

ACORNSOFT

Into

The word 'VIEW' is rendered in a large, bold, white sans-serif font. Behind each letter of 'VIEW' is a corresponding letter in a lighter shade of green or yellow, which is itself set against a background of more letters in various shades of green and yellow, creating a layered, 3D effect. The letters are slightly offset and overlapping, giving the impression of depth and movement.

VIEW

**An introduction to word processing
on the BBC Microcomputer**

Into VIEW

An introduction to word processing
on the BBC Microcomputer

ACORNSOFT

The VIEW Word Processor was designed and developed by Protechnic Computers Limited, Cambridge, England.

Into VIEW was written by Information Transfer Limited, Cambridge, England.

Copyright © Acornsoft Limited 1983, 1984

All rights reserved

First published in 1983 by Acornsoft Limited

No part of this book may be reproduced by any means without the prior consent of the copyright holder. The only exceptions are as provided for by the Copyright (photocopying) Act or for the purpose of review or in order for the software herein to be entered into a computer for the sole use of the owner of the book.

SECOND EDITION

ISBN 0907876 81 1

Published by Acornsoft Limited, Betjemen House, 104 Hills Road, Cambridge
CB2 1LQ, England
Telephone (0223) 316039

Contents

1 What is word processing?	1
<hr/>	
2 The VIEW word processor	3
2.1 BASIC	3
<hr/>	
3 Word processing with VIEW	5
3.1 Switching on	5
3.2 The command mode display	6
3.3 Text mode	6
3.4 The text mode display	7
3.5 Screen and page	7
3.6 Typing text	8
<hr/>	
4 Screen modes and ruler	11
4.1 The ruler	11
4.2 Do-it-yourself rulers	13
4.3 Further experiments with rulers	14
<hr/>	
5 Tabbing	15
5.1 Tabbing after typing	15
5.2 TAB characters	16
5.3 Text outside the ruler	17
5.4 Tables and formatting	18
<hr/>	
6 Saving and loading files	19
6.1 Disc systems	19
6.1.1 Setting up	19
6.1.2 Formatting	19
6.1.3 Using discs	19
6.1.4 Locking files	21
6.1.5 Help!	21
6.2 Cassette systems	21
6.2.1 Recording files	22
6.2.2 Reading files	22
6.2.3 Printing	23
6.3 Names of files	23

7 Printing	24
7.1 General procedure for printing	24
7.2 PRINT	25
7.3 SHEETS	26
7.4 Editing procedures	27
7.5 SCREEN	27
7.6 Highlights	28
7.7 Resetting highlight codes	29
7.8 Printing from cassette	29
8 Moving and changing text	30
9 Stored commands and page layout	34
9.1 Book and report work	35
9.2 Number registers	38
10 Macros	39
10.1 Modified Macros	40
10.2 Macros for mail shots	43
10.3 Automatic layout	45
11 Change . . . replace . . . search	49
11.1 REPLACE	50
11.2 SEARCH	50
11.3 Limited searching and changing	51
11.4 Finding and changing phrases	51
12 Formatting and margins	53
12.1 Protecting text	53
12.2 The use of TABs	54
12.3 The left margin	54
13 Special features	56
13.1 COUNT	56
13.2 FORMAT	56
13.3 Editing BASIC programs	57

14 Continuous processing	59
14.1 The EDIT method	59
14.2 Finishing	61
15 Conclusion	62
15.1 The Second Processor	62
15.2 Mode switches	62
15.3 *FX commands	62
15.4 Colours	62
16 List of commands	65
17 Index	68

1 What is word processing?

The easiest way to describe word processing is to contrast it with normal typing. Think about what happens when you type a document.

First you type it in rough so that it can be edited. You edit it, changing words, swapping paragraphs around, changing paragraph lengths, adding headings, redrafting some of it and putting other sections in tabular form.

After that you type it all again. You check it, correct it, retype parts in a narrower column with side headings, perhaps retype whole pages where there are too many corrections.

After all that, you have your 'top' copy. If you want other copies, you have to photocopy, use carbons, or type it all again.

All this may seem quite normal - until you try word processing.

With a word processor, you type the text in as before, with the difference that the text appears on a monitor or television screens instead of on paper. The continual rapping and buzzing of conventional typewriters is replaced by the soft rattle of keys, and when you make a mistake, instead of going to a great deal of trouble to make a fairly adequate correction, you have only to replace one character image on the screen with another and the job is done, quickly and perfectly.

You record your rough draft on a magnetic disc or on tape, and you can cause the printer to type out copies on paper for checking.

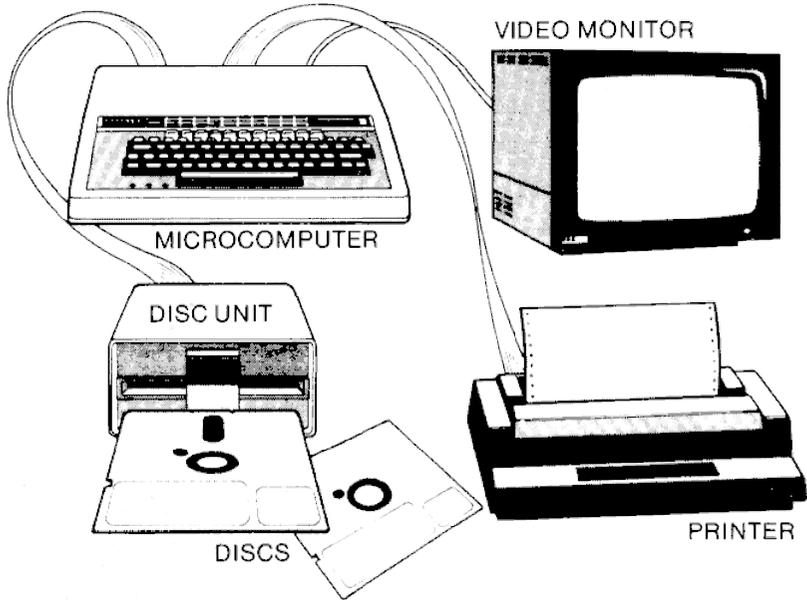
You edit the draft. When you come to 'retype' it you may find that large chunks of it are correct. With word processing there is no need ever to type these again, since they are recorded for you to use as many times as you like, in this or any other document.

You make your corrections on the screen, very easily. You insert and delete lines, move blocks of text around, and restructure the TABs even though the text is already typed! If you want to see what the text blocks like in a narrower column, you can try it out in a matter of seconds.

Then you print it out again, perhaps with three 'top' copies - since it is all recorded, this is just as easy as putting another piece of paper into the machine.

And if you want to send another, almost identical copy to firm XYZ Ltd instead of ABC Ltd, with changed names throughout, there are some very cunning and very easy ways of doing that too.

The item that makes all the difference in a word processor is the microcomputer that controls it all. A typical word processor layout is like this:



All the processing is done in the microcomputer and the results are displayed on the monitor, before being recorded for future use on discs, and printed out.

2 The VIEW word processor

The VIEW word processor package is designed to do all the things described in the last chapter, and much more besides. It is especially convenient in that the processor is installed in the BBC Microcomputer. When you want word processing you have only to request it. You can even arrange your computer in such a way that when you switch on you are immediately in word processing mode.

A typical VIEW word processing layout consists of:

- A BBC Microcomputer Model B, with a VIEW word processing ROM fitted. (ROM means 'Read Only Memory' and the VIEW ROM contains all the memory that is needed to convert the BBC Microcomputer into a word processor. It should be fitted by your dealer, but if you are fitting it yourself see the instructions issued with the ROM.)
- A BBC Microcomputer disc drive.
- A good quality black and white video monitor (for example a Kaga Denshi or MicroVitec). Some people claim that a green screen tires the eyes less. You can also use a colour monitor or, less satisfactorily, a television set.
- A printer. What printer you choose depends on your requirements. If you want the text to look as if it is typed on an electric typewriter, you should choose a good daisy wheel printer.

To connect up your computer, monitor, disc unit and printer, see the manuals that accompany these pieces of equipment.

Before starting to use VIEW you should slip the function key card under the clear plastic strip at the top of the computer keyboard, lining it up so that **FORMAT BLOCK** comes immediately above the red function key **f0**.

The function key card will be your guide when issuing commands to VIEW while you are processing text.

2.1 BASIC

Having the VIEW word processor ROM installed in your computer does not prevent you from using the machine as a normal computer whenever you wish. To

switch back to BASIC, type

*BASIC **RETURN**

If you are already in BASIC and want to begin word processing, type

*WORD **RETURN**

BREAK

Pressing the **BREAK** key leaves the system in command mode, including the No text message, which means that your text has vanished. To recover text lost in this way type OLD and press **RETURN**.

3 Word processing with VIEW

This introduction to VIEW is designed to be read in two ways.

Ideally you should place it beside your computer, with your VIEW GUIDE alongside, and work with both of them so that you really get to know how word processing is done on a BBC Microcomputer - and get to know the GUIDE too, so that it becomes your constant reference for word processing methods.

If you do not always have ready access to a computer, you may prefer to read this introduction separately, trying out methods it suggests whenever you get the chance.

However you study the system remember that the objective of word processing is to process words - with keyboard, screen and printer. So for the rest of this book we shall assume that you are sitting at your computer trying everything out as you go. The more often this is the case the quicker you will learn.

3.1 Switching on

Whenever you switch on you may be in VIEW immediately, or you may have to ask the computer for it.

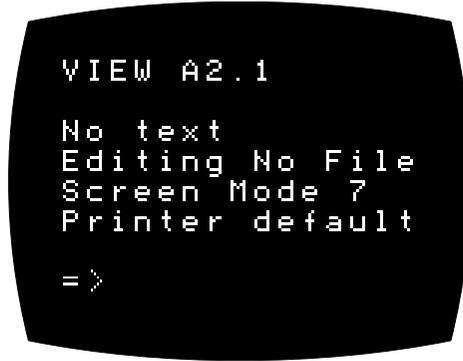
VIEW is encoded on a ROM - a 'Read Only Memory' - which is plugged into one of four sockets in your computer. What you see when you switch the computer on depends on where VIEW is plugged in and what other ROMs are plugged in with it. If you see the command mode display (see below) you can go ahead immediately with word processing. If not, type

*WORD **RETURN**

to get into VIEW.

