



DECUS

PROGRAM LIBRARY

DECUS NO.	FOCAL8-52a
TITLE	FOCAL 5/69
AUTHOR	Edward A. Taft III
COMPANY	Old Neck Road Manchester, Massachusetts
DATE	July 1972
SOURCE LANGUAGE	FOCAL

ATTENTION

This is a USER program. Other than requiring that it conform to submittal and review standards, no quality control has been imposed upon this program by DECUS.

The DECUS Program Library is a clearing house only; it does not generate or test programs. No warranty, express or implied, is made by the contributor, Digital Equipment Computer Users Society or Digital Equipment Corporation as to the accuracy or functioning of the program or related material, and no responsibility is assumed by these parties in connection therewith.

FOCAL 5/69

DECUS Program Library Write-up

DECUS No. FOCAL8-52a

GENERAL

This is a new version of FOCAL, based on FOCAL-W, 8/68, which has been expanded and rewritten to remove numerous bugs and restrictions and to provide a large number of new commands and extended capabilities.

Some of the new features are:

- a) Better control over I/O devices, including the high-speed punch.
- b) New I/O formats, including buffered input that accepts expressions as well as numbers, input and output of single ASCII characters, and a tabulation controller.
- c) A group of "OPTION" commands that perform minor functions such as suppressing or restoring keyboard echo and changing I/O modes.
- d) A more compact extended function package, resulting in an enlarged user area; also a command for deleting the extended functions.
- e) Extended command formats; also a provision for using calculated line numbers.

This documentation is a supplement to the FOCAL manual, and presumes that the user is already familiar with FOCAL-W.

LOADING

The following **binary** versions of FOCAL 5/69, are available:

- a) FOCAL 5/69 Binary - This tape contains the basic version, and is in two sections. The second (smaller) section contains the extended functions and enables the LIBRARY DELETE command for deleting them at a later time. The extended functions may be omitted at load time, in which case the LIBRARY DELETE command is also omitted, by loading only the first section of the tape. The extended functions may also be restored at any time by loading the second section of the tape. CAUTION: there may be trouble if extended functions overlap a user program; therefore, load the extended functions only after issuing an ERASE ALL command, or with a short program present (less than about 1500 characters).

Loading Instructions (Non disk)

Be sure the BINARY LOADER is in core, then perform the following steps:

1. Set 7777 into the Switch Register and press LOAD ADDRESS.
2. If the high-speed reader is being used, set SR bit 0 to zero.
3. Place FOCAL 5/69 Binary in the selected reader. If the teletype reader is being used, set the reader switch to START.
4. Press START, the tape will be read in. It will stop at the end of the first section. If accumulator lights are on checksum error has occurred and the loading procedure should be repeated.
5. The extended functions are loaded if desired by pressing press CONTINUE. When the reader stops, check the accumulator as above.
6. Turn the teletype to LINE. Set 0200 into the Switch Register, press LOAD ADDRESS and then START. The teletype will print:
C-FOCAL, 5/69
and type a back-arrow on the next line, signaling that it is ready for commands.

Disk-Monitor

To save FOCAL 5/69 on the systems device, ensure the Monitor is in core and start by striking CTRL/C or starting at 7600.

<pre>.LOAD) *IN-R:) * * *OPT-2 ST=0) ↑↑↑↑↑↑</pre>	<p>Call the Binary Loader. Specify input device (T; , T: if no high-speed reader).</p> <p>Two-pass load. Return to Monitor after load. After each arrow, strike CTRL/P to continue loading. At the end of the second section of tape it should be repositioned to the beginning.</p>
--	--

```
.SAVE FOCAL ! 0-3577, 5000-7577; 200 )
```

To save FOCAL without the extended functions:

<pre>LOAD) *IN-R:) * *OPT-2 ST=0) ↑↑↑↑↑</pre>	<p>Reposition the tape at the end of the first section.</p>
--	---

```
.SAVE FOCAL ! 0-3577, 5600-7577; 200 )
```

b) FOCAL 5/69 Two-User Binary - This version requires 8K or more of memory, a second teletype and PT08 control. The tape is a binary-punched version, containing the extended functions, the LIBRARY DELETE command, and a starting patch in the user area which transfers FOCAL to Field 1 and performs the necessary modifications for two-user operation. The coding in Field 0 is for the standard teletype (TTY 0), and the coding in Field 1 is for the PT08 teletype (TTY 1). The system is arranged so that the computer will swap back-and-forth between users at least 1000 times per second (PDP-8/I), thus providing a pseudo-time-shared system.

Loading Instructions (Non disk)

Load FOCAL 5/69 Two-User Binary with the Binary Loader as previously described. The tape has one section. Turn on both teletypes, set 0200 into the switch register (the Data Field and Instruction Field switches should be all set to zero), press LOAD ADDRESS and START. Both teletypes will print C-FOCAL, 5/69. The teletypes may now be independently operated.

