JUST TEXT FORMATTER

INTRODUCTION

One of the most common uses of small computers is for word processing. Word processors generally perform two main functions. These are Editing of text, and Formatting of the edited text. Some of the word processors are combined editors and formatters. They allow editing of the text, and show the user immediately what the text will look like in its final form on the printed page. The text rearranges itself on the screen if any change is made in a format command. These processors require that the editor be what is called a "screen editor" as opposed to a "line editor". Line editors operate on one line at a time in a text file. A screen editor will allow the user to move the cursor around on the screen and edit at the cursor location.

Other word processing systems break the two functions into two programs. The editor is used to enter the text. Commands for the formatter are inserted to control the format of the final output. Of course the file as it is being edited does not necessarily resemble the final format. ***JUST***, however, can be used to format the text to a terminal rather than to the printer. The user then sees where the text breaks for pages and in fact, exactly how it will look when printed, so that he can cause early paging to keep a paragraph or a table intact. He can add extra blank lines to separate a paragraph from a table, insert blank lines for a drawing on the final printout, change the left margin and text width for insertion of photographs or insets, start page numbers at a number other than 1, etc.

JUST is a text formatter that will accept commands that are placed within the text by the user. It is actually the second half of a two-part word processing system, the first half being an editor. There are a number of good editors available, and most FLEX9 users have their favorite. Though ***JUST*** is a formatter that has been set up to work at its best with an Epson MX-80 with the Graftrax option, it may be used without that option, and it may be adapted to work with other printers. ***JUST*** has no provision for operating with printers that have "proportional spacing" of letters. It is intended primarily for dot matrix printers.

There are a few terms that need to be explained before getting into the detailed instructions. Someone new to text processing and text formatters might easily be confused by the terms "fill" and "justify". Many text processors are capable of both functions.

Fill, generally means to put as many words as will fit on each line, and nothing more. Filling by itself makes the right ends of lines uneven, but they are as even as the word lengths will allow. This paragraph is filled but not justified. This mode is frequently used for letters since people tend to think of justified text as "printed". Text that is only filled looks more "personal".

When lines that are first filled are also justified, extra spaces are inserted in the text between words in the line to make the right margin come out even. Some care must be used in the choice of the placement of extra spaces. A random distribution would be the most pleasing to the eye. A systematic distribution, such as always placing the extra spaces between words starting at the left of each line is most disturbing. The text appears to be "light" on the left and "heavy" or dark on the right. A good compromise is to distribute the spaces from the left on every other lines and from the right in the remaining lines. Since ***JUST*** maintains a line count for each page, it decides from which end to insert extra spaces based on whether the line count is odd or even.

The fact that the formatter will fill lines makes it possible to edit normal page width text on a terminal or small computer with limited width. The TRS Color Computer, for example, with FLEX, has a 51 character line length limit (ie. to fit a line on the screen). You can edit text with a maximum width less than 51 and set the width to 65 or 72. ***JUST*** will expand the lines to the proper length. It will center titles, though they are input at the left of the screen, and it will even move short lines (for example name and address) to the far right of a line. This greatly facilitates editing text on a narrow screen.

INSTRUCTIONS FOR FORMAT PROGRAM

JUST uses commands embedded in the text and a several built-in constants for page length and format. The built in constants may be changed by overlaying certain memory locations with the desired values. Some of these constants are used to set the initial values of parameters that can be changed by in-line commands. The initial value of these parameters are indicated below, and instructions for changing them are included later.

,W:nn W sets line width to the number nn. For Example ,W:72 The initial width (called the default value) is 66. That is, if a ,W instruction is not used the width will be 66.

,M:nn M sets left margin to the number nn. The default left margin is 8. The default width and margin will properly center text output on a 10 character per inch printer to an 8 1/2 by 11 sheet of paper, though the paper guides may have to be moved a bit to center the paper within the print area.

,P Starts a new paragraph which is automatically filled and may be right justified if the Justify mode is on.

,S:nn Skips nn lines. Default is 1 if :nn is omitted. May be used on the first blank line after a paragraph, to turn the fill mode off, otherwise a blank line may be used, and will be passed through the formatter without change.

,R:nn Right justifies the next nn lines. That is, the text is moved to the right end of the line so it ends at the right margin. Default for nn is 1.

,C:nn Centers the text on the next nn lines. Default is 1.

,I:n Single indent on the next line, n spaces. This value may be negative. The sum of the indent and the margin values must be greater than or equal to zero. The paragraph indent may be made zero, and the ,p preceded by a negative indent. This is useful for numbered items that may then be justified as in this list of commands which have (on the two lines preceding the text) the commands ,i:-10 and ,p the paragraph indent was set to zero above with a ,h:0 command.

,G:n Without the :n will cause output to page (go to the top of the next page). With the argument, this command sets the current page number.

,H:n Sets the number of spaces to indent at the start of each paragraph. Value must be positive but may be zero.

,V:n Sets the vertical spacing, eg ,V:2 for double spaced.

,J Turns on Justify mode. Text within paragraphs is always filled. When Justify is on, lines are also padded with spaces to achieve right justification. Justify is a MODE. It need only be turned on once at the start of a program where even right margins are desired.

,Q Cancels or turns off the Justify mode. ***JUST*** initializes itself to the Justify mode being OFF. If you want it on, you must include a ,J instruction at the beginning of your text. Of course you may turn it on or off anywhere within the text by the ,J or ,Q instruction at the beginning of an otherwise blank line.

,Y This sets the double wide mode for one line. ,Y is a smart double wide mode. It automatically adjusts the left margin and line length so that it can properly right justify and center titles done in double wide mode. This feature (for obvious reasons) will work best if left margin and line length are even numbers (they must be divided by two for double wide to come out even). After a line in the double wide mode, the original margin and line length are restored.

,U This is a modal underline command that will provide underline with any of the Epson printers, since they support backspace. It works by inserting a backspace and an underline character after every printable (non-space) character in a line. It cancels itself at the end of the line. This underline mode does not underline spaces.

The following are special commands for formatting control with an EPSON MX-80 or MX-100.

,E	**Sets the emphasized print mode.**
,D	**Sets the double strike mode.**
,B	**Sets both, for boldface titles.**
,N	Cancels all of the above.

Note that these commands must appear at the beginning of a line, and therefore the printing mode cannot be changed within a line by using them. They work with the MX-80 without the Graftrax option as well as with it.

NOTE

Text is only justified within a paragraph. The Fill mode is turned off by any other command OR A BLANK LINE. That is, termination of a paragraph by a RETURN, followed by another blank line (ie. a second return) OR a command such as ,S for example, will print the remainder of the previous paragraph from the line buffer. At that point, the Fill mode is off, and text will be printed exactly as input. A ,P command will restore the Fill mode and until the end of a paragraph, all text will be filled again. If the Justify mode is on, it will also be justified.

The text file should start with the ,W and ,M specification in no particular order, if the initial or default values are not satisfactory. Commands must be at the beginning of a line, and terminated by a Carriage Return. Erratic results will be encountered if there is any text or a space after a command. The formatter will accept either upper or lower case letters for all commands. A comma is used as the command delimiter because there is no normal situation in which a line would start with a comma.

Note that the Epson printer is set to the double wide mode with the control code $0E. It remains in that mode for one line. A CR terminates the double wide mode, so that no cancel instruction is necessary. Space in the table is provided for a cancel code, however, and ***JUST*** will output the cancel code when it outputs a line in the double wide mode. If your printer requires such a code you may install it by means of editing EPCON (see below). ***JUST*** will automatically assume that the double wide mode is terminated after a double wide line, and reset the margin and line length.

