

```

FFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFF
FFF
FFF
FFF
FFF
FFF
FFF
FFFFFFFFFFF
FFFFFFFFFFF
FFFFFFFFFFF
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
111111111
111111111
111111111

```

```

111
111
111
111111
111111
111111
111
111
111
111
111
111
111
111
111
111
111
111
111111111
111111111
111111111

```

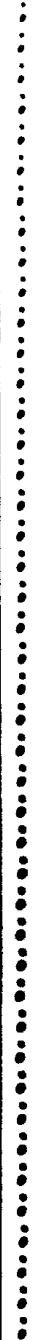
```

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

```

```
CCCCCCCC RRRRRRRR EEEEEEEEEE FFFFFFFFFF CCCCCCCC BBBB88888
CCCCCCCC RRRRRRRR EEEEEEEEEE FFFFFFFFFF CCCCCCCC BBBB88888
CC        RR      RR EE          FF          CC        BB      BB
CC        RR      RR EE          FF          CC        BB      BB
CC        RR      RR EE          FF          CC        BB      BB
CC        RRRRRRRR EEEEEEEE FFFFFFFF CCCCCCCC BBBB88888
CC        RRRRRRRR EEEEEEEE FFFFFFFF CCCCCCCC BBBB88888
CC        RR  RR   EE          FF          CC        BB      BB
CC        RR  RR   EE          FF          CC        BB      BB
CC        RR      RR EE          FF          CC        BB      BB
CC        RR      RR EE          FF          CC        BB      BB
CCCCCCCC RR      RR EEEEEEEEEE FFFFFFFF CCCCCCCC BBBB88888
CCCCCCCC RR      RR EEEEEEEEEE FFFFFFFF CCCCCCCC BBBB88888
```

```
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
LL        II      SS
LLLLLLLLLL IIIIII SSSSSSSS
LLLLLLLLLL IIIIII SSSSSSSS
```



```

1 0001 0 MODULE CREFCB (
2 0002 0
3 0003 0     LANGUAGE (BLISS32),
4 0004 0     IDENT = 'V04-000'
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     These routines create and initialize a file control block
38 0038 1     from the given file header.
39 0039 1
40 0040 1 ENVIRONMENT:
41 0041 1
42 0042 1     STARLET operating system, including privileged system services
43 0043 1     and internal exec routines. These routines must be called in
44 0044 1     kernel mode.
45 0045 1
46 0046 1
47 0047 1 --
48 0048 1
49 0049 1
50 0050 1 AUTHOR: Andrew C. Goldstein, CREATION DATE: 14-Dec-1976 16:48
51 0051 1
52 0052 1 MODIFIED BY:
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 **

```

CREFCB  
V04-000

C 14  
16-Sep-1984 00:54:07  
14-Sep-1984 12:29:25

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

CRE  
V04

```

: 58      0058 1
: 59      0059 1
: 60      0060 1 LIBRARY 'SYSS$LIBRARY:LIB.L32';
: 61      0061 1 REQUIRE 'SRC$:FCPDEF.B32';
: 62      0376 1
: 63      0377 1
: 64      0378 1 FORWARD ROUTINE
: 65      0379 1 CREATE_FCB,
: 66      0380 1 UPDATE_FCB      : NOVALUE;
```

