WORD PROCESSING TUTORIAL

USING FUNNELWEB AND TI-WRITER

by Col Christensen
Brisbane User Group

AIMS OF TUTORIAL

This tutorial on word processing on the 99/4A is aimed at the newcomer as well as the not so new user. Word Processing has so many facets that it would be impossible to assimilate all the knowledge the first time through. For that reason I suggest that whether you are a beginner or an expert, you should reread this series or the TI-Writer manual from time to time. There is always something new, maybe a tip or maybe a whole procedure, that you can make use of. I don't claim to be an expert but through research into the manual and by testing out "new" and previously unused features, I feel now qualified to present this series. I have attempted to make it as simple but precise as possible in an effort to smooth the way on the newcomers bumpy road.

WHAT DO I NEED?

TI-Writer, the word processor published by Texas Instruments in 1982 came in the form of a sizeable hard covered manual, a solid state cartridge, one disk housing the Editor and Formatter programs and practice examples and a keyboard overlay. As a newcomer to word processing you should have, apart from the actual software, two important items. One is the keyboard OVERLAY to show the relevant keypresses for the top row of keys, and the other is this tutorial as a guide. The TI-Writer Manual is also useful but a little harder to locate a particular topic in it. Obviously you'll already have an expanded system with disk drives and a printer connected to the system.

The word processor (WP) program you will be using will most likely not be the original TI one. Most everybody today uses the editor program supplied with and loaded through Tony McGovern's FUNNELWEB program. Tony has spent days and weeks and months on each upgrade to the original word processor program and there have been quite a number of upgraded versions. This software is FAIRWARE and users need to send off to Tony a reasonable payment if they haven't already done so as some recompense for all his efforts. Let's face it, you probably will use your computer for word processing more than anything else. With FUNNELWEB you not only get a top notch word procesor but a computer operating system as well. From here on I shall use the abbreviation WP for Tony's version of the word processor as the term TI-Writer is not now appropriate.

TEXT EDITOR

To start with, you need to know th two modes in which you use the WP, th Command mode and the Editor mode.

THE COMMAND MODE

When the WP first loads you see the cursor at the top left of the screen or what is screen line 2 with some command names shown on the line above. This is the COMMAND mode. On this line you type commands that allow you to do something with the document you have written. In the Command mode you have a selection of over a dozen one-letter or two-letter commands to process the text but more about these later. Pressing <E> for edit takes you from the Command mode to the Edit mode .

THE EDIT MODE

When in the EDIT mode, the top line of command prompts disappears and the whole 24 screen lines can be used for typing text. Its in this mode that you'll spend most of your WP time. Pressing FCTN/9 takes you from the Edit mode back to the Command mode . Now let's try that again on the keyboard. From the Command mode press <E> to go the the Edit mode and from the Edit mode press FCTN/9 to escape to the Command mode. In the Edit mode there are quite a number of keypress combinations that facilitate different operations but more later on these, too.

SCREEN WINDOWS

One small problem with most computers using TVs as monitors, and the 4 is no exception, is the limitation on the width of the screen when compared with normal paper width. Your printer under most conditions will be required to print 8 characters on each line. But the WP screen is only 40 characters wide so a system of windowing is used. We can have a line 8 characters long but we can't see all of it at once. Three screen windows are used and they show columns 0 to 39 (left half), 2 to 59 (middle half) and 40 to 79 (right half) respectively. The program starts up with the left window showing on the screen but further presses of the FCTN/5 key bring up the next window in the sequence left, middle, right, left, middle, right, etc.

WORD WRAP

Word wrap mode can be turned off or on using the CTRL/0 (that's a zero) key combination. By CTRL/0 I mean that you hold down the CTRL key while you tap the 0 key once. Word wrap is ON when the WP first begins and an indication of this mode is the appearance of the cursor, a SOLID rectangle. With word wrap ON, typing continues across the screen to the right tab position then automatically restarts on the next line at the left tab position. If at the end of a line only part of a word will fit, the whole word is automatically moved to the beginning of the next line. Another effect also is when typing continues to and past the right edge of the screen the screen flips to display the next window.

With word wrap OFF the cursor appears as a HOLLOW rectangle. In this mode typing continues across the screen to the right tab position only and will not drop to the start of the next line unless <ENTER> is pressed.

SETTING THE TABS

Tab settings control the way your document appears on the screen. As well as the standard tab position presets, you can also preset the Left margin tab, the Right margin tab and the paragraph Indentation tab. The Indent tab sets the position or column on the screen that a new paragraph begins at. The Right tab sets the screen column past which typing is not allowed and the Left tab defines the column where each line of text begins at near the left of the screen.

To set the tabs go to the Command mode and press <T> for Tabs. You will see above the cursor a numberline that starts at an invisible 0 and extends to the number 79 (that you would see if you window across to the right window). On this line certain letters are placed to indicate the tab settings required. "L" is for left margin, "R" is for right margin, "I" is for paragraph indenting and "T" is for tab.

Under the Tab line is the cursor on top of the letter "L". That seems a suitable place for the left margin so leave the "L" there and move the cursor over to position number 5. Type an "I" over the "T" here to indicate the position for paragraph indentation. Suppose, for example, in typing a document you don't want to have the WP flipping from one window to the next all the time. You can overcome this with a prudent setting of the right margin. Put an "R" for right at position 39. Then all typing will take place on the left screen window.

Tab position presets are fixed by using the letter "T" where required on the tab ruler. You can blank out any unwanted "T"s still showing between the "L" and the "R" settings. If (FCTN/7) is pressed when typing, the cursor will jump to the next "T" setting on that screen line. Use this facility when typing lists or tables in vertical columns. If two or more "L" settings are entered, the rightmost of them only is accepted and, in similar vein, of multiple "R" settings, the leftmost one is accepted.

Whenever you save a document to disk, the tab settings will be saved with it. Conversely, whenever you load a saved document from disk the tab settings will be retrieved with it and will be in effect when the loaded text appears on the screen.

After setting tabs, you must hit <ENTER> to confirm the settings and you will be returned to the Edit mode with the cursor at the point where it was just before escaping to the Command mode.

SCREEN COLOUR SELECTION

The default screen colours are white characters on a blue background but you can toggle through a selection of 10 colour combinations by successive presses of CTRL/3.

TRYING IT OUT

Now let's see what we can do with what we have seen so far. I hope you are making a list of the special keypresses mentioned to keep for future reference. Not all those mentioned will be found on the overlay.

Starting from first loading the WP, the cursor appears in the Command mode on screen line 2. First set the tabs as above or to your liking. Remember to press <T> to access the tab numberline. When done press <ENTER>. What mode are we now in? Yes, the Edit mode where we can type to our hearts content. But what's this? You've probably set a paragraph indent tab somewhere in from the left margin and the cursor is at the left margin setting on the screen.

Well, the computer isn't dumb. You haven't told it that you are to begin a new paragraph. So press <ENTER>, the signal in the Edit mode to end a paragraph, and there's the cursor at the correct indent position on the second line.

Notice the line numbers down the left side of the screen. Really the only time you need to see these is when copying, moving or deleting parts of your text. Besides, with the line numbers showing you won't see the whole 40 columns of text. Pressing FCTN/0 will toggle the line numbers ON or OFF. Repeat pressing FCTN/0 to see the effect.

