

FFFFFFFFFFFFFFFF	111	111	AAAAAAAA	
FFFFFFFFFFFFFFFF	111	111	AAAAAAAA	
FFFFFFFFFFFFFFFF	111	111	AAAAAAAA	
FFF	111111	111111	AAA	AAA
FFF	111111	111111	AAA	AAA
FFF	111111	111111	AAA	AAA
FFF	111	111	AAA	AAA
FFF	111	111	AAA	AAA
FFF	111	111	AAA	AAA
FFFFFFFFFFFFFF	111	111	AAA	AAA
FFFFFFFFFFFFFF	111	111	AAA	AAA
FFFFFFFFFFFFFF	111	111	AAA	AAA
FFF	111	111	AAAAAAAAAAAAAAAA	
FFF	111	111	AAAAAAAAAAAAAAAA	
FFF	111	111	AAAAAAAAAAAAAAAA	
FFF	111	111	AAA	AAA
FFF	111	111	AAA	AAA
FFF	111	111	AAA	AAA
FFF	111	111	AAA	AAA
FFF	1111111111	1111111111	AAA	AAA
FFF	1111111111	1111111111	AAA	AAA
FFF	1111111111	1111111111	AAA	AAA

```
EEEEEEEEEE XX XX TTTTTTTTTT FFFFFFFFFF CCCCCCCC BBBBBBBBBB
EEEEEEEEEE XX XX TTTTTTTTTT FFFFFFFFFF CCCCCCCC BBBBBBBBBB
EE XX XX TTT TT FF CC BB
EE XX XX TTT TT FF CC BB
EE XX XX TTT TT FF CC BB
EEEEEEEE XX XX TTT TT FFFFFFFF CCCCCCCC BBBBBBBBBB
EEEEEEEE XX XX TTT TT FFFFFFFF CCCCCCCC BBBBBBBBBB
EE XX XX TTT TT FF CC BB
EE XX XX TTT TT FF CC BB
EE XX XX TTT TT FF CC BB
EEEEEEEEEE XX XX TTT TT FFFFFFFF CCCCCCCC BBBBBBBBBB
EEEEEEEEEE XX XX TTT TT FFFFFFFF CCCCCCCC BBBBBBBBBB
.....
.....
.....
.....
```

```
LL !IIIII SSSSSSSS
LL !IIIII SSSSSSSS
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 !IIIII SSSSSSSS
LLLLLLLLLL !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
```

EXTFCB
V04-000

0 9
16-Sep-1984 01:03:11
14-Sep-1984 12:29:33

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

```

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

```

**F1


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

```
SE      FF4C      CE 9E 00002
```

```
: 0384
:
```


		0C	AC	D5	00007	TSTL	MODE	:	0450
			18	12	0000A	BNEQ	1\$	:	
		04	AC	DD	0000C	PUSHL	HEADER	:	0453
		04	AE	9F	0000F	PUSHAB	DUMMY_FCB	:	
	0000G	CF	02	FB	00012	CALLS	#2, INIT_FCB	:	
	38	50	CF	DO	00017	MOVL	PRIMARY_FCB, R0	:	0454
		A0	38	AE	CO	ADDL2	DUMMY_FCB+56, 56(R0)	:	
			50	D4	00021	CLRL	R0	:	0466
				04	00023	RET		:	
		04	AC	DD	00024	PUSHL	HEADER	:	0467
	0000G	CF	01	FB	00027	CALLS	#1, CREATE_FCB	:	
	1A	A0	01	B0	0002C	MOVW	#1, 26(NEW_FCB)	:	0468
		51	08	AC	DO	MOVL	FCB, R1	:	0469
	0C	A1	50	DO	00034	MOVL	NEW_FCB, 12(R1)	:	
		61	60	OE	00038	INSQUE	(NEW_FCB), (R1)	:	0470
		51	0000G	CF	DO	MOVL	PRIMARY_FCB, R1	:	0474
		0B	0C	AC	E9	BLBC	MODE, 2\$	:	0471
	2C	A0	38	A1	CO	ADDL2	56(R1), 44(NEW_FCB)	:	0474
	38	A1	38	A0	CO	ADDL2	56(NEW_FCB), 56(R1)	:	0475
				04	0004E	RET		:	0471
	52	2C	A0	38	A1	ADDL3	56(R1), 44(NEW_FCB), R2	:	0478
			51	0000G	CF	MOVL	CURRENT_FIB, RT	:	0479
			52	18	A1	ADDL2	24(R1), R2	:	
	2C	A0	52	0000G	CF	SUBL3	UNREC_COUNT, R2, 44(NEW_FCB)	:	
				04	00065	RET		:	0484

; 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,NOPI,ALIGN(2)

Library Statistics

File	Symbols		Pages Mapped	Processing Time
	Total	Loaded Percent		
_\$255\$DUA28:[SYSLIB]LIB.L32;1	18619	8 0	1000	00:02.0

EXTFCB
V04-000

H 9
16-Sep-1984 01:03:11
14-Sep-1984 12:29:33

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

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LISS:EXTFCB/OBJ=OBJ\$:EXTFCB MSRC\$: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

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

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY