Quick intro to elvis 2.1, with links to source code and binaries

CONTENTS

---

1. About this file

This file is written in the HTML markup language. You can view it with any WWW viewer, such as Netscape. You can also use elvis 2.1i-beta to view it; this version of elvis has the ability to view HTML documents, and print them.

This file has many hypertext links. Use them! If you're using elvis 2.1 to browse this file, then hypertextual references will appear as underlined text. (Except on color PCs; since color video cards don't support underlining, hypertextual references will be colored -- white on red, by default.) To follow the hypertext link, move the cursor onto the underlined text and press (Enter). To go back, press (Control-T). The (Tab) key moves the cursor forward to the next hypertext reference.

If elvis 2.1 doesn't automatically start up in HTML mode when you view this file, then you'll need to force it into HTML mode by giving the command ":display html".

2. Differences between vi and elvis 2.1

Elvis is a superset of vi. It runs on more operating systems than vi, it is free, and you can obtain the source code. Elvis also has many new features. These new features are described in the first chapter of the online manual, which hypertext links to the other parts of the manual where those features are described in detail. Here’s a just brief list:

* Multiple edit buffers, so you can edit several files at once.
* Multiple windows, so you can see multiple edit buffers, or different parts of the same edit buffer.
* Multiple user interfaces, including graphical interfaces under Windows95/98/NT and X11.
* A variety of display modes, including syntax coloring and HTML.
* Online help, with hypertext links.
* Enhanced tags, to support overloading in C++.
* Network support, so you can load/save files via FTP, or even use elvis as a light-weight Web browser.
* Aliases, which allow you to define new ex commands.
* Built-in calculator

3. Differences between 2.0 and 2.1

The following is a summary of changes made since the release of elvis 2.0. These are in addition to bug changes.

3.1 New ex commands

:only
: This closes all windows except the current window.

```
:only
```

:browse
:sbrowse
: These build a list of all tags which match given criteria (e.g., all tags in a particular file), and make an HTML document with links to all the definition points in your source code. The :browse command saves the cursor position on the stack and then displays the table in the current window. The :sbrowse command creates a new window showing that table.

```
:browse foo.c
:sbrowse class:/Ball
```

:bbrowse
:sbbrowse
: These build an HTML document with links to all the edit buffers. The :bbrowse command saves the cursor position on the stack and then displays the table in the current window. The :sbbrowse command creates a new window showing that table.

```
:bbrowse
```

:alias
:unalias
: The :alias command creates an alias -- a new ex command which you can implement via a series of existing ex commands. Aliases can accept arguments, and perform complex operations. Many sample aliases are included in the distribution.

```
:alias lower !% s/.*/\L&/
```

:message
:warning
:error
: These output messages of various types. In particular, the :error message type has the side-effect of aborting any pending macros or aliases.

```
:if !< == !>
```

```
:then error Address range required
```

:local
This is handy in aliases. It is like :set, except that the options will be restored to their previous value when the alias exits.

```
:alias total {
        "Sum the integers in a range of lines
        local t=0 report=0 nosaveregexp
        !% s/[0-9]\+/let t=t+\1/x
        eval !% c (t)
}
```

:try
This executes a command, and then checks the success/failure indicator returned by that command. The then/else flag is set accordingly. Regardless of whether the command succeeded or not, :try itself always succeeds; this is significant because when a command fails (anywhere except as the argument of a :try command), any pending macros or aliases are aborted.

```
:alias togglecase {
        try !% s/.*/\U&
        else !% s/.*/\L&
}
```

:while
:do
This executes a command in a loop, as long as a given condition holds true.

```
:let i = 1
:while i <= 10
:do {
        calc i
        let i = i + 1
}
```

3.2 Options

Many new options have been added. Some exist only in a particular user interface, though. These are all described below.

bufid, bufferid
Each user buffer is automatically assigned a numeric identifier, which is stored in this option. These can be used later as a shorthand for either the buffer’s filename or buffername.

```
:e #1
:1,20 cop (1)$
```

locked, lock
When set, this option prevents any changes from being made to the buffer. This is a slightly stronger version of the "readonly" option.

readeol, reol
writeeol, weol
Together, these two options replace the old "binary" option. readeol can be set to one of **dos**, **unix**, **mac**, **text**, or **binary** to indicate the newline translations that were used to read a file into a buffer. The writeeol option can be set to any of those values, or the special value **same** to use the same translation as each buffer was read with.

partiallastline, pll
Traditionally, when vi is invoked on a file which doesn't end with a newline, vi adds one. Elvis does that too now, but since elvis can also be used on binary files it needs to remember whether it has added a newline, so it can omit that bogus newline when doing a binary write. That's what this option is for.