Now is the time to type something, anything that comes into your head but first release the alpha lock key for lower case characters and use the shift key for capitals. Just keep typing on and on without worrying about mistakes and don't forget to throw in an occasional comma or full stop from time to time. Notice the "bell" sound as you get near to the right margin setting. Look to see how the word wrap functions. When you have exhausted the phase of the topic you are writing, press <ENTER> twice. The second press of the enter key serves the purpose of leaving one line spacing between paragraphs. Notice the funny character that comes on the screen where the enter key is pressed. This is a signal to a printer to do a carriage return and a line feed at that point. The WP won't allow you to type over a $^{c}_{r}$ symbol in word wrap mode, the symbol just gets pushed ahead of the cursor. However, the $^{c}_{r}$ can be deleted using the FCTN/1 delete char combination when necessary. Type another paragraph or so in similar vein expanding on your topic of discussion. Now it's time to correct any errors and make changes.

ERROR CORRECTION AND TEXT MANIPULATION

DELETECHAR (FCTN/1) To remove individual characters, move the cursor to the required point and press FCTN/1 for each character to be deleted or hold the keys down for repeated deletions.

DELETELINE (FCTN/3) The whole line from left margin to right margin will be deleted with FCTN/3.

INSERT CHAR (FCTN/2) Place the cursor at the appropriate position for an insertion and press FCTN/2. Notice that any text to the right of the cursor drops to the line below leaving space after the cursor. Type what has to be added, whether one letter, one word or a number of lines of text. If necessary you can insert again in a part that has already been inserted. The screen will probably look untidy with bits here and there on different lines. The text then needs to be reformatted as explained shortly.

In the non word wrap mode the insert char puts a character under the cursor and moves the remainder of the line to the right of the cursor one position to the right. Any character already at the right margin will be lost.

INSERT LINE (FCTN/8) Pressing FCTN/8 will move all text on the screen and in memory that is on the cursor line and below it down one line. It then blanks the line that the cursor is on.

REFORMATTING (CTRL/2) With word wrap on, move the cursor to a point from which you need to tidy up the text and press CTRL/2 and, Bingo! all the text just pops back into place where it should be. Reformatting takes place between the left and right margins as set on the tab numberline and continues down the file in memory until a $^{c}_{r}$ symbol is reached, i.e. till the end of the paragraph. Try altering the L and R tab settings and reformatting your document paragraph by paragraph. Place the cursor over the first character in the first paragraph and press FCTN/2 (insert char) then CTRL/2 (reformat). Press CTRL/4 to get to the start of the next paragraph and reformat it the same way. Play around with inserting, deleting and reformatting until you are familiar with their intricacies.

OOPS (CTRL/1) This aptly named function can undo deletions and insertions only if no other keys have been pressed since the deletion or insertion. It can only go back to the point before the wrong keypress or series of similar keypresses and take up from there. If, for example, you wanted to insert char (FCTN/2) but pressed the FCTN/3 (erase line) key by mistake. Too late, the line disappears. Not to worry. Press CTRL/1 before touching any other key. Presto! the missing line reappears. The main thing to remember is that if you make a booboo, press OOPS anyway. You never know, your error might be recovered.

CASE CONVERSION (CTRL/> and CTRL/:) If you are like me and watch the keys as you type, you'll often look up and find the

wrong sized characters on the screen, usually upper case because you forgot to reset the alpha lock key. Tony McGovern has added this little beauty to the word processor. Upper case will be changed to lower case as the cursor moves over them with CTRL/> held down. Isn't that terrific? I use it regularly. The opposite effect, lower to upper case conversion, is obtained by using the CTRL/: keys.

LINES

The L command will display the four lines options: Move, Copy, Delete and Show lines. To execute any one of them you don't have to go to Lines first, just go directly to it in the command mode by typing M, C, D or S then pressing <ENTER>.

COPY and MOVE

This is the cut and paste function used by our WP. The main difference between the two is that the COPY function leaves the original text intact and merely makes a copy somewhere else while the MOVE deletes the original section after making the copy. It is, unfortunately, restricted in its use to only whole lines of text. This is a slight drawback but the limitation can be overcome with a little extra effort.

To effect a COPY or a MOVE, you need to know the first line number and the last line number of the section to be copied or moved, and the line number after which the text is to be placed. Escape to the command mode and type C to copy or M to move and press <ENTER>. The prompt, "start line, stop line, after line" appears. Suppose you want to copy lines 19 to 24 inclusive and place them after line 16. You now type your line numbers in either of two ways, whichever you are more comfortable with. You can type 19,24,16 with commas to separate each or 19 24 16 using spaces instead, then press <ENTER>. After a brief delay depending on how much text has to be manipulated it will be done and your cursor will be ready waiting for you. Try out both the Copy and the Move and check the text to convince yourself that what you wanted to happen actually did.

Note that the WP won't allow you to copy or move text to a line number that is not currently existent. Suppose your text so far ends at line number 41. You can move text after line 41 but not after line 42 or higher. Note also, and this applies to most places where line numbers are involved, that line number 0 is valid and indicates the first line or before line 1 and line number E (for End) indicates the last line. So, in a MOVE, an entry such as 68,E,0 will move lines 68 to the end of your text and put them before line 1. Note thirdly that when you COPY some lines of text, you will end up with a larger number of lines than before, but if you MOVE lines, the final line count will be the same as before.

Now suppose you want to move, say, a long sentence and place it in a different position in your text. The sentence will be sure to start and end somewhere in the middle of a line. Murphy's Law makes sure of that. As the lines commands operate only on entire lines, we have a few extra steps to do. The idea is to use the insert keystroke to split both the start and ending lines so that the complete sentence and nothing else occupies a unique set of line numbers ready for moving. You also need to split the line into which the insertion must go.

I won't go into detail on the steps to follow to effect a MOVE but leave it for you to nut out. But just one little pointer though. When you use the insert keystroke to split lines of text and wish to move the cursor down the screen, do so with the arrow keys and not the <ENTER> which will leave a "¶" symbol that you most likely do not want. By all means use the <ENTER> key after the insert if you want that point eventually to be the end of a paragraph.

MORE ON REFORMATTING

Now that you are making insertions and moves within a paragraph and leaving a mess in the text buffer and on the screen you'll probably like to tidy things up by reformatting. What I want to impress is that it is not necessary to go way up to the beginning of a paragraph to do this. Just move the cursor to a point anywhere before the mess, press insert (FCTN/2) and press reformat (CTRL/2).

Although this section relates to tab settings, it has a bearing on reformatting too, for, as I explained before, reformatting takes place between the left and right tab settings. Tony McGovern has incorporated dual TAB sets into the later versions of our WP. You can now have one set of tabs for part of your document and different tab settings for another part. Both sets of tabs will be saved to disk with your document and retrieved again the

next time you load it into memory. To change the tabs, escape to the command mode and type ST (swap tabs). If the alternate set had not been set up, do so now, and press <ENTER> to accept the new set. The screen format and reformatting will follow the new tab settings. To revert to the other tab setting at some point in the text, escape to the command mode, type ST and press <ENTER> twice, one to accept the "ST" input and one to accept the tab settings. It is so simple to make the change over and is a very useful addition to the WP.

MARGIN RELEASE (CTRL/Y) The cursor movement is limited by the tab margin settings so that it can only move within the left and right margins. If you find a need, however, to move the cursor outside these settings, it can be done on the next keypress after pressing CTRL/Y. In other words you need to move the cursor to the margin you wish to cross, press CTRL/Y and then the appropriate arrow key.

DELETE LINE

Normally you would delete a line or two of text by using FCTN/3 but there are times when a large number of lines have to be deleted. This is done in the command mode after typing D and pressing <ENTER>. The prompt, "start line and stop line" tells what to do. Separate the relevant line numbers with either a comma or a space and press <ENTER> when you are sure you have typed the numbers correctly. OOPS won't help you recover from an error here. Tony has greatly improved the speed of the delete function in later versions.

SHOW LINE

The S command allows you to control which line numbers will appear at the top of the screen. Suppose the assignment you are writing is nearing 500 lines in length and you want to refer back to the first paragraph. SHOW will speed up the process of displaying it for you. Escape to command mode, type S and press <ENTER> and type a suitable line number. That part of the text will appear the instant you press the <ENTER> key. The line number, E for End, is valid and quickly shows the very last line of your work. Roll down (FCTN/4) and Roll up (FCTN/6), remember, also move the text up or down 24 lines at a time.

FILE HANDLING

PRINTING FILES

PF is the command to send the text in the memory buffer to a printer. You are prompted for a devicename for your printer. Valid devicenames normally used are PIO (Parallel Input/Output) if you have a parallel printer or RS232 if using the less common type of printer with a serial input. In this discussion I shall use the devicename, PIO, as hardly anyone today is using a serial printer.

So for printfile devicename, making the entry, PIO, followed by <ENTER> will print out the whole of the text in memory just as it would appear on the screen except for printer codes. The topic of printer codes will be covered in a later issue.

But the WP is more versatile than just being able to print out the complete text buffer. You can direct the printout to cover a specific part of the whole file by preceding the devicename with numbers or letters. The following are valid input for the devicename, PIO.

1 50 PIO prints only lines from 1 to 50.

48 E PIO prints lines 48 through to the end of the text. Remember the "E" represents the last line number.

L PIO prints the line numbers as well as the text but only the first 74 characters of text on each line. A line would normally be up to 80 characters long but the other 6 are taken up by the 4 digits of the line number and the two spaces following it.

L 33 46 PIO prints (can you guess?), yes, the lines from 33 to 46 including the line numbers. Amazing!

F PIO prints the text in fixed 80 format. One use for this function is to edit an assembly object code file and to save it back to disk by using a disk filename instead of a printer devicename. WP text files are normally saved as VARIABLE length records.

C PIO strips out any control characters like carriage returns, line feeds or new page before printing. I've never used this so I must try it out. Maybe you can think of a good use for it.

DELETING FILES

P purges all text in the memory buffers if you answer "Yes" to the prompt. You then have a clean slate again to start on.

DF for delete file appeared in the TI-Writer Editor so that particular disk files could be deleted. Its function has been greatly superceded and enhanced in Funnelweb's word processor to become a SD (show directory) command. You need to consult the Funnelweb docs to discover all its finer points of disk and file management.

SAVING AND LOADING FILES FROM DISK

Yes, SF is the abbreviated command name for this utility. A prompt then requires the devicename to save the file to. There I go again. I'm sure a preposition is not the correct thing to end a sentence with. There are additional refinements in this command like those above for printing files. You can save parts of your text by preceding the devicename with line numbers. Here are some valid entries:

```
DSK1.FILENAME
1 100 DSK1.FILENAME
26 E DSK1.FILENAME
```

To load a file, type LF on the command line and then at the prompt the pathname to the file on disk such as:

```
DSK1.FILENAME
16 120 DSK1.FILENAME loads in just those
   line numbers from the disk and stores
   them in the memory buffer numbered
   from line 1, of course.
0 40 DSK1.FILENAME
100 E DSK1.FILENAME
```

Things now get deeper. You can merge a disk file with what is already in memory with:

10 DSK1.FILENAME merges all of DSK1.FILENAME and places it after the existing line 10 and before the existing line 11. Obviously the old lines 11 onwards will now have much higher line numbers.

E DSK1.FILENAME merges all the disk file at the end of the existing file in memory.

0 DSK1.FILENAME merges all the disk file at the beginning of the existing file in memory.

And deeper yet! Merge PART of a disk file into the existing text with:

14 50 64 DSK1.FILENAME. It should be no trouble to work that one out if you remember that the first number is the line in the current memory buffer after which the insertion is to be made.

SEARCH

This group allows finding a particular "word" or replacing it with another.

FS (FindString) is followed by a prompt to input the string or word for which a search is to be made. The string must be preceded by a slash and followed by a slash. For example, to find the string "word processor" your entry would be /word processor/ . The search is case sensitive and will not find a string in upper case (WORD PROCESSOR) when the input is in lower case (/word processor/). After pressing <ENTER>, the text will be searched from the position of the cursor onwards. If you want the whole lot to be searched, do a Show Line 1 first to get the cursor to the beginning. The search will end if successful with the cursor over the first character in the first matching string found. If not successful, the cursor will appear after the end of the last line of text.

RS (ReplaceString) requires an input of the string to be searched for as well as the string to replace it with. An input such as /RSI/Repetitive Strain Injury/ will replace the abbreviation with the full name for the "affliction". But when the cursor appears over the first occurrence of the string "RSI" this list of prompts can be seen on the command line -

Yes, No, All or Stop?

So type
Y to replace this one and find the next.
N to ignore this one and go on to the next.
A to replace all occurrences of the string.
S to stop and escape to the edit mode.

NOTE WELL. If in the word-wrap mode with the solid cursor, all string replacements will be accompanied by automatic reformatting of that paragraph using the current tab settings. This can be disconcerting if a particular replacement

is in a section of your work that has been set out in tabular form. Reformatting will close it all up to just one space between each item. To overcome this, turn word wrap off before making the change in that section.

TEXT FORMATTER

The text formatter is a program loaded separately which allows the printing of a text file according to the formatting commands that are imbedded in the text file. These commands in the text file referred to as dot commands are the ones that set the current left and right margins, paragraph indentation, page length and line spacing etc. for printing .

When the formatter program is loaded, the first prompt asks for the INPUT FILENAME. This is the name under which the text file had been saved and will be the name of the file you want to print.

The second prompt of PRINT DEVICENAME will usually be answered as PIO.LF unless you have a serial printer. Then RS232.LF will be the devicename but consult your printer manual in case some special baud rate is needed in the serial devicename. In either case the formatter itself will issue line feed commands to the printer at the required places. To all the other prompts on the formatter screen, just press <ENTER> for now to accept the defaults shown and printing should begin.

THE FORMATTER COMMANDS

The formatter commands, are always in upper case and preceded by a dot (hence the name dot commands) and placed at the appropriate positions in the text on a line by themselves. More than one command can be placed on a line. There needs to be only one dot used only at the beginning of a line and each command must be separated by a semicolon. e.g. The very first line in a text file might have the following formatter commands:

.LM5;RM75;IN+5;FI;AD;PL56;CE2

It is also permissible to leave spaces between the command and the number parameter following it, such as .LM 5;RM 75 etc. The following explanations will throw some light on how these commands will influence the final printed output.

FORMATTER TEXT DIMENSION COMMANDS

LEFT MARGIN

.LM 6 sets the left margin at column 6 on the page.
.LM +5 adjusts the left margin inwards 5 columns more than the previous setting.
.LM -5 adjusts the left margin outwards 5 columns less than the previous setting.

