

HiSoft ED80

Full-Screen Programmer's Editor

System Requirements:

CP/M 2 or CP/M 3 disk system with at least 36K TPA.

Copyright © HiSoft 1988

ED80 Manual, Second Edition January 1988

Set using an Apple Macintosh™, Microsoft Word™ and Apple Laserwriter™.

All Rights Reserved Worldwide. No part of this publication may be reproduced or transmitted in any form or by any means, including photocopying and recording, without the written permission of the copyright holder. Such written permission must also be obtained before any part of this publication is stored in a retrieval system of any nature.

It is an infringement of the copyright pertaining to **ED80** and its associated documentation to copy, by any means whatsoever, any part of **ED80** for any reason other than for the purposes of making a security back-up copy of the object code.

Contents

SECTION 1 Getting Started & Tutorial **1**

1.1 Introduction	1
1.2 Using ED80	1
1.3 The Look of the Editor	2
1.3.1 The Status Lines	2
1.3.2 The Text Window	2
1.3.3 Typing Modes	2
1.4 A Quick Tutorial	3

SECTION 2 Installing the Editor **7**

2.1 Starting up the Install Program	7
2.2 Terminal Installation	8
2.3 Redefining the Editor Commands	11
2.4 Re-Defining User Options	13
2.5 Use of Installation Files	13
2.6 Leaving the Installation Program	14

SECTION 3 Command Reference Guide **15**

3.1 Cursor-Moving Commands	15
3.2 Text Deleting Commands	16
3.3 Block Commands	17
Printing a Block	18
3.4 Quick Cursor Movement	18
3.5 Find and Substitute	18

3.6 Leaving the Editor	19
Printing the File	20
3.7 Toggles	20
3.8 Miscellaneous	21

SECTION 4 Prompts and Messages **23**

4.1 Prompts	23
4.2 Rules for Filenames	24
4.3 Error Messages	25

SECTION 5 Technical Details **27**

5.1 Internal File Format	27
5.2 Non-Printing Characters	28
5.3 Data and External Devices	28

SECTION 1

Getting Started & Tutorial

1.1 Introduction

On your **HiSoft** program disc is a full-screen editor called ED80.COM.

ED80 is simply an editor which can be used to produce text for whatever application you need, you can even write letters with it.

ED80 is fully configurable, you can tailor the commands to suit your taste and you can also install the editor to suit your computer's particular screen codes.

The supplied disc contains the following files connected with the editor package:

- | | | |
|----|--------------|-------------------------------------|
| 1) | ED80.COM | (The editor) |
| 2) | ED80.HLP | (Help file for ED80) |
| 3) | JABBER.WOK | (Example text file) |
| 4) | EXTRA.WOK | (Example text file) |
| 5) | ED80INST.COM | (The installation program for ED80) |
| 6) | ED80INST.MSG | (Needed by ED80INST) |

1.2 Using ED80

You enter ED80 from CP/M by typing:

```
ED80 [RETURN]
```

You may, however, add a filename preceded by a space e.g.

```
ED80 MYFILE.TXT [RETURN]
```

This will cause that filename to become the *Current File*. If the Current File exists then it is loaded from the disc otherwise you simply start from scratch with an empty file.

When you leave ED80, you are returned to CP/M.

1.3 The Look of the Editor

1.3.1 The Status Lines

The upper status-line (the top line) has six sections and appears like this:-

```
JABBER.WOK      LINE:1 COL:1      INSERT
```

On the left is the name of the Current File. To the right is the line and column at which the cursor is positioned. The space to the right of the column number is for the commands you give to the editor. To the right of this is the mode (see next section) and finally there is a large space for messages.

The lower status-line (the bottom line) has three sections and looks like this:-

```
FREE:XXXXX      $                $
```

On the left is the amount of free space you have left in memory which will vary depending on the size of the current text. The value is approximately equal to the number of characters you can type before the editor's buffer becomes full of text. The two \$ signs mark the start of the Find and Substitute strings which are currently undefined.

1.3.2 The Text Window

The screen is best looked upon as a window onto the current text. This window may be moved in all four directions so that you can view any part of the file. If the window moves downwards then the text appears to move upwards and this is called upward scrolling. In the same way, if the window moves to the right then the text appears to move to the left (leftward scrolling). The scrolling is handled automatically and intelligently to give you the most convenient window onto the text.

1.3.3 Typing Modes

There are two basic typing modes: INSERT and CHANGE. INSERT mode is the more normal method of text entry. When you type a character in INSERT mode, all of the line to the right of the cursor is moved right one position before the character is entered. This means that the current line becomes longer by one character. In CHANGE mode, the character you type overwrites the current character (i.e. the one at the cursor) and thus the line remains the same length. You are not allowed to go over the end of the line in CHANGE mode.

In general, INSERT mode is used to build up a file and CHANGE mode is used to alter small sections within a line.

