

13 Special features

13.1 COUNT

To find out the number of words in your text, switch to command mode and type

COUNT **RETURN**

If you want to know the number of words in part of the text only, set markers 1 and 2 before and after the part concerned type

COUNT 1 2 **RETURN**

It is important, however, to recognise what VIEW is doing when it counts words. It would be truer to say it was counting the spaces between the words. VIEW cannot read, so the only way it can discover a word is by the fact that words have spaces each side of them. So if you have typed 'WORD' it will recognise it as one word. If you have typed 'W O R D' VIEW will count it as four words, but if you are aware of how VIEW goes about this task, COUNT is a very useful facility.

13.2 FORMAT

This is the global formatting command. It is similar in its effect to the FORMAT BLOCK command which formats paragraphs. FORMAT is used in command mode and can format all the text in memory.

As such it must be used with caution. It may be tempting to decide to use a narrower column, for example, and apply this throughout the whole text. All you have to do is set a shorter ruler, go into command mode and type:

FORMAT **RETURN**

However, you should check the text carefully before using FORMAT. Some parts of it may be unsuitable for formatting - for example where you have used a narrow column and insert side headings; or where you have a table which is not protected from the effects of formatting.

To limit the effect of the FORMAT command, you can set markers each side of the text to be formatted and type:

For more information on the effects of TAB on formatting, see chapter 5 of this book.

13.3 Editing BASIC programs

It is possible to use VIEW to edit BASIC programs. While you may not often wish to do this - since it is quite possible to edit them in BASIC itself - it is sometimes useful to be able to place a part of a program within a VIEW text. For example you may have invented a new BASIC routine and want to write an article describing how it works. It would be very boring to have to type all the program lines into a VIEW file, when you have already got them in a BASIC file on your disc. It could also lead to a lot of mistakes.

The method of getting BASIC files into VIEW and back into BASIC again is as follows.

To place a BASIC program in a file which VIEW can read

Type: *BASIC **RETURN** (To get into BASIC.)

Place the disc (or cassette) containing the BASIC program in the drive (or cassette recorder).

Type: LOAD "(filename)" **RETURN** (To load the program)

Type: *SPOOL (newfile) **RETURN** (To create a new file)

Type: LIST **RETURN** (To put the program into it.)

Type: *SPOOL **RETURN** (To close the new file.)

To read the program into VIEW and edit it

Type: *WORD **RETURN** (To get into VIEW.)

Type: NEW **RETURN** (To clear text mode.)

Type: READ (newfile) **RETURN** (To read in the file.)

You can now edit the program and SAVE it back in ' newfile or in another text file

if you wish.

To use the edited version as a BASIC program

Type: *BASIC **RETURN** (To get into BASIC.)

Type: *EXEC (newfile) **RETURN**

The program is now in memory and can be listed and run in BASIC.

The point of all this routine is that VIEW cannot use BASIC files, so you have to create a file with the program in it which VIEW can use. This is done by using the *SPOOL command, which makes a file out of whatever is on the screen - and the program is placed on the screen by listing it.

The *EXEC command has the opposite function, of making a file which BASIC can use out of a VIEW text - which must of course be in the correct form for a BASIC program.

Two points are worth stressing:

- Always use READ, not LOAD or you will have a completely unusable column of text in VIEW.
- Lines of more than 132 characters cannot be used.