RIGHT MARGIN

.RM 70 sets the right margin at column 70 on the printed page.
.RM +5 adjusts the right margin outwards 5 columns more than the previous setting.
.RM -5 adjusts the right margin inwards 5 columns less than the previous setting.

INDENT

.IN 8 indents the first line after a ^M (carriage return) to column 8.
.IN +10 sets the indentation to 10 columns inwards from the current LM setting.
.IN -5 sets indenting to 5 columns less than the LM setting.

LINE MANIPULATION

.FI (Fill) puts as many whole words as possible on each line to fill within the left and right margin limits.

.NF (No fill) cancels the FI command and prints the part of the document following the .NF exactly as it would appear on the screen.

.AD (Adjust) in conjunction with a Fill command spreads the spacing between words so that the printed text reaches the right margin exactly. Lines ending with a ^M will not be adjusted nor will they need to be. The formatter cannot differentiate between printer control codes placed within the text and ordinary text, so adjusted lines containing control codes may not completely reach the right margin.

.NA (No adjust) cancels the adjust command and the printout following it will have raggedy ends.

LINE SPACING

.LS 2 causes printing on every second line only.

PAGE LENGTH

.PL 60 prints 60 lines then starts a new page.
.PL +5 adjusts the page length relative to the previous setting
.PL -4 as above

BEGIN PAGE

.BP forces a new page break. The current PL value is then restored and countdown starts again.

INTERNAL FORMAT COMMANDS

CENTRE TEXT

.CE centres the next line between the current left and right margins.
.CE 3 centres the next 3 lines.

SPACE

.SP leaves one blank line on the printed page.
.SP 5 leaves 5 blank lines.

The above will cover the basic intricacies in using formatting commands to manipulate the way in which some text is printed. Remember two things:

1. The Tab settings govern the way the text will look on the screen and the dot commands in the text will control the way the text is printed.

2. On a line of formatter dot commands, an INdent with + or - relative positioning, relates to the last mentioned LM setting. On a line like the example earlier where many dot commands are placed, it is good practice to place an LM first if making changes to this margin.

I guess by now you have discovered what the given one-line formatting example above will do.

HIGHLIGHTING AND SPECIAL EFFECTS

Certain characters are used by the Text Formatter as flags to effect some print output features. The characters in question are the & for underlining, @ for overstriking and ^ for required space. (ampersand, at and circumflex).

REQUIRED SPACE with ^

The circumflex (^) tells the Text Formatter that words joined by it are to be treated as one word during underlining, overstriking, filling text or adjusting text. For example, you may want to be sure a group of words such as 24 January, 1992 will appear all on one line and not be split partly on one line and partly on the next. Just use the circumflex to replace each space like this, 24^January,^1992. The required space is useful also in underlining and overstriking groups of words as explained in the next two paragraphs.

UNDERLINING with &

Wherever the Text Formatter encounters the & symbol, the text from there to the next space character will be underlined. So if your text shows &IMPORTANT, the formatter will print IMPORTANT.

But if you wish to underline a group of words there are two ways to go about it. The first is to use the & symbol before each word. Typing &The &Storm &King will print to The Storm King.

The other way is to tie the words together with the required space. In &The^Storm^King, the whole title, spaces and all, will be underlined to become The Storm King as the printout. Although this is the preferred and easiest method, there can be a small price to pay in the FI;AD format. If the title is towards the end of a line and cannot be fitted as a unit on that line, it will be wrapped in full onto the next line. That would leave the previous line with a great number of spaces between words. You may not find the appearance of such a spacy line very elegant.

OVERSTRIKING with @

In a similar way to underlining in the paragraph above, the symbol @ is used to invoke overstriking. In doing this, the Text Formatter prints over the particular word four times before finally performing a line feed. The dark printing resulting stands out very clearly and is more prominent than even the printer's emphasized style of print. If a group of words is required to be overstruck, the required space symbol (^) can be utilised in a similar fashion to that shown in the paragraph on underlining.

Although the formatter removes the & and @ flags for underlining or

overstriking, you can still make it print one of these characters. Just type the character twice and the formatter will "understand".

TRANSLITERATES

This unusual word (Latin: litera=a letter and trans=across) simply means the use of one character to represent either another character or even a group of characters. That's like assigning a "," as a kind of variable to represent some long word or some printer code that might, say, change the print mode to enlarged italic elite print.

The formatter, when it encounters a circumflex (^) or an asterisk anywhere in the text or a period (.)placed first on a line will remove it and execute some routine that is flagged by that symbol. To be able to purposely print one of those three, you need to transliterate some other little used characters to represent any you want to print. e.g.

```
.TL 124:94<,     "|" becomes "^"
.TL 96:42<,      "`" becomes "*"
.TL 95:46<,      "_" becomes "."
```

The <, symbols above represent the carriage return character that appears on the screen when the <ENTER> key is pressed. The Formatter, on receiving the first character, will intercept it and send the second character to the printer instead.

```
.TL 123:70,85,78,78,69,76,87,69,66<,
```

In this one the character "{" is transliterated to a whole group of characters. Note the commas to separate each. So, in fact, wherever the formatter encounters a "{", it prints the word "FUNNELWEB" instead. Unfortunately, it does not format the rest of the printed line accordingly. In place of the one character, it prints 9 so there would be an additional 8 characters extending past the right margin with FI;AD in force.

```
.TL 125:27,88,1,28<,
```

The "}" character this time is interpreted to be the string of four characters following the colon in the TL. This string is a printer code for setting the left and right margins. It is probably shown in the printer manual as ESC "X" (n1) (n2). In the TL, the 27 is the ESC character, 88 is the "X", and the 1 and the 28 are the left and right margin settings for the printer. The printer, on receiving these four characters, will interpret them as a printer control code sequence and remove them from the input stream of characters. They won't be printed out,

but from here on, printing will only take place on lines that are 28 characters long. Becuase, in place of one character in the text (the "{"), none were printed, the print line will be one character short of the right margin.

INCLUDE FILE

This is another dot command that adds versatility to the Formatter. The command could look like .IF DSK1.RESUME2. When the Text Formatter encounters such a command, it prints the contents of that disk file too. The command can be at the beginning, the end or in the middle of the main file. There is no limit to the number of IFs used in the main file but IFs cannot be chained, i.e. a file that has itself been IFfed cannot have an IF command in it. Suppose you have done up a review that occupies 4 disk files named REVIEW1 to REVIEW4. Then to format them what you can do is to create another file to print out the whole review such as:

```
.CO Review:- The Storm King<,
.CO Date :- 23 March, 1991<,
.LM10;RM70;IN+8;FI;AD
.HE^^^^^^^^^^^^@THE^STORM^KING<,
.FO^^^^^^^^^^^^^^^^^Page^#<,
.IF DSK1.REVIEW1<,
.IF DSK1.REVIEW2<,
.IF DSK1.REVIEW3<,
.IF DSK1.REVIEW4<,
```

A few more dot commands have cropped up here.

COMMENT

The .CO flags a comment just for the benefit of the reader. The whole line is ignored by the Formatter.

HEADER

.HE is the dot command that produces a heading on each page printed. The header is placed on the first line on the page and the text begins immediately on the next line. In the above example, the

title, THE^STORM^KING would be printed at the top of each page. The eleven circumflexes forces the title to be spaced over 11 positions from the indent position and the ? ensures that the heading is overstruck four times. Reusing the command, .HE, without anything after it cancels any previous header command. If wording follows a new .HE, the new one will be printed.

