TEXT (Version 3.0)
Text Formatter

Dr. James J. Gillogly
2520 S. Chard Ave.
Topanga CA 90290
(213) 455-1407
10 May 1980

1. Description

Text is a text formatting program for Heath's HDOS operating system greatly extended from format [1], by Kernighan and Plauger. It will format documents (such as this software description) for printing on a typewriter or line printer. Though small, Text contains all of the important functions that a text formatter should have, including filling, spacing, indenting, right margin justification, underlining, and pagination.

2. Usage

To call Text from HDOS, give the command

```
text infile
```
or
```
text infile outfile
```
or
```
text [switches] infile [outfile]
```
or
```
text -v
```

where infile is the raw Text input file, outfile is the desired location of the finished document, and the brackets surround optional material. Omitting outfile sends output to the terminal. The filename may be "LP:" to direct output to the line printer.

The -v switch prints Text's version number. Other switches are:

-w - wait between pages for the user to type a newline on the controlling terminal; beeps between pages

-t - controlling terminal is not an H89/H19: do not put underlining or other overstrikes in inverse video

-o3 - start output at page 3 (use any number here)

The switches can be used together, e.g.

```
text -w -o7 doc.txt lp:
```

to output to the line printer starting at page 7, waiting between pages.

3. Inputting Text

Input to Text is prepared with any standard editor; for example, the Pie screen editor available from Walter Bilofsky (14478 Glorietta Drive, Sherman Oaks CA 91423) or the EDIT program supplied with HDOS. For straightforward text, simply type it as it should appear on the output. The text is normally filled (i.e. the line lengths are evened out) and right-justified.

Two-letter commands beginning with a period invoke special processing features of Text. For example, underlined words or phrases appear on a separate line preceded by the command ".ul". A list of all the legal commands appears in Section 15. Each command must be on a separate line and the period must be in the leftmost column.

Text honors indentations, so that a paragraph can be started with an arbitrary number of spaces. If the usual 5 spaces are desired, the .pp command can be used to separate paragraphs. Extra spaces typed within a line are retained, and a period at the end of an input line is followed by two spaces on output.

4. Text Filling and Justifying

Text fills each line with words until it runs out of room, ignoring the line endings of the input text. To prevent filling, use the command

```
.nf
```

This is useful for including tables such as the command list of Section 15 exactly as they were input. The inside address of a letter is typed in no-fill mode. The command

```
.fi
```

restores fill mode. Both .nf and .fi stop processing the current output line when they are encountered. That is, they cause a "break".

A break can be explicitly requested using the command

.br

which terminates the current output line and begins a new one with the text from the next input line.

While filling, Text right-justifies the lines by adding extra spaces between words to make the right margin line up exactly. The spaces are added alternately from each side to prevent obtrusive rivers of spaces from appearing in the output text. To stop justifying, use the command

.nj

and to resume justifying use the command

.ju

5. Indenting

Text normally prints the entire document starting each line at the first available character position. This can be changed using the command

.in value

The value can be an absolute number of columns to indent, or it can be a relative value. For example,

.in 6

would indent subsequent text by 6 character positions (giving a 1-inch left margin for many printers).

.in +5

would indent 5 characters deeper than the current indentation level, as for an indented paragraph.

Hanging indents This paragraph was created by giving the commands

.in +17
.hi "Hanging indents" -17

to achieve an indented and labeled block paragraph. The .hi command can be followed with a value to specify the indentation level for the label:

.hi string value

If the value is omitted, the label will begin at the left margin.

To get an indentation only for the next input line, use the command

```
.ti value
```

where the value can again be either relative or absolute.

The initial line of a paragraph can be indented by any number of spaces. To get the standard 5-letter indentation, use the command

```
.pp
```

which is equivalent to

```
.ti +5
```

6. Centering, Underlining and Overstriking

The command

```
.ce integer
```

will center the specified number of input lines without filling them. If no number is specified, only the next input line is centered. To center a large number of lines without counting them, say

```
.ce 1000          or some other large number
line 1 to be centered
more text to be centered
...
.ce 0             to stop the centering
```

The text is centered between the current indentation level and the right margin (see Section 8).

The commands

```
.ul integer
.cu integer
```

cause the specified number of input lines to be underlined. ".ul" is used for underlining individual words, and ".cu" is used for continuous underlining, including interword spacing. If no number is specified only the next input line will be underlined. The same trick used to center many lines (above) can be used to underline a lot of text. Filling and justifying are not affected by underlined text. On an H19 terminal (e.g. in an H89) Text will use the inverse video function to highlight underlined text directed to the screen.

Text underlines text by outputting one line containing only the underline characters, then a carriage return without a line feed, and finally the text to be underlined. If the output device does not recognize a carriage return, the command

```
.bs 1
```

will cause subsequent underlining to type the character followed by a backspace character and an underline for every character. This is slower than the default method (which can be re-entered with the

```
.bs 0
```

command).

To overstrike a character, the string "\b" can be used to represent one backspace. For example, the Welsh word "môr-leider" (pirate) is input as "mo\b^r-leider".

7. Page Control

The standard page length is 66 lines. To change this to a different length, use the command

```
.pl value
```

The value can be either absolute or relative.

