

FFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFF
FFF
FFF
FFF
FFF
FFF
FFF
FFFFFFFFFFFFFF
FFFFFFFFFFFFFF
FFFFFFFFFFFFFF
FFF
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
111
1111111111
1111111111
1111111111

111
111
111
111111
111111
111111
111
111
111
111
111
111
111
111
111
111
111
111
111
111
1111111111
1111111111
1111111111

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
AAA AAA
AAA AAA

```
EEEEEEEEEE XX XX TTTTTTTTTT HH HH DDDDDDDD RRRRRRRR
EEEEEEEEEE XX XX TTTTTTTTTT HH HH DDDDDDDD RRRRRRRR
EE XX XX TT HH HH DD DD RR RR
EE XX XX TT HH HH DD DD RR RR
EE XX XX TT HH HH DD DD RR RR
EEEEEEEEE XX XX TT HH HH DD DD RRRRRRRR
EEEEEEEEE XX XX TT HH HH DD DD RRRRRRRR
EE XX XX TT HH HH DD DD RR RR
EE XX XX TT HH HH DD DD RR RR
EEEEEEEEEE XX XX TT HH HH DDDDDDDD RR RR
EEEEEEEEEE XX XX TT HH HH DDDDDDDD RR RR
```

```
LL IIIIIII SSSSSSSS
LL IIIIIII 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
LLLLLLLLLLL IIIIIII SSSSSSSS
LLLLLLLLLLL IIIIIII SSSSSSSS
```

```

0001 0 MODULE EXTHDR (
0002 0
0003 0          LANGUAGE (BLISS32),
0004 0          IDENT = 'V04-000'
0005 0 ) =
0006 1 BEGIN
0007 1
0008 1 *****
0009 1 *
0010 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0011 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0012 1 * ALL RIGHTS RESERVED.
0013 1 *
0014 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0015 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0016 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0017 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0018 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0019 1 * TRANSFERRED.
0020 1 *
0021 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0022 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0023 1 * CORPORATION.
0024 1 *
0025 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0026 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0027 1 *
0028 1 *
0029 1 *****
0030 1
0031 1 **
0032 1
0033 1 FACILITY: F11ACP Structure Level 1
0034 1
0035 1 ABSTRACT:
0036 1
0037 1     This routine creates an extension file header for the given file
0038 1     header.
0039 1
0040 1 ENVIRONMENT:
0041 1
0042 1     STARLET operating system, including privileged system services
0043 1     and internal exec routines.
0044 1
0045 1 --
0046 1
0047 1
0048 1 AUTHOR: Andrew C. Goldstein, CREATION DATE: 27-Jul-1977 10:15
0049 1
0050 1 MODIFIED BY:
0051 1
0052 1     V02-001 LMP0005          L. Mark Pilant,          29-Dec-1981 15:00
0053 1     Add support for Cathedral windows.
0054 1
0055 1     A0100 ACG0001          Andrew C. Goldstein, 10-Oct-1978 20:01
0056 1     Previous revision history moved to f11A.REV
0057 1

```

EXTHDR
V04-000

K 9
16-Sep-1984 01:03:41
~~14-Sep-1984 12:29:33~~

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

```
.. 58          0058 1 !**  
.. 59          0059 1  
.. 60          0060 1  
.. 61          0061 1 LIBRARY 'SYSSLIBRARY:LIB.L32';  
.. 62          0062 1 REQUIRE 'SRCS:FCPDEF.B32';
```

EXT
V04-

.....