Feel free to experiment with any keys - you cannot do the computer any harm by pressing them.

3.2 The command mode display



```
VIEW A2.1
No text
Editing No File
Screen Mode 7
Printer default
=>
```

The top line of the display shows the version of VIEW you are using. The words `No text` mean just that: when you enter text into VIEW or load a file from disc or tape, the words `No text` will change to the amount of memory available. `Editing No File` will also change when you load a text file, to show the name of the file. `Screen Mode 7` is the BBC Microcomputer's Teletext mode, giving 24 lines of 34 characters each on the screen. `Printer default` refers to the printer driver. Briefly this is a program which manages printing, and the word 'default' implies that VIEW's built-in printer driver program is active. Other drivers can be loaded for particular printers. The sign `=>` shows where commands are typed in.

3.3 Text mode

If you wish to write text, you need to open up text mode. This is done by typing

NEW **RETURN**

Once you have done this, you can switch at will between command mode (the mode in which you issue general commands to the system) and text mode (the mode in which you write text). To switch between these modes press **ESCAPE**.

Notice that the words `No text` have changed to `Bytes Free`. As a rough guide, one byte corresponds to one character, ie one letter, number, sign or space.

Switch to text mode by pressing **ESCAPE** and try some typing. You will soon discover that you are typing in capitals all the time. This is because at start-up the 'caps lock' is on. To switch it off press the **CAPS LOCK** key.

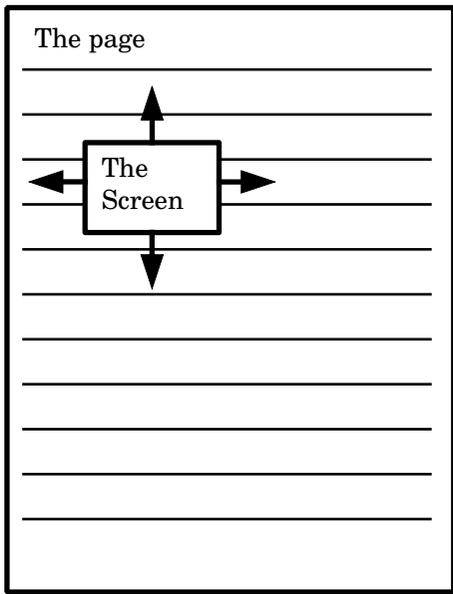
3.4 The text mode display

```
FJ .....*.....*.....*.....*.<
_
*****
```

The top line is the ruler, and this controls the width of the column of text under it. The asterisks in the ruler are TAB stops. The letter F means format and J means justify; these are explained in detail later in this chapter. The horizontal bar is the cursor, and shows where any new text will be typed. The row of asterisks marks the bottom of the column of text.

3.5 Screen and page

The four arrow keys move the cursor about the screen, but if you use them at this stage it is important to realise what they are doing. VIEW' stext area is not limited to the screen itself. The best way to regard the text area is a very large ' page'only a little of which is visible to you through the screen. The screen in fact is rather like a window which you can move (using the arrow keys) to any part of the page you wish.



3.6 Typing text

When you are in text mode, you can use the commands on the red function keys. Look at the function key card, which you should keep inserted under the clear plastic strip at the top of the keyboard. The most frequently needed commands are on the bottom row, and these are used without pressing either **SHIFT** or **CTRL**. Type in anything you like and try them out.

Now let us try out some of the features of VIEW word processing. Type in the paragraph below, without ever pressing **RETURN**, since the system will move you on to another line automatically, as soon as you type a word beyond the right margin.

When typing in text on the VIEW word processor, there is no need to press RETURN unless you wish to go on to another line without having reached the margin on the current one. Text is automatically formatted as you type it in. If you are using Mode 7 it is formatted on to a 34-character line.

Notice how each line you type is justified, ie the right-hand margin is even.

If this is not happening you have switched off the justification by mistake. At this stage the best thing to do is to return to the start-up position as follows.

- Press **BREAK** to reset the system.
- Press **CAPS LOCK** to switch that off.
- Type NEW and press **RETURN**.
- Press **ESCAPE** to get back into text mode.

... and start again.

Having typed in your text, correct any mistakes. If you made none just rewrite a bit instead. (Do not worry at this stage if any characters go beyond the right margin.)

To replace characters, use the arrow keys to place the cursor under the character you want replaced. Then simply type over the character and the new character will replace the old one.

To delete a character, place the cursor under the character you want to delete and press **DELETE CHARACTER** - which is function key **f9**.

To insert a character, place the cursor in the character space to the right of where you want the new character to be, like this:

WORD PROCESSING

Then press **INSERT CHARACTER** (function key **f8**). The text will open up like this:

WORD PRO_ESSING

and you can insert the new character.

You can also try inserting and deleting lines. The **INSERT LINE** command (function key **f6**) inserts a blank line above the line which the cursor is on, and the **DELETE LINE** command (function key **f7**) deletes the line which the cursor is on.

TOP OF TEXT and **BOTTOM OF TEXT** are easily demonstrated too, as are **BEGINNING OF LINE** and **END OF LINE**. At first you will probably find yourself using the arrow keys most of the time to move the cursor about, but as you become more proficient you will find these more powerful commands save a lot of time. All these commands are described in the **GUIDE** in more detail.

When you have finished making all your changes, you will probably find that your text no longer has that neat, justified right edge you started with. Probably it looks rather like our version below.

When typing in a paragraph on the **VIEW** word processor, there is no need to press **RETURN** unless you wish to go on to another line without having reached the margin on the current one. Text is automatically formatted as you type it in. If you are using Mode 7 it is formatted on to a 34-character line.

You can tidy up a column of text like this by reformatting it. Place the cursor anywhere on the top line and press function key **f0**, immediately below the words **FORMAT BLOCK** on the function key card. Make sure you do not have

' shift lock' on or you will get a bleep and no action.

The whole paragraph is then changed into something like this:

When typing in a paragraph on the VIEW word processor, there is no need to press RETURN unless you wish to go on to another line without having reached the margin on the current one. Text is automatically formatted as you type it in. If you are using Mode 7 it is formatted on to a 34-character line.

Of course not everyone likes justified text, and for some purposes it looks too formal. To switch off justification, hold down **CTRL** and press function key **f3** (labelled JUSTIFY MODE on the function key card). The small letter J at top left on the screen disappears, although the F remains - we are still formatting.

To rearrange the text in unjustified lines, all you have to do is to carry out the same procedure as before: place the cursor on the top line and press the function key for FORMAT BLOCK. The result should be something like this:

When typing in a paragraph on the VIEW word processor, there is no need to press RETURN unless you wish to go on to another line without having reached the margin on the current one. Text is automatically formatted as you type it in. If you are using Mode 7 it is formatted on to a 34-character line.

4 Screen modes and ruler

So far everything in this book can be done in screen mode 7, which is probably the mode in which your BBC Microcomputer will start up (ie unless you have arranged for it to start otherwise).

There are in fact eight possible screen modes, and VIEW will work in any of them. In practice most people use MODE 3 since its 74-character line is very suitable for laying out typewritten material.

Mode	Characters	Lines	Mode	Characters	Lines
MODE 0	74	31	MODE 1	34	31
MODE 2	34	31	MODE 3	74	24
MODE 4	34	31	MODE 5	16	31
MODE 6	34	24	MODE 7	34	24

To switch to another mode, type MODE followed by the mode number and **RETURN**.

You should be aware of the computer's memory if you are to use VIEW effectively, and the mode you are in affects the amount of memory available for text. Switch to command mode by pressing **ESCAPE** and you will see the amount of free memory shown in the form Bytes free ...

This shows the amount of memory left for you to use after the text already there is taken into account. A byte is a unit of memory, equivalent to a single character on the screen. However, the different screen modes themselves take up different amounts of memory.

If you start in MODE 7, you have about 25000 bytes of memory to play with, whereas MODE 3 only allows you about 10000 and in MODE 0 you are down to about 6000 words before you start.

On balance MODE 3 probably offers the most useful compromise. As a rough guide, a typical A4 page takes about 2000 bytes.

4.1 The ruler

For some time now you will have been aware of the ruler, and of its functions in limiting the length of the line when you are typing in or formatting text. Below is the default ruler you will see in MODE 3:

Now move the cursor on to the top line of the paragraph and press **FORMAT BLOCK** (function key **f0**). Suddenly the whole paragraph is reset to a different line length, like this:

Formatting relates the length of the text lines to the current ruler. Press **FORMAT MODE** to turn on (F is displayed at top left). Press **FORMAT MODE** again to turn off (nothing is displayed). In format mode, when text is typed in, any word which overflows the right margin is transferred whole to the following line.

This method works equally well with justified and unjustified text. Try a few more line lengths for yourself.

Whatever you do with the rulers, remember the basic rule: *the system obeys the last ruler above the line that the cursor is on.*

4.2 Do-it-yourself rulers

Since you are allowed as many as 128 rulers in any document you can feel free to use rulers wherever it is convenient to do so. Modifying the default ruler is often the best way to start, but having modified it once you may feel it easier to make another adjustment to your modified ruler, rather than start from scratch with another default ruler.

Copying the current ruler can be done quite simply by holding down **SHIFT** and pressing **COPY**. The copy ruler will appear on the line where the cursor is.

Alternatively if you want something quite unlike the default ruler you may find it easiest to make up your own ruler. To do this first place the cursor on the line where you want it; then hold down **CTRL** and press function key **f8** (**MARK AS RULER**). Two dots appear in the margin and you can make up your own ruler to the right of these.

