elvis - a clone of the ex/vi text editor

SYNOPSIS

elvis [**-V...**] [**-a**] [**-r**] [**-R**] [**-e**] [**-i**] [**-S**] [**-s**|**-**] [**-f** *session*] [**-o** *logfile*] [**-G** *gui*] [**-c** *command*|**+***command*] [**-t** *tag*] [**-w** *scroll*] [**-b** *blksize*] [*file*]...

DESCRIPTION

elvis is a text editor. It is intended to be a modern replacement for the classic *ex/vi* editor of UNIX fame. *elvis* supports many new features, including multiple edit buffers, multiple windows, multiple user interfaces (including an X11 interface), and a variety of display modes.

For a more complete description, you should see *elvis*'s on-line documentation. To view this documentation, start elvis and then give the command ":help".

To exit *elvis,* you can give the command ":q" in each of its windows. If you've modified the buffer in a window, and you want to abandon those changes, then give the command ":q!" instead.

OPTIONS

-V "Verbose" mode, causes elvis to output more status messages. You can use multiple **-V** flags to get even more detailed information. **-VVV** can be handy when elvis isn't initializing itself quite the way you expected. See also the **-o***logfile* option, described below.

-a Instructs elvis to load all files named on the command line *simultaneously* with a separate window for each.

-r This is used for recovering an edit session after a crash. Each elvis process uses a single "session file" to store the contents of all edit buffers. While elvis is running, a flag is set near the beginning of the session file so that other elvis processes won't try to use it at the same time. If an elvis process dies abnormally, though, it will leave the session file lying around with that flag set; the **-r** flag allows elvis to open a session file even if it is marked as being in use.

-R This sets the "defaultreadonly" option, which causes all new buffers to be marked as "readonly" so you won't accidentally overwrite the original file.

"ex" implies this.

-i Causes *elvis* to start each window in input mode, instead of visual command mode. Novice users may prefer this.

-S Sets the "safer" option, making *elvis* paranoid about certain potentially harmful commands. The ./.exrc file and modelines are executed with the "safer" option temporarily turned on regardless of whether **-S** was given. The **-S** flag is just used to make *elvis* permanently paranoid, for the duration of this process.

-f *session*
Makes elvis use the session file named "*session*" instead of the default file. Session files are discussed in the description of the **-r** flag, above.

-o *logfile*
Redirects messages and trace information out to *logfile* instead of going to stdout/stderr as usual. This is useful under Windows95, where stdout/stderr don't show anywhere. If you're having trouble configuring WinElvis, try running "winelvis -VVV -olog" and then you can find the trace and error messages in the file "log".

-G *gui* Makes elvis use the named *gui* user interface instead of the default. To see a list of supported user interfaces, give the command "elvis -?".

-c *command* or **+***command*
Gives an *ex* command to be executed after loading

the first file.

-s Read an *ex* script from stdin, and execute it. This is similar to the **-Gscript** flag, except that **-s** has the additional side-effect of bypassing all initialization scripts.

-t *tag* Causes editing to begin at the location where the given *tag* is defined. See the **ctags(1)** command for more information about tags.

-w *scroll*
This sets the "window" option, which has very little effect in *elvis.*

-b *blksize*
If a new session file is created, this causes it to use blocks of size *blksize*.

maintain backward compatibility, although their future use is discouraged.

+*command*
Like **-c** *command*, this causes the ex *command* to be executed after the first file is loaded. If the *command* is omitted, it is understood to be "$", which causes the cursor to move to the last line of the file.

- Like **-s**, this causes elvis to read a script from stdin and execute it.

TERMCAP INTERFACE

The termcap interface is the one you'll use most often on non-graphic terminals. It looks and acts a heck of a lot like the traditional vi. The biggest addition is the support for multiple windows. For more information on how to use multiple windows, start elvis and give the command ":help ^W". The short form of that help is: ^Ws splits the screen to form an additional window, ^Wq closes the window, and ^W^W switches the cursor from one window to another.

If your terminal supports ANSI color escape sequences, then you can use the ":color" command to assign different colors to the six basic fonts: normal, bold, italic, underlined, fixed, and emphasized. You must assign a normal color first, e.g., ":color normal yellow".

There are three additional options when using the termcap interface: term, ttyrows, and ttycolumns. The term option contains the name of the termcap entry being used; it should correspond to the type of terminal you're using. The ttyrows and ttycolumns options give the size of the screen.

The DOS and (text mode) Win32 version of the termcap interface support the mouse. The mouse behaves almost exactly like the X11 mouse, described below. The only differences are that the mouse can't be used to cut & paste to the clipboard, and on a two-button mouse you can simulate a middle button by simultaneously pressing the left and right buttons.

X11 INTERFACE

The x11 interface is used under X-Windows on UNIX systems. It provides a scrollbar and mouse support, and allows you to select which fonts to use.

X11 Options

To specify a normal font, use **-font** *fontname* or **-fn** *font-* don't specify a normal font, then elvis will use a font named "fixed" by default. (This default can be overridden by a "set normalfont=..." command in the *elvis.ini* file. The default *elvis.ini* file does this, making the new default font be 18-point Courier.)

To specify a bold font, use **-fb** *fontname.* The specified font should have the same size character cell as the normal font, but elvis does not verify this. If you don't specify a bold font, then elvis will fake it by smearing the normal font rightward one pixel.

To specify an italic font, use **-fi** *fontname.* The specified font should have the same size character cell as the normal font, but elvis does not verify this. If you don't specify an italic font, then elvis will fake it by sliding the top half of the normal font rightward one pixel.

If you want to use Courier fonts, there is a shortcut: **-courier** *size* will use the normal, bold, and italic versions of the Courier font in the requested size.

You can force elvis to use only black and white with the **-mono** flag; this is the default if your display only has one bitplane. For color displays, **-fg** *color* and **-bg** *color* can be used to set the normal text color and the background color, respectively.

Elvis has a built-in icon, which is generally a good thing. Some window managers won't allow you to assign a new icon to a program that has a built-in one, so elvis has a **-noicon** flag which disables the built-in icon.

The **-fork** client causes elvis to run in the background, so that your shell prompt returns immediately.

The **-client** option causes elvis to look for an already-running elvis process on the same X server and, if there is one, send the new arguments to it. This causes the old elvis process to create new windows for file arguments. The new elvis process then exits, leaving the old one to do the real work and allowing your shell program to prompt for a new command immediately. For the sake of uniformity, if **-client** fails, then a new elvis process starts up as though you had used the **-fork** argument instead.

The **-client** option is implemented in an interesting way: the client elvis simply sends a series of ex commands to an existing window of the server elvis. For each file name argument, the client elvis sends a ":split *file*" command. For **-t***tag*, the client elvis sends a ":stag *tag*" command. For **-c***command*, the client elvis simply sends the command is executed, for security reasons. Second, the command is executed by the server's existing window, not the new one, so (for example) "elvis -client -c 20 foo" creates a new window for the file "foo", and then moves the OLD WINDOW's cursor to line 20 of whatever file it was showing.

X11 Mouse

I've tried to reach a balance between the mouse behavior of **xterm(1)** and what makes sense for an editor. To do this right, elvis has to distinguish between clicking and dragging.

Dragging the mouse always selects text. Dragging with button 1 pressed (usually the left button) selects characters, dragging with button 2 (the middle button) selects a rectangular area, and dragging with button 3 (usually the right button) selects whole lines. These operations correspond to elvis' v, ^V, and V commands, respectively. When you release the button at the end of the drag, the selected text is immediately copied into an X11 cut buffer, so you can paste it into another application such as xterm. The text remains selected, so you can apply an operator command to it.

Clicking button 1 cancels any pending selection, and moves the cursor to the clicked-on character. Clicking button 3 moves the cursor without cancelling the pending selection; you can use this to extend a pending selection.

Clicking button 2 "pastes" text from the X11 cut butter. If you're entering an ex command line, the text will be pasted into the command line as though you had typed it. If you're in visual command mode or input mode, the text will be pasted into your edit buffer. When pasting, it doesn't matter where you click in the window; elvis always inserts the text at the position of the text cursor.

Double-clicking button 1 simulates a ^] keystroke, causing elvis to perform tag lookup on the clicked-on word. If elvis happens to be displaying an HTML document, then tag lookup pursues hypertext links so you can double-click on any underlined text to view the topic that describes that text. Double-clicking button 3 simulates a ^T keystroke, taking you back to where you did the last tag lookup.

ENVIRONMENT VARIABLES

HOME
: This is used to determine where your home directory is. It is always defined under UNIX. For WindowsNT, if HOME is undefined then elvis will derive it from the values of HOMEDRIVE and HOMEPATH, which are always defined; the default value is typically contains the elvis program as your home directory.

  This is significant because your home directory is a popular location for storing configuration files. The .exrc or ELVIS.RC file resides there.

SHELL
: This is the name of your command-line interpreter. Elvis needs to know this so you can run programs from inside elvis.

ELVISPATH
: If ELVISPATH is defined in the environment, then its value is copied into the elvispath option, which is a list of directories that elvis should search through when looking for its support files. If ELVISPATH is undefined, elvis will use a default list which usually includes the your home directory, and maybe a system-wide default location.

SESSIONPATH
: This is a list of directories where elvis might be able to create the session file. Elvis uses the first writable directory from the list, and ignores all others.

INCLUDE
This is a list of directories where the "syntax" display mode should look for #include files.

LC_ALL, LC_MESSAGES, or LANG
If LANG is defined, then elvis will look for for its message translations in $LANG/elvis.msg, and only use just plain elvis.msg if it can't find $LANG/elvis.msg. The LC_MESSAGES and LC_ALL environment variables work similarly, and take precedence over LANG.

EXINIT If defined, the value of this option is typically interpreted as a series of EX commands when elvis starts up. This behavior isn't built into elvis though; it is handled by the standard *elvis.ini* file.

Elvis has commands which can examine any environment variable. Because of this, a comprehensive list of environment variables isn't possible.

FILES

~ This is your home directory. Whenever elvis sees a ~ at the start of a pathname, it replaces the ~ with the value of the "home" option. The "home" option is initialized from the HOME environment the elvis program resides.

lib In this man-page, "lib" is a placeholder for the name of a directory in which elvis' configuration files reside. The "elvispath" option's value is a list of directories that elvis will check for each configuration file.

elvis*.ses
These are the default names for sessions files. Session files store the contents of all edit buffers for a given edit session. You can instruct elvis to use a specific session file via the **-f***session* command-line flag. Note that sessions specified via **-f***session* normally reside in the current directory, but when elvis chooses its own session file name, it will place it in the first writable file named in the SESSIONPATH environment variable.

tags This file stores the tags for the files in a directory. It is used by the :tag command, among others.

lib/elvis.ini
: This file contains EX commands which are executed whenever elvis starts up. Elvis searches through the ELVISPATH for the file.

~/.exrc or ~\elvis.rc
: The *.exrc* (for UNIX) or *elvis.rc* (for non-UNIX) file in your home directory will generally be executed as a series of EX commands, unless the EXINIT environment variable is defined. This behavior isn't built into elvis though; it is handled by the standard *elvis.ini* file.

./.exrc or elvis.rc
: The *.exrc* (for UNIX) or *elvis.rc* (for non-UNIX) file in the current directory is interpreted as a series of EX commands, but only if the **exrc** option has been set. This behavior isn't built into elvis; it is handled by the standard *elvis.ini* file.

lib/elvis.brf
: This file contains EX commands which are executed **B**efore **R**eading a **F**ile. Typically this is used to distinguish a binary file from a text file by examining the file name extension of the file. The "readeol" option is then set accordingly.

lib/elvis.arf
: doesn't contain an *elvis.arf* file, then elvis searches through the ELVISPATH for the file. Typically, this is used to set the **bufdisplay** option for a buffer, based on the file name extension of the file.

lib/elvis.bwf
: This file contains EX commands which are executed **B**efore **W**riting a **F**ile. If the current directory doesn't contain an *elvis.bwf* file, then elvis searches through the ELVISPATH for the file. Typically, *elvis.bwf* is used to rename the old file to "filename.BAK" before writing out the new file.

lib/elvis.lat
: This contains a series of ":digraph" commands which set up the digraph table appropriately for the ISO Latin-1 symbol set. The "lib/elvis.ini" file executes this script during initialization, except under OS/2, MS-DOS, or text-mode Win32.

lib/elvis.pc8
: This contains a series of ":digraph" commands which set up the digraph table appropriately for the PC-8 symbol set. This is the normal symbol set for MS-

DOS, OS/2, and text-mode Win32 (although the graphical Win32 uses "lib/elvis.lat"). The "lib/elvis.ini" file executes this script during initialization if under MS-DOS.

lib/elvis.msg
: This file's data is used for mapping the elvis' terse error messages into verbose messages.

lib/elvistrs.msg
: Elvis doesn't use this file directly; it is simply meant to serve as a resource from which you can extract the terse form of one of elvis' messages; you can then add the terse form and your own custom verbose form to the "lib/elvis.msg" file.

lib/elvis.syn
: This file controls how elvis' "syntax" display mode highlights the text for a given language.

lib/elvis.x11
: This file contains a series of ex command. This file is sourced by *elvis.ini* if elvis is using its X11 user interface. It configures up the toolbar and default colors.

lib/elvis.ali
: isn't the MS-DOS version) then these aliases will be loaded automatically. They are partly intended to serve as examples of what aliases can do and how to write them, but mostly these aliases are intended to be truly useful. To see a list of the aliases, give the command ":alias".

lib/elvis*.html
: These files contain the online documentation. If the current directory doesn't contain the required *elvis*.html* file, then elvis searches through the ELVISPATH for the file.

lib/*.man
: These contain the man pages -- shorter summaries of the programs, with descriptions of the command-line flags.

guix11/*.xpm and guix11/elvis.xbm
: These contain icon images for use with X-windows. The "elvis.xbm" image is a 2-color bitmap, and it is compiled into elvis. The other files, such as "elvis.xpm," contain color images. The "insticon.sh" shell script (which is invoked as part of the "make install" operation) tries to copy these into appropriate directories.

SEE ALSO

ex(1), vi(1), ctags(1)

You should also view the on-line documentation, via ":help".

AUTHOR

Steve Kirkendall
kirkenda@cs.pdx.edu

Man(1) output converted with man2html