Disk-Monitor

To save Two User Focal on the systems device, place the FOCAL 5/69 Two-User binary tape in the selected reader and give this command sequence:

. LOAD)	
*IN-R:)	(Or T: if using the teletype reader.)
*		
*OPT-2		Two-pass load.
ST=0)	Return to monitor after load.
↑↑↑↑		CTRL/P as before. Reposition the tape to the beginning when it reaches the end.
. SAVE FOCAL ! 0-4177, 5000-7577; 200)		

Note that it is not necessary to SAVE anything from Field 1 (as it was with FOCAL 8/68, Two-User). All the necessary coding is initially present in Field 0 and is swapped to Field 1 after starting.

c) Restarting - All versions may be restarted at location 0200 in Field 0. The teletype will respond by printing ?00.00, indicating manual restart. (In two-user FOCAL, only TTY 0 will respond; however, both users will have been restarted.)

COMMAND CHANGES AND NEW FEATURES

The following is an alphabetical list of FOCAL commands, explaining all the changes in the FOCAL, 5/69 version

ASK

This command now operates in two modes, Interpretive and Character. It is initially set to

Interpretive mode and may be changed by the OPTION commands.

In Interpretive mode FOCAL accepts either numbers or valid numerical expressions (composed of numbers, functions, and defined variables). The following are correct responses:

```
:3.4576
:23*4.68E-5
:3/FSQT(77)
:A+3*B-2
```

(assuming A and B are defined)

RUBOUT may now be used to correct mistakes (as in command input), and the back-arrow works normally. Leading spaces are ignored. The number or expression must be terminated by a carriage-return, comma, or CTRL/L character. The user may also specify another character as a terminator (see OPTION S). Note that space and ALTMODE are no longer terminators. If there is an error in the input expression, FOCAL will type an error message and stop, giving the step number of the ASK command as the location of the error.

There is also a provision for attaching comments to the input: anything typed after a semicolon will be considered a comment and ignored.

The ASK command types a colon for each variable whose value it is asking. The colon printout may be suppressed, if desired, by the OPTION X command. All other features of ASK work normally (as in TYPE); see TYPE for the change to % and the addition of &.

In Character mode, which is selected by the OPTION C command, FOCAL accepts a single ASCII character for each variable and converts the character to its decimal code (see the list of decimal ASCII codes). Any printing or nonprinting character may be input by this method, and there is, of course, no need to terminate the input. Example:

```
← OPTION C; FOR I=1,6; ASK X(I), " "
:A :B :X :# :e :3 ←
← FOR I=1,6; TYPE %3.0, X(I)
ABX#@3 ←
← OPTION I; FOR I=1,6; TYPE X(I)
193 194 216 163 192 179 ←
```

(Note: in FOCAL 8/68, the only valid input is a number, and RUBOUT cannot be used.)

COMMENT

No change. Notice that a blank line is also a comment, as is step 00.00 (the heading C-FOCAL, 5/69, which is typed without its step number).

DO

The DO command works normally. Note that DO 3 is equivalent to DO 3.00.

There are two new features available in the DO command. First, more than one line number may be given, separated by commas; FOCAL will execute them in order, as if they had been given in separate DO commands. For example,

DO 3.12, 5.54, 4, 14, 1.56;

is equivalent to

DO 3.12; DO 5.54; DO 4; DO 14; DO 1.56;

This feature is also implemented in WRITE.

Second, a line number may be any valid numerical expression if it is preceded by a sign (+ or -). Thus,

DO +N-.1

will DO step 3.2 if N is equal to 3.3. The expression is evaluated and truncated to the next lower hundredth to arrive at a line number; if it is negative or greater than 15.99, an error message will result.

WARNING: in some cases, due to errors in approximating a decimal number in binary, a number like 4.30 will be stored as 4.299999, and will be truncated to the step number 4.29; if this occurs, simply add a factor of .005 to the line number expression.

This feature, which is also available in ERASE, GOTO, IF, MODIFY, and WRITE, is very useful within a program for allowing variables to determine what steps are to be executed.

Note that a line number must be in standard form (XX.XX, X.XX, XX.X, X.X, XX or X) if it is not preceded by a sign. Note also that FOCAL uses over five times as much time evaluating a line number in floating point as it does in fixed point.

ERASE

Erase works normally, with the exception that the command ERASE \emptyset will no longer destroy FOCAL as it did in FOCAL 8/68. ERASE without a line number or ALL erases only the variable storage; the command does not terminate the line as it did before. When a line number or ALL is given, however, FOCAL returns to command mode immediately after executing the command; it also erases the variables (this was true of FOCAL 8/68 as well, although not made clear by the manual). This is because the variable pointers are upset by erasing, modifying, or adding steps (program and variables share the same storage).

Summary

{ ERASE ERASE \emptyset	Erase the variables and continue running.
ERASE 12.34	Erase step 12.34 and variables.
ERASE 3	Erase group 3 and variables.
ERASE ALL	Erase program and variables.

Note: calculated line numbers are legal as in DO.

FOR

This command has been improved. The format is the same as it was before (either FOR I=n1, n2, n3; ... or FOR I=n1, n2; with an assumed increment of 1), but there are, in addition, two new features:

1) A negative increment may be given for backward counting. For example, FOR I= 4,-1,-3 will count: 4,3,2,1,0,-1,-2,-3.