1.4 A Quick Tutorial

If you are quite familiar with the use of word-processors then a quick glance over this tutorial to clarify the points of difference with the one you are used to should be sufficient. If, however, you are in any doubt as to the full capabilities of a word-processor then it is strongly recommended that you work through this tutorial on your computer. Using a disc containing ED80.COM, ED80.HLP, JABBER.WOK and EXTRA.WOK, enter CP/M and type:

```
ED80 JABBER.WOK [RETURN]
```

On the screen at the moment is the text of *Jabberwocky* from Lewis Carroll's *Alice Through the Looking-glass*. This is just a sample piece of text to enable you to become familiar with the editor. As a re-assuring start, verify that the help key ([CTRL]-J) will list the various one-key commands. After a quick look at the help page, pressing [RETURN] will get you back to the file.

An important factor in the efficient use of full-screen editors is gaining confidence in moving the cursor around the file. It is worthwhile trying out the two main cursor-moving commands which are:-

character left ([CTRL]-S) and character right ([CTRL]-D)

If you keep pressing the character-right key until the cursor is just past the Y of JABBERWOCKY in the title and then press it a further time, you'll notice that the cursor wraps around to the start of the next line. Also notice how the line and column numbers change in the upper status-line.

If character spaces were the only way you could move the cursor then it would take a very long time to get to the end of some files and so, of course, there are many other ways to position the cursor. You can move the cursor one word to the left ([CTRL]-A) or one to the right ([CTRL]-F) or even straight to the beginning ([CTRL]-Q S) or end ([CTRL]-Q D) of the line.

Now move the cursor down ([CTRL]-X) and place it at line 7 column 1 just below All mimsy etc. As you may have realised, there is a line missing from the poem and the fourth line of the first stanza should read:

```
And the mome raths outgrabe
```

Type this line in preceded by the correct number of spaces and press [RETURN] at the end. You may have noticed that the second line of this verse is also incorrect and should read:

```
Did gyre and gimble in the wabe;
```

Move the cursor up to where the word `the` has been missed out i.e. the space after `in` and simply type the word `the`.

Note that the rest of the line is moved to the right after each letter. This is because we are in `INSERT` mode (see the upper status-line). You can change (toggle) the typing mode by pressing `[CTRL]-V`. Now the word `CHANGE` appears in the upper status-line. Move the cursor to the letter `o` of `wobe` which should be an `a` and simply type the letter `a`. Note that in `CHANGE` mode, the text is not moved to the right and the characters typed simply overwrite the existing ones.

If you finish off the line by typing `be;` then you can also see that you are not allowed to overwrite the end-of-line in `CHANGE` mode.

A word of explanation is in order here about end-of-line. At the end of each line is an invisible character (its value is 13) which can be cursoried-over, deleted, found and substituted as can any other character. In fact the installation program gives you the option of displaying all end-of-line characters so you can see where you are more easily. The most usual way of displaying them is with `<`.

The second verse has been omitted from the poem and this is a good excuse to test the deleting commands and the auto-indent feature. Firstly go into `INSERT` mode (`[CTRL]-V`) and then move the cursor to the start of the blank line below the end of the first verse. Now press `[RETURN]` twice to give a good separation between the stanzas. You can toggle auto-indent by pressing (`[CTRL]-OI`) and the message `I/AUTO` will appear on the upper status-line. Now type the first line preceded by the correct number of spaces so that the line starts directly under those of the first verse:

```
Beware the Jabberwock, my son!
```

If you make a mistake then you can delete the character before the cursor (`[DEL]`) or delete the character in front of the cursor (`[CTRL]-G`). Press `[RETURN]` at the end of the line and note that the new line starts immediately under the one above. This is due to the auto-indent and is an extremely useful feature when writing programs to improve legibility. Now type in the other three lines of the stanza ending all with `[RETURN]`:

```
The jaws that bite, the claws that catch!  
Beware the Jubjub bird, and shun  
The frumious Bandersnatch!
```

Typing mistakes are usually very common when entering programs and the editor has been designed to minimise the effects of the more common errors. Thus, whenever you delete a line, it is stored until you start to edit another line and can be recovered.

To illustrate this, move the cursor until it rests on the line starting: Beware the Jabberwock and give the command to delete the line ([CTRL]-Y). As you can see, the line disappears from the text, but the restore-line command ([CTRL]-O R) can be used to get the line back and in fact you may move the cursor to an entirely different place and restore the line again as many times as required. This manoeuvre can be quite useful for moving a line or copying it to some other place in the text.

If you edit a line i.e. type in a few characters to an existing line and then use the restore-line command, the line is restored to what it was originally.

In a large program using the Find and Substitute facilities is often the best way of getting to a known point in a program. To illustrate this, position the cursor on the first line of the poem, and give the find-first command ([CTRL]-QF). This puts the cursor just after the first dollar sign on the bottom status-line. Now type in Jabberwock. (you can use the delete-character-left ([DEL]) if you make a mistake) and then [RETURN]. Just press [RETURN] for the second or Substitute string.

The cursor will now be positioned on line 34. Note that the two previous occurrences of Jabberwock are not followed by a full-stop. If you now put the cursor to the top of the file ([CTRL]-Q R) and redefine the Find string as Jabberwock (without the full-stop) then the cursor will first rest on line 10. Now the find next command ([CTRL]-L) will go to line 23.

The next part of the tutorial illustrates the way you can manipulate blocks of text, rather than just characters and lines.

