

-e Start editing in ex mode, as if the command name were **ex**.

-F Don't copy the entire file when first starting to edit. (The default is to make a copy in case someone else modifies the file during your edit session.)

-R Start editing in read-only mode, as if the command name was **view**, or the **readonly** option was set.

-r Recover the specified files, or, if no files are specified, list the files that could be recovered. If no recoverable files by the specified name exist, the file is edited as if the **-r** option had not been specified.

-S Run with the **secure** edit option set, disallowing all access to external programs.

-s Enter batch mode; applicable only to **ex** edit sessions. Batch mode is useful when running **ex** scripts. Prompts, informative messages and other user oriented messages are turned off, and no startup files or environment variables are read. This is the POSIX 1003.2 interface for the historic "-" argument. **nex**/**nvi** supports both the old and new syntax.

-t *tag* Start editing at the specified *tag* (see ctags(1)).

-v Start editing in vi mode, as if the command name was **vi**.

-w *size* Set the initial window size to the specified number of lines.

Command input for **ex**/**vi** is read from the standard input. In the **vi** interface, it is an error if standard input is not a terminal. In the **ex** interface, if standard input is not a terminal, **ex** will read commands from it regardless; however, the session will be a batch mode session, exactly as if the **-s** option had been specified.

FAST STARTUP

This section will tell you the minimum amount that you need to do simple editing tasks using **vi**. If you've never used any screen editor before, you're likely to have problems even with this simple introduction. In that case you should find someone that already knows **vi** and have them walk you through this section.

vi is a screen editor. This means that it takes up almost the entire screen, displaying part of the file on each screen line, except for the last line of the screen. The last line of the screen is used for you to give commands to **vi**, and for **vi** to give information to you.

The other fact that you need to understand is that **vi** is a modeful editor, i.e., you are either entering text or you are executing commands, and you have to be in the right mode to do one or the other. You will be in command mode when you first start editing a file. There are commands that switch you into input mode. There is only one key that takes you out of input mode, and that is the <escape> key.

In this manual, key names are denoted with < and >, e.g., <escape> means the "escape" key, usually labeled "Esc" on your terminal's keyboard. If you're ever confused as to which mode you're in, keep entering the <escape> key until **vi** beeps at you. Generally, **vi** will beep at you if you try and do something that's not allowed. It will also display error messages.

To start editing a file, enter the following command:

```
$ vi file
```

The command you should enter as soon as you start editing is:

```
:set verbose showmode
```

This will make the editor give you verbose error messages and display the current mode at the bottom of the screen.

The commands to move around the file are:

h Move the cursor left one character.

j Move the cursor down one line.

k Move the cursor up one line.

l Move the cursor right one character.

<**cursor-arrows**>
The cursor arrow keys should work, too.

/text Search for the string "*text*" in the file, and move the cursor to its first character.

The commands to enter new text are:

a Append new text, after the cursor.

i Insert new text, before the cursor.

o Open a new line below the line the cursor is on, and start entering text.

O Open a new line above the line the cursor is on, and start entering text.

<**escape**> Once you've entered input mode using one of the **a**, **i**, **o** or **O** commands, use <**escape**> to quit entering text and return to command mode.

The commands to copy text are:

yy Copy the line the cursor is on.

p Append the copied line after the line the cursor is on.

The commands to delete text are:

dd Delete the line the cursor is on.

x Delete the character the cursor is on.

The commands to write the file are:

:w Write the file back to the file with the name that you originally used as an argument on the **vi** command line.

:w *file_name*
Write the file back to the file with the name *file_name*.

The commands to quit editing and exit the editor are:

:q Quit editing and leave **vi** (if you've modified the file, but not saved your changes, **vi** will refuse to quit).

:q! Quit, discarding any modifications that you may have made.

One final caution: Unusual characters can take up more than one column on the screen, and long lines can take up more than a single screen line. The above commands work on "physical" characters and lines, i.e., they affect the entire line no matter how many screen lines it takes up and the entire character no matter how many screen columns it takes up.

REGULAR EXPRESSIONS

ex/**vi** supports regular expressions (REs), as documented in re_format(7), for line addresses, as the first part of the **ex substitute**, **global** and **v** commands, and in search patterns. Basic regular expressions (BREs) are enabled by default; extended regular expressions (EREs) are used if the **extended** option is enabled. The use of regular expressions can be largely disabled using the **magic** option.

The following strings have special meanings in the **ex**/**vi** version of regular expressions:

- An empty regular expression is equivalent to the last regular expression used.

- '\<' matches the beginning of the word.

- ‘\>’ matches the end of the word.

- ‘~’ matches the replacement part of the last **substitute** command.

BUFFERS

A buffer is an area where commands can save changed or deleted text for later use. **vi** buffers are named with a single character preceded by a double quote, for example **"**<*c*>; **ex** buffers are the same, but without the double quote. **nex**/**nvi** permits the use of any character without another meaning in the position where a buffer name is expected.

