


```

GGGGGGGG  EEEEEEEEEE  TTTTTTTTTT  FFFFFFFFFF  IIIIIII  BBBB88888
GGGGGGGG  EEEEEEEEEE  TTTTTTTTTT  FFFFFFFFFF  IIIIIII  BBBB88888
GG        EE          TT          FF          II          BB      BB
GG        EE          TT          FF          II          BB      BB
GG        EE          TT          FF          II          BB      BB
GG        EE          TT          FF          II          BB      BB
GG        EEEEEEEE  TT          FFFFFFFF  II          BBBB88888
GG        EEEEEEEE  TT          FFFFFFFF  II          BBBB88888
GG  GGGGGG  EE          TT          FF          II          BB      BB
GG  GGGGGG  EE          TT          FF          II          BB      BB
GG        GG  EE          TT          FF          II          BB      BB
GG        GG  EE          TT          FF          II          BB      BB
GGGGGG  EEEEEEEEEE  TT          FF          IIIIIII  BBBB88888
GGGGGG  EEEEEEEEEE  TT          FF          IIIIIII  BBBB88888

```

```

LL        IIIIIII  SSSSSSSS
LL        IIIIIII  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  IIIIIII  SSSSSSSS
LLLLLLLLLL  IIIIIII  SSSSSSSS

```

.....

```

1 0001 0 MODULE GETFIB (
2 0002 0 LANGUAGE (BLISS32),
3 0003 0 IDENT = 'V04-000'
4 0004 0 ) =
5 0005 1 BEGIN
6 0006 1
7 0007 1
8 0008 1 *****
9 0009 1 *
10 0010 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY *
11 0011 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. *
12 0012 1 * ALL RIGHTS RESERVED. *
13 0013 1 *
14 0014 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
15 0015 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE *
16 0016 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER *
17 0017 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
18 0018 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY *
19 0019 1 * TRANSFERRED. *
20 0020 1 *
21 0021 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE *
22 0022 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT *
23 0023 1 * CORPORATION. *
24 0024 1 *
25 0025 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS *
26 0026 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL. *
27 0027 1 *
28 0028 1 *
29 0029 1 *****
30 0030 1
31 0031 1 **
32 0032 1
33 0033 1 FACILITY: F11ACP Structure Level 1
34 0034 1
35 0035 1 ABSTRACT:
36 0036 1
37 0037 1 This routine obtains the address of the FIB for this operation.
38 0038 1
39 0039 1 ENVIRONMENT:
40 0040 1
41 0041 1 STARLET operating system, including privileged system services
42 0042 1 and internal exec routines.
43 0043 1
44 0044 1 --
45 0045 1
46 0046 1
47 0047 1 AUTHOR: Andrew C. Goldstein, CREATION DATE: 7-Jan-1977 01:02
48 0048 1
49 0049 1 MODIFIED BY:
50 0050 1
51 0051 1 V03-001 LMP0219 L. Mark Pilant, 24-Mar-1984 23:15
52 0052 1 Preset FIB$L_ACL_STATUS to $$$_NORMAL.
53 0053 1
54 0054 1 A0100 ACG0001 Andrew C. Goldstein, 10-Oct-1978 20:01
55 0055 1 Previous revision history moved to f11A.REV
56 0056 1
57 0057 1 **

```

GETFIB
V04-000

N 11
16-Sep-1984 01:05:44
14-Sep-1984 12:29:37

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[F11A.SRC]GETFIB.B32;1 Page 2 (1)

```
: 58      0058 1
: 59      0059 1
: 60      0060 1 LIBRARY 'SYSSLIBRARY:LIB.L32';
: 61      0061 1 REQUIRE 'SRCS:FCPDEF.B32';
```

