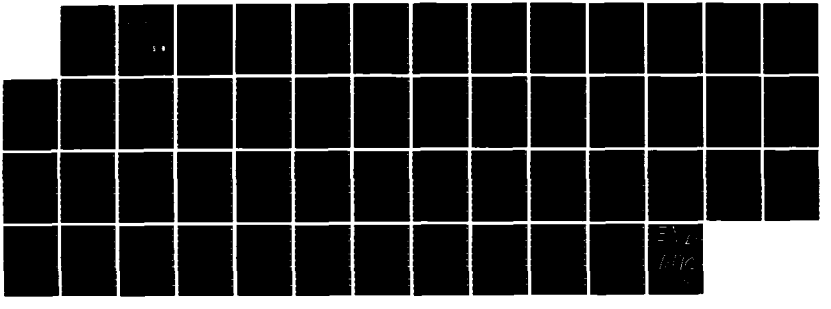
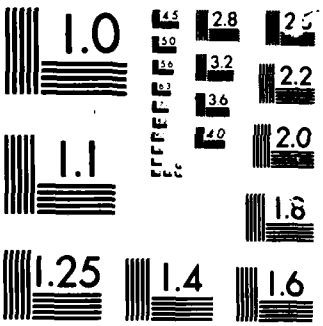
AD-A167 362 *FORM: A TEXT FORMATTING PROGRAM(U) MICHIGAN UNIV ANN ARBOR COMMUNICATIONS AND SIGNAL PROCESSING LAB L C RUSSELL MAR 86 021535-1-M N00014-84-K-0017 1/1

UNCLASSIFIED F/G 9/2 NL





MICROCOPY CHART

Report 021535-1-M

*FORM: A TEXT FORMATTING PROGRAM

L.C. Russell

COMMUNICATIONS AND SIGNAL PROCESSING LABORATORY
Department of Electrical Engineering and Computer Science
The University of Michigan
Ann Arbor, Michigan 48109

March 1986

Technical Memorandum No. 119
Approved for public release; distribution unlimited.

Prepared for
OFFICE OF NAVAL RESEARCH
Department of the Navy
Arlington, Virginia 22217

First Edition .... OCTOBER 1984
Second Edition ... JULY 1985
Third Edition .... MARCH 1986

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS NONE
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) 021535-1-M	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Communications and Signal Processing Laboratory 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION Office of Naval Research Code 4250A
6c. ADDRESS (City, State, and ZIP Code) The University of Michigan Ann Arbor, Michigan 48109	7b. ADDRESS (City, State, and ZIP Code) 800 North Quincy Street Arlington, Virginia 22217
8a. NAME OF FUNDING/SPONSORING ORGANIZATION 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER Contract No. N00014-84-K-0017
8c. ADDRESS (City, State, and ZIP Code)	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. NR 083-490 / WORK UNIT ACCESSION NO.
11. TITLE (Include Security Classification) *FORM: A Text Formatting Program	
12. PERSONAL AUTHOR(S) Russell, Linda C.	
13a. TYPE OF REPORT Tech. Memorandum 13b. TIME COVERED FROM 3/84 TO 3/86	14. DATE OF REPORT (Year, Month, Day) March 1986 15. PAGE COUNT 50
16. SUPPLEMENTARY NOTATION	

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Word Processor Text Formatting

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

The program *FORM is a text formatting program which prints the contents of PDS text files in paginated form. It is designed to provide great flexibility to the user. The user can have very little understanding of *FORM and yet still produce suitably formatted text.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☐ UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION Unclassified
22a. NAME OF RESPONSIBLE INDIVIDUAL Carol S. Van Aken	22b. TELEPHONE (Include Area Code) (313) 764-5220 22c. OFFICE SYMBOL

DD FORM 1473, 84 MAR 83 APR edition may be used until exhausted. All other editions are obsolete.

TABLE OF CONTENTS

THE *FORM PROGRAM

The program *FORM is a text formatting program which prints the contents of PDS text files in paginated form. It is designed to provide great flexibility to the user. The user can have very little understanding of the capabilities of *FORM and yet still produce suitably formatted text. On the other hand, very personalized output can be obtained by exploiting the full potential of *FORM.

The user has control of the following features (this is by no means an exhaustive list):

MARGINS (for top,bottom,left,right),
LENGTH AND WIDTH OF PAGE,
PAGE NUMBERING,
LINE SPACING,
HEADERS AND FOOTERS,
FILLING OF TEXT,
ADJUSTING OF TEXT,
DIFFERENT SPECIFICATIONS FOR ODD/EVEN PAGES,
FIGURES,
LEADERS AND TRAILERS,
PRINTING OF PAGES,
FORMATTING EQUATIONS,
NUMBERING EQUATIONS
CHARACTER STRING SUBSTITUTION
NUMERICAL VARIABLE SUBSTITUTION
USER DEFINED COMMANDS
REPEATED SOURCE INPUT,
PRINTING ON THE MTS XEROX 9700 PAGE PRINTER
CHARACTER FONTS ON XEROX 9700 AND EPSON FX-80
OUTPUTTING TO LINC TAPE FOR LATER PRINTING,

The first thing a user (especially a novice user) should do is select one of the 4 internal formats. These

correspond to the following: Cooley Report Format, Correspondence Format, Documentation Format, and Rackham Dissertation Format. The default format is the Documentation Format. When an internal format is selected all parameters are set appropriately and the user is assured of at least producing output suitable for the intended use even if *FORM is given no additional commands. To select a format see the description of the command .FMT in the section titled COMMANDS. Also see Appendix A.

The program is invoked using the command

```
RUN *FORM FILE1+FILE2+...+FILE16 [P=PARAMETERS]
```

An integer number in the parameter string indicates on which output text page printing is to start. This is useful for replacing spoiled pages. (Default for this is page number 1)

The control parameters recognized are:

O:	Print only odd pages
E:	Print only even pages
S:	Halt at the top of each page and wait until any key is hit. This allows for manual paper feed.
V:	Print version ID only, do not execute
D:	Printing device = Decwriter
H:	Printing device = Hytype
T:	Printing device = Paper Tiger (on Port #2)
P:	Printing device = Epson FX-80 (on Port #2)
N:	Indicates use of narrow paper.
M:	Output is to go on magnetic tape for printing on the Xerox 9700. "P=M" cannot be used on the MP12.
L:	Output is to go on linc tape on unit 1 for

later printing on the Xerox 9700, or listing using the program *RFORM. Output will be written onto unit 1 linc tape starting at block 0. Any information on this tape will be written over. For a description of the use of the program *RFORM see the section in this document entitled *RFORM. "P=L" cannot be used on the MP12.

If no printing device is supplied in the command line the pages are vertically expanded by 2 . (In other words an 11 inch page will cover 22 inches of paper.)

The following characters receive special treatment when found in text:

"."	When immediately preceded by a carriage return, a period (.) indicates a command line.
CTRL-R	Repeat following sequence (see section on equations for a complete description)
CTRL-U	Up (see section on equations)
CTRL-D	Down (see section on equations)
CTRL-F	Forward (see section on equations)
CTRL-B	Back (see section on equations)
CTRL-X	Set a mark. See section on equations and description of CTRL-E below.
CTRL-E	Return to location of mark within the word or equation. If no mark has been made return to start of word or equation. See section on equations.
CTRL-G	Substitute String (see section on string substitution, and description of .SDEF command).
"_"	Non-trivial space (underline character). In equation mode this character always prints as

an underline character. In non-equation mode this character prints as a non-trivial space unless CTRL-B has been performed to get to its position, in which case it prints as an underline.

"-" Line continuation character (hyphen). If next character is a carriage return *FORM ignores it. In equation mode this character is treated as a minus sign and not as a line continuation character. Warning: Unless in equation mode, *FORM will assume that a word can be split between two lines if it contains a hyphen.

CTRL-S Halt character. *FORM will wait for any key to be struck before continuing.

CTRL-P Next character is to be printed using the alternate font type. (See section on font types).

GENERAL INFORMATION:

All command lines must begin in column 1.

If filling is OFF all spaces are non-trivial. If filling is ON only leading spaces are non-trivial (they indicate the start of a new text block).

Any hyphenated word may be "split" at the hyphen. A hyphen at the end of a line indicates that the following carriage return is to be ignored (unless the subsequent line is a command line). In order for a "-" not to be treated as a hyphen enter equation mode.

A new text block is started upon entry to the program or whenever a text block is terminated. A text block is terminated by:

1) an empty or blank line
2) first character on next line is a space
3) any command which causes a break.

If filling is ON, the current output line is not adjusted if the block ends. If this line is null it is discarded. Output lines containing only one word are not adjusted.

A "word" is terminated by a trivial space or a carriage return. When filling is ON and a "word" ends with any of the characters:

".", "?", "!", ";", or ":";
or with ".", "?", or "!" followed by """ or ")",

2 blanks will precede the immediately following word (if it is placed on the same output line).

To adjust text to fit between the defined boundaries extra spaces are distributed as uniformly as possible between the words. To present a more balanced appearance the adjusting is started alternately at the right and left boundaries as successive lines are processed.

COMMANDS:

*FORM provides the user with the following commands (see Appendix D for a summary of the commands provided):

.SET — Sets user defined labels or program parameters. User labels must already have been defined by the .DEF command. When specifying parameters in inches the value of "lines per inch" (LPI) or "characters per inch" (CPI) should already have been set (otherwise program uses default values - default for CPI is 10; default for LPI is 6). The .SET command will accept many parameters to accuracy of one tenth (unless the value is in inches it will be rounded to the half integer as follows: 1.4 → 1.0, 1.5 → 1.5, 1.6 → 2.0, etc.) Inch specifications will not be rounded off. (Caution: Certain parameters, by their definition, must be in terms of integers. Any attempt to define them with a non-integer quantity will result in truncating of that quantity.) .SET can also be used to specify special characters. .SET does not cause a break. Program parameters are updated following the printing of the next line whereas user defined parameters are updated immediately.