2) If the initial value is already beyond the final value, the object statements (the rest of the line after the FOR command) will not be executed at all. (FOCAL 8-68 would always execute the object statements at least once).
For example:

```
← FOR I=1,3; TYPE "X"  
  XXX ←  
← FOR I=1,0; TYPE "X"  
← FOR I=5,-2,1; TYPE "X"  
  XXX ←  
← FOR I=5,-2,6; TYPE "X"  
← SET N=77; FOR I=5,-1,N; TYPE "X"  
←
```

GO

The GO command works normally. Note that GO is equivalent to GOTO 0.

GOTO

GOTO allows calculated line numbers as in DO.

HELLO

This is a new command which quickly and easily initializes FOCAL to its original conditions. HELLO performs the following commands:

```
TYPE %8.4; OPTION K, T, I, E, :, S; ERASE ALL.
```

Use of this command will prevent confusion, especially to beginners, resulting from output being typed in unexpected formats or a previous program not being erased.

IF

IF command operates as before. Calculated line numbers are allowed as in DO. The space between IF and the left parenthesis is now no longer necessary.

Note: The expression must be enclosed by curved parentheses; square or angle brackets are not allowed.

LIBRARY DELETE

Library Delete is a new command which deletes the extended functions FSIN, FCOS, FATN, FLOG, and FEXP in order to free more core for user area. The user area is 816 words long with the extended functions present; the figure is increased to 1126 words when they are removed.

WARNING: the extra space is not available until FOCAL has once returned to command mode; thus, LIBRARY DELETE should be given as a direct command only.

This command is not enabled unless the extended functions were initially loaded with FOCAL. (They are automatically loaded into both fields in Two-User FOCAL.) Thus if the functions were omitted by not loading the second section of the tape, LIBRARY DELETE is an illegal command.

LIBRARY is designed as an expandable command; LIBRARY DELETE is the only form implemented now, but others could be added if desired by sacrificing some user area. (There is presently no free space left within FOCAL.) Interested users can obtain the listing at a handling charge of \$5.00

MODIFY

MODIFY works as usual. Calculated line numbers are accepted. Remember that the use of MODIFY erases the variables and returns to command mode.

CAUTION: It is possible, using MODIFY, to make a line more than 72 characters long (including line number). FOCAL will not object to this. However, if the program is punched and later loaded back in via the reader, the line will be too long to read and will cause an error message.

OPTION

This is a new family of commands for performing operations useful mainly to the advanced programmer. An OPTION command consists of the word OPTION followed by one or more words (which may be abbreviated to their initial letters) separated by commas. For example:

OPTION KEY, PUNCH, ECHO, X
OPT K; OPT P; OPT E; OPT X
O K, P, E, X

are all equivalent.

COMMANDS

OPTION R Switch all input to the high-speed reader. All subsequent data and command input is taken from the reader until an OPTION K command is given or the reader runs out of tape. To read in a program, give OPTION R as a direct command with the tape in the reader.

- OPTION K Switch input to the teletype keyboard (or teletype reader). Input is also switched to the keyboard if the reader runs out of tape, an error occurs, or CTRL/C is struck.
- OPTION P Transmit all output to the high-speed punch. This new facility punches all the output that would normally be typed, including keyboard echo and back-arrows. (Key echo may be suppressed by the OPTION N command.)
- OPTION T Switch output to the teletype; this restores FOCAL to normal teletype output. Output is also switched to the teletype if an error occurs or CTRL/C is struck.
- OPTION I Set the ASK and TYPE commands to Interpretive Input mode and Numeric Output mode. (FOCAL is initially set to this mode.) See ASK and TYPE.
- OPTION C Set ASK and TYPE to Character Input/Output mode, whereby a single ASCII character is read or printed for each variable or expression; the character is stored by FOCAL as a decimal number corresponding to the character code.
- OPTION X Suppress the colon printout, which occurs when FOCAL is executing an ASK command. This command is useful for providing cleaner output, especially when inputting data from the high-speed reader. When OPTION X is in effect, FOCAL simply pauses for input of each variable.
- OPTION : Restore the colon printout, if deleted as above.
- OPTION S
number Set the number following "S" as the character code for an extra input terminator. Normal terminators (used in response to an ASK command) are return, comma and CTRL/L. The user may define space to be a terminator by typing OPTION S 160 where 160 is the code for the character space. (Leading spaces are ignored on input and will not terminate). OPTION S 138 will define line-feed to be the terminator.
- Note: only one special terminator is in effect at any one time.
- OPTION S (With no number): delete the special terminator.

- OPTION N No Echo: normally, all characters typed on the keyboard or input from the low-speed reader are printed, or "echoed," on the teletype. This feature may be suppressed by the OPTION N command. Note that a back-arrow will still be printed to signal command mode, and a backslash will still be typed in response to rubout.
- OPTION E Restore the echo, if deleted as above.
- OPTION M Start the Disk Monitor. The Disk/DECtape monitor starts at location 7600; OPTION M causes the computer to exit FOCAL at the completion of all printing and jump to this location, thereby starting the monitor, which responds with a period. Obviously, this command should not be used if the computer does not have a monitor system. The OPTION M command may be suppressed if necessary by changing the contents of location 3271 from 4225 to 4526.