Whenever you are making up your own ruler or modifying an existing or default ruler, you should always make sure that the finished product actually looks like a ruler. For example, it would be possible to make a perfectly valid ruler consisting simply of two dots in the left margin and a margin stop on the right, but it would also be fatally easy to delete such a ruler by mistake, spoiling all the text under

it. So fill it in with dots at least.

We also suggest that you do not alter the left margin except for a special purpose which will be mentioned later in this book.

4.3 Further experiments with rulers

Try out what happens if you have a very long ruler. Try extending the ruler beyond the right-hand edge of the screen, using INSERT CHARACTER key and filling up the spaces with dots.

If you then type in text, as the cursor reaches the right-hand edge of the screen, the text to the left disappears and you find yourself typing on into blackness.

What has happened is that the screen has moved to the right over the 'page', as described in the last chapter.

In fact you may construct a ruler which is up to 132 characters wide. However you should remember that the end product of word processing is printing words onto paper, so you have to make sure that your printer can handle a line of this length.

The text behaves in this way because the size of the TAB characters has been changed by the resetting of the TAB stops on the ruler.

If you wish to work with TABs a good deal you should study the relevant pages of the *GUIDE*.

5.3 Text outside the ruler

Having got this far with the placing and manipulation of text, we are ready for a rather more sophisticated operation. You have probably noticed in many reports, legal documents and leaflets that the main text often occupies the central part of the page only, with side headings and comment to left and right.

Since we need the ruler to align and format all text, how can we produce a layout like that?

The answer is to type the text first and the headings afterwards, like this.

```
.. ..>.....<
      The left margin is set
      towards the middle of the
      page to leave room for
      headings and notes at the
      sides. The next can be
      justified or not as required
      and TABs can be set as
      usual.
```

After the text is typed, release the margins by pressing **RELEASE MARGINS (SHIFT and function key f2)**. Then move the cursor to the positions where you want the side headings to be, using the arrow keys. Type the headings and notes in like this:

```
.. ..>.....<
SIDE HEADINS The left margin is set NOTES AND
TYPED towards the middle of the REFERENCES
HERE page to leave room for TYPED HERE
      headings and notes at the
      sides. The next can be
      justified or not as required
      and TABs can be set as
      usual.
```

. . . and when you have finished the job, don't forget to restore the margins by pressing `RELEASE MARGINS` (**SHIFT** and function key **f2**) again.

Note: If you are using the method described here, always format your text before placing the headings and notes. If you try to format it afterwards, `VIEW` will assume that you want everything in the lines concerned included in the formatting, so all your headings will be collapsed into the text and you will have to start again!

5.4 Tables and formatting

Formatting text can do a great deal of damage to tables embedded in it, unless you take action to prevent the damage. This is related to the way in which formatting deals with `TAB`s.

Suppose you have a line with a `TAB` in it which you format - with `FORMAT BLOCK`, global format, or the formatting that occurs when you reach the end of a justified line. In such a case `VIEW` treats the `TAB` as a space, and so will redistribute spaces in the line to accommodate the `TAB`. Naturally this would ruin any tabular layout.

The easiest way to prevent this from happening is to start each line of your table with a single `TAB`, or with one or more spaces. This acts as a signal to `VIEW` and it will not disturb the table.

If you want the table to line up with the left margin the simplest method is to place a ruler above the table with no right margin stop on it. This prevents formatting and protects the table.

These methods are described in more detail in chapter 12.

6 Saving and loading files

By now you must be getting impatient at seeing your disc unit or cassette recorder standing there doing nothing, to say nothing of your printer, if you have one. So this chapter is devoted to getting them working.

Advice on the disc unit and cassette recorder in this book and the VIEW Guide is confined to matters affecting word processing. For connecting up and general use refer to your *BBC Microcomputer System User Guide* and *Disc Filing System User Guide*.

6.1 Disc systems

6.1.1 Setting up

Before you can use discs your system must be in disc mode. It may in fact be set to default to disc mode (ie this may be its normal state) in which case you can go ahead and use the disc drive immediately. If it defaults to cassette mode, however, you must first enter command mode and type

```
*DISC RETURN
```

6.1.2 Formatting

Before you can record anything onto a disc you must format the disc. The Disc Filing System User Guide tells you how to do this. The formatting routine can be carried out while the system is in VIEW, using command mode. Note, however, that formatting a disc will destroy any text currently in the computer's memory.

6.1.3 Using Discs

Saving and loading files is all done in command mode. To save a file, use

```
SAVE (filename) RETURN
```

If there is another file of that name already on the disc, this command wipes it out and replaces it with the new one - which can be very useful for updating texts, but very frustrating if you do it by mistake. So always keep back-up copies of the texts you cannot afford to lose. (You can't say we didn't warn you! Of course we realise that nobody makes adequate back-up copies at first, and everybody has to

learn the hard way . . . that' s life!)

Taking a file from the disc and placing it in the computer' s memory is done with the command **LOAD**:

```
LOAD (filename) RETURN
```

or

```
L (filename) RETURN
```

Before the new file is loaded, any text currently in the computer' s memory is wiped. If you wish not to wipe all the current text but to keep it and add the new text from the file to it, you should use the command **READ**:

```
READ (filename) RETURN
```

So whether you **READ** or **LOAD** depends on whether you want to add to what is already in the computer' s memory, or whether you want to replace it with new text.

Save a few pieces of text onto the disc for yourself, checking the procedure in the Guide. Try out **LOAD** and **READ** and watch their effects. If you want to delete any files you have saved, use

```
*DELETE (filename) RETURN
```

but do not delete all the files, since we need some later for printing.

A useful variant on the **SAVE** command described above is to use **SAVE** by itself. Whenever you load a file the editing line in the command mode message shows the name of the file you have loaded:

```
Editing (filename)
```

If you then modify the file and want to return it to the disc under the same name, as an updated version, you do not need to name it in the **SAVE** command. Simply type

```
SAVE RETURN
```

and it will be saved under the name given in the editing line. This can be a time saver, but it is always worth checking the editing line to make sure that the name there is really the one under which you want to save the text.

6.1.4 Locking files

If there are any files you particularly wish to preserve, you can 'lock' them with the command

```
*ACCESS (filename) L RETURN
```

Try this with one of the files you have saved. Once a file is locked, you cannot update it by saving another file of the same name. If you try to, the system replies `File Locked`. To unlock it, use the `ACCESS` command again without the `L`.

To find out what files are on the disc (including which files are locked) type

```
*. RETURN or *CAT RETURN
```

Locked files are marked with the letter `L`.

6.1.5 Help!

It may be helpful to know that there is a list of disc commands available on the screen, which can serve as a rapid reminder. You can consult the list by typing

```
*HELP DFS RETURN
```

However, for the finer points of disc operation you should consult the *Disc Filing System User Guide*.

6.2 Cassette systems

If at all possible, use a cassette recorder with a remote control (REM) socket. Otherwise you will find yourself constantly rewinding and estimating positions on the tape.

Before you use a cassette recorder, your system must be in cassette mode. It may in fact be set to default to cassette mode (ie this may be its normal state), in which case you can go ahead and use the cassette recorder immediately. If it defaults to disc, however, you must first enter command mode and type

```
*TAPE RETURN
```

6.2.1 Recording files

Make sure there is a blank cassette in the recorder.

Type

```
SAVE (filename) RETURN
```

The 'cassette motor' lamp turns on. The message appears on the screen:

```
RECORD THEN RETURN
```

Press **RECORD** on your cassette recorder and **RETURN** on the computer. When the prompt (=>) reappears, the file is recorded.

If your cassette recorder has no motor control (**REM**) socket, stop it quickly. If it has a **REM** connection it will stop automatically.

6.2.2 Reading files

When using **VIEW** with cassettes, there is no **LOAD** command, ie no command which automatically wipes the current text from memory and replaces it with new text from a tape file. Instead there is the **READ** command which adds new text from the file to any text currently in memory. If you wish to wipe text from memory first type

```
NEW RETURN
```

Wind back the cassette to the appropriate point.

Type

```
READ (filename) RETURN
```

The 'cassette motor' lamp comes on.

Press **PLAY** on the cassette recorder. Depending on the cassette recorder, you may hear high- or low-pitched sounds. When the prompt (=>) reappears the file has been read in.

6.2.3 Printing

If you have difficulty in printing out files from cassette, in particular if you keep getting `Block?` messages, you may be able to solve the problem by an alternative method of recording onto cassette. This is covered in the next chapter.

6.3 Name of files

One final point affecting both disc and tape: how should you name your files?

One thing to beware of is the use of spaces in filenames. For example, you might type the first chapter of a report and save it under the name 'ABC' you might then go on to type the next chapter and save it in a second file which, quite naturally, you would call 'ABC 2' .

Unfortunately the computer takes the space between the C and the 2 to indicate that this is the end of the filename. To the computer, therefore, you have saved one file called 'ABC' and then another file called 'ABC' which overwrites the first. The cure for this is to miss out the space and call the second file 'ABC2' .

Think out your filenames carefully. Ideally they should be unique. If some of them relate to each other (like chapters in a report) they should reflect this fact. It is also best if they actually mean something and are not just random sequences of letters which you will have difficulty in remembering.

7 Printing

Printing can be done in two ways:

- Printing out whatever text is currently held in the computer's memory.
- Printing directly from a file on disc or tape.

If you print directly from a file on disc or tape, this has no effect at all on the text in memory, so there is no reason why you should not pause while editing the text in memory to print out text from files.

For general advice on the management of printers, see your *BBC Microcomputer System User Guide*, and the manual for the printer itself. In these pages we shall concentrate on the word processing aspects.

When you have everything set up so that you can print out text quickly and easily, printing will seem a very straightforward operation, but there is a good deal more going on behind the simple command `PRINT`.

`VIEW` does not itself directly manage the printer. Instead it sends codes to a *printer driver program* which in turn sends codes to the printer to produce the effect you have specified in your text and stored commands. This may seem an unnecessarily complicated way of running things, but it has an important benefit: `VIEW` can be made to work with many different printers, simply by using different printer driver programs.

