<u>Command: R</u>

The previously compiled object code will be executed, but only if the source has not been expanded in the meantime - see Section 0.2 for more detail.

<u>Command: T n</u>

This is the 'T'ranslate command. The current source is compiled from line n (or from the start if n is omitted) and, if the compilation is successful, you will be prompted with 'Ok?'; if you answer 'Y' to this prompt then the object code produced by the compilation will be moved to the end of the runtimes (destroying the compiler) and then the runtimes and the object code will be dumped out to tape with a filename equal to that previously defined for the 'find' string. You may then, at a later stage, load this file into memory, using the HP4T loader, whereupon it will automatically execute the object program. Note that the object code is located at and moved to the end of the runtimes so that, after a 'T'ranslate you will need to reload the compiler - however this should present no problems since you are only likely to 'T'ranslate a program when it is fully working.

If you decide not to continue with the dump to tape then simply type any character other than 'Y' to the 'Ok?' prompt; control is returned to the editor which will still function perfectly since the object code was not moved.

<u>4.2.4 Other Commands.</u>

<u>Command: B</u>

This simply returns control to the operating system. For details of how to re-enter the compiler refer to the HP4T Alteration Guide and your Implementation Note.

<u>Command: O n,m</u>

Remember that text is held in memory in a tokenised form with leading spaces shortened into a one character count and all HP4T Reserved Words reduced to a one character token. However if you have somehow got some text in memory, perhaps from another editor, which is not tokenised then you can use the 'O' command to tokenise it for you. Text is read into a buffer in an expanded form and then put back into the file in a tokenised form - this may of course take a little time to perform. A line range must be specified, or the previously entered values will be assumed.

<u>Command: S d</u>

This command allows you to change the delimiter which is taken as separating the arguments in the command line. On entry to the editor the comma ',' is taken as the delimiter; this may be changed by the use of the 'S' command to the first character of the specified string d. Remember that once you have defined a new delimiter it must be used (even within the 'S' command) until another one is specified.

Note that the separator may not be a space.