RSTS/E
RUNOFF User's Guide

Order No. DEC-11-URUNA-B-D

AA-3337B-TC

digital **software**

RSTS/E
RUNOFF User's Guide

Order No. DEC-11-URUNA-B-D

AA-3337B-TC

The postage prepaid READER'S COMMENTS form on the last page of this
document requests the user's critical evaluation to assist us in pre-
paring future documentation.

The following are trademarks of Digital Equipment Corporation:

DIGITAL	DECsystem-10	MASSBUS
DEC	DECtape	OMNIBUS
PDP	DIBOL	OS/8
DECUS	EDUSYSTEM	PHA
UNIBUS	FLIP CHIP	RSTS
COMPUTER LABS	FOCAL	RSX
COMTEX	INDAC	TYPESET-8
DDT	LAB-8	TYPESET-10
DECCOMM		TYPESET-11

2/77-17

CONTENTS

PREFACE

This updated manual describes RUNOFF, a system library program
distributed with the RSTS/E Version V06A system. RUNOFF is a
formatting program that aids in the production of documents. It is
recommended that the user refer to the RSTS-11 System User's Guide and
the RSTS/E Documentation Directory for further information about the
system.

Section 1 is an introduction and explanation of the program for the
inexperienced RUNOFF user.

Section 2 provides a step-by-step guide for executing RUNOFF,
including input and output file specification.

Section 3 lists the format for I/O files, commands, case information,
and some optional modes of operation.

Section 4 lists all RUNOFF commands and their abbreviations. This
chapter can serve as a reference for the experienced RUNOFF user.
There are four categories of commands: page formatting, text
formatting, mode setting, and parameter setting. In each command
specification the function, format, initial or default setting,
duration, and disabling command (if there is one) are described.

Section 5 describes RUNOFF error processing and the errors which can
be encountered.

Provided in the standard system library are two files: RUNOFF.RNO and
RUNOFF.DOC. The former is an ASCII file containing RUNOFF commands
and textual material. The latter is a RUNOFF output file containing
the formatted text. The user can compare the listings of these two
files to see the effect of various RUNOFF commands.

1.0 INTRODUCTION

RUNOFF is a BASIC-PLUS program implemented on the RSTS/E system to aid
in the preparation of textual materials. The user can prepare
material on any RSTS/E terminal, and write it into a file using the
EDIT program to insert text. The file contains textual material along
with RUNOFF commands which are denoted by a period at the beginning of
the line. These commands contain all case and formatting
instructions. RUNOFF then reproduces the file according to command
specifications, performing all formatting and case shifting,
justification, page numbering and titling, indexing and footnoting as
directed. RUNOFF commands do not, of course, appear in the final
output.

Files prepared for use with RUNOFF can be edited and corrected easily.
Material can be added or deleted, and unchanged material need not be
retyped. After a set of changes is made, the program can be executed
to produce a new copy which is properly paged and formatted.
Documentation can thus be updated as necessary without extensive
retyping.

2.0 ON-LINE OPERATION

RUNOFF is used after a file has been created containing both textual material and RUNOFF commands. Like most RSTS system programs, RUNOFF is initiated by the "RUN" command to the monitor. The user types "RUN $RUNOFF." The program first responds with its title and version number, and then asks:

> Source File?

The source file designation (device name, file name and extension) must then be typed. The default extension for RUNOFF source files is .RNO. The default input device is the keyboard (KB:). If the source file designation is typed terminated by pressing ESCAPE (or ALTMODE), a default output file with the same file name and .DOC extension is assumed, underscoring will be done in SEPARATE mode, form feed characters will not be simulated, and there will be no pause. If a carriage return terminates the source designation, the program then asks:

> Output File?

The output file designation (device name, file name and extension) must be typed. The default output device is the keyboard (KB:), which may be limited in capabilities (e.g., upper-case only). Alternately, a disk file name or other device and file name may be given, and the final image will be written into it. The default extension for RUNOFF output files is .DOC.