FOCAL is initially set (and is set by the HELLO command) to the following states:

- OPTION K Keyboard input.
 OPTION T Teletype output.
 OPTION I Interpretive/Numeric I/O mode.
 OPTION E Echo feature operative.
 OPTION : Colon printout enabled at ASK commands.
 OPTION S Special input terminator deleted.

QUIT

No change.

RETURN

No change. Note that RETURN is equivalent to QUIT if there is nothing to return to.

SET

Works normally. FOCAL will type an error message if the expression on the right of the equals sign contains an undefined variable (one that has not been given a value in a SET, FOR, or ASK command).

TYPE

TYPE has been improved in several ways. FOCAL will now no longer hang up typing zeroes when it encounters certain format errors. Also, TYPE no longer prints an equal sign before

each number to be output, as it did in FOCAL.8/68.

When the input/output mode is Interpretive (see OPTION I and ASK), the output will be typed in the numerical format (specified by use of the % sign). When the mode is Character (see OPTION C), TYPE will instead convert the output value to a character and print that single character. For example:

```
← SET A=195; SET B=183; SET C=164; SET D=221
← OPTION I; TYPE %3.00, A, B, C, D
    195 183 164 221 ←
← OPTION C; TYPE A, B, C, D
    C7$J ←
```

All characters may be output in this mode (including non-printing characters); legal codes run from 0 to 255. If a code outside this range is given, it will be reduced modulo 256 to a number between 0 and 255.

The following changes are effective in both TYPE and ASK commands:

<u>Symbol</u>	<u>Example</u>	
%	%7.3	This symbol is used for setting the format in which numbers are to be typed. "TYPE %7.3. means type 7 digits, of which 3 are to be to the right of the decimal point. (It is now no longer necessary to specify %7.03). Examples:

```
← SET A=3.472
← TYPE %5.4, A
    3.4720 ←
← TYPE %9.2, A
    3.47 ←
← TYPE %2, A
    3 ←

← TYPE %4.1, A
    3.5 ←
← TYPE %6.6, A/10
    .347200 ←
```

Other improvements:

- 1) The last example would not have been legal in FOCAL.8/68. In FOCAL.5/69, the leading zero before the decimal point is suppressed if necessary.
- 2) Rounding is now done correctly. In %2 format, the number 7.5 will be rounded to 8, and 7.499 will be rounded to 7.

Symbol Example

3) Output fields are no longer restricted to 31 characters.

%
%0.0 This is the signal for E format output.
Numbers are printed with six decimal places (but no leading zero), followed by a sign and tens exponent.

%0.4 This means type in E format, but with only four digits.

Note: the format is initially set (and set by HELLO) to %8.4.

\$ \$ This is the signal for FOCAL to type the values of all defined variables. The \$ does not terminate the command line as it did in FOCAL 8/68. The values are typed in numerical format even if the I/O mode is Character. Also, the full, signed subscript is typed; FOCAL 8/68 could only print positive subscripts up to 99. Legal subscripts are -2048 to +2047.

Note: the values are typed in whatever format is in effect at the moment.

Example:

```
← ERASE; SET X=2.354; SET A(44)=-2345.6
← SET VW(-554)=X+A(44); TYPE $; ERASE ALL
X@ (+00)=        - 2.3540
A@ (+44)=       -2345.6000
VW(-554)= -2343.2400
←
```

& &22
& N
& B+3 This new symbol is a tabulation controller. For tabulation purposes, the teletype columns are numbered from 1 to 72 from left to right, and the specification & N, where N is an expression, will cause the teletype to tabulate to the Nth column (if it is already at or past the Nth column, the command will be ignored). This feature is useful for aligning input data, providing more concise output, and plotting. Note that non-printing characters will not be counted and, therefore, will not throw off the alignment.

The symbols ", !, and # will work normally. Note that it is not necessary to put a comma after a !, #, or right-hand quote.

WRITE

WRITE works normally. Several line or group numbers may be given after a single WRITE command, separated by commas, and the numbers may be calculated (see DO). A WRITE command no longer terminates the command line.

Note: WRITE without an argument is equivalent to WRITE ALL.

A convenient way to WRITE groups of steps is as follows: to WRITE all the steps from 3.42 to 3.77, give this command:

```
← F I=3.42, .Ø1, 3.77; W + I
```

This will work even if there are only a few steps.

COMMAND FORMATS

a) The character @ is no longer ignored by FOCAL on input, and is treated like any other character. Thus, it is not suitable for leader. For this reason, tapes prepared on FOCAL 8/68 with @@@@ for leader, will cause an error message at the start of a load.

b) Blank characters, generated by CTRL/SHIFT/P or by pressing HERE-IS (if the answer-back drum has not been coded), are now suitable for leader, and for this purpose they will echo on-line. They are ignored by FOCAL at all times. To prepare a tape of a program:

- 1) Type WRITE ALL, but do not press RETURN.
- 2) Turn on the tape punch.
- 3) Generate leader, using HERE-IS.
- 4) Press RETURN, and the program will be typed and punched.
- 5) When it is finished, generate more leader, turn off the punch, and remove the tape.

