

1 OVERVIEW OF TEX 2.0 ENHANCEMENTS

This document is an addendum to the Digital Research manual, "CP/M TEX Text Formatter User's Guide." It overviews the operation of TEX version 2.0 and highlights its new features. Section 1.1 describes options which can be invoked at TEX run-time. Read this section before running TEX. Section 1.2 describes the escape sequences TEX accepts within the source text, and within words, for added formatting control. This section need not be read until you are familiar with TEX. Section 1.3 surveys the TEX formatting commands. It is organized by function rather than in alphabetical order, and includes additional commands not described in the TEX manual. Finally, Section 1.4 summarizes the new features in this version. It is recommended that you use the appendixes as references for the TEX commands and usage options.

TEX accepts lines of input text interspersed with formatting control lines and produces a printable, paginated document whose format is user-designed. Format control lines begin with a control character, usually a . (period) or ` (accute accent), followed by a two-letter command and optional command arguments. The produced document is a single column of formatted text.

The source text file can be prepared using any standard text editor such as the Digital Research editor EDtm. However, if not using EDtm, be sure your editor is in non-document mode and produces standard ASCII files without non-printing characters other than the standard tabs, carriage returns, and line feeds.

The TEX 2.0 filling and margin adjustment routines have been enhanced to provide greater flexibility and improved document appearance. The new fill and no fill commands give the user control over text filling. Text insertions can be made either from the console or from a separate file, simplifying production of personalized form letters. Usage flexibility is increased by the addition of new TEX options.

A number of special commands have been implemented for use on the Diablo 1640/1650 printer with the XMEM2 optional printed circuit board assembly available from Xerox. With this hardware, TEX can underline, double strike characters for a bold appearance, and proportionally space to produce evenly spaced words in adjusted lines. These facilities will be provided for other hardware in future versions through a user-modifiable output module, similiar in concept to the BIOS in CP/M. Headings and footings can contain the page number and can be printed in bold face on the Diablo. Appendix D describes the formatting commands which require the Diablo 1640/1650 and XMEM2 PCB assembly combination.

TEX 2.0 can automatically generate an index and a Table of Contents. You can include additional source files (.SO command) and chain to the next source file (.NX command). You may also specify multiple input files when you invoke TEX.

1.1 Usage and Options

The general form to invoke TEX 2.0 at the CP/M or MP/M command level is:

TEX files options

where files represents one or more file specifications separated by commas or spaces, and options represents a string of option arguments preceded by a "\$". The maximum length of the command line can not exceed 127 characters. Option arguments have the forms:

\$t \$td \$tn \$tn,m \$tf

where

t represents the option type,
d represents the option argument device,
n and m represent numeric arguments, and
f represents a CP/M file name.

The only required "\$" is for the first option argument. Option arguments should be separated by a comma or spaces, although a separator is not absolutely required unless the previous option specified in the list can have a numeric or file name argument.

The option argument device, d, may be a letter in the range "A" through "P" corresponding to one of sixteen logical drives, or one of the following special characters:

X - console

Y - printer

Z - byte bucket

Numeric arguments of options may be either a positive (+n) or negative (-n) increment, or an absolute value up to 65,536. The options, which may appear in any order so long as they appear after the files, are:

Option	Effect
Td	Location of TEX source file. The default TEX file type is "tex". The special devices X, Y and Z are not allowed. The default location if this parameter is not used is the current default drive.
Pd	Location of PRN output file. The output file name will be the same as that of the first source file. The special devices X, Y and Z are allowed. The default location if this parameter is not used is the location of the source file.
Ed	Destination of error messages resulting from text processing. Only the special devices X and Y may be used. The default for this parameter is the console (X) device.
Xd	Location of the IX index and table of contents output file. The name and default location are those of the source file. The special device Z suppresses this output. The X and Y devices are not allowed.
S	Stop after each page until a console key is pressed. This option allows single sheet insertion for text printer output.
F	Place form-feed characters in output. An initial form-feed is output before any text and one form-feed is output at the end of each page. Note, this will not cause page skipping even if the bottom margin is zero.
C	Chaining control. When a .NX command is encountered in the source text the query: "Continue (Y/N) ?" is displayed. A yes response will execute the .NX command while any other response will not.
D	Diablo printer. This option enables the Diablo 1640/1650 formatting commands. It should also be used if the source text contains special non-printing control characters.
Q	When it follows the D option in the command line, the Q option initiates proportional spaced printing (equivalent to a leading .PS command in the source text).
R f	Read insertions from the file f whenever a .RD (read insertion) command is encountered in the source text. This parameter makes it easy to generate form letters with a source file which chains to itself, (equivalent to a leading .RF command in the source text).
N n	Set initial page number to n. This command is overridden by a Page Number (.PN n) command in the text.
O n,m	Omit output of pages not in the range n to m.