An optional switch can be appended to either the source or output file designation to QUEue the output file for printing. The switch name is QUE (just Q is sufficient); it is separated from the file designation by a slant (/) character. The value of the switch, separated from the name by a colon (:) or equals (=) character is the line printer designator (LPn:) and switches as in a QUE command (see the <u>RSTS-11 System User's Guide</u>). For example,

> <u>Output File?</u> MONTH.RPT/QUE:LP1:/AFTER:18:00

A second optional switch can be appended to the output file designation. The /CONVERT switch causes certain characters in the output file to be converted to different characters. The switch name is CONVERT (just C is sufficient); it is separated from the output designation by a slant (/) character. The value of the switch, separated from the name by a colon (:) or equals (=) character, is the specification of the file containing the conversion information. For example,

> <u>Output File ?</u> ABC/CONVERT=CNVRT.OUT

As a result, RUNOFF reads the file CNVRT.OUT on the current user account in the public structure to obtain the conversion information.

The conversion file must be an ASCII file with one or more lines, each terminated by a RETURN character. Each line must contain one and only one pair of character codes; the codes are separated by a comma. The codes are the decimal values of the ASCII characters involved in the conversion. The first code of a pair denotes the character in the output file which RUNOFF is to convert. The second code of a pair is the character to be substituted in the output file for every occurrence of the specified character.

The /CONVERT switch is valuable if the RUNOFF output device does not have a special character required in the text. For example, if the output device has no opening and closing brackets ([and]), the following two ·lines in the conversion file cause a substitution.

        91,32
        93,32

RUNOFF substitutes the space character (decimal value 32) in the output for every occurrence of the opening and closing brackets (decimal values 91 and 93). ·After output to the hard copy device, the desired characters can be manually inserted in the space generated.

When the output file designation has been entered, the program asks:

        Underscore (B, L, S or N)?

Answer L (LINE) for a line printer with underlining or for files that will eventually be copied to such a line printer. This causes underscoring (where specified by the ampersand (&) character) to be performed by overprinting an underline. B (BACKSPACE) is used for a terminal such as a 2741 Selectric typewriter. Underscoring is performed by outputting a backspace-then-underline following each underscored character. S (FOLLOWING) causes underscoring to be performed by minus-sign (-) characters on a following line. N (NO) causes underscoring requests to be disregarded. NO is the default condition if this question is not answered and the carriage is returned.

When the response to the Underscore question is other than N, RUNOFF underlines text where the underscore special character (&) is in effect in the input file. Imbedded spaces, unless they are quoted, are not underlined. The response to the Underscore question can contain a switch to include underlining of imbedded spaces in the text. The switch name is SPACES (just S is sufficient); it is separated from the response by a slant (/) character. For example.

        Underscore (B, L, S or N) ?   L/SPACES

As a result of appending the /SPACES switch, the following source characters in the input file

        ^&<RSTS-11 ^SYSTEM ^USER'S ^GUIDE\&

are printed as follows

        RSTS-11 System User's Guide

In this manner, the blank space between words in a multiple word title can be underlined without having to insert quoted spaces. After the Underscore question is answered, the program next asks the question concerning form feeds.

        Simulate Form Feed (Y or N)?

The user must reply Y or N. NO (N) is normal and means that a form-feed character (control-L) will be used to advance the output to the top of a page. This mode is used for line printers. NO is the default condition. YES (Y) means that the program advances to a new page by outputting the appropriate number of line-feed characters. This must be used with any device that does not have automatic mechanical page advancing. If output is to the terminal (KB:), the program then asks:

       Pause?

A YES (Y) answer instructs the program to stop at the top of each page to allow the paper to be positioned manually. The bell rings if the terminal has one; the user must type carriage return to resume printing. This mode is appropriate for typing the output onto pages of duplicating masters or other non-continuous forms. If this question is not answered and the carriage is returned, no pause is the default condition.

After RUNOFF has completed a job, the program asks for another source file. The user may exit from the program and return to the Monitor by typing either CTRL/Z (control-Z) or CTRL/C (control-C).

For information about other RSTS/E system programs or logout procedures, see the RSTS-11 System User's Guide.

3.0   SOURCE FILE FORMAT

3.1   CONTENTS OF SOURCE FILE

The source file contains the textual material which is to appear on
the final copy along with information specifying formatting. Upper
and lower case information can also be supplied so that copy can be
prepared on the Teletype or other such device which can normally input
only upper case letters. All command information consists of regular
ASCII printing characters so that a listing of the source file can be
examined if the final copy is not exactly as desired.