To punch a program using the high-speed punch, simply turn on the punch, press FEED to generate leader, and type:

```
← OPTION P; WRITE ALL; OPTION T
```

When it is done, press FEED again to generate more leader.

Blank tape may also be generated by an operating program by the use of OPTION C; leader has codes of Ø and 128.

c) FOCAL maintains an input buffer and an output buffer. However, whereas the output buffer is 15 characters long, allowing FOCAL to continue processing while it is still typing previous output, the input buffer is only one character long. Input buffer overflow occurs when this buffer is filled faster than it can be used, and causes FOCAL to stop and print the message ?11.18.

Due to the construction of the keyboard control, when the low-speed reader is on and contains tape, the tape is read continuously, whether or not FOCAL is ready for the next character. When reading in long tapes, the reader can easily get more than 16 characters ahead of the output and cause an input buffer overflow. There are two ways to prevent this:

- 1) Every ten or fifteen lines, stop the reader momentarily, until the printing has caught up.
- 2) If a listing of the program is not needed, give an OPTION N command before reading in the tape. The program will not be typed as it is read in. To restore the echo at the end, type OPTION E.

Note that this problem does not occur with the high-speed reader; reading takes place only as fast as FOCAL can use it.

d) FOCAL has a Command Buffer, which is 72 characters long, for holding direct commands as they are being typed in; there is also an ASK Buffer, which is 46 characters long, for holding input typed in response to the ASK command. If either buffer is overflowed, FOCAL will type an error message and then read in the rest of the line, which will also probably cause an error. Therefore it is important that 1) program steps do not exceed 72 characters, including the line number in its standard form (XX.XX), and 2) data input expressions do not exceed 46 characters (including comment, if any).

e) FOCAL recognizes commands by their first letter and ignores the rest of the command word. Thus FOCAL commands must be properly terminated so that FOCAL can see where they end. In FOCAL 8/68, the only terminator is space, and it is necessary to type a space after the command word. In FOCAL 5/69, the following characters will terminate a command word:

space ! " . + - ([< , ;

Thus, in a long program where space is at a premium, some storage may be saved in some cases by omitting spaces. In the following example, the commands on the left, formerly illegal, are now equivalent to the commands on the right:

<pre>T !! A "WA="WA, !! I(WA-1.2)3.21; T-1.1*WA</pre>	<pre>T !! A "WA=" WA, !! I (WA-1.2) 3.21; T -1.1*WA</pre>	<pre>TYPE !! ASK "WA=" WA, !! IF (WA-1.2) 3.21; TYPE -1.1*WA</pre>
---	---	--

Note: GOTO must be one word, since it is a single command. If GO TO is typed (as two words), FOCAL will only skip to the first (separating) space, then, finding no line number, will interpret the command as "GOTO \emptyset ."

f) In FOCAL 8/68, there was a restriction of about 10 consecutive digits in a number, which, if exceeded, caused an error message "literal number too large." This restriction has been removed. Any number of digits may be typed; however, digits beyond about the first eight will have no significance and will be processed only as place markers.

MISCELLANEOUS CHANGES AND RESTRICTIONS

a) The arithmetic priorities have been changed to conform to standard algebra and to FORTRAN. Multiplication and division are on the same priority level, as are addition and subtraction. In the absence of parentheses, FOCAL will perform multiplication and division together in left-to-right order, followed by addition and subtraction in left-to-right order. The priority list is now as follows:

↑	Exponentiation.
* and /	Multiplication and Division
+ and -	Addition and Subtraction.

b) It is now permissible to raise a number to a negative power. Legal exponents are integers from -2048 to $+2047$. For exponents outside this range, or which are not **integers**, ↑ cannot be used. Instead, use:

$$N^X = e^{X \ln N} = \text{FEXP}(X * \text{FLOG}(N))$$

c) Changes to functions:

1) FITR(X) returns the value of the greatest integer less than or equal to X. FOCAL 8/68 worked improperly for numbers between zero and -1.

FITR(3.1)= 3
FITR(2.9)= 2
FITR(1.5)= 1
FITR(-.1)= -1
FITR(-.9)= -1
FITR(-1)= -1
FITR(-5.4)= -6

2) FSGN(X), the sign function, is equal to 1 if X is positive and nonzero, 0 if X is zero, and -1 if X is negative. Previously, FSGN(0) was given as 1.

FSGN(-27.9)= -1
FSGN(13.4E7)= 1
FSGN(0)= 0
FSGN(2E-555)= 1

3) FRAN(), the random number function, is equal to a random fraction between zero and one. Unlike FRAN() in FOCAL 8/68, this random number generator is suitable for statistical use of any kind. The numbers are well-distributed and, in fact, are not expected to repeat for over four billion numbers.

d) The * (asterisk) command, which controlled the high-speed reader in FOCAL 8/68, is

no longer legal, having been superseded by OPTION R (to turn it on) and OPTION K (to turn it off). It is for this reason that FOCAL's "Ready" character has been changed from asterisk to back-arrow, so that the "Ready" character at the end of a punched tape will not result in an error message when the tape is read back in (a back-arrow at the beginning of a line is simply ignored). Thus, when FOCAL types a back-arrow, it is in command mode, and this is a signal, "Your turn to type."

e) The FOCAL 8/68 provision for "Alphabetic numbers" is not implemented in FOCAL 5/69. Numbers are composed only of signs, digits, a decimal point and "E;" and a number such as ØABC is not permitted. Alphabetic information in response to the ASK command is now interpreted as a variable name rather than as an "alphabetic number." To program for user responses such as "YES" and "NO," simply define the variables YE and NO before giving the ASK statement. More sophisticated alphabetic input (and also output) may be handled in character mode.

ALT-MODE is no longer a special ASK character. In FOCAL 8/68, it was used to skip over the ASK command. The same result may be achieved in FOCAL 5/69 by just typing the name of the variable that is being ASKed.

f) When FOCAL 8/68 encountered a variable which had not yet been defined, it did not recognize this fact and took that variable's value to be zero. In practice this caused confusion because users were usually not aware that they were dealing with undefined variables.

This has been fixed in FOCAL 5/69. When an undefined variable name is found in an expression which is being evaluated, an error message is printed. This reminds the user to initialize his variables before he uses them in expressions.

g) FOCAL 5/69 will not operate on a PDP-5.

USER AREA INFORMATION

The user area is the area of core from 3432 to 5112, or from 3432 to 5577 if the extended functions have been deleted. In decimal, these figures are 816 and 1126 words, respectively. The user area is used for storing the user program ("text"), the names and values of all defined variables, and the push-down list which is used by FOCAL for keeping track of operations. The text is stored in the user area starting at the bottom. Variables are stored starting at the end of the text (this is why they are erased every time the text is changed). The push-down list is filled from the top of the user area backwards. Eleven words are reserved for the push-down list when execution starts, but usually more is needed. When the push-down list and the variable storage run into each other, the message ?Ø2.81 is printed and execution stops.

The following is a discussion of storage requirements:

a) Every defined variable requires 5 words of storage.

b) Every line of text requires three words of storage plus one word for every two printing characters on the line, not including the line number and the space immediately after it. Other spaces, however, take half a word each like printing characters. Nonprinting characters take a full word each.

c) The length of the push-down list varies during the running of a program, and tends to take up more space the more complex the program is. Here are some guides:

1) Each DO requires 7 words for the duration of the DO. FOCAL uses these locations to remember where it was before the DO was executed, so that it can return to it later.

2) Each FOR command requires 12 words, which are used, until the FOR is exited, for storing the loop values.

3) Extra space is also needed while FOCAL evaluates an expression; the amount needed depends on the complexity of the expression.

d) A useful rule-of-thumb is as follows:

$$\text{Space required (in words)} = 3n + \frac{c}{2} + 5v$$

where n = the number of lines in the program.
 c = the number of characters in the program, not including line numbers.
 v = the number of variables defined within the program. Note that the separate subscripted elements of an array count the same as single variables

Subtract the resulting figure from the total space available (816 or 1126) to find the amount of remaining space (which must also serve the needs of the push-down list).

Saving FOCAL Programs On the Disk

Often-used programs may be saved on the systems device and called back into FOCAL at a later time.

Before this may be done, it is necessary to SAVE a file called "RESTART," used for re-starting FOCAL after OPTION M has been used to call a user program. This is done as follows:

```
.FOCAL
C-FOCAL, 5/69
←OPTION M
.SAVE REST ! 7200-7577; 200
.RESTART
?00.00 (indicating FOCAL has restarted)
←
```

Saving a Program

Compose and debug the program as usual on FOCAL. Now type:

```
← OPTION M
.SAVE NAME : 0, 3400-5177; (where NAME is the
.RESTART                name desired for the
?00.00                  user file)
←
```

The program is still in core and has not been affected by the save. If the extended functions are deleted at the time of the SAVE, the following core specifications should be used in the SAVE command:

```
.SAVE NAME : 0, 600, 3400-5577;
```

Calling a Program

FOCAL must be in core before a program can be called from the disk.

```
← OPTION M
.CALL NAME
.RESTART
?00.00
←
```

The called program replaces the previous contents of the user area. If the extended functions were deleted when the called program was saved, they will be deleted now. If the extended functions were present at save time but deleted in FOCAL before the user program was called, they will be deleted now, but the extra space will not be available until a LIBRARY DELETE command is given.

CAUTION: Do not attempt to use RESTART unless FOCAL is in core.

Saving and Calling in TWO-USER FOCAL

It is equally simple to save user programs while running Two-User FOCAL. The file RESTART is saved the same way as for standard FOCAL.

Since the Disk Monitor operates only from TTY 0, all calls and saves must be made from that teletype. To save a user 0 program, be sure neither terminal is active and use the same procedure as that explained above. A program saved from Field 0 will be loaded into Field 0 when called; there is no way to load it into Field 1. When either saving or calling, Field 1 programs will not be affected.

The following commands are typed on TTY 0.