1.2 Escape Sequences

Various special functions may be introduced anywhere in the input text (except control lines) by means of an escape character, usually `\`. For example, the sequence `\c` allows a line to be continued on the next non-control line. This makes it possible for certain formatting commands to be applied to partial words or single letters in words. Two backslashes (`\\`) produce a single backslash in the output text. A backslash followed by character which does not form a special escape sequence produces that character in the output text. The escape character may be changed with the `.EC` (escape character) command. The defined escape sequences are:

Escape Sequence	Meaning
\\	\ (to allow a backslash in the output).
\(space)	Unpaddable space character. The output text line will not be broken at this position.
\-	- The output text line will not be broken following this hyphen.
\"	Beginning of a comment. The comment is terminated by a RETURN.
\b	Backspace. This produces a backspace in the output text to facilitate overstriking. For example, the sequence: +\b produces +. Note: the Diablo printer option (\$D) must be used and the characters immediately following the \b cannot be another escape character sequence.
\c	Interrupt text processing. The next encountered input text line will be considered a continuation of the same line of input text. If a word is interrupted, the next encountered text will be taken as a continuation of the interrupted word. If intervening control lines cause a break, any partial line will be forced out along with any partial word.
\d	Down 1/2 line. This causes a positive half-linefeed on the Diablo 1640/1650. Note: the Diablo printer option (\$D) must be used.
\u	Up 1/2 line. This causes a negative half-linefeed on the Diablo 1640/1650. For example, the sequence: \u\d produces ¹ . Note: the Diablo printer option (\$D) must be used.
\(RETURN)	Concealed (ignored) RETURN. The next line of input is taken to be a continuation of the current line.
\X	X. X is any character not listed above.
\s n	Shadow print. In headings and footings only, (.HE and .FT), this works exactly like the .SH (shadow print) command in regular text. Note: the Diablo printer option (\$D) must be used.

1.3 Formatting Commands

This section contains a brief overview of the commands in TEX: for additional information refer to the command descriptions in the Digital Research manual: "CP/M TEX Text Formatter User's Guide." The following notation is used for command arguments:

- +n (or +m) denote a numeric argument which can be a positive (+n) or negative (-n) increment, or an absolute number (n).
- n (or m) denote an absolute numeric argument.
- c denotes a single ASCII character.
- s denotes a string argument which should be separated from the command by one space.
- f denotes a CP/M file name which may have a drive designation and a file type. The source file drive and type "TEX" are the defaults.
- b indicates that the command normally causes a break. The break function forces the output of any partially filled line.

Page Control

- .BP +n b** Begin page. The current page is ejected and a new page is begun. If +n or n is given, the new page will be numbered +n or n. If no text has been printed on the current page, the .BP command is ignored. Instead of n the word "odd" may be used forming the special command: **.sh .BP ODD. .qs** When used, the current page is ejected and the next page is numbered the next odd page number (useful for forcing sections to begin on a facing page.)
- .NE n** Need n lines of veritical space. This will cause a .BP command if n lines are not available on the current page.
- .CP n** Conditional page, equivalent to the .NE need lines command.
- .PA n b** Page advance, n pages are printed with only headings, footings and page numbering. If the current position is the first line of a new page, n is decreased by 1.

- .PL +n** Page length. The page length is set to +n lines or to n lines. All headings and footings are included in the page length. The initial value is 66 lines.
- .PN +n** Page number. Page numbering is turned on and the current page number is set to +n or n, if given. This is initially on. Note: this does not effect the printing of page numbers in the heading or footing.
- .OP** Omit page numbers. Page numbering is turned off. Page numbers will not be printed below the footing, although they are still counted. This is initially off. Note: this does not effect the printing of page numbers in the heading or footing.
- .PO +n** Page Offset. The current left margin is set to +n or n spaces. The initial value is 8 spaces. The space size will vary depending on the point size.

Text Filling, Adjusting and Centering

- .BR** b Break. The filling of the line currently being collected is stopped and the line is output without adjustment.
- .FI** b Fill subsequent output text lines. This causes TEX to place as many words as possible on each output line. Fill is initially on.
- .NF** b Nofill. Subsequent input text lines will be output without filling of words or margin adjustment. Indentation and other TEX parameters are not affected.
- .AD** b Adjust text to stretch output lines out to the right margin. if fill mode is not on, adjustment will be deferred until fill mode is back on.
- .NA** b Noadjust. Adjustment is turned off causing the right margin to be ragged. Output line filling still occurs if fill mode is on.
- .LI** b Literal. This turns off adjustment, fill and indentation temporarily until the next formatting command is encountered. Input is copied without formatting to the output. In proportional spacing mode a fixed character width is set to preserve columnar alignment in tables.
- .CE n** b Center the next n input text lines within the current linelength. If n is 0 centering stops.

Vertical Spacing

- .LS ± Line spacing. The spacing is set or adjusted by ± n. The default is one (single spacing).
- .SP n b Space n output line spaces. A line space is determined by the current line spacing (.LS). If the bottom of the page is reached, the next page is begun and spacing is stopped. The default is one space.
- .SS Single space output lines. This is equivalent to .LS 1. This is initially on.
- .DS Double space output lines; equivalent to .LS 2. This is initially off.

Headings and Footings

- .MT ±n Top margin. This command sets or adjusts the number of lines to be left blank at the top of the page. Any heading which is defined will print within the top margin. The initial value is 6 lines. Note: all margins are included in the page length.
- .MB ±n Bottom margin. This command sets or adjusts the number of lines to be left blank at the bottom of the page. Any footing which is defined will print within the bottom margin. If page numbering is in effect the page number will be printed centered between the footing and the bottom of the page. The initial value is 5 lines.
- .HE s Heading. This command defines a one line heading which will be printed at the top of the page. The heading is left justified. The heading may be shadow printed on the Diablo 1640/1650 using the escape sequence: \sn, where \ is the current escape character and n is an optional number representing the HMI. This works exactly the same as a .SH n command in normal text. The sequence \\ produces a single \. The heading can be discontinued by placing a .HE command without any string in the text. This produces a null heading. If the heading contains the page number character (initially %), the page number will be inserted at that point each time the heading is printed. The initial value is a null heading.
- .HM ±n Heading margin. This command positions the heading within the top margin. The heading margin is the number of lines between the heading and the first line of text on a page (the bottom of the top margin). The initial

value is 3 lines.

