Ex changes – Version 3.1 to 3.5

This update describes the new features and changes which have been made in converting from version 3.1 to 3.5 of *ex*. Each change is marked with the first version where it appeared.

Update to Ex Reference Manual

Command line options

3.4 A new command called *view* has been created. *View* is just like *vi* but it sets *readonly*.

3.4 The encryption code from the v7 editor is now part of *ex*. You can invoke *ex* with the −x option and it will ask for a key, as *ed*. The *ed x* command (to enter encryption mode from within the editor) is not available. This feature may not be available in all instances of *ex* due to memory limitations.

Commands

3.4 Provisions to handle the new process stopping features of the Berkeley TTY driver have been added. A new command, *stop*, takes you out of the editor cleanly and efficiently, returning you to the shell. Resuming the editor puts you back in command or visual mode, as appropriate. If *autowrite* is set and there are outstanding changes, a write is done first unless you say "stop!".

3.4 A

:vi <file>

command from visual mode is now treated the same as a

:edit <file> or :ex <file>

command. The meaning of the *vi* command from *ex* command mode is not affected.

3.3 A new command mode command *xit* (abbreviated *x*) has been added. This is the same as *wq* but will not bother to write if there have been no changes to the file.

Options

3.4 A read only mode now lets you guarantee you won't clobber your file by accident. You can set the on/off option *readonly* (*ro*), and writes will fail unless you use an ! after the write. Commands such as *x, ZZ,* the autowrite option, and in general anything that writes is affected. This option is turned on if you invoke ex with the −R flag.

3.4 The *wrapmargin* option is now usable. The way it works has been completely revamped. Now if you go past the margin (even in the middle of a word) the entire word is erased and rewritten on the next line. This changes the semantics of the number given to wrapmargin. 0 still means off. Any other number is still a distance from the right edge of the screen, but this location is now the right edge of the area where wraps can take place, instead of the left edge. Wrapmargin now behaves much like fill/nojustify mode in *nroff*.

3.3 The options *w300, w1200,* and *w9600* can be set. They are synonyms for *window,* but only apply at 300, 1200, or 9600 baud, respectively. Thus you can specify you want a 12 line window at 300 baud and a 23 line window at 1200 baud in your EXINIT with

:set w300=12 w1200=23

3.3 The new option *timeout* (default on) causes macros to time out after one second. Turn it off and they will wait forever. This is useful if you want multi character macros, but if your terminal sends escape sequences for arrow keys, it will be necessary to hit escape twice to get a beep.

3.3 The new option *remap* (default on) causes the editor to attempt to map the result of a macro mapping again until the mapping fails. This makes it possible, say, to map q to #

and #1 to something else and get q1 mapped to something else. Turning it off makes it possible to map ^L to l and map ^R to ^L without having ^R map to l.

3.3 The new (string) valued option *tags* allows you to specify a list of tag files, similar to the "path" variable of csh. The files are separated by spaces (which are entered preceded by a backslash) and are searched left to right. The default value is "tags /usr/lib/tags", which has the same effect as before. It is recommended that "tags" always be the first entry. On Ernie CoVax, /usr/lib/tags contains entries for the system defined library procedures from section 3 of the manual.

Environment enquiries

3.4 The editor now adopts the convention that a null string in the environment is the same as not being set. This applies to TERM, TERMCAP, and EXINIT.

Vi Tutorial Update

Deleted features

3.3 The "q" command from visual no longer works at all. You must use "Q" to get to ex command mode. The "q" command was deleted because of user complaints about hitting it by accident too often.

3.5 The provisions for changing the window size with a numeric prefix argument to certain visual commands have been deleted. The correct way to change the window size is to use the z command, for example z5<cr> to change the window to 5 lines.

3.3 The option "mapinput" is dead. It has been replaced by a much more powerful mechanism: ":map!".

Change in default option settings

3.3 The default window sizes have been changed. At 300 baud the window is now 8 lines (it was 1/2 the screen size). At 1200 baud the window is now 16 lines (it was 2/3 the screen size, which was usually also 16 for a typical 24 line CRT). At 9600 baud the window is still the full screen size. Any baud rate less than 1200 behaves like 300, any over 1200 like 9600. This change makes *vi* more usable on a large screen at slow speeds.

Vi commands

3.3 The command "ZZ" from vi is the same as ":x<cr>". This is the recommended way to leave the editor. Z must be typed twice to avoid hitting it accidently.

3.4 The command ^Z is the same as ":stop<cr>". Note that if you have an arrow key that sends ^Z the stop function will take priority over the arrow function. If you have your "susp" character set to something besides ^Z, that key will be honored as well.

3.3 It is now possible from visual to string several search expressions together separated by semicolons the same as command mode. For example, you can say

```
/foo/;/bar
```

from visual and it will move to the first "bar" after the next "foo". This also works within one line.

3.3 ^R is now the same as ^L on terminals where the right arrow key sends ^L (This includes the Televideo 912/920 and the ADM 31 terminals.)

3.4 The visual page motion commands ^F and ^B now treat any preceding counts as number of pages to move, instead of changes to the window size. That is, 2^F moves forward 2 pages.

Macros

3.3 The "mapinput" mechanism of version 3.1 has been replaced by a more powerful mechanism. An "!" can follow the word "**map**" in the *map* command. **Map!**'ed macros only apply during input mode, while **map**'ed macros only apply during command mode. Using "**map**" or "**map!**" by itself produces a listing of macros in the corresponding mode.

3.4 A word abbreviation mode is now available. You can define abbreviations with the *abbreviate* command

```
:abbr foo find outer otter
```

which maps "foo" to "find outer otter". Abbreviations can be turned off with the *unabbreviate* command. The syntax of these commands is identical to the *map* and *unmap* commands, except that the ! forms do not exist. Abbreviations are considered when in visual input mode only, and only affect whole words typed in, using the conservative definition. (Thus "foobar" will not be mapped as it would using "map!") Abbreviate and unabbreviate can be abbreviated to "ab" and "una", respectively.