FOOTER

.FO ensures that a footer is printed at the bottom of each page just after a blank line. The dot command can be followed by any text to be printed and/or the % character if required. The % instructs the Formatter to print consecutive page numbers. The example above will print both the word "Page" as well as its number. Likewise, page numbers can also be included in headers.

It seems that the F'web formatter prints out three lines less than the .PL command setting. On a setting of 60 lines per page, for example, the format printed is:

1 line either a header or blank
2 lines blank
52 lines of text
1 line blank
1 line either a footer or blank

Concerning page numbers in headers or footers, what if we want to start numbering the pages from something other than one or even skip a page number somewhere where we intend to use a page for illustration purposes? Well, there's another command to take care of that.

PAGE NUMBER SET AND RESET

.PA followed by a number sets header or footer pages to begin from that number. Relative values such as +2 or -1 can also be used to reset page numbers.

ALTERNATE CHARACTER SET

This important character set is useful for transliteration and for printer control. The normal set of characters range in ASCII values from 32 to 127 as shown in your Basic manual. The WP editor similarly limits the range of characters in the word wrap or non word wrap mode to this range. But there is another mode provided in the editor where you can type ASCII values from 0 to 31 as well. To toggle to and from this mode, you press CTRL/U. In this alternate input mode (but I will call it the CTRL/U mode from now on), the cursor appears as an underline character. What happens in CTRL/U mode, all key ASCII values are reduced by 64. So a ? (ASCII 64) shows up as ASCII 0, an A (65) becomes a 1 and so on.

With the ability to type ASCII values below 32 you can directly control your printer from within your text. e.g. the printer code for expanded or enlarged print is ESC W 1. The ESC is character 27, the W is character 87, and the 1, would you believe, is character 1. In your text you can type this printer code just before any heading you want to enlarge. First the ESC (27) requires the CTRL/U mode, then a [which is ASCII 91. The CTRL/U mode subtracts 64 and character 27 shows on the screen as a tiny dash-B. In hexadecimal arithmetic that represents 1B which is 16+11=27. To type the W you now get out of the CTRL/U mode and type W normally. The 1 requires the CTRL/U mode again and pressing SHIFT/A (not a) makes the little 1 appear. So you should see on the secreen a dash-B followed by the W then the tiny 1. Going over the keystrokes again, we have CTRL/U FCTN/R CTRL/U then W then CTRL/U SHIFT/A CTRL/U.

All the tiny characters are depicted in hexadecimal except for decimal 10, 12 and 13. These are special printer control characters that control the paper feed and print head position of the printer. They are called LineFeed, FormFeed and CarriageReturn respectively and show on the screen as LF, FF and CR. They can be typed in the CTRL/U mode by pressing SHIFT/J, L and M respectively.

MORE ON TRANSLITERATION

I should make passing mention here of a valuable disk that is probably in your Club program library. Once you have got the general hang of transliterates and the IFfing of files ask your program librarian for Jack Shugrue's disk called PLUS!. Print out the documents and try out the files on the disk.

Here is a sample file based on Jack's ideas that contains a lot of TLs for printer control through the Text Formatter. By using TLs most printer code sequences, no matter how long or involved, can be invoked by placing just a single character in the text. The file below has codes specifically for a Star printer and would suit most printers. The file should be saved as DSK1.*TL and stored on every disk

you use for word processing. The reason for the asterisk in the filename is to ensure that this filename appears near the top of a directory listing and will not appear among the filenames of normal text files. To make use of the printer codes, one of the first lines of any text file should be .IF DSK1.*TL.

When typing the file below, firstly just type the transliterate code and press <ENTER> to get the ^r symbols where they are shown. Then you can come back if you wish, in the non word wrap mode, to type the comments after each.

```
.TL 0:0^r          @ Reserved for 0=off
.TL 1:1^r          A Reserved for 1=On
.TL 2:27,72,32^r       B Dble strike off
.TL 3:32,15^r          C Condensed on
.TL 4:18,32^r          D Condensed off
.TL 5:32,27,69^r       E Emphasized on
.TL 6:27,70,32^r       F Emphasized off
.TL 7:32,27,71^r       G Dble strike on
.TL 8:8^r          H=Backspace reserved
.TL 9:32,27,52^r       I Italics on
.TL 10:10^r        J=Line feed reserved
.TL 11:27,53,32^r      K Italics off
.TL 12:12^r        L=Form feed reserved
.TL 13:13^r        M=Carr retn reserved
.TL 14:32,27,50^r      N 1/6 line spacing
.TL 15:32,27,48^r      O 1/8 line spacing
.TL 16:32,27,80^r      P Pica size print
.TL 17:32,27,51,17^r   Q sQuashed lines
.TL 18:32,27,83,0^r    R supeRscript
.TL 19:32,27,83,1^r    S Subscript
.TL 20:27,84,32^r      T Cancel sub/supe
.TL 21:32,27,45,1^r    U Underline on
.TL 22:27,45,0,32^r    V Underline off
.TL 23:32,27,87,1^r    W Wide enlarged on
.TL 24:27,87,0,32^r    X Wide enlarged off
.TL 25:32,27,120,1^r   Y NLQ characters
.TL 26:27,120,0,32^r   Z Draft chars
.TL 27:27^r        [ ESCAPE reserved
.TL 28:42^r            \ Asterisk
.TL 29:32,27,77^r      ] Elite print
.TL 30:94^r            ^ Circumflex
.TL 31:46^r            _ Period
```

Now that you are conversant with typing characters in the CTRL/U mode, we can look back at the transliterate file listed above. We now can send a particular printer code string through the formatter by typing just one control character that transliterates to that string. By choosing control characters from 0 to 31 for the transliterates, the full set of normal characters is left free for use in the text. Look at one of the TLs in the file:

```
.TL 5:32,27,69^r E Emphasized on
```

The printer code for turning emphasized print on is ESC E. i.e. characters 27 and 69 which you see in the TL above. 5 is the ASCII character to use to start Emphasized printing. To type the 5, first press CTRL/U to get the underline type cursor, then press SHIFT/E as shown in the comment after the transliterate above. The tiny character 5 appears. Press CTRL/U again to get back to normal cursor mode. Type E to complete the code followed by whatever word/s you want to appear emphasized when printed. Then cancel the emphasis (printer ESC F) by typing after the word/s the character, 6. That is, CTRL/U SHIFT/F and finally CTRL/U to return to normal cursor mode. Simpler when you're actually doing it rather than trying to grasp it mentally.

Referring again to the TL code above then, following the character 5 and the colon, are three character values assigned to the character 5. You've seen how the 27 and the 69 come from the printer ESC E code. That just leaves the 32 which is a space character. If the 5 were to be transliterated to the printer code for emphasized print style and encountered by the Formatter, the 5 would be removed from the text, acted upon in setting the printer code and the line filled to the right margin. But the Formatter fills to the right margin before removing the character 5, so that line will end up one character short of the right margin. The space character, therefore, is included to be printed to compenstate for that loss of one character.

If you intend to use the TL file above, you will need a reference to consult when using your word processor. The main thing to record is the list of SHIFT characters and what printer code they control. So start off by typing a list beginning with:

```
in CTRL/U mode
SHIFT   Effect
  B   Dble strike off
  C   Condensed on
  D   Condensed off
etc
```

Maybe you have been observant and noticed that, for all the times I've said in this series not to do this and not to do that, I have broken the rules I set. You've probably noticed some lines beginning with periods and several occurrences of asterisks and circumflexes etc.

