3. INPUT MODE

You can't type text into your file directly from visual command mode. Instead, you must first give a command which will put you into input mode. The commands to do this are A, C, I, O, R, S, a, i, o, and s.

The S, s, C, and c commands temporarily place a $ at the end of the text that they are going to change.

In input mode, all keystrokes are inserted into the text at the cursor's position, except for the following:

KEY	FUNCTION
^@	(Nul) Insert a copy of last input text, then exit
^A	Insert a copy of the last input text
^D	Delete one indent character
^H	(Backspace) Erase the character before the cursor
^I	(Tab) Depends on the inputtab option
^K	Compose the next two characters as a digraph
^M	(Enter) Insert a newline (^J, linefeed)
^O	Read & run a single vi command, then resume input
^P	Insert the contents of the cut buffer
^R	Redraw the window, like ^L
^T	Insert an indent character
^U	Backspace to the beginning of the line
^V	Insert the following keystroke, even if special
^W	Backspace to the beginning of the current word
^X	Use the next two keystrokes as hex digits of 1 byte
^[	(ESCape) Exit from input mode, back to command mode

Also, on some systems, **^S** may stop output, **^Q** may restart output, and **^C** may interrupt execution.

The R visual command puts you in overtype mode, which is a slightly different form of input mode. In overtype mode, each time you insert a character, one of the old characters is deleted from the file.

3.1 Arrow keys in Input Mode

The arrow keys can be used to move the cursor in input mode. (This is an extension; the real Vi doesn't support arrow keys in input mode.) The **PgUp**, **PgDn**, **Home**, and **End** keys work in input mode, too. The **Delete** key deletes a single character in input mode. The **Insert** key toggles between input mode and replace mode.

3.2 Digraphs

Elvis supports digraphs as a way to enter non-ASCII characters. A digraph is a character which is composed of two other characters. For example, an apostrophe and the letter i could be defined as a digraph which is to be stored and displayed as an accented i.

There is no single standard for extended ASCII character sets, so elvis uses a configurable digraph table. You can view or edit the digraph table via the :digraph! ex command.

When inputing text, there are two ways you can type in a digraph. The first way is to type the first character, then hit **backspace**, and then type the second character. Elvis will then substitute the non-ASCII character in their place. Digraphs entered this way will not be recognized unless you've turned on the digraph option.

The other way is to type **^K** followed by the two characters, without a backspace. This method works regardless of the value of the digraph option.

3.2.1 Latin-1 digraphs

The most common standard for non-ASCII symbol sets is called Latin-1 (or ISO 8859/1, or ECMA-94 -- these are all the same). This symbol set is used by X-windows and graphical Windows95 programs, but not MS-DOS.

Elvis is distributed with a file named lib/elvis.lat which contains the :digraph commands necessary to set up the digraph table appropriately for Latin-1. Most of these digraphs are fairly straight-forward. For example, to generate an accented "i", type **^K**, **i**, and an apostrophe character. You can list all of the digraphs via the command ":dig!"

The default lib/elvis.ini file reads the lib/elvis.lat file automatically. This is very handy if your system really does use Latin-1, but if it doesn't then you'll probably want to disable the automatic loading. You can disable it by editing the lib/elvis.ini file, searching for the "source! (elvispath("elvis.lat"))" line, and inserting a double-quote character at the start of that line.

3.2.2 PC-8 digraphs

MS-DOS doesn't use Latin-1, and neither do OS/2 or Win32 console applications, such as elvis. So for Microsoft operating systems, elvis loads its digraph table via a file named lib/elvis.pc8. This sets up the digraph table appropriately for the 437 code page, which is Microsoft's default. If you use a different code page, you may wish to edit lib/elvis.pc8.

3.3 Abbreviations

Elvis can expand abbreviations for you. You define an abbreviation with the :abbr command, and then whenever you type in the abbreviated form while in input mode, Elvis will immediately replace it with the long form. COBOL programmers should find this useful. :-)

Elvis doesn't perform the substitution until you type a non-alphanumeric character to mark the end of the word. If you type a control-V before that non-alphanumeric character, then Elvis will not perform the substitution.

Elvis also requires the abbreviated text to either appear at the start of a line, or immediately after a non-alphanumeric character. This prevents elvis from trying to treat the ends of words as abbreviations. This is necessary so that if you set up an abbreviation which maps "ed" to "editor", elvis won't mangle "confused" into "confuseditor".

Abbreviations are allowed to contain punctuation characters. For example, **TeX** users might find it useful to map "\b" to "\begin".

3.4 Auto-Indent

With the autoindent option turned on, Elvis will automatically insert leading whitespace at the beginning of each new line that you type in. The leading whitespace is copied from the preceding line.

To add more leading whitespace, type control-T. To remove some whitespace, type control-D.

If you ":set noautotab", then the whitespace generated by control-T will always consist of spaces -- never tabs. Some people seem to prefer this.

Elvis' autoindent mode isn't 100% compatible with vi's. In Elvis, 0^D and ^^D don't work, ^U can wipe out all indentation, and sometimes Elvis will use a different amount of indentation than vi would.