To define a block of text you have to mark its start and end. The original Lewis Carroll poem duplicates the first verse at the end and so the aim is to mark the whole of the first verse as a block and then copy it to the end. However, this will be done in rather an unusual way to illustrate the block buffer in the editor. Put the cursor to the start of line 4 i.e. the start of the first line of the first verse and mark this point as the start of a block ([CTRL]-K B).

Now position the cursor at the end of the last line of the poem (the space after outgrabe.) and mark this as the end of the block ([CTRL]-K K). Now the unusual part: delete this block ([CTRL]-K Y)!!

You can now see that there is a star (*) after the figure showing the amount of free space left on the bottom status-line. This star means that there is a block in the block buffer.

You can see the size of the block by using the free-space toggle ([CTRL]-OF). You get the block back by using the paste block command ([CTRL]-OP). *Do this now.* Note that the block is still there and can be pasted as many times as required.

Now move the cursor to the end of the file ([CTRL]-QC) and paste the block again and lo! the last verse is duplicated. Exactly the same effect would have been achieved by marking the block and then copying it ([CTRL]-KC) except that the block would not have been put in the buffer. Both block delete and block move (which is exactly equivalent to delete followed by paste) put the block in the buffer if there is space.

It is possible to write a block to the disc ([CTRL]-KW) and read a block from the disc. As an exercise, move the cursor right to the end of the file and then issue the command to read a block ([CTRL]-KR). You are now prompted for the filename of the file you wish to read in. Type:-

```
EXTRA.WOK [RETURN]
```

(you can use [DEL] if you make a mistake typing in the name).

Naturally Alice did not understand the explicit and obscure sexual connotations of the poem as we do today and the poem stands as an interesting if rather distressing insight into Dodgson's dark, tulgey mind!

Finally, it is obviously important to be able to save an edited file to the disc. There are three ways to quit. Firstly, you can abandon the file ([CTRL]-KQ). Here nothing is saved and the current text is lost.

Or you can save with no backup ([CTRL]-OQ). This will save the current text on the disc deleting a file of the same name if it existed. This method is generally used when space is low on the disc.

The most normal method of leaving the editor is to save with a backup ([CTRL]-KX). Here, if there is a file with the same name it is converted to type .BAK and thus is preserved.

Do the save-with-backup command ([CTRL]-KX). You are prompted for a filename, with the Current Filename already given for convenience. The Current Filename may be deleted if you require and the text saved under another name. In this case, however, just press [RETURN].

When silence and the A> prompt rules again, a look at the disc directory with DIR will show that the original file is now called JABBER.BAK and there is a new JABBER.WOK that is the file we have just edited.

It should now be perfectly possible (with frequent forays into the help-pages) to edit your own files. Before doing so it is advisable to cast a quick glance over the reference section as there are some very useful features documented there that have not been covered in this tutorial.

SECTION 2

Installing the Editor

The process of installing the editor involves three phases. The editor is first read in from the disc. Then, sections of the program are modified and finally it is written back out to the disc (as a .COM file) together with a help (.HLP) file. Thus the process involves a permanent change to the editor.

There are two reasons that you might want to run the installation program. Primarily, it may be that there are problems with the screen layout and the editor seems not to work at all. This will be due to incorrect terminal codes and in this case you should read the section on **Terminal Installation**. Alternatively, you may wish to modify some of the commands or options to suit either keyboard or taste. This procedure is covered in the section **Re-Defining the Editor Commands** and **Re-Defining User Options**. In either case you should first read the next section.

2.1 Starting up the Install Program

To run the installing program in order to install ED80, insert your back-up disc and type:-

```
ED80INST [RETURN]
```

You will now see the copyright message. The purpose of the installation process is to alter the copy of your editor on the disc. To this end, some copy of the program (called the working copy) is read in from the disc into the machine. The first question is thus:-

```
Normally the working copy of ED80 is  
read in from a file called ED80.COM
```

```
Use another file instead  
(Y/N) ?
```

To install ED80 you should answer N to this question; if you answer Y you will get the prompt:-

```
[ESC] to abort  
Omit file type (.COM assumed)  
Enter filename
```

to which a filename should be typed in (omitting the filetype). Whether you replied N to the opening question or Y and then specified a filename, the working copy will now be read in to the machine from the disc and the Installation Menu will appear.

There is now a copy of the editor in the memory of your machine ready to be altered and the Installation Menu on the screen.

ED80 INSTALLATION MENU

1. Return to CP/M
2. Alter screen codes
3. Save ED80 as <working copy filename> (normally ED80.COM)
4. Save ED80 as another file
5. Alter command codes
6. Alter user options
7. Load installation from .E80 file
8. Save installation to .E80 file

Type desired number:

If you are a first-timer using the installation program because the screen codes in the editor were wrong then turn first to the section **Terminal Installation** and then to **Leaving the Install Program**. The other sections in this chapter are **Re-Defining the Editor Commands**, **Re-Defining User Options** and **Use of Installation Files**.

2.2 Terminal Installation

Select option 2 from the main menu to alter the screen codes. You will be asked:

How many screen columns (80) ? and then

How many screen rows (24) ?

In answer to each of these questions you should type in the correct number followed by [RETURN]. Pressing [RETURN] alone is equivalent to giving the answer in brackets.

