

FFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFF
FFF
FFF
FFF
FFF
FFF
FFF
FFFFFFFFFFFFFF
FFFFFFFFFFFFFF
FFFFFFFFFFFFFF
FFF
FFF
FFF
FFF
FFF
FFF
FFF
FFF
FFF
FFF
FFF
FFF

111
111
111
111111
111111
111111
111
111
111
111
111
111
111
111
111
111
111
111
111
111
1111111111
1111111111
1111111111

111
111
111
111111
111111
111111
111
111
111
111
111
111
111
111
111
111
111
111
111
111
1111111111
1111111111
1111111111

AAAAAAAAAA
AAAAAAAAAA
AAAAAAAAAA
AAA
AAA
AAA
AAA
AAA
AAA
AAA
AAA
AAA
AAA
AAA
AAA
AAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAA
AAA
AAA
AAA
AAA
AAA
AAA
AAA
AAA
AAA

```

EEEEEEEEEE  NN      NN  TTTTTTTTTT  EEEEEEEEEE  RRRRRRRR
EEEEEEEEEE  NN      NN  TTTTTTTTTT  EEEEEEEEEE  RRRRRRRR
EE          NN      NN      TT          EE          RR      RR
EE          NN      NN      TT          EE          RR      RR
EE          NNNN     NN      TT          EE          RR      RR
EE          NNNN     NN      TT          EE          RR      RR
EEEEEEEEEE  NN  NN  NN      TT          EEEEEEEEE  RRRRRRRR
EEEEEEEEEE  NN  NN  NN      TT          EEEEEEEEE  RRRRRRRR
EE          NN      NNNN     TT          EE          RR  RR
EE          NN      NNNN     TT          EE          RR  RR
EE          NN      NN      TT          EE          RR  RR
EEEEEEEEEE  NN      NN      TT          EEEEEEEEEE  RR      RR
EEEEEEEEEE  NN      NN      TT          EEEEEEEEEE  RR      RR

```

```

LL          IIIIII  SSSSSSSS
LL          IIIIII  SSSSSSSS
LL          II      SS
LL          II      SS
LL          II      SS
LL          II      SS
LL          II      SSSSSS
LL          II      SSSSSS
LL          II      SS
LL          II      SS
LL          II      SS
LLLLLLLLLL  IIIIII  SSSSSSSS
LLLLLLLLLL  IIIIII  SSSSSSSS

```

.....

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57

```

0001 0 MODULE ENTER (
0002 0
0003 0     LANGUAGE (BLISS32),
0004 0     IDENT = 'V04-000'
0005 1 BEGIN
0006 1
0007 1
0008 1 *****
0009 1 *
0010 1 *  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0011 1 *  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0012 1 *  ALL RIGHTS RESERVED.
0013 1 *
0014 1 *  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0015 1 *  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0016 1 *  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0017 1 *  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0018 1 *  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0019 1 *  TRANSFERRED.
0020 1 *
0021 1 *  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0022 1 *  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0023 1 *  CORPORATION.
0024 1 *
0025 1 *  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0026 1 *  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0027 1 *
0028 1 *
0029 1 *****
0030 1
0031 1 ++
0032 1
0033 1 FACILITY: F11ACP Structure Level 1
0034 1
0035 1 ABSTRACT:
0036 1
0037 1     This routine enters the given file name in the specified directory.
0038 1
0039 1 ENVIRONMENT:
0040 1
0041 1     STARLET operating system, including privileged system services
0042 1     and internal exec routines.
0043 1
0044 1 --
0045 1
0046 1
0047 1 AUTHOR: Andrew C. Goldstein, CREATION DATE: 30-Mar-1977 15:05
0048 1
0049 1 MODIFIED BY:
0050 1
0051 1     V03-001 LMP0241 L. Mark Pilant, 26-Apr-1984 10:44
0052 1     Include the FIB in the MAKE_NAMEBLOCK routine call.
0053 1
0054 1     V02-002 ACG0238 Andrew C. Goldstein, 29-Dec-1981 19:37
0055 1     Invalidate RMS caches when a directory is superseded
0056 1
0057 1     V02-001 ACG0167 Andrew C. Goldstein, 7-May-1980 18:49