In fact `VIEW` itself contains a 'default' printer driver program, which is adequate for straightforward printing on most printers. But if you are using a high quality daisy wheel printer such as a RICOH or QUME you need some way of taking advantage of the special effects these printers can provide - such as bold or underlined type.

This is done by installing a special printer driver program for that printer. Programs are available for many printers, and the *Printer Drivers* booklet goes into much more detail on their use and how they work.

7.1 General procedure for printing

If you need a printer driver to run your printer, load it from disc or cassette. The command is `PRINTER` followed by the name of the printer and **RETURN**. For example:

PRINTER RICOH **RETURN**

If your printer of the serial RS423 type, the computer must be switched to this type of output. The printer driver may do this for you, but space for the driver's facilities is limited, so you may have selected RS423 output for yourself. See the *Printer Driver Generator* package published by Acornsoft. To select for RS423 output type:

*FX5, 2 **RETURN**

For similar reasons you may have to set the baud rate. This is done with a command beginning *FX8 followed by a digit selecting the rate as follows.

Baud rate	Command
75	*FX8, 1
150	*FX8, 2
300	*FX8, 3
1200	*FX8, 4
2400	*FX8, 5
4800	*FX8, 6
9600	*FX8, 7
19200	*FX8, 8

Your printer may allow microspacing. The normal way in which VIEW justifies a line is by distributing any spaces left over throughout the line, adding the extra spaces to some of the spaces already present in the line. This is the standard practice because many printers only have whole character spaces.

Some printers however have a facility for adjusting the position of characters by 120th of an inch. This allows VIEW to adjust the line more finely by dividing the spaces left over into units of 120th of an inch and distributing these units evenly throughout the line, giving a much neater appearance.

This is microspacing. If you printer and printer driver have this facility, you can call it into action by typing:

MICROSPACE **RETURN**

If you are not sure, type it anyway, and if it is not available VIEW will tell you so.

Use the command PRINT or SHEETS to print the text, as described below.

7.2 PRINT

This is very straightforward to use. It prints out the entire file from disc or cassette, or the entire text in memory.

To print out the text in memory, use

```
PRINT RETURN
```

To print a file, use

```
PRINT (filename) RETURN
```

or, if you want to print several files continuously, use

```
PRINT (filename1 filename2 filename3...etc) RETURN
```

When VIEW is printing and comes to the end of a page, either because the set number of lines is used up or because it encounters a 'pageject' it carries out a page eject (ie the printer winds on whatever number of lines would be needed to take care of the bottom margin on one page and the header margin on the next) and just carries on printing. Headers and footers, and other matters concerned with page layout, are dealt with in chapter 9.

If you use PRINT with several files in succession, you can set automatic page numbering (see chapter 9) at the beginning of the first file and it will continue throughout the batch.

To stop printing press **ESCAPE**.

7.3 SHEETS

If you are using separate pages fed into the printer one at a time you will have to use this command. It is also useful when you do not want to print all the pages in the file, since it allows you to miss out pages.

To print out the text in memory, type

```
SHEETS RETURN
```

To print a file type

```
SHEETS (filename) RETURN
```

or, if you want to print several files continuously, use

SHEETS (filename1 filename2 filename3...etc) **RETURN**

The prompt appears:

Page 1..

To print, press any key except M, Q, **ESCAPE**, **COPY** or **BREAK**

To miss out a page, press M

To stop printing, press Q or **ESCAPE**

When page 1 is printed or missed out, VIEW goes on to prompt Page 2.. etc. until the file is finished.

7.4 Editing procedures

The fact that VIEW allows you to print out from memory using either PRINT or SHEETS can be useful for revising a lengthy report, for example.

Suppose you have already printed out pages 1 to 20 and subsequently want to revise page 9. You load the file and carry out your revisions. You record the amended file back on disc.

All you need to print again is page 9. The simplest way of doing this is to delete everything in memory coming before and after page 9, by setting markers 1 and 2 each side of the text to be deleted and pressing DELETE BLOCK (**CTRL f0**). This is quite safe since the full version remains on disc. Then use PRINT or SHEETS alone, and page 9, which is all that remains in memory, will be printed.

7.5 SCREEN

The SCREEN command allows you to 'flickthrough' the pages of a file, or of the text in memory, to get an idea of how it will be printed out. In this way you can check the formatting and see where the page breaks occur.

To display the text in memory, type

SCREEN **RETURN**

To display the contents of a file, type

SCREEN (filename) **RETURN**

To display the contents of several files, type

SCREEN (filename1 filename2 filename3...etc) **RETURN**

As soon as you press **RETURN** the first page moves up the screen. To move on to the next screenful press **SHIFT**. If you have specified headers or footers (for example, page numbers) these will be displayed.

7.6 Highlights

We mentioned earlier that some printers permit special effects such as underlined or bold type. Text to be printed in such a way is marked using the HIGHLIGHT 1 and HIGHLIGHT 2 keys.

When the text is printed, these highlight codes signal to the printer driver that a special effect is required, and the printer driver sends instructions to the printer.

To mark highlight 1

- Place the cursor under the first character of the text concerned and press HIGHLIGHT 1. The text will jump to the right and an underbar will appear.
- Move the cursor to the space after the last character of the text concerned and press HIGHLIGHT 1 again. A second underbar will appear and if there is any text to the right it will move to make room for the underbar.

Text marked for highlight 1 looks like this on the screen:

See the _VIEW Guide_ for details.

and would normally result in:

See the VIEW Guide for details.

To mark highlight 2 use the same procedure as for highlight 1. Instead of an underbar the marker in the text is an asterisk. Text marked for highlight 2 looks like this on the screen:

See the *VIEW Guide* for details.

and would normally result in:

See the **VIEW Guide** for details.

7.7 Resetting highlight codes

We say that these would ' normally be the results because some printer drivers and printers have other facilities than underlining and bold type - facilities such as an additional character set or superscripts.

Since there are only two highlight codes, VIEW allows you to reset these codes temporarily to send alternative signals to the printer driver. The stored command HT is used for this, and is described in more detail in the VIEW Guide.

7.8 Printing from cassette

When printing from cassette VIEW reads in one block of text at a time and prints it before reading in the next block. On some cassette recorders the BBC Microcomputer cannot stop the cassette motor quickly enough to prevent the tape running on to the next block. If it moves into the next block you will get a Block? error message.

Obviously one way to avoid the problem is to load the file into memory and print out from memory, as described earlier in the chapter.

Alternatively you can re-record the file, leaving more tape between blocks. This is done as follows:

Type: *OPT 3,10 **RETURN**

Type: WRITE (filename) **RETURN**

For more information on *OPT consult the *BBC Microcomputer System User Guide*.

8 Moving and changing text

One of the pleasures of word processing is the ease with which you can modify text - inserting words, moving whole paragraphs from one point to another, copying sections and deleting them.

We have already done a little of this, inserting and deleting a character at a time, but VIEW has much more to offer.

Type in the following example to try out the more powerful commands.

----- * -----

Word processing in business

Introducing word processing into a business represents an opportunity to think about parts of the business that involve the production of print, and about how print fits into the business.

If you think of a word processing system as a glorified typewriter, you will not get the best out of it.

Now suppose you get that printed out, and then proceed to edit it, rather heavily like this.

----- * -----

Word processing in business /upper case

Introducing word processing into a business represents an opportunity to think about parts of the business that involve the production of printed text and about how print fits into the business.

those

unrivalled

ed text

If you think of a word processing system as a glorified typewriter, you will not get the best out of it.

----- * -----

Editing text like this is done by moving the cursor and using the red function keys for command such as INSERT CHARACTER, DELETE CHARACTER, INSERT MODE, SWAP CASE, and the commands involved in moving blocks of text such as MOVE BLOCK.

Changing the title to upper case letters only is a simple matter - the SWAP CASE commands is indicated. Place the cursor under the 'o' of 'Word' and press **SHIFT f1**. The case changes from lower to upper and the cursor moves on. All you do is keep pressing SWAP CASE until the whole title is converted. In practice of course you might decide to retype a short heading like this, but remember that retyping is where many mistakes occur.

Now to change 'opportunity' into 'an unrivalled opportunity'. Place the cursor under the first 'o' of 'opportunity' and press **CTRL f4**. This puts VIEW into INSERT MODE so that any characters typed will cause the text to the right to move right to make room for the new text. A letter I appears at top left on the screen to remind you. Type in the additional word with a space after it and move on to the other two corrections which can be dealt with in the same way. When you have done them all, cancel INSERT MODE by pressing **CTRL f4** again.

The next job is to move the three lines at the bottom to the top under the title. Operations such as moving, copying to deleting blocks of text are controlled by setting markers. To move a piece of text, set markers 1 and 2 before and after it as follows:

- Place the cursor on the line above the text you wish to move.
- Press SET MARKER (**SHIFT f7**) and MK appears at top left on screen.
- Press 1.
- Place the cursor on the line below the text you wish to move.
- Press SET MARKER again.
- Press 2.

Carry out these operations slowly and carefully. If you make a mistake you can just set the markers again. You can also cancel markers 1 and 2 by changing to command mode and typing

CLEAR **RETURN**

Having set markers 1 and 2, move the cursor up to the blank line under the title and press MOVE BLOCK (**SHIFT f0**). The three lines will appear there and the other lines will open up to make room for them.

Finally we want to copy the rule design at the top and place it at the bottom. Once again we set markers above and below the rule, and place the cursor where we want it copied to, but this time instead of pressing a red function key we press the **COPY** key at bottom right of the keyboard.

After that the text should look like this.

----- * -----

Word processing in business

If you think of a word processing system as a glorified typewriter, you will not get the best out of it.

Introducing word processing into a business represents an unrivalled opportunity to think about those parts of the business that involve the production of printed text, and about how print fits into the business.

----- * -----

To justify the text again you have only to reformat the second paragraph by placing the cursor anywhere on the top line of that paragraph and pressing **FORMAT BLOCK** (function key **f0**) and the job is done.