The rest of the questions concern how the screen controller works on your machine. If you are in doubt about any of the questions, consult the manual for your machine. You are now asked for the:-

Cursor position lead-in sequence

() () -

When the editor is in operation it has to be able to tell the screen controller to put the cursor at a certain position on the screen. To do this, it tells the controller the row and the column required. Most screen controllers require a special sequence of codes to indicate that the values to follow represent a row and a column. Thus inside the first set of brackets there will be the sequence as it is currently defined with the decimal values of the codes in that sequence in the second set of brackets. If the sequence is correctly set up then just press [RETURN] and move on to the next question. If the sequence is incorrect then it must be changed.

Assuming your screen controller does have a Cursor Position lead-in sequence (on Amstrad CP/M Plus computers it is [ESC] Y) then you should enter it now code by code (up to a maximum of four codes) terminated by [RETURN]. Each code may either be entered as a single keypress or as its decimal value terminated by [RETURN].

As an example, if the correct sequence for your controller was [CTRL]-K =. You could enter this either by typing

[CTRL]-K = [RETURN] or by typing

11 [RETURN] 61 [RETURN] [RETURN]

note the two [RETURN]s at the end. The first is to terminate the 61 and the second is to terminate the whole sequence.)

The next question asked is:

Is the row sent before the column ()
(Y/N/ENTER) ?

The screen controller may require the row before the column, or the column before the row. As above, pressing [RETURN] is equivalent to giving the answer in brackets.

You are now asked:

Offset for column () ? and then

Offset for row () ?

When the values for the row and the column are sent, many screen controllers require an offset to be added to each. The values required for the offsets are those required to position the cursor at the top left of the screen (i.e. if the correct offsets for your machine were both 32 then sending the Cursor Position lead-in sequence, then 32, then 32 will put the cursor at the top left of your screen). If the value in brackets is correct then just press [RETURN] otherwise type in the correct value terminated by [RETURN]. You should consult the manual for your machine if in any doubt.

The next text to appear is:

Clear Screen sequence
() () -

The layout is identical with that for the cursor positioning sequence detailed above. Press [RETURN] alone if the sequence for clearing the screen is correct or enter the correct code terminated by [RETURN] as above. If your controller does not recognize a sequence to clear the screen (possible but unlikely) then press D. Next:

Clear to End of Line Sequence
() () -

prompts you for the sequence to clear to the end of the current line. Respond to the prompt exactly as above for the clear-screen sequence. It is quite possible that your screen controller does not recognize a sequence for clearing to the end of the current line. If this is so then press D to delete the sequence and the editor will perform the function by software (although more slowly than the controller would do it).

Use lead-in ()
(Y/N/ENTER) ?

Use lead-out ()
(Y/N/ENTER) ?

These final questions concern the use of lead-in and lead-out sequences. These options allow you to send a command to the screen controller or run a small program at the start and end of an editing session. For example, this facility might be used to put your machine into 80 column mode for editing and reset back to 40 column mode on exit. However, unless you have an important reason for wanting to use this facility, it is advisable to answer N to both questions. If you answer Y to either you will be asked to specify a code sequence to send to the screen controller which you should enter as described above.

You will now be returned to the installation menu.

2.3 Redefining the Editor Commands

Pressing 5 from the main menu will allow you to alter the command definitions.

All of the commands will be shown and you have the opportunity to change the definition or accept it and pass on to the next command. After the last command you are returned to the main menu. For each of the commands the display format is:

Command name (keystroke definition | decimal definition)

where the keystroke definition is the sequence of keys you press to give the command and the decimal definition is the decimal ASCII value of those keys. These are alternatives.

At any stage you have the option to go back to consider the previous command, to retain the current definition or to change the current definition.

- 1) To backtrack to the previous command, press B
- 2) To retain the current definition press [RETURN]. The process then repeats for the next command. At the end you are returned to the main menu.

- 3) To change the current definition the new definition should be typed in element by element (up to a maximum of four elements) and terminated by [RETURN] after which the redefined command appears. If you are now sure that the definition is correct then press [RETURN] to pass to the next command, otherwise type in another definition and the whole process is repeated.
- 4) Definition elements are of two types. The first type is simply a keystroke and the second type is a sequence of digits terminated by [RETURN]. For example, the two ways to include a [CTRL]-Y (which has an ASCII value of 25) in the definition are:-

a) hold down the [CTRL] key and press Y

b) press 2 then 5 then [RETURN]

The two modes of entry of elements may not be mixed within the same definition. Thus if the first character of a definition was a number then all subsequent numbers are treated as their numerical value. However, if the first character was not a number then all subsequent numbers are treated as ASCII characters. This feature is included so that command definitions such as [CTRL]-K0 can be entered directly (i.e. hold down [CTRL] and press K, then press 0)

If the definition given is the same as that of a previous command or a prefix to a previous command then this message will appear:

```
WARNING : There is a conflict between the
          and                commands.
          Do you wish to continue anyway (Y/N) ?
```

A response of Y will ignore the duplication and N will allow the current command to be re-defined. Note that if the editor is saved to the disc with two commands identical, the use of one of the commands will be lost.

It is recommended that you consult the reference section of the manual if in any doubt as to the meaning of some of the commands.