```

ENTER
V04-000

⁶
16-Sep-1984 01:00:47
14-Sep-1984 12:29:30

VAX-11 Bliss-32 V4.0-742
DISK\$VMMASTER:[F11A.SRC]ENTER.B32;1 Page (1)

EXT
V04

```
.. 58      0058 1 |  
.. 59      0059 1 |**  
.. 60      0060 1 |  
.. 61      0061 1 |  
.. 62      0062 1 LIBRARY 'SYSSLIBRARY:LIB.L32';  
.. 63      0063 1 REQUIRE 'SRCS:FCPDEF.B32';
```

Previous revision history moved to F11A.REV

.....

```

65 0378 1 GLOBAL ROUTINE ENTER (ABD, FIB, RESULT_LENGTH, RESULT) : NOVALUE =
66 0379 1
67 0380 1 ++
68 0381 1
69 0382 1 FUNCTIONAL DESCRIPTION:
70 0383 1
71 0384 1 This routine enters the given file name in the specified directory.
72 0385 1
73 0386 1 CALLING SEQUENCE:
74 0387 1 ENTER (ARG1, ARG2, ARG3, ARG4)
75 0388 1
76 0389 1 INPUT PARAMETERS:
77 0390 1 ARG1: address of buffer descriptor packet
78 0391 1 ARG2: address of FIB of operation
79 0392 1
80 0393 1 IMPLICIT INPUTS:
81 0394 1 NONE
82 0395 1
83 0396 1 OUTPUT PARAMETERS:
84 0397 1 ARG3: address of longword to receive length of result string
85 0398 1 ARG4: address of result string buffer
86 0399 1
87 0400 1 IMPLICIT OUTPUTS:
88 0401 1 DIR_RECORD: record number of new directory entry
89 0402 1
90 0403 1 ROUTINE VALUE:
91 0404 1 NONE
92 0405 1
93 0406 1 SIDE EFFECTS:
94 0407 1 directory altered, result string and length written into buffer packet
95 0408 1
96 0409 1 --
97 0410 1
98 0411 2 BEGIN
99 0412 2
100 0413 2 MAP
101 0414 2 ABD : REF BBLOCKVECTOR [,ABD$C_LENGTH],
102 0415 2 ! descriptor list arg
103 0416 2 FIB : REF BBLOCK, ! FIB argument
104 0417 2 RESULT : REF VECTOR [,BYTE]; ! result string arg
105 0418 2
106 0419 2 LOCAL
107 0420 2 NAMEBLOCK : BBLOCK [NMB$C_LENGTH],
108 0421 2 ! RAD-50 name block
109 0422 2 COUNT, ! count of name string
110 0423 2 STRINGP, ! address of name string
111 0424 2 RECADDR : REF BBLOCK; ! pointer to found directory entry
112 0425 2
113 0426 2 EXTERNAL
114 0427 2 CLEANUP_FLAGS : BITVECTOR, ! cleanup action flags
115 0428 2 HIGHEST_VERSION : SIGNED WORD, ! highest version encountered
116 0429 2 LOWEST_VERSION : SIGNED WORD, ! lowest version number encountered
117 0430 2 FIRST_FREE, ! record number of free slot in directory
118 0431 2 DIR_RECORD, ! record number of entry created
119 0432 2 SUPER_FID : BBLOCK; ! file ID of superseded file
120 0433 2
121 0434 2 EXTERNAL ROUTINE

