



DECUS

PROGRAM LIBRARY

DECUS NO.	FOCAL8-100
TITLE	ADDITIONS TO FOCAL W
AUTHOR	Alan S. Fields
COMPANY	Naval Ship Research and Development Laboratory Annapolis, Maryland
DATE	October 8, 1969
SOURCE LANGUAGE	FOCAL

ADDITIONS TO FOCAL W

DECUS Program Library Write-up

DECUS No. FOCAL8-100

These are additions to FOCAL W, of 8/68, for both 4 and 8K systems. Assembly language listings are included as Appendices.

Appendix 1 gives the simple patch required to permit group numbers to go up to 31.99 rather than 15.99. This may be useful only with 8K. Included in Appendix 1 is the symbol table used by PAL-3 for all the additions.

Appendix 2 gives a simple routine which on repeated calls turns the echo on and off. This simplifies "read-in" of data tapes if only a low speed reader is available. The routine uses the DXS function slots in the tables. Systems with scopes will have to provide a change. The routine occupies unused locations in the floating point package. For a tape with N data points, programming can be as follows:

```
2.1 S X = FECO ( ) ; A X(1); F I = 1, N; A X (I)
```

```
2.2 S X = FECO ( )
```

Note that the extra ask statement is for the first = sign, which serves as a terminator. It is assumed that the data is prepared by a FOCAL routine with no spurious carriage returns (a listing is produced separately). Spaces are a convenient leader and trailer.

Appendix 3 gives a patch for 8K FOCAL, which stores dimensioned, 3-word floating point arrays in field 1. The routine uses the NEW slots on the lists and occupies space for normal variables as well as extra room in the floating point package.

As seen on the explanation page in Appendix 3, an initialization statement is employed to set a pointer to the next available (usually for text) address (BUFR). Each array must then be dimensioned. This updates the pointer and sets a FOCAL variable, one for each array, referred to as the array variable, to hold both the start of the array and its dimension. Statements are then available for storing and retrieving data within the arrays.

The routine may be called recursively to any level. The following example stores the sum of arrays A and B in array S:

```
1.1 F I = 0, N; S X = FARY(S,I, FARY(A,I) + FARY(B,I) )
```

Arrays A, B and S were dimensioned N or greater.

The Appendix also describes error messages for the routine. Note that the data is lost on any change of text or alteration of an array variable.

Appendix 4 gives a simple routine for running a CALCOMP from FOCAL. The routine uses the DIS slots in the tables and occupies normal variable space. It must be loaded with different origins depending on what extended functions are retained at FOCAL initialization time. The

Appendix and tapes are for all extended functions. This can be used with no or some extended functions with subsequent core wastage.

The routine includes the DEC plotting 8-12-U. The rest is simply interface between FOCAL and this routine. Thus, no negative numbers, X greater than 2047, or Y greater than 1023 (for a .01" plotter) can be employed.

The opinions and assertions made in this paper are those of the author and are not to be construed as official or reflecting the views of the Department of the Navy or the Naval service at large.

CLEARED FOR OPEN PUBLICATION 30 JAN 1970 - CHIEF OF NAVAL MATERIAL (PA)

APPENDIX 1

Group Number Patch

/ASF 7/69

/FOCAL PATH TO PERMIT LINE NUMBERS 1.01 TO 31.99

/(INSTEAD OF ONLY TO 15.99)

*2261

2261 7041 CIA

2262 7100 CLL

2263 7000 NOP

*2267

2267 7630 SZL CLA

/SYMBOL TABLE FOR FOCAL ADDITIONS

/FOCAL ADDRESSES

BOTTOM=27
FNTABF=376
FNTABL=2173
PUSHJ=4501
EVAL=1603
INTEGER=52
FINT=4407
FLAC=44
BUFR=134
CHAR=142
ERROR2=4526
ERROR3=4526
END=76
ERROR4=4526
PUSHA=4503
POPA=1413
SPNOR=4521
EFUN31=100
PT1=154
PUSHF=4504
POPF=4505
C200=15