3.2   RUNOFF COMMAND FORMAT

All material in the source file is taken to be source text except
those lines beginning with a period. A line beginning with a period
is assumed to be a RUNOFF command, and must match one of those listed
in this manual.   The commands provide formatting information and
control various optional modes of operation. Only as many characters
as necessary to uniquely specify a command need be included. One- to
three- character abbreviations are available for most commands.
Numeric arguments can be separated by spaces, tabs, or commas.
Numeric arguments preceded by + or - can be used in CENTER, NUMBER,
LEFT MARGIN, and RIGHT MARGIN commands, to increment or decrement
prior or assumed values.

Multiple commands are permitted on a single line if separated by
semicolons (;).   Comments may follow commands if the comments are
preceded by an exclamation point (!). Neither multiple commands nor
comments are possible on commands with text arguments. Either text or
another command may follow the semi-colon. If a command follows, the
period is a sufficient separator and the semi-colon may be omitted.

3.3   FORMATTING THE TEXT

The text is usually filled and justified as it is processed. That is,
the program FILLs a line by adding successive words from the source
text until one more word would cause the right margin to be exceeded.
The line is then JUSTIFYed by making the word spacings larger until
the last word in the line exactly meets the right margin.   A double
space normally follows a period, exclamation point, question mark,
colon or semi-colon, provided it is immediately preceded by two
alphabetic characters.   This feature can be disabled by a RUNOFF
command.

In fill mode, spaces and carriage returns occurring in the source text
are treated as word separators only. Multiple separators are ignored.

Some commands cause a BREAK in the output. A BREAK means that the
current line is output without justification, and the next word is
positioned at the beginning of the next line. This occurs at the ends
of paragraphs, for example.

Occasionally, the user may wish to reproduce the source text exactly, which is done by disabling filling and justification. The program can also be set to fill but not to justify, in which case the output will be normal except that lines will not be justified to the right margin.

The program advances to new pages as necessary, and places the title (if given) and the page number at the top of each page. The user can explicitly call for a page advance where desired, and can inhibit the occurrence of a page advance within specified material.

3.4 CASE INFORMATION

Specification of case for files prepared on an upper case terminal is done with two characters -- circumflex (^, shift-N) and back-slash (\, shift-L). The appearance of a circumflex causes the letter immediately following to be transmitted in upper case. The appearance of a back-slash causes the letter immediately following to be converted to lower case. Any letter not preceded by one of these characters is transmitted in the current mode. The mode is initially upper case and is changed by the occurrence of two successive case control characters or by the LOWER CASE command. Two circumflexes (^^) cause the mode to be set to upper case, and two back-slashes (\\) cause the mode to be set to lower case. This usage corresponds to the use of the shift and shift-lock keys on a typewriter. The following example (lower case is the current mode) shows the use of the case control characters:

    ^HERE IS A ^SAMPLE ^SENTENCE IN ^^UPPER CASE\\ AND LOWER CASE.

becomes:

    Here is a Sample Sentence in UPPER CASE and lower case.

                              NOTE

            Case conversion takes place only on
            ASCII codes 65 to 90 decimal, that is,
            the upper case letters. Any actual
            lower case letters (codes 97 to 122
            decimal) appearing in the source will be
            transmitted unchanged. If the source is
            prepared on a device such as a Model 37
            Teletype or 2741 Selectric typewriter
            which produces letters of the proper
            case, the mode can be left unchanged
            from its original upper case setting.

3.5 SPECIAL CHARACTERS

3.5.1 Ampersand (&)

This character is used to specify underscoring, and when typed it causes the character following it to be underscored. For example:

&s&o&f&t&w&a&r&e

becomes:

software

in the output.

The appearance of an ampersand preceded by a circumflex (^&) causes the underlining of all following characters except for blanks. (If underlining of blanks is required, the /SPACES switch can be typed with the response to the Underscore question of RUNOFF.) The usage of the circumflex is essentially a shift-lock of the underscoring operation. The appearance of ampersand preceded by back-slash (\&) disables this mode.

For example:

^&PLATO\& was a very ^&wise \&man.

becomes:

PLATO was a very wise man.

3.5.2 Number Sign (#)

As it is occasionally necessary to include spaces in the text which are not word separators, RUNOFF interprets the number sign (#) as a quoted space. It prints as exactly one space in the output, can not be expanded for purposes of justifying the line, and can not be interpreted as the end of a line.

3.5.3 Less-Than (<)*

This character preceding a word capitalizes the entire word up to the first space. This is the same as preceding the word with ^^ and following it with \\.

