

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	1111111111	1111111111	XXX	XXX
FFF	1111111111	1111111111	XXX	XXX
FFF	1111111111	1111111111	XXX	XXX

Symb

IOCS  
IO\_C  
IO\_C  
IO\_D  
IO\_F  
IO\_S  
KICL

KILL  
KILL  
LB\_E  
LB\_C  
LB\_F  
LB\_H  
LB\_L  
LOCAL  
LOCK

LOCK  
LOCK  
LOCK  
LOC\_

LOC\_

L\_CC  
L\_CC  
L\_DA  
L\_DA

MAIN  
MAKE  
MAKE  
MAKE  
MAKE  
MAKE

MAKE  
MAKE  
MAP\_

MAP  
MARI  
MARI  
MARI  
MARI

```

NN      NN  XX      XX  TTTTTTTTTT  HH      HH  DDDDDDDD  RRRRRRRR
NN      NN  XX      XX  TTTTTTTTTT  HH      HH  DDDDDDDD  RRRRRRRR
NN      NN  XX      XX      TT      HH      HH  DD      DD  RR      RR
NN      NN  XX      XX      TT      HH      HH  DD      DD  RR      RR
NNNN    NN      XX  XX  TT      HH      HH  DD      DD  RR      RR
NNNN    NN      XX  XX  TT      HH      HH  DD      DD  RR      RR
NN  NN  NN      XX      TT      HHHHHHHHHH  DD      DD  RRRRRRRR
NN  NN  NN      XX      TT      HHHHHHHHHH  DD      DD  RRRRRRRR
NN      NNNN    XX  XX  TT      HH      HH  DD      DD  RR  RR
NN      NNNN    XX  XX  TT      HH      HH  DD      DD  RR  RR
NN      NN  XX      XX  TT      HH      HH  DD      DD  RR      RR
NN      NN  XX      XX  TT      HH      HH  DDDDDDDD  RR      RR
NN      NN  XX      XX  TT      HH      HH  DDDDDDDD  RR      RR