.....
S
R
E
L
I
A
C

```

: 64 0377 1 GLOBAL ROUTINE EXTEND_HEADER (OLD_HEADER, FCB) =
: 65 0378 1
: 66 0379 1  +-+
: 67 0380 1
: 68 0381 1  FUNCTIONAL DESCRIPTION:
: 69 0382 1
: 70 0383 1      This routine creates an extension file header for the given file
: 71 0384 1      header.
: 72 0385 1
: 73 0386 1
: 74 0387 1  CALLING SEQUENCE:
: 75 0388 1      EXTEND_HEADER (ARG1, ARG2)
: 76 0389 1
: 77 0390 1  INPUT PARAMETERS:
: 78 0391 1      ARG1: address of present last file header
: 79 0392 1      ARG2: address of present last FCB or 0
: 80 0393 1
: 81 0394 1  IMPLICIT INPUTS:
: 82 0395 1      CURRENT_WINDOW: address of file window or 0
: 83 0396 1      PRIMARY_FCB: primary FCB of file
: 84 0397 1      CURRENT_VCB: address of VCB of volume
: 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 new file header
: 94 0407 1
: 95 0408 1  SIDE EFFECTS:
: 96 0409 1      file header created, window turned, FCB created
: 97 0410 1
: 98 0411 1  --
: 99 0412 1
100 0413 2 BEGIN
101 0414 2
102 0415 2 MAP
103 0416 2     OLD_HEADER      : REF BBLOCK,      ! file header arg
104 0417 2     FCB         : REF BBLOCK;      ! FCB arg
105 0418 2
106 0419 2 LOCAL
107 0420 2     OLD_FID       : BBLOCK [FID$C_LENGTH], ! file ID of old header
108 0421 2     EXT_FID       : BBLOCK [FID$C_LENGTH], ! file ID of new header
109 0422 2     VBN          : BBLOCK [VBN$C_LENGTH], ! index file VBN of new header
110 0423 2     LBN          : BBLOCK [LBN$C_LENGTH], ! LBN of new header
111 0424 2     MAP_AREA     : REF BBLOCK,      ! file header map area
112 0425 2     NEW_HEADER   : REF BBLOCK;      ! address of new file header
113 0426 2
114 0427 2 EXTERNAL
115 0428 2     CLEANUP_FLAGS  : BITVECTOR,      ! cleanup action flags
116 0429 2     USER_STATUS   : VECTOR,         ! I/O status block for user
117 0430 2     PRIMARY_FCB   : REF BBLOCK,      ! primary FCB of file
118 0431 2     CURRENT_WINDOW : REF BBLOCK,      ! window of file
119 0432 2     CURRENT_VCB   : REF BBLOCK,      ! VCB of volume
: 120 0433 2     HEADER_LBN    : REF BBLOCK,      ! LBN of current file header