/ANALOG TO DIGITAL CONVERTER INSTRUCTIONS

ADSF=6531
ADCV=6532
ADRB=6534

/MULTIPLEXER INSTRUCTIONS

SMF=6541
MC1=6542
MC2=6544

/PLOTTER INSTRUCTIONS

PLSF=6501
PLPU=6504
PLDU=6512
PLPL=6521
PLPD=6524
PLCF=6502
PLPR=6511
PLDD=6514
PLUD=6522

/EXTENDED MEMORY INSTRUCTIONS

CDF0=6201
CDF1=6211
FIXTAB
PAUSE

APPENDIX 2

Echo Switch

/ASF 7/69

/FOCAL ADDITION TO SWITCH ECHO

*FNTABF+5
~~0403~~ 6571 FECO /REPLACES DXS
 *FNTABL+5
~~2200~~ 2551 2551 /MISH-MASH CODE FOR ECO (SEE EFUN2)

/S X=FECO() SWITCHES BETWEEN ECHO AND NO
 /ECHO TO PERMIT PAPER TAPE INPUT
 / ECHO REQUIRES 4301, NON ECHO REQUIRES ~~7000~~ IN 2477
 / $4301 + 7000 = A+B = 3301 = S$
 / IF PRESENT CODE IS A, $-A+(A+B)=B$
 / IF PRESENT CODE IS B, $-B+(A+B)=A$
 / THUS, THE CODE IS CHANGED BY ADDING MINUS THE
 / PRESENT CODE TO THE SUM OF THE TWO.

*6571
 6571 1777 FECO, TAD I ADDR /GET PRESENT CONTROL
 6572 ~~7041~~ CIA
 6573 1376 TAD S
 6574 3777 DCA I ADDR /DEPOSIT NEW CODE
 6575 ~~5500~~ JMP I EFUN3I
 6576 3301 S, 3301
 6577 2477 ADDR, 2477

ADDR 6577
 FECO 6571
 S 6576

APPENDIX 3

Array Addition

/A.S.FIELDS 8/69

/FOCAL ADDITION FOR ARRAYS IN 8K

/EXPLANATION OF FARY:

/S X=FARY(-1)

/ INITIALIZATION STATEMENT
/ MUST BE USED AFTER ANY ERASE, MODIFICATION OR
/ CHANGE IN ARRAY VARIABLES
/ INITIALIZES ARRAY POINTER TO BOTTOM OF TEXT
/ X IS A DUMMY VARIABLE

/S A=FARY(1, N)

/ DIMENSION STATEMENT
/ RESERVES N+1 SETS OF THREE LOCATIONS FOR A ARRAY
/ SETS FOCAL VARIABLE A SO THAT;
/ EXPONENT= \emptyset
/ MSH=PRESENT LOCATION OF ARRAY POINTER
/ (WHICH IS START OF STORAGE OF A ARRAY IN MEM1)
/ LSH=N - AN INTEGER GREATER THAN 1, LESS THAN 1300
/ ****A MUST NOT BE ALTERED****

/S X=FARY(A, N, Z)

/ STORE STATEMENT
/ STORES Z IN THE N'TH SET OF THREE LOCATION
/ RESERVED FOR A ARRAY
/ N MUST BE LESS THAN OR EQUAL TO THE DIMENSION FOR
/ THE A ARRAY
/ X IS A DUMMY VARIABLE

/S Z=FARY(A, N)

/ GET STATEMENT
/ SETS Z EQUAL TO THE CONTENTS OF THE N'TH SET
/ OF THREE LOCATIONS IN THE A ARRAY
/ N MUST BE LESS THAN OR EQUAL TO THE DIMENSION FOR
/ THE A ARRAY

/FARY ERROR MESSAGES

/ ?13.;4 TEXT HAS CHANGED - ARRAYS NO LONGER AVAILABLE
/ ?13.51 NOT ENOUGH ROOM FOR LAST ARRAY DIMENSIONED
/ ?13.42 NO INDEX IN DIMENSION STATEMENT
/ ?13.75 NO INDEX IN GET OR STORE STATEMENT
/ ?13.83 INDEX BEYOND DIMENSION OR ARRAY VARIABLE CHANGED
/ ?13.89 ARRAY VARIABLE CHANGED
/ ?13.31 FIRST ARGUMENT NOT +OR- 1 OR ARRAY VARIABLE