putstyle, ps
This is only meaningful for cut buffers, not normal edit buffers. It indicates whether the cutr buffer's contents should be pasted as a series of characters, as a rectangular block, or as whole lines. (Previously this information was stored in the first line of a cut buffer, but that made cut buffers hard to edit.)

matchchar, mc
The visual % command has been extended to match any pairs of characers. This option stores the list of character pairs.

completebinary, cob
Elvis performs file name completion -- you can type a partial file name and hit <Tab> to have elvis complete the name for you (or as much of the name as possible before ambiguities are encountered). Normally elvis ignores binary files when looking for possible completions; it only considers the names of text files. Setting this option makes it check the names of binary files as well as text files.

showname, snm
showtag, st
These cause elvis to display extra information at the bottom of the window, whenever it has nothing better to show there. showname causes it to display the name of the current file. showtag causes it to display the name of the tag which is defined on the current line, or the nearest line above it which has a tag. The showtag option can make elvis take more time to load files, since it must find the location of every tag for

that file.

true, True
false, False
These options mostly exist so you can use the symbols true and false in the expressions used by :if and the like. The value of the false option is used as a logical false value, just like "", "0", or "false"; anything else is considered to be a logical true value.

Any boolean option returns the value of the true option if it is set, or the false option if it is reset. Non-english speakers may wish to set true and false to the local words for those logical states. You can do that by setting those options explicitly, or by adding lines to the "elvis.msg" file (e.g., "true:verdad" and "false:falso").

submit, Submit
cancel, Cancel
help, Help
The "x11" user interface uses these like the true and false options, to translate a few words that it uses. Actually, help doesn't do anything yet, but the values of submit and cancel are used as the labels of buttons on dialog windows.

scrollbar, sb
toolbar, tb
statusbar, stb
menubar, mb
These enable or disable certain parts of the a GUI window. They are supported by the "windows" GUI (for Windows95/98/NT) and the "x11" GUI (for Unix). Actually menubar isn't support by "x11" because "x11" doesn't have a menu bar yet.

font, fnt
normalstyle, nfn
boldstyle, bfn
italicstyle, ifn
fixedstyle, ffn
emphasizedstyle, efn
underlinedstyle, nfn
These control the appearance of text in the "windows" interface (for Windows95/98/NT). font stores the base font, and the other options tell elvis how to derive a display font from the base font.

underline, uln
scrollbarleft, xsl
borderwidth, xbw
textcursor, tc
outlinemono, om
These are all "x11" options.

saveregexp, sre
Many scripts and aliases reset this. Normally when you perform a search or substitution, elvis remembers the regular expression

you used so you can perform the same search/substitution again later without retyping the whole regular expression. That's handy. But many aliases and scripts perform their own internal searches and substitutions, and those shouldn't cause elvis to forget the your regular expression. So aliases often do :local nosre to protect your regular expression.

tagprg, tp
tagprgonce, tpo
These allow you to use an external program to perform tag searches.

blkhit, bh
blkmiss, bm
blkwrite, bw
These provide extra information about elvis' internal block cache.

program, argv0
This stores the name by which elvis was invoked. The default "elvis.ini" file uses this information to distinguish between "ex" behavior and "view" behavior from the normal "vi" behavior.

lpnumber, lpn
lpheader, lph
These affect the way text is printed. Setting lpnumber causes line numbers to be printed, just as number causes line number to be shown on the screen. For the "normal" or "syntax" display modes, lpheader causes pages to have a header showing the file name, page number, and date.

previousdir, pdir
This stores the previous directory. You can use "~-" (tilde minus) at the start of a file name as a shorthand for the previous directory name.

3.3 New functions

The following functions have been added to the built-in calculator. This is used by :if, :let, and others.

htmlsafe(*string***)**
This replaces the &, <, >, and " characters with **&**, **<**, **>**, and **"**, respectively.

quote(*chars*, *str***)**
unquote(*chars*, *str***)**
These add or remove backslashes. quote() adds a backslash before each existing backslash, and before each character listed in the *chars* string. unquote() does the opposite.

current(*what***)**
This examines elvis' internal variables. It can be any of the following:

line	cursor's current line number
column	cursor's current column number
word	word at the cursor position
mode	mode name as reported by showmode option
next	next file in args list
prev	previous file in args list
tag	current tag as reported by showtag option

fileeol(*file*)
Examine the contents of a file, and make a guess about what value the readeol option should be set to.

getcwd()
Return the name of the current directory.

absolute(*file*)
Return the fully-specified name of a given file. In other words, if it is in the current directory then combine the current directory name with the file name.

alias(*name*)
Test for the existence of a given alias.

3.4 New ports, and new GUI features

Graphical Win32 port
In addition to a text-mode version, there is now a fully graphical version of elvis for Windows95/98/NT, named WinElvis. It is included in the Win32 binary distribution.

OS/2 port
A text-mode port of elvis for OS/2 has been completed.

