NAME

ex, vi, view – text editors

SYNOPSIS

ex [**−eFRrsv**] [**−c** *cmd*] [**−t** *tag*] [**−w** *size*] [*file ...*]
vi [**−eFRrv**] [**−c** *cmd*] [**−t** *tag*] [**−w** *size*] [*file ...*]
view [**−eFRrv**] [**−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 Execute *cmd* immediately after starting the edit session. Particularly useful for initial positioning in the file, however *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 Enter batch mode; applicable only to **ex** edit sessions. Batch mode is useful when running **ex** scripts. Prompts, informative messages and other user oriented message are turned off, and no startup files or environmental variables are read. This is the POSIX 1003.2 interface for the historic "−" argument. **Nex/nvi** supports both the old and new syntax.

−t Start editing at the specified tag. (See ctags(1)).

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

−v Start editing in vi mode, as if the command name was **vi** or **view**.

−X Reserved for X11 interfaces. *No X11 support is currently implemented.*

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.

`Ex/vi` exits 0 on success, and greater than 0 if an error occurs.

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. (Key names are written using less-than and greater-than signs, 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 command "`vi file_name<carriage-return>`". The command you should enter as soon as you start editing is "`:set verbose showmode<carriage-return>`". 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<carriage-return>
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 the 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<carriage-return>
Write the file back to the file with the name that you originally used as an argument on the **vi** com-

mand line.

:w file_name<carriage-return>
Write the file back to the file with the name ‘‘file_name’’.

The commands to quit editing and exit the editor are:

:q<carriage-return>
Quit editing and leave vi (if you’ve modified the file, but not saved your changes, **`vi`** will refuse to quit).

:q!<carriage-return>
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.

VI COMMANDS

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

[count] <control-A>
Search forward `count` times for the current word.

[count] <control-B>
Page backwards `count` screens.

[count] <control-D>
Scroll forward `count` lines.

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

[count] <control-F>
Page forward `count` screens.

<control-G>
Display the file information.

<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 nonblank 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.

<control-U>

Scroll backwards `count` lines.

<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.

<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 **`ex`** commands or cancel partial commands.

<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)
Replace text with results from a shell command.

[count] # +|-|#
Increment or decrement the cursor number.

[count] $
Move the cursor to the end of a line.

% Move to the matching character.

& Repeat the previous substitution command on the current line.

'<character>
`<character>
Return to a context marked by the character `<character>`.

[count] (
Back up `count` sentences.

[count])
Move forward `count` sentences.

[count] ,
Reverse find character `count` times.

[count] -
Move to first nonblank of the previous line, `count` times.

[count] .
Repeat the last **`vi`** command that modified text.

/RE<carriage-return>
/RE/ [offset]<carriage-return>
?RE<carriage-return>
?RE? [offset]<carriage-return>
N
n Search forward or backward for a regular expression.

0 Move to the first character in the current line.

: Execute an ex command.

[count] ;
Repeat the last character find `count` times.

[count] < motion

[count] > motion
Shift lines left or right.

@ buffer
Execute a named buffer.

[count] A
Enter input mode, appending the text after the end of the line.

[count] B
Move backwards `count` bigwords.

[buffer] [count] C
Change text from the current position to the end-of-line.

[buffer] D
Delete text from the current position to the end-of-line.

[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` 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.

[count] J
Join lines.

[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.

[buffer] P
Insert text from a buffer.

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

[count] R
Enter input mode, replacing the characters in the current line.

[buffer] [count] S
Substitute `count` lines.

[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.

[buffer] [count] Y
Copy (or "yank") `count` lines into the specified buffer.

ZZ Write the file and exit **vi**.

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

[count]]]

Move forward `count` section boundaries.

^ Move to first nonblank character on the current line.

[count] _
Move down `count - 1` lines, to the first nonblank character.

[count] a
Enter input mode, appending the text after the cursor.

[count] b
Move backwards `count` words.

[buffer] [count] c motion
Change a region of text.

[buffer] [count] d motion
Delete a region of text.

[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.

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.

[buffer] p
Append text from a buffer.

[count] r <character>
Replace `count` characters.

[buffer] [count] s
Substitute `count` characters in the current line starting with the current character.

[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.

[count] w
Move forward `count` words.

[buffer] [count] x
Delete `count` characters.

[buffer] [count] y motion
Copy (or "yank") a text region specified by the `count` and motion into a buffer.

[count1] z [count2] -|.|+|^|<carriage-return>
Redraw, optionally repositioning and resizing the screen.

[count] {
Move backward `count` paragraphs.

[count] |
Move to a specific *column* position on the current line.

[count] }
Move forward `count` paragraphs.

[count] ~
Reverse the case of the next `count` character(s).

[count] ~ motion

Reverse the case of the characters in a text region specified by the `count` and `motion`.

<interrupt>
Interrupt the current operation.

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 the previous autoindent character.

^<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 the cursor forward to a column immediately after the next column which is an even multiple of the **shiftwidth** option.

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

<literal next>
Quote the next character.

<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.

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

di[splay] b[uffers] | s[creens] | t[ags]
Display buffers, screens or tags.

e[dit][!] [+cmd] [file]
ex[!] [+cmd] [file]
Edit a different file.

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

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

fg [name]
Vi mode only. Foreground the specified screen.

[range] g[lobal] /pattern/ [commands]
[range] v /pattern/ [commands]
Apply commands to lines matching (or not matching) 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.

n[ext][!] [file ...]
Edit the next file from the argument list.