----- * -----

Word processing in business

If you think of a word processing system as a glorified typewriter, you will not get the best out of it.

Introducing word processing into a business represents an unrivalled opportunity to think about those parts of the business that involve the production of printed text, and about how print fits into the business.

----- * -----

9 Stored commands and page layout

We have already used many immediate commands, such as DELETE CHARACTER and SET MARKER, and some command mode commands such as SAVE and LOAD.

VIEW also has a third set of commands which are entered in text mode, but do not have any effect at all as they are entered. These are stored commands, and their purpose is to influence the way the document is printed.

Perhaps the most obvious in its effect is PE, which means 'page eject'. This is a command to the printer to stop printing and eject the page.

Stored commands are used in text mode, and they appear in the left margin, under the two dots to the left of the ruler. It is as if they were marginal notes, telling the printer what to do as it progresses down the text.

The stored command page eject would be entered like this:

- Position the cursor on the line where you want the command.
- Press EDIT COMMAND (**SHIFT f8**).
- The cursor moves into the left margin.
- Type PE **RETURN**.

When the file is printed, as soon as the printer comes to the line on which you entered PE the page will eject.

Some of the other stored commands are used like this, some with numbers beside them, and some with text. For example suppose you want to tell the printer to work to a page length of 45 lines. The stored command for this is PL (page length) and it is entered in the margin just like PE. Beside it on the same line goes the number of lines: PL 45.

- Press EDIT COMMAND (**SHIFT f8**).
- The cursor moves into the left margin.
- Type PL **RETURN**.
- Type 45.

Similarly if you want to centre the words ' TheEuropean Community' on a line, you can do so by typing it on the left and placing the stored command CE beside it, like this:

```
CE The European Community
```

If you want to range text to the right, as for example with instructions on a form, you would use the stored command RJ like this:

```
RJ Membership subscription £.....  
RJ Handbook £.....  
RJ Badge £.....  
RJ Magazine £.....
```

```
RJ _____
```

```
RJ TOTAL £_____
```

The effect when printed would be:

```
Membership subscription £.....  
                    Handbook £.....  
                    Badge £.....  
                    Magazine £.....
```

```
                    _____  
                    £_____
```

Line spacing can also be controlled with stored commands. VIEW normally assumes solid text, but if you want part of your document with extra spacing between the lines you can use the command LS followed by the number of lines spacing, eg LS 2.

9.1 Book and report work

Stored commands come into their own when you are processing books and reports, although they can be very useful for letters too. In book and report work, you need to have at your disposal at least some of the features normally associated with the printing of books, such as top and bottom margin setting, lines at the head and foot of the page (headers and footers), and methods of emphasising type.

-)
-)Top margin, four lines
-)
-)
-)Header, one line
-)
-)Header margin, four lines
-)
-)

Lissajoux figures

Lissajoux figures are fascinating patterns that can form the basis for many weird and wonderful programs. The method for drawing Lissajoux figures is similar to the polar-coordinate method for drawing a circle.

For the circle, the angle from which the x- and y-coordinates are derived is the same. Different Lissajoux figures are obtained when these angles are out of phase.

There are many ways in which the basic Lissajoux patterns can be enhanced. Here is a program that uses straight lines to join the points that trace out two intermeshing figures. The pattern obtained depends on the random numbers chosen at lines 50 and 60.

-)
-)Footer margin, four lines
-)
-)
-)Footer, one line
-)
-)Bottom margin, four lines
-)
-)

VIEW provides all these. In its standard (default) layout it assumes four blank lines at the top of the page, followed by a header line, containing the title or series heading for the page. Below this are another four blank lines, after which the text begins. Footers (normally containing page numbers and report codes) are similarly separated from the text and from the bottom of the page by four blank lines.

Of course all these spacings can be changed and the headers and footers themselves can be cancelled if you wish. Remember that once you define a header or footer, VIEW will continue to print it until you issue other instructions. So if you define your header as ' CHAPTER3' you will have every page after that headed ' CHAPTER 3' until you redefine the header as ' CHAPTER 4' .

column, and select the bar chart option you convert the column into a horizontal bar chart. Of course it is necessary to make the

the same column of numbers on the sheet you can show the effect of the bar chart facility by moving the cursor and changing some of them. The sheet will recalculate and the line of

The illustration shows a page from a guide to the spreadsheet package 'ViewSheet'. To make up this page all spacings have been reset, and headers and footers have been redefined.

The top margin has been reset to three spaces, instead of the default four, by the stored command `TM 3` which is used in the same way as the page length command described above.

The header margin, ie the lines between the header and the text, has also been reset with `HM 3`. Footer and bottom margins have been similarly reset to two and three lines respectively with the commands `FM 3` and `BM 2`.

The header has been defined so as to display both the title of the book and the title of the section. Headers always have three components: left, centre and right.

The define header command is `DH`, and when this is used it must always be followed by the three components divided by spacers (normally the slash `/`). If one of the components is blank, simply place two slashes together. So the general rule is:

`DH /left component/centre component/right component/`

In the present case the header would have left and right components but no centre:

Footers can be set in exactly the same way as headers, and also have left, centre and right components.

Once a header or footer has been defined VIEW will print it on every page until it is cancelled. To cancel a header use the stored command:

```
HE
```

To cancel a footer use:

```
FO 0
```

The Guide goes into all this in more detail.

9.2 Number registers

VIEW has 26 number registers, labelled A to Z. Only two of these are allocated: P for pages and L for lines. The rest you can set as you wish.

In fact it is possible to set number register P to the number of your first, and it will automatically increase by one every time the printer completes a page. All you have to do then is to define your footer so as to print out the value of P and your page number will be printed automatically.

To set register P to value 1 for the first page, use the stored command SR (set register), like this:

```
SR P 1
```

You have then only to define the footer as the value of P. In order to distinguish between the character P and the value of register P you must place the vertical line symbol in front of it. So the footer is defined as:

```
DF //|P//
```

That is to say: no left component, centred page number, no right component.

Other stored commands which can cause number registers to be printed out are: DH (define header), CE (centre), RJ (right justify) and LJ (left justify). In fact the command LJ exists simply for the purpose of printing out registers, since VIEW always justifies left unless told to do otherwise.

10 Macros

One of the main reasons for using word processors is that you can make the computer do all the really boring jobs.

Suppose you have to type the same letter or invitation 30 times over. The chances are that you will make half a dozen mistakes in your letters which you will only discover when it is too late.

VIEW offers a facility which avoids all such problems. It allows you to make up any collection of text and commands into a package. You give the package a name, and you can then cause the whole package to be printed as often as you like, merely by entering that name in the margin like a stored command.

Such packages are known as 'macros'. To show how they work, suppose you have to send out innumerable copies of the following invitation.

```
        ABC Computers Ltd
request the pleasure of your company
        at the launching of the
```

```
        LOGOMANIAC
```

```
        Word Processing Package
        at the Queen's Hotel
on Friday 13 February at 12.00pm
```

Obviously you would use the CE command to centre all the lines, like this:

```
CE ABC Computers Ltd
CE request the pleasure of your company
CE at the launching of the
```

```
CE LOGOMANIAC
```

```
CE Word Processing Package
CE at the Queen's Hotel
CE on Friday 13 February at 12.00pm
```

To make this into a macro so that you could print out many copies with the least possible trouble you need to:

- Give it a name.
- Signal to VIEW that it is a macro.
- ' Call' it, ie to tell VIEW to use it.

Names of macros can consist of any combination of two letters that is not already used as a stored command. The stored command to define (give a name to) a macro is DM, so to name the macro AZ we simply place before it the stored command: DM AZ and after it the stored command EM (which of course means ' end macro').

The finished macro therefore looks like this:

```
DM AZ
CE ABC Computers Ltd
CE request the pleasure of your company
CE at the launching of the

CE LOGOMANIAC

CE Word Processing Package
CE at the Queen's Hotel
CE on Friday 13 February at 12.00pm
EM
```

When you have written, defined, named and ended your macro you have only to use it. As it stands it will not print out at all. To make it print you have to call it, ie to enter its name in the margin as if it were a stored command. Press EDIT COMMAND (**SHIFT f6**), type AZ and press **RETURN**.

Try entering it several times as a test. When you set up the printer and type PRINT, the text will be printed out as many times as you have entered the macro' s name in the margin.

10.1 Modified macros

When composing standard letters, invitations and other documents, we frequently need a piece of text which is essentially the same in each example, but has to be modified so as to ' personaliseit. Normal copying techniques fail here, but VIEW has a special facility which makes its macros even more effective.

Suppose we want to make up a standard, all-purpose invitation, which can be used to invite anybody to anything.

To start with we shall try a macro with just one modified item. Suppose we want to design an invitation into which any name can be inserted, an invitation that begins like this:

```
          ABC Computers Ltd
request the pleasure of the company of
          Mr Bill Brewer
```

In cases like this, VIEW allows you in effect to leave blanks in the macro into which you can fit any name you like at the printing stage. The method is to replace the words which are to be changed with the symbols @0, @1, @2 . . . @9. Each time the macro is called for printing, these parameters are filled in by typing the words on the same line as its name.

In this case we would simply place @0 in place of the name:

```
DM BY
CE ABC Computers Ltd
CE request the pleasure and company of
CE @0
CE at the launching of the

CE LOGOMANIAC

CE Word Processing Package
CE at the Queen's Hotel
CE on Friday 13 February at 12.00pm
EM
```

Later in the text, you enter the macro as if it were a stored command, with the appropriate name beside it.

```
BY Mr Bill Brewer
BY Mr Jan Stewer
BY Mr Peter Gurney
```

After that, all you have to do is to issue a print command and the macro will be printed, its blanks filled in, as many times as you have entered its name in the margin.

