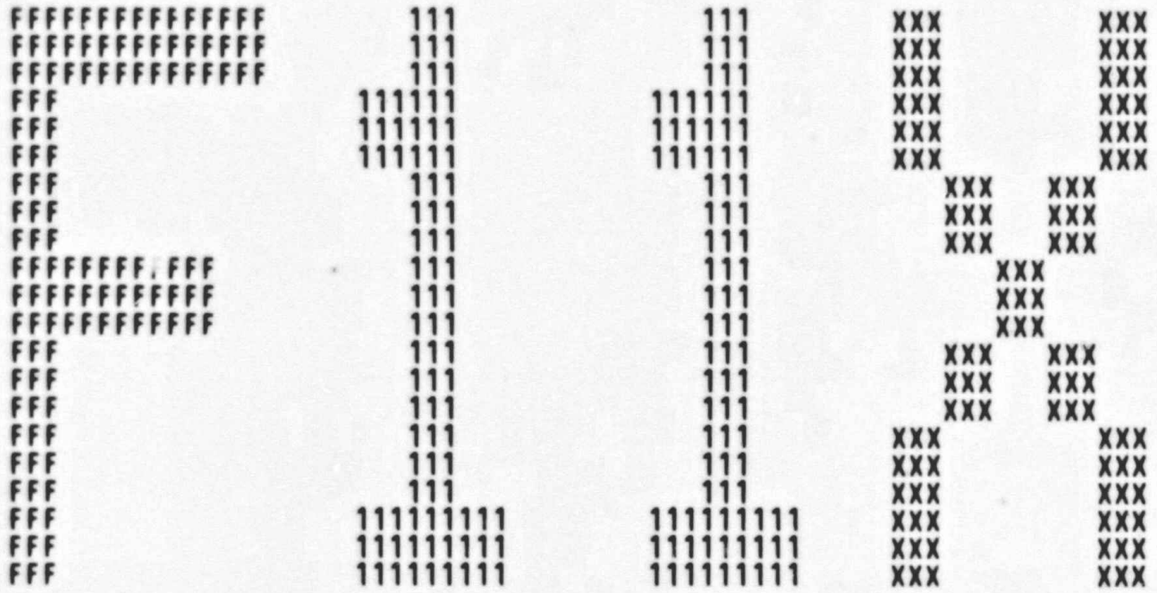

_$2

Symb

IOCS
IO_C
IO_C
IO_D
IO_F
IO_S
KICL

KILL
KILL
LB_B
LB_D
LB_F
LB_K
LB_L
LOCA
LOCA
LOCK

LOCK
LOCK
LOCK
LOC_
LOC_
L_CO
L_CO
L_DA
L_DA
MAIN
MAKE
MAKE
MAKE
MAKE
MAKE
MAKE

MAKE
MAKE
MAP_
MAP_

MAP
MARK
MARK
MARK
MARK
MARK

FILEID**CHKHD2

```
  CCCCCCCC  HH      HH  KK      KK  HH      HH  DDDDDDDD     222222
  CCCCCCCC  HH      HH  KK      KK  HH      HH  DDDDDDDD     222222
CC          HH      HH  KK      KK  HH      HH  DD      DD  22      22
CC          HH      HH  KK      KK  HH      HH  DD      DD  22      22
CC          HH      HH  KK    KK    HH      HH  DD      DD          22
CC          HH      HH  KK    KK    HH      HH  DD      DD          22
CC          HHHHHHHHHH  KKKKKK      HHHHHHHHHH  DD      DD        22
CC          HHHHHHHHHH  KKKKKK      HHHHHHHHHH  DD      DD        22
CC          HH      HH  KK    KK    HH      HH  DD      DD      22
CC          HH      HH  KK    KK    HH      HH  DD      DD      22
CC          HH      HH  KK      KK  HH      HH  DD      DD    22          ....
CC          HH      HH  KK      KK  HH      HH  DD      DD    22          ....
  CCCCCCCC  HH      HH  KK      KK  HH      HH  DDDDDDDD    2222222222  ....
  CCCCCCCC  HH      HH  KK      KK  HH      HH  DDDDDDDD    2222222222  ....

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 CHKHD2 (
;     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 2
;    34          0034  1 !
;    35          0035  1 ! ABSTRACT:
;    36          0036  1 !
;    37          0037  1 !       This routine verifies that the block given it is in fact a
;    38          0038  1 !       file header. If file number and/or file seqence number are also
;    39          0039  1 !       supplied, they are checked as well.
;    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.
;    45          0045  1 !
;    46          0046  1 !--
;    47          0047  1 !
;    48          0048  1 !
;    49          0049  1 ! AUTHOR:  Andrew C. Goldstein,  CREATION DATE:  13-Dec-1976  16:11
;    50          0050  1 !
;    51          0051  1 ! MODIFIED BY:
;    52          0052  1 !
;    53          0053  1 !       V03-005 ACG0408         Andrew C. Goldstein,    23-Mar-1984  11:31
;    54          0054  1 !               Remove external reference to USER_STATUS
;    55          0055  1 !
;    56          0056  1 !       V03-004 CDS0003         Christian D. Saether    18-Jan-1984
;    57          0057  1 !               ERR_STATUS macro declares USER_STATUS as an external.
```

```
;    58          0058  1 !                     Explicitly declare it to avoid truncation errors.
;    59          0059  1 !
;    60          0060  1 !               V03-003 CDS0002         Christian D. Saether    17-Jan-1984
;    61          0061  1 !                       Ooops.  Cannot use L_NORM linkage because this module
;    62          0062  1 !                       gets pulled out into SYSINIT and MOUNTSHR images, at least.
;    63          0063  1 !                       Remove test for EXTFID flag in CURRENT_VCB (we always use
;    64          0064  1 !                       extended file ID format).
;    65          0065  1 !
;    66          0066  1 !               V03-002 CDS0001         Christian D. Saether    29-Dec-1983
;    67          0067  1 !                       Use L_NORM linkage and BIND_COMMON macro.
;    68          0068  1 !
;    69          0069  1 !               V03-001 ACG0325         Andrew C. Goldstein,    3-Apr-1983  17:11
;    70          0070  1 !                       Change use of header area length symbol
;    71          0071  1 !
;    72          0072  1 !               V02-003 ACG0156         Andrew C. Goldstein,    12-Mar-1980  15:21
;    73          0073  1 !                       Fix header invalidation bug
;    74          0074  1 !
;    75          0075  1 !               B0102   ACG0146         Andrew C. Goldstein,    22-Feb-1980  21:40