All buffers are either in *line mode* or *character mode*. Inserting a buffer in line mode into the text creates new lines for each of the lines it contains, while a buffer in character mode creates new lines for any lines *other* than the first and last lines it contains. The first and last lines are inserted at the current cursor position, becoming part of the current line. If there is more than one line in the buffer, the current line itself will be split. All **ex** commands which store text into buffers do so in line mode. The behaviour of **vi** commands depend on their associated motion command:

- <**control-A**>, **h**, **l**, **,**, **0**, **B**, **E**, **F**, **T**, **W**, ^, **b**, **e**, **f** and **t** make the destination buffer character-oriented.

- **j**, <**control-M**>, **k**, **’**, **-**, **G**, **H**, **L**, **M**, _ and **|** make the destination buffer line-oriented.

- **$**, **%**, **‘**, **(**, **)**, /, **?**, **[[**, **]]**, **{** and **}** make the destination buffer character-oriented, unless the starting and end positions are the first and last characters on a line. In that case, the buffer is line-oriented.

The **ex** command **display buffers** displays the current mode for each buffer.

Buffers named ‘a’ through ‘z’ may be referred to using their uppercase equivalent, in which case new content will be appended to the buffer, instead of replacing it.

Buffers named ‘1’ through ‘9’ are special. A region of text modified using the **c** (change) or **d** (delete) commands is placed into the numeric buffer ‘1’ if no other buffer is specified and if it meets one of the following conditions:

- It includes characters from more than one line.

- It is specified using a line-oriented motion.

- It is specified using one of the following motion commands: <**control-A**>, **‘**<*character*>, **n**, **N**, **%**, /, **{**, **}**, **(**, **)**, and **?**.

Before this copy is done, the previous contents of buffer ‘1’ are moved into buffer ‘2’, ‘2’ into buffer ‘3’, and so on. The contents of buffer ‘9’ are discarded. Note that this rotation occurs *regardless* of the user specifying another buffer. In **vi**, text may be explicitly stored into the numeric buffers. In this case, the buffer rotation occurs before the replacement of the buffer’s contents. The numeric buffers are only available in **vi** mode.

VI COMMANDS

The following section describes the commands available in the command mode of the **vi** editor. The following words have a special meaning in the commands description:

bigword A set of non-whitespace characters.

buffer Temporary area where commands may place text. If not specified, the default buffer is used. See also *BUFFERS*, above.

count A positive number used to specify the desired number of iterations of a command. It defaults to 1 if not specified.

motion A cursor movement command which indicates the other end of the affected region of text, the first being the current cursor position. Repeating the command character makes it affect the whole current line.

word A sequence of letters, digits or underscores.

buffer and *count*, if both present, may be specified in any order. *motion* and *count*, if both present, are effectively multiplied together and considered part of the motion.

<**control-A**>
Search forward for the word starting at the cursor position.

[*count*] <**control-B**>
Page backwards *count* screens. Two lines of overlap are maintained, if possible.

[*count*] <**control-D**>
Scroll forward *count* lines. If *count* is not given, scroll forward the number of lines specified by the last <**control-D**> or <**control-U**> command. If this is the first <**control-D**> command, scroll half the number of lines in the current screen.

[*count*] <**control-E**>
Scroll forward *count* lines, leaving the current line and column as is, if possible.

[*count*] <**control-F**>
Page forward *count* screens. Two lines of overlap are maintained, if possible.

<**control-G**>

Display the following file information: the file name (as given to **vi**); whether the file has been modified since it was last written; if the file is read-only; the current line number; the total number of lines in the file; and the current line number as a percentage of the total lines in the file.

[*count*] <**control-H**>
[*count*] **h**

Move the cursor back *count* characters in the current line.

[*count*] <**control-J**>
[*count*] <**control-N**>
[*count*] **j**

Move the cursor down *count* lines without changing the current column.

<**control-L**>
<**control-R**>

Repaint the screen.

[*count*] <**control-M**>
[*count*] **+**

Move the cursor down *count* lines to the first non-blank character of that line.

[*count*] <**control-P**>
[*count*] **k**

Move the cursor up *count* lines, without changing the current column.

<**control-T**>

Return to the most recent tag context.

[*count*] <**control-U**>

Scroll backwards *count* lines. If *count* is not given, scroll backwards the number of lines specified by the last <**control-D**> or <**control-U**> command. If this is the first <**control-U**> command, scroll half the number of lines in the current screen.

<**control-W**>

Switch to the next lower screen in the window, or to the first screen if there are no lower screens in the window.

[*count*] <**control-Y**>

Scroll backwards *count* lines, leaving the current line and column as is, if possible.

<**control-Z**>
Suspend the current editor session.

<**escape**>
Execute the **ex** command being entered, or cancel it if it is only partial.

<**control-]**>
Push a tag reference onto the tag stack.

<**control-^**>
Switch to the most recently edited file.