EXAMPLE: The following example sets the even page left margin (ELM) to 1.5 inches, sets the odd page left margin to be equal to the even page left margin, increases the page number (PN) by 5, makes '*' the non-trivial space character (NTS), and makes

the line spacing 1.

```
.SET ELM="1.5 OLM=ELM PN=PN+5 NTS='*' LSP=1
```

See Appendix B for a complete listing of the parameters which can be set using the .SET command.

.SSET — Similar to the .SET command above except .SSET is used to set defined character strings to other defined character strings.

EXAMPLE: The following example sets the string A to the string EXP and sets the string B to the string EXP1.

```
.SSET A=EXP B=EXP1
```

All strings must have been previously defined by the use of the .SDEF command. (See description below). This command does not cause a break.

.DEF — Defines user-defined numerical parameters with symbol names of up to 8 characters in length. These may be defined in terms of expressions involving integers, single character strings, program parameters or previously defined user-parameters. A user-defined parameter must first be defined before it can appear in a .SET command. This command does not cause a break.

EXAMPLE:

```
.DEF A=10 B=HTT+TBF C='d'
```

.SET PTH=B-A B=LSP NTS=C

.SDEF n — Defines a character string. The next line contains the name (up to 8 characters in length) of the string. The n lines after this contain the string. (Default for n is 1). Carriage returns within the string are not ignored. Later in the text whenever the string's name flanked by CNTRL-G's appears the defined string will be substituted in. The string may contain any special characters. See the section "string substitution". This command does not cause a break.

.C — Comment line. Not printed.

.BR — Causes a text break. Terminates a text block (such as a paragraph). If last line of text block is non-empty it is printed out unadjusted.

.A — Causes text to be adjusted (spread) to fit between indentation limits. Filling is turned on. Causes a break (see description of .BR) .-A is same as .RR. Most of the text on this page is an example of adjusted text.

.RR — Use ragged right. Turns filling off. Text is printed as it appears starting at the left hand indentation. The right hand limit is ignored (except for page boundary). If filling is turned back on text is contained within the indentation limits. Causes a break. .-RR is same as

.A.

This is an example of ragged right.

.RL Use ragged left. Turns filling off. Lines are aligned to the right hand indentation limit. If line is too long it will overflow the right hand margin. If filling is turned back on text is contained within the indentation limits and is right justified. Causes a break. .-RL is same as .A.

This is an example of ragged left.

.CE Centering is turned on. Causes filling to be turned off. Subsequent lines are printed centered within the existing indentation limits. If the line is too long it will overflow into the right hand margin. Causes a break. .-CE is same as .A.

This is an example of centering.

.F Turns filling on. When filling is ON *FORM ignores carriage returns within a block of text. When filling is OFF all carriage returns supplied in the input file are honored. This command does not change the justification mode (A, CE, RL, or RR). It does cause a break. .-F turns filling off.

.PA Terminates current page. Any footers or figures are printed. Causes a break. Does not cause new page to be started. Does

nothing if "between" pages.

.HT 'left'center'right'	Set header for all pages. Text between 1st and 2nd string delimiters is left justified. Text between 2nd and 3rd string delimiters is centered. Text between 3rd and 4th string delimiters is right justified. A "#" is substituted by the current page number. The string delimiter can be any non-alphanumeric character. The only special control characters which are allowed in headers and footers are ^P, ^O, ^R, ^G, and ^S. Not allowed are ^F, ^B, ^U, ^D, ^X, and ^E.
.FT 'left'center'right'	Set footer for all pages. See description of .HT.
.EHT 'left'center'right'	Set header for even pages only. See description of .HT.
.EFT 'left'center'right'	Set footer for even pages only. See description of .HT.
.ODP	Advance page numbering to next odd page. Causes a page termination and break.

.RNP — All page numbering will be in Roman numerals. .-RNP turns Roman numeral page numbering off.

The following commands perform their functions on the next n input lines following the command:

.BL n — Causes n blank text line spacings (defined by LSP) to be produced. n can also be specified in terms of inches. Causes a break. If there are n or fewer lines remaining on the current page, causes the page to be terminated. Default: n=1.

.VT n — Causes a vertical tab to line n. Causes a break. If current line is past line n, has no effect other than causing the break.

.UL n — Enter underline mode for next n lines. *FORM only underlines printed text and non-trivial spaces, which are not super-scripted or sub-scripted.

.EQ n — Enter equation mode for next n lines. The next n lines will be treated as one word. (See section "EQUATIONS").

.ENUM n — For use when an equation is to appear alone on its own line with an equation number lined up against the right margin. The following line will contain the form of the equation number typed exactly as it is to appear. The n lines after this contain the equation (see the section: Equations). Default for n is 1. This command causes a break both before and after the equation.

The equation number may not contain any special carriage control characters (^F, ^B, ^U, ^D, ^E, or ^X).