For example, if the current mode is lower case,

<DIGITAL OF ^MAYNARD, <MA

becomes:

DIGITAL of Maynard, MA

in the output.

---

* This special character is initially disabled. See the descriptions of the FLAGS and NO FLAGS commands in Section 4.4.

3.5.4 Greater-Than (>)*

This special character enters the word immediately following it (up to the first space) into the index, including any case shifts that are included in the text.

3.5.5 Percent (%)*

It is occasionally useful to create special symbols in the output by superimposing one character over another. For example, the slant (/) character superimposed on the equals (=) character creates the not equals symbol (≠). The action is referred to as overstrike and is denoted by the percent character (%) in the RUNOFF input file.

If the FLAGS OVERSTRIKE command is in effect and a percent character appears in the text, overstriking occurs in the output. The character immediately following the percent is superimposed on the character preceding the percent.

The method of overstriking is determined by the response to the Underscore question. (See Section 2.0 for a description of the Underscore question.) RUNOFF can superimpose only one character on any given character. Since RUNOFF treats the underscore character (_) as an overstriking character, a character can not both be underscored and have another character superimposed on it.

For compatibility with previous implementations of RUNOFF, the percent special character is initially disabled. See the descriptions of the FLAGS and NO FLAGS commands in Section 4.4.

3.5.6 Outputting Special Characters

To allow the appearance of the special characters (circumflex, back-slash, ampersand, number sign, less-than, greater-than, or percent in the output, the character underline (_) is used as a quote character.

The character immediately following an underline is transmitted to the output with no formatting effect. The underline itself is thus another case requiring quoting. The following eight cases occur:

```
    _ ^    becomes        ^
    _ \                   \
    _ &                   &
    _ #                   #
    _ <                   <
    _ >                   >
    _ %                   %
    _ _                   _
```

---

* This special character is initially disabled. See the descriptions of the FLAGS and NO FLAGS commands in Section 4.4.

4.0 COMMANDS

The RSTS/E RUNOFF commands are recognized if they occur at the beginning of a line started with a period. Any line in the source file beginning with a period is assumed to contain one of these commands. If it does not, an error diagnostic is typed and the line is ignored. Some commands take one or more decimal arguments. The letter n is used to represent this argument, which in some cases may be negative. These numbers are separated from the command by a space or a tab.

Note that only as many characters as needed to uniquely specify a command need be typed. There are abbreviations for most commands; the abbreviations are listed next to the commands in the sections that follow. Multiple commands are permitted on a single line if separated by a semi-colon (;).

Finally, in reading each command description, note that some commands must appear directly before the line of text to which they apply. For example, the CENTER command applies only to the next line and causes whatever is on that line to be centered. Also note the duration of each command. In most cases, if a command is permanent, the disabling command is listed with it.

The commands implemented in RUNOFF on the RSTS/E system are sufficient to format most documents, memos, or other textual material.

RUNOFF automatically performs two operations to improve the format and readability of text. The first operation is filling a line. RUNOFF extracts words from the source text and creates a line of predefined length in the output text. Words are filled in the output line until the addition of another word causes the line length to be exceeded. In filling a line, RETURN and LINE FEED characters in the source text are discarded and a RETURN and LINE FEED pair are inserted after the last character in the filled line. This filling operation ensures that each line contains as many words as possible.

The second operation is justifying a line. RUNOFF adds blank characters between words of a filled line to align the last word on each line with the right margin of the output text. Justification is practical with filled lines only. RUNOFF, therefore, does not justify lines automatically when filling is disabled.

Some RUNOFF operations are initially enabled or disabled as default conditions each time a new source file is processed. The default condition acts as if one of the RUNOFF commands had been executed from the source file. The defaults can be altered by the presence of the appropriate alternate command. The RUNOFF commands and default conditions (if any) are summarized in Table 4-1.

Table 4-1
Summary of RUNOFF Commands and Defaults

Command	Short Form	Argument	Initial Default	Comments
AUTOPARAGRAPH	.AP	-	.NAP	
BLANK	.B	n	-	If n is omitted, 1 is used.
BREAK	.BR	-	-	
CENTER	.C	n	-	If n is omitted, current paper width is used, (see .PAGE SIZE command.) .CENTRE is also valid.
COMMENT	.COMMENT	ccc...ccc	-	
DATE	.D	-	.ND	
FIGURE	.FG	n	-	
FILL	.F	-	.F	
FIRST TITLE	.FT	-	-	
FLAGS	.FL	name c	-	See NO FLAGS.
FOOTNOTE	.FN	n	-	
HEADER	.HD	x	MIXED	X can be UPPER, LOWER, or MIXED.
INDENT	.I	n	-	
INDEX	.X	iii...iii	-	
JUSTIFY	.J	-	.J	
LEFT MARGIN	.LM	n	0	
LOWER CASE	.LC	-	.UC	
NO AUTOPARAGRAPH	.NAP	-	.NAP	
NO DATE	.ND	-	.ND	
NO FILL	.NF	-	.F	
NO FLAGS	.NFL	name	<,>,%	CAPITALIZE, INDEX, and OVERSTRIKE initially disabled.
NO HEADER	.NHD	-	.HD MIXED	
NO JUSTIFY	.NJ	-	.J	
NO NUMBER	.NNM	-	.NM	

Table 4-1 (cont.)
Summary of RUNOFF Commands and Defaults

Command	Short Form	Argument	Initial Default	Comments
NO PERIOD	.NPR	—	.PR	
NUMBER	.NM	n	.NM	
PAGE	.PG	—	—	
PAGE SIZE	.PS	n,m	58,60	PAPER SIZE also valid.
PARAGRAPH	.P	n,v,t	5,-1,2	
PERIOD	.PR	—	.PR	
PRINT INDEX	.PX	—	—	Must be last command in file.
RIGHT MARGIN	.RM	n	60	
SKIP	.S	n	—	If n is omitted, 1 is used.
SPACING	.SP	n	1	
SUBTITLE	.ST	sss...sss	blank	.SUBTTL also valid.
TAB STOPS	.TS	n,n,...,n	—	If n is omitted, every 8 is used, up to 32 allowed
TEST PAGE	.TP	n	—	
TITLE	.TITLE	ttt...ttt	blank	
UPPER CASE	.UC	—	.UC	

4.1 TEXT FORMATTING COMMANDS

The commands in this section affect text formatting and paging.

BREAK

.BREAK or .BR

Causes a break, i.e., the current line is output with no justification and the next word of the source text is placed at the beginning of the next line.

SKIP

.SKIP n or .S n

Causes a break after which n x (line spacing) lines are left blank. If the skip would leave room for less than two printed lines in the page, the output is advanced to the top of the next page. If n is omitted, its value is assumed to be 1.

BLANK

.BLANK n or .B n

Exactly like SKIP, except that n, rather than n x (line spacing), lines are specified. BLANK is used where space is to be left independent of the line spacing; SKIP is used where the space left should be relative to the size of line spacing. Here, too, the default value of n is 1.

FIGURE

.FIGURE n or .FG n

Like BLANK, except that if less than n lines remain on the current page, the page is advanced and n blank lines are left at the top of the new page. Principally used where it is desired to leave room for a figure to be drawn in manually.

TEST PAGE

.TEST PAGE n or .TP n

Causes a break followed by a conditional page advance. If there are n or more lines remaining on the current page, no advance is made and no lines are skipped; otherwise, the page is advanced. This command should be used to ensure that the following n lines are all placed on the same page.

PAGE

.PAGE or .PG

Causes a break and an unconditional advance to a new page. This command does nothing if the current page is empty. It performs titling and numbering as for automatic page advance.

PARAGRAPH

.PARAGRAPH n,v,t or .P n,v,t

The number n is optional and, if present, sets the number of spaces which paragraphs are to be indented. The initial setting is 5. As with INDENT which follows, paragraphs may be indented to the left of the left margin by using a negative value of n. The parameter v determines vertical spacing between paragraphs. A value of 1 specifies single spacing, 2 specifies double spacing, and so on, up to a maximum of 5. If v is specified as -1 (the original, or default, setting), line spacing between paragraphs is (m+1)/2, where m is the regular line spacing. The number t, if present, causes an automatic TEST PAGE t before beginning the paragraph. This parameter is initially set to 2.

The PARAGRAPH command causes a BREAK, followed by vertical line spacing according to the value of v. The following line is indented according to the value of n.

The parameters set by the PARAGRAPH command retain their values until they are explicitly changed.