;    76          0076  1 !                       Change file sequence number check to no such file
;    77          0077  1 !
;    78          0078  1 !               B0101   acg0003         Andrew C. Goldstein,  10-Nov-1978  19:29
;    79          0079  1 !                       Add multi-volume support
;    80          0080  1 !
;    81          0081  1 !               B0100   ACG00001        Andrew C. Goldstein,  10-Oct-1978  19:59
;    82          0082  1 !                       Previous revision history moved to [F11B.SRC]F11B.REV
;    83          0083  1 !**
;    84          0084  1
;    85          0085  1
;    86          0086  1 LIBRARY 'SYS$LIBRARY:LIB.L32';
;    87          0087  1 REQUIRE 'SRC$:FCPDEF.B32';
```

```
;    89          1078  1 GLOBAL ROUTINE CHECK_HEADER2 (HEADER, FILE_ID, HEADER_STATUS) =
;    90          1079  1
;    91          1080  1 !++
;    92          1081  1 !
;    93          1082  1 ! FUNCTIONAL DESCRIPTION:
;    94          1083  1 !
;    95          1084  1 !       This routine verifies that the block given it is in fact a
;    96          1085  1 !       file header. If file number and/or file seqence number are also
;    97          1086  1 !       supplied, they are checked as well.
;    98          1087  1 !
;    99          1088  1 ! CALLING SEQUENCE:
;   100          1089  1 !       CHECK_HEADER (ARG1, ARG2, ARG3)
;   101          1090  1 !
;   102          1091  1 ! INPUT PARAMETERS:
;   103          1092  1 !       ARG1: address of header image
;   104          1093  1 !       ARG2: address of file ID
;   105          1094  1 !
;   106          1095  1 ! IMPLICIT INPUTS:
;   107          1096  1 !       NONE
;   108          1097  1 !
;   109          1098  1 ! OUTPUT PARAMETERS:
;   110          1099  1 !       ARG3: (optional) address to store status return code
;   111          1100  1 !
;   112          1101  1 ! IMPLICIT OUTPUTS:
;   113          1102  1 !       USER_STATUS contains code if not valid
;   114          1103  1 !
;   115          1104  1 ! ROUTINE VALUE:
;   116          1105  1 !       0 if garbage
;   117          1106  1 !       1 if valid and correct file header
;   118          1107  1 !       2 if deleted file header
;   119          1108  1 !       4 if valid header but wrong sequence number
;   120          1109  1 !
;   121          1110  1 ! SIDE EFFECTS:
;   122          1111  1 !       NONE
;   123          1112  1 !
;   124          1113  1 !--
;   125          1114  1
;   126          1115  2 BEGIN
;   127          1116  2
;   128          1117  2 MAP
;   129          1118  2         HEADER          : REF BBLOCK,   ! file header arg
;   130          1119  2         FILE_ID         : REF BBLOCK,   ! file ID arg
;   131          1120  2         HEADER_STATUS   : REF VECTOR [,WORD]; ! status output arg
;   132          1121  2
;   133          1122  2 MACRO
;   134        M 1123  2         EXIT    (STATUS_CODE, HEADER_STATE) =
;   135        M 1124  2                 BEGIN
;   136        M 1125  2                 STATUS = HEADER_STATE;
;   137        M 1126  2                 IF ACTUALCOUNT GEQU 3
;   138        M 1127  2                 THEN IF .HEADER_STATUS[0]
;   139        M 1128  2                     THEN HEADER_STATUS[0] = STATUS_CODE;
;   140        M 1129  2                 RETURN .STATUS;
;   141        M 1130  2                 END
;   142          1131  2                 %;
;   143          1132  2
;   144          1133  2 LOCAL
;   145          1134  2         STATUS,                         ! return value of routine
```

```
;  146          1135  2          MAP_AREA        : REF BBLOCK;   ! pointer to header map area
;  147          1136  2
;  148          1137  2 EXTERNAL ROUTINE
;  149          1138  2          CHECKSUM;                       ! compute file header checksum
;  150          1139  2
;  151          1140  2
;  152          1141  2 ! First check the structure level.
;  153          1142  2 !
;  154          1143  2