[*count*] <**space**>
[*count*] **l**
Move the cursor forward *count* characters without changing the current line.

[*count*] **!** *motion shell-argument(s)* <carriage-return>
Replace the lines spanned by *count* and *motion* with the output (standard output and standard error) of the program named by the **shell** option, called with a **-c** flag followed by the *shell-argument(s)* (bundled into a single argument). Within *shell-argument(s)*, the ‘%’, ‘#’ and ‘!’ characters are expanded to the current file name, the previous current file name, and the command text of the previous **!** or **:!** commands, respectively. The special meaning of ‘%’, ‘#’ and ‘!’ can be overridden by escaping them with a backslash.

[*count*] **# #|+|-**
Increment (trailing ‘#’ or ‘+’) or decrement (trailing ‘-’) the number under the cursor by *count*, starting at the cursor position or at the first non-blank character following it. Numbers with a leading ‘0x’ or ‘0X’ are interpreted as hexadecimal numbers. Numbers with a leading ‘0’ are interpreted as octal numbers unless they contain a non-octal digit. Other numbers may be prefixed with a ‘+’ or ‘-’ sign.

[*count*] **$**
Move the cursor to the end of a line. If *count* is specified, additionally move the cursor down *count* - 1 lines.

% Move to the **matchchars** character matching the one found at the cursor position or the closest to the right of it.

& Repeat the previous substitution command on the current line.

'<*character*>
`<*character*>

Return to the cursor position marked by the character *character*, or, if *character* is '' or ``, to the position of the cursor before the last of the following commands: <**control-A**>, <**control-T**>, <**control-]**>, **%**, **'**, **`**, **(**, **)**, /, **?**, **G**, **H**, **L**, **[[**, **]]**, **{**, **}**. The first form returns to the first non-blank character of the line marked by *character*. The second form returns to the line and column marked by *character*.

[*count*] **(**
[*count*] **)**

Move *count* sentences backward or forward, respectively. A sentence is an area of text that begins with the first nonblank character following the previous sentence, paragraph, or section boundary and continues until the next period, exclamation point, or question mark character, followed by any number of closing parentheses, brackets, double or single quote characters, followed by either an end-of-line or two whitespace characters. Groups of empty lines (or lines containing only whitespace characters) are treated as a single sentence.

[*count*] **,**

Reverse find character (i.e., the last **F**, **f**, **T** or **t** command) *count* times.

[*count*] **-**

Move to the first non-blank character of the previous line, *count* times.

[*count*] **.**

Repeat the last **vi** command that modified text. *count* replaces both the *count* argument of the repeated command and that of the associated *motion*. If the **.** command repeats the **u** command, the change log is rolled forward or backward, depending on the action of the **u** command.

/*RE* <carriage-return>
/*RE*/ [*offset*] [**z**] <carriage-return>
?*RE* <carriage-return>
?*RE*? [*offset*] [**z**] <carriage-return>
N
n Search forward ('/') or backward ('?') for a regular expression. **n** and **N** repeat the last search in the same or opposite directions, respectively. If *RE* is empty, the last search regular expression is used. If *offset* is specified, the cursor is placed *offset* lines before or after the matched regular expression. If either **n** or **N** commands are used as motion components for the **!** command, there will be no prompt for the text of the command and the previous **!** will be

executed. Multiple search patterns may be grouped together by delimiting them with semicolons and zero or more whitespace characters. These patterns are evaluated from left to right with the final cursor position determined by the last search pattern. A **z** command may be appended to the closed search expressions to reposition the result line.

0 Move to the first character in the current line.

: Execute an **ex** command.

[*count*] **;**
Repeat the last character find (i.e., the last **F**, **f**, **T** or **t** command) *count* times.

[*count*] **<** *motion*
[*count*] **>** *motion*
Shift *count* lines left or right, respectively, by an amount of **shiftwidth**.

@ *buffer*
Execute a named *buffer* as **vi** commands. The buffer may include **ex** commands too, but they must be expressed as a **:** command. If *buffer* is ‘@’ or ‘*’, then the last buffer executed shall be used.

[*count*] **A**
Enter input mode, appending the text after the end of the line. If a *count* argument is given, the characters input are repeated *count* - 1 times after input mode is exited.

[*count*] **B**
Move backwards *count* bigwords.

[*buffer*] **C**
Change text from the current position to the end-of-line. If *buffer* is specified, "yank" the deleted text into *buffer*.

[*buffer*] **D**
Delete text from the current position to the end-of-line. If *buffer* is specified, "yank" the deleted text into *buffer*.

[*count*] **E**
Move forward *count* end-of-bigwords.

[*count*] **F** <*character*>

Search *count* times backward through the current line for <*character*>.

[*count*] **G**
Move to line *count*, or the last line of the file if *count* is not specified.

[*count*] **H**
Move to the screen line *count* - 1 lines below the top of the screen.

[*count*] **I**
Enter input mode, inserting the text at the beginning of the line. If a *count* argument is given, the characters input are repeated *count* - 1 more times.