Saving a TTY 1 program with extended functions:

```
← OPTION M  
.SAVE NAM2 ; 0, 3400-15177;  
.RESTART  
?00.00  
←
```

To save when the extended functions are deleted, use this specification:

```
.SAVE NAM2 : 0, 600, 3400-15577;
```

Calling a TTY 1 program:

```
← OPTION M  
.CALL NAM2  
.RESTART  
?00.00  
←
```

In all of the above cases, the Field 0 program will not be affected.

It is important to be aware, when calling programs off the disk, to remember which field they were saved from. User programs will always be loaded into the field from which they were saved.

TWO-USER OPERATION

In TWO-USER FOCAL the teletypes operate independently of each other. Operation for each user is the same as for standard FOCAL, with the exception of disk saving and calling, as described above, and the changes described below:

- 1) The LIBRARY DELETE command is automatically available on both teletypes, as are the extended functions. The functions may be deleted on either terminal without affecting the other.
- 2) The OPTION M command is illegal on TTY 1, and will result in an error message.
- 3) The high-speed reader may be used by either user. Reader usage is determined by Switch Register bit 0 on the computer console. If it is set to 0 it may be used by User 0, and if it is set to 1 it is used by User 1. If one user attempts to use the reader and the switch is set to the other user, FOCAL will proceed as if the reader has run out of tape, and will swap to keyboard input for that user; the other user will experience no difficulty.
- 4) High-speed punch usage is not switched. The punch may be used by either user; however, it is important that only one user attempt to use it at any one time. Attempts at simultaneous use will have unpredictable results.

Note: The I/O swapping routines have been improved in this Two-User version. In particular, FOCAL 5/69 does not have the problem of the reader operating slowly and roughly when the other user is active.

DECIMAL ASCII CODES FOR FOCAL, 5/69

Code	Character	Code	Character	Code	Character
128	CTRL/SHFT/P (leader)	172	,	216	X
129	CTRL/A	173	-	217	Y
130	CTRL/B	174	.	218	Z
132	CTRL/D	175	/	219	[
133	CTRL/E	176	∅	220	\
134	CTRL/F	177	1	221]
135	CTRL/G (bell)	178	2	222	↑
136	CTRL/H	179	3	223	←
137	CTRL/I	180	4		
138	line feed	181	5	255	rubout
139	CTRL/K	182	ó		
140	CTRL/L	183	7		
141	return	184	8		
142	CTRL/N	185	9		
143	CTRL/O	186	:		
144	CTRL/P	187	;		
145	CTRL/Q	188	<		
146	CTRL/R	189	=		
147	CTRL/S	190	>		
148	CTRL/T	191	?		
149	CTRL/U	192	@		
150	CTRL/V	193	A		
151	CTRL/W	194	B		
152	CTRL/X	195	C		
153	CTRL/Y	196	D		
154	CTRL/Z	197	E		
155	CTRL/SHFT/K	198	F		
156	CTRL/SHFT/L	199	G		
157	CTRL/SHFT/M	200	H		
158	CTRL/SHFT/N	201	I		
159	CTRL/SHFT/O	202	J		
160	space	203	K		
161	!	204	L		
162	"	205	M		
163	#	206	N		
164	\$	207	O		
165	%	208	P		
166	&	209	Q		
167	'	210	R		
168	(211	S		
169)	212	T		
170	*	213	U		
171	+	214	V		
		215	W		

Codes 224 to 250 are for lower-case letters on some teletypes. On most terminals, printing these characters will result in characters identical to 192 through 218.

Codes 251 to 254 vary on different teletypes.

FOCAL considers the printing characters to be codes 160 through 223.

Code 131 (CTRL/C) is FOCAL's break character.

The characters [, \ and] are generated by SHFT/K, SHFT/L and SHFT/M respectively.

ERROR DIAGNOSTICS OF FOCAL, 5/69

Messages in these forms:

?nn.nn Error in direct command.
 ?nn.nn@ nn.nn Error number@step number.

- ?00.00 - Manual restart, or restart from monitor.
- ?01.00 - Interrupted by CTRL/C.
- ?01.34 - Illegal group zero usage.
- ?01.42 - Line number with zero fractional part.
- ?01.113 - Error in step number.
- ?02.45 - Nonexistent step referenced by DO.
- ?02.63 - Nonexistent group referenced by DO.
- ?02.81 - Push-down list overflowed. May be too many variables, expression too complex, or DO's nested too deeply (something DOing itself?).
- ?03.10 - Nonexistent line referenced by GOTO or IF.
- ?03.32 - Illegal command.
- ?04.01 - IF not followed by parenthesized expression.
- ?04.26 - Left of = in error in FOR or SET.
- ?04.35 - Excess right parentheses in FOR or SET.
- ?04.42 - Bad format in FOR
- ?05.26 - Excess right parentheses or bad format in TYPE.
- ?05.61 - Illegal or nonexistent step number after MODIFY.
- ?06.07 - Illegal use of function or number.
- ?06.56 - Undefined variable used in expression.
- ?06.62 - Variable storage exceeded.
- ?07.12 - Comma or operator missing in an expression.
- ?07.32 - Operator missing before left parenthesis.
- ?07.108 - Double operators in an expression.
- ?07.123 - Function name not immediately followed by parenthesized expression.
- ?08.02 - Illegal function name.
- ?08.59 - Mismatched parentheses.
- ?11.18 - Input buffer overflowed. Something typed on keyboard or read from teletype reader when FOCAL was not ready for it.
- ?12.44 - Input line too long, or storage filled by program.
- ?13.57 - OPTION M attempted on teletype 1.
- ?22.46 - Zero or negative argument for FLOG(X).
- ?23.11 - Double periods in a number.
- ?24.22 - Fixed-point number too large.
- ?26.42 - Illegal option name.
- ?26.113 - Illegal LIBRARY command
- ?28.123 - Division by zero or zero to a negative power.
- ?30.21 - Raising to a power greater than 2047 or less than -2048
- ?30.52 - Square root of a negative number.
- ?31.127 - Unavailable command or function, illegal character, or miscellaneous error.