```

```

122 0435      PMS_START SUB,           ! start subfunction metering
123 0436      PMS_END SOB,         ! end subfunction metering
124 0437      DIR_ACCESS,        ! access the directory
125 0438      MAKE_NAMEBLOCK,    ! construct name block
126 0439      DIR_SCAN,          ! search the directory
127 0440      EXTEND_DIR,        ! extend the directory
128 0441      DIRGET,           ! read a directory record
129 0442      DIRPUT,           ! write a directory record
130 0443      UPDATE_DIRSEQ,     ! update volume directory sequence
131 0444      MAKE_STRING,      ! convert directory entry to string
132 0445      RETURN_DIR;       ! return data to buffer packet
133 0446
134 0447
135 0448      ! Start metering for this subfunction.
136 0449
137 0450
138 0451      PMS_START_SUB (PMS_ENTER);
139 0452
140 0453      ! The file ID to be entered must be non-zero
141 0454
142 0455
143 0456      IF .FIB[FIB$W_FID_NUM] EQL 0
144 0457      THEN ERR_EXIT (SS$BADPARAM);
145 0458
146 0459      ! Find the name string in the buffer packet. Construct the RAD-50 name block
147 0460      ! from it and the other data in the FIB. Mask out the wild card bits, since
148 0461      ! they are ignored.
149 0462
150 0463
151 0464      COUNT = .ABD[ABD$C_NAME, ABD$W_COUNT];
152 0465      STRINGP = .ABD[ABD$C_NAME, ABD$W_TEXT] + ABD[ABD$C_NAME, ABD$W_TEXT] + 1;
153 0466
154 0467      MAKE_NAMEBLOCK (.FIB, .COUNT, .STRINGP, NAMEBLOCK);
155 0468      IF .NAMEBLOCK[NMBS$W_WILD]
156 0469      THEN ERR_EXIT (SS$BADFILENAME);
157 0470
158 0471      CH$MOVE (FIB$S_FID, FIB[FIB$W_FID], NAMEBLOCK[NMBS$W_FID]);
159 0472      NAMEBLOCK[NMBS$W_FLAGS] = .FIB[FIB$W_NMCTL] AND FIB$M_NEWVER;
160 0473      NAMEBLOCK[NMBS$W_CONTEXT] = 0;
161 0474
162 0475      ! Access the directory.
163 0476
164 0477
165 0478      DIR_ACCESS (.FIB, 1);
166 0479
167 0480      ! Now search the directory. If the search succeeds, we have a duplicate
168 0481      ! entry. If superseding is enabled, save away the old file ID for
169 0482      ! deletion later on. If the file being superseded is of the form
170 0483      ! xxx.DIR;1, invalidate the RMS caches by updating the directory
171 0484      ! sequence count.
172 0485
173 0486
174 0487      RECADDR = DIR_SCAN (NAMEBLOCK, 1);
175 0488      IF .RECADDR NEQ 0           ! if success, a duplicate exists
176 0489      THEN
177 0490          BEGIN
178 0491          IF NOT .FIB[FIB$V_SUPERSEDE] ! see if supersede mode

```

```

179 0492 THEN ERR_EXIT (SS$_DUPFILENAME);
180 0493
181 0494 CH$MOVE (FIB$$_FID, RECADDR[NMBSW_FID], SUPER_FID);
182 0495 CLEANUP_FLAGS[CLF_SUPERSEDE] = 1;
183 0496 CLEANUP_FLAGS[CLF_REENTER] = 1;
184 0497
185 0498 IF .RECADDR[NMBSW_TYPE] EQL %RAD50_11 'DIR'
186 0499 AND .RECADDR[NMBSW_VERSION] EQL 1
187 0500 THEN KERNEL_CALL (UPDATE_DIRSEQ);
188 0501 END
189 0502
190 0503 ! Given that the search failed, compute the new version number. Then position
191 0504 ! to the free slot found in the directory scan and insert the new entry.
192 0505 ! If no empty record has been found, the directory must be extended.
193 0506
194 0507
195 0508 ELSE
196 0509 BEGIN
197 0510 IF .NAMEBLOCK[NMBSW_VERSION] EQL 0
198 0511 OR (.NAMEBLOCK[NMBSW_NEWVER] AND .NAMEBLOCK[NMBSW_VERSION] LEQ .HIGHEST_VERSION)
199 0512 THEN NAMEBLOCK[NMBSW_VERSION] = .HIGHEST_VERSION + 1;
200 0513
201 0514 IF .FIRST_FREE EQL 0
202 0515 THEN
203 0516 BEGIN
204 0517 RECADDR = EXTEND_DIR ();
205 0518 END
206 0519
207 0520 ELSE
208 0521 BEGIN
209 0522 DIR_RECORD = .FIRST_FREE;
210 0523 RECADDR = DIRGET (.FIRST_FREE, 0);
211 0524 END;
212 0525 END;
213 0526
214 0527 ! Set the higher and lower version flags according to what the directory
215 0528 ! scan found.
216 0529
217 0530
218 0531 IF .HIGHEST_VERSION GTR .NAMEBLOCK[NMBSW_VERSION]
219 0532 THEN FIB[FIB$V_HIGHVER] = 1;
220 0533 IF .LOWEST_VERSION LSS .NAMEBLOCK[NMBSW_VERSION]
221 0534 THEN FIB[FIB$V_LOWVER] = 1;
222 0535
223 0536 ! Check the new version number for legality. Copy the new directory entry
224 0537 ! into the record and write it back.
225 0538
226 0539
227 0540 IF .NAMEBLOCK[NMBSW_VERSION] GTRU 32767
228 0541 THEN ERR_EXIT (SS$_BADFILEVER);
229 0542
230 0543 CH$MOVE (16, NAMEBLOCK, .RECADDR);
231 0544 DIRPUT (.RECADDR);
232 0545 CLEANUP_FLAGS[CLF_REMOVE] = 1;
233 0546
234 0547 ! Finally convert the name block into the resultant string.
235 0548

```

```

: 236
: 237
: 238
: 239
: 240
: 241
: 242
: 243
: 244
: 245
0549 2
0550 2 .RESULT_LENGTH = MAKE_STRING (NAMEBLOCK, .RESULT);
0551 2 KERNEL_CALL (RETURN_DIR, ..RESULT_LENGTH, .RESULT, .ABD);
0552 2
0553 2 ! Stop metering of this subfunction
0554 2 !
0555 2
0556 2 PMS_END_SUB ();
0557 2
0558 1 END;

```

! end of routine ENTER

.TITLE ENTER
.IDENT \V04-000\

```

.EXTRN CLEANUP_FLAGS, HIGHEST_VERSION
.EXTRN LOWEST_VERSION, FIRST_FREE
.EXTRN DIR_RECORD, SUPER_FID
.EXTRN PMS_START_SUB, PMS_END_SUB
.EXTRN DIR_ACCESS, MAKE_NAMEBLOCK
.EXTRN DIR_SCAN, EXTEND_DIR
.EXTRN DIRGET, DIRPUT, UPDATE_DIRSEQ
.EXTRN MAKE_STRING, RETURN_DIR
.EXTRN SYSSCMKRNL