- .FT s** Footing. This command defines a one line footing which will be printed at the bottom of the page. The footing is left justified. The footing may be shadow printed on the Diablo 1640/1650 using the escape sequence: \sn, in the same manner as in the heading command. The footing can be discontinued with a null .FT command, that is .FT(RETURN). If the footing contains the page number character (initially %), the page number will be inserted at that point each time the footing is printed. The initial value is a null footing.
- .FM +n** Footing margin. This command positions the footing within the bottom margin. The footing margin is the number of lines between the last line of text on the page and the footing. The initial value is 1 line.
- .PC c** Set page number character to c. If c is not given the page number character is set to its initial value of %. After this command the character c in a heading or footing will be replaced by the current page number.

Line Length and Indenting

- .LL +n** Line length is set to + n. This is the maximum line length for fill mode. The line length includes any indents but not the page offset.
- .IN +n** b Indentation is set to + n. A value of zero cancels indentation.
- .TI +n** b Temporary indent. The next output line will be indented a distance of +n with respect to the current indent. The current indent is not changed.
- .QI** b Quit indenting, equivalent to .IN 0. of +n with respect to the current indent. The current indent is not changed.
- .PP +n, +m b**

Paragraph. This produces the command sequence:

```
.sp
.ne m    m is initially 2 but can be reset with .pp n m
.ti n    n is initially 6 but can be reset with .pp n
```

The need (.NE) command insures that the paragraph will not begin on the bottom line of a page, the temporary indent is the paragraph indentation and the space puts one blank line between paragraphs. If no arguments are

given the current values of *n* and *m* are used.

Input File Switching

- .SO f** **b** Switch source file. After this command, input is taken from the file named *f*. When the new file ends input is again taken from the original file. Note: .SO commands cannot be nested or used at the same time as read insertions (.RD) from a file.
- .NX f** **b** Next file is *f*. The current file is considered ended, and the input taken from the new file named *f*. If no filename is given, .NX is the same as an end-of-file. If the chaining control option (\$C) is used, TEX allows the user to choose to either execute the chaining command or treat it as an end-of-file.

Insertions

- .RD s** Read insertion. The console bell is sounded and the string *s* is displayed on the user's terminal as a prompt, and input is switched to come from the console keyboard. Input will continue from the keyboard until two RETURNS are entered in a row. Insertions can also be taken from a file through the use of the .RF (read from file) command before the .RD command, or through the use of the \$R option.
- .RF f** Read from file named *f*. Subsequent read insertion (.RD) commands will take input from the file named *f*. Each insertion is still ended with two RETURNS in a row. This is equivalent to the \$R option. Note: Read insertions from a file cannot be made from a source file included with the .SO (switch source file) command.

Index and Table of Contents Entries

.IX s Index entry. The string *s* is saved as an index entry in the IX file. This file has the same name as the first source file and the type IX. The location of this file is the same as the source file unless redirected with the X option. After TEX is finished, the INDEX program can be run to generate an index for the document. The index entry string consists of,

key word, modifier

The index will be alphabetized by key words and each key word can have an optional modifier. The indexing procedure is explained further in Section 3 of this document.

.TC s Table of contents entry. The string *s* is saved as a table of contents entry in the IX file (see above). After TEX is finished, the INDEX program can be run to generate a table of contents for the document. The table of contents entry should be input exactly as it should appear, including any desired indentation. The table of contents generation procedure is explained further in Section 3 of this document.

Control Character Translations

.EC c Set escape character to *c*. If *c* is not given the escape character is set to its original value of \. See section 1.2 for the escape sequences supported in TEX 2.0.

.CC c Set the basic control character to *c*, or reset to '.' if *c* is not given. The control character indicates a formatting command when it is the first character in a source text line.

.C2 c Set the nobreak control character to *c*, or reset to '^' if *c* is not given. The nobreak control character indicates a TEX formatting command in the same manner as does the basic control character (see above), except that it suppresses any break normally produced by the command.

Miscellaneous

- .TM s** Terminal message. The console bell is sounded, and the string *s* is displayed on the user's console. This is useful for indicating the beginning of a new source file when multiple source files are being processed.
- .TA +n, +n, +n, ... +n**
- Set tabs. This command sets or adjusts the TEX tab stops. Tab stops are stored as columns, where columns begin at the left side of the page, not at the page offset. A maximum of 19 tab stops can be set. The tab stops are initially at columns 9, 17, 25, 33, ..., 129.
- .CL s** Command line. The string *s* must begin with a \$ and consist of valid TEX 2.0 run-time options, it is treated exactly as the option string typed when invoking TEX.
- .IG** Ignore. Subsequent input text lines are ignored until the next TEX formatting command.
- .ST s** b Stop. When the .ST command is encountered, TEX sounds the console bell, displays the string *s* as a prompt on the terminal, and displays the message: "Hit any key to continue:." When a console key is depressed TEX continues its processing. This allows TEX output to be interrupted for operator intervention, such as the changing of print wheels.