THIS DECTAPE (LINCTAPE) CONTAINS THE SOURCE AND BINARY FILES FOR EDWARD A TAFT III'S FOCAL 5/69 (DECUS: FOCAL8-52A) AND AN 8K OVERLAY (DECUS: FOCAL8-189) DEVELOPED BY MAGNUS LUNDIN. THIS TAPE HAS BEEN SUBMITTED INDEPENDENTLY FOR THE CONVENIENCE OF PS/8-OS/8 PROGRAMMERS AND HENCE MIGHT NOT REFLECT CHANGES MADE BY THE AUTHORS TO THESE PROGRAMS AT A LATER DATE. THE FILEDATE INDICATES THE VERSION INCLUDED. MINOR EDITING OF THE ORIGINAL PDP-10 SOURCE HAS BEEN PERFORMED SO THAT ASSEMBLY MAY BE DONE WITH PAL8. THE PROCEDURE IS GIVEN ON THE FIRST PAGE OF THE SOURCE WHICH MAY BE LISTED ON THE TTY WITH PIP. NOTE THAT THE BINARY FILES AS WELL AS A CORE-IMAGE FILE ARE INCLUDED SO THAT ASSEMBLY IS ONLY NECESSARY FOR PURPOSES OF MODIFICATION. THE LISTING FILES MAY BE PROCESSED WITH CREF IF DESIRED.

CONTENTS:

WRITE .UP	7/15/73	THIS FILE
FOCAL .SV	7/25/72	SINGLE-USER, 8K, EXTENDED FUNCTIONS
FOCAL5.69	7/25/72	FOCAL LANGUAGE PROCESSOR (SOURCE)
FLOAT5.69	7/25/72	ARITHMETIC PACKAGE (SOURCE)
EXTND5.69	7/25/72	SIN, COS, ATN, LOG, EXP (SOURCE)
8KOV5.69	6/29/71	SINGLE USER 8K OVERLAY (SOURCE)
2USER5.69	7/25/72	TWO USER PATCH (SOURCE)
FOCAL .BN	7/25/72	MAIN BINARY FILE
EXTND .BN	7/25/72	EXTENDED FUNCTIONS
8KOV5 .BN	6/29/71	8K PATCH
2USER .BN	7/25/72	DUAL TTY
FOCAL .LS	7/25/72	LISTING
EXTND .LS	7/25/72	"
8KOV5 .LS	6/29/71	"
2USER .LS	7/25/72	"

JIM VAN ZEE
 UNIV. OF WASH.
 DEPT. OF CHEM.
 SEATTLE, WASH.
 98195

/**** FOCAL 5/69 ****

PAL8-V7 7/25/72 PAGE 1

/**** FOCAL 5/69 ****

/E.A.TAFT - REVISION OF FOCALW 8/68

/EAT/ 25-JUL-72

/ASSEMBLY INSTRUCTIONS FOR DECUS DECTAPE VERSION:

/INPUT FILES:

```
/          FOCAL5.69          FOCAL LANGUAGE PROCESSOR
/          FLOAT5.69          FLOATING POINT PACKAGE
/          EXTND5.69          EXTENDED FUNCTION PACKAGE
/          8KOVRS.69          8K SINGLE USER PATCH
/          2USERS.69          2-USER OVERLAY (8K SYSTEM)
```

/ASSEMBLY USING PAL8-V7 (ASSUMES INPUT FILES ON UNIT 1)

```
/          .AS DTA1 DSK
/          .R PAL8
/          *FOCAL, FOCAL*FOCAL5.69, FLOAT5.69
/          .R PAL8
/          *EXTND, EXTND*EXTND5.69
/          .R PAL8
/          *8KOVRS, 8KOVRS*8KOVRS.69
/          .R PAL8
/          *2USER, 2USER*2USERS.69
```

/EXAMPLE BELOW LOADS AND SAVES 8K SINGLE USER VERSION WITH
/EXTENDED FUNCTIONS:

```
/          .R ABSLDR
/          *FOCAL, EXTND, 8KOVRS
/          .SA SYS FOCAL
```

/CALLING SEQUENCE:

```
/          .R FOCAL
/
/          C-FOCAL 5/69, 8K, F
/          -
```