[*count*] **J**
Join *count* lines with the current line. The spacing between two joined lines is set to two whitespace characters if the former ends with a question mark, a period or an exclamation point. It is set to one whitespace character otherwise.

[*count*] **L**
Move to the screen line *count* - 1 lines above the bottom of the screen.

M Move to the screen line in the middle of the screen.

[*count*] **O**
Enter input mode, appending text in a new line above the current line. If a *count* argument is given, the characters input are repeated *count* - 1 more times.

[*buffer*] **P**
Insert text from *buffer* before the current column if *buffer* is character-oriented or before the current line if it is line-oriented.

Q Exit **vi** (or visual) mode and switch to **ex** mode.

[*count*] **R**
Enter input mode, replacing the characters in the current line. If a *count* argument is given, the characters input are repeated *count* - 1 more times upon exit from insert mode.

[*buffer*] [*count*] **S**
Substitute *count* lines. If *buffer* is specified, "yank" the deleted text into *buffer*.

[*count*] **T** <*character*>

Search backwards, *count* times, through the current line for the character after the specified <*character*>.

U Restore the current line to its state before the cursor last moved to it.

[*count*] **W**
Move forward *count* bigwords.

[*buffer*] [*count*] **X**
Delete *count* characters before the cursor, on the current line. If *buffer* is specified, "yank" the deleted text into *buffer*.

[*buffer*] [*count*] **Y**
Copy (or "yank") *count* lines into *buffer*.

ZZ Write the file and exit **vi** if there are no more files to edit. Entering two "quit" commands in a row ignores any remaining file to edit.

[*count*] **[[**
Back up *count* section boundaries.

[*count*] **]]**
Move forward *count* section boundaries.

^ Move to the first non-blank character on the current line.

[*count*] _
Move down *count* - 1 lines, to the first non-blank character.

[*count*] **a**
Enter input mode, appending the text after the cursor. If a *count* argument is given, the characters input are repeated *count* number of times.

[*count*] **b**
Move backwards *count* words.

[*buffer*] [*count*] **c** *motion*
Change the region of text described by *count* and *motion*. If *buffer* is specified, "yank" the changed text into *buffer*.

[*buffer*] [*count*] **d** *motion*

Delete the region of text described by *count* and *motion*. If *buffer* is specified, "yank" the deleted text into *buffer*.

[*count*] **e**

Move forward *count* end-of-words.

[*count*] **f** <*character*>

Search forward, *count* times, through the rest of the current line for <*character*>.

[*count*] **i**

Enter input mode, inserting the text before the cursor. If a *count* argument is given, the characters input are repeated *count* number of times.

m <*character*>

Save the current context (line and column) as <*character*>.

[*count*] **o**

Enter input mode, appending text in a new line under the current line. If a *count* argument is given, the characters input are repeated *count* - 1 more times.

[*buffer*] **p**

Append text from *buffer*. Text is appended after the current column if *buffer* is character oriented, or after the current line otherwise.

[*count*] **r** <*character*>

Replace *count* characters with *character*.

[*buffer*] [*count*] **s**

Substitute *count* characters in the current line starting with the current character. If *buffer* is specified, "yank" the substituted text into *buffer*.

[*count*] **t** <*character*>

Search forward, *count* times, through the current line for the character immediately before <*character*>.

u Undo the last change made to the file. If repeated, the **u** command alternates between these two states. The **.** command, when used immediately after **u**, causes the change log to be rolled forward or backward, depending on the action of the **u** command.

[*count*] **w**

Move forward *count* words.

[*buffer*] [*count*] **x**

Delete *count* characters at the current cursor position, but no more than there are till the end of the line.

[*buffer*] [*count*] **y** *motion*

Copy (or "yank") a text region specified by *count* and *motion* into a buffer.

[*count1*] **z** [*count2*] **type**

Redraw, optionally repositioning and resizing the screen. If *count2* is specified, limit the screen size to *count2* lines. The following **type** characters may be used:

+ If *count1* is specified, place the line *count1* at the top of the screen. Otherwise, display the screen after the current screen.

<**carriage-return**>

Place the line *count1* at the top of the screen.

. Place the line *count1* in the center of the screen.

- Place the line *count1* at the bottom of the screen.

^ If *count1* is given, display the screen before the screen before *count1* (i.e., 2 screens before). Otherwise, display the screen before the current screen.

[*count*] **{**

Move backward *count* paragraphs.

[*column*] **|**

Move to a specific *column* position on the current line. If *column* is omitted, move to the start of the current line.

[*count*] **}**

Move forward *count* paragraphs.

[*count*] **~** *motion*

If the **tildeop** option is not set, reverse the case of the next *count* character(s) and no *motion* can be specified. Otherwise *motion* is mandatory and **~** reverses the case of the characters in a text

region specified by the *count* and *motion*.

<**interrupt**>
Interrupt the current operation. The <interrupt> character is usually <control-C>.