1.4 Features and Performance

1. In a benchmark in which the Digital Research document: "CP/M 2.0 User's Guide for CP/M 1.4 Owners" was formatted, TEX 2.0 ran 25% faster than TEX 1.0.
2. The text filling and adjusting algorithms in TEX 2.0 have been improved. Multiple inter-word space characters in the input text are retained, except for trailing spaces. Adjustment of the right margin is controlled by the commands `adjust (.AD)` and `noadjust (.NA)`. The filling of words to the output text line is controlled by the commands `fill (.FI)` and `nofill (.NF)`. Input files without formatting commands produce reasonable output, with adjusted margins. Their indentation and paragraphing is preserved.
3. Multiple source files may be processed with a single TEX invocation through several mechanisms. The easiest way is to simply specify multiple input files in the source line. These will be processed in order of specification. Multiple source filenames in the command line may be separated by either spaces, commas or a combination of commas and spaces.

Alternatively, the chaining or include facilities may be used to direct subsequent input to be taken from the file specified in the input text. The next source file command `(.NX)` causes an end of file condition for the current source file and switches input to the new file. The source file command `(.SO)` is an include facility which causes TEX to take subsequent input from the file specified until the end of the file and then revert to the current primary source file for input.

4. In addition to the standard command control character `'.'`, a `^^` at the start of an input text line also indicates a formatting command. The acute accent, however, suppresses the break function caused by some of the commands. The break function normally forces the output of any partially filled line.
5. Indexing and automatic table of contents generation are now supported through the use of the formatting commands `.IX` (index entry) and `.TC` (table of contents entry), and the use of a separate program (INDEX) which creates a tex source file with the table of contents or index entries with the associated document page numbers.
6. The TEX 2.0 command line options format has been changed to conform with other Digital Research software products, such as LINK-80, MAC, RMAC and ASM-86.

7. Special purpose commands have been added supporting the word processing capabilities of Diablo 1640/1650 printers with the Xmem2 PCB assembly. These commands facilitate Bold Overprinting, Shadow Printing, Underlining, Incremental Justification and Proportional Spacing. Note that these options will not function unless the printer has the Xerox Xmem2 PCB assembly or an equivalent board.
8. Escape sequences have been added for additional formatting control. These are initiated by the use of the backslash (\) as an escape character. The pair \\ produces a single backslash in the output text. Escape sequences make it possible to boldface individual characters in a word and to override the padding of blanks and breaking of words on inbedded hyphens.
9. Insertions into the tex file either from the console or from another file are possible through the use of the read (.RD) and read file (.RF) commands. A .RD command indicates that subsequent input is to be taken from the console keyboard until two 'returns' are entered in a row. The .RF command and \$R option direct subsequent .RD commands to take the insertion from the file specified. This facility allows the easy generation of personalized form letters.
10. Striking any key causes the prompt: "Abort (Y/N) ?", to be displayed on the console. A 'y' response terminates TEX, deleting all output files. Any other response continues text processing.
11. TEX output files are placed on the drive designated for the source file unless explicitly directed to other destinations in the command line options.
12. Many functions may be turned on or off in blocks by using an arbitrarily large argument (less than 255) for number of lines to start the function and the same command with an argument of 0 to terminate the function. This means it is not necessary to count lines in many cases. Examples of this concept are:

.CE 100	.BO 100	.IN 10	.UL 100
.CE 0	.QB	.IN 0	.QU
13. Changing the linelength inside of an indent subtracts the indentation from the new linelength. (This was not done in TEX 1.0)
14. The page number can be printed in a heading or footing by designating its location with the "%" as a page number character. If the page number is printed in a heading or footing it will not be printed at the bottom of the page even if page numbering is in effect. The page count is incremented irregardless of whether page numbers are printed or not.

15. Comments may follow commands having no arguments or a single numeric argument. These comments do not require the escape sequence `\` and extend to the end of the line.
16. Most of the TEX commands are compatible with the UNIX text formatters: NROFF and TROFF.
17. TEX 2.0 requires a minimum of 16K of memory, that is an 10K code segment, 2K initialized data area and 4K of free buffer space. TEX is compatible with CP/M 1.4 and subsequent versions. INDEX 2.0 requires a minimum of 46K of memory.

2 TEXT FILLING AND ADJUSTING

This section discusses text filling and adjusting using TEX 2.0 and describes the use of the fill and no fill commands. The fill and no fill command descriptions supplement the adjust, noadjust and literal command descriptions in the Digital Research document:

"CP/M TEX Text Formatter User's Guide."

Normally, words are collected from input text lines and assembled into an output text line until some word doesn't fit. If the word contains a hyphen, an attempt is made to break the word following the hyphen in an effort to assemble part of it on the output line. Once an output line is completed, the spaces between the words on the output line are increased to spread out the line to the current line length minus any current indent. In the normal adjust mode (.AD), this is done by padding the spaces with additional space characters. The first line following a break is padded with blanks from right to left and subsequent lines are padded alternately from left to right and right to left. When padding from left to right spaces following a period are not padded until all spaces on the line have been padded once. This produces uniform padding when periods at the end of a sentence are followed by two blanks.

An input text line ending with ., ?, or ! is taken to be the end of a sentence, and an additional space character is provided during filling. Multiple inter-word space characters in the input text are retained, except for trailing spaces.

Adjustment of the right margin is controlled by the commands adjust (.AD) and noadjust (.NA). The filling of words to the output text line is controlled by the commands fill (.FI) and nofill (.NF). If the fill mode is turned off when margin adjustment is on, margin adjustment will be deferred until the fill mode is turned back on.

