

```
FFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFF
FFF
FFF
FFF
FFF
FFF
FFF
FFFFFFFFFFFF
FFFFFFFFFFFF
FFFFFFFFFFFF
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
111111111
111111111
111111111
```

```
111
111
111
111111
111111
111111
111
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
AAA      AAA
AAA      AAA
AAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAA
AAA      AAA
AAA      AAA
AAA      AAA
AAA      AAA
AAA      AAA
AAA      AAA
AAA      AAA
```

```

EEEEEEEEEE XX      XX  TTTTTTTTTT FFFFFFFFFF CCCCCCCC BBBB88888
EEEEEEEEEE XX      XX  TTTTTTTTTT FFFFFFFFFF CCCCCCCC BBBB88888
EE          XX      XX  TT          FF          CC          BB      BB
EE          XX      XX  TT          FF          CC          BB      BB
EE          XX      XX  TT          FF          CC          BB      BB
EE          XX      XX  TT          FF          CC          BB      BB
EEEEEEEEEE      XX  TT          FFFFFFFF CC          BBBB88888
EEEEEEEEEE      XX  TT          FFFFFFFF CC          BBBB88888
EE          XX      XX  TT          FF          CC          BB      BB
EE          XX      XX  TT          FF          CC          BB      BB
EE          XX      XX  TT          FF          CC          BB      BB
EEEEEEEEEE XX      XX  TT          FF          CC          BB      BB
EEEEEEEEEE XX      XX  TT          FF          CC          BB      BB

```

```

LL          !IIIII  SSSSSSSS
LL          !IIIII  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
LLLLLLLLLLL !IIIII  SSSSSSSS
LLLLLLLLLLL !IIIII  SSSSSSSS

```

```

1 0001 0 MODULE EXTFCB (
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 updates an FCB for one extension header of the file.
38 0038 1     If the file is accessed, it creates an extension FCB. Regardless,
39 0039 1     it updates the file size in the primary FCB.
40 0040 1
41 0041 1 ENVIRONMENT:
42 0042 1
43 0043 1     STARLET operating system, including privileged system services
44 0044 1     and internal exec routines. This routine must be called in
45 0045 1     kernel mode.
46 0046 1
47 0047 1 --
48 0048 1
49 0049 1
50 0050 1 AUTHOR: Andrew C. Goldstein, CREATION DATE: 25-Jul-1977 10:55
51 0051 1
52 0052 1 MODIFIED BY:
53 0053 1
54 0054 1     V03-001 LMP0307          L. Mark Pilant,          29-Aug-1984 14:02
55 0055 1     Remove the call to the window turner as it is done elsewhere.
56 0056 1
57 0057 1     V02-002 LMP0005          L. Mark Pilant,          5-Jan-1982 15:55

```

: Rc

:  
: :  
: :

:  
: :  
: :  
: :

:  
: :  
: :  
: :

:  
: :