```

```

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 NXTHDR (
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 *
11 0011 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY *
12 0012 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. *
13 0013 1 * ALL RIGHTS RESERVED. *
14 0014 1 *
15 0015 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
16 0016 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE *
17 0017 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER *
18 0018 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
19 0019 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY *
20 0020 1 * TRANSFERRED. *
21 0021 1 *
22 0022 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE *
23 0023 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT *
24 0024 1 * CORPORATION. *
25 0025 1 *
26 0026 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS *
27 0027 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL. *
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 routine reads the next extension header, if any, of the
38 0038 1 given file.
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.
44 0044 1
45 0045 1 --
46 0046 1
47 0047 1
48 0048 1 AUTHOR: Andrew C. Goldstein, CREATION DATE: 22-Jul-1977 17:40
49 0049 1
50 0050 1 MODIFIED BY:
51 0051 1
52 0052 1 V03-001 CDS0001 Christian D. Saether 30-Dec-1983
53 0053 1 Use L_NORM linkage and BIND_COMMON macro.
54 0054 1
55 0055 1 B0100 ACG0001 Andrew C. Goldstein, 10-Oct-1978 20:00
56 0056 1 Previous revision history moved to [F11B.SRC]F11B.REV
57 0057 1 **
    
```



```

63 1052 1 GLOBAL ROUTINE NEXT_HEADER (HEADER, FCB, EXT_FID, SEGNUM) : L_NORM =
64 1053 1
65 1054 1 !++
66 1055 1
67 1056 1 FUNCTIONAL DESCRIPTION:
68 1057 1
69 1058 1 This routine reads the next extension header, if any, of the
70 1059 1 indicated file. Extension data is taken from either the indicated
71 1060 1 file header or the arguments.
72 1061 1
73 1062 1
74 1063 1 CALLING SEQUENCE:
75 1064 1 NEXT_HEADER (ARG1, ARG2, ARG3, ARG4)
76 1065 1
77 1066 1 INPUT PARAMETERS:
78 1067 1 ARG1: address of current file header or 0
79 1068 1 ARG2: address of corresponding FCB or zero
80 1069 1 ARG3: extension file ID, if present
81 1070 1 ARG4: extension segment number, if present
82 1071 1
83 1072 1 IMPLICIT INPUTS:
84 1073 1 NONE
85 1074 1
86 1075 1 OUTPUT PARAMETERS:
87 1076 1 NONE
88 1077 1
89 1078 1 IMPLICIT OUTPUTS:
90 1079 1 NONE
91 1080 1
92 1081 1 ROUTINE VALUE:
93 1082 1 Address of header read or 0 if none
94 1083 1
95 1084 1 SIDE EFFECTS:
96 1085 1 File header may be read
97 1086 1
98 1087 1 --
99 1088 1
100 1089 2 BEGIN
101 1090 2
102 1091 2 MAP
103 1092 2 HEADER : REF BBLOCK, ! file header arg
104 1093 2 FCB : REF BBLOCK, ! FCB arg
105 1094 2 EXT_FID : REF BBLOCK; ! extension file ID arg
106 1095 2
107 1096 2 LOCAL
108 1097 2 NEW_HEADER : REF BBLOCK, ! address of extension file header read
109 1098 2 EXT_FCB : REF BBLOCK, ! address of extension FCB
110 1099 2 FILE_ID : BBLOCK [FID$C_LENGTH], ! file ID of extension header
111 1100 2 SEG_NUMBER : WORD; ! segment number of file header
112 1101 2
113 1102 2 BIND_COMMON;
114 1103 2
115 1104 2 EXTERNAL ROUTINE
116 1105 2 READ_HEADER : L_NORM; ! read a file header
117 1106 2
118 1107 2
119 1108 2 ! Get the extension file number of the file header. If it is zero, then

```

```

: 120      1109 2 ! there is no extension header. If it is non-zero, read the header, using
: 121      1110 2 ! the extension FCB if one exists.
: 122      1111 2 !
: 123      1112 2 !
: 124      1113 2 IF ACTUALCOUNT LSS 4
: 125      1114 2 THEN
: 126      1115 2     BEGIN
: 127      1116 2     CH$MOVE (FID$ LENGTH, HEADER[FH2$W_EXT_FID], FILE_ID);
: 128      1117 2     SEG_NUMBER = .HEADER[FH2$W_SEG_NUM]+ 1;
: 129      1118 2     END
: 130      1119 2 ELSE
: 131      1120 2     BEGIN
: 132      1121 2     CH$MOVE (FID$ LENGTH, .EXT_FID, FILE_ID);
: 133      1122 2     SEG_NUMBER = .SEGNUM;
: 134      1123 2     END;
: 135      1124 2
: 136      1125 2 IF .FILE_ID[FID$W_NUM] EQL 0
: 137      1126 2 AND (
: 138      1127 2     IF .CURRENT_VCB[VCB$V_EXTFID]
: 139      1128 2     THEN .FILE_ID[FID$B_NMX] EQL 0
: 140      1129 2     ELSE 1
: 141      1130 2 )
: 142      1131 2 THEN RETURN 0;
: 143      1132 2 EXT_FCB =
: 144      1133 2     (IF .FCB NEQ 0
: 145      1134 2     THEN .FCB[FCB$L_EXFCB]
: 146      1135 2     ELSE 0
: 147      1136 2 );
: 148      1137 2 NEW_HEADER = READ_HEADER (FILE_ID, .EXT_FCB);
: 149      1138 2
: 150      1139 2 ! Check the segment number of the header read for consistency.
: 151      1140 2 !
: 152      1141 2 !
: 153      1142 2 IF .SEG_NUMBER NEQ .NEW_HEADER[FH2$W_SEG_NUM]
: 154      1143 2 THEN ERR_EXIT (SS$_BADFILEHDR);
: 155      1144 2
: 156      1145 2 RETURN .NEW_HEADER;
: 157      1146 2
: 158      1147 1 END;