;  155          1144  2 IF .HEADER[FH2$B_STRUCLEV] NEQ 2
;  156          1145  2 THEN EXIT (SS$_FILESTRUCT, 0);
;  157          1146  2
;  158          1147  2 ! Check the area offsets and the retrieval pointer use counts for
;  159          1148  2 ! consistency.
;  160          1149  2 !
;  161          1150  2
;  162          1151  2 IF .HEADER[FH2$B_IDOFFSET] LSSU $BYTEOFFSET (FH2$L_HIGHWATER)/2
;  163          1152  2 OR .HEADER[FH2$B_MPOFFSET] LSSU .HEADER[FH2$B_IDOFFSET]
;  164          1153  2 OR .HEADER[FH2$B_ACOFFSET] LSSU .HEADER[FH2$B_MPOFFSET]
;  165          1154  2 OR .HEADER[FH2$B_RSOFFSET] LSSU .HEADER[FH2$B_ACOFFSET]
;  166          1155  2 OR .HEADER[FH2$B_MAP_INUSE] GTRU .HEADER[FH2$B_ACOFFSET] - .HEADER[FH2$B_MPOFFSET]
;  167          1156  2 THEN EXIT (SS$_BADFILEHDR, 0);
;  168          1157  2
;  169          1158  2 ! At this point, we have verified that the block at least once was a
;  170          1159  2 ! valid file header.
;  171          1160  2 !
;  172          1161  2 ! Look at the file number in the header. If zero, this is a
;  173          1162  2 ! deleted header.
;  174          1163  2 !
;  175          1164  2
;  176          1165  2 IF .HEADER[FH2$W_FID_NUM] EQL 0
;  177          1166  2 AND .HEADER[FH2$B_FID_NMX] EQL 0
;  178          1167  2 THEN EXIT (SS$_NOSUCHFILE, 2);
;  179          1168  2
;  180          1169  2 ! Now compute the header checksum.
;  181          1170  2 !
;  182          1171  2
;  183          1172  2 IF NOT CHECKSUM (.HEADER)
;  184          1173  2 THEN EXIT (SS$_BADCHKSUM, 2);
;  185          1174  2
;  186          1175  2 ! Check file number and file sequence number.
;  187          1176  2 !
;  188          1177  2
;  189          1178  2 IF .HEADER[FH2$W_FID_NUM] NEQ .FILE_ID[FID$W_NUM]
;  190          1179  2 OR .HEADER[FH2$B_FID_NMX] NEQ .FILE_ID[FID$B_NMX]
;  191          1180  2 THEN EXIT (SS$_FILENUMCHK, 2);
;  192          1181  2
;  193          1182  2 IF .HEADER[FH2$W_FID_SEQ] NEQ .FILE_ID[FID$W_SEQ]
;  194          1183  2 THEN EXIT (SS$_NOSUCHFILE, 4);
;  195          1184  2
;  196          1185  2 ! Header is ok.
;  197          1186  2 !
;  198          1187  2
;  199          1188  2 RETURN 1;
;  200          1189  2
;  201          1190  1 END;                                     ! end of routine CHECK_HEADER
```

```
                                                        .TITLE  CHKHD2
                                                        .IDENT  \V04-000\

                                                        .EXTRN  CHECKSUM

                                                        .PSECT  $CODE$,NOWRT,2

                                    000C 00000          .ENTRY  CHECK_HEADER2, Save R2,R3                       ; 1078
                          50    04  AC  D0 00002        MOVL    HEADER, R0                                      ; 1144
                          02    07  A0  91 00006        CMPB    7(R0), #2                                       ;
                                    13  13 0000A        BEQL    1$                                              ;
                                    52  D4 0000C        CLRL    STATUS                                          ; 1145
                          03        6C  91 0000E        CMPB    (AP), #3                                        ;
                                    7F  1F 00011        BLSSU   5$                                              ;
                          7B    0C  BC  E9 00013        BLBC    @HEADER_STATUS, 5$                              ;
                    0C    BC  08C0  8F  B0 00017        MOVW    #2240, @HEADER_STATUS                           ;
                                    73  11 0001D        BRB     5$                                              ;
                          50    04  AC  D0 0001F 1$:    MOVL    HEADER, R0                                      ; 1151
                          26        60  91 00023        CMPB    (R0), #38                                       ;