: S  
: R  
: E  
: L  
: L  
: M  
: C

```
.. 58      0058 1  |      Add support for Cathedral windows.
.. 59      0059 1  |
.. 60      0060 1  |      A0101  ACG26369      Andrew C. Goldstein,      28-Dec-1979  15:44
.. 61      0061 1  |      Fix multi-header interlock bug
.. 62      0062 1  |
.. 63      0063 1  |      A0100  ACG00001      Andrew C. Goldstein,      10-Oct-1978  20:02
.. 64      0064 1  |      Previous revision history moved to F11A.REV
.. 65      0065 1  |      !**
.. 66      0066 1  |
.. 67      0067 1  |
.. 68      0068 1  |      LIBRARY 'SYSS$LIBRARY:LIB.L32';
.. 69      0069 1  |      REQUIRE 'SRCS:FCPDEF.B32';
```

```
71 0384 1 GLOBAL ROUTINE MAKE_EXTFCB (HEADER, FCB, MODE) =
72 0385 1
73 0386 1 ++
74 0387 1
75 0388 1 FUNCTIONAL DESCRIPTION:
76 0389 1
77 0390 1 This routine updates an FCB for one extension header of the file.
78 0391 1 If the file is accessed, it creates an extension FCB. Regardless,
79 0392 1 it updates the file size in the primary FCB.
80 0393 1
81 0394 1
82 0395 1 CALLING SEQUENCE:
83 0396 1 MAKE_EXTFCB (ARG1, ARG2, ARG3)
84 0397 1
85 0398 1 INPUT PARAMETERS:
86 0399 1 ARG1: address of extension file header
87 0400 1 ARG2: address of last file FCB or 0
88 0401 1 ARG3: 0 if just building primary FCB for read attributes
89 0402 1 1 if file is being accessed (to create new FCB)
90 0403 1 2 if file is being extended (creating new header)
91 0404 1
92 0405 1 IMPLICIT INPUTS:
93 0406 1 PRIMARY_FCB: address of primary FCB of file
94 0407 1 CURRENT_WINDOW: address of file window or 0
95 0408 1 UNREC_COUNT: count of allocated but unrecorded blocks
96 0409 1
97 0410 1 OUTPUT PARAMETERS:
98 0411 1 NONE
99 0412 1
100 0413 1 IMPLICIT OUTPUTS:
101 0414 1 NONE
102 0415 1
103 0416 1 ROUTINE VALUE:
104 0417 1 address of new FCB created or 0
105 0418 1
106 0419 1 SIDE EFFECTS:
107 0420 1 new FCB created, primary FCB modified, window modified
108 0421 1
109 0422 1 --
110 0423 1
111 0424 2 BEGIN
112 0425 2
113 0426 2 MAP
114 0427 2 HEADER : REF BBLOCK, ! file header arg
115 0428 2 FCB : REF BBLOCK; ! FCB arg
116 0429 2
117 0430 2 LOCAL
118 0431 2 DUMMY_FCB : BBLOCK [FCB$C_LENGTH], ! dummy FCB to compute file size
119 0432 2 NEW_FCB : REF BBLOCK; ! address of new FCB created
120 0433 2
121 0434 2 EXTERNAL
122 0435 2 PRIMARY_FCB : REF BBLOCK, ! address of primary FCB of file
123 0436 2 CURRENT_WINDOW : REF BBLOCK, ! address of file window if any
124 0437 2 CURRENT_FIB : REF BBLOCK, ! FIB of operation
125 0438 2 UNREC_COUNT; ! count of allocated but unrecorded blocks
126 0439 2
127 0440 2 EXTERNAL ROUTINE
```

```

128 0441 2 CREATE_FCB, ! create a new FCB
129 0442 2 INIT_FCB; ! initialize existing FCB
130 0443
131 0444
132 0445 ! If the file is not accessed, this is for a read attributes which just needs
133 0446 ! a primary FCB. All we do is compute the file size of the header and add
134 0447 ! it into the primary FCB.
135 0448
136 0449
137 0450 IF .MODE EQL 0
138 0451 THEN
139 0452 BEGIN
140 0453 INIT_FCB (DUMMY_FCB, .HEADER); ! get file size in dummy FCB
141 0454 PRIMARY_FCB[FCB$$_FILESIZE] = .PRIMARY_FCB[FCB$$_FILESIZE] + .DUMMY_FCB[FCB$$_FILESIZE];
142 0455 RETURN 0;
143 0456 END
144 0457
145 0458 ! If the file is accessed, we create a new FCB if there isn't one already
146 0459 ! and link it in. On an access we attempt a window turn to VBN 1 in case
147 0460 ! this is a large contiguous file. On an extend we must compute the new FCB
148 0461 ! starting VBN from the extend data lying around, since the primary FCB isn't
149 0462 ! updated until the very end.
150 0463
151 0464
152 0465 ELSE
153 0466 BEGIN
154 0467 NEW_FCB = CREATE_FCB (.HEADER);
155 0468 NEW_FCB[FCB$$_ACNT] = 1;
156 0469 FCB[FCB$$_EXFCB] = .NEW_FCB;
157 0470 INSQUE (.NEW_FCB, FCB[FCB$$_FCBFL]);
158 0471 IF .MODE
159 0472 THEN
160 0473 BEGIN
161 0474 NEW_FCB[FCB$$_STVBN] = .NEW_FCB[FCB$$_STVBN] + .PRIMARY_FCB[FCB$$_FILESIZE];
162 0475 PRIMARY_FCB[FCB$$_FILESIZE] = .PRIMARY_FCB[FCB$$_FILESIZE] + .NEW_FCB[FCB$$_FILESIZE];
163 0476 END
164 0477 ELSE
165 0478 NEW_FCB[FCB$$_STVBN] = .NEW_FCB[FCB$$_STVBN] + .PRIMARY_FCB[FCB$$_FILESIZE]
166 0479 + .CURRENT_FIB[FCB$$_EXSZ] - .UNREC_COUNT;
167 0480
168 0481 RETURN .NEW_FCB;
169 0482 END;
170 0483
171 0484 1 END; ! end of routine MAKE_EXTFCB