.FIT n — If at least n printing lines fit on the current page, continue processing text. Otherwise, terminate the current page printing any footers, start a new page and continue processing text. n may also be supplied in inches, such as: .FIT "2. .FIT does not cause a break.

.FIG op m n — Save the next n lines for use in making up a figure. (Default for n is 1). The figure may contain special control characters, commands, string substitutions, etc. just like regular text. The value m is the minimum expected vertical extent of the printed figure. The user must supply this. m may be either in absolute line spacing counts or in inches. m is only important if the figure is to appear at the bottom of the page. Anywhere else on the page the figure is printed using as little space as possible regardless of the value given for m. There are 4 choices for the option "op": T, B, I, or none. If op=T the figure is printed at the top of the next page. If op=B the figure is printed as soon as possible at the very bottom of a page. If op=I the figure is printed immediately if there is room, otherwise it is printed at the top of the next page. If no op is specified (op=none) the figure is printed as soon as possible either at the top or the bottom of a page. No more than

two figures may be queued at once. They will be printed out in the same order in which they were saved off, i.e. the second figure will never print until after the first figure has printed even if the second figure was given an op of I. This command does not cause a break. The printing of the figure does not cause a break either. The values of the program parameters which control the printing of the figure are those which are in effect when the figure is printed, not those in effect when the figure command was given.

.DEBUG n — Turn debugging on for next n lines. See section on Error Messages.

.LHT m n — The following n input lines are to be used as left hand leader text of width m in columns LM+LI-m-2 thru LM+LI-2. Margin limits are ignored, page width limits are not. Causes a break. Text is fit between these columns as best can using ragged right. If a break occurs before leader is finished main text waits until leader is completely printed. The only special control characters which are allowed in leaders are ^P, ^S, ^R, and ^G. Not allowed are ^U, ^D, ^F, ^B, ^E, and ^X.

.RHT m n — The following n input lines are to be used as right hand trailer text of width m in columns LM+LW-RI+2 thru LM+LW-RI+2+m. Margin limits are ignored; page widths are not. Causes a break. Text is fit between these columns as best can using ragged

right. If a break occurs before trailer is finished main text waits until trailer is completely printed. The only special control characters which are allowed in trailers are ^P, ^S, ^R, and ^G. Not allowed are ^U, ^D, ^F, ^B, ^E, and ^X.

.FMT m — Sets Format. "m"=1 corresponds to Cooley Report Format, "m"=2 corresponds to Correspondence Format, "m"=3 corresponds to Documentation Format, "m"=4 corresponds to Rackham Dissertation Format.

*FORM

AN EXAMPLE:

The following is an example of how one can use *FORM to produce a desired output. The desired output is section 4 of the PDS manual (to make it fancier I underlined the heading). Comments follow the **s.

```
.FMT 3                          ** Selects documentation format
.HT 'DRAFT 1-25-77'''           ** Sets the top of page header
.FT '''PDS MANUAL 4-#'          ** Sets the bottom of page footer
.EFT 'PDS MANUAL 4-#'''         ** Sets the footer for even pages
.CE                             ** Turns line centering on
.UL                             ** Turns underlining on for 1 line
4. The Symbolic Assembler
.BL 3                           ** Leaves 3 blank lines
.A                              ** Turns centering off, adjusting on
 The assembler is used to convert programs
written in assembly language into a form
which can be loaded into the computer.
The input to the assembler comes from PDS text
files and the output is normally placed
into a PDS file.  The output of the assembler is in
a format which is often referred to as "binary".  The binary
format can be converted into RUN module form using the
PDS program, *LOAD.

 The assembler is placed into execution using a RUN command
of the form:

.CE                             ** Turns centering on
RUN *ASM FILE1+FILE2+FILE3 BINARY [p=PARAMETERS]

.A                              ** Turns centering off, adjusting on
 A total of 15 input files can be used to supply the source
input to the assembler (if an output file is not designated,
a total of 16 input files can be used).  The binary output is
written into the system scratch area and is copied into the
permanent output file when the system is re-entered at the
completion of the assembly.

 The control parameters recognized by the assembler are:

.SET LI=12                      ** Sets left indent to 12 spaces
.LHT 6                          ** Next line will be a left hand trailer
L
generate a listing of the assembled program using the
operator's console as the output device

.LHT 7                          ** Left hand trailer of width 7
```

```
*FORM


HL
same as L but use the line printer as the output device (device
address 66)

.LHT 6                          ** Left hand trailer of width 6
P
punch the binary output onto paper tape using the console punch
rather than placing it into an output file
```

The produced output is shown on the following page.

4. The Symbolic Assembler

The assembler is used to convert programs written in assembly language into a form which can be loaded into the computer. The input to the assembler comes from PDS text files and the output is normally placed into a PDS file. The output of the assembler is in a format which is often referred to as "binary". The binary format can be converted into RUN module form using the PDS program, *LOAD.