: R

```

68 0381 1 GLOBAL ROUTINE CREATE_FCB (HEADER) =
69 0382 1
70 0383 1 !++
71 0384 1
72 0385 1 FUNCTIONAL DESCRIPTION:
73 0386 1
74 0387 1     This routine creates an FCB and initializes it according to
75 0388 1     the given file header.
76 0389 1
77 0390 1 CALLING SEQUENCE:
78 0391 1     CREATE_FCB (ARG1)
79 0392 1
80 0393 1 INPUT PARAMETERS:
81 0394 1     ARG1: address of file header
82 0395 1
83 0396 1 IMPLICIT INPUTS:
84 0397 1     NONE
85 0398 1
86 0399 1 OUTPUT PARAMETERS:
87 0400 1     NONE
88 0401 1
89 0402 1 IMPLICIT OUTPUTS:
90 0403 1     NONE
91 0404 1
92 0405 1 ROUTINE VALUE:
93 0406 1     ADDRESS OF FCB
94 0407 1
95 0408 1 SIDE EFFECTS:
96 0409 1     FCB created and initialized
97 0410 1
98 0411 1 !--
99 0412 1
100 0413 2 BEGIN
101 0414 2
102 0415 2 MAP
103 0416 2     HEADER          : REF BBLOCK;    ! file header argument
104 0417 2
105 0418 2 LOCAL
106 0419 2     FCB              : REF BBLOCK;    ! address of FCB created
107 0420 2
108 0421 2 EXTERNAL ROUTINE
109 0422 2     ALLOCATE,          ! allocate dynamic memory
110 0423 2     INIT_FCB;         ! initialize contents of FCB
111 0424 2
112 0425 2 ! Allocate an FCB sized and typed block. Then use the common routine
113 0426 2 ! to init it.
114 0427 2
115 0428 2
116 0429 2 FCB = ALLOCATE (FCB$C_LENGTH, FCB_TYPE);
117 0430 2 FCB[FCB$S_WLFL] = FCB[FCB$S_WLFL]; ! init null window list
118 0431 2 FCB[FCB$S_WLBL] = FCB[FCB$S_WLFL];
119 0432 2 FCB[FCB$S_STVBN] = 1; ! init start VBN to 1
120 0433 2 INIT_FCB (.FCB, .HEADER);
121 0434 2 RETURN .FCB;
122 0435 2
123 0436 1 END; ! end of routine CREATE_FCB

```

			0004 00000	.TITLE	CREFCB	
			7E D4 00002	.IDENT	\V04-000\	
			8F 9A 00004	.EXTRN	ALLOCATE, INIT_FCB	
			02 FB 00008	.PSECT	\$CODE\$,NOWRT,2	
			50 D0 0000D	.ENTRY	CREATE_FCB, Save R2	: 0381
0000G	7E	B4	01 D0 0001A	CLR	-(SP)	: 0429
	CF		AC DD 0001E	MOVZBL	#180, -(SP)	
	52		52 DD 00021	CALLS	#2, ALLOCATE	
10	A2	10	02 FB 00023	MOVL	RO, FCB	
	52		04 0002B	MOVAB	16(FCB), 16(FCB)	: 0430
14	A2	10		MOVAB	16(FCB), 20(FCB)	: 0431
	A2			MOVL	#1, 44(FCB)	: 0432
2C	A2	04		PUSHL	HEADER	: 0433
				PUSHL	FCB	
0000G	CF			CALLS	#2, INIT_FCB	: 0434
	50			MOVL	FCB, RO	: 0436
				RET		

: Routine Size: 44 bytes, Routine Base: \$CODE\$ + 0000

```

125 0437 1 GLOBAL ROUTINE UPDATE_FCB (HEADER) : NOVALUE =
126 0438 1
127 0439 1 !++
128 0440 1
129 0441 1 FUNCTIONAL DESCRIPTION:
130 0442 1
131 0443 1 This routine updates the file attributes of the file's primary FCB,
132 0444 1 if any, with the file attributes of the given header. The file size
133 0445 1 is preserved.
134 0446 1
135 0447 1
136 0448 1 CALLING SEQUENCE:
137 0449 1 UPDATE_FCB (ARG1)
138 0450 1
139 0451 1 INPUT PARAMETERS:
140 0452 1 ARG1: address of file header
141 0453 1
142 0454 1 IMPLICIT INPUTS:
143 0455 1 NONE
144 0456 1
145 0457 1 OUTPUT PARAMETERS:
146 0458 1 NONE
147 0459 1
148 0460 1 IMPLICIT OUTPUTS:
149 0461 1 PRIMARY_FCB: address of file FCB or 0
150 0462 1
151 0463 1 ROUTINE VALUE:
152 0464 1 NONE
153 0465 1
154 0466 1 SIDE EFFECTS:
155 0467 1 FCB is updated if it exists
156 0468 1
157 0469 1 --
158 0470 1
159 0471 2 BEGIN
160 0472 2
161 0473 2 MAP
162 0474 2 HEADER : REF BBLOCK; ! file header arg
163 0475 2
164 0476 2 LOCAL
165 0477 2 FCB : REF BBLOCK, ! local pointer to FCB
166 0478 2 MAP_AREA : REF BBLOCK, ! pointer to header map area
167 0479 2 MAP_POINTER : REF BBLOCK; ! pointer to scan map
168 0480 2
169 0481 2 EXTERNAL
170 0482 2 PRIMARY_FCB : REF BBLOCK, ! FCB of file in process
171 0483 2 HEADER_LBN; ! LBN of file header
172 0484 2
173 0485 2
174 0486 2 FCB = .PRIMARY_FCB;
175 0487 2 IF .FCB EQL 0 THEN RETURN;
176 0488 2
177 0489 2
178 0490 2 ! Get the known constants and the simple stuff from the file header
179 0491 2 ! (i e., header LBN, file ID, starting VBN, file owner and file protection).
180 0492 2
181 0493 2

```

```

182 0494 2 FCB[FCBSL_HDLBN] = .HEADER_LBN;
183 0495 2 FCB[FCBSW_FID_NUM] = .HEADER[FH1$W_FID_NUM];
184 0496 2 FCB[FCBSW_FID_SEQ] = .HEADER[FH1$W_FID_SEQ];
185 0497 2 FCB[FCBSW_UICMEMBER] = .HEADER[FH1$B_UICMEMBER];
186 0498 2 FCB[FCBSW_UICGROUP] = .HEADER[FH1$B_UICGROUP];
187 0499 2 FCB[FCBSW_FILEPROT] = .HEADER[FH1$W_FILEPROT];
188 0500 2 IF .HEADER[FH1$V_SPOOL] THEN FCB[FCBSV_SPOOL] = 1;
189 0501 2 FCB[FCBSL_EFBLK] = ROT (.BBLOCK[HEADER[FH1$W_RECATTR], FAT$E_EFBLK], 16);
190 0502 2 IF .FCB[FCBSL_EFBLK] NEQ 0
191 0503 2 AND .BBLOCK[HEADER[FH1$W_RECATTR], FAT$W_FFBYTE] EQL 0
192 0504 2 THEN FCB[FCBSL_EFBLK] = .FCB[FCBSL_EFBLK] - 1;
193 0505
194 0506 2 ! Now scan the map area. Get the starting LBN if the f : is contiguous.
195 0507 2 !
196 0508
197 0509 2 MAP_AREA = .HEADER + .HEADER[FH1$B_MPOFFSET]*2;
198 0510 2 MAP_POINTER = .MAP_AREA + FM1$C_POINTERS;
199 0511 2 FCB[FCBSW_SEGN] = .MAP_AREA[FM1$B_EX_SEGNUM];
200 0512
201 0513 2 FCB[FCBSL_STLBN] = 0; ! assume non-contiguous file
202 0514 2 IF .HEADER[FH1$V_CONTIG]
203 0515 2 THEN
204 0516 2 BEGIN
205 0517 2 FCB[FCBSL_STLBN] = .MAP_POINTER[FM1$W_LOWLBN]; ! get low order LBN
206 0518 2 (FCB[FCBSL_STLBN]) < 16,85 = .MAP_POINTER[FM1$B_HIGHLBN]; ! and high order
207 0519 2 END;
208 0520
209 0521
210 0522 2 IF .FCB[FCBSL_EFBLK] GTR .FCB[FCBSL_FILESIZE]
211 0523 2 THEN FCB[FCBSL_EFBLK] = .FCB[FCBSL_FILESIZE];
212 0524
213 0525 1 END; ! end of routine UPDATE_FCB

```

					.EXTRN	PRIMARY_FCB, HEADER_LBN	
					.ENTRY	UPDATE_FCB, Save R2,R3,R4	: 0437
		52	0000G	CF D0 00002	MOVL	PRIMARY_FCB, FCB	: 0486
				64 13 00007	BEQL	4\$	: 0487
	34	A2	0000G	CF D0 00009	MOVL	HEADER_LBN, 52(FCB)	: 0494
		S3		04 AC D0 0000F	MOVL	HEADER, R3	: 0495
	24	A2		02 A3 D0 00013	MOVL	2(R3), 36(FCB)	
	58	A2		08 A3 9B 00018	MOVZBW	8(R3), 88(FCB)	: 0497
	5A	A2		09 A3 9B 0001D	MOVZBW	9(R3), 90(FCB)	: 0498
	70	A2		0A A3 80 00022	MOVW	10(R3), 112(FCB)	: 0499
	04	A3		04 E1 00027	BBC	#4, 13(R3), 1\$	: 0500
		A2		10 88 0002C	BISB2	#16, 34(FCB)	
		S4	3C	A2 9E 00030	MOVAB	60(FCB), R4	: 0501
	64	A3		10 9C 00034	ROTL	#16, 22(R3), (R4)	
				07 13 00039	BEQL	2\$	: 0502
			1A	A3 85 0003B	TSTW	26(R3)	: 0503
				02 12 0003E	BNEQ	2\$	
				64 D7 00040	DECL	(R4)	: 0504
		50	01	A3 9A 00042	MOVZBL	1(R3), R0	: 0509
		51		6340 3E 00046	MOVAV	(R3)[R0], MAP_AREA	
		50	0A	A1 9E 0004A	MOVAB	10(R1), MAP_POINTER	: 0510



2A	A2		61	9B	0004E	MOVZBW	(MAP AREA), 42(FCB)	:	0511
		30	A2	D4	00052	CLRL	48(FCB)	:	0513
		0C	A3	95	00055	TSTB	12(R3)	:	0514
			09	18	00058	BGEQ	3\$	:	
30	A2	02	A0	3C	0005A	MOVZWL	2(MAP POINTER), 48(FCB)	:	0517
32	A2		60	90	0005F	MOVB	(MAP POINTER), 50(FCB)	:	0518
38	A2		64	D1	00063	3\$: Cmpl	(R4), 56(FCB)	:	0522
			04	15	00067	BLEQ	4\$	:	
	64	38	A2	D0	00069	MOVL	56(FCB), (R4)	:	0523
			04	0006D	4\$: RET			:	0525

: Routine Size: 110 bytes, Routine Base: \$CODE\$ + 002C

```

: 214      0526 1
: 215      0527 1 END
: 216      0528 0 ELUDOM

```

PSECT SUMMARY

Name	Bytes	Attributes
\$CODE\$	154	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	31	0	1000	00:01.9

COMMAND QUALIFIERS

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

```

: Size:      154 code + 0 data bytes
: Run Time:  00:08.3
: Elapsed Time: 00:25.9
: Lines/CPU Min: 3830
: Lexemes/CPU-Min: 17470
: Memory Used: 102 pages
: Compilation Complete

```

FCPOEF B32	ACPCNTR LIS	CHKSUM LIS	CHKPRO LIS	DEACCS LIS
BADSEN LIS	CLENUP LIS	CPYNAM LIS	CHKHDR LIS	COMMON LIS
CREHDR LIS	CREWIN LIS	ACCESS LIS	ALLOB LIS	CHKDMD LIS
CREATE LIS	CREFCB LIS	DELETE LIS		