VI TEXT INPUT COMMANDS

The following section describes the commands available in the text input mode of the **vi** editor.

<**nul**> Replay the previous input.

<**control-D**>
Erase to the previous *shiftwidth* column boundary.

^<**control-D**>
Erase all of the autoindent characters, and reset the autoindent level.

0<**control-D**>
Erase all of the autoindent characters.

<**control-T**>
Insert sufficient <tab> and <space> characters to move forward to the next *shiftwidth* column boundary. If the **expandtab** option is set, only insert <space> characters.

<**erase**>
<**control-H**>
Erase the last character.

<**literal next**>
Escape the next character from any special meaning. The <literal next> character is usually <control-V>.

<**escape**>
Resolve all text input into the file, and return to command mode.

<**line erase**>
Erase the current line.

<**control-W**>
<**word erase**>
Erase the last word. The definition of word is dependent on the **altwerase** and **ttywerase**

options.

<**control-X**>[**0-9A-Fa-f**]**+**
Insert a character with the specified hexadecimal value into the text.

<**interrupt**>
Interrupt text input mode, returning to command mode. The <interrupt> character is usually <control-C>.

EX COMMANDS

The following section describes the commands available in the **ex** editor. In each entry below, the tag line is a usage synopsis for the command.

<**end-of-file**>
Scroll the screen.

! *argument(s)*
[*range*] **!** *argument(s)*
Execute a shell command, or filter lines through a shell command.

" A comment.

[*range*] **nu**[**mber**] [*count*] [*flags*]
[*range*] **#** [*count*] [*flags*]
Display the selected lines, each preceded with its line number.

@ *buffer*
***** *buffer*
Execute a buffer.

[*range*] **<**[**<** **...**] [*count*] [*flags*]
Shift lines left.

[*line*] **=** [*flags*]
Display the line number of *line*. If *line* is not specified, display the line number of the last line in the file.

[*range*] **>**[**>** **...**] [*count*] [*flags*]
Shift lines right.

ab[**breviate**] *lhs rhs*
vi only. Add *lhs* as an abbreviation for *rhs* to the abbreviation list.

[*line*] **a**[**ppend**][**!**]
The input text is appended after the specified line.

ar[**gs**] Display the argument list.

bg **vi** only. Background the current screen.

[*range*] **c**[**hange**][**!**] [*count*]
The input text replaces the specified range.

chd[**ir**][**!**] [*directory*]
cd[**!**] [*directory*]
Change the current working directory.

[*range*] **co**[**py**] *line* [*flags*]
[*range*] **t** *line* [*flags*]
Copy the specified lines after the destination *line*.

cs[**cope**] **add** | **find** | **help** | **kill** | **reset**
Execute a Cscope command.

[*range*] **d**[**elete**] [*buffer*] [*count*] [*flags*]
Delete the lines from the file.

di[**splay**] **b**[**uffers**] | **c**[**onnections**] | **s**[**creens**] | **t**[**ags**]
Display buffers, Cscope connections, screens or tags.

[**Ee**][**dit**][**!**] [+*cmd*] [*file*]
[**Ee**]**x**[**!**] [+*cmd*] [*file*]
Edit a different file. The capitalized command opens a new screen below the current screen.

exu[**sage**] [*command*]
Display usage for an **ex** command.

f[**ile**] [*file*]
Display and optionally change the file name.

[**Ff**]**g** [*name*]

vi mode only. Foreground the specified screen. The capitalized command opens a new screen below the current screen.

[*range*] **g**[**lobal**] /*pattern*/ [*commands*]
[*range*] **v** /*pattern*/ [*commands*]

Apply commands to lines matching (‘global’) or not matching (‘v’) a pattern.

he[**lp**] Display a help message.

[*line*] **i**[**nsert**][**!**]

The input text is inserted before the specified line.

[*range*] **j**[**oin**][**!**] [*count*] [*flags*]

Join lines of text together.

[*range*] **l**[**ist**] [*count*] [*flags*]

Display the lines unambiguously.

map[**!**] [*lhs rhs*]

Define or display maps (for **vi** only).

[*line*] **ma**[**rk**] <*character*>
[*line*] **k** <*character*>

Mark the line with the mark <*character*>.

[*range*] **m**[**ove**] *line*

Move the specified lines after the target line.

mk[**exrc**][**!**] *file*

Write the abbreviations, editor options and maps to the specified *file*.

[**Nn**][**ext**][**!**] [*file ...*]

Edit the next file from the argument list. The capitalized command opens a new screen below the current screen.

pre[**serve**]

Save the file in a form that can later be recovered using the **ex -r** option.

[**Pp**]**rev**[**ious**][**!**]

Edit the previous file from the argument list. The capitalized command opens a new screen below the current screen.

[*range*] **p**[**rint**] [*count*] [*flags*]
Display the specified lines.

[*line*] **pu**[**t**] [*buffer*]
Append buffer contents to the current line.

q[**uit**][**!**]
End the editing session.

[*line*] **r**[**ead**][**!**] [*file*]
Read a file.

rec[**over**] *file*
Recover *file* if it was previously saved.

res[**ize**] [+|-]*size*
vi mode only. Grow or shrink the current screen.

rew[**ind**][**!**]
Rewind the argument list.

se[**t**] [option[=[value]] ...] [nooption ...] [option? ...] [*all*]
Display or set editor options.

sh[**ell**] Run a shell program.

so[**urce**] *file*
Read and execute **ex** commands from a file.

[*range*] **s**[**ubstitute**] [/*pattern*/*replace*/] [*options*] [*count*] [*flags*]
[*range*] **&** [*options*] [*count*] [*flags*]
[*range*] **~** [*options*] [*count*] [*flags*]
Make substitutions. The *replace* field may contain any of the following sequences:

‘&’ The text matched by *pattern*.

‘~’ The replacement part of the previous **substitute** command.

‘%’ If this is the entire *replace* pattern, the replacement part of the previous **substitute** command.

‘\#’ Where ‘#’ is an integer from 1 to 9, the text matched by the #’th subexpression in *pattern*.

‘\L’ Causes the characters up to the end of the line of the next occurrence of ‘\E’ or ‘\e’ to be converted to lowercase.

‘\l’ Causes the next character to be converted to lowercase.

‘\U’ Causes the characters up to the end of the line of the next occurrence of ‘\E’ or ‘\e’ to be converted to uppercase.

‘\u’ Causes the next character to be converted to uppercase.

su[**spend**][**!**]
st[**op**][**!**]
<**suspend**>
Suspend the edit session. The <suspend> character is usually <control-Z>.

[**Tt**]**a**[**g**][**!**] *tagstring*
Edit the file containing the specified tag. The capitalized command opens a new screen below the current screen.

tagn[**ext**][**!**]
Edit the file containing the next context for the current tag.

tagp[**op**][**!**] [*file* | *number*]
Pop to the specified tag in the tags stack.

tagpr[**ev**][**!**]
Edit the file containing the previous context for the current tag.

tagt[**op**][**!**]
Pop to the least recent tag on the tags stack, clearing the stack.

una[**bbreviate**] *lhs*
vi only. Delete an abbreviation.

u[**ndo**] Undo the last change made to the file.

unm[**ap**][**!**] *lhs*
Unmap a mapped string.

ve[**rsion**]
Display the version of the **ex**/**vi** editor.

[*line*] **vi**[**sual**] [*type*] [*count*] [*flags*]
ex mode only. Enter **vi**.

Vi[**sual**][**!**] [*+cmd*] [*file*]
vi mode only. Edit a different file by opening a new screen below the current screen.

viu[**sage**] [*command*]
Display usage for a **vi** command.

vs[**plit**] [*+cmd*] [*file*]
Edit a different file by opening a new screen to the right of the current screen.

[*range*] **w**[**rite**][**!**] [>>] [*file*]
[*range*] **w**[**rite**] **!***shell-command*
[*range*] **wn**[**!**] [>>] [*file*]
[*range*] **wq**[**!**] [>>] [*file*]
Write the entire file, or *range*. ‘!’ overwrites a different, preexisting file. ‘>>’ appends to a file that may preexist. Whitespace followed by ‘!’ pipes the file to *shell-command*. **wn** moves to the next file if writing succeeds. **wq** exits the editor if writing succeeds, unless there are more files to edit; ‘!’ exits regardless.

[*range*] **x**[**it**][**!**] [*file*]
Exit the editor, writing the file if it has been modified.

[*range*] **ya**[**nk**] [*buffer*] [*count*]
Copy the specified lines to a buffer.

[*line*] **z** [*type*] [*count*] [*flags*]
Adjust the window.

SET OPTIONS

There are a large number of options that may be set (or unset) to change the editor’s behavior. This

section describes the options, their abbreviations and their default values.

In each entry below, the first part of the tag line is the full name of the option, followed by any equivalent abbreviations. The part in square brackets is the default value of the option. Most of the options are boolean, i.e., they are either on or off, and do not have an associated value.

Options apply to both **ex** and **vi** modes, unless otherwise specified.

altnotation [off]
Display control characters less than 0x20 in <C-char> notations. Carriage feed, escape, and delete are marked as <Enter>, <Esc>, and <Del>, respectively.

altwerase [off]
vi only. Select an alternate word erase algorithm.

autoindent, **ai** [off]
Automatically indent new lines.

autoprint, **ap** [on]
ex only. Display the current line automatically.

autowrite, **aw** [off]
Write modified files automatically when changing files or suspending the editor session.

backup [""]
Back up files before they are overwritten.

beautify, **bf** [off]
Discard control characters.

cdpath [environment variable CDPATH, or current directory]
The directory paths used as path prefixes for the **cd** command.

cedit [no default]
Set the character to edit the colon command-line history.

columns, **co** [80]
Set the number of columns in the screen.