The assembler is placed into execution using a RUN command of the form:

```
RUN *ASM FILE1+FILE2+FILE3 BINARY [p=PARAMETERS]
```

A total of 15 input files can be used to supply the source input to the assembler (if an output file is not designated, a total of 16 input files can be used). The binary output is written into the system scratch area and is copied into the permanent output file when the system is re-entered at the completion of the assembly.

The control parameters recognized by the assembler are:

L	generate a listing of the assembled program using the operator's console as the output device
HL	same as L but use the line printer as the output device (device address 66)
P	punch the binary output onto paper tape using the console punch rather than placing it into an output file

*FORM

CHARACTER FONTS:

Unless otherwise indicated all characters will be printed in the main character font. In order to select the alternate character font for a particular character the character must be immediately preceded by a CTRL-P.

Character fonts are only valid when printing is to occur on the Xerox 9700 or on the EPSON printers. On the following page is a list of mappings between digraphs represented by a CTRL-P followed immediately by a normal ASCII printing character and the associated symbol to be printed by the 9700, or the EPSON printer. In order to print the alternate character set on the EPSON printer, the printer must be initialized. This entails running the program *EPSON prior to running *FORM. If *EPSON is not run first the alternate characters will be printed as the italic characters which is the default alternate character set for the EPSON printers. A summary of the character sets is presented in Appendix E.

A	∝	proportional sign	a	α	alpha
B	•	big dot	b	β	beta
C	©	copyright sign	c	χ	chi
D	Δ	delta	d	δ	delta
E	∈	element of	e	ε	epsilon
F	Φ	phi	f	φ	phi
G	Γ	gamma	g	γ	gamma
H			h	η	eta
I	∫	integral sign	i	∞	infinity
J			j		
K			k	κ	kappa
L	Λ	lambda	l	λ	lambda
M			m	μ	mu
N	∇	nabla	n	ν	nu
O			o	∂	partial derivative
P	Π	pi	p	π	pi
Q	Θ	theta	q	θ	theta
R	®	registered sign	r	ρ	rho
S	Σ	sigma	s	σ	sigma
T	™	trademark sign	t	τ	tau
U	⊤	upsilon	u	υ	upsilon
V			v		
W	Ω	Omega	w	ω	omega
X	Ξ	xi	x	ξ	xi
Y	Ψ	psi	y	ψ	psi
Z			z	ζ	zeta
0	°		+	∪	set union
1	¹		'		
2	²		-	¬	not sign
3	³		.	∩	set intersection
4	⁴		/	÷	divide sign
5	⁵		;		
6	⁶		<	⊂	contained in
7	⁷		=	≡	identically equal to
8	⁸		>	⊃	contains
9	⁹		?	ℓ	likelihood ratio
@	■	square block	[	{	nicer {
!	√	radical	\		
"	·	dot	]	}	nicer }
#	←	left arrow	^	↑	up arrow
$	→	right arrow	_	↓	down arrow
%	±	plus or minus			
&	∑	summation	{	{	big {
'	°	degree sign	|	¶	paragraph sign
(	(	bigger (	}	}	big }
)	)	bigger)	~	≈	approximately equal to
*	×	times sign			
:					

EQUATION FORMATTING:

If an equation is not trivial or contains spaces or minus signs, the command .EQ should be given to enter equation mode. Once in equation mode everything on the n lines after the .EQ n command will be treated as a single "word". It will not be split up, nor will extra spaces be added. .EQ does not cause a break.

Control characters are defined to indicate cursor motion:

UP:	CTRL-U
DOWN:	CTRL-D
FORWARD:	CTRL-F
BACK:	CTRL-B
SET A MARK:	CTRL-X
RETURN TO MARK:	CTRL-E

Up and down work in half-line increments, forward and back work in full character spaces.

Another control character which is very useful is CTRL-R, the repeat control character. It operates a total of n times on whatever follows it enclosed in apostrophes. Its format is as follows:

CTRL-R n 'whatever is to be repeated'

where n is an integer and the phrase to be repeated may be of any length and contain any equation control characters or alpha-numeric characters. CTRL-R's may occur anywhere in the text and may be nested up to six deep.

Liberal use of marks (CTRL-X's) and return to marks (CTRL-E's) is useful for keeping down the size of the line

buffer to prevent overflows.

Example:

```
The equation tried was
.EQ 2
M^DL^U=C^D^P1^UT^U-5^D(e^U^U-C^D^P2^U/LT^D^D-1)^U-1
^D.
This was found to be very suitable.
```

Produces:

The equation tried was $M_L=C_1T^{-5}(e^{-C_2/LT}-1)^{-1}$.
This was found to be very suitable.

Example:

```
^R5'*'^R4'^D^B*'^R4'^B^B*'^R3'^U^B*'^F^D*
```

Produces:

```
*****
*   *
*****
```

Example:

```
.SDEF 1
EQ1
H(f)e^Uj2^Ppf^Pt

.ENUM 1
(6.4a)
LTI Output:  ^GEQ1^G
```

Produces:

LTI Output: $H(f)e^{j2\pi f\tau}$ (6.4a)

STRINGS SUBSTITUTION:

String substitution is a very powerful yet flexible tool which can be used two ways:

1) To make the insertion and modification of commonly repeated series of keystrokes easier, or to build up user-defined commands (see following section).

