

## LEMMY Version 0.5-E (ELVIS for Windows)

LEMMY is a VI clone for MS-Windows, Windows 95, and Windows NT based heavily on Steve Kirkendall's ELVIS. If you are a vi user, LEMMY will be familiar. If you've only heard of vi, and use MS-Windows, LEMMY may be the first vi you've ever found to be usable!

For those of you who have never heard of ELVIS, it is the VI clone that comes with LINUX distributions. If you haven't heard of LINUX, it is freely-distributable implementation of Unix for PC Compatibles with an 80386 or higher CPU.

LEMMY comes in 16-Bit and 32-Bit versions. The latest builds are always available via anonymous FTP at **ftp.accessone.com** in the **/pub/misc/jai** directory.

Menu of sections:

### What's New

What's new in version 0.5-E.

### VI Help

Help with VI commands. Disclaimer - THIS IS NOT A COMPLETE REFERENCE.

### Feedback

I am interested in hearing what people think of this and/or to hear about ways that it can be broken. Also where is the WWW site?

### Elvis man page

A copy of the man page from elvis. Contains important information on ELVIS initialization parameters and startup command line flags. These apply to LEMMY as well.

## Send your Comments - Version 0.5-E

Credit for ELVIS itself goes to Steve Kirkendall. This is a VI clone for windows, based heavily on ELVIS. It started life as a port, and is gradually evolving from there.

Thanks goes to all who have provided feedback, and have helped to make this a better product. You know who you are - the responses received have had a direct impact on the evolution and improvement of this editor.

Send comments or suggestions to my Internet E-Mail address below:

**[jai@accessone.com](mailto:jai@accessone.com)**

Here is the URL for my website, where updates are available:

**<http://www.accessone.com/~jai>**

## What's New - Version 0.5-E

Version 0.5-E contains a variety of enhancements that people have been asking for, and that improve its behavior in the 32-bit world. Enhancements include (but not limited to):

- \* **User definable colors.**
- \* **Support for long filenames and space-embedded filenames.**
- \* **Edited file history available from the "File" menu.**
- \* **Search/Replace pattern history.**
- \* **Definable startup window rows and columns.**
- \* **No longer requires BWCC.DLL.**
- \* **Mouse-drag selection option.**
- \* **Support for TrueType (scalable) fonts.**
- \* **Improved scrollbar scrolling.**

Jl

## ELVIS "man" page excerpts...

### NAME

elvis, ex, vi, view, input - The editor

### SYNOPSIS

elvis [flags] [+cmd] [files...]

### DESCRIPTION

Elvis is a text editor which emulates vi/ex.

On systems which pass the program name as an argument, such as Unix and Minix, you may also install elvis under the names "ex", "vi", "view", and "input". These extra names would normally be links to elvis; see the "ln" shell command.

When elvis is invoked as "vi", it behaves exactly as though it was invoked as "elvis". However, if you invoke elvis as "view", then the readonly option is set as though you had given it the "-R" flag. If you invoke elvis as "ex", then elvis will start up in the colon command mode instead of the visual command mode, as though you had given it the "-e" flag. If you invoke elvis as "input" or "edit", then elvis will start up in input mode, as though the "-i" flag was given.

### OPTIONS

-r To the real vi, this flag means that a previous edit should be recovered. Elvis, though, has a separate program, called elvrec(1), for recovering files. When you invoke elvis with -r, elvis will tell you to run elvrec.

-R This sets the "readonly" option, so you won't accidentally overwrite a file.

-t tag  
This causes elvis to start editing at the given tag.

-m [file]  
Elvis will search through file for something that looks like an error message from a compiler. It will then begin editing the source file that caused the error, with the cursor sitting on the line where the error was detected. If you don't explicitly name a file, then "errlist" is assumed.

-e Elvis will start up in colon command mode.

-v Elvis will start up in visual command mode.

-i Elvis will start up in input mode.