```

```

.TITLE EXTFCB
.IDENT \V04-000\

.EXTRN PRIMARY_FCB, CURRENT_WINDOW
.EXTRN CURRENT_FIB, UNREC_COUNT
.EXTRN CREATE_FCB, INIT_FCB

```

```
.PSECT $CODE$,NOWRT,2
```

```
.ENTRY MAKE_EXTFCB, Save R2
MOVAB -180(TSP), SP
```

```
SE FF4C CE 9E 0000
0004 0000
```

```
: 0384
:
```

		0C	AC	05	00007	TSTL	MODE		: 0450
			18	12	0000A	BNEQ	1\$		: 0453
		04	AC	DD	0000C	PUSHL	HEADER		: 0454
		04	AE	9F	0000F	PUSHAB	DUMMY_FCB		: 0466
0000G	CF		02	FB	00012	CALLS	#2, INIT_FCB		: 0467
	50	0000G	CF	DO	00017	MOVL	PRIMARY_FCB, R0		: 0468
38	A0	38	AE	CO	0001C	ADDL2	DUMMY_FCB+56, 56(R0)		: 0469
			50	D4	00021	CLRL	R0		: 0470
				04	00023	RET			: 0471
		04	AC	DD	00024	PUSHL	HEADER		: 0474
0000G	CF		01	FB	00027	CALLS	#1, CREATE_FCB		: 0475
1A	A0		01	B0	0002C	MOVW	#1, 26(NEW_FCB)		: 0477
	51	08	AC	DO	00030	MOVL	FCB, R1		: 0478
0C	A1		50	DO	00034	MOVL	NEW_FCB, 12(R1)		: 0479
	61		60	OE	00038	INSQUE	(NEW_FCB), (R1)		: 0484
	51	0000G	CF	DO	0003B	MOVL	PRIMARY_FCB, R1		: 0485
	0B	0C	AC	E9	00040	BLBC	MODE, 2\$		: 0486
2C	A0	38	A1	CO	00044	ADDL2	56(R1), 44(NEW_FCB)		: 0487
38	A1	38	A0	CO	00049	ADDL2	56(NEW_FCB), 58(R1)		: 0488
				04	0004E	RET			: 0489
52	2C	38	A1	C1	0004F	ADDL3	56(R1), 44(NEW_FCB), R2		: 0490
		0000G	CF	DO	00055	MOVL	CURRENT_FIB, R1		: 0491
			18	A1	CO	0005A	ADDL2	24(R1), R2	: 0492
2C	A0	0000G	CF	C3	0005E	SUBL3	UNREC_COUNT, R2, 44(NEW_FCB)		: 0493
				04	00065	RET			: 0494

: Routine Size: 102 bytes, Routine Base: \$CODE\$ + 0000

```

: 172      0485 1
: 173      0486 1 END
: 174      0487 0 ELUDOM

```

PSECT SUMMARY

Name	Bytes	Attributes
\$CODE\$	102	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	8	0	1000	00:02.0

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LISS:EXTFCB/OBJ=OBJS:EXTFCB MSRCS:EXTFCB/UPDATE=(ENHS:EXTFCB)

: Size: 102 code + 0 data bytes  
: Run Time: 00:06.6  
: Elapsed Time: 00:21.1  
: Lines/CPU Min: 4427  
: Lexemes/CPU-Min: 14145  
: Memory Used: 82 pages  
: Compilation Complete

.....



Grid of technical diagrams, including: EXTFCB LIS, DELF JL LIS, DIRGET LIS, EXTIOX LIS, LOCKON LIS, IODONE LIS, ENTER LIS, GETREQ LIS, GETTIM LIS, DISPAT LIS, DIRFCB LIS, EXTHDR LIS, DIRSON LIS, LOGDEL LIS, LOCKDB LIS, DIRACC LIS, INIFCB LIS, EXTDIR LIS, and EXTEND LIS.