

FFFFFFFFFFFFFFFF	111	111	XXX	XXX
FFFFFFFFFFFFFFFF	111	111	XXX	XXX
FFFFFFFFFFFFFFFF	111	111	XXX	XXX
FFF	111111	111111	XXX	XXX
FFF	111111	111111	XXX	XXX
FFF	111111	111111	XXX	XXX
FFF	111	111	XXX	XXX
FFF	111	111	XXX	XXX
FFF	111	111	XXX	XXX
FFFFFFFF.FFF	111	111	XXX	XXX
FFFFFFFFFFFF	111	111	XXX	XXX
FFFFFFFFFFFF	111	111	XXX	XXX
FFF	111	111	XXX	XXX
FFF	111	111	XXX	XXX
FFF	111	111	XXX	XXX
FFF	111	111	XXX	XXX
FFF	111	111	XXX	XXX
FFF	111	111	XXX	XXX
FFF	111	111	XXX	XXX
FFF	111111111	111111111	XXX	XXX
FFF	111111111	111111111	XXX	XXX
FFF	111111111	111111111	XXX	XXX

_S25
Symt

IOCI
IO_C
IO-C
IO-D
IO-E
IO-F
IO-S
KICL
KILL
KILL
LB_E
LB_C
LB-F
LB_P
LB-L
LOCAL
LOCK
LOCK
LOCK
LOCK
LOC-
LOC-
L-CC
L-CC
L-CC
L-DA
L-DA
MAIN
MAKE
MAKE
MAKE
MAKE
MAKE
MAKE
MAKE
MAKE
MAKE
MAKE
MAKE
MAKE
MAKE
MAKE
MAKE
MAKE
MAKE
MAP-
MAP-
MAP
MARF
MARF
MARF
MARF
MARF


```

1 0001 0 MODULE EXTFCB (
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 2
34 0034 1
35 0035 1 ABSTRACT:
36 0036 1
37 0037 1 This module contains a routine which will build the
38 0038 1 extension fcb chain for the given fcb, if necessary.
39 0039 1
40 0040 1 ENVIRONMENT:
41 0041 1
42 0042 1 VAX/VMS operating system, including privileged system services
43 0043 1 and internal exec routines. This routine must be called in
44 0044 1 kernel mode.
45 0045 1
46 0046 1 --
47 0047 1
48 0048 1
49 0049 1 AUTHOR: Andrew C. Goldstein, CREATION DATE: 25-Jul-1977 10:55
50 0050 1
51 0051 1 MODIFIED BY:
52 0052 1
53 0053 1 V03-007 CDS0005 Christian D. Saether 29-Aug-1984
54 0054 1 Add optional second argument to BUILD_EXT_FCBS
55 0055 1 to specify primary fcb other than PRIMARY_FCB.
56 0056 1
57 0057 1 V03-006 CDS0004 Christian D. Saether 21-Aug-1984

```

```
58 0058 1 | Update EFBLK after turning back to primary when
59 0059 1 | building extension fcb chain.
60 0060 1 |
61 0061 1 | V03-005 CDS0003 Christian D. Saether 14-Aug-1984
62 0062 1 | Replace MAKE_EXTFCB routine with BUILD_EXT_FCBS.
63 0063 1 |
64 0064 1 | V03-004 CDS0002 Christian D. Saether 19-Apr-1984
65 0065 1 | Use REFCNT instead of ACNT.
66 0066 1 | Set up FCBSL_LOCKBASIS to be that of primary fcb.
67 0067 1 |
68 0068 1 | V03-003 CDS0001 Christian D. Saether 30-Dec-1983
69 0069 1 | Use L_NORM linkage and BIND_COMMON macro.
70 0070 1 |
71 0071 1 | V03-002 LMP0059 L. Mark Pilant, 21-Dec-1982 10:51
72 0072 1 | Always create an FCB for a file header accessed. This
73 0073 1 | eliminates a lot of special casing for FCB handling.
74 0074 1 |
75 0075 1 | V03-001 ACG0272 Andrew C. Goldstein, 23-Mar-1982 10:17
76 0076 1 | Clean up use of dummy FCB
77 0077 1 |
78 0078 1 | B0102 ACG26369 Andrew C. Goldstein, 28-Dec-1979 15:44
79 0079 1 | Fix multi-header interlock bug
80 0080 1 |
81 0081 1 | B0101 ACG0003 Andrew C. Goldstein, 19-Dec-1978 17:40
82 0082 1 | Add multi-volume support
83 0083 1 |
84 0084 1 | B0100 ACG00001 Andrew C. Goldstein, 10-Oct-1978 20:00
85 0085 1 | Previous revision history moved to [F11B.SRC]F11B.REV
86 0086 1 | **
87 0087 1 |
88 0088 1 |
89 0089 1 | LIBRARY 'SYSS$LIBRARY:LIB.L32';
90 0090 1 | REQUIRE 'SRC$:FCPDEF.B32';
```