FOOTNOTE

.FOOTNOTE n or .FN n

Allocates n x (line spacing) lines at the bottom of the current page for a footnote (1). If insufficient room remains on the current page, space will be allocated at the bottom of the following page. The text for the footnote begins on the line following the command, and it may contain any appropriate commands (e.g., CENTER, SKIP) necessary to format the footnote. The footnote is terminated by a line beginning with an exclamation point (the remainder of which is ignored). The lines delimited by this line and the FOOTNOTE command are put into a buffer to be processed when the output moves to within the designated number of lines from the bottom of the page. If a page has multiple footnotes the allocated space is the sum of the allocations for all footnotes assigned to the page. The user must include his choice of footnote-designating symbols within the text. This command does not cause a break.

The current values of margins, justification, case lock, spacing and fill are saved and restored after processing of footnotes; therefore, a footnote may contain commands which change these parameters, and the effect will be limited to the footnote text.

The actual space taken by the footnote may be more or less than that specified by n. The n merely allocates space and should be the user's best guess. If it is considerably off, the footnote lines may overflow the page, or extra space may be left at the bottom. The user may wish to adjust n after examining a first draft printout.

INDEX

.INDEX iii...iii or .X iii...iii

This command treats the remaining text (iii...iii) on the line as a key word or words and adds it, along with the current page number, to the internal index buffer. The command does not cause a break. It must appear immediately before the item to be indexed. A key word or words may be indexed more than once; all pages are indicated when the index is printed. An INDEX command must accompany each occurrence of the key word or words to appear in the printed index.

---

(1) This is a footnote. This text and the dividing line above it were specified by text and commands following a FOOTNOTE 5 command.

PRINT INDEX

.PRINT INDEX or .PX

Causes a break followed by a printout of the entire contents of the index buffer. Entries are printed in alphabetical order, and are set against the left margin using regular line spacing, except that a blank line is left between entries of different first letters. The number of the first page on which each entry appeared is put on the same line as the entry, beginning at the middle of the line (midway between the left and right margins). Additional page numbers for multiple entries follow, separated by commas. The index buffer is left empty.

The PRINT INDEX command must follow all other commands and text in the source file because any data after the PRINT INDEX is not processed.

The INDEX and PRINT INDEX commands are operational only if the RSTS SORT package is available in the system library. If the SORT package is not available, the PRINT INDEX command results in an error message indicating that the command is unavailable.

INDENT

.INDENT n or .I n

Causes a break and sets the beginning of the next line n spaces to the right of the left margin. The number n, if negative, causes the line to begin to the left of the left margin (useful for numbered paragraphs).

CENTER

.CENTER n or .CENTRE n or .C n

This command causes a break after which the following line in the source file is centered. The centering is over the column n/2, independent of the setting of the left and right margins. If n is missing, it is assumed to be the paper width, initially set to 60 (See PAPER SIZE command).

COMMENT

.COMMENT ccc...ccc

This command is treated as a comment and is otherwise ignored.

4.2 PAGE FORMATTING COMMANDS

The commands in this section control page headings.

TITLE

.TITLE ttt...ttt or .T ttt...ttt

This command treats the remaining text on the line as the title. This text appears at the top of all subsequent pages, at position 0 on the second line with the page number. The title is initially blank.

SUBTITLE

.SUBTITLE sss...sss or .SUBTTL sss...sss or
.ST sss...sss

This command treats the remaining text on the line as the subtitle. This text appears on the line immediately following the title and page number. The subtitle is initially blank. The subtitle is not indented, but may contain leading quoted spaces (#) to achieve the same effect, if desired.

NUMBER

.NUMBER n or .NM n

Turns on page numbering (normal) and, if n is supplied, resets the current page number to n.

NO NUMBER

.NO NUMBER or .NNM

Turns off page numbering (disables the NUMBER command). Pages continue to be counted, so the normal page number appears if numbering is re-enabled.

DATE

.DATE or .D

Prints the current system date right-justified directly under the page number in each page header. If the page number does not appear, the date is not printed. The initial, or default, setting is NO DATE.

NO DATE

.NO DATE or .ND

Suppresses printing of the current system date in the page header. The initial, or default, setting is NO DATE.

HEADER

.HEADER x or .HD x

Enables the printing of headers. Sets the word "page" in the page header to either all upper case (PAGE), all lower case (page), or mixed (Page). Thus, x can have one of three values: UPPER, LOWER, or MIXED. The initial, or default, setting is MIXED.