comment [off]

vi only. Skip leading comments in shell, C and C++ language files.

directory, **dir** [environment variable TMPDIR, or */tmp*]
The directory where temporary files are created.

edcompatible, **ed** [off]
Remember the values of the ‘c’ and ‘g’ suffixes to the **substitute** commands, instead of initializing them as unset for each new command.

errorbells, **eb** [off]
ex only. Announce error messages with a bell.

escapetime [1]
The tenths of a second **ex**/**vi** waits for a subsequent key to complete an <escape> key mapping.

expandtab, **et** [off]
Expand <tab> characters to <space> when inserting, replacing or shifting text, autoindenting, indenting with <**control-T**>, outdenting with <**control-D**>, or when filtering lines with the **!** command.

exrc, **ex** [off]
Read the startup files in the local directory.

extended [off]
Use extended regular expressions (EREs) rather than basic regular expressions (BREs). See re_format(7) for more information on regular expressions.

filec [<tab>]
Set the character to perform file path completion on the colon command line.

fileencoding, **fe** [auto detect]
Set the encoding of the current file.

flash [on]
Flash the screen instead of beeping the keyboard on error.

hardtabs, ht [0]
Set the spacing between hardware tab settings. This option currently has no effect.

iclower [off]

Makes all regular expressions case-insensitive, as long as an upper-case letter does not appear in the search string.

ignorecase, **ic** [off]
Ignore case differences in regular expressions.

inputencoding, **ie** [locale]
Set the encoding of your input characters.

keytime [6]
The tenths of a second **ex**/**vi** waits for a subsequent key to complete a key mapping.

leftright [off]
vi only. Do left-right scrolling.

lines, **li** [24]
vi only. Set the number of lines in the screen.

lisp [off]
vi only. Modify various search commands and options to work with Lisp. This option is not yet implemented.

list [off]
Display lines in an unambiguous fashion.

lock [on]
Attempt to get an exclusive lock on any file being edited, read or written.

magic [on]
When turned off, all regular expression characters except for ‘^’ and ‘$’ are treated as ordinary characters. Preceding individual characters by ‘\’ re-enables them.

matchchars [[]{}()]
Character pairs looked for by the **%** command.

matchtime [7]
vi only. The tenths of a second **ex**/**vi** pauses on the matching character when the **showmatch** option is set.

mesg [on]

Permit messages from other users.

msgcat [/usr/share/vi/catalog/]
Selects a message catalog to be used to display error and informational messages in a specified language.

modelines, **modeline** [off]
Read the first and last few lines of each file for **ex** commands. This option will never be implemented.

noprint [""]
Characters that are never handled as printable characters.

number, **nu** [off]
Precede each line displayed with its current line number.

octal [off]
Display unknown characters as octal numbers, instead of the default hexadecimal.

open [on]
ex only. If this option is not set, the **open** and **visual** commands are disallowed.

optimize, **opt** [on]
vi only. Optimize text throughput to dumb terminals. This option is not yet implemented.

paragraphs, **para** [IPLPPPQPP LIpplpipbp]
vi only. Define additional paragraph boundaries for the **{** and **}** commands.

path [""]
Define additional directories to search for files being edited.

print [""]
Characters that are always handled as printable characters.

prompt [on]
ex only. Display a command prompt.

readonly, **ro** [off]
Mark the file and session as read-only.

recdir [/var/tmp/vi.recover]
The directory where recovery files are stored.

redraw, **re** [off]
vi only. Simulate an intelligent terminal on a dumb one. This option is not yet implemented.

remap [on]
Remap keys until resolved.

report [5]
Set the number of lines about which the editor reports changes or yanks.

ruler [off]
vi only. Display a row/column ruler on the colon command line.

scroll, **scr** [window size / 2]
Set the number of lines scrolled.

searchincr [off]
Makes the **/** and **?** commands incremental.

sections, **sect** [NHSHH HUnhsh]
vi only. Define additional section boundaries for the **[[** and **]]** commands.

secure [off]
Turns off all access to external programs.

shell, **sh** [environment variable SHELL, or */bin/sh*]
Select the shell used by the editor.

shellmeta [~{[*?$‘’"\]
Set the meta characters checked to determine if file name expansion is necessary.

shiftwidth, **sw** [8]
Set the autoindent and shift command indentation width.

showmatch, **sm** [off]
vi only. Note the left matching characters when the right ones are inserted.

showmode, **smd** [off]

vi only. Display the current editor mode and a "modified" flag.

sidescroll [16]
vi only. Set the amount a left-right scroll will shift.

slowopen, **slow** [off]
Delay display updating during text input. This option is not yet implemented.

sourceany [off]
Read startup files not owned by the current user. This option will never be implemented.

tabstop, **ts** [8]
This option sets tab widths for the editor display.

taglength, **tl** [0]
Set the number of significant characters in tag names.

tags, **tag** [tags]
Set the list of tags files.