```

```
121 0434 2 NEW_FID; ! file number of new extension header
122 0435 2
123 0436 2 EXTERNAL ROUTINE
124 0437 2 INIT_FCB, ! initialize FCB
125 0438 2 TURN_WINDOW, ! update window
126 0439 2 CHECKSUM, ! checksum file header
127 0440 2 MARK_DIRTY, ! mark header for writeback
128 0441 2 CREATE_HEADER, ! create a new file ID and header
129 0442 2 INVALIDATE, ! discard a block buffer
130 0443 2 RESET_LBN, ! reassign buffer LBN
131 0444 2 WRITE_HEADER, ! write file header
132 0445 2 READ_HEADER, ! read file header
133 0446 2 MAKE_EXTFCB; ! create an extension FCB
134 0447 2
135 0448 2
136 0449 2 ! Save the file ID of the current last header. If the file is accessed, fix
137 0450 2 ! up the FCB if it is not the primary and turn the window to include blocks
138 0451 2 ! from the header if possible. Then prepare the old header for write-back.
139 0452 2
140 0453 2
141 0454 2 OLD_FID[FID$W_NUM] = .OLD_HEADER[FH1$W_FID_NUM];
142 0455 2 OLD_FID[FID$W_SEQ] = .OLD_HEADER[FH1$W_FID_SEQ];
143 0456 2 OLD_FID[FID$W_RVN] = 0;
144 0457 2
145 0458 2 IF .FCB NEQ 0 AND .FCB NEQ .PRIMARY_FCB
146 0459 2 THEN KERNEL_CALL (INIT_FCB, .FCB, .OLD_HEADER);
147 0460 2
148 0461 2 IF .CURRENT_WINDOW NEQ 0
149 0462 2 THEN IF NOT .CURRENT_WINDOW[WCB$V_CATHEDRAL]
150 0463 2 THEN KERNEL_CALL (TURN_WINDOW, .CURRENT_WINDOW, .OLD_HEADER, .PRIMARY_FCB[FCB$L_FILESIZE], .FCB[FCB$L_STVBN]
151 0464 2
152 0465 2 CHECKSUM (.OLD_HEADER);
153 0466 2
154 0467 2 ! Now create the new file ID. Map and read the corresponding block in the
155 0468 2 ! index file to extract the file sequence number; then punt the buffer.
156 0469 2 !
157 0470 2
158 0471 2 NEW_HEADER = CREATE_HEADER ();
159 0472 2 EXT_FID[FID$W_NUM] = .NEW_HEADER[FH1$W_FID_NUM];
160 0473 2 EXT_FID[FID$W_SEQ] = .NEW_HEADER[FH1$W_FID_SEQ];
161 0474 2 LBN = .HEADER_LBN;
162 0475 2 INVALIDATE (.NEW_HEADER);
163 0476 2
164 0477 2 ! Get back the old file header, which may or may not have been written out
165 0478 2 ! due to buffer pool thrashing. Check the segment number for overflow.
166 0479 2 ! Plug in the header extension linkage and write it.
167 0480 2 !
168 0481 2
169 0482 2 NEW_HEADER = READ_HEADER (OLD_FID, .FCB);
170 0483 2 MAP_AREA = .NEW_HEADER + .NEW_HEADER[FH1$B_MPOFFSET]*2;
171 0484 2 IF .MAP_AREA[FM1$B_EX_SEGNUM] GEQU 255
172 0485 2 THEN ERR_EXIT (SS$HEADERFULL);
173 0486 2 MAP_AREA[FM1$W_EX_FILNUM] = .EXT_FID[FID$W_NUM];
174 0487 2 MAP_AREA[FM1$W_EX_FILSEQ] = .EXT_FID[FID$W_SEQ];
175 0488 2
176 0489 2 CHECKSUM (.NEW_HEADER);
177 0490 2 WRITE_HEADER ();
```

```

: 178      0491 2 NEW_FID = 0;                                ! header extension is complete
: 179      0492 ~~~~~
: 180      0493 ! We now build the new file header over the old one in the same buffer, thus
: 181      0494 ! keeping the attributes.
: 182      0495 !
: 183      0496 ~~~~~
: 184      0497 RESET_LBN (.NEW_HEADER, .LBN);
: 185      0498 MARK_DIRTY (.NEW_HEADER);
: 186      0499 HEADER_LBN = .LBN;
: 187      0500 ~~~~~
: 188      0501 NEW_HEADER[FM1$W_FID_NUM] = .EXT_FID[FID$W_NUM];
: 189      0502 NEW_HEADER[FM1$W_FID_SEQ] = .EXT_FID[FID$W_SEQ];
: 190      0503 ~~~~~
: 191      0504 MAP_AREA[FM1$B_EX_SEGNUM] = .MAP_AREA[FM1$B_EX_SEGNUM] + 1;
: 192      0505 MAP_AREA[FM1$W_EX_FILNUM] = 0;
: 193      0506 MAP_AREA[FM1$W_EX_FILSEQ] = 0;
: 194      0507 MAP_AREA[FM1$B_INOSE] = 0;
: 195      0508 ~~~~~
: 196      0509 CH$FILL (0, .MAP_AREA[FM1$B_AVAIL]*2, .MAP_AREA+FM1$C_POINTERS);
: 197      0510 ~~~~~
: 198      0511 ! Finally create an extension FCB if the file is accessed.
: 199      0512 !
: 200      0513 ~~~~~
: 201      0514 IF .FCB NEQ 0
: 202      0515 THEN KERNEL_CALL (MAKE_EXTFCB, .NEW_HEADER, .FCB, 2);
: 203      0516 ~~~~~
: 204      0517 RETURN (.NEW_HEADER);
: 205      0518 ~~~~~
: 206      0519 1 END;                                ! end of routine EXTEND_HEADER