After the last command, you are returned to the main menu.

2.4 Re-Defining User Options

You can change the user options by selecting 6 from the main menu. There are four user options. They are as follows:

Size of tabs () ?

to which you should type in the tab size required followed by [RETURN] or [RETURN] alone to retain the value in brackets.

Tabs per scroll () ?

When the cursor in the editor moves off the right-hand edge of the screen, the text window moves to the right (i.e. the text appears to scroll to the left). This left/right scroll works in units of one tab. On most screens, a value of one or two is best for this parameter. Enter the value as above.

End of line display () () ?

End of file display () () ?

A single key response is needed for both of these. If you don't wish the end of lines or the end of file to be displayed then press D to delete the current value, otherwise type the character you would like to be used for each (< is a common end-of-line marker with maybe | for end-of-file).

Although not normally used in word-processors, the markers can be useful in a program editor for distinguishing spaces and tabs at the end of lines and end of file.

After responding to these options you are returned to the main menu.

2.5 Use of Installation Files

There are many features of the editor that are alterable by the user.

Every copy of the editor naturally contains one set of these options. There is a type of file, however, called an *Installation File* that consists solely of the set of the alterable options. An Installation File is of type .E80.

To save the current installation information in a file, select option 7 from the main menu. You will then be prompted for a filename which you should type in terminated by [RETURN].

It is possible that you will see the error message

```
Too many characters in commands
```

If so, you should decrease the number of characters used to define the commands.

To load an installation file, select option 8 from the main menu. As above, you will be prompted for a filename. If the file you give does not exist then the prompt will be repeated. You can press [ESC] to quit.

When the installation file is loaded into memory, it will overwrite the alterable options already present in the copy of the editor in memory.

The main use of Installation Files is when you are in the long-term process of tailoring your version of the editor to suite your own preferences. If you save each successive change you make to the installation then any changes you find undesirable can be overwritten by using the last installation file rather than going all the way through the commands.

2.6 Leaving the Installation Program

You can leave the install program by selecting option 1 from the main menu, but BEWARE! If you select option 1 then nothing will be changed on the disc. Thus if you are satisfied with the changes you have made in the last installation session, you should first use either option 3 or option 4. Both will save a copy of the editor (as a .COM file) and a help file (as a .HLP file) on the disc.

Option 3 will save both files under the name you specified at the beginning of the session (normally ED80) whereas option 4 allows you to change the name by which you will invoke the editor.

You may have more than one copy of the editor on the disc at the same time (under different names, of course) without a clash of help files.

Thus the normal method of leaving the install program will be first to select option 3 and then option 1. If you don't wish to save the results of your installing labours then select option 1 alone.

You may, when saving, get the error message

```
Too many characters in commands
```

in which case you should decrease the size of your command definitions.

It is well worth spending some time deciding on the design of the command definitions. A well-designed and succinct set will be easier to use and will also lead to quicker and more efficient editing.

SECTION 3

Command Reference Guide

This Section is intended as a short reference guide to the commands and features of the editor. In all cases, the default command is given in brackets; where possible, the default command is the same as the Wordstar™ command. [CTRL]- means that the control key is to be held down, [RETURN] indicates that you should press [RETURN] or [ENTER] on your keyboard, and <CR> indicates a byte of the value #0D (ASCII 13).

3.1 Cursor-Moving Commands

Character left/right

[CTRL]-S : [CTRL]-D

Move the cursor one character position left/right. Moving past the end of a line positions the cursor at the beginning of the next line. Likewise moving past the beginning of a line puts the cursor at the end of the previous line. This feature is hereafter called *wraparound*).

Word left/right

[CTRL]-A : [CTRL]-F

Move the cursor to the beginning of the last/next word. Characters that constitute the boundaries between words are :-

" () [] { } = + - * / < > ^ - ; : , # \$ & \ [TAB]

and wraparound operates.

Tab left/right

[CTRL]-OS : [CTRL]-OD

Move the cursor to the last/next tab position. Wraparound operates.

Start/End of line

[CTRL]-QS : [CTRL]-QD

Move the cursor to the start/end of the current line. Wraparound does not operate.

Line up/down

[CTRL]-E : [CTRL]-X

Move the cursor up/down one line. After moving up or down one line, the cursor column is always the same. Thus it may appear that the cursor is positioned beyond the end of a line. If another line up/down or page up/down command is issued then the cursor will move as described. However, if any other key is pressed, the editor will behave as though the cursor was at the end of the current line (*ambiguous cursor*).

Top/Bottom of screen

[CTRL]-OE : [CTRL]-OX

Move the cursor to the top/bottom of the screen.

Page up/down

[CTRL]-R : [CTRL]-C

Move the text window down/up by one less than the number of non-status lines displayed on the screen. Thus a page up command on a 32-line screen will move the text window up by 29 lines (32 screen lines-2 status lines-1) and the old top line becomes the new bottom line. Ambiguous cursor operates.

Start/End of file

[CTRL]-QR : [CTRL]-QC

Move the cursor to the start/end of the file.

3.2 Text Deleting Commands

Delete line

[CTRL]-Y

Delete the current line. Note that the line is placed into the editing buffer and can be recalled into the text by use of the restore line command. The deleted line will be overwritten when the user next makes a change to any line.