term, **ttytype**, **tty** [environment variable TERM]
Set the terminal type.

terse [off]
This option has historically made editor messages less verbose. It has no effect in this implementation.

tildeop [off]
Modify the **~** command to take an associated motion.

timeout, **to** [on]
Time out on keys which may be mapped.

ttywerase [off]
vi only. Select an alternate erase algorithm.

verbose [off]
vi only. Display an error message for every error.

w300 [no default]

vi only. Set the window size if the baud rate is less than 1200 baud.

w1200 [no default]
vi only. Set the window size if the baud rate is equal to 1200 baud.

w9600 [no default]
vi only. Set the window size if the baud rate is greater than 1200 baud.

warn [on]
ex only. This option causes a warning message to be printed on the terminal if the file has been modified since it was last written, before a **!** command.

window, **w**, **wi** [environment variable LINES - 1]
Set the window size for the screen.

windowname [off]
Change the icon/window name to the current file name.

wraplen, **wl** [0]
vi only. Break lines automatically, the specified number of columns from the left-hand margin. If both the **wraplen** and **wrapmargin** edit options are set, the **wrapmargin** value is used.

wrapmargin, **wm** [0]
vi only. Break lines automatically, the specified number of columns from the right-hand margin. If both the **wraplen** and **wrapmargin** edit options are set, the **wrapmargin** value is used.

wrapscan, **ws** [on]
Set searches to wrap around the end or beginning of the file.

writeany, **wa** [off]
Turn off file-overwriting checks.

ENVIRONMENT

COLUMNS The number of columns on the screen. This value overrides any system or terminal specific values. If the COLUMNS environment variable is not set when **ex**/**vi** runs, or the **columns** option is explicitly reset by the user, **ex**/**vi** enters the value into the environment.

EXINIT A list of **ex** startup commands, read after */etc/vi.exrc* unless the variable NEXINIT is also set.

HOME The user's home directory, used as the initial directory path for the startup *$HOME/.nexrc* and *$HOME/.exrc* files. This value is also used as the default directory for the **cd** command.

LINES The number of rows on the screen. This value overrides any system or terminal specific values. If the LINES environment variable is not set when **ex**/**vi** runs, or the **lines** option is explicitly reset by the user, **ex**/**vi** enters the value into the environment.

NEXINIT A list of **ex** startup commands, read after */etc/vi.exrc*.

SHELL The user's shell of choice (see also the **shell** option).

TERM The user's terminal type. The default is the type "unknown". If the TERM environment variable is not set when **ex**/**vi** runs, or the **term** option is explicitly reset by the user, **ex**/**vi** enters the value into the environment.

TMPDIR The location used to store temporary files (see also the **directory** edit option).

ASYNCHRONOUS EVENTS

SIGALRM **vi**/**ex** uses this signal for periodic backups of file modifications and to display "busy" messages when operations are likely to take a long time.

SIGHUP
SIGTERM If the current buffer has changed since it was last written in its entirety, the editor attempts to save the modified file so it can be later recovered. See the **vi**/**ex** reference manual section *Recovery* for more information.

SIGINT When an interrupt occurs, the current operation is halted and the editor returns to the command level. If interrupted during text input, the text already input is resolved into the file as if the text input had been normally terminated.

SIGWINCH The screen is resized. See the **vi**/**ex** reference manual section *Sizing the Screen* for more information.

FILES

/bin/sh The default user shell.

/etc/vi.exrc System-wide **vi** startup file. It is read for **ex** commands first in the startup sequence. Must be owned by root or the user, and writable only by the owner.

/tmp Temporary file directory.

/var/tmp/vi.recover The default recovery file directory.

$HOME/.nexrc First choice for user's home directory startup file, read for **ex** commands right after */etc/vi.exrc* unless either NEXINIT or EXINIT are set. Must be owned by root or the user, and writable only by the owner.

$HOME/.exrc Second choice for user's home directory startup file, read for **ex** commands under the same conditions as *$HOME/.nexrc*.

.nexrc First choice for local directory startup file, read for **ex** commands at the end of the startup sequence if the **exrc** option was turned on earlier. Must be owned by the user and writable only by the owner.

.exrc Second choice for local directory startup file, read for **ex** commands under the same conditions as *.nexrc*.

EXIT STATUS

The **ex** and **vi** utilities exit 0 on success, and >0 if an error occurs.

SEE ALSO

ctags(1), iconv(1), re_format(7)

vi/ex reference manual, https://docs.freebsd.org/44doc/usd/13.viref/paper.pdf.

STANDARDS

nex/**nvi** is close to IEEE Std 1003.1-2008 ("POSIX.1"). That document differs from historical **ex**/**vi** practice in several places; there are changes to be made on both sides.

HISTORY

The **ex** editor first appeared in 1BSD. The **nex**/**nvi** replacements for the **ex**/**vi** editor first appeared in 4.4BSD.

AUTHORS

Bill Joy wrote the original version of **ex** in 1977.