Transliteration has been the key to overcoming most of those hurdles which become limitations no longer. My only difficulty has been deciding whether I need to use an actual transliteration at a certain point or just to show an example of one as an illustration. Anyway, the whole business of transliteration can be quite complicated if you like to go into it deeply as I found out when I set out to transliterate a tilde to download a properly formed ~ symbol to my printer. The best way to learn all the intricacies is firstly to have a need to use them and secondly to actually use the processes that produce the results you require.

MAKING A SETUP FILE

Most times that you start up your word processor you need to set up your favourite tabs, margins and indent positions and on the first few lines to prepare a set of print margins, transliterates, comments etc. Why not have a standard layout on a special disk file that will do all of the above for you each time you need it? Then, when you start, its a simple matter to do a LoadFile of that filename and simply carry on typing using your own default screen margins, tabs and indent positions as well as the formatter attributes as outlined on those first few lines loaded. A sample of such a file with filename *SETUP could be:-

```
.LM10;RM70;IN+5;FI;AD;PL60^M
.IF DSK1.*TL^M
.CO Here place name/purpose of file^M
.HE If required^M
.FO If required^M
```

Before you save this file to disk, you need to set the tabs also so that they, too, will automatically be saved with the file. So each WP disk you use would have in readiness two files, *SETUP and *TL, on them.

FORM LETTERS

Form letters are those sent to a number of persons and containing the same information but generally personalised for each recipient. Certain parts of the letter like name, address, salutation change from letter to letter. Such a letter can be created in the normal way on the editor but where the wording has to differ, a variable must be placed. The variable takes the form of *n* where n is a number from 1 to 99.

A value file of data to replace the variables must be created so that it can be called by the Text Formatter while printing. If the .FI command has been used, the data for each variable will be slotted smoothly into the text. The value file can take two forms. Probably the more common is the Mail List file on disk and the other is the list typed in through the keyboard during the course of printing in response to screen prompts. Which method you use depends on a number of factors.

Mainly, if you have need to send out form letters to the same people on more than one occasion, the mailing list is the more efficient and it is too, if you have many letters to send. With just a few variables in each of a small number of letters the list typed in while printing takes place seems the more attractive.

Here is a sample of a form letter complete with variables more suited to a disk based mailing list file. The ^M where it appears represents the carriage return symbol that appears on the screen when the <ENTER> key is pressed.

```
*1* *2* *3*,^M
*4*,^M
*5*.^M
^M
Dear *1* *3*,^M
The Weeders' Digest's bonanza prize
draw is to take place very soon and
there could be much rejoicing at *4* at
that time. All you need to do, *1* *3*,
is to complete the enclosed form and
return it together with the YES sticker
and your luck might change.^M
```

MAIL LIST AS A VALUE FILE

A mailing list is compiled with a text editor keeping in mind certain protocols when typing the list. The first character on a line must be a number (1-99) which corresponds to the *n* in the form letter. Next on the same line is a space, then the data for that variable and lastly a ^M symbol by pressing <ENTER>. Other lines of data follow using the same format until all data for one form letter is entered. On the next line type an asterisk and press <ENTER>. This indicates that that is all the data for that letter. Continue on similarly with data for all letters. Here is a sample mailing list saved to disk as DSK1.MLIST1.

```
1 Mr",
2 J.",
3 Brown",
4 15 Redwood Street",
5 Forestvale Qld 4076",
*",
1 Mrs",
2 E.",
3 Smith",
4 28 Brolga Avenue",
5 Gumdale Qld 4090",
*",
```

FORM LETTERS USING A MAILING LIST

When the text formatter is loaded, the list of screen prompts allows you to incorporate mailing lists in the form letters printed out. If using a mailing list, these prompts will appear:

ENTER INPUT FILENAME Answer the disk filename of the form letter.

ENTER PRINT DEVICENAME Answer PIO.LF or your usual RS232 name complete with the final .LF or you can send the output to a disk file without the .LF of course. If you happen to print out this disk file, however, you will need to add the .LF to the printer devicename then.

USE MAILING LIST? Answer "Y" this time.

WHICH LETTER(S)? (ALL) Answer "A" for all letters if using the full mailing list or specify by numbers which letters to print. A selection of letters could look like this: 1,5,9-17,19-20,23,25-E. Each individual number or group must be separated by commas and each group tied by a dash. The "E" can be used to signify the end of the mailing list.

PAUSE AT END OF PAGE? Answer as you normally do. "Y" if using single sheet manually fed paper, otherwise answer "N".

MAILING LIST NAME? There are two ways to inform the Text Formatter the location and filename of the data list. A mail list dot command such as .ML DSK1.MLIST1 towards the top of the text file would take care of that and there would be no need to give an answer to this prompt. Just press the <ENTER> key. If no dot command is included in the text file, it's obvious that you would have to input the filename here.

Printing then begins with a screen display of "PRINTING LETTER NUMBER n" tucked in amongst the other screen text. As printing continues the number of letters shown on the screen and yet to be done counts down.

FORM LETTERS USING ALTERNATE INPUT

If you answer "N" to the prompt, USE MAILING LIST?, and there are variables in the form letter the Text Formatter expects input of data from the keyboard. The screen inputs are slightly different too.

ENTER INPUT FILENAME Same as above.

ENTER PRINT DEVICENAME Same as before.

USE MAILING LIST? Answer "N".

WHAT PAGE(S)? (ALL) The input here is ignored anyway so just press <ENTER>.

NUMBER OF COPIES? This prompt takes the place of the previous one. Here enter the number of form letters you are going to do. During the course of printing this number on the screen will count down as each letter is printed.

PAUSE AT END OF PAGE? Same as above.

When the formatter encounters the first variable in the form letter file, another screen prompt appears saying, "ENTER DATA FOR VARIABLE *1*" and invites you to supply the data for it. Up to 28 characters (a whole screen line) are allowed for each data input. Once you press <ENTER>, printing will continue if there is sufficient text to send to the printer before the next variable crops up. So you must to be on hand to input all the data values as they arise for every letter to be printed. You need to know the kind of value to type for each variable, like a name for *1*, a street for *2* and a city for *3* etc.

This process of alternate keyboard input through non-descriptive prompts like ENTER DATA FOR VARIABLE *6* leaves a little to be desired. A define prompt dot command (.DP) has been provided to allow you to tailor your own wording for each prompt. Your own prompt would guide you as to the type of information to type in. To achieve

this a list of define prompts must be placed in the form letter prior to the first occurrence of a variable. The list might look like this:

```
.DP1 :NAME?",
.DP2 :STREET?",
.DP3 :CITY ETC?",
.DP4 :Dear",
```

Then as each variable is encountered, the prompt you defined for that variable number pops up to jog your memory on what kind of response to give.

You've seen how variables such as *4* are placed in form letters and treated in a particular way by the Text Formatter. The variable starts with an asterisk followed by a number from 1 to 99 and terminates with another asterisk. If this format is not adhered to, the formatter can do strange things to the text. That is why a printed asterisk must be transliterated from some other character non-sensitive to the formatter.

MORE ON PRINTERS

One aspect of word processing that needs more coverage, I feel, is the use of printer control code sequences to manipulate the various printer functions. These code sequences range in length from just one character to a great number. The most common sequences, however, are from one to three characters.