Developing this into a genuinely all-purpose invitation is merely a matter of extending the number of parameters, so as to leave blanks for almost everything.

```
.....
requests the pleasure of the company of
.....
on the occasion of
.....
.....
at .....
on .....at.....
```

Try it yourself, before looking at the finished macro below.

```
DM CX
CE @0
CE requests the pleasure of the company of
CE @1
CE on the occasion of
CE @2
CE @3
CE at @4
CE on @5 at @6
EM
```

When you come to use it the method is much the same as before:

```
CX XYZ Computers Ltd, Jim Smith, the lanuching, of ASTROCALC,
... and so on.
```

One variation is worth noting. Suppose we want to invite Tom, Dick and Harry, who naturally go everywhere together. What about the comma after ' Tom' ? Commas are used as spacers between parameters, so if we put it in it would have the wrong effect. To avoid this problem VIEW has the rule that if you want to include commas in a parameter you place it in angle brackets, like this:

CX XYZ Computers Ltd,<Tom, Dick and Harry>,the launching ...

Apart from this there is virtually no restriction in the use of macros. The macro below, for example, consists almost entirely of commands. Its purpose is to provide an automatic layout for the top of a letter.

DM LH

RJ @0

@1

@2

@3

@4

RJ @5

Dear @6

CE @7

EM

Try for yourself setting out the parameter line to make the beginning of the letter read:

25 June 1984

John Smith Esq
25 Alpha Road
Middletown
Loamshire

Ref. 25/3

Dear Mr Smith

Hire Purchase Agreement

10.2 Macros for mail shots

Using macros in this way, you can develop quite a sophisticated system for standard letters, all of which can be individually addressed and have a number of different key words and numbers in their contents.

Suppose we wish to send the following letter to applicants for a course, making changes of date, student, course, fee, etc.

Mr A B Carter 15 Sept 1984
10 Old Street
Newtown CX3 9JJ

Dear Mr Carter

Thank you for your application. We confirm your enrolment for the Intermediate Course C. Your student number is 552.

You will receive further details in a few days. Meanwhile will you please send the acceptance payment as shown in our brochure of £22.50.

If you have any queries please address them to Dept 4B.

Yours sincerely

The equivalent macro would be:

```
DM LL
@0           @1 Sept 1984
@2
@3
```

Dear @4

Thank you for your application. We confirm your enrolment for the Intermediate Course C. Your student number is 552.

You will receive further details in a few days. Meanwhile will you please send the acceptance payment as shown in our brochure of £22.50.

If you have any queries please address them to Dept 4B.

Yours sincerely

EM

and appropriate parameter lines would be as follows:

```
LL Mr A B Carter,15,10 Old Street,Newtown CX3 9JJ,Mr Carter,C,552,£22.50,4B
```

```
LL Ms Jane Brown,15,The Cottage,<Stenby, Powys>,Ms Brown,433,£19.25,9A
```

10.3 Automatic Layout

If you need to produce reports or books to a standard format, it is possible to use macros in conjunction with number registers to automate the layout in quite a sophisticated way.

Suppose you produce a series of reports, each with many chapters with headings and sub-headings, all numbered as follows:

CHAPTER 1

Chapter title

1.1 Section heading

1.1.1 Sub-section heading

Text
Text text text text text text text text text text text text text
Text text text text text text text text text text text text text text

1.2 Section heading

Text
Text text text text text text text text text text text text text text
Text text text text text text text text text text text text text text

Page 1

The most efficient way to manage such a series would be to set up a file containing macros to control all the standard features: spacing, bold type, and in particular numbers. For example, if we set register C to 1, and then cause a macro to increase register C by 1 each time a chapter heading is printed, we can print chapter headings completely automatically. Numbers for section and sub-section headings, and page numbers could be dealt with in the same way.

In the case of chapter headings the settings of the register would take the form:

SR C 1

and each time a chapter heading was printed a macro would be called containing the line:

SR C |C+1

which means that the new value of C is set to 1 greater than the old value of C. Later in the macro the value of C would be printed out by

LJ *CHAPTER |C*

which uses the LJ (Left justify) command to print out the chapter heading, the asterisks to print it in bold, and the vertical bar to indicate that it is register C that it is being printed, not the letter C.

Using these principles we can construct a set of macros to manage a whole series of reports. The macros would be saved on disc, and used as part of the print command whenever a report was printed. For example if the set of macros is given the name ' BOOK and the report files are ' A1 to ' A5 the print command would be:

```
PRINT BOOK A1 A2 A3 A4 A5
```

The macros below are an example of how this can be done. The way they do this may not be immediately apparent, but they will repay study.

```
DF ///Page|P/      (Define footers to display page numbers as
SR P 0             register P which is automatically increased
SR C 0             by 1 as each page is printed; also set
SR S 0             chapter, section and sub-section heading
SR T 0             numbers to zero.)
```

Chapter heading macro:

```
DM CH              (Define chapter heading macro.)
SR C |C+1          (Increase chapter number by 1.)
SR S 0             (Set section number to 0.)
SR T 0             (Set sub-section number to 0.)
PE                (Eject page - so chapter begins on new page.)
LJ *CHAPTER |C*   (Print chapter heading.)
CE *@0*           (Print chapter title, bold, centred.)
EM
```

Section heading macro:

```
DM SE              (Define section heading macro.)
SR S |S+1          (Increase section number by 1.)
SR T 0             (Set sub-section number to 0.)
PE 5               (Eject page if within five lines of bottom.)
LJ *|C.|S @0*     (Print chapter and section numbers, and
EM                section heading in bold.)
```

Sub-section heading macro:

```
DM  SS                (Define sub-section heading macro.)
SR  T |T+1           (Increase sub-section number by 1.)
PE  3                (Eject page if within three lines of bottom.)
LJ  *|C.|S.|T @0*   (Print chapter, section, and sub-section
EM                                numbers, and sub-heading in bold.)
```

When the text is written in VIEW, the footers appear automatically, showing the page numbers, and the numbers in the chapter, section and sub-section headings are provided by the macros, the words of these headings being parameters to the macros, as follows.

```
CH  Chapter
SE  Section heading
SS  Sub-section heading
    Text text text text text text text text text text
    Text text text text text text text text text text text
    Text text text text text text text text text text text

SE  Section heading
    Text text
    Text text text text text text text text text text text
    Text text text text text text text text text text text
```

If the macros above are made into a file called ' MACRO'and the text into a file called ' TEXT' then the command

```
PRINT MACRO TEXT RETURN
```

will reproduce the text shown earlier in this chapter. You will notice, however, that it first prints out a page which is blank apart from the footer: Page 0. If you can work out why it does this you will have no difficulty in understanding macros.

The reason is that the chapter heading macro causes each chapter to be printed on a new page by doing a page eject. The setting-up sequence sets the page number register P to 0, and the chapter heading macro then ejects page 0, and P is incremented to 1. This gives the correct starting page and chapter (chapter 1 page 1), but also gives a blank 0.

11 Change . . . replace . . . search

Another of those boring and time consuming tasks which seem to attract errors is going through a document to change a word every time it occurs. For example, you may already have typed a letter which sets out an agreement with a Mr Q R Smith, and you now want to send an identical letter to make a similar agreement with a Mr H I Jones. Or maybe you are typing a text about computers in which you have used the spelling 'disk' - only to find that your editor absolutely insists on the spelling 'disc'.

With traditional typing, making a change like this must count as the most boring of all possible jobs, but with word processing it could not be simpler. In the last example all you would do would be to switch to command mode and type:

```
CHANGE disk disc RETURN
```

and the word would be changed automatically throughout the whole text. Try a few examples for yourself.

Notice how the `CHANGE` command copies the case of the word to be changed. So with the command above, for example, 'DISK' would be changed to 'DISC' and 'Disk' into 'Disc'.

Naturally you can use this not only to correct spelling but to change the terminology itself. For example, if you are writing about antiques and mention 'grandfather clocks', you may find your editor insisting that 'long-case clocks' is the correct horological term. No problem:

```
CHANGE grandfather long-case RETURN
```

and again 'Grandfather' will become 'Long-case' too.

The way in which `VIEW` copies the case of the original can be very useful, but it can also prevent you from altering the case if you want to. For example you may be producing a report about an antibiotic called 'erythromycin'. You have nearly finished when you find that this is a proprietary name, and should be written 'Erythromycin'.

To make a correction like this you need to switch off 'VIEW' facility to copy the case of the original. This facility is known as 'folding' and you can switch it off by using the command FOLD (in command mode) with the number 0 after it:

```
FOLD 0 RETURN
```

After this you can issue the instruction:

```
CHANGE erythromycin Erythromycin RETURN
```

and the change will be made. To turn folding on again, replace the 0 with a 1 - and if you forget whether folding is on or off, just type

```
FOLD RETURN
```

and VIEW will tell you.

11.1 REPLACE

The REPLACE command is used when you want to change some of the words only, and want to make a decision on each one. Suppose you refer to a report in your text, giving the author as 'Green'. You subsequently discover that the author was 'Stephenson', so naturally you use the CHANGE command. To your dismay you later discover that you have also changed 'the green screen' to 'the stephenson screen'.

In cases like this you should use the REPLACE command instead of CHANGE. It allows you to make a separate decision on each occurrence of the word. The command is given in much the same way as CHANGE:

```
REPLACE Green Stepehnson RETURN
```

The system switches to text mode and the first occurrence of 'Green' is signalled by the cursor being placed on the first letter. If you do not want that word replaced, press N. If you want it replaced press Y. VIEW then moves on to the next occurrence of the word, until you have made your decision on them all.

Folding works with REPLACE in the same way as it does with CHANGE.

11.2 SEARCH

If you want to find a particular word but do not want to change it, or cannot

predict how you will change it until you see it, the most useful command is `SEARCH`. The command is given in the same way as the commands to change text:

```
SEARCH Green RETURN
```

The system changes to text mode and the cursor rests on the first occurrence of the word named. To find the next press `NEXT MATCH` (**CTRL f1**).

So what happens when you search for a word which you know is there, but the `SEARCH` command cannot find it? Perhaps you have mistyped it? - in which case the `SEARCH` command will never find it. Here again `VIEW` comes to the rescue with the ' wild search' facility.