```

.PSECT \$CODE\$,NOWRT,2

.ENTRY ENTER, Save R2,R3,R4,R5,R6,R7,R8,R9

59	0000G	CF	03FC	00000	MOVAB	HIGHEST_VERSION, R9	0378
58	00000000G	9F	9E	00007	MOVAB	@#SYSSCMKRNL, R8	
5E		28	C2	0000E	SUBL2	#40, SP	
		07	DD	00011	PUSHL	#7	0451
0000G	CF	01	FB	00013	CALLS	#1, PMS_START_SUB	
50		08	AC	00018	MOVL	FIB, R0	0456
		04	A0	0001C	TSTW	4(R0)	
			03	12	BNEQ	1\$	
			14	BF	CHMU	#20	0457
			04	00023	RET		
50		04	AC	00024	MOVL	ABD, R0	0464
52		12	A0	00028	MOVZWL	18(R0), COUNT	
51		10	A0	0002C	MOVAB	16(R0), R1	0465
50			61	3C	MOVZWL	(R1), R0	
50		01	A140	9E	MOVAB	1(R1)[R0], STRINGP	
		4001	8F	00038	PUSHR	#*M<R0,SP>	0467
			52	DD	PUSHL	COUNT	
		08	AC	0003E	PUSHL	FIB	
0000G	CF	04	F8	00041	CALLS	#4, MAKE_NAMEBLOCK	
05		11	AE	00046	BLBC	NAMEBLOCK+17, 2\$	0468
		0818	8F	0004A	CHMU	#2072	0469
			04	0C04E	RET		
56		08	AC	0004F	MOVL	FIB, R6	0471
10	6E	04	A6	00053	MOV3	#6, 4(R6), NAMEBLOCK	
	AE	14	A6	00058	BICW3	#-513, 20(R6), NAMEBLOCK+16	0472
			FDFE	8F	CLR3	NAMEBLOCK+38	0473
			26	AE	PUSHL	#1	0478
			01	DD	PUSHL	R6	
			56	DD	CALLS	#2, DIR_ACCESS	
0000G	CF	02	FB	00067			

				01	DD	0006C		PUSHL	#1	0487
			04	AE	9F	0006E		PUSHAB	NAMEBLOCK	
	0000G	CF		02	FB	00071		CALLS	#2, DIR_SCAN	
		57		50	DO	00076		MOVL	R0, RECADDR	
				34	13	00079		BEQL	4\$	0488
	05	15	A6	02	E0	0007B		BBS	#2, 21(R6), 3\$	0491
			0868	8F	BF	00080		CHMU	#2152	0492
					04	00084		RET		
0000G	CF	67		C6	28	00085	3\$:	MOV3	#6, (RECADDR), SUPER_FID	0494
	0000G	CF	00800020	8F	C8	0008B		BISL2	#8388640, CLEANUP_FLAGS	0496
	1A7A	8F	0C	A7	B1	00094		CMPW	12(RECADDR), #6778	0498
				47	12	0009A		BNEQ	9\$	
		01	0E	A7	B1	0009C		CMPW	14(RECADDR), #1	0499
				41	12	000A0		BNEQ	9\$	
				7E	D4	000A2		CLRL	-(SP)	0500
				5E	DD	000A4		PUSHL	SP	
			0000G	CF	9F	000A6		PUSHAB	UPDATE_DIRSEQ	
		68		03	FB	000AA		CALLS	#3, SYSSCMKRN	
				34	11	000AD		BRB	9\$	0488
			0E	AE	B5	000AF	4\$:	TSTW	NAMEBLOCK+14	0510
				0B	13	000B2		BEQL	5\$	
	0B	11	AE	01	E1	000B4		BBC	#1, NAMEBLOCK+17, 6\$	0511
		69	0E	AE	B1	000B9		CMPW	NAMEBLOCK+14, HIGHEST_VERSION	
				05	14	000BD		BGTR	6\$	
0E	AE	69		01	A1	000BF	5\$:	ADDW3	#1, HIGHEST_VERSION, NAMEBLOCK+14	0512
		52	0000G	CF	DO	000C4	6\$:	MOVL	FIRST_FREE, R2	0514
				07	12	000C9		BNEQ	7\$	
	0000G	CF		00	FB	000CB		CALLS	#0, EXTEND_DIR	0517
				0E	11	000D0		BRB	8\$	
	0000G	CF		52	DO	000D2	7\$:	MOVL	R2, DIR_RECORD	0522
				7E	D4	000D7		CLRL	-(SP)	0523
				52	DD	000D9		PUSHL	R2	
	0000G	CF		02	FB	000DB		CALLS	#2, DIRGET	
		57		50	DO	000E0	8\$:	MOVL	R0, RECADDR	
	0E	AE		69	B1	000E3	9\$:	CMPW	HIGHEST_VERSION, NAMEBLOCK+14	0531
				09	15	000E7		BLEQ	10\$	
		50	08	AC	DO	000E9		MOVL	FIB, R0	0532
	15	A0	80	8F	88	000ED		BISB2	#128, 21(R0)	
	0E	AE	0000G	CF	B1	000F2	10\$:	CMPW	LOWEST_VERSION, NAMEBLOCK+14	0533
				09	18	000F8		BGEQ	11\$	
		50	08	AC	DO	000FA		MOVL	FIB, R0	0534
	15	A0	40	8F	88	000FE		BISB2	#64, 21(R0)	
	7FFF	8F	0E	AE	B1	00103	11\$:	CMPW	NAMEBLOCK+14, #32767	0540
				05	1B	00109		BLEQU	12\$	
			0820	9F	BF	0010B		CHMU	#2080	0541
					04	0010F		RET		
	67	6E		10	28	00110	12\$:	MOV3	#16, NAMEBLOCK, (RECADDR)	0543
				57	DD	00114		PUSHL	RECADDR	0544
	0000G	CF		01	FB	00116		CALLS	#1, DIRPUT	
	0000G	CF	40	8F	88	0011B		BISB2	#64, CLEANUP_FLAGS+2	0545
			10	AC	DD	00121		PUSHL	RESULT	0550
			04	AE	9F	00124		PUSHAB	NAMEBLOCK	
	0000G	CF		02	FB	00127		CALLS	#2, MAKE_STRING	
	0C	BC		50	DO	0012C		MOVL	R0, @RESULT_LENGTH	
			04	AC	DD	00130		PUSHL	ABD	0551
			10	AC	DD	00133		PUSHL	RESULT	
			0C	BC	DD	00136		PUSHL	@RESULT_LENGTH	

ENTER
V04-000

M 6
16-Sep-1984 01:00:47
14-Sep-1984 12:29:30

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[F11A.SRC]ENTER.B32;1

Page 8
(2)

		03	DD	00139	PUSHL	#3	
		5E	DD	0013B	PUSHL	SP	
	0000G	CF	9F	00130	PUSHAB	RETURN DIR	
	0000G	68	06	FB	00141	CALLS	#6, SYSSCMKRN
	CF	00	FB	00144	CALLS	#0, PMS_END_SUB	
			04	00149	RET		

0556
0558

: Routine Size: 330 bytes, Routine Base: \$CODE\$ + 0000

```

: 246      0559 1
: 247      0560 1 END
: 248      0561 0 ELUDOM

```

PSECT SUMMARY

Name	Bytes	Attributes
\$CODE\$	330	NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)

Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
_\$255\$DUA28:[SYSLIB]LIB.L32;1	18619	25	0	1000	00:01.9

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS\$:ENTER/OBJ=OBJ\$:ENTER MSRC\$:ENTER/UPDATE=(ENH\$:ENTER)

```

: Size:          330 code + 0 data bytes
: Run Time:      00:09.8
: Elapsed Time: 00:29.7
: Lines/CPU Min: 3441
: Lexemes/CPU-Min: 14846
: Memory Used:  142 pages
: Compilation Complete

```

EXT
V04

0165 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

EXTFCB LIS											
DELJL LIS		DIRGET LIS					EXTIOX LIS			TODONE LIS	LOCKON LIS
					ENTER LIS						
								GETREQ LIS	GETTIM LIS		
			DISPAT LIS						INIFCP LIS		
		DIRFCB LIS					EXTHDR LIS				
			DIRSCN LIS								LOGDEL LIS
								FIND LIS	GETFIB LIS		
		DIRACC LIS							INIFCB LIS		
					EXTDIR LIS	EXTEND LIS					