                                    27  1F 00026        BLSSU   2$                                              ;
                          60    01  A0  91 00028        CMPB    1(R0), (R0)                                     ; 1152
                                    21  1F 0002C        BLSSU   2$                                              ;
                    01    A0    02  A0  91 0002E        CMPB    2(R0), 1(R0)                                    ; 1153
                                    1A  1F 00033        BLSSU   2$                                              ;
                    02    A0    03  A0  91 00035        CMPB    3(R0), 2(R0)                                    ; 1154
                                    13  1F 0003A        BLSSU   2$                                              ;
                          51    02  A0  9A 0003C        MOVZBL  2(R0), R1                                       ; 1155
                          53    01  A0  9A 00040        MOVZBL  1(R0), R3                                       ;
                          51        53  C2 00044        SUBL2   R3, R1                                          ;
          51        3A    A0        08  00  ED 00047    CMPZV   #0, #8, 58(R0), R1                              ;
                                    13  1B 0004D        BLEQU   3$                                              ;
                                    52  D4 0004F 2$:     CLRL    STATUS                                          ; 1156
                          03        6C  91 00051        CMPB    (AP), #3                                        ;
                                    78  1F 00054        BLSSU   10$                                             ;
                          76    0C  BC  E9 00056        BLBC    @HEADER_STATUS, 11$                             ;
                    0C    BC  0810  8F  B0 0005A        MOVW    #2064, @HEADER_STATUS                           ;
                                    78  11 00060        BRB     12$                                             ;
                          50    04  AC  D0 00062 3$:    MOVL    HEADER, R0                                      ; 1165
                                08  A0  B5 00066        TSTW    8(R0)                                           ;
                                    0A  12 00069        BNEQ    4$                                              ;
                                0D  A0  95 0006B        TSTB    13(R0)                                          ; 1166
                                    05  12 0006E        BNEQ    4$                                              ;
                          52        02  D0 00070        MOVL    #2, STATUS                                      ; 1167
                                    56  11 00073        BRB     9$                                              ;
                                04  AC  DD 00075 4$:    PUSHL   HEADER                                          ; 1172
                    0000G CF        01  FB 00078        CALLS   #1, CHECKSUM                                    ;
                          14        50  E8 0007D        BLBS    R0, 6$                                          ;
                          52        02  D0 00080        MOVL    #2, STATUS                                      ; 1173
                          03        6C  91 00083        CMPB    (AP), #3                                        ;
                                    52  1F 00086        BLSSU   12$                                             ;
                          4E    0C  BC  E9 00088        BLBC    @HEADER_STATUS, 12$                             ;
                    0C    BC  0808  8F  B0 0008C        MOVW    #2056, @HEADER_STATUS                           ;
                                    46  11 00092 5$:    BRB     12$                                             ;