Below is a sample of text with commands, followed by the formatted output.

```
 ,w:64
 ,m:8
 ,s:3
 ,c
 This is a Test
 ,p
 This is how a paragraph might look as input by the programmer
 into the text formatter program.  It would be "jagged" at the
 right edge and could
 contain very short lines
 such as the one above, due to editing the text extensively.

            This is a table........................1
            More table.............................2
            Still more.............................3
 ,p
 This is the next paragraph, which will start the Fill mode
 again. The paragraph will be nicely formatted.
 Blanks used to fill the
 line for justification are inserted alternately at the left and

 right of the line.
```

```
                         This is a Test

     This  is  how  a  paragraph  might  look  as  input  by the
programmer  into  the  text  formatter  program.  It  would   be
"jagged"  at  the  right edge and could contain very short lines
such as the one above, due to editing the text extensively.

          This is a table......................1
          More stuff...........................2
          Still more...........................3

     This is the next paragraph which will start the  Fill  mode
again.  The  paragraph will be nicely formatted.  Blanks used to
fill the lines for justification  are  inserted  alternately  at
the left and right end of the line.
```

That is how the text would be handled by the formatter. The parameter nn must be separated from the command by a colon ":". The program is supplied with page length set to 66 (11 inches at 6 lines per inch). Each page, including the first, starts with 5 blank lines. The page ends with two blank lines, a centered page number, and two more blank lines. At the end of the text file, linefeeds are output down to the page number, which is output followed by a formfeed. The lines per page and number of blank lines top and bottom may be modified by the user.

An added feature is that intentional double spaces will not have more spaces added. This is to facilitate double spaces at the end of a sentence, without having them opened up to four, which looks bad in the output text.

The text may be generated with any available text editor. If you have an Assembler you may change the constants in the source program by editing and reassembling the configure program supplied, and then appending it to ***JUST***.

RUNNING THE FORMATTER

In order to avoid confusion with the disk format commands, (some suppliers use FORMAT for the utility name), this formatter uses JUST.CMD (for justify) as a filename. You simply type:

```
JUST,filename,destination
```

Filename is any valid FLEX filename. The default extension is .TXT, and the default drive is the working drive. The filename LETTER would be the same to ***JUST*** as if you had typed LETTER.TXT.1 when you have set up the working drive as 1. Destination may be a filename, in which case output will be to that file. If there is no destination, ie. JUST,filename ... the output will be directed to the terminal or if the command P,JUST,filename is used, to the printer. If pause is enabled, it will be honored for output to the terminal. That is, the text will stop and it may be started again with the escape key, or the operation aborted by RETURN. If output is to the printer ***JUST*** will automatically shut PAUSE off, but output may still be stopped by using the escape key. On exit, ***JUST*** will automatically restore the PAUSE status that was in effect before running ***JUST***.

Output to a disk file will be formatted according to the embedded commands in the input text file. All printer control commands will be passed intact into the output file. If you use the TSC LIST utility later in an attempt to print the file, these commands will not be passed on to the printer since all of the control sequences start with the ESC (escape) character $1B. FLEX LIST will not pass any control codes (code lower than $20).

There is therefore a companion Utility to ***JUST***. It is called FPRINT. FPRINT does not suppress the ESC character. Thus, you may write a formatted output file with printer commands embedded, and list it to the printer with full formatting and printer control by means of FPRINT. Without this utility ***JUST*** would have to be used to output a text directly to the printer. In cases where you are going to run several copies of a text, it will save time to output to a disk file and use FPRINT to list the file a number of times.

There is one caution regarding output of printer commands to a file. FLEX text files use a technique called space compression. This operation is hidden within the FLEX operating system itself. When a text file is saved to a disk, spaces are not stored as consecutive space characters, but they are counted and stored as a "horizontal tab" character and a count of spaces. For example, a row of ten spaces is represented as $09,$0A. The $09 is the control character used to signal horizontal tab. Therefore, you may not store control code $09 in a text file, since on reading the file it will be interpreted by FLEX as Horizontal Tab and the next character will be interpreted as a space count. All other control codes, however, are usable. If you have a printer other than the Epson, and you modify the control strings for your printer, and it happens to use a Horizontal Tab for control, you will not be able to output a formatted text to a file, but will have to output directly to the printer.