/FARY PAGE 2

```
*END
0076 3372 BUFBEQ
      *FNTABF + 15
0413 3232 FARY
      *FNTABL + 15
2210 2601 2601 /MISH-MASH CODE FOR ARY - SEE EFUN2
      *3232 /STARTS AT END OF FOCAL (VAR. STOR. START)
3232 1044 FARY, TAD FLAC /CHECK EXPONENT
3233 7750 SPA SNA CLA
3234 5276 JMP GORS /NOT +OR- 1
3235 1046 TAD FLAC+2 /CHECK LOW ORDER
3236 7640 SZA CLA
3237 4526 ERROR2 /NOT +OR- 1 OR LEGITIMATE ARRAY
      /VARIABLE (AS EXPONENT WAS +)
3240 1045 TAD FLAC+1 /CHECK HIGH ORDER
3241 7106 CLL RTL /LOOK AT ALL BUT BIT 1
3242 7650 SNA CLA
3243 5251 JMP DIM /+1*2↑N, TREAT AS 1 (N IS +)
```

/RESET ROUTINE

```
3244 1134 RES, TAD BUFR /GET BOTTOM OF TEXT
3245 3350 DCA TOP /STORE FOR CHECKING
3246 1350 TAD TOP
3247 3347 DCA LOC / INITIALIZE ARRAY POINTER
3250 5500 JMP I EFUN3I
```

/FARY PAGE 3

/DIMENSION ARRAY

3251	4355	DIM,	JMS TCMA	
3252	4526		ERROR4	/NO COMMA AFTER 1
3253	1347		TAD LOC	
3254	4503		PUSHA	/STORE LOC
3255	4754		JMS I LGET3N	/GET DIMENSION
3256	7100		CLL	
3257	1347		TAD LOC	
3260	1351		TAD P3	/ADD 1 MORE LOC FOR 0'TH TERM
3261	1015		TAD C200	/ADD MAX. LOCATION
3262	7430		SZL	/ENOUGH ROOM?
3263	4526		ERROR2	/NOT ENOUGH ROOM
3264	1352		TAD MDOT	
3265	3347		DCA LOC	/UPDATE LOC
3266	3044		DCA FLAC	/CLEAR EXPONENT
3267	1413		POPA	/GET START ADDRESS OF THIS ARRAY
3270	7104		CLL RAL	/CHECK FOR 4000
3271	7450		SNA	
3272	7100		CLL	
3273	7010		RAR	/CHANGE 4000 to 0
3274	3045		DCA FLAC+1	/LOAD START OF THIS ARRAY
			/FLAC+2 STILL CONTAINS DIMENSION	
3275	5500		JMP I EFUN3I	

/GET OR STORE PROBLEM

3276	3047	GORS,	DCA FLAC+3	/CLEAR OVERFLOW REGISTER
3277	1044		TAD FLAC	
3300	7650		SNA CLA	/NEED UN-NORMALIZING?
3301	5304		JMP .+3	/NO
3302	4753		JMS I LDIV2	/ROTATE ONCE
3303	5277		JMP .-4	/CHECK FOR MORE NEEDED
3304	1046		TAD FLAC+2	
3305	4503		PUSHA	/STORE DIMENSION
3306	1045		TAD FLAC+1	
3307	7450		SNA	/CORRECT FOR 4000 TO 0
3310	7130		CLL CML RAR	
3311	4503		PUSHA	/STORE START LOCATION
3312	4355		JMS TCMA	
3313	4526		ERROR4	/NO COMMA
3314	4754		JMS I LGET3N	/GET DESIRED LOCATION
3315	1413		POPA	/ADD LOCATION ON LIST
3316	3346		DCA ELOC	/STORE LOCATION TO BE ADDRESSED
3317	1046		TAD FLAC+2	/GET DESIRED LOCATION
3320	7041		CIA	
3321	1413		POPA	/ADD DIMENSION
3322	7710		SPA CLA	
3323	4526		ERROR2	/BEYOND DIMENSION

/FARY PAGE 4

/DIMENSION ARRAY

3324	1347	TAD LOC	
3325	7041	CIA	
3326	7100	CLL	
3327	1346	TAD ELOC	
3330	7630	SZL CLA	
3331	4526	ERROR2	/BEYOND LISTS - SOMETHING IS WRONG
3332	1346	TAD ELOC	
3333	4503	PUSHA	/STORE ELOC
3334	4355	JMS TCMA	
3335	5343	JMP GET	/NO THIRD VARIABLE, GET PROBLEM OR ERROR
		/STORE PROBLEM	
3336	4501	PUSHJ	
3337	1602	EVAL-1	/GET VARIABLE TO BE STORED
3340	1413	POPA	/GET STORAGE START LOCATION
3341	5742	JMP I LMOVES	
3342	5755	LMOVES, MOVES	
		/GET PROBLEM	
3343	1413	GET, POPA	/GET STORAGE START LOCATION
3344	5745	JMP I LMOVEG	
3345	7154	LMOVEG, MOVEG	
		/STORAGE	
3346	0000	ELOC, 0	
3347	0000	LOC, 0	
3350	0000	TOP, 0	
3351	0003	P3, 3	
3352	7600	MBOT, -200	
3353	6750	LDIV2, 6750	
3354	6171	LGET3N, GET3N	
		/ROUTINE TO LOOK FOR COMMA AND CHECK TEXT	
3355	0000	TCMA 0	
3356	1134	TAD BUFR	/CHECK THAT TEXT HASN'T CHANGED
3357	7041	CIA	
3360	1350	TAD TOP	
3361	7640	SZA CLA	
3362	4526	ERROR2	/TEXT CHANGED - ARRAYS UNAVAILABLE
3363	4521	SPNOR	/MOVE ON TO NEXT TEXT CHARACTER
3364	1142	TAD CHAR	
3365	1371	TAD MCMA	
3366	7650	SNA CLA	
3367	2355	ISZ TCMA	/EXIT WITH SKIP IF COMMA
3370	5755	JMP I TCMA	

/FARY PAGE 5

3371 7524 MCMA, -254
BUFBEQ=.

/THESE ROUTINES ARE FIT INTO UNUSED LOCATIONS IN
/THE FLOATING POINT PACKAGE TO PRESERVE ROOM.

/ROUTINE TO EVALUATE NEXT ARGUMENT, CONVERT TO INTEGER
/LEAVES 3 TIMES THE RESULT IN AC, THE INTEGER IN FLAC+2
*6171

6171 0000 GET3N, 0
6172 4501 PUSHJ
6173 1602 EVAL-1
6174 4452 JMS I INTEGER
6175 7104 RAL CLL
6176 1046 TAD FLAC+2
6177 5771 JMP I GET3N
*7154 /GETS STORED VALUE

7154 3373 MOVEG, DCA LOCG
7155 1372 TAD M3G
7156 3374 DCA CNTG
7157 1375 TAD GOP
7160 3363 DCA EOP2
7161 6211 CDF1
7162 1773 TAD I LOCG
7163 0000 EOP2, 0
7164 2363 ISZ EOP2
7165 2373 ISZ LOCG
7166 2374 ISZ CNTG
7167 5362 JMP EOP2-1
7170 6201 CDF 0
7171 5500 JMP I EFUN31
7172 7775 M3G, -3
7173 0000 LOCG, 0
7174 0000 CNTG, 0
7175 3044 GOP, DCA FLAC
*5755 /STORES VALUE

5755 3373 MOVES, DCA LOCS
5756 1374 TAD M3S
5757 3375 DCA CNTS
5760 1376 TAD SOP
5761 3363 DCA EOP1
5762 6211 CDF1

/FARY PAGE 6

5763	0000	EOP1,	∅
5764	3773		DCA I LOCS
5765	2373		ISZ LOCS
5766	2363		ISZ EOP1
5767	2375		ISZ CNTS
5770	5363		JMP EOP1
5771	6201		CDF∅
5772	5500		JMP I EFUN3I
5773	0000	LOCS,	∅
5774	7775	M3S,	-3
5775	0000	CNTS,	∅
5776	1044	SOP,	TAD FLAC

BUFBEG	3372
CNTG	7174
CNTS	5775
DIM	3251
ELOC	3346
EOP1	5763
EOP2	7163
FARY	3232
GET	3343
GET3N	6171
GOP	7175
GORS	3276
LDIV2	3353
LGET3N	3354
LMOVEG	3345
LMOVES	3342
LOC	3347
LOCG	7173
LOCS	5773
MBOT	3352
MCMA	3371
MOVEG	7154
MOVES	5755
M3G	7172
M3S	5774
P3	3351
RES	3244
SOP	5776
TCMA	3355
TOP	3350

APPENDIX 4

Plotting Addition

/ASF 7/69

/ADDITION TO FOCAL FOR PLOTTING ON THE 565

/ALL EXTENDED FUNCTIONS

```

*27
0027 4355 CPLOT-1      /TOP OF CORE
          *FNTABF+3
0401 4356 CPLOT      /FUNCTION LIST ADDITION
          *FNTABL+3
2176 2647 2647      /CODE FOR PLO
          *4356

/FOCAL CALLING STATEMENT: SET Z=FNEW(X, Y, C)
/WHERE -
/Z=DUMMY VARIABLE
/X=NEW X POINT
/Y=NEW Y POINT
/C=CONTROL
/
/      -1 CLEARS PLOT MEMORY (OLDX AND OLDY =0)
/      0 PLOT WITH PEN DOWN
/      1 PLOT WITH PEN UP (MOVE WITHOUT PLOTTING)

4356 6002 CPLOT,      IOF
4357 4452          JMS I INTEGER      /SET UP X
4360 3365          DCA PX
4361 4371          JMS NGET           /GET Y
4362 3366          DCA PY
4363 4371          JMS NGET           /GET CONTROL
4364 4777          JMS I LPLOT        /PLOT
4365 0000 PX,      0           /X VALUE
4366 0000 PY,      0           /Y VALUE
4367 6001          ION
4370 5500          JMP I EFUN3I

/SUBROUTINE TO GET ARGUMENTS

4371 0000 NGET,      0
4372 4521          SPNOT      /IGNOR SPACES
4373 4501          PUSHJ
4374 1602          EVAL-1      /SET UP NUMBER
4375 4452          JMS I INTEGER
4376 5771          JMP I NGET
4377 4400 LPLOT,      PLOTX

```

```

/FOCAL PLOTTING ADDITION - PAGE 2
/SUBROUTINE PLOTX
/DIGITAL 8-12-U
/PLOT SUBROUTINE
/CALLING SEQUENCE

```

```

/   C(AC)=-1;  INITIALIZE
/   C(AC)= 0;  PLOT WITH PEN DOWN
/   C(AC)= 1;  PLOT WITH PEN UP
/           JMS PLOTX
/           X CO-ORDINATE (IN STEPS) (RETURN IF AC=-1)
/           Y CO-ORDINATE (IN STEPS)

```

```

PLOTX,  0
        SPA           /MOVE THE PEN?
        JMP PLOTA    /NO: CONTINUE
        TAD PLOTPN   /ADD PEN STATUS
        CLL RTR
        SPA CLA      /ANY CHANGE?
        JMP PLOTI    /NO: CONTINUE
        SNA CLA
        JMP .+4      /LOWER THE PEN
        DCA PLOTPN   /RAISE THE PEN
        PLPU
        JMP .+3
        ISZ PLOTPN   /LOWER THE PEN
        PLPD
        JMS PLOTWT   /WAIT FOR FLAG
        JMP PLOTI    /CONTINUE
PLOTA,  CLA
        PLPU           /RAISE THE PEN
        DCA PLOTPN
        DCA PLOTNX   /0 TO X CO-ORDINATE
        DCA PLOTNY   /0 TO Y CO-ORDINATE
        JMS PLOTWT
        JMP I PLOTX

```

/FOCAL PLOTTING ADDITION - PAGE 3

/PICK UP ARGUMENTS

PLOT1,	TAD PLOTNX	/FETCH PREVIOUS X CO-ORDINATE
	CIA CLL	
	TAD I PLOTX	/FORM NX-NPX
	SNL	/L=∅: NX<NPX
	CIA	
	DCA PLOTDX	/ABSOLUTE VALUE OF DIFFERENCE
	RAL	
	DCA PLOTMV	/SAVE SIGN BIT
	TAD I PLOTX	/SET NEW
	DCA PLOTNX	/PREVIOUS X
	ISZ PLOTX	/INCREMENT POINTER
	TAD PLOTNY	/ FETCH PREVIOUS Y CO-ORDINATE
	CIA CLL	
	TAD I PLOTX	/FORM NY-NPY
	SNL	/<=∅: NPY<NY
	CIA	
	DCA PLOTDY	/ABSOLUTE VALUE OF DIFFERENCE
	TAD PLOTMV	/SAVE SIGN BIT
	RAL	/BIT 1∅(1)= DRUM-DOWN(POSITIVE)
	DCA PLOTMV	/BIT 11(1)=PEN-LEFT (POSITIVE)
	TAD I PLOTX	/SET NEW
	DCA PLOTNY	/PREVIOUS Y
	ISZ PLOTX	/INCREMENT POINTER
	TAD PLOTDX	
	CIA CLL	
	TAD PLOTDY	
	SNL CLA	/L=∅: DELTA Y < DELTA X
	JMP PLOT2	
	TAD PLOTDX	/REVERSE NUMBERS
	DCA PLOTNA	
	TAD PLOTDY	
	DCA PLOTDX	
	TAD PLOTNA	
	DCA PLOTDY	
	IAC	/SET MAJOR MOTION
	AND PLOTMV	/INSTRUCTION
	TAD PLOTT1	
	JMP .+4	

/FOCAL PLOTTING ADDITION - PAGE 4

PLOT2,	TAD PLOTMV CLL RAR TAD PLOTT2 DCA PLOTNA TAD I PLOTNA DCA PLOT4 TAD PLOTMV TAD PLOTT3 DCA PLOTMV TAD I PLOTMV DCA PLOTD3 TAD PLOTDX CLL RAR DCA PLOTNA TAD PLOTDX CMA DCA PLOTMV	/SET COMBINED MOTION
PLOT3,	ISZ PLOTMV SKP JMP I PLOTX TAD PLOTNA TAD PLOTDY DCA PLOTNA TAD PLOTNA CMA CLL TAD PLOTDX SZL CLA JMP PLOT4	/ALL DONE
PLOTDB,	Ø TAD PLOTDX CIA TAD PLOTNA DCA PLOTNA JMS PLOTWT JMP PLOT3	/SINGLE MOTION /COMBINED MOTION
PLOT4,	Ø	
PLOTT1,	JMP .-3 .+1 PLPR PLPL	/PEN-RIGHT /PEN-LEFT
PLOTT2,	.+1 PLDU PLDD	/DRUM-UP /DRUM-DOWN

/FOCAL PLOTTING ADDITION - PAGE 5

```

PLOTT3, .+1
        PLDU PLPR /UP-RIGHT
        PLUD PLPL /UP-LEFT
        PLDD PLPR /DOWN-RIGHT
        JMS .+1 /DOWN-LEFT
        Ø
        PLDD
        PLPL
        JMP I .-3
PLOTPN, Ø
PLOTNX, Ø
PLOTNY, Ø
PLOTDX, Ø
PLOTDY, Ø
PLOTNA, Ø
PLOTMV, Ø

PLOTWT, Ø
        PLSF /WAIT FOR DONE FLAG
        JMP .-1 /NOT YET
        PLCF /CLEAR FLAG
        JMP I PLOTWT /EXIT

```

\$

```

CPLOT 4356
LPLOT 4377
NGET 4371
PLOTA 442Ø
PLOTDB 4531
PLOTDX 4564
PLOTDY 4565
PLOTMV 4567
PLOTNA 4566
PLOTNX 4562
PLOTNY 4563
PLOTPN 4561
PLOTT1 4542
PLOTT2 4545
PLOTT3 455Ø
PLOTWT 457Ø
PLOTX 44ØØ
PLOT1 4427
PLOT2 4475
PLOT3 4516
PLOT4 454Ø
PX 4365
PY 4366

```