All you have to do is to operate the search while substituting a '?' for each character which may be wrong. So if you want to find ' Erythromycinbut are not sure how it is spelt, you will find it quite quickly if you ask for ' Erythro????' .

With more common words it will of course find other words of similar spelling, but you can quickly move on using `NEXT MATCH`.

Even **RETURN** and **TAB** can be specified in your searches, using the vertical bar and the tilde (~) respectively. Consult the *Guide* for more detail on this.

11.3 Limited searching and changing

You can cause the commands `SEARCH`, `CHANGE` and `REPLACE` to act on a limited area of the text if you set markers 1 and 2 before and after the text concerned and then name these markers in the command:

```
CHANGE disk disc 1 2 RETURN
```

This will change ' disk' to ' disc' between the markers only.

11.4 Finding and changing phrases

The space between the words of a `CHANGE` command has a definite function: it signals to `VIEW` that you want the word before the space changed into the word after it. To replace one group of words with another, therefore, requires a slightly different method.

The method is to enclose the phrases within slashes, like this:

CHANGE/Read Only Memory/Random Access Memory/ **RETURN**

The space or slash in `CHANGE` and similar commands is known as a 'delimiter' since its function is to show where one thing ends and another begins. The first character after `CHANGE` indicates to `VIEW` which delimiter is being used, and it must then be used consistently through the command.

12 Formatting and margins

12.1 Protecting text

Formatting is done in blocks, which frequently correspond to paragraphs of text. Once VIEW starts to format a block it goes on until it comes to any one of the following:

- A blank line.
- A line beginning with a space.
- A line beginning with a TAB.
- A line with a stored command on it.

These 'stops' can be used deliberately to limit the effect of formatting and so prevent VIEW from formatting tables and lists which you have designed in a particular form and wish to keep that way.

Another method of prevent formatting which is particularly valuable for tables is to set a ruler above the table with the right margin stop removed. For example:

```
PE 12
..
.....*.....*.....*.....*.....
NAME          Week 1    Week 2    Week 3    Week 4
DAVIES, M     56        9         25        43
EVANS, G      20        35        13        26
HUGHES, T     18        32         6         45
JENKINS, P    32        34        53         9
JONES, D      16        45        23         8
REES, H       12        22        54        32
THOMAS, J     37        22        19        32
WILLIAMS, K   23        25        18        28
```

This could be a standard layout for tables in a report. The stored command PE 12 makes sure that there is room for the table on the page, and the absence of the right margin on the ruler protects the table against disturbance by formatting.

A stored command which is particularly convenient for protecting such items as addresses is LJ (left justify):

12.2 The use of TABs

TABs should be used in much the same way as on a typewriter, ie they should be the first character(s) in a line and have the effect of placing the text to the right. It is of course possible to use TABs *within* a line of text, but if you reformat such text with a different line length this is bound to cause confusion.

The length of a TAB character depends on the setting of the TAB stops in the current ruler. As the text moves between lines during reformatting, the alignments originally created by the TABs must become distorted and the text thrown into disorder.

When VIEW is in justify mode it attempts to cope with this problem by replacing the TABs with spaces before justifying the text, but in general it is best to avoid TABs within the text altogether.

12.3 The left margin

We mentioned earlier that the default left margin stop is the left edge of the default ruler, but that you can set your own left margin stop (>) anywhere on the ruler you like, provided that it is to the left of the right margin stop.

Setting a left margin stop in this way creates a left margin, in which it is possible to place side headings, as described in chapter 5. This is done by releasing margins, typing in the text, and restoring margins again.

In fact it is possible to operate within the left margin with much greater freedom than might at first appear. Most commands including TAB, INSERT CHARACTER, DELETE CHARACTER and HIGHLIGHTs function normally in this area. The use of TABs allows you to build up a series of headings and references like this.

```
.. .....>.....*.....*.....*.....*.....*.....*.....<
  1.      The first paragraph is typed as usual, with the
          paragraph number placed to the left with the
          margins released.
```

.. ..*.....>.....*.....*.....*.....*.....*.....<

a) The second paragraph number is aligned by replacing the left margin stop with a TAB stop. Again, margins are released to type in the number.

.. ..*.....*.....>.....*.....*.....*.....*.....*.....<

(i) And so the process continues. One advantage of doing things this way is that the text is fully protected when formatting, ie the paragraphs are formatted but the layout is preserved.

In text such as this, the left margin is in fact a left margin TAB character, which you can prove by deleting it and so causing the text to move left. What happens in effect is that when you type the first character on a line, VIEW checks for a left margin stop. If you have not set one, VIEW places the cursor in the first character position on the line. If you have, VIEW inserts a left margin TAB character filling the space between the leftmost position on the line and the left margin stop.

The length of a heading in the left margin is obviously limited by the margin width; in fact VIEW limits it to one character less than this. If you type a longer heading than this the text between the margin stops will move right, preserving one space between heading and text.

If you type a piece of text with a left margin set and then remove the left margin, the text will align to the left, but with one space between the text and the left edge. This is because the left margin TAB is still there, but reduced to one character space now that the margin stop has been removed.

When you reformat the text, the space to the left disappears. Similar if you define a left margin for a piece of text which had none before, you will need to reformat the text. In fact you should make it a rule always to reformat the text after adjusting the margin stops.

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.

14 Continuous processing

Until now, we have been thinking in terms of composing text, putting it onto disc, and retrieving it one file at a time, using `SAVE` and `LOAD`, or perhaps `READ` if we want to add material to a file we are already editing.

This is quite satisfactory for short items, but suppose you are processing a long report and want to go through it from beginning to end, correcting and editing. You will find yourself constantly interrupting your editing to `LOAD` and `SAVE` files.

A convenient way round this problem is offered by the `EDIT` facility, and when you have become accustomed to using `VIEW` you may find yourself using this as much as the separate `SAVE` and `LOAD` commands.

What happens in principle is that the computer reads in material from one file, holds it in memory while you edit it, and saves it in another file, bringing in more material from the first file automatically. The process continues until you have edited all the material and passed it on to the second file.

Obviously if you are going to use the `EDIT` system, you have to make sure that you have plenty of memory left on your disc before you start, or you will get a `Can't Extend` message and you will be left with a computer full of edited material and nowhere to put it. With care, however, it can be an effective way of processing files which contain much more material than will fit into the computer's memory.

14.1 The `EDIT` method

If you have material in the computer's memory which you wish to preserve, save it now. We are about to compact the disc so as to use disc memory more efficiently, but compacting normally destroys any text in the computer's memory.

Check the files on your disc. Type `*.` **RETURN** or `*CAT` **RETURN**. Are there any files you want to delete? If so type `*DELETE (filename)` **RETURN**.

Compact your disc. Compacting rearranges the files in the most efficient manner.
Type

*COMPACT **RETURN**

These first three instructions are intended to leave your disc with its programs arranged so as to leave the maximum space available for the EDIT process. Remember that your editing may result in a much larger file than you started with.

You are now ready to start the EDIT procedure, by reading in your file and giving a name to the file to which the text you are processing will go. Type

EDIT (file-in) (file-out) **RETURN**

(' File-in' is the original file, ' file-out' the destination file.)

VIEW reads in text from ' file-in'. You process it. When you are ready to go on to the next batch type

MORE **RETURN**

The text you have processed is written into your ' file-out' and new text is read in from your ' file-in' .

When you wish to stop editing, even if you have not finished the document, type

FINISH **RETURN**

and the text in memory is put into your ' file-out'. Any unread text left in ' file-in' is read and transferred to ' file-out' .

You are now left with two files: the original file (' file-in') and the destination file (' file-out'). If you no longer need ' file-in' type

*DELETE (file-in) **RETURN**

The command QUIT is an alternative to FINISH, which simply abandons editing, leaving ' file-in' intact but ' file-out' incomplete. You should delete ' file-out' .

You should be particularly careful about naming your files if you are to use EDIT. It is easy enough to get confused with normal saving and loading, but using two files at the same time can be tricky if you are careless about naming them.

14.2 Finishing

Note the procedure for finishing in the instructions. The point is that you can end your editing session in three ways. You can keep on asking for `MORE` until you finish the job. You can type

`FINISH` **`RETURN`**

if you have done a good deal of the job and want to preserve your text in a half-edited condition, to continue editing it later. `FINISH` leaves you with all the edited text in ' file-out' along with the rest of the text which you have not yet edited. In your next session you have only to keep on asking for `MORE` until you come to where you left off.

If on the other hand you have been interrupted when you have just started to edit, and what you have done so far is not worth saving, you do not want to wait while the computer goes through all the process of transferring unedited text from ' file-into ' file-out' as it would have to do if you used `FINISH`. Instead you can use the command `QUIT`, which simply abandons the operation, so that you can start again in your next session with the same ' file-in' .

One final refinement in continuous processing. When you are in the middle of editing, you may wish to transfer part of the text you are working on to ' file-out' in order to get some more out of ' file-in' perhaps to compare or transfer parts of it.

You can do this by setting marker 1 at the point up to which text should be transferred to ' file-out' Then type `MORE 1` **`RETURN`**. The first part of the text will then go into ' file-out' , and more material will arrive from ' file-in' .

15 Conclusion

We have now gone through the commands available on the VIEW word processor, and discussed some of the things you can do with them.

Once you have processed a few documents, even short ones, you will find yourself discovering new ways of doing your work. For example, there may be many pieces of text which you find yourself typing time and time again as parts of longer documents. Why not record these as files? Then when you need them, you can just READ them into the text you are composing and move them to wherever you want them at the time. The methods suggested in chapter 10 for macros can be developed a good deal further too.

15.1 The Second Processor

As you work with VIEW you will probably come to use larger and larger files. The EDIT facility can be useful here, but if you need more text memory within the computer you should consider the Second Processor module. This provides 30K of memory in any screen mode, so if you use MODE 3 you will have about three times as much memory for text.

15.2 Mode switches

If you normally want your computer to start up in a particular mode - say MODE 3 when using VIEW - you may find it convenient to have mode switches fitted. These are installed below the keyboard and should be fitted by your dealer.