: Re

.....
NUM

.....
1

```
92 1081 1 GLOBAL ROUTINE BUILD_EXT_FCBS (PRIMHDR, PFCB) : L_NORM NOVALUE =
93 1082 1
94 1083 1 +-
95 1084 1
96 1085 1 FUNCTIONAL DESCRIPTION:
97 1086 1
98 1087 1 Build the extension fcb chain starting with the primary
99 1088 1 fcb and header. Update the size in the primary fcb. Turn
100 1089 1 the header back when done.
101 1090 1
102 1091 1 SIDE EFFECTS:
103 1092 1 new FCBs created, primary fcb modified
104 1093 1
105 1094 1 --
106 1095 1
107 1096 2 BEGIN
108 1097 2
109 1098 2 MAP
110 1099 2 PRIMHDR : REF BBLOCK; ! file header arg
111 1100 2
112 1101 2 BIND_COMMON;
113 1102 2
114 1103 2 EXTERNAL ROUTINE
115 1104 2 CREATE_FCB : L_NORM, ! create a new FCB
116 1105 2 NEXT_HEADER : L_NORM;
117 1106 2 READ_HEADER : L_NORM;
118 1107 2
119 1108 2 LOCAL
120 1109 2 FCB : REF BBLOCK,
121 1110 2 PRIMFCB : REF BBLOCK,
122 1111 2 NEW_FCB : REF BBLOCK,
123 1112 2 HEADER : REF BBLOCK,
124 1113 2 NEW_HEADER : REF BBLOCK;
125 1114 2
126 1115 2 IF ACTUALCOUNT EQL 2
127 1116 2 THEN
128 1117 3 BEGIN
129 1118 3 PRIMFCB = .PFCB;
130 1119 3
131 1120 3 ! This is a flag for READ_HEADER to tell it not to update FILE HEADER.
132 1121 3 ! This prevents it from being set when dealing with directory headers.
133 1122 3 ! The flag is a one-shot cleared by READ_HEADER (which may be called
134 1123 3 ! by NEXT_HEADER).
135 1124 3
136 1125 3 STSFLGS [STS_LEAVE_FILEHDR] = 1;
137 1126 3 END
138 1127 2 ELSE
139 1128 2 PRIMFCB = .PRIMARY_FCB;
140 1129 2
141 1130 2 FCB = .PRIMFCB;
142 1131 2 HEADER = .PRIMHDR;
143 1132 2
144 1133 2 UNTIL (NEW_HEADER = NEXT_HEADER (.HEADER, .FCB)) EQL 0
145 1134 2 DO
146 1135 3 BEGIN
147 1136 3 HEADER = .NEW_HEADER;
148 1137 3
```

```

149 1138 3 IF ACTUALCOUNT EQL 2
150 1139 THEN
151 1140 NEW_FCB = CREATE_FCB (.HEADER, .PRIMFCB)
152 1141 ELSE
153 1142 NEW_FCB = CREATE_FCB (.HEADER);
154 1143
155 1144 CURRENT_VCB [VCBSW_TRANS] = .CURRENT_VCB [VCBSW_TRANS] + 1;
156 1145 NEW_FCB [FCBSW_REFcnt] = 1;
157 1146 NEW_FCB [FCBSL_LOCKBASIS] = .PRIMFCB [FCBSL_LOCKBASIS];
158 1147 NEW_FCB [FCBSL_STVBN] = .NEW_FCB [FCBSL_STVBN] + .PRIMFCB [FCBSL_FILESIZE];
159 1148 PRIMFCB [FCBSL_FILESIZE] = .PRIMFCB [FCBSL_FILESIZE]
160 1149 + .NEW_FCB [FCBSL_FILESIZE];
161 1150 FCB [FCBSL_EXFCB] = .NEW_FCB;
162 1151 FCB = .NEW_FCB;
163 1152
164 1153 ! Set it up for the next NEXT_HEADER or the possible READ_HEADER
165 1154 ! if we drop out of this loop.
166 1155 !
167 1156
168 1157 IF ACTUALCOUNT EQL 2
169 1158 THEN
170 1159 STSFLGS [STS_LEAVE_FILEHDR] = 1;
171 1160
172 1161 END;
173 1162
174 1163 IF .FCB NEQ .PRIMFCB
175 1164 THEN
176 1165 BEGIN
177 1166 HEADER = READ_HEADER (0, .PRIMFCB);
178 1167
179 1168 PRIMFCB [FCBSL_EFBLK] = ROT (.BBLOCK[HEADER[FH2$W_RECATTR], FAT$S_EFBLK], 16);
180 1169
181 1170 IF .PRIMFCB [FCBSL_EFBLK] NEQ 0
182 1171 AND .BBLOCK[HEADER[FH2$W_RECATTR], FAT$W_FFBYTE] EQL 0
183 1172 THEN
184 1173 PRIMFCB [FCBSL_EFBLK] = .PRIMFCB [FCBSL_EFBLK] - 1;
185 1174
186 1175 IF .PRIMFCB [FCBSL_EFBLK] GTR .PRIMFCB [FCBSL_FILESIZE]
187 1176 THEN
188 1177 PRIMFCB [FCBSL_EFBLK] = .PRIMFCB [FCBSL_FILESIZE];
189 1178
190 1179 END;
191 1180
192 1181 STSFLGS [STS_LEAVE_FILEHDR] = 0;
193 1182
194 1183 1 END;