Blank lines in the input text produce blank lines in the output after causing a break in the filling of text. Leading blanks on an input line also cause a break and are passed through to the output, simulating a temporary indent (.TI) command. These two features allow input files without formatting commands to produce reasonable output. Such files produce justified output with paragraphing intact.


```
*****
*   .FI   *
*****
```

FILL

The fill command causes TEX to place as many words as possible on a line but does not justify the right margin unless margin adjustment (.AD) is on. Once given, the fill command remains in effect until either a literal (.LI) or no fill (.NF) command is given. The following example should clarify the effect of the fill command:

INPUT

```
.LL 40
.NA
.FI
```

This is an example
of the effect of the
fill
command.

Note that blank lines produce blank lines and
leading
spaces are preserved.
Otherwise as many words
as will fit
are placed on each output line.

OUTPUT

This is an example of the
effect of the fill command.

Note that blank lines produce
blank lines and leading spaces are
preserved. Otherwise as many words
as will fit are placed on each
output line.


```
*****
*   .NF   *
*****
```

NO FILL

The no fill Command offers an additional degree of latitude in formatting text. No fill mode is similar to the literal (.LI) mode except that indentation and line spacing are preserved. As in literal mode, lines exceeding the Line Length setting will **not** be broken following the last word fitting on the line. If margin adjustment (.AD) is on it will be deferred until fill mode is back on. The effect of the no fill command is best shown by example. The following example compares the effects of the adjust, noadjust, fill, no fill and literal commands:

INPUT

```
.LL 40
.IN 6
.SP
.LI
```

The CP/M Operating System
can be altered to execute with almost any
microcomputer configuration.

```
.SP
.NF
```

The CP/M Operating System
can be altered to execute with almost any
microcomputer configuration.

```
.SP
.FI
```

The CP/M Operating System
can be altered to execute with almost any
microcomputer configuration.

```
.SP
.NA
```

The CP/M Operating System
can be altered to execute with almost any
microcomputer configuration.

```
.SP
.AD
```

The CP/M Operating System
can be altered to execute with almost any
microcomputer configuration.

OUTPUT

The CP/M Operating System
can be altered to execute with almost any
microcomputer configuration.

The CP/M Operating System
can be altered to execute with almost any
microcomputer configuration.

The CP/M Operating System can
be altered to execute with almost
any microcomputer configuration.

The CP/M Operating System can
be altered to execute with almost
any microcomputer configuration.

The CP/M Operating System can
be altered to execute with almost
any microcomputer configuration.

3 INDEX AND TABLE OF CONTENTS GENERATION

This section discusses the use of the **INDEX** program and the **.IX** and **.TC** commands in **TEX** to produce a document table of contents and index. These are created as additional **TEX** source files which can be edited further, if desired. These files can then be processed with **TEX** to produce the final table of contents and index for the document. The first section discusses the procedure if the document is a single source file, and the second the case of multiple source files. The third and fourth sections discuss the **.IX** and **.TC** commands, and the last section describes the **INDEX** program.

3.1 Single Source File

The index and table of contents for a single source file is done in six steps:

1. Place the index (**.IX**) or table of contents (**.TC**) entries in the **TEX** source document file. For example, suppose the source file is

XXXXXX.TEX.

Edit **XXXXXX.TEX**, placing one **.IX** entry for each location in the document to be indexed, and one **.TC** entry for each section title or subtitle to go into the table of contents.

2. Run **TEX 2.0** to produce the document (**XXXXXX.PRN**) file or listing. This produces an intermediate file named:

XXXXXX.IX

This contains all the index and table of contents entries from the document.

3. Run the index program creating a file with the type **TEX**.
4. Run the index program again, this time creating a file of type **TEX** containing the document's index.
5. Edit the index and table of contents files, if desired, and edit the source file **XXXXXX.TEX**, placing **.SO TC** and **.SO INDEX** (switch source file) commands at the locations of the table of contents and index, respectively.

6. Finally, run **TEX** to produce the finished document.

Note:

The document page numbering must be started with a **.PN 1** (or some other number) after the table of contents and the beginning of the file should have a **.OP** to stop the printing of page numbers.

Example:

command		produces
A>ED X.TEX	----	X.TEX document source (containing .IX and .TC commands)
A>TEX X	----	X.PRN document (without index or table of contents)
	----	X.IX index and table of contents intermediate file
A>INDEX TC X	----	TC.TEX table of contents for X.PRN
A>INDEX INDEX X	----	INDEX.TEX index for X.PRN
A>ED X.TEX	----	X.TEX document text (add the commands: .SO TC .SO INDEX)
A>TEX X \$XZ	----	X.PRN final document

3.2 Multiple Source Files

With large documents it is usually preferable to break the document into separate files for each section. For example, the document X, in the example above might have 4 sections numbered 1 through 4. It could then be broken into the tex source files:

XTP.TEX X1.TEX X2.TEX X3.TEX X4.TEX XINDEX.TEX

This organization makes the index and table of contents generation process easier. It now has 5 steps:

1. Place the index (.IX) or table of contents (.TC) entries in the TEX source document files. Edit X1.TEX, X2.TEX, X3.TEX and X4.TEX, placing one .IX entry for each location in the document to be indexed, and one .TC entry for each section title or subtitle to go into the table of contents.
2. Use the TEX 2.0 command:

TEX X1,X2,X3,X4 \$PY

to print the body of the document on the printer. This also produces an intermediate file named:

X1.IX

containing all the index and table of contents entries from the document.