When the printer receives a valid single control character, the character is removed from the text string and the text string is acted upon according to the function that it controls. In a sequence, the first control character will usually be the character 27 and it will be followed by one or more other characters. The control character 27, usually referred to as ESCape, will be removed as well as a predetermined number of other characters. The number depends on what the second character is.

As a matter of interest, The control characters from 0 to 31 in value were named in the days of early systems of electronic text communication and became more evident in the days of teletype transmissions. Some of the mnemonics that generally depict the function of the characters are:

ASCII	MNE	Meaning
2	STX	Start of text.
3	ETX	End of text
6	ACK	Acknowledge
7	BEL	Ding a ling
8	BS	Back space
12	FF	Form feed
14	SO	Shift out
20	DC4	Device control 4
27	ESC	Escape

THE PRINTER CONTROL CODES

Printer control sequences can be grouped into the following categories:

1. Text mode settings
2. Print positioning
3. Page formatting
4. Graphic bit imaging
5. Downloading characters
6. Printer status

For each of these groups the printer functions and control code sequences are given as both mnemonics and their equivalent characters.

TEXT MODE SETTINGS

Function	Mnemonic	Characters
Expanded printing for one line	SO	14
Cancel expanded	DC4	20
Begin condensed mode	SI	15
Cancel condensed	DC2	18
Begin Pica size	DC2	18
Begin Elite size	ESC :	27 58
	or ESC M	27 77
Begin Italics	ESC 4	27 52
Upright print	ESC 5	27 53
Begin emphasized	ESC E	27 70
Cancel emphasized	ESC F	27 71
Begin underlining	ESC - 1	27 45 1
Stop underlining	ESC - 0	27 45 0
Begin dble strike or near letter quality	ESC G	27 72
Cancel above	ESC H	27 73
Begin subscript	ESC S 1	27 83 1
Begin superscript	ESC S 0	27 83 0
Cancel either	ESC T	27 84
Begin expanded print	ESC W 1	27 87 1
Cancel expanded	ESC W 0	27 87 0
Begin overlining	ESC _ 1	27 95 1
Cancel overlining	ESC _ 0	27 95 0

123456789012345678901234567890123456789012345
EXPANDED IN WIDTH
PICA STYLE IS 10 CHARACTERS PER INCH
ELITE SIZE IS 12 CHARACTERS PER INCH
CONDENSED SIZE IS 17 CHARACTERS PER INCH
EMPHASIZED IS DARKER AND BROADER
NEAR LETTER QUALITY IS TOPS
Overlining Underlining Both
ITALICS PICA and ELITE VARIETY
SUBSCRIPT and SUPERSCRIPT

Some of the different text modes can be combined as you have probably noticed in the examples above. For convenience, those that will combine are more easily seen if the codes are put into sub-groups. You can use any code from one of the sub-groups with one from any or each of the others but don't be disappointed if some discrepancies occur. Your printer manual might list some of the restrictions and which ones have priority over others.

1 Expanded
2 Pica, Elite, Condensed
3 Subscript, Superscript
4 Italics, Upright
5 Emphasized or not
6 Underlined
7 Overlined
8 Doublestrike (NLQ) or not

PRINT POSITIONING

Function	Mnemonic	Characters
Line feed	LF	10
Form feed	FF	12
Carriage return	CR	13
Set perforation skip	ESC N n	27 78 n
Cancel perf skip	ESC O	27 79
Set margins	ESC X m n	27 88 m n

The perforation skip refers to the space left unprinted at the perforation of continuous (fan fold) paper. It sets the bottom of the page margin to "n" lines. That is, the distance from the last print line on one page and the first on the next page.

The margins code allows setting of the printer's left margin (m) and the right margin (n). If you want to list a basic program in 28 columns just like it appears on the screen, set these two values to 1 and 28 or if you want to print further over on the page, set it to a pair of higher numbers.

PAGE FORMATTING

Function	Mnemonic	Characters
Set 1/8" line spacing	ESC 0	27 48
Set 7/72" spacing	ESC 1	27 49
Set 1/6" line spacing	ESC 2	27 50
Set n/216" spacing	ESC 3 n	27 51 n
Set page length lines	ESC C n	27 67 n
Set page length inches	ESC C 0 n	27 67 0 n

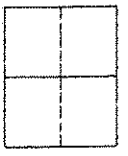
The standard line spacing as set on the printer dip switces is 1/6". Double spaced print can be set with the code ESC 3 H, the H with ASCII 72 making 72/216" or twice 1/6". 1/8" line spacing can be used for listings if you want to conserve paper.

GRAPHIC BIT IMAGING

These codes are used by such programs as TI Artist, My Art and Page Pro in their output of graphic designs to paper. The complications of their use would be beyond the scope of this artice. Briefly put, the picture data is sent out, row by row, in streams of bytes each representing 8 vertical dots of the picture. The 8 dots are converted to data numbers in much the same way as in redefining a character in a CALL CHAR statement in Basic.

Most printers already have a set of graphic characters restricted to horizontals and verticals that could be suitable for doing line drawing. Their ASCII values are usually greater than 128 so are not readily available in word processing. They can be printed however by using transliterates such as these that suit my printer:

```
.CO Use CTRL/U & SHIFT
.TL 1:218    A  TL CORNER
.TL 2:194    B  T INTERSECTION
.TL 3:191    C  TR CORNER
.TL 4:195    D  L INTERSECTION
.TL 5:197    E  CROSS
.TL 6:180    F  R INTERSECTION
.TL 7:192    G  BL CORNER
.TL 8:193    H  B INTERSECTION
.TL 9:217    I  BR CORNER
.TL 16:196   P  HORIZ LINE
.TL 17:179   Q  VERT LINE
```



```
APPPPBPPPPC
Q    Q    Q
Q    Q    Q
DPPPPEPPPPF
Q    Q    Q
Q    Q    Q
GPPPPHPPPPI
```

DOWNLOADING

Redefined characters can be down loaded to the printer using code from this group. Another code allows selection of the downloaded character set or the standard set for printing. This data transfer sends bytes each representing 8 vertical dots to be printed. The 8/9 pin type of printer would take about 13 bytes to define a charater. The more pins your printer has the more bytes that have to be sent to define a character. For my printer which has a character definition 48 dots deep and 36 wide in the high quality mode it takes 224 bytes to redefine just one character and 96 disk sectors to house a full set from ASCII 32 to 126.

Modern printers have a range of selectable fonts such as Courier, Sanserif, Orator etc that give a good variety in output. These can be selected either by using control codes or by push buttons on the printer.

PRINTER STATUS

Function	Mnemonic	Characters
Reset printer	ESC @	27 64
Unidirectional print	ESC U 1	27 85 1
Bidirectional print	ESC U 0	27 85 0

Resetting the printer cancels all previously set codes. Unidirectional print is necessary for printing graphics as variations in the printer's horizontal registration otherwise tend to get wavy shapes in vertical lines.

THE HEXADECIMAL NUMBER SYSTEM

Let's look at numbers up to 31 in decimal and hexadecimal, the latter usually being prefixed with a > or a H or suffixed with a H. You can apply the principles of decimal notation to hexadecimal. In decimal the two "houses" are tens and units while in hexadecimal they are sixteens and units. So >1B is equivalent to 1 sixteen + B (11 decimal) units totalling 27 decimal.