Delete last character

[DEL]

Delete the character to the left of the cursor. Wraparound operates.

Delete this character

[CTRL]-G

Delete the character under the cursor. Wraparound operates.

Delete word left/right

[CTRL]-OT : [CTRL]-T

Delete from the cursor to the beginning of the last/next word. The characters that constitute the boundaries between are given under the Word left/right command above. Wraparound operates.

Delete to start of line

[CTRL]-Q[DEL]

Delete from the cursor to the beginning of the current line.

Delete to end of line

[CTRL]Q-Y

Delete from the cursor to the end of the current line.

3.3 Block Commands

Mark start/end of block

[CTRL]-KB : [CTRL]-KK

Place the block markers. A marker will be positioned at the cursor position. The markers are lost if the line containing the marker is altered subsequently.

Move block

[CTRL]-KV

Delete the currently marked block from the text and place in the block buffer, then insert the block at the cursor position. If there is enough space the block will be retained in the buffer, but the less space there is, the longer the command will take.

Copy block

[CTRL]-KC

Copy the currently marked block from the text to the cursor position.

Delete block

[CTRL]-KY

Delete the currently marked block from the text and place in the block buffer. The less space there is, the longer this command will take. This is due to the procedure required to place the block in the buffer rather than abandoning it altogether.

Thus, if the amount of free space is very small (less than 256) you are asked whether to abandon the block. If you press Y then the block will be deleted from the text and not placed in the block buffer. If you don't want to completely abandon the block then N should be pressed, the block should be written out to the disc (from where it may later be read back in if desired) and then deleted.

Paste block

[CTRL]-OP

Insert at the cursor the block currently in the block buffer. The block remains in the buffer if there is sufficient space.

Read block

[CTRL]-KR

You are asked for a filename. [RETURN] alone aborts the command. A filename followed by [RETURN] will search the disc for the filename given and insert it at the cursor. The response RDR: will read the block from the logical reader device.

Write block

[CTRL]-KW

You are asked for a filename. [RETURN] alone aborts the command. A filename followed by [RETURN] will write the currently marked block to the disc with the filename given.

Printing a Block

In response to the filename prompt, LST: will send the block to the current logical list device and may thus be used to print a block of text. The response PUN: will send the block to the current logical punch device. The whole file may thus be printed by setting the block markers to the start and end of the file and writing the block to LST: (but see **Printing the File** below).

3.4 Quick Cursor Movement

Goto line

[CTRL]-OG

You are prompted to enter a line number. This should be entered digit by digit (the DELETE CHAR LEFT command may be used as a destructive backspace) and after [RETURN] the cursor will be positioned at the start of the line given. This command is extremely convenient for quick access to an error reported by a compiler or assembler.

Goto start/end of block

[CTRL]-QB : [CTRL]-QK

Move the cursor to the start/end block marker.

Remember position

[CTRL]-KO

The current cursor position is stored. The marker is lost if the line in which it lies is subsequently changed.

Return to position

[CTRL]-QO

The cursor is positioned at the stored position.

3.5 Find and Substitute

Find first

[CTRL]-QF

The current Find string is displayed. [RETURN] will retain the current string, otherwise you should type in the required Find string (up to a maximum of 32 characters) and then press [RETURN].

[DEL] may be used as a destructive backspace, [CTRL]-R will redisplay the previous string and [CTRL]-U will abort the operation leaving the strings as they were.

A control code may be entered by pressing the control meta-key ([CTRL]-P) (see **Miscellaneous**) and then the control character (e.g. [CTRL]-P then [RETURN] enters a <CR> or [CTRL]-M into the string). Pressing the meta-key and then ? will return a value which is displayed as ? and counts as a wild-character when in the Find string.

After [RETURN] is pressed the operation is repeated for the Substitute string, and then the cursor is positioned at the start of the first occurrence of the Find string in the file.

Find next [CTRL]-L

The file is searched for the next occurrence of the Find string starting from one character after the cursor. A wild-character in the Find string will match with any character at all in the file.

Substitute and find [CTRL]-OL

The file is searched for the next occurrence of the Find string starting from the cursor. A wild-character in the Find string will match with any character at all in the file. When the string is found, it is replaced by the Substitute string and the cursor is positioned after the last character of the Substitute string. Finally the file is searched for the next occurrence of the Find string starting from the cursor.

Substitute all [CTRL]-OA

Starting from the cursor, all occurrences of the Find string in the file are replaced by the Substitute string. A wild-character in the Find string will match with any character at all in the file. The cursor is then placed after the last string substituted.

3.6 Leaving the Editor

Quit and Exit [CTRL]-KQ

You are asked whether to abandon the file. Pressing Y will cause a return to CP/M and the current text will be abandoned. Any other response will abort the command.

Exit without backup

[CTRL]-OQ

The Current Filename is displayed after the prompt `Filename:.` This may be deleted using the `DELETE CHAR LEFT` command ([DEL]) and altered. When you are satisfied with the filename, [RETURN] will cause the current text to be saved on the disc under the filename given. You are then returned to CP/M.

A file already on the disc with the same name will be lost.