[line] o[pen] /pattern/ [flags]
Enter open mode.

pre[serve]
Save the file in a form that can later be recovered using the **ex −r** option.

prev[ious][!]
Edit the previous file from the argument list.

[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.

sp[lit] [file ...]

`Vi` mode only. Split the screen.

[range] s[ubstitute] [/pattern/replace/] [options] [count] [flags]
[range] & [options] [count] [flags]
[range] ~ [options] [count] [flags]

Make substitutions.

su[spend][!]
st[op][!]
<suspend>

Suspend the edit session.

ta[g][!] tagstring

Edit the file containing the specified tag.

tagp[op][!] [file | number]

Pop to the specified tag in the tags stack.

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 new file.

viu[sage] [command]

Display usage for a **`vi`** command.

[range] w[rite][!] [>>] [file]
[range] w[rite] [!] [file]
[range] wn[!] [>>] [file]
[range] wq[!] [>>] [file]

Write the file.

[range] x[it][!] [file]

Write 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.

altwerase [off]
: **Vi** only. Select an alternate word erase algorithm.

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

autoprint, ap [off]
: **Ex** only. Display the current line automatically.

autowrite, aw [off]
: Write modified files automatically when changing files.

beautify, bf [off]
: Discard control characters.

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

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

comment [off]
: **Vi** only. Skip leading comments in 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" suffices 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.

exrc, ex [off]
: Never read startup files in the local directory.

extended [off]
: Regular expressions are extended (i.e. `egrep`(1) style) expressions.

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

hardtabs, ht [8]
: Set the spacing between hardware tab settings.

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

keytime [6]
: The 10th's 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.

magic [on]
Treat certain characters specially in regular expressions.

matchtime [7]
Vi only. The 10th's of a second **ex/vi** pauses on the matching character when the **showmatch** option is set.

mesg [on]
Permit messages from other users.

modelines, modeline [off]
Read the first and last few lines of each file for **ex** commands.

This option will never be implemented.

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.

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

readonly, ro [off]
Mark the file 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 / 2]
Set the number of lines scrolled.

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

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

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

showdirty [off]

`Vi` only. Display an asterisk on the colon command line if the file has been modified.

showmatch, sm [off]
`Vi` only. Note matching "{" and "(" for "}" and ")" characters.

showmode [off]
`Vi` only. Display the current editor mode (command or input).

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 /var/db/libc.tags /sys/kern/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
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]
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 the terminal if the file has been modified, since it was last written, before a **!** command.

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

wrapmargin, wm [0]
`Vi` only. Break lines automatically when they reach the right-hand margin.

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

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

ENVIRONMENTAL VARIABLES

`COLUMNS` The number of columns on the screen. This value overrides any system or terminal specific values. If the COLUMNS environmental 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 if the variable `NEXINIT` is not 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 **vi cd** command.

`LINES` The number of rows on the screen. This value overrides any system or terminal specific values. If the LINES environmental 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.

`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 environmental 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 stored temporary files (see also the **directory** 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 entitled "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 entitled "Sizing the Screen" for more information.

SIGCONT

SIGQUIT

SIGTSTP **Vi/ex** ignores these signals.

BUGS

See the file `nvi/docs/bugs.current` for a list of the known bugs in this version.

FILES

`/bin/sh`	The default user shell.
`/etc/vi.exrc`	System-wide vi startup file.
`/tmp`	Temporary file directory.
`/var/tmp/vi.recover`	The default recovery file directory.
`$HOME/.nexrc`	1st choice for user's home directory startup file.
`$HOME/.exrc`	2nd choice for user's home directory startup file.
`.nexrc`	1st choice for local directory startup file.
`.exrc`	2nd choice for local directory startup file.

SEE ALSO

`ctags`(1), `more`(1), `curses`(3), `dbopen`(3)

The "Vi Quick Reference" card.

‘‘An Introduction to Display Editing with Vi’’, found in the ‘‘UNIX User’s Manual Supplementary Documents’’ section of both the 4.3BSD and 4.4BSD manual sets. This document is the closest thing available to an introduction to the **vi** screen editor.

‘‘Ex Reference Manual (Version 3.7)’’, found in the ‘‘UNIX User’s Manual Supplementary Documents’’ section of both the 4.3BSD and 4.4BSD manual sets. This document is the final reference for the **ex** editor, as distributed in most historic 4BSD and System V systems.

‘‘Edit: A tutorial’’, found in the ‘‘UNIX User’s Manual Supplementary Documents’’ section of the 4.3BSD manual set. This document is an introduction to a simple version of the **ex** screen editor.

‘‘Ex/Vi Reference Manual’’, found in the ‘‘UNIX User’s Manual Supplementary Documents’’ section of the 4.4BSD manual set. This document is the final reference for the **nex/nvi** text editors, as distributed in 4.4BSD and 4.4BSD-Lite.

Roff source for all of these documents is distributed with **nex/nvi** in the `nvi/USD.doc` directory of the **nex/nvi** source code.

The files ‘‘autowrite’’, ‘‘input’’, ‘‘quoting’’, and ‘‘structures’’, found in the `nvi/docs/internals` directory of the **nex/nvi** source code.

HISTORY

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

STANDARDS

Nex/nvi is close to IEEE Std1003.2 (‘‘POSIX’’). That document differs from historical **ex/vi** practice in several places; there are changes to be made on both sides.