Decimal	Hexadecimal	Decimal	Hexadecimal
0	>0	16	>10
1	>1	17	>11
2	>2	18	>12
3	>3	19	>13
4	>4	20	>14
5	>5	21	>15
6	>6	22	>16
7	>7	23	>17
8	>8	24	>18
9	>9	25	>19
10	>A	26	>1A
11	>B	27	>1B
12	>C	28	>1C
13	>D	29	>1D
14	>E	30	>1E
15	>F	31	>1F

In the CTRL/U mode type the upper case alphabet from A to Z while watching each resultant character on the screen. Which keypress produced a tiny 1 on the screen? Of course, the "A", the 1st letter of the alphabet. And which key put a tiny 9 on the screen? Right again! The 9th letter of the alphabet. "I" is the ninth letter. But when we get to the Z that's only the 26th. How do we get the others up to 31? Look up an ASCII table to find the next characters. The one after Z is the 27th, a "[". What keys do you press to type "[". The magical FCTN/R of course gives us the ESC character, 27 as we already knew but didn't know why.

One other number is missing, the 0. It surely comes before 1 which is obtained with SHIFT/A. The ASCII table shows the "A" character is preceded by "@" so that's it. Play with them for a while if it's still a little hazy. Notice that you can type your own CR, LF and FF symbols on the screen.

USING THE CODES

You can use printer control codes in your text whether you intend to process it through the formatter or just print it straight out from the editor using the PF command. AS I may have said in an earlier article, it is preferable to use transliterates when using the formatter as these tend to keep the adjusted right margin more even.

Now comes the process of typing control code sequences into your text. You will have to refer to both the mnemonic and the characters listed for that particular function.

Firstly, we'll take the code to set the printer to subscript, i.e. ESC S 1 with the characters 27 83 1. As you know, we need the CTRL/U mode to type characters less than 32. Starting at ESC which is character 27 and should show on the screen as a small B with a little vertical dash before it, press CTRL/U and FCTN/R and the character will appear then CTRL/U again to

get back to a normal cursor. Now, beside that, we need an "S" typed in the normal way. Just press SHIFT/S. Next comes the character, 1. We need CTRL/U again for this one, and SHIFT/, and CTRL/U again.

Your code on the screen should read dash-B S tiny-1 to represent ESC S 1. Anything typed after that will print out in subscript style. You do not want all of your text to be in subscript, so you must cancel that style somewhere. Looking up the codes table, we find the code to effect that is ESC T. To get it press CTRL/U FCTN/R CTRL/U SHIFT/T. Got it? Now you should be able to handle the twenty-seven keystrokes and fifteen typed characters needed to be able to print the chemical formula for battery acid which is H_2SO_4.

If you can manage that exercise, you can class yourself capable of mastering control of your printer. Try to use other printer codes to bring your printer's hidden talents to the light of day. I know, you will have to refer to the code lists above or your printer manual from time to time. Who doesn't? What I find helpful is little lists stuck here and there over my console and, guess what, they ALL refer to printer codes. Without them as ready references, I might not bother to use printer codes. I would get by without making the printer do what I would like it to and never be really satisfied with the result.

I hope by now that some of the fog surrounding the use of printer codes is beginning to clear and a ray or two of golden sunshine is beginning to peep through.

RANDOM NOTES

* The text formatter defaults are NoFill, NoAdjust, LineSpacing=1 and PageLength=60.
* HEader, FOoter, DefinePrompt and TransLiterate must be on a line by themselves or be the last command on a line.
* The ADjust command must be preceded either on the same line or separately by a FIll command to function at all.
* .IN relative (+,-) is based on the preceding LM setting.
* Other relative value commands must be preceded by an absolute value for that same command name.
* Your printer must have the automatic perforation skip disabled by setting one of the dip switches accordingly. You are likely to tear your hair out trying to format pages correctly if you don't. Both the formatter and the printer would be issuing form feed commands resulting in blank pages and pages with just a few lines printed on them.
* Sending print through the RS232 card can be halted by pressing FCTN/4. The printer will stop when its print buffer is empty.
* Printer problems such as jammed paper can be stopped on most printers by pressing the ON LINE button. Then press FCTN/4 on the computer to cancel its output.
* When testing the formatter output, you can use a disk filename as the print devicename. You can then inspect the file to see the effect of the formatting.
* The Text Formatter prints 5 lines less than the .PL command setting probably to allow for a header and a footer. Check this out using a disk filename instead of a print devicename.
* You can outdent when creating lists of definitions or notes such as these. Set .LM inwards and .IN relatively less with a command such as .LM5;IN-4.
* FindString will also search text column numbers. e.g. 18 27/REBATE/ will search columns 18 to 27 only. Handy for finding a match in a chart or table.
* FindString will still search for a word even if no slashes surround the input.
* ReplaceString can take quite a while in a lengthy text file with word wrap on and reformatting taking place. Don't be tempted to to turn word wrap off, though, if the new string is longer than the one to be replaced. You could lose characters spilled off the right of the page.
* ReplaceString must have word wrap turned off if searching through any columnar text such as lists and tables. Make a habit of doing a SaveFile prior to any RS anyway just in case there is some part of your text that is marked for NoFill.
* Use TransLiterates to print asterisks and circumflexes.
* Never print out a listed Basic program through the formatter. The multiply sign (*) and power sign (^) will disappear and the string concatenation

(&) and the string variable (@) will produce confusing results.

That about concludes the tutorial on our current word processor. I must compliment Tony McGovern on the sterling work he has done to manipulate the old TI-Writer program into such a useful and powerful word processor. One thing I haven't mentioned is the ShowDirectory routine which Tony has expanded so that it is more like a file management program. Although it forms part of our WP, I feel that information on its use is best left to the maestro himself so look up his Funnelweb documents for all the gen.

TABLE OF KEYPRESSES

1. Shown on the keyboard overlay.

FCTN	CTRL
1 Delete Char	1 Oops!
2 Insert Char	2 Reformat
3 Delete Line	3 Screen colour
4 Roll Down	4 Next Paragraph
5 Next Window	5 Duplicate Line
6 Roll Up	6 Last Paragraph
7 Tab	7 Word Tab
8 Insert Line	8 New Paragraph
9 Command/Escape	9 New Page
0 Line Numbers	0 Word Wrap
= Quit/Escape	= No action

2. Duplicate keystrokes and additional new ones.

CTRL/T Back Tab	
CTRL/V Beginning of Line	
CTRL/C Command/Escape	= FCTN/9
CTRL/F Delete Char	= FCTN/1
CTRL/K Delete End of Line	
CTRL/N Delete Line	
CTRL/X Down Arrow	= FCTN/X
CTRL/L Home Cursor	
CTRL/O Insert Line	= FCTN/8
CTRL/G Insert Char	= FCTN/2
CTRL/H Last Paragraph	= CTRL/6
CTRL/S Left Arrow	= FCTN/S
CTRL/Y Margin Release	
CTRL/P New Page	= CTRL/9
CTRL/M New Paragraph	= CTRL/8
CTRL/J Next Paragraph	= CTRL/4
CTRL/Z Oops!	= CTRL/1
CTRL/R Reformat	= CTRL/2
CTRL/D Right Arrow	= FCTN/D
CTRL/A Roll Down	= FCTN/4
CTRL/B Roll Up	= FCTN/6
CTRL/I Tab	= FCTN/7
CTRL/E Up Arrow	= FCTN/E
CTRL/W Word Wrap	= CTRL/7

CONTENTS