NO HEADER

.NO HEADER or .NHD

Stops the printing of page headers; namely, no title, subtitle, page number, or spacing appear at the top of the page. Thus, the first line of every page will always be blank.

FIRST TITLE

.FIRST TITLE or .FT

Causes the page header with TITLE and SUBTITLE to be printed at the top of the first page. This command must precede any text.

4.3 MODE SETTING COMMANDS

Mode setting commands affect document appearance, i.e., case, filling, justifying. The effect of these commands continues until reset by the corresponding opposite commands.

LOWER CASE

.LOWER CASE or .LC

Sets the case lock to lower case. Equivalent to \\. All letters are output as lower case unless a case shift is indicated.

UPPER CASE

.UPPER CASE or .UC

Sets the case lock to upper case. Equivalent to ^^. Any upper case letters in the source file are output as upper case; this is the initial setting.

NOTE

Since case control applies only to upper case characters in the source file (ASCII codes 65 to 90 decimal), UPPER CASE mode causes the text to be reproduced as it is in the source file. This is especially useful if the source is created on a terminal that can input both upper and lower case characters.

PERIOD

.PERIOD or .PR

Causes insertion of at least two spaces after certain punctuation characters (period, colon, semi-colon, exclamation point, and question mark) in the text. This is the initial mode.

NO PERIOD

.NO PERIOD or .PR

Disables the insertion of the extra space after the punctuation characters.

JUSTIFY

.JUSTIFY or .J

Causes a break and sets subsequent output lines to be justified. This is the initial setting.

NO JUSTIFY

.NO JUSTIFY or .NJ

Causes a break and prevents justification of subsequent output lines. Disables JUSTIFY mode.

FILL

.FILL or .F

Causes a break and specifies that subsequent output lines be filled. Sets the justification mode to that specified by the last appearance of JUSTIFY or NO JUSTIFY. This is the initial setting.

NO FILL

```
.NO FILL or .NF
```

Causes a break and prevents filling of subsequent
output lines. Also turns off justification.

NOTE

The nofill-nojustify mode need be used only
where there are several lines of material to
be copied exactly. A single line example
will not require using these commands if
there are breaks before and after.

Normally FILL and NO FILL are used to turn
both filling and justification on and off.
It is usually desirable to do both. A
subsequent appearance of a justification
command will override the fill command,
however.

Because of the action of FILL, a single
occurrence of NO JUSTIFY will cause the
remainder of the file to be unjustified, with
filling as specified. In order to justify
but not fill (not recommended), a JUSTIFY
command must follow any NO FILL command.

AUTOPARAGRAPH

```
.AUTOPARAGRAPH or .AP
```

Causes any line that starts with a space, including
totally blank lines, to be considered the start of a
new paragraph. This allows normally typed text to
create justified text without special commands. A
blank line followed by a command is not considered to
begin a paragraph. This mode is initially turned off.

NO AUTOPARAGRAPH

```
.NO AUTOPARAGRAPH or .NAP
```

Disables the AUTOPARAGRAPH mode. This is the initial
setting.

4.4 PARAMETER SETTING COMMANDS

These commands specify margins, spacing, paper size, and define special flag characters.

PAGE SIZE

.PAGE SIZE n,m or .PAPER SIZE n,m or .PS n,m

Sets the number of lines per page to n. The value of n must be greater than 10; the initial setting is 58. The value of n includes the top margin of 5 lines; the page number and title appear on the second line. The second argument, m, is optional. If present, it sets the paper width in columns (initially 60). It must be greater than the left margin, and the right margin is set to this value as if a RIGHT MARGIN command had also been typed. This command is normally used only at the beginning of a file, but may be used throughout if needed.

LEFT MARGIN

.LEFT MARGIN n or .LM n

Causes a break after which the left margin is set at column n. The initial setting of n is 0. This may be changed to any number less than the right margin, but not less than 0. The amount of any indent plus the left margin must never be less than 0.

RIGHT MARGIN

.RIGHT MARGIN n or .RM n

Causes a break after which the right margin is set at column n. The number n must be greater than the left margin. The initial setting is 60.

The number of characters on a line is equal to or less than the right margin minus the left margin minus any indenting which may be specified. Even if filling has been disabled, lines are not extended past the right margin.