```

! end of routine NEXT\_HEADER

					.TITLE	NXTHDR		
					.IDENT	\V04-000\		
					.EXTRN	READ_HEADER		
					.PSECT	\$CODE\$,NOWRT,2		
					.ENTRY	NEXT_HEADER, Save R2,R3,R4,R5,R6	:	1052
		5E		08 C2 00002	SUBL2	#8, SP	:	
		04		6C 91 00005	CMPB	(AP), #4	:	1113
				10 1E 00008	BGEQU	1\$	:	
		56	04	AC D0 0000A	MOVL	HEADER, R6	:	1116
6E	0E	A6		06 28 0000E	MOV3	#6, 14(R6), FILE_ID	:	
52	04	A6		01 A1 00013	ADDW3	#1, 4(R6), SEG_NUMBER	:	1117
				09 11 00018	BRB	2\$	:	1113
6E	0C	BC		06 28 0001A 1\$:	MOV3	#6, @EXT_FID, FILE_ID	:	1121

		52	10	AC B0 0001F	MOVW	SEGNUM, SEG_NUMBER	:	1122
				6E B5 00023 2\$:	TSTW	FILE_ID	:	1125
				0E 12 00025	BNEQ	3\$	:	
		50	98	AA D0 00027	MOVL	-104(BASE), R0	:	1127
2F	0B	A0		05 E1 0002B	BBC	#5, 11(R0), 7\$	:	
				05 AE 95 00030	TSTB	FILE_ID+5	:	1128
				2A 13 00033	BEQL	7\$	:	
		50	08	AC D0 00035 3\$:	MOVL	FCB, R0	:	1133
				06 13 00039	BEQL	4\$	:	
		50	0C	A0 D0 0003B	MOVL	12(R0), EXT_FCB	:	1134
				02 11 0003F	BRB	5\$	:	
				50 D4 00041 4\$:	CLRL	EXT_FCB	:	1133
				50 DD 00043 5\$:	PUSHL	EXT_FCB	:	1137
			04	AE 9F 00045	PUSHAB	FILE_ID	:	
0000G	CF			02 FB 00048	CALLS	#2, READ HEADER	:	
	51			50 D0 0004D	MOVL	R0, NEW_HEADER	:	
	04	A1		52 B1 00050	CMPW	SEG_NUMBER, 4(NEW_HEADER)	:	1142
				05 13 00054	BEQL	6\$	:	
			0810	8F BF 00056	CHMU	#2064	:	1143
				04 0005A	RET		:	
		50		51 D0 0005B 6\$:	MOVL	NEW_HEADER, R0	:	1145
				04 0005E	RET		:	
				50 D4 0005F 7\$:	CLRL	R0	:	1147
				04 00061	RET		:	

; Routine Size: 98 bytes, Routine Base: \$CODE\$ + 0000

```
; 159      1148  1
; 160      1149  1 END
; 161      1150  0 ELUDOM
```

PSECT SUMMARY

Name	Bytes	Attributes
\$CODE\$	98	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	24	0	1000	00:02.0

NXTHDR  
V04-000

E 14  
16-Sep-1984 00:48:15  
14-Sep-1984 12:30:39

VAX-11 Bliss-32 V4.0-742  
DISK\$VMMASTER:[F11X.SRC]NXTHDR.B32;1 Page (2)

COMMAND QUALIFIERS

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

: Size: 98 code + 0 data bytes  
: Run Time: 00:15.8  
: Elapsed Time: 00:36.5  
: Lines/CPU Min: 4364  
: Lexemes/CPU-Min: 53214  
: Memory Used: 198 pages  
: Compilation Complete



0171 AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY

