

LIBRARY COMMANDS

LIBRARY IN FILENM Abbr: L I FILENM	Load a library input file with a six-character or less file name (FCL extension assumed)
	Input is terminated by end of medium, end of file, or (*) as part of a direct command within file.
LIBRARY OUT FILENM Abbr: L O FILENM	The named FILENM (six characters or less) is initialized for output.
LIBRARY WRITE ALL L W A	Writes entire stored program onto opened output file
LIBRARY WRITE (Line #) L W (Line #)	Writes a single line onto opened output file
LIBRARY WRITE (Group #) L W (Group #)	Writes entire group onto opened output file
LIBRARY WRITE " (Direct command) L W " (Direct command)	Write any FOCAL direct command into the output file
LIBRARY CLOSE L C	Terminates current output file
LIBRARY KILL L K	Terminates output action but does not save output file
Example:	
LIBRARY INPUT TRIANG	Load the file named TRIANG and begin program when direct command beginning with an * is encountered
LIBRARY OUTPUT TRIAN1	Open output file called TRIAN1
LIBRARY WRITE " ERASE ALL	Set file to clear text storage when loaded
LIBRARY WRITE ALL	Write all on output file
LIBRARY WRITE " *; GOTO 10.1	Set file to begin execution at line 10.1 when loaded
LIBRARY CLOSE	Close file and terminate library output mode

digital

digital

pdp9 -
pdp9/L

FOCAL

FOCAL OPERATIONS AND THEIR SYMBOLS

Exponentiation (↑) Multiplication (*) Division (/) Addition (+) Subtraction (←)

Up to three alphanumeric letters are accepted as variable names and all may be subscripted. Any of the following parenthetical pairs may be used—(), [], and < > nested to any depth. All commands except Write, Modify, Quit and Erase may be combined on the same line if separated by a semicolon.

Step numbers from 1.01 to 99.99 are acceptable for indirect commands.

INTERNAL FOCAL FUNCTIONS

FSQT()	Square Root
FABS()	Absolute Value
FSGN()	Sign Part of the Expression
FITR()	Integer Part of the Expression
FRAN()	A Noise Generator
FEXP()	Natural Base to the Power
FSIN()	Sine
FCOS()	Cosine
FATN()	Arctangent
FLOG()	Naperian Log

FOCAL ERROR DIAGNOSTICS

If the user makes an error, FOCAL automatically prints out a code describing the problem. A line number will be printed if an indirect command caused the error.

CODE	MEANING
?01	Illegal character at beginning of line
?02	Group number illegal as line number
?03	Group number too large
?04	Illegal type/ask format
?05	Too many periods
?06	Line number too large
?07	Line number missing
?08	Illegal group number
?09	Push-down list overflow
?10	Illegal command
?11	Illegal "IF" format
?12	Left of equals in error on "FOR" or "SET"
?13	Excess right parenthesis
?14	Illegal "FOR" format
?15	Illegal variable name
?16	Text/variable buffer overflow
?17	Illegal expression format
?18	Operator missing before parenthesis
?19	Missing left parenthesis
?20	Illegal function name
?21	Double operator
?22	Parenthesis error
?23	"ERASE" or "WRITE" argument error
?24	Negative line number
?25	Zero argument for log
?26	Input overflow
?27	Number too large
?28	Negative power illegal
?29	Division by zero illegal
?30	Square root of a negative number
?31	Illegal command during library output
?32	Illegal library command
?33	Illegal file name
?34	File not found
?35	No library output file open
?36	.OTS error from FORTRAN IV arithmetic package
?37	COMMON format error
??nn	User defined function error

FOCAL COMMAND SUMMARY

Command	Abbr.	Example of Form	Explanation
TYPE T		TYPE FSQT(AL 3 + FSQT(B))	Evaluates expression, types out result in current output format
		TYPE "TEXT STRING"!	Types text. Use ! to generate carriage return line feed
WRITE W	WRITE ALL		Focal prints the entire indirect program
		WRITE 1	Focal types out all group 1 lines
		WRITE 1.1	Focal print line 1.1
IF I	IF (X)	1.2, 1.3, 1.4;	Where X is identifier or expression
			Control is transferred to the first, second, or third line number if (X) is less than, equal to, or greater than zero respectively. If the semicolon is encountered prematurely then the remainder of the line is executed.
MODIFY M	MODIFY	1.15	Enables editing of characters on line 1.15
			The next character typed becomes the search character. FOCAL will then position itself after the search character; then the user may:
			1) type new text, or
			2) push altmode to go to the next occurrence, or
			3) push ↑ (control key) bell to change the search character, or
			4) push rubout to delete backwards, or
			5) push ↑ (control key) U to kill entire line backwards, or
			6) push carriage return to end the line, or
			7) push line-feed to save the rest of the line.
QUIT	Q	QUIT	Returns control to user
Return	R	RETURN	Terminates DO sub-routines
SET	S	SET A=5/B*SCALE(3)	Substitution statement
ASK	A	ASK ALPHA (1+2*J)	FOCAL types a colon for each variable; the user types a value terminated by a carriage return or altmode to define each variable.
COMMENT	C	COMMENT	If a line begins with the letter C and is not COMMON the remainder of the line will be ignored.
CONTINUE	C	C	

COMMON	COMMON	A, B(1)	Defines these variables as permanent
	COMMON	(C, X, Y, Z)	Defines these array elements as permanent (see FOR for description of x,y,z)
DO	D	DO 4.14	Execute line 4.14; return
		DO 4	Execute all group 4 lines, return when group is expended or when a RETURN is encountered.
		DO	Execute entire indirect text as a subroutine.
ERASE	E	ERASE	Erases the symbol table.
		ERASE 2	Erases all group 2 lines.
		ERASE 2.1	Deletes line 2.1
		ERASE ALL	Deletes all user text.
		ERASE COMMON	Erases all permanent variables
FOR	F	FOR 1=x,y,z; TYPE I	The command string following the semicolon is executed for each value. x,y,z are constants, variables, or expressions. x=initial value of I. y= value added to I until I is greater than z. y is assumed=1 if omitted.
GO	G	GO	Starts indirect program at lowest numbered line number.
		GO?	Traces entire indirect program
GOTO	G	GOTO 3.4	Starts indirect program at line 3.4