3. Run the index program creating a file with the type TEX.
4. Run the index program again, this time creating a file of type TEX containing the document's index.
5. Edit the index and table of contents files, if desired. Now print the title page and table of contents with the command:

TEX XTP,TC \$PY

and, assuming that the last page number in the document body is 80, print the index with the command:

TEX INDEX \$N 81 \$PY

Example:

command	produces
A>TEX X1,X2,X3,X4 \$PY	
---->	main body of document printed on printer (last page is 80)
---->	X.IX index and table of contents intermediate file
A>INDEX XTC X	
---->	XTC.TEX table of contents
A>INDEX XINDEX X	
---->	XINDEX.TEX index
A>TEX XTP,XTC	
---->	table of contents printed on the printer
A>TEX XINDEX \$N 81 PY	
---->	index printed on the printer beginning on page 81

3.3 Index Entry Command

The .IX (index entry) command has the format:

.IX keyword {, modifier }

where,

keyword represents the word(s) to be indexed, and
modifier represents the optional word(s) qualifying
the entry.

Note: The page number matched with each index entry is the current page when the .IX command is encountered, so be sure to place them at the document position you wish to index.

Example:

command form		produces
.IX Commands	---->	Commands, 23
.IX Commands, MT	---->	Commands MT, 23
.IX Commands, BP		
.IX BP command	---->	BP command, 15
. . .		Commands
.IX Commands, BP		(see also name)
.IX Commands, PA		BP, 15, 35
.IX Commands, (see also name)		PA, 35

3.4 Table of Contents Entry Command

The .TC (table of contents entry) command has the format:

.TC entry

where,

entry represents the (indented) section heading

Note: The page number matched with each table of contents entry is the current page number when the .TC command is encountered, so be sure to place them on the first page of each section. The following example produced a portion of the table of contents for this document:

Example:

.TC 1 OVERVIEW OF TEX 2.0 ENHANCEMENTS

.

.tc 1.1 Usage and Options

. . . (section 1.1) . . .

.tc 1.2 Escape Sequences

. . . (section 1.2) . . .

.tc 1.3 Formatting Commands

. . . (section 1.3) . . .

.tc 1.4 Features and Performance

. . . (section 1.1) . . .

3.5 The Index Program

The INDEX program is used as follows:

1. First invoke the program INDEX.COM or INDEX.PRL with the command line:

INDEX destination source

where

destination represents a CP/M file name of type TEX which will contain the index or table of contents. The type TEX is assumed if not entered. The destination can also be one of the following special devices:

\$CON	System Console
\$LST	System List Device (Printer)
\$RDR	System Reader Device
\$PUN	System Punch Device

source represents a CP/M file name of the intermediate file (the type IX is assumed), containing the index and table of contents entries.

2. The INDEX program displays the following:

Options:

- 1 - Table of Contents
- 2 - Table of Contents (with dots)
- 3 - Index

Option?

Respond with the desired choice.

3. If the Table of Contents with dots is selected, you are prompted for the linelength, (usually 70).

If the Index option is selected, you are asked if you wish to trace the building of the index. The usual answer is n (no).

The index is created using the Digital Research Product: BT-80tm to organize the index. This produces two temporary files on the default drive named TEXIDX.TMP and TEXIDX1.TMP which are deleted when the index is finished. The trace option can be used to monitor the index generation process. The only limiting factor regarding index size is the disk capacity of the default drive.

APPENDIX A

SUMMARY OF TEX COMMANDS

Function -----	Break? -----	Command Format -----	Initial Value -----	Default Value -----
Adjust Margins	yes	.AD	on	
Begin Page	yes	.BP +n		+1
Begin Next Odd page	yes	.BP ODD		
Break	yes	.BR		
Command Character	no	.CC c	:	:
Nobreak Command Chr	no	.C2 c		
Center	yes	.CE n		1
Command Line	no	.CL \$options		
Conditional Page	yes *	.CP n		
Double Space	no	.DS	off	
Escape Character	no	.EC c	\	\
Fill	yes	.FI	off	
Footing Margin	no	.FM +n	1	
Footing	no	.FT s	off	blank
Heading	no	.HE s	off	blank
Heading Margin	no	.HM +n	2	
Ignore	no	.IG		
Indent	yes	.IN +n	0	0
Index	no	.IX key, modifier		
Literal	yes	.LI		
Line Length	no	.LL +n	70	
Line Spacing	no	.LS n	1	
Margin, Bottom	no	.MB +n	5	
Margin, Top	no	.MT +n	6	
Noadjust	yes	.NA	off	
Need Lines	yes *	.NE n		
No Fill	yes	.NF	off	
Next Source File	yes	.NX filename		
Omit Page Numbers	no	.OP	off	
Page Advance n	yes	.PA n	1	
Page Length	no	.PL +n	66	
Page Number	no	.PN +n	1	current n
Page Offset	yes	.PO +n	8	0
Paragraph	yes	.PP n	6	last n
Quit Indent	yes	.QI		
Read Insertion	no	.RD s	off	console
Read From File	no	.RF filename	off	
Source File Include	yes	.SO filename		
Space n Lines	yes	.SP n	1	1
Single Space	no	.SS	on	
Tab settings	no	.TA +n {, +n }		
Table of Contents	no	.TC s		
Temporary Indent	yes	.TI +n	0	
Terminal Message	no	.TM s		