2) To substitute the decimal values of user-defined or program defined parameters into the text.

String substitution can occur literally anywhere in the text. Strings may be nested within strings. Whenever CTRL-G appears in the text it indicates a text string substitution. There are two modes (as indicated above):

1) If the CTRL-G is followed by the name of a string which has been already defined by the .SDEF command the character string will be substituted into the text. A second CTRL-G should terminate the string's name.

2) If CTRL-G is followed by an apostrophe (') it indicates a value substitution. The apostrophe should be followed by the name of a user-defined or program defined parameter whose decimal value is to be substituted into the text (see description of the .DEF and .SET commands). As above a second CTRL-G should terminate the parameter's name.

If a string substitution appears in an .SDEF command the string will not be substituted in until it is referenced in the text. This allows for variable strings within strings. One string may be set equal to another through use of the .SSET command.

*FORM

Example:

```
1       .DEF EN=1 E1=1 E2=0 E3=0 E4=0
2       .SDEF
3       EQN
4       (^G'EN^G)
5       .ENUM
6       ^GEQN^G
7       firstequation
8       .SET EN=EN+1 E2=EN
9       .ENUM
10      ^GEQN^G
11      secondequation
12      .SET EN=EN+1 E3=EN
13      .ENUM
14      ^GEQN^G
15      thirdequation
16      Compare equation (^G'E2^G) with equation (^G'E3^G)
```

Produces:

```
firstequation                                                (1)
secondequation                                               (2)
thirdequation                                                (3)
Compare equation (2) with equation (3)
```

If later another equation is to be inserted between equation (1) and equation (2) just insert the following between lines 7 and 8:

```
7.1     .SET EN=EN+1
7.2     .DEF E1A=EN
7.3     .ENUM
7.4     ^GEQN^G
7.5     addedequation
```

The output now looks like:

```
firstequation                                                (1)
addedequation                                                (2)
```

```
secondequation                                    (3)
thirdequation                                     (4)
Compare equation (3) with equation (4)
```

The above simple example is designed just to demonstrate the basic principles of string substitution. Strings are allowed to contain as many lines as desired including command lines (see description of .SDEF command). A long series of commands and text lines could be inserted into the text with a few keystrokes once the block had been defined by .SDEF. The potential power of string substitution is limited only by the user's imagination (and the size of the string buffer).

Any marks made by CTRL-X within a string will be local to that string only. In this manner marks may be nested within nested strings. A CTRL-E within a string will only return to the mark made within that string or to the start of the string if no mark has been made within that string. Only one mark (the most recently defined) will be operational within each string.

*FORM

USER-DEFINED COMMANDS:

User-defined commands can be built up by combining a series of program defined commands (see section: COMMANDS) and text if desired within a defined string. Whenever a period "." is encountered in column 1 *FORM looks up the following name in the command symbol table. If the name can not be found in the command table it looks into the user-defined string table. If the name is found here, the first name from the string is looked up in the symbol tables, etc. Commands may be embedded to any depth.

This has the effect of eliminating the need to flank the strings name by CTRL-G's when the string is to be used for command substitution.

EXAMPLE: Kurt wants to define a command, .KFIG, which will leave 3 inches for an immediate figure, update the current figure number and label the figure. He defines a string as follows:

```
.SDEF 6
KFIG
FIG I "3 5
.BL "3
.SET FIGNUM=FIGNUM+1
.CE
FIG.^G'FIGNUM^G
.A
```

Note that the first command line must be missing the period (since the period is in the text already read).

Now whenever the following is inserted into the text:

.KFIG

a three inch blank labeled figure will be immediately

inserted.

EXAMPLE: Kurt wants to define a command which will leave a blank page without causing a break. He wants it to be called: .BP. He defines a string as follows:

```
.SDEF 2
BP
FIG T "1
.PA
```

Now whenever the command .BP appears in the text a blank page will be left (without causing a break) at the first available opportunity.

THE PROGRAM *RFORM:

The program *RFORM is designed to read *FORM produced output off of linc tape for listing on a selected terminal. The information on the linc tape must have been produced by running the program *FORM with parameter P=L. The linc tape must be mounted on unit 1.

The program is invoked by using the command

RUN *RFORM [P=PARAMETERS]

An integer decimal number in the parameter string causes that number of copies of the document to be produced.

The control parameters recognized are:

O:	Print alternating pages starting with the first.
E:	Print alternating pages starting with the second.
S:	Halt at the top of each page and wait until return is hit. This allows for manual paper feed.
V:	Print version ID only, do not execute.
D:	Printing device = Decwriter
H:	Printing device = Hytype
T:	Printing device = Paper Tiger (on port #2)
P:	Printing device = Epson FX-80 (on port #2)
N:	Indicates use of narrow paper.

If no printing device is supplied in the command line the pages are vertically expanded by 2. (In other words an 11 inch page will cover 22 inches of paper.)