GET
V04

000

```
63 0376 1 GLOBAL ROUTINE GET_FIB (ABD) =
64 0377 1
65 0378 1 :++
66 0379 1
67 0380 1 FUNCTIONAL DESCRIPTION:
68 0381 1
69 0382 1 This routine obtains the address of the FIB for this operation.
70 0383 1 It copies the FIB from the buffer packet into local storage
71 0384 1 and zero extends it to maximum length.
72 0385 1
73 0386 1 CALLING SEQUENCE:
74 0387 1 GET_FIB (ARG1)
75 0388 1
76 0389 1 INPUT PARAMETERS:
77 0390 1 ARG1: buffer descriptor list
78 0391 1
79 0392 1 IMPLICIT INPUTS:
80 0393 1 CURRENT_WINDOW: address of user's window or 0
81 0394 1 IO_PACKET: address of user's I/O packet
82 0395 1
83 0396 1 OUTPUT PARAMETERS:
84 0397 1 NONE
85 0398 1
86 0399 1 IMPLICIT OUTPUTS:
87 0400 1 NONE
88 0401 1
89 0402 1 ROUTINE VALUE:
90 0403 1 address of FIB
91 0404 1
92 0405 1 SIDE EFFECTS:
93 0406 1 file ID may be written into FIB
94 0407 1 channel window pointer write-back inhibited
95 0408 1 result string buffers zeroed
96 0409 1
97 0410 1 --
98 0411 1
99 0412 2 BEGIN
100 0413 2
101 0414 2 MAP
102 0415 2 ABD : REF BBLOCKVECTOR [,ABD$C_LENGTH];
103 0416 2 ! buffer descriptors
104 0417 2
105 0418 2 LOCAL
106 0419 2 FCB : REF BBLOCK, ! FCB of file
107 0420 2 FIBL; ! length of user FIB
108 0421 2
109 0422 2 EXTERNAL
110 0423 2 LOCAL_FIB : BBLOCK, ! internal copy of user FIB
111 0424 2 IO_PACKET : REF BBLOCK, ! I/O packet of this operation
112 0425 2 PRIMARY_FCB : REF BBLOCK, ! FCB of current file
113 0426 2 CURRENT_FIB : REF BBLOCK, ! pointer to current FIB in use
114 0427 2 CURRENT_WINDOW : REF BBLOCK; ! user's window
115 0428 2
116 0429 2
117 0430 2 ! Get the length of the user-supplied FIB. If there is a window,
118 0431 2 ! and there is no user FIB, use the file ID from
119 0432 2 ! the window's FCB. Also use the FCB's file ID if the file number
```

```
120 0433 2 ! in the user FIB is zero.
121 0434 2 !
122 0435 2 !
123 0436 2 FIBL = .ABD[ABD$C_FIB, ABD$W_COUNT];
124 0437 2 !
125 0438 2 CH$COPY (.FIBL,
126 0439 2 .ABD[ABD$C_FIB, ABD$W_TEXT] + ABD[ABD$C_FIB, ABD$W_TEXT] + 1,
127 0440 2 0
128 0441 2 FIB$C_LENGTH,
129 0442 2 LOCAL_FIB);
130 0443 2 CURRENT_FIB = LOCAL_FIB;
131 0444 2 !
132 0445 2 LOCAL_FIB[FIB$L_ACL_STATUS] = SSS_NORMAL; ! Preset to success
133 0446 2 !
134 0447 2 IF .CURRENT_WINDOW NEQ 0
135 0448 2 THEN
136 0449 2 BEGIN
137 0450 2 FCB = .CURRENT_WINDOW[WCB$L_FCB];
138 0451 2 IF .LOCAL_FIB[FIB$W_FID_NUM] EQL 0
139 0452 2 THEN CH$MOVE (FIB$S_FID, FCB[FCB$W_FID], LOCAL_FIB[FIB$W_FID]);
140 0453 2 !
141 0454 2 ! If the file ID in the FIB does not match that in the FCB, this operation
142 0455 2 ! is not on the open file; clear the FCB and window addresses (except in
143 0456 2 ! the case of a DEACCESS, in which we force the file ID to that of the open
144 0457 2 ! file and signal an error).
145 0458 2 !
146 0459 2 !
147 0460 2 IF .LOCAL_FIB[FIB$W_FID_NUM] NEQ .FCB[FCB$W_FID_NUM]
148 0461 2 OR .LOCAL_FIB[FIB$W_FID_RVN] NEQ .FCB[FCB$W_FID_RVN]
149 0462 2 THEN
150 0463 2 BEGIN
151 0464 2 IF .IO_PACKET[IRP$V_FCODE] EQL IOS_DEACCESS
152 0465 2 THEN
153 0466 2 BEGIN
154 0467 2 CH$MOVE (FIB$S_FID, FCB[FCB$W_FID], LOCAL_FIB[FIB$W_FID]);
155 0468 2 ERR_STATUS (SS$_BADPARAM);
156 0469 2 END
157 0470 2 ELSE
158 0471 2 BEGIN
159 0472 2 CURRENT_WINDOW = 0;
160 0473 2 PRIMARY_FCB = 0;
161 0474 2 END;
162 0475 2 END;
163 0476 2 END
164 0477 2 !
165 0478 2 ! If there is no file open, there must be a minimum FIB.
166 0479 2 !
167 0480 2 !
168 0481 2 ELSE
169 0482 2 BEGIN
170 0483 2 IF .FIBL LSS FIB$C_ACCDATA
171 0484 2 THEN ERR_EXIT (SS$_INSFARG);
172 0485 2 END;
173 0486 2 !
174 0487 2 ! If the directory ID is -1,-1, convert it to 4,4,0 to be compatible with
175 0488 2 ! the old RSX MFD kluge.
176 0489 2 !
```


