## Name

**ex, vi, view** – text editors

## SYNOPSIS

**vi** [ **-eFlRv**] [ **-c** *command*] [ **-r** *file*] [ **-t** *tag*] [ **-w** *size*] [ **-x** *aw*]
**view** [ **-eFlRv**] [ **-c** *command*] [ **-r** *file*] [ **-t** *tag*] [ **-w** *size*] [ **-x** *aw*]
**ex** [ **-eFlRsv**] [ **-c** *command*] [ **-r** *file*] [ **-t** *tag*] [ **-w** *size*] [ **-x** *aw*]

## DESCRIPTION

**Ex** is a line-oriented text editor. **Vi** is a screen oriented text editor, based on **ex**. **View** is the same as using the **-R** (read-only) option of **vi**. **Ex** and **vi** are really different interfaces to the same program, and it is possible to switch back and forth during the same edit session.

This manual page is the one provided with the **nex** and **nvi** versions of the vi text editors. **Nex** and **nvi** are intended as bug-for-bug compatible replacements for the original Fourth Berkeley Software Distribution (4BSD) **ex** and **vi** programs.

The following options are available:

**-c** Execute *command* immediately after starting the edit session. Useful for initial positioning in the file, but *command* is not limited to positioning commands. This is the POSIX 1003.2 interface for the historic "+command" syntax. **Nex/nvi** supports both the old and new syntax.

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

**-F** Don't copy the entire file when first starting to edit. (The default is to make a copy in case someone else tries to edit the file too.)

**-l** List the files that may be recovered using the **-r** option of **vi**. This is the new interface for the historic **-r** without a file argument syntax. **Nex/nvi** supports both the old and new syntax.

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

**-r** Recover the specified file.

**-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 things are turned off. This is the POSIX 1003.2 interface for the historic **--** syntax. **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. Not currently implemented.

## ADDITIONAL FEATURES

This manual page is the one distributed with the **nvi** and **nex** reimplementations of **ex/vi**. There are a few additional features in **nex/nvi**.

8-bit clean data, large lines, files

> **Nvi/nex** will edit any format file. Line lengths are limited by available memory, and file sizes are limited by available disk space. The command "^Vx[0-9A-Fa-f]*", in input mode, will insert any legal character value into the text.

Split screens

> The command ":sp[lit] [file ...]" splits the screen in vi mode. The key "^W" switches between the foreground screens, and the ":resize count" command can be used to grow or shrink any particular screen.

Background and foreground screens

> The command ":bg" backgrounds the current screen, and the command ":fg [file]" foregrounds the backgrounded screeen that is editing the specified file, or, by default, the first background screen on the queue. The command ":di[splay] s[creens]" lists the background screens.

Shell screens

> The command ":sc[ript] [file ...]" runs a shell in the screen. Editing is unchanged, with the exception that a <carriage-return> enters the current line (stripped of any prompt) as input to the shell.

Tag stacks

> Tags are now maintained in a stack. The command "^T" returns to the previous tag location. The command ":tagpop [number  file]" returns to the most recent tag location by default, or, optional-

ly to a specific tag number in the tag stack, or the most recent tag from the specified file. Use the command ":di[splay] t[ags]" to view the tags stack. The command ":tagtop" returns to the top of the tag stack.

New displays
: The command ":di[splay] b[uffers] s[creens] t[ags]" can be used to display, respectively, the current cut buffers, the backgrounded screens, and the tags stack.

Infinite undo
: The changes made during an edit session may be rolled backward and forward. A '.' command immediately after a 'u' command continues either forward or backward depending on whether the 'u' command was an undo or a redo.

Usage information
: The command ":exu[sage [cmd]" and "viu[sage] [key]" provide usage information for all of the ex and vi commands by default, or, optionally, for a specific command or key.

Extended regular expressions
: The ":set extended" command treats search and other command regular expressions as extended (egrep(1) style) regular expressions.

Word search
: The command "^A" searches for the word referenced by the cursor.

Number increment
: The command "#" increments the number referenced by the cursor.

Previous file
: The command ":prev[ious][!]" edits the previous file from the argument list.

Left-Right scrolling
: The command ":set leftright" makes **nvi** do left-right screen scrolling, instead of the traditional **vi** line wrapping.

## RECOVERY

There is no recovery program for nvi, nor does it run setuid. Users may recover any file which they may read, and the superuser may recover any edit session.

Edit sessions are backed by files in `/var/tmp/vi.recover`, and are named "vi.XXXX", where "XXXX" is a number related to the process id. When a file is first modified, a second file, which contains an email message for the user, is created, and is named "recover.XXXX", where, again, "XXXX" is associated with the process id. Both files are removed at the end of a normal edit session, but will remain if the edit session is abnormally terminated or the user enters the ex/vi "preserve" command. The use of the `/var/tmp` directory may be changed setting the option "recdir" in the user's or system startup information.

The recovery directory should have the "sticky-bit" set so that only the owners of files may remove them. If this is not possible on the system, then a pseudo-user should own the recovery directory. The recovery directory must be both read and write-able by any user.

The recovery file has all of the necessary information in it to enable the user to recover the edit session. In addition, it has all of the necessary email headers for sendmail. When the system is rebooted, all of the files in `/var/tmp/vi.recover` named "recover.XXXX" should be sent by email, using the **−t** flag of sendmail (or a similar mechanism in other mailers). A simple way to do this is to insert the following script into your rc.local (or other startup) file:

```
# Recover nvi editor files.
virecovery=/var/tmp/vi.recover/recover.*
if [ "$virecovery" != "/var/tmp/vi.recover/recover.*" ]; then
        echo 'Recovering vi editor sessions'
        for i in $virecovery; do
                sendmail -t < $i
        done
fi
```

If **nex/nvi** receives a hangup (SIGHUP) signal, it will email the recovery information to the user itself.

If you don't have the sendmail program on your system, the source file `nvi/recover.c` will have to be modified to use your mail delivery programs.

## BUGS

This man page is woefully incomplete.

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

## FILES

| | |
|---|---|
| `/tmp` | Storage for temporary files. |
| `/var/tmp/vi.recover` | Storage for recovery files. |

## SEE ALSO

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

The "Vi Quick Reference" card.

"Introduction to Display Editing with Vi", found in the "UNIX User's Manual Supplementary Documents".

"Edit: A tutorial", found in the "UNIX User's Manual Supplementary Documents".

"Ex Reference Manual (Version 3.7)", found in the "UNIX User's Manual Supplementary Documents".

**Nroff/troff** source for the previous three documents are distributed with **nex/nvi** in the `nvi/docs/USD.doc` directory of the **nex/nvi** source code.

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

## HISTORY

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

## STANDARDS

**Nex/nvi** is fairly close to IEEE Std1003.2 ("POSIX"), but it's not there yet.