Exit with a backup

[CTRL]-KX

Identical with above except that a file already on the disc with the same name as the Current Filename will be renamed as a `.BAK` file and any `.BAK` file with the same name will be lost.

Printing the File

With the Exit without backup command in response to the filename prompt, `LST:` will cause the current text to be written to the current logical list device and may thus be used to send a file to the printer. `PUN:` will write the text to the current logical punch device.

Note that after both these responses you will abandon the current text and the disc copy of the Current Filename will be unaltered. A better way of printing the whole file is to set the block markers at the start and end of the file and then write the block to `LST:` (see **Printing a Block** above).

3.7 Toggles

Toggle insert mode

[CTRL]-V

Switch between INSERT and CHANGE mode. A character typed in INSERT mode will only be entered after the characters to the right of the cursor on the same line have been moved right one character position. A character typed in CHANGE mode will overwrite the current character. A [CR] may not be overwritten in CHANGE mode.

Toggle auto indent

[CTRL]-OI

Auto indent will only operate in INSERT mode. The message `INSERT` will become `I/AUTO`. When indent is on and [RETURN] is pressed in INSERT mode, the next line will be indented so that it starts at the same column as the line above.

Toggle free space

[CTRL]-OF

A star (*) following the amount of free space indicates that there is a block in the block buffer. The free space toggle is used to check the size of the block.

3.8 Miscellaneous

Deliver tab

[CTRL]-I

Will return a tab character ([CTRL]-I or ASCII 9).

The size of tabs may be defined. Tabs will be entered as a ASCII 9 in the file and will not be changed to spaces. They are treated in the main like any other character in the file.

Restore line

[CTRL]-OR

If you are in the process of editing a line then this command will restore the line to what it was when you first positioned the cursor on it. If you are not in the process of editing a line then this command will insert in front of the current line, the last edited line.

This aspect of the command is useful because the DELETE LINE command places the deleted line into the line-buffer exactly as though it had just been edited. You may thus move a line from one place to another by deleting it, moving the cursor to the desired place and then issuing a RESTORE LINE command.

Disc directory

[CTRL]-KF

The prompt `Filename:` is given. See **Rules for Filenames**. A reply of [RETURN] or [SPACE] alone will abort the command.

After the filename is terminated by [RETURN] or [SPACE], the screen is cleared and a directory is printed (in fact the directory given will be the same as that seen after the equivalent DIR command). Any key will then return you to the current text.

Erase file

[CTRL]-KJ

The prompt `Filename:` is given. See **Rules for Filenames**. A reply of [RETURN] or [SPACE] alone will abort the command. After the filename is terminated by [RETURN] or [SPACE] the named file or files will be deleted from the disc.

Control meta-key

[CTRL]-P

Any key pressed after the meta-key will be entered into the file as its literal value. This may thus be used to enter control characters into the file that are normally commands or prefixes to commands (e.g. [CTRL]-P then 8 enters a [CTRL]-H).

The meta-key can also be used in the same way to enter control characters in the Find and Substitute strings. In this case if ? is pressed after the meta-key a character is returned that is displayed as ? but acts as a wild-character in the find string.

Help key

[CTRL]-J

Pressing the help key will display help pages giving information on the commands available from the editor and how to access them.

SECTION 4

Prompts and Messages

4.1 Prompts

There are four prompts produced by the editor which normally appear on the upper status line. They are as follows:

Abandon block: Sure?

This prompt requires a single character response. It appears if the user has issued the DELETE-BLOCK command and there are less than 256 bytes free. If you respond Y then the block will be deleted and lost (note that the block is normally saved in the block buffer and thus not lost) while any other response will abort the command.

The prompt also appears if you have issued any command the execution of which would overwrite the block in the block buffer. If you respond Y then the block in the block buffer will be lost and the command executed while any other response will abort the command.

Abandon file: Sure?

This prompt requires a single character response. It is produced after the QUIT command. Respond Y and the current text will be lost and you will be returned to CP/M. Any other response will abort the command.

Filename:

This prompt requires a string of characters terminated by [RETURN]. It is produced after any command that requires reading from or writing to the disc. The response is interpreted as a filename and the maximum allowed length is 14 characters (See **Rules for Filenames**). In building up the filename, the currently defined DELETE CHARACTER LEFT can be used as a destructive backspace. When [RETURN] is pressed, the Current Filename is displayed for convenience; it may be deleted if desired and another name substituted.

If the name returned is null i.e. [RETURN] alone then the command is aborted.

There are three responses to this prompt that are not interpreted as a filename, viz LST: PUN: RDR: . These address the logical list device, the logical punch device and the logical reader device respectively.

Go to line:

This prompt requires a string of numbers terminated by [RETURN]. It is produced after the GO TO LINE command. The response is interpreted as a line number and the maximum allowed length is 4 characters (only numbers are accepted). In building up the number the currently defined DELETE CHARACTER LEFT can be used as a destructive backspace. [RETURN] alone aborts the command.

