

Thus the loader will automatically load the compiler and runtimes into memory for you.

When the compiler has been successfully loaded it will execute automatically and produce the message:

Top of RAM?

You should respond to this by either entering a positive decimal number up to 65536 (followed by RETURN) or by hitting RETURN (See Implementation Note).

If you enter a number then this is taken to represent the highest RAM location + 1, otherwise the first non-RAM location is automatically computed. The compiler's stack is set to this value, thus you can reserve high memory locations (perhaps for extensions to the compiler) by giving a value less than the true top of RAM. In the ZX Spectrum version the true top of RAM is taken to be the start of the user-defined graphics area (UDG in the Sinclair manual).

You will then be prompted with:

Top of RAM for 'T'

Here you can enter a decimal number or default to the 'Top of RAM' value previously specified. What you enter will be taken as the stack when the resultant object code is executed after using the editor 'T' command (See Section 4 for details). You will need to define a runtime stack different from the top of RAM if, for example, you have written extensions to the runtimes and wish to place them safely in high memory locations.

Finally you will be asked:

Table size?

What you enter here specifies the amount of memory to be allocated to the compiler's symbol table.

Again, either enter a positive decimal number followed by RETURN or simply RETURN by itself (in which case a default value of (available RAM divided by 16) will be taken as the symbol table size. In nearly all cases the default value provides more than enough space for symbols. The symbol table may not extend above machine address 23000\* (32766 decimal). If you specify so large a value that this happens then you will be prompted again for 'Top of RAM' etc.

You may, optionally, include an 'E' before the number after this prompt - if you do so then the internal line editor will not be retained for use by the compiler. So do this if you wish to use your own editor with the compiler (see the HP4T Installation Guide for details on how to do this).

At this point the compiler and integral editor (if retained) will be relocated at the end of the symbol table and execution transferred to the supported editor.

\*Note: throughout this Manual, the pound sign '#' is replaced by the number sign or hash (decimal) 35, hexadecimal 23, shift '3' on all systems which do not use U.K. ASCII. Numbers that are preceded by this symbol are in hexadecimal.

## 0.1. Scope of this manual.

This manual is not intended to teach you Pascal; you are referred to the excellent books given in the Bibliography if you are a newcomer to programming in Pascal.

This manual is a reference document, detailing the particular features of the soft Pascal 4.

Section 1 gives the syntax and the semantics expected by the compiler.

Section 2 details the various predefined identifiers that are available within Pascal 4, from CONSTANTS to FUNCTIONS.