

LIBRARY COMMANDS

LIBRARY IN FILENM Abbr: L I TRIANG and Example	Load a Focal library input file with a six-character or less file name (FCL extension assumed) from the .DAT slot 3 device.† Input is terminated by end of medium, end of file, or (*) as part of a direct command within file.
LIBRARY OUT FILENM L O TRIANG	The named file (six characters or less) is initialized for output on the .DAT slot 5 device.
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 " ERASE ALL	Write any FOCAL direct command into the output file. Here the direct command, ERASE ALL, sets file to clear text storage when loaded.
L W "*; GOTO 10.1	Terminate input from .DAT slot 3 device and set file to begin execution at line 10.1 when loaded.
LIBRARY CLOSE L C	Terminates current output file.
LIBRARY KILL L K	Terminates output action but does not save output file.

†NOTE: Refer to **PDP-15 FOCAL Programming Manual** for explanation of .DAT slot and device assignments.

DATA COMMANDS

DATA OUT FILENM Abbr: DATA O LOAN and Example	Initializes the device assigned to .DAT 10 and, if file-oriented, enters the filename (e.g. LOAN) in the file directory.
DATA CLOSE FILENM DATA C LOAN	Closes the output file and returns the user to Teletype mode.
DATA KILL DATA K	Aborts the output file and returns the user to Teletype mode.
DATA IN FILENM DATA I LOAN	Initializes the device assigned to .DAT 7 and reads in the file named (e.g., LOAN).

digital

digital

pdp15
FOCAL

FOCAL OPERATIONS AND THEIR SYMBOLS

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

Up to three alphanumeric letters are accepted as variable names and all may be subscripted. First character must be alphabetic and not the letter "F", since "F" is used for FOCAL functions. 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()	Random Number 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
?00	Function not implemented
?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	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.
	TYPE %x.yz	% changes output format. x = number of digits to be output. yz = number of digits to the right of the decimal point, and must be followed by a comma. If x.yz is omitted, output format is reset to floating-point (comma not needed).
	T %4.02, A	
	T %	
	TYPE \$	Type out the current symbol table.
WRITE W	WRITE ALL	Focal prints the entire indirect program.
	WRITE 1	Focal types out all group 1 lines.
	WRITE 1.1	Focal types 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 type:
		1) New text.
		2) ALT MODE to find next occurrence of search character.
		3) CTRL BELL and new search character.
		4) RUBOUT to delete preceding (search) character.
		5) CTRL U to delete entire portion of line to the left (except line number).
		6) Carriage Return to end line.
		7) LINE FEED to save rest of line.
QUIT	Q QUIT	Returns control to user.
SET	S SET A=5/B*C	Substitution statement. Each occurrence of A during program execution is replaced by the value of the expression.

ASK	A	ASK A, B, C	FOCAL types a colon for each variable; the user types a value terminated by a carriage return or ALT-MODE to define each variable.
COMMENT	C	COMMENT	Allows comments to be inserted in the program. Any line starting with a "C" (except COMMON) is treated as a comment and ignored.
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; control returns to statement following DO.
		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 (same as GO).
ERASE	E	ERASE	Erases the symbol table.
		ERASE 2	Erases all group 2 lines.
		ERASE 2.1	Erases line 2.1
		ERASE ALL	Erases all user text.
		ERASE COMMON	Erases all permanent variables.
FOR	F	FOR A=x, y, z; DO 2.0	The command string following the semicolon is executed for each value. x, y, z are constants, variables, or expressions. x = initial expression of A. y = value added to A until A 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.
RETURN	R	RETURN	Exit from a DO subroutine, and return to command following DO command.