During the first draft of a document, the page breaks may appear in inconvenient places. To force a page break (e.g. to keep a paragraph together), use the command

```
.bp
```

which will immediately eject the current page. If it is followed by an integer, e.g.

```
.bp 7
```

the next page will be page 7. (Page numbers are discussed in Section 10.)

A conditional page break allows more sophisticated control of the page ejection. The command

```
.ne integer                ("need")
```

tells Text to eject the page if there are less than the specified number of lines remaining on the page. Thus, for example, the command

```
.ne 4
```

might be used in no-fill mode to keep a four-line table together.

8. Setting the Line Length

The line length is controlled by setting the left and right margins. Indentation handles the left margin. E.g.

```
.in 6
```

would specify a typical left margin for a line printer. Indentation is discussed in more detail in Section 5. The command

```
.rm value
```

modifies the right margin. The value may be relative or absolute. Initially the right margin is set at 60.

9. Vertical Spacing

To get blank vertical space in the output file, either leave blank lines in the input (each becomes a blank output line) or use the command

```
.sp integer
```

to get the specified number of blank lines. If no integer is specified, one blank line is output.

To get more than one newline between each output line, use the command

```
.ls integer
```

For example,

```
.ls 2
```

forces double spaced output.

10. Headers and Footers

Each line may have a 1-line header and/or a 1-line footer, either of which may include automatically counted page numbers (see also ".bp integer", Section 7). These titles have three parts: left-adjusted, centered, and right-adjusted. The commands

```
.he "left string"centered string"right string"
.fo "left string"centered string"right string"
```

are used to define the header or footer lines. Leading or trailing blanks may appear in any string, and will be counted in the adjusted or centering. The quoting character may be any non-blank character in place of the double quote shown in the examples. If any title string includes a # sign it is replaced

with the page number of the current page. The margins used for adjusting the three-part titles are those in effect at the time the last title was defined.

The top and bottom margins are initially set at 5 blank lines. A one-line header and/or footer, if defined, becomes the third of the five lines. The space above and below the header and footer lines are initially set at 2 lines above the header, 3 lines from header to text (including the header), 3 lines from text to footer (including the footer), and 2 lines below the footer. These may be changed by the user:

```
.h1 value   lines above the header
.h2 value   lines below and including the header
.f1 value   lines above and including the footer
.f2 value   lines below the footer
```

The values may be either relative or absolute.

11. Including other files

To include the text from another file use the command

```
.rf filename              ("read file")
```

Text will interrupt processing of the original file, process all the text from the specified file, and then switch back to the original file. The new file may include another .rf command, to a nesting limit of about 5 files.

This feature is particularly useful for including boilerplate such as the copyright notice at the beginning of this document. It may also be used to include files from another drive, if the document is too big to fit on one diskette.

Unless the filename explicitly specifies the drive (e.g. SY2:BOILER.TXT), the .rf file will be taken from the same drive as the original file.

12. Terminal Messages

While processing text to a file or printer, it is sometimes useful to see confirmation on the screen that something is happening. The command

```
.tm Any text can appear here
```

will cause the text to be printed on the controlling terminal when the command is encountered.

13. String Substitution

Text provides for 26 user-defined strings, identified by the letters a-z. To define string x, use the command

```
.sb x string
```

To invoke the string, embed \sx in the text wherever the string is to occur. For example, at the beginning of this file is the definition

```
.sb t  \bT \be \bx \bt
```

which defines the word Text. Each underlined occurrence of Text in the text is obtained by using the abbreviation \st.

14. Escapes

The escape symbol "\" is used to evoke special Text processing features. In earlier sections the \b (backspace) and \s (string substitution) features were described. The other escapes are:

```
\\ - give a single \ on output
\" - everything on the line after this is ignored (comment)
\& - null character, mainly used for getting a "." at the
     beginning of a line without having it introduce
     a command.
```

15. COMMAND SUMMARY

The commands implemented in Text are:

Command	Description	Default	Section
.bp [n]	begin page numbered n	---	7
.br	start new output line	---	4
.bs n	n=1: use <BS> for underline	0	6
.ce n	center n input lines	1	6
.cu n	continuous underline n lines	1	6
.f1 [+/-]n	lines above & incl footer	3	10
.f2 [+/-]n	lines below footer	2	10
.fi	fill mode	YES	4
.fo "str"str"str"	three-part footer	none	10
.h1 [+/-]n	lines above header	2	10
.h2 [+/-]n	lines below & incl header	3	10
.he "str"str"str"	three-part header	---	10
.hi string [[-]n]	hanging indent	n=0	5
.in [+/-]n	indent n characters	---	5,8
.ju	right justify (must fill)	YES	4
.ls n	double spacing or more	1	9
.ne n	need n more lines on page	---	7
.nf	stop filling text	NO	4
.nj	end right justification	NO	4
.pl [+/-]n	set page length to n	66	7
.pp	paragraph indent of 5	---	5
.rf filename	read specified file	---	11
.rm [+/-]n	set right margin to n	60	8
.sb x string	substitute string for \sx	---	13
.sp n	space down n output lines	1	6
.ti [+/-]n	indent next output line	---	5
.tm message	print message on screen	---	12
.ul [n]	underline n input lines	1	6
\ escapes	special character functions	---	14

Reference

[1] Kernighan, B.W. and Plauger, P.J. Software Tools. Addison-Wesley Publishing Company, Reading MA, pp. 219-250.