```

```

.TITLE EXTHDR
.IDENT  \V04-000\

.EXTRN CLEANUP_FLAGS, USER STATUS
.EXTRN PRIMARY_FCB, CURRENT_WINDOW
.EXTRN CURRENT_VCB, HEADER_LBN
.EXTRN NEW_FID, INIT_FCB
.EXTRN TURN_WINDOW, CHECKSUM
.EXTRN MARK_DIRTY, CREATE_HEADER
.EXTRN INVALIDATE, RESET_LBN
.EXTRN WRITE_HEADER, READ_HEADER
.EXTRN MAKE_EXTFCB, SYSS$CMKRNL

.PSECT $CODE$,NOWRT,2

.ENTRY EXTEND_HEADER, Save R2,R3,R4,R5,R6,R7 : 0377
MOVAB @#SYSS$CMKRNL, R7
SUBL2 #16, SP
MOVQ  OLD_HEADER, R2 : 0454
MOVL  2(R2), OLD_FID
CLRW  OLD_FID+4 : 0456
TSTL  R3 : 0458
BEQL  1$,
CMLL  R3, PRIMARY_FCB
BEQL  1$,
PUSHL R2 : 0459

```

```

00FC 0000
57 0000000G 9F 9E 00002
5E 10 C2 00009
08 52 04 AC 7D 0000C
AE 02 A2 D0 00010
OC AE B4 00015
53 D5 00018
16 13 0001A
0000G CF 53 D1 0001C
OF 13 00021
52 DD 00023

```

			53	DD	00025		PUSHL	R3		
			02	DD	00027		PUSHL	#2		
			5E	DD	00029		PUSHL	SP		
			0000G	CF	9F 00028		PUSHAB	INIT_FCB		
	67		05	FB	0002F		CALLS	#5, SYSSCMKRN		
	51		0000G	CF	D0 00032	1\$:	MOVL	CURRENT_WINDOW, R1		0461
			1D	13	00037		BEQL	2\$		
18	0B	A1	06	E0	00039		BBS	#6, 11(R1), 2\$		0462
		2C	A3	DD	0003E		PUSHL	44(R3)		0463
		50	0000G	CF	D0 00041		MOVL	PRIMARY_FCB, R0		
			38	A0	DD 00046		PUSHL	56(R0)		
			06	BB	00049		PUSHR	#*M<R1,R2>		
			04	DD	0004B		PUSHL	#4		
			5E	DD	0004D		PUSHL	SP		
			0000G	CF	9F 0004F		PUSHAB	TURN_WINDOW		
		67	07	FB	00053		CALLS	#7, SYSSCMKRN		
			52	DD	00056	2\$:	PUSHL	R2		0465
	0000G	CF	01	FB	00058		CALLS	#1, CHECKSUM		
	0000G	CF	00	FB	0005D		CALLS	#0, CREATE_HEADER		0471
		56	50	D0	00062		MOVL	R0, NEW_HEADER		
		6E	02	A6	D0 00065		MOVL	2(NEW_HEADER), EXT_FID		0472
		54	0000G	CF	D0 00069		MOVL	HEADER_LBN, LBN		0474
			56	DD	0006E		PUSHL	NEW_HEADER		0475
	0000G	CF	01	FB	00070		CALLS	#1, INVALIDATE		
			53	DD	00075		PUSHL	R3		0482
		0C	AE	9F	00077		PUSHAB	OLD_FID		
	0000G	CF	02	FB	0007A		CALLS	#2, READ_HEADER		
		56	50	D0	0007F		MOVL	R0, NEW_HEADER		
		50	01	A6	9A 00082		MOVZBL	1(NEW_HEADER), R0		0483
		52	6640	3E	00086		MOVAV	(NEW_HEADER)[R0], MAP_AREA		
	FF	8F	62	91	0008A		CMPB	(MAP_AREA), #255		0484
			05	1F	0008E		BLSSU	3\$		
		08C8	8F	BF	00090		CHMU	#2248		0485
			04	00094			RET			
	02	A2	6E	D0	00095	3\$:	MOVL	EXT_FID, 2(MAP_AREA)		0486
			56	DD	00099		PUSHL	NEW_HEADER		0489
	0000G	CF	01	FB	0009B		CALLS	#1, CHECKSUM		
	0000G	CF	00	FB	000A0		CALLS	#0, WRITE_HEADER		0490
			0000G	CF	D4 000A5		CLRL	NEW_FID		0491
			54	DD	000A9		PUSHL	LBN		0497
			56	DD	000AB		PUSHL	NEW_HEADER		
	0000G	CF	02	FB	000AD		CALLS	#2, RESET_LBN		
			56	DD	000B2		PUSHL	NEW_HEADER		0498
	0000G	CF	01	FB	000B4		CALLS	#1, MARK_DIRTY		
	0000G	CF	54	D0	000B9		MOVL	LBN, HEADER_LBN		0499
	02	A6	6E	D0	000BE		MOVL	EXT_FID, 2(NEW_HEADER)		0501
			62	96	000C2		INCB	(MAP_AREA)		0504
		02	A2	D4	000C4		CLRL	2(MAP_AREA)		0505
		08	A2	94	000C7		CLRB	8(MAP_AREA)		0507
		50	09	A2	9A 000CA		MOVZBL	9(MAP_AREA), R0		0509
		50	02	C4	000CE		MULL2	#2, R0		
50			00	2C	000D1		MOVCS	#0, (SP), #0, R0, 10(MAP_AREA)		
			0A	A2	000D6					
			08	AC	D5 000D8		TSTL	FCB		0514
			12	13	000DB		BEQL	4\$		
			02	DD	000DD		PUSHL	#2		0515
		08	AC	DD	000DF		PUSHL	FCB		