*RFORM is useful when 2 or more copies of a given

document are desired or printing on both sides of the paper is desired. It is also useful if it is desirable to do the slow *FORM processing on one system and printing on another.

ERROR MESSAGES:

Whenever an error occurs an error mesage is printed along with information to locate the error. The information given is the input block number, the input file number, and the line number within the file of the last line read before the error was detected. This information is printed as follows:

```
ERROR LOC IS: BLOCK NUMBER =
               FILE NUMBER =
               LINE NUMBER =
```

All numbers are printed in decimal.

Following is a list of the error messages that *FORM can send, along with an explanation of each:

OVERFLO OCCUR	Indicates an overflow of either the single word buffer or the single line buffer. Number of characters in a single word or a single line is limited to 512 (this includes control chars.)
UNDEFINED OP	Indicates that the following char is not a recognized operator for use in command strings.
SYM TABL FULL	Symbol Table is full. User is allowed 210 user-defined symbols (see description of .DEF command)
STRNG TAB FUL	Substitute string buffer is full. The total number of characters in all the defined strings plus the number of characters in the stack must not exceed 3072.
STACK UNDRFLO	Programming problem. See L. C. Russell.
COMERR TOKN=	Indicates command format is not correct. Value of TOKEN is given. Definitions are as

follows:

```
'+'    : 0001
'-'    : 0002
'EOL'  : 0003    (End Of Line)
'='    : 0004
''''   : 0005
','    : 0006
'W'    : 0007
Any other non-hexidecimal char: 0010

Decimal Number: 7000
Defined Symbol: 4000
Undefined Sym.: 3000
```

If this error message occurs check command for correct structure.

ILLEGAL COMND	Illegal command has been given. Check for correct spelling.
IN SYMBOL TAB	Attempt has been made to define an already defined symbol.
NOT IN SYMTAB	Reference has been made to a symbol which has not been previously defined.
SYMBOL TYPE:	Indicates what type of symbol was being sought (either COMMAND, CHAR. STRING, or NUM. PARAMETER).
LAST SYMBOL:	Last SYMBOL to be referenced is listed.
LAST CHAR:	Last character to be gotten from input file is listed.
!FORM(19FEB85)	Version of *FORM
*PARAMETERS:	When DEBUG is on and a .SET command has occured what follows is a list of program parameter values. Values occur in the order in which they are listed in Appendix B.
NO INTERFACE!	Interface to printer is not present.
NO L,M ON MP12	P=L or P=M are not allowed on the MP12.
LHT, RHT ERROR	Command error specific to the commands .LHT or .RHT.
SYM SUB ERROR	Error occurred during string substitution or numerical parameter substitution. Check

CTRL-G closely.

*LINE OVERFLO! When output is going onto magnetic tape this error message indicates that a line had to be truncated because it exceeded the acceptable line length for printing on the Xerox 9700. This error does not halt *FORM.

*FORM

APPENDIX A

Parameters Set By The .FMT n Command

There are four parameters which are set by the .FMT command.** These are: the Even Page Left Margin (ELM), the Odd Page Left Margin (OLM), the Line Width (LW), and the Line SPacing (LSP). There are 4 format types:

1 Cooley Report Format (n=1)
2 Correspondence Format (n=2)
3 Documentation Format (n=3)
4 Rackham Dissertation Format (n=4)

The following table lists the values of the parameters for each of the format types:

n	ELM	OLM	LW	LSP
1	10 spaces	15 spaces	60 spaces	1.5
2	10 "	10 "	65 "	1.5
3	10 "	10 "	65 "	1.0
4**	15 "	15 "	60 "	1.5

**In addition, the Rackham Dissertation Format (n=4) sets 6 more parameters as follows: RI=0, PL=66, PTH=4, HTT=3, TBF=3, FPB=2.

*FORM

APPENDIX B

Program Parameters

There are 18 program parameters which are used during the execution of the program. They all can be modified as desired by the user through use of the .SET command or .FMT command (see Appendix A for description of .FMT). Most of these parameters are related to the layout of the page (margins, length, width, spacing, etc). (See Appendix C for a picture of the page layout). Exceptions are: PI (paragraph indent), PN (current page number), NTS (the non-trivial space char.), CPI (characters per inch - for converting from inch specifications to absolute value), LPI (lines per inch).

Program parameters which can be .SET:

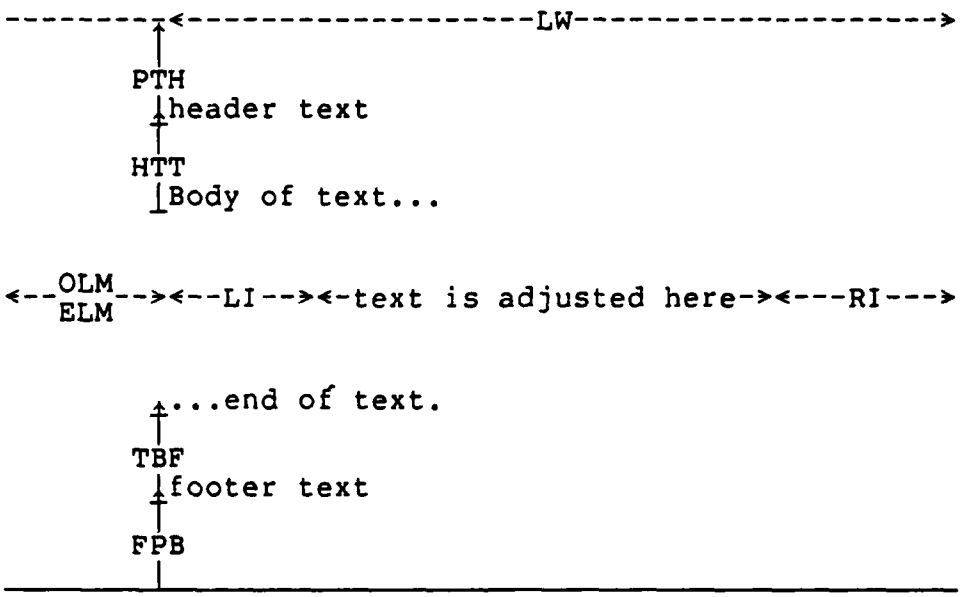
Mnemonic	Type	Inches?	Default	Description
ELM	+INT	YES	10	Even Page Left Margin
OLM	+INT	YES	10	Odd Page Left Margin
LW	+INT	YES	65	Line Width
LSP	+REAL	YES	1.0	Line Spacing
RI	+INT	YES	0	Right Indentation
PL	+REAL	YES	66	Page Length
PTH	+REAL	YES	3	Space from Page Top to Header
HTT	+REAL	YES	3	Space from Header to Text Top
TBF	+REAL	YES	3	Space from Text Bottom to Footer
FPB	+REAL	YES	3	Space from Footer to Page Bottom
PN	+INT	NO	1	Page Number
PI	+INT	YES	5	Paragraph Indent
LI	+INT	YES	0	Left Indentation
NTS	CHAR	NO	"_"	Non-Trivial Space Character
CPI	+INT	NO	10	Characters per Inch
LPI	+INT	NO	6	Lines per Inch

```
*FORM
```

APPENDIX C

Page Layout

The page layout is shown below. The page length is PL.



*FORM

APPENDIX D

Summary of Commands

The following is a summary of the commands which FORM uses. For a detailed description of each command see the section entitled 'COMMANDS' contained in this document.

Mnemonic	Brief Description
.SET	Sets user defined labels or program parameters
.SSET	Sets defined character strings
.DEF	Defines user defined numerical parameters
.SDEF n	Defines character strings
.C	Comment line
.BR	Causes a text break
.A	Turns on adjusting and filling modes
.RR	Turns ragged right mode on and filling mode off
.RL	Turns ragged left mode on and filling mode off
.CE	Turns centering mode on and filling mode off
.F	Turns filling mode on
.PA	Terminates current page
.HT ''''	Sets header for all pages
.FT ''''	Sets footer for all pages
.EHT ''''	Sets header for even pages only
.EFT ''''	Sets footer for even pages only
.ODP	Advances page numbering to next odd page
.RNP	Turns on Roman Numeral page numbering
.BL n	Produces n blank text line spacings
.VT n	Causes a vertical tab to line n
.UL n	Turns on underline mode for next n lines
.EQ n	Turns on equation mode for next n lines
.ENUM n	Sets equation numbering for following equation
.FIT n	Prints following n lines as a "block"
.FIG op m n	Saves the next n lines to make a figure
.DEBUG n	Turns on debug mode for next n lines
.LHT m n	Uses next n lines as a left hand leader
.RHT m n	Uses next n lines as a right hand trailer
.FMT n	Sets format

*FORM

APPENDIX E

Summary of Characters

The following is a summary of the allowed characters and the alternate character set they map into.

A	∝	a	α	0	⁰	+	∪
B	•	b	β	1	¹	,	
C	∈	c	χ	2	²	-	¬
D	Δ	d	δ	3	³	.	∩
E	∈	e	ε	4	⁴	/	÷
F	Φ	f	ϕ	5	⁵	;	
G	Γ	g	γ	6	⁶	<	⊂
H		h	η	7	⁷	=	≡
I	∫	i	∞	8	⁸	>	⊃
J		j		9	⁹	?	∂
K		k	κ	@	■	[	{
L	Λ	l	λ	!	✓	\	
M		m	μ	"	•	]	}
N	∇	n	ν	#	←	^	↑
O		o	∂	$	→	_	↓
P	Π	p	π	%	±	{	{
Q	Θ	q	θ	&	∑	|	¶
R	®	r	ρ	'	°	}	}
S	Σ	s	σ	(	(	~	≃
T	™	t	τ	)	)		
U	T	u	∪	*	×		
V		v		:			
W	Ω	w	ω				
X	Ξ	x	ξ				
Y	Ψ	y	ψ				
Z		z	ζ				

END

DTIC

6-86