```

! end of routine BUILD_EXT_FCBS

.TITLE EXTFCB
.IDENT \V04-000\.EXTRN CREATE_FCB, NEXT_HEADER
.EXTRN READ_HEADER

.PSECT \$CODE\$,NOWRT,2

007C 0000

.ENTRY BUILD_EXT_FCBS, Save R2,R3,R4,R5,R6

: 1081

	02		6C	91	00002		CMPB	(AP), #2	:	1115	
			0A	12	00005		BNEQ	1\$:		
A6	52	08	AC	D0	00007		MOVL	PF'CB, PRIMFCB	:	1118	
	AA		08	88	0000B		BISB2	#8, -90(BASE)	:	1125	
			04	11	0000F		BRB	2\$:	1115	
	52	08	AA	D0	00011	1\$:	MOVL	8(BASE), PRIMFCB	:	1128	
	55		52	D0	00015	2\$:	MOVL	PRIMFCB, FCB	:	1130	
	54	04	AC	D0	00018		MOVL	PRIMHDR, HEADER	:	1131	
0000G	CF		30	BB	0001C	3\$:	PUSHR	#^M<R4,R5>	:	1133	
	56		02	FB	0001E		CALLS	#2, NEXT HEADER	:		
			50	D0	00023		MOVL	R0, NEW_HEADER	:		
	54		49	13	00026		BEQL	6\$:		
	02		56	D0	00028		MOVL	NEW_HEADER, HEADER	:	1136	
			6C	91	0002B		CMPB	(APT), #2	:	1138	
			0B	12	0002E		BNEQ	4\$:		
			52	DD	00030		PUSHL	PRIMFCB	:	1140	
0000G	CF		54	DD	00032		PUSHL	HEADER	:		
			02	FB	00034		CALLS	#2, CREATE_FCB	:		
			07	11	00039		BRB	5\$:		
0000G	CF		54	DD	0003B	4\$:	PUSHL	HEADER	:	1142	
			01	FB	0003D		CALLS	#1, CREATE_FCB	:		
	53		50	D0	00042	5\$:	MOVL	R0, NEW_FCB	:		
	50	98	AA	D0	00045		MOVL	-104(BASE), R0	:	1144	
		0C	A0	B6	00049		INCR	12(R0)	:		
18	A3		01	B0	0004C		MOVW	#1, 24(NEW_FCB)	:	1145	
4C	A3	4C	A2	D0	00050		MOVL	76(PRIMFCB), 76(NEW_FCB)	:	1146	
2C	A3	38	A2	C0	00055		ADDL2	56(PRIMFCB), 44(NEW_FCB)	:	1147	
38	A2	38	A3	C0	0005A		ADDL2	56(NEW_FCB), 56(PRIMFCB)	:	1149	
0C	A5		53	D0	0005F		MOVL	NEW_FCB, 12(FCB)	:	1150	
	55		53	D0	00063		MOVL	NEW_FCB, FCB	:	1151	
	02		6C	91	00066		CMPB	(APT), #2	:	1157	
			B1	12	00069		BNEQ	3\$:		
A6	AA		08	88	0006B		BISB2	#8, -90(BASE)	:	1159	
			AB	11	0006F		BRB	3\$:	1133	
	52		55	D1	00071	6\$:	CMPL	FCB, PRIMFCB	:	1163	
			28	13	00074		BEQL	8\$:		
			52	DD	00076		PUSHL	PRIMFCB	:	1166	
			7E	D4	00078		CLRL	-(SP)	:		
0000G	CF		02	FB	0007A		CALLS	#2, READ HEADER	:		
	54		50	D0	0007F		MOVL	R0, HEADER	:		
3C	A2	1C	10	9C	00082		ROTL	#16, 28(HEADER), 60(PRIMFCB)	:	1168	
			08	13	00088		BEQL	7\$:	1170	
		20	A4	B5	0008A		TSTW	32(HEADER)	:	1171	
			03	12	0008D		BNEQ	7\$:		
		3C	A2	D7	0008F		DECL	60(PRIMFCB)	:	1173	
	38	A2	3C	A2	D1	00092	7\$:	CMPL	60(PRIMFCB), 56(PRIMFCB)	:	1175
			05	15	00097		BLEQ	8\$:		
	3C	A2	38	A2	D0	00099		MOVL	56(PRIMFCB), 60(PRIMFCB)	:	1177
	A6	AA	08	8A	0C09E	8\$:	BICB2	#8, -90(BASE)	:	1181	
			04	000A2			RET		:	1183	

: Routine Size: 163 bytes, Routine Base: \$CODE\$ + 0000

: 195 1184 1
: 196 1185 1 END
: 197 1186 0 ELUDOM

PSECT SUMMARY

```
:  
: Name Bytes Attributes  
: $CODE$ 163 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 27 0 1000 00:02.0
```

COMMAND QUALIFIERS

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

```
: Size: 163 code + 0 data bytes  
: Run Time: 00:17.5  
: Elapsed Time: 00:30.5  
: Lines/CPU Min: 4075  
: Lexemes/CPU-Min: 50577  
: Memory Used: 218 pages  
: Compilation Complete
```


[Screenshot]	[Screenshot]	[Screenshot]	[Screenshot]	[Screenshot]	[Screenshot]	[Screenshot]	[Screenshot]	[Screenshot]	[Screenshot]
[Screenshot]	[Screenshot]	[Screenshot]	[Screenshot]	[Screenshot]	[Screenshot]	[Screenshot]	[Screenshot]	[Screenshot]	[Screenshot]
[Screenshot]	[Screenshot]	[Screenshot]	[Screenshot]	[Screenshot]	EXTIDX LIS	[Screenshot]	[Screenshot]	[Screenshot]	[Screenshot]
[Screenshot]	[Screenshot]	[Screenshot]	[Screenshot]	[Screenshot]	[Screenshot]	[Screenshot]	[Screenshot]	[Screenshot]	[Screenshot]
[Screenshot]	[Screenshot]	[Screenshot]	[Screenshot]	[Screenshot]	[Screenshot]	[Screenshot]	[Screenshot]	[Screenshot]	[Screenshot]
[Screenshot]	[Screenshot]	[Screenshot]	[Screenshot]	[Screenshot]	[Screenshot]	[Screenshot]	[Screenshot]	[Screenshot]	[Screenshot]
[Screenshot]	[Screenshot]	[Screenshot]	[Screenshot]	[Screenshot]	[Screenshot]	[Screenshot]	[Screenshot]	[Screenshot]	[Screenshot]
[Screenshot]	[Screenshot]	[Screenshot]	[Screenshot]	[Screenshot]	[Screenshot]	[Screenshot]	[Screenshot]	[Screenshot]	[Screenshot]
[Screenshot]	[Screenshot]	[Screenshot]	[Screenshot]	[Screenshot]	[Screenshot]	[Screenshot]	[Screenshot]	[Screenshot]	[Screenshot]
[Screenshot]	[Screenshot]	[Screenshot]	[Screenshot]	[Screenshot]	[Screenshot]	[Screenshot]	[Screenshot]	[Screenshot]	[Screenshot]

ERASE
LIS

EXTEND
LIS

EXTHDR
LIS

FILUTL
LIS

GETLOC
LIS

GETREQ
LIS

GETTIM
LIS

FIND
LIS

GETFIB
LIS

INIFC2
LIS

INIFCP
LIS

EXTFCB
LIS

FILESERV
LIS

FILESIZE
LIS

GETPFR
LIS