* only if page eject occurs

APPENDIX B

TEX RUN-TIME OPTIONS

Option	Effect
Td	Location of TEX source file. The special devices X, Y and Z are not allowed.
Pd	Location of PRN output file. The output file name will be the same as that of the first source file. The special devices X, Y and Z are allowed.
Ed	Destination of error messages resulting from text processing. Only the special devices X and Y may be used. The default for this parameter is the console (X) device.
Xd	Location of the IX index and table of contents output file. The name and default location are those of the source file. The special device Z suppresses this output. The X and Y devices are not allowed.
S	Stop after each page until a console key is pressed.
F	Place form-feed characters in output.
C	Chaining control. When a .NX command is encountered in the source text the query: "Continue (Y/N) ?" is displayed. A yes response will execute the .NX command while any other response will not.
D	Diablo printer. This option enables the Diablo 1640/1650 formatting commands. It should also be used if the source text contains special non-printing control characters.
Q	When it follows the D option in the command line, the Q option initiates proportional spaced printing (equivalent to a leading .PS command in the source text).
R f	Read insertions from the file f whenever a .RD (read insertion) command is encountered in the source text. (equivalent to a leading .RF command in the source text).
N n	Set initial page number to n. This command is overridden by a Page Number (.PN n) command in the text.
O n,m	Omit output of pages not in the range n to m.

APPENDIX C

SUMMARY OF ERROR MESSAGES

TEX generates error messages to indicate various problems encountered in the input text file. Some of these messages are fatal, i.e., they cause the TEX program to terminate without further processing of the source file. Other messages merely indicate the occurrence of an error or of a problem situation.

The format of all error messages is as follows:

ERROR: error-condition statement

CONTEXT:

text containing error condition

If the error resulted from an invalid command line (TEX invocation) the original command line is echoed up to the point of the error and followed by a "?". If the error is a fatal error, an additional statement prints on the error device:

Tex Aborted

The error-condition statement will vary according to the error. See the following page for a summary of these conditions.

The text following the CONTEXT: designation is printed to give the user as much information as possible for locating the problem area. The error generally is found at the end of this segment of text.

ERROR-CONDITION STATEMENTS: -----	EXPLANATION / CAUSE: -----	ACTION TO CORRECT: -----
File not found	Invalid or missing source file name.	Correct source file name.
Invalid parameter	Invalid character in parameter string.	Correct parameter (see Section 1.1).
Invalid file	Invalid file format. Filename parsing error.	Use a valid CP/M file name.
Disk Write	Cannot write output file from memory to disk.	Disk or disk directory full; condense disk or transfer files.
Word too long	Usually, a long word cannot be formatted successfully into a short line length.	Reword text or increase line length.
Undefined ?? Command	Invalid ?? command after a "." or "`" at the beginning of input line.	Correct command; check for accidental "." or "``" at the beginning of line.
HM COMMAND	Heading margin value greater than or equal to top margin.	Correct HM or MT value; if changing both, reset MT value first. (TEX resets HM=MT-1)
MB COMMAND	Bottom margin greater than or equal to PL value - MT value.	Correct MB or PL value; if changing both, reset PL value first.
MT COMMAND	Top margin greater than or equal to PL value - MB value.	Correct MT or PL value; if changing both, reset PL value first.
PP COMMAND	Paragraph indent value greater than line length.	Correct PP or LL value.
?? COMMAND	The command ?? was used at an invalid time or with an argument out of range.	Read command documentation. Try changing command argument value first.

APPENDIX D

DIABLO 1640/1650 WORD PROCESSING COMMANDS

The Diablo commands require the use of the \$D (Diablo) option or they will be ignored. An overstrike capability is provided for output devices which support the backspace character. Superscripts and subscripts can be produced on the Diablo. Both these features use escape sequences documented in Section 1.2. In addition the following features are supported on the Diablo 1640/1650 with the Xerox Xmem2 PCB assembly:

Proportional Spacing

Shadow Print

Bold Overprint

Auto Underline

SUMMARY OF DIABLO OPTION TEX COMMANDS

Function -----	Break? -----	Command Format -----	Initial Value -----	Default Value -----
Bold Overprint	no	.BO n	off	1
Character Size	no	.CS <u>n</u>	0	current
Proportional Spacing	yes	.PS n	off	0
Quit Bold Overprint	no	.QB		
Quit Proportional	yes	.QP		
Quit Shadow Print	no	.QS		
Quit Auto Underline	no	.QU		
Shadow Print	no	.SH n	off	0
Auto Underline	no	.UL n	off	1

1. Proportional Spacing

.PS Start Proportional Spacing

.QP Quit Proportional Spacing

Proportional spacing requires the use of a Proportional Space print wheel on the Diablo 1640/1650. The **.PS** command initiates proportional spacing, overriding the setting of the PROP SPACE switch on the 1640/1650 Operator Control Panel. Proportional spacing may be turned off with the **.QP** command. The Literal (**.LI**) command temporarily turns off proportional spacing and sets a fixed character spacing of 16/120", sufficient for the widest characters on the proportional space print wheel. This preserves columnar alignment for the printing of tabular material. Exiting the Literal mode restarts proportional spacing.

This command is equivalent to the \$Q option.

2. Character Size

.CS Character Size

Forms of the **.CS** command are:

(a) **.CS**

(b) **.CS n**

Form (a) leaves the current character size unchanged. Form (b) reduces the character size by n%. This is useful when the document being prepared is destined for the Diablo. The Diablo (**\$D**) option and the Proportional Spacing (**\$Q**) option reduce character size by 5% and 20% respectively. This command allows these defaults to be preset to insure that the document paging will not change when the document is printed on the Diablo.

3. Shadow Print

.SH Start Shadow Print

.QS Quit Shadow Print

Characters printed after the .SH command are struck twice with 1/120" carriage and ribbon movement between strikes. Increased carriage settling time (.CS) may provide improved Shadow Print quality. The Shadow Print mode continues until the end of the line unless a Quit Shadow Print (.QS) command is used. When Shadow Print is used in conjunction with the Center (.CE) command, the .SH command should follow the .CE <n> command. Forms of the .SH command are:

- (a) .SH
- (b) .SH n

In non-proportional spacing mode the argument n sets the HMI (Horizontal Motion Index). The HMI controls the horizontal spacing of characters. The HMI represents the number of 1/120" increments used for each character; thus the HMI corresponding to a pitch of 10 characters per inch is 12. If the argument n is omitted or is less than 9 the HMI is set by the PITCH switch on the Diablo 1640/1650 control panel.

In Proportional Spacing (.PS) mode the HMI is ignored and an offset of 3 is added to the table of character widths to compensate for the carriage movement between characters.

INPUT

.LL 50

This is an example of

.SH

TEX 1.2

.QS

shadow printing:

.SP

.SH 15

Shadow printing with an HMI of 15.

.BR

.SH 10

Shadow printing with an HMI of 10.

.SP

Note that shadow printing stops at the end of the line.

OUTPUT

This is an example of **TEX 1.2 shadow printing**:

Shadow printing with an HMI of 15.

Shadow printing with an HMI of 10.

Note that shadow printing stops at the end of the line.

4. Bold Overprint

.BO Start Bold Overprint

.QB Quit Bold Overprint

Characters printed after the .BO command are struck twice with no intervening carriage motion and normal ribbon movement between strikes. Bold Overprint and Shadow Print are mutually exclusive modes. When used in conjunction with centering the .BO command should follow the .CE <n> command. When Bold Overprint is used in Proportional Spacing (.PS) mode an offset of 2 is added to the table of character widths to improve document appearance. Forms of the .BO command are:

- (a) .BO
- (b) .BO n

When form (a) is used the Bold Overprint mode continues until the end of the line or a Quit Bold Overprint command.

When form (b) is used the Bold Overprint mode continues for n lines or until a Quit Bold Overprint command, which ever comes first. If several words within a sentence are to be overprinted it is wise to use the command .BO 2 and place a .QB command after the last overprinted word. This insures that the desired text will be overprinted even if some of the text falls on the next line.

INPUT

.LL50

.CE

.BO

.UL

Bold Overprint Example

.SP

This is an example of the use of the

.BO 2

Bold Overprint

.QB

command.

OUTPUT

Bold Overprint Example

This is an example of the use of the Bold Overprint command.

5. Auto Underline

.UL Start Auto Underline

.QU Quit Auto Underline

Automatic underlining is initiated by the .UL command, causing the current carriage position to be stored in the Diablo memory. When the end of the line being printed or a Quit Auto Underline (.QU) command is encountered the text between the starting position and current position is underlined. When this command is used in conjunction with either the Center, Bold Overprint or Shadow Print commands, it should be placed immediately before the text to be underlined as the presence of other commands may introduce spaces between the starting and ending markers for underlining. Similarly, the Quit Auto Underline command should always be placed on the next line following the text to be underlined. Forms of the .UL command are:

- (a) .UL
- (b) .UL n

When form (a) is used TEX will underline text following the .UL command until the end of the line or a Quit Auto Underline command.

When form (b) is used TEX will underline the next n lines of output text unless a Quit Auto Underline command is encountered first. If there is some doubt about whether the text to be underlined will fit on one line the command .UL 2 should be used in conjunction with the Quit Auto Underline command. This insures that the desired text will be underlined even if some text falls on the next line.

INPUT

.LL50

.CE 2

.BO 2

.UL 2

Auto Underline

Example

.SP

This is an example of the use of the

.UL 2

Auto Underline

.QU

command.

OUTPUT

Auto Underline

Example

This is an example of the use of the Auto Underline command.

* .FT string *

FOOTING

The FT command allows the user to define a footing to be printed at the bottom of subsequent pages. The footing is placed one double space above the page number and requires a minimum bottom margin of six lines. Once a footing is defined, it will print on every page following the command, until the footing is changed or eliminated. A footing can be changed by simply entering another FT command to redefine the footing. A footing is eliminated, or nullified, by typing an empty FT command into the source file (see form (b) below).

Valid formats for the FT command are as follows:

(a) .FT string

(b) .FT

where "string" is any character string, not to exceed the line length set by the LL command. One space must separate ".FT" and the heading string. All character positions (characters, blanks, etc.) following this blank, and preceding the carriage return, are considered to be the contents of the footing. If the footing exceeds the current line length, it will be truncated (from the right) to the current line length. Form (b) sets the footing to blank.

NOTE: The left margin set by a PO command remains in effect when printing the footing; however, other indenting values are ignored.

The following is an example showing a footing definition for a technical manual:

.mt -3

.mb +3

.ft (All Information Contained Herein is Proprietary to ABC Co.)

.bp