		56	DD	000E2	PUSHL	NEW_HEADER	:
		03	DD	000E4	PUSHL	#3	:
		5E	DD	000E6	PUSHL	SP	:
	0000G	CF	9F	000E8	PUSHAB	MAKE_EXTFCB	:
67		06	FB	000EC	CALLS	#6, SYSSCMKRN	:
50		56	D0	000EF	MOVL	NEW_HEADER, R0	: 0517
		04	000F2	4\$:	RET		: 0519

: Routine Size: 243 bytes, Routine Base: \$CODE\$ + 0000

```

: 207      0520 1
: 208      0521 1 END
: 209      0522 0 ELUDOM

```

PSECT SUMMARY

Name	Bytes	Attributes
\$CODE\$	243	NOVEC,NOWRT, RD, EXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)

Library Statistics

File	----- Symbols -----		Pages	Processing
	Total	Loaded		
_\$255\$DUA28:[SYSLIB]LIB.L32;1	18619	18	0	1000 00:01.9

COMMAND QUALIFIERS

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

```

: Size:          243 code + 0 data bytes
: Run Time:      00:09.1
: Elapsed Time: 00:26.1
: Lines/CPU Min: 3441
: Lexemes/CPU-Min: 15843
: Memory Used: 129 pages
: Compilation Complete

```

This image shows a large grid of approximately 150 small terminal windows, likely representing a collection of system utilities or application programs. Each window displays a different interface, often featuring a header line with the program name followed by "LIS" (List), such as "EXTFCB LIS", "DIRJL LIS", "DIRGET LIS", "EXTIOX LIS", "TODONE LIS", "LOCKDN LIS", "ENTER LIS", "GETREQ LIS", "GETTIM LIS", "DISPAT LIS", "DIRFCB LIS", "EXTHDR LIS", "DIRSCN LIS", "LOGDEL LIS", "DIRACC LIS", "EXTDIR LIS", "EXTEND LIS", "FIND LIS", "GETFIB LIS", "INIFCB LIS", and "LOCKDB LIS". The windows also display various data lists, status indicators, and control prompts. Some windows are partially obscured or overlapping, creating a dense, multi-layered visual effect. The overall appearance is that of a multi-screen terminal environment from the VAX/VMS era.