It is a great convenience to be able to output a file to the terminal to see how it is going to format, and where it will page. Note that when output is to the terminal, the left margin spaces are suppressed. This makes maximum use of the terminal line length. It also fouls up the display of negative single indent lines. ***JUST*** will honor the FLEX Pause feature so that the text will stop every screenful until you hit your escape key (or whatever key is defined as escape with the TTYSET utility). ***JUST*** automatically ignores Pause when outputting to your printer or a text file.

ADDED FEATURES FOR USE WITH GRAFTRAX

If you have an Epson printer with the Graftrax option, it is possible to *italicize* a word or words in the middle of a line. It is also possible to **emphasize** or **boldface** a word or words. If your printer is equipped with GRAFTRAX+ (as opposed to GRAFFTRAX 80), you can also turn the underline mode on and off. Since the control characters must be sent to the printer in the middle of a line, a new feature of ***JUST*** has been implemented to handle such commands. The backslash character (\)is used to signal ***JUST*** that the next character describes a control string to be sent to the printer. There are 31 possible strings designated by the characters 0 through 9, A through F following the backslash. An Assembler source program called EPCON is supplied on the disk with ***JUST***. EPCON defines the strings starting with the one sent for code zero. Each string is 8 characters long. As supplied, EPCON has the strings defined as follows:

Underline On	\0	underlined text	(works with GRAFTRAX+
Underline Off	\1		and not GRAFTRAX80)
Italics On	\2	*italic text*	
Italics Off	\3		
Emphasize On	\4	**emphasized text**	
Emphasize Off	\5		
Boldface On	\6	**boldface text**	
Boldface Off	\7		
Superscript On	\8	$^{\text{superscript text}}$	
Superscript Off	\9		
Subscript On	\A	$_{\text{subscript text}}$	
Subscript Off	\B		

\C through \V is unused. If you want to print a backslash, you simply use two in a row \\ and one will print.

MODIFYING CONSTANTS

The formatting constants and initial values for some of the variables are settable by means of a constants in the EPCON file as well. Locations of the values are:

$000F	Default line width, initially 66
$0010	Default left margin, initially 8
$0011	Default paragraph indent, initially 5
$0012	Vertical margin (ie. blank lines at top and bottom of page, set to 5)
$0013	Page length (total lines) set to 66

The next several locations contain the control strings for the commands to the printer for four mode switches. For the Epson, these are defined as follows:

$0014	Normal mode	ESC,'H,ESC,'F,0,0
$001A	Double Strike	ESC,'G,ESC,'F,0,0
$0020	Emphasized	ESC,'E,ESC,'H,0,0
$0026	Both	ESC,'E,ESC,'G,0,0
$002C	Double Width	$0E,0,0,0,0,0
$0032	Cancel Double	$14,0,0,0,0,0

The command strings may be changed by editing and reassembling the EPCON file and appending it to ***JUST***, or by using the command GET ***JUST***, exiting to a monitor and putting the desired strings in memory starting at $1200 and at $000F. Remember to use 6 or 8 bytes for each as required, filling unused bytes with null (00). Then SAVE JUSTNEW.CMD.0,0,124F,0. Try JUSTNEW, and if it works as desired, delete JUST.CMD and RENAME JUSTNEW.CMD, JUST. If you have made an error, delete JUSTNEW.CMD, and try again. Of course you will always save the original disk on which this software is supplied with the unappended version of ***JUST*** so that you can start fresh if you change printers or terminals on your system.

With these two user configurable tables, it should be possible to adapt ***JUST*** to a number of printers with other features and control codes than those of the Epson. If character width is changed, (as in use of double wide printing) the line length will change. ***JUST*** will automatically adjust both the left margin and the line length to provide proper centering or right justification. Since these values must be divided by 2, centering of double wide will work better if the margin and line length are both even numbers.

JUST has been tested thoroughly. There are, however, an unbelievable number of possible combinations of conditions to be checked. It is not only possible, but highly likely that some bugs will show up. If you think you have found one, please document it (send a printed copy of the formatter output and a LIST of the input file). We will send corrections and/or additions to ***JUST*** and advise all customers of new releases with added features as they become available.

ADAPTING TO YOUR SYSTEM

The disk supplied contains several files. The source file for ***JUST*** is JUST.TXT. It was written in Windrush PL9, and is included for completeness. JUST.BIN is the output of the compiler. It has the Epson specific strings for the comma commands, but not the backslash commands. The file EPCON.TXT is the source for a short assembler program that defines both sets of command strings. You may edit the text file for another printer, assemble it and append the binary output file to JUST.BIN to make JUST.CMD. This has been done, and if you have a standard Epson with Graftrax you may use JUST.CMD directly. You may want to copy it to your system disk.

IMPORTANT NOTE

I've found that there are Epson printers without Graftrax, some with Graftrax, some with Graftrax+, and some newer ones with all the features of Graftrax and some others built in as supplied. There are some differences in the control sequences to do various things with these. If some feature here does not work with your printer, consult the manual and make the necessary changes.

These instructions were processed by ***JUST***. In order that you may look at a long text with ***JUST*** commands, the file JUSTINST.TXT has been included also. It is the source file for these instructions. Copy it onto another disk and feel free to experiment with it.

FPRINT.TXT and FPRINT.CMD are the source and output files for the FPRINT utility. It was also written in PL9 and it uses the FLEX.LIB library file of PL9, as does ***JUST***. FLEX.LIB contains all the file handling procedures.

AUTHOR'S NOTE

JUST was first written in Pascal. It was a chapter for my book "From BASIC to Pascal" (Tab #1466 1982). In its Pascal form, it reached version 17. Each version was a bit more capable than the preceding one. Its present form as a PL9 program is recent. I found that I could reduce the code nearly to half of the original by taking advantage of PL9's features. I found working on it a bit like eating popcorn. As I ran each new listing I would look at it and see a further possible improvement or feature. I had thought of the "backslash commands" for some time, and decided to implement them. I decided near the end of the project that most of the purchasers wouldn't have PL9 so I would have to implement a way for them to customize ***JUST*** to a printer other than the Epson. The Assembler overlay was the obvious answer. I had not implemented the double wide because it seemed incompatible with centering and right justification. It dawned on me that I could divide the margin and line length values by 2 and if they were even, everything would work out properly, so that was implemented.

Strangely, that last addition was about the easiest part of ***JUST***.

Programmers might note the CASE statement that is the command table. Commands that could be implemented in a single statement were included in the CASE structure itself:

```
CASE 'J THEN JUSTIFY = TRUE;
```

Commands that required several statements were made PROCEDURES and the case statement simply calls the proper procedure. If you have PL9 and want to play with ***JUST***, just add a case for a new command character and then add a procedure to cause the proper action to take place.

I have one note of caution. Text processors seem simple in concept and they are. There are, however many many special cases that must be handled. These cases are not usually obvious until the processor produces some unexpected result. Writing ***JUST*** was a real challenge. Should you have any ideas for enhancements, or should you make some improvements to ***JUST***, I would appreciate hearing about them. (BY LETTER, PLEASE). Also, if you work out additional backslash codes or a different set that are compatible with a particular model printer, Epson or any other, I would like a copy of your version of Epcon along with the make and model number of the printer with which it works, so that I might include such information in future releases of JUST.

Ronald W. Anderson
3540 Sturbridge Ct.
Ann Arbor, MI 48105