SPACING

.SPACING n or .SP n

Causes a break after which the line spacing is set to n. (1 means single spacing, 2 means double spacing, etc.) Single spacing (n=1) is the initial setting. The value of n must be within the range 1 to 5.

TAB STOPS

.TAB STOPS n...n or .TS n...n

This command clears all previous tab stops and sets new ones in the columns specified. The maximum number of tab stops is 32. The value(s) of n must be greater than zero and in increasing order. These values of n are the positions of tab stops independent of the setting of the left margin, although any that are less than the left margin are ignored. The initial setting of tab stops is the standard hardware setting at eight-column intervals.

Tabs should be used only in lines that are not justified or filled, i.e., where filling is disabled or a break immediately follows. Clearly, the spaces specified by a tab character should not be expanded to justify the line; this defeats the effect of tab formatting. The appearance of a tab in the source text is translated to one or more quoted spaces, the amount necessary to advance to the next tab stop. If a tab appears at a point on the line where no further tab stops have been set, the tab is treated as though it had been a quoted space.

FLAGS

.FLAGS name c or .FL name c

Enables the special character(s) named by <u>name</u> and redefines it as the character c, if c is present. The names of the special characters are:

CONTROL   column 1 command indicator (.)
ENDFOOTNOTE   column 1 end of footnote (!)
UPPERCASE   upper case and case lock (^)
LOWERCASE   lower case and case unlock (\)
UNDERLINE   underlining (&)
SPACE   quoted space (#)
QUOTE   quoting character (_)
CAPITALIZE   capitalize following word (<)
INDEX   following word into index (>)
OVERSTRIKE   superimpose the following character on
               the preceding character (%)
ALL   all but . and ! (column 1 control
               characters)
null   same as ALL

NO FLAGS

.NO FLAGS name or .NFL name

Disables the special character(s) <u>name</u>. Initially, CAPITALIZE, INDEX and OVERSTRIKE are disabled.

5.0 ERROR PROCESSING

RUNOFF denotes errors in the source file by printing messages at the terminal. If an incorrect or unrecognized command appears in the source text, the message gives the page number in which the error exists followed by the text "Unrecognized Command" or "Invalid Command" and a description of the error. When an otherwise valid command contains an incorrect argument, RUNOFF optionally precedes the error message with text describing the error condition. Table 5-1 lists and explains the error conditions.

Table 5-1
RUNOFF Error Conditions

Error Text	Meaning
Can't Justify Line	No blank spaces to be expanded.
Invalid Inside Footnote	A FOOTNOTE command is not allowed within the text of another FOOTNOTE command.
Missing or Invalid Argument	The command requires an argument and does not have one or the argument given is improper. See Table 4-1 for a summary of commands and arguments.
Negative Indent Attempted	A negetive argument in an INDENT command may force the text beyond column 0 (and off the page image). Use the LEFT MARGIN command to shift the lefthand margin further to the right so that the negative indent does not overflow column 0.
'PRINT INDEX' Unavailable	The PRINT INDEX command requires the BASIC-PLUS Sort package and the Sort programs are not in the system library.
Too Many Tab Stops	The TAB STOPS command allows no more than 32 arguments. User must reformat the text.

INDEX

READER'S COMMENTS

NOTE: This form is for document comments only. DIGITAL will
        use comments submitted on this form at the company's
        discretion. Problems with software should be reported
        on a Software Performance Report (SPR) form. If you
        require a written reply and are eligible to receive
        one under SPR service, submit your comments on an SPR
        form.

Did you find errors in this manual? If so, specify by page.

_____
_____
_____
_____
_____
_____

Did you find this manual understandable, usable, and well-organized?
Please make suggestions for improvement.

_____
_____
_____
_____
_____
_____
_____

Is there sufficient documentation on associated system programs
required for use of the software described in this manual? If not,
what material is missing and where should it be placed?

_____
_____
_____
_____
_____
_____

Please indicate the type of user/reader that you most nearly represent.

☐ Assembly language programmer
☐ Higher-level language programmer
☐ Occasional programmer (experienced)
☐ User with little programming experience
☐ Student programmer
☐ Non-programmer interested in computer concepts and capabilities

Name_____ Date_____

Organization_____

Street_____

City_____ State_____ Zip Code_____
                                                         or
                                                      Country