15.3 *FX commands

Finally you should not neglect your *BBC Microcomputer System User Guide*. We mention in the *VIEW Guide* the *TV commands for preventing screen flicker and adjusting the height of the screen image, and the *FX commands for managing printing, but there are many more. For example you will have noticed that if you hold a key down the character auto-repeats on the screen after a short delay. Both the delay and the auto-repeat rate can be reset as follows:

To change the delay: *FX11, X **RETURN**
where X is the delay in hundredths of a second.

To change the auto-repeat rate: *FX12, X **RETURN**
where X is the number of characters per second.

To reset both rates to the default values: *FX12, 0 **RETURN**

15.4 Colours

The *BBC Microcomputer System User Guide* also goes into a good detail of detail concerning colour, most of which is of no relevance to word processing. If you use a colour monitor or a television set, however, you may wish to take advantage of alternative screen colours.

Both text and background colours can be changed. The colours available are:

- 0 - black
- 1 - red
- 2 - green
- 3 - yellow
- 4 - blue
- 5 - magenta
- 6 - cyan
- 7 - white

Try the following examples in MODE 3.

For a green screen effect:

Hold down **CTRL** and press S
Press: 1 2 0 0 0

For white text on blue background:

Hold down **CTRL** and press S
Press: 0 4 0 0 0

For black text on white background

Hold down **CTRL** and press S
Press: 0 7 0 0 0
Hold down **CTRL** and press S
Press 1 0 0 0 0

The first digit indicates whether it is text or background colour that is to be changed. 0 means background and 1 means text. The second digit indicates the colour, as in the table above.

Changing modes resets colours to white on black.

The purpose of this book is to help you become aware of the opportunities which VIEW offers, but using them effectively is a matter of practice. As you work with VIEW, place the *VIEW Guide* beside your computer for quick reference. It is fully indexed and groups the facilities so as to minimise page turning. Above all remember that word processing is a partnership between you and the system, but that the system depends on you for judgement and control. The more you understand about VIEW and about your computer the more effective you will be in using both.

16 List of commands

Immediate commands (text mode)

FORMAT BLOCK	f0
TOP OF TEXT	f1
BOTTOM OF TEXT	f2
DELETE END OF LINE	f3
BEGINNING OF LINE	f4
END OF LINE	f5
INSERT LINE	f6
DELETE LINE	f7
INSERT CHARACTER	f8
DELETE CHARACTER	f9

MOVE BLOCK	SHIFT	f0
SWAP CASE	SHIFT	f1
RELEASE MARGINS	SHIFT	f2
DELETE UP TO CHARACTER	SHIFT	f3
HIGHLIGHT 1	SHIFT	f4
HIGHLIGHT 2	SHIFT	f5
GO TO MARKER	SHIFT	f6
SET MARKER	SHIFT	f7
EDIT COMMAND	SHIFT	f8
DELETE COMMAND	SHIFT	f9

DELETE BLOCK	CTRL	f0
NEXT MATCH	CTRL	f1
FORMAT MODE	CTRL	f2
JUSTIFY MODE	CTRL	f3
INSERT MODE	CTRL	f4
DEFAULT RULER	CTRL	f7
SPLIT LINE	CTRL	f6

CONCATENATE LINES

MARK AS RULER

Move left one character

Move right one character

Move up on line

Move down one line

Move to beginning of last word

Move to beginning of next word

Move screen one screen size up

Move screen one screen size down

Copy block

Copy current ruler

Switch between text and command modes

Delete last character typed

Insert a TAB and move to next stop

CTRL **f7**
CTRL **f8**

right cursor

left cursor

up cursor

down cursor

SHIFT cursor left

SHIFT cursor right

SHIFT cursor up

SHIFT cursor down

COPY

SHIFT **COPY**

ESCAPE

DELETE

TAB

Stored commands (text mode)

TM Top margin

PL Page length

HE Headers on/off

HM Header margin

DH Define header

CE Centre text

RJ Right justify

LJ Left justify

DM Define macro

EM End macro

PE Page eject

OP Eject - odd page

EP Eject - even page

LM Left margin

LS Line spacing

CO Comment

HT Highlight

FO Footers on/off

DF Define footer

FM Footer margin

BM Bottom margin

TS Two-sided on/off

Command mode commands (command mode)

SEARCH	S	Search for strings*
CHANGE	C	Change strings*
REPLACE	R	Conditionally change strings*
FOLD		Match case of string in CHANGEing
SCREEN		Print text on screen
PRINT	P	Print document
SHEETS		Print document, pausing after each page

READ		Read in file and insert at marker
WRITE		Write out file between two markers

LOAD	L	Load in whole file
SAVE		Save whole file

EDIT		Start editing a file longer than memory
MORE	M	Save text and read in more*
FINISH	F	Finish editing current file
QUIT		Abandon editing current file

MODE		Change screen mode
------	--	--------------------

FIELD		Set field character
FORMAT		Format text*

PRINTER		Load new printer driver
MICROSPACE		Enable microspacing
COUNT		Count words*
CLEAR		Clear markers 1 and 2
WILD		Set wild characters for SEARCH
NEW		Clear text from memory
OLD		Revert to old text after BREAK

* These commands can be used with markers.

Index

- ACCESS 21
- Arrow keys 7, 8
- Auto-repeat 62,63
- BM 37
- BASIC 3
- *BASIC 4, 57
- Baud rate 25
- BBC Microcomputer 3
- BEGINNING OF LINE 9
- Bleep! 15
- Block? 23, 29
- Bold type 28
- Book layout 35-38
- Bottom margin 36
- BOTTOM OF TEXT 9
- BREAK 8
- Bytes free 11
- CE 35, 38
- Can't extend 59
- CAPS LOCK 6
- Case (upper and lower) 31, 49, 50
- Cassette, printing from 29
- Cassette systems 21, 22
- *CAT 21, 59
- Centre line 35
- CHANGE 49
- Chapter headings 46
- Characters 11
- CLEAR 32
- Clearing text 22, 57
- Colours 63
- Command mode 6
- *COMPACT 60
- Continuous processing 59-61
- COPY 13, 32
- COUNT 56
- Cursor 7
- DF 38
- DH 37, 38
- DM 40
- DEFAULT RULER 12
- Define footer 38
- Define header 37, 38
- Define macro 40
- DELETE 16
- *DELETE 20, 59,60
- DELETE BLOCK 27
- DELETE CHARACTER 9, 12, 16, 31, 54
- DELETE LINE 9
- Delimiter 52
- *DISC 19
- Disc memory 19
- EM 40
- EDIT 59, 60
- EDIT COMMAND 34
- Editing BASIC programs 57
- Editing (filename) 6, 20
- End macro 40
- END OF LINE 9
- ESCAPE 6
- *EXEC 58
- FM 37
- FO 38
- Filenames 23
- Files 19-23
- Finding words 50, 51
- FINISH 60, 61
- FOLD 50
- Footer margin 36
- Footers 36-38
- FORMAT 56
- FORMAT BLOCK 9, 10, 18, 33
- FORMAT MODE 13

Formatting 9, 10-13, 18, 53-55
 *FX commands 25, 62, 63
 Global formatting 56, 57
 Green screen 63
 HE 38
 HM 37
 HT 29
 Header margin 36
 Headers 36-38
 *HELP DFS 21
 Highlights 28, 29, 54
 INSERT CHARACTER 9, 12, 14, 31, 54
 INSERT LINE 9
 INSERT MODE 31
 Justification 10
 JUSTIFY MODE 10
 LJ 38, 46, 53
 LS 35
 Layout of pages 35-38, 45
 Left justify 38, 46, 53
 Left margin 54, 55
 Letters (macros) 44, 45
 Line spacing 35
 LIST 57
 LOAD 20, 57, 59
 Locking files 21
 Macros 39-48
 Mail shots 43-45
 Margins 53-55
 MARK AS RULER 13
 Markers 32, 51, 56, 57
 MICROSPACE 25
 MODE 12
 Mode switches 62
 Modes 6, 11
 MORE 60, 61
 MOVE BLOCK 31
 Moving text 30
 NEW 6, 22, 57
 NEXT MATCH 51
 No text 6
 Number registers 38, 46-48
 *OPT 29
 PE 34
 PL 34
 Page 7, 14
 Page eject 34
 Page layout 35-38, 45
 Page length 34
 Page numbers 38
 Parameters 42-48
 PRINT 24-27
 PRINTER 24
 Printer default 6
 Printer drivers 6, 24
 Printing 3, 24-29
 Printing highlights 28, 29
 Printing macros 39-48
 Programs (BASIC) 3, 4, 57
 Protecting text 53, 54
 QUIT 60, 61
 RJ 35, 38
 READ 20, 22, 57, 58, 59
 RECORD THEN RETURN 22
 Recording text 19-23
 Registers (number) 38, 46-48
 RELEASE MARGINS 17
 REPLACE 50
 Replacing characters 8
 Right justify 35, 38
 Ruler 11-14
 SR 38, 47
 SAVE 19-23, 59
 SCREEN 27, 28
 Screen 7, 14
 Screen modes 6, 11
 SEARCH 50, 51
 Second Processor 62
 SET MARKER 32
 Set register 38, 47
 SHEETS 26, 27

SHIFT COPY 13

Side headings 17, 18

*SPOOL 57, 58

Stored commands 34-38

SWAP CASE 31

TM 37

TAB characters 16-18, 54

TAB stops 7, 16-18

Tables 18

*TAPE 22

Text mode 6, 7

Text outside the ruler 17, 18

Top margin 36

TOP OF TEXT 9

Underlined type 28

VIEW ROM 3, 5

Wild search 51

*WORD 5, 57

Word count 56

Word processing 1

Words (finding and changing) 49-52

WRITE 29

The VIEW Word processor was designed and developed by protechnic Computers Limited, Cambridge, England

Into VIEW was written by Information Transfer Limited, Cambridge, England