X11 improvements
The "x11" user interface has been improved through the addition of a status bar and a configurable tool bar. Tool bar buttons can even have configurable dialog windows.

3.5 Miscellany

Name completion
The <Tab> key is used for name completion on ex command lines. It can complete file names, ex command names, option names, and tag names. You can also make it perform tag name completion in your own files by running ":set inputtab=identifier".

Tilde in a file name
Elvis 2.0 allowed you to use "~" at the start of a file name to

access files in your home directory. 2.1 adds the ability to use "~+" for the current directory's absolute path name, and "~-" for the previous directory. Also, under Unix you can use "~*user*" for the home directory of *user*. ~name, ~+, ~-

Network protocols

Elvis can read via HTTP, and read or write via FTP. Just use a Web URL in place of a file name, like this:

```
:e ftp://ftp.cs.pdx.edu/pub/elvis/README.html
```

Together with elvis' "html" display mode, this allows elvis to be used as a light-weight Web browser.

"tex" display mode

A "tex" display mode has been implemented. It handles the simplest TeX documents moderately well. Unlike real TeX formatters, though, elvis' "tex" mode is not programmable so it doesn't handle custom commands.

Enhanced tags

Elvis' tags implementation has been enhanced to support overloaded tags as required for C++. There is a whole chapter in the manual describing all the wonderful things you can do with tags now.

Multi-line ex clauses

Some ex commands accept another ex command as an argument. The :g command is an example of this. Traditionally, vi has allowed multiple commands by placing a backslash before the end of each line except the last, but this didn't always work. Elvis supports that, and adds a better alternative: If you give a '{' in place of the command, then elvis will read the following lines up to the next matching '}', and use the intervening lines as the ex command. This is both more reliable and more readable than the backslash notation.

Confirming substitutions

Elvis 2.0's implementation of the :s/// command lacked support for the c flag. Elvis 2.1 supports :s///c, at long last!

3. The future of elvis

One of the biggest tasks on my list is to rewrite the ctags program so that it will be able to parse C++ code better. It should read the language descriptions from "elvis.syn", and do at least a half-assed job of generating tags for any language described there. I’m also toying with the idea of a statistical tags generator, but that’s still pretty nebulous at this point.

Windowing will be more versatile. Currently the GUI versions of elvis always split detached windows; sometimes it would be nice if they could be attached to the existing window, as a horizontal or vertical pane.

I intend to add a true extension language to elvis someday, but it isn’t in 2.1. The language interface will be general enough to support a variety of languages. The first language supported will probably be PERL, followed rapidly by Python and TCL.

4. Links to related files

If the main site (ftp.cs.pdx.edu) is too slow, try the mirror site at ftp.false.com.

Most of the following are binary files, not text or HTML files, so you can't view then with your Web browser. But you can use your browser to download the files. For Netscape, use <Shift-Click>; for MSIE, use <RightClick> and "download".

untar.c
: This is the complete source code for "untar", a little program which extracts files from a gzipped tar archive. Comments near the top of "untar.c" describe how to compile and use it. If you already have the gzip and tar utilities, then you don't need this.

untardos.exe
: This is an MS-DOS executable, produced from the above "untar.c" file. It can also be run under Windows 3.1, in a Dos-prompt window. For brief instructions on how to use untardos, run it with no arguments.

untarw32.exe
: This is a Win32 executable, produced from the above "untar.c" file. It runs under WindowsNT and Windows95. It runs somewhat faster than the MS-DOS version. It also supports long file names. For brief instructions on how to use untarw32, run it with no arguments, in a text-mode window.

: **NOTE:** MS-Windows95 and MS-DOS use incompatible methods for mapping long file names to short ones. So if you extract the files under Windows95, DOS programs won't be able to find them with their expected names, and vice versa. Consequently, you must use untardos.exe to unpack elvis-2.1-msdos.tar.gz, and untarw32.exe to unpack elvis-2.1-win32.tar.gz.

untaros2.exe
: This is an OS/2 executable, produced from the above "untar.c" file. For brief instructions on how to use untaros2, run it with no arguments.

elvis-2.1.tar.gz
: This is a gzipped tar archive of the source code and documentation for Elvis 2.1 and its related programs.

elvis-2.1-msdos.tar.gz
: This archive contains the documentation and MS-DOS executables for Elvis 2.1.

elvis-2.1-win32.tar.gz
: This archive contains the documentation and Win32 executables for Elvis 2.1. These were compiled and tested under Windows95, but should work under WindowsNT 3.51 (or later) as well.

elvis-2.1-os2.tar.gz

This archive contains the documentation and OS/2 executables for Elvis 2.1.

ftp://ftp.fh-wedel.de/pub/fh-wedel/staff/herbert/elvis/00-index.html
This is where the OS/2 maintainer stores his most up-to-date versions. It may be better than the elvis-2.1-os2.tar.gz file, above.