                          50    04  AC  7D 00094 6$:    MOVQ    HEADER, R0                                      ; 1178
                          61    08  A0  B1 00098        CMPW    8(R0), (R1)                                     ;
                                    07  12 0009C        BNEQ    7$                                              ;
```

```
                     05    A1      0D    A0  91 0009E          CMPB    13(R0), 5(R1)                           ; 1179
                                         14  13 000A3          BEQL    8$                                      ;
                           52            02  D0 000A5 7$:      MOVL    #2, STATUS                              ; 1180
                           03            6C  91 000A8          CMPB    (AP), #3                                ;
                                         2D  1F 000AB          BLSSU   12$                                     ;
                           29      0C    BC  E9 000AD          BLBC    @HEADER_STATUS, 12$                     ;
                     0C    BC    08B0    8F  B0 000B1          MOVW    #2224, @HEADER_STATUS                   ;
                                         21  11 000B7          BRB     12$                                     ;
                           51      04    AC  D0 000B9 8$:      MOVL    HEADER, R1                              ; 1182
                           50      08    AC  D0 000BD          MOVL    FILE_ID, R0                             ;
                     02    A0      0A    A1  B1 000C1          CMPW    10(R1), 2(R0)                           ;
                                         16  13 000C6          BEQL    13$                                     ;
                           52            04  D0 000C8          MOVL    #4, STATUS                              ; 1183
                           03            6C  91 000CB 9$:      CMPB    (AP), #3                                ;
                                         0A  1F 000CE 10$:     BLSSU   12$                                     ;
                           06      0C    BC  E9 000D0 11$:     BLBC    @HEADER_STATUS, 12$                     ;
                     0C    BC    0910    8F  B0 000D4          MOVW    #2320, @HEADER_STATUS                   ;
                           50            52  D0 000DA 12$:     MOVL    STATUS, R0                              ;
                                             04 000DD          RET                                             ;
                           50            01  D0 000DE 13$:     MOVL    #1, R0                                  ; 1188
                                             04 000E1          RET                                             ; 1190

; Routine Size:  226 bytes,    Routine Base:  $CODE$ + 0000

;  202          1191  1
;  203          1192  1 END
;  204          1193  0 ELUDOM



;                                  PSECT SUMMARY

;      Name                    Bytes                           Attributes

;  $CODE$                        226  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)


;                               Library Statistics

;                                          --------  Symbols  --------      Pages       Processing
;      File                                Total     Loaded   Percent       Mapped      Time

;  _$255$DUA28:[SYSLIB]LIB.L32;1           18619         36         0         1000        00:02.0


;                                  COMMAND QUALIFIERS
```

```
;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:CHKHD2/OBJ=OBJ$:CHKHD2 MSRC$:CHKHD2/UPDATE=(ENH$:CHKHD2)

; Size:          226 code + 0 data bytes
; Run Time:          00:11.6
; Elapsed Time:      00:22.4
; Lines/CPU Min:      6197
; Lexemes/CPU-Min:  26197
; Memory Used:   168 pages
; Compilation Complete
```

CHKDMO
LIS

CPYNAM
LIS

CLENUP
LIS

CHKPRO
LIS

CHARGEQ
LIS

COMMON
LIS

CREATE
LIS

BADSCN
LIS

CHKHD2
LIS

CHKSUM
LIS

ALLOCB
LIS