-w winsize  
Sets the "window" option's value to winsize.

+command or -c command

If you use the +command parameter, then after the first file is loaded command is executed as an EX command. A

typical example would be "elvis +237 foo", which would cause elvis to start editing foo and then move directly to line 237. The "-c command" variant was added for UNIX SysV compatibility.

#### FILES

/tmp/elv\*

During editing, elvis stores text in a temporary file. For UNIX, this file will usually be stored in the /tmp directory, and the first three characters will be "elv". For other systems, the temporary files may be stored someplace else; see the version-specific section of the documentation.

tags This is the database used by the :tags command and the -t option. It is usually created by the ctags(1) program.

.exrc or elvis.rc

On UNIX-like systems, a file called ".exrc" in your home directory is executed as a series of ex commands. A file by the same name may be executed in the current directory, too. On non-UNIX systems, ".exrc" is usually an invalid file name; there, the initialization file is called "elvis.rc" instead.

#### ENVIRONMENT

TERM This is the name of your terminal's entry in the termcap or terminfo database. The list of legal values varies from one system to another.

#### TERMCAP

Optional. If your system uses termcap, and the TERMCAP variable is unset, then  
will read your terminal's definition from /etc/termcap. If TERMCAP is set to the full pathname of a file (starting with a '/') then will look in the named file instead of /etc/termcap. If TERMCAP is set to a value which doesn't start with a '/', then its value is assumed to be the full termcap entry for your terminal.

#### TERMINFO

Optional. If your system uses terminfo, and the TERMINFO variable is unset, then  
will read your terminal's definition from the database in the /usr/lib/terminfo database. If TERMINFO is set, then its value is used as the database name to use instead of /usr/lib/terminfo.

#### LINES, COLUMNS

Optional. These variables, if set, will override the screen size values given in the termcap/terminfo for your terminal. On windowing systems such as X, has other ways of determining the screen size, so you should probably leave these variables unset.

#### EXINIT

Optional. This variable can hold EX commands which will be executed before any .exrc files.

#### SHELL

Optional. The SHELL variable sets the default value for the "shell" option, which determines which shell program is used to perform wildcard expansion in file names, and also which is used to execute filters or

external programs. The default value on UNIX systems is `"/bin/sh"`.

Note: Under MS-DOS, this variable is called COMSPEC instead of SHELL.

HOME This variable should be set to the name of your home directory. It looks for its initialization file there; if HOME is unset then the initialization file will not be executed.

TAGPATH Optional. This variable is used by the "ref" program, which is invoked by the shift-K, control-J, and :tag commands. See "ref" for more information.

TMP, TEMP These optional environment variables are only used in non-UNIX versions of . They allow you to supply a directory name to be used for storing temporary files.

SEE ALSO  
ctags(1), ref(1), virec(1)

Elvis - A Clone of Vi/Ex, the complete elvis documentation.

BUGS  
There is no LISP support. Certain other features are missing, too.

Auto-indent mode is not quite compatible with the real vi. Among other things, 0^D and ^^D don't do what you might expect.

Long lines are displayed differently. The real vi wraps long lines onto multiple rows of the screen, but elvis scrolls sideways.

AUTHOR  
Steve Kirkendall  
kirkenda@cs.pdx.edu

Many other people have worked to port elvis to various operating systems. To see who deserves credit, run the :version command from within elvis, or look in the system-specific section of the complete documentation.

# VI Help

The VI editor has many features and capabilities not documented here. It is my goal over time to make this a complete documentation. For now, it introduces the basics.

Click below to go to the appropriate section:

**VI Command Mode**

**EX Command Mode**

**INSERT/REPLACE Mode**

**Regular Expressions**

## VI Command mode

When in VI Command Mode, the keyboard is not what it seems. Every key, both upper and lowercase has a special meaning. The functions provided via the keys in command mode provide a variety of complex editing functions.

**Most commands accept multipliers before they are used.** For example, keying in '10j' will move the cursor ten characters to the right.

Below is a list of some of the more common key commands:

ke	what it does	new mode
y		
a	Append after cursor position	<u>INSERT</u>
b	Move back by word	
dd	Delete line(s)	
e	Move to end of word	
h	Cursor left	
i	Insert text before cursor	<u>INSERT</u>
j	Cursor down	
k	Cursor up	
l	Cursor right	
n	Repeat last search	
o	Open new line below current line	<u>INSERT</u>
p	Put yanked/deleted text after or below cursor	
r	Replace character at position	(replace 1)
s	Replace current character with characters	<u>INSERT</u>
u	Undo the last change	
w	Move forward by word	
x	Delete character at cursor	
yy	Yank line(s)	
A	Append text at end of line	<u>INSERT</u>
B	Move back by word ignoring punctuation	
C	Change text to end of current line	<u>INSERT</u>
D	Delete to end of current line	
E	Move to end of word ignoring punctuation	
G	Goto specified line	
H	Move cursor to home position (top left of screen)	
I	Insert text at beginning of line	<u>INSERT</u>
J	Join two lines	
L	Move cursor to last position in window	
M	Move cursor to middle of window	
N	Repeat last search in reverse	
O	Open new line above cursor	<u>INSERT</u>
P	Put yanked/deleted text before or above cursor	
R	Replace characters	<u>REPLACE</u>
S	Change entire line	<u>INSERT</u>
W	Move to beginning of next word, ignore punctuation	

X Delete character before cursor  
Y Copy line  
Z Save changes and exit  
Z  
: Issue an EX command  
/ Forward search  
? Backward search

EX  
EX/Search  
EX/SearchEX

## VI Insert and Replace modes.

INSERT or REPLACE modes are typically entered from VI Command mode when certain keyboard commands are issued. These are the modes that allow you to type text directly into your file.

When in INSERT or REPLACE mode, you are keying in text. The keyboard becomes what you'd expect. What you type is what you see. To get back to command mode, you use the **ESCAPE** key.

## EX Command Mode.

EX Mode is entered by hitting a ':' from VI Command mode. EX itself is a line-mode editor for UNIX which exists on its own. Its functionality is included in VI, and provides extended complex editing functions.

Here is a partial list of EX commands (again, by no means complete):

<b>Command</b>	<b>Description</b>
<code>:w</code>	Save current file.
<code>:w!</code>	Force save current file (exclamation point forces many operations).
<code>:wq</code>	Save and quit.
<code>:r !<i>command</i></code>	Execute the command, and insert its standard output at the cursor position.
<code>:r <i>file</i></code>	Insert the contents of the named file at the cursor position.
<code>:s/<i>pattern</i>/<i>with</i>/</code>	Replace ' <i>pattern</i> ' with ' <i>with</i> ' on current line.
<code>!<i>command</i></code>	Execute the command.
<code>:e!</code>	Throw away all edits and revert to original file contents.
<code>:<i>number</i></code>	Go to the specified line.
<code>:\$</code>	Go to the last line in the file.
<code>:f</code>	Show the current filename
<code>/<i>pattern</i></code>	Forward search.
<code>?<i>pattern</i></code>	Backward search.

In the above list, patterns can include [regular expressions](#) if the 'magic' option is enabled.

## Regular Expressions.

Regular expressions allow for powerful pattern matching. In a regular expression, certain characters have special meanings as to how to perform a match on the text.

Character	Meaning
.	Match any single character
*	Match zero or more of any character
^	Anchor at beginning of line (^dog only matches dog at the beginning of a line)
\$	Anchor at end of line (dog\$ only matches dog at the end of a line)
\	Take the following special character literally ( matches a period)
[...]	Match a single position to anything between the brackets (d[io]g matches dig or dog)
\(...\)	Very powerful. I'll include more info in a future version of this help.