4.2 Rules for Filenames

A filename consists of three fields. The drivename, the filename and the filetype e.g. B: MYFILE .GEN

When giving a filename:-

- 1) The drivename is optional and if not given the current logged-in drive is assumed.
- 2) In a command where ambiguous filenames are allowed (i.e. ERASE FILE and DISC DIRECTORY) a ? may be used to represent any single character and a * may be used as if the remainder of the field in which it occurs (barring the drivename field) were filled out with ?s.
- 3) In the ambiguous filename commands a response of the drivename field alone is interpreted as though the filename and filetype were *.

For example:

B:MYFILE.*

Addresses files of any filetype on disc B called MYFILE

FILE*.TXT

Addresses files of type .TXT on the current disc whose filename starts with the letters FILE

FILE?.TXT

Addresses files of type .TXT on the current disc whose filename contains 5 letters and starts with the letters FILE

B: or B:*.*

The first form (drivename alone) can be used only for the DISC DIRECTORY command. Addresses all files on drive B

4.3 Error Messages

There are twelve messages produced by the editor and they appear mainly on the upper status-line as do the prompts.

Out of memory

Indicates that there is not enough space in the machine to carry out the proposed command.

Line is too long

Produced when the length of the line would exceed the maximum allowed length (255 characters) if the proposed action was taken. This might either be simply the press of a key, or the deletion of a <CR>.

Undefined command

Indicates that the initial key of a command is correct, but the second or subsequent keys do not form a valid command.

Block start unmarked/Block end unmarked

Produced after any block operation if the start/end of the block has not been marked or the mark has been lost (i.e. the line containing the mark has been edited).

Block marks reversed

Produced after any block operation if the start of the block occurs after the end.

Invalid destination

Produced after a MOVE BLOCK or COPY BLOCK and indicates that the destination (cursor) lies between the start and end of the block.

Block too big

Produced after a READ BLOCK command and indicates that the file on the disc is too large to fit into memory.

No block in buffer

Produced after a PASTE BLOCK operation and is self-explanatory.

Marker lost

Produced after a RETURN TO POSITION command and indicates that the position marker has not been placed or has been lost.

No file/Bad filename

Produced after any command which prompts the user for a filename. The command indicates either that the filename is badly formed or inappropriate or the file does not exist.

Disc full

Produced after any command that tries to write to the disc. Indicates that either the disc or the disc directory is full. You should consider using the ERASE FILE command.

No such line

Produced after the GO TO LINE command and indicates that the line number given is greater than the number of lines in the file.

SECTION 5

Technical Details

5.1 Internal File Format

Text is held simply as a string of ASCII characters. The end-of-line sequence is <CR> (ASCII 13) rather than <CR><LF>, allowing the user greater text space. The end-of-file is marked by a <NULL> (ASCII 0).

When the text is written to the disc, however, <CR> is replaced with <CR><LF> and the <NULL> is replaced by [CTRL]-Z (ASCII 26) thus making the disc file written by the editor compatible with normal CP/M text files.

The maximum line-length is 255 characters. Note that the cursor column number may exceed 255 due to tab and control characters (in which case the column number displayed on the status-line is 255).

The maximum number of lines in the file is limited only by memory considerations, but note that the line number display on the status-line is only of four digits (due to space considerations) and the GO TO LINE command can only reach line 9999 and no further. All other commands will work as normal if the number of lines exceeds 9999 (although note also that on most systems the average number of characters per line would have to be about three for the line numbers to exceed 9999).

5.2 Non-Printing Characters

Characters of ASCII value less than 32 decimal (Control characters) are treated as any other character. They may be entered into the file by pressing the control meta-key ([CTRL]-P) and then the control character desired. If the terminal is capable of producing characters of value greater than 127 decimal, then these characters are entered as any other into the file and are displayed as ?.

The meta-key may also be used to specify control characters in the find and substitute strings in the same way as above. An obvious use for this feature is to find the end-of-line character (reached by [meta-key][RETURN] or [meta-key][CTRL]-M). The ? character when pressed after the meta-key returns #80. This character is displayed as ? in the find and substitute strings, but is treated as a wild-character in the find string i.e. it will match with any character at all in the file. (Note that if a terminal has a key that can return #80 then this will be in all respects identical to [meta-key] followed by ?).

5.3 Data and External Devices

Whenever you give a command that would normally access the disc (i.e. READ BLOCK, WRITE BLOCK, EXIT WITHOUT BACKUP) there are three responses to the prompt `Filename:` that are interpreted as logical external devices.

- 1) **LST:** if used for a write operation will send the data to the current logical list device which is normally a printer (but may, of course, be set from CP/M using STAT). When the data is sent a <LF> character (ASCII 10) is sent after every <CR> as is usual for CP/M files.
- 2) **PUN:** if used for a write operation will send the data to the current logical punch device. As above, every <CR> is sent as <CR><LF>, but unlike the use of LST: a [CTRL]-Z (ASCII 26) is sent after the data to mark the end-of-file. [CTRL]-Z is the standard CP/M end-of-file character.
- 3) **RDR:** when used for a read operation is designed to be compatible with PUN: or indeed any standard CP/M data transfer operation. The top bit of every character is reset (thus masking out any parity bits sent by the transmitting hardware) and all <LF> characters are ignored to produce the standard editor internal format. RDR: requires a [CTRL]-Z character to mark the end-of- data. Files may thus easily be transferred from one machine to another from inside the editor by use of PUN: and RDR:.