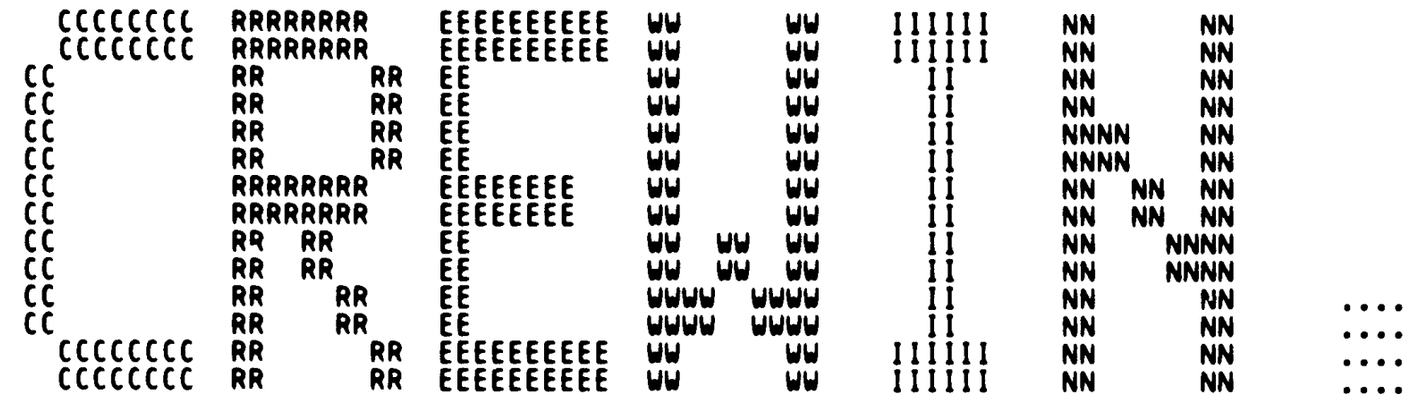
```
FFFFFFFFFFFFFFF         111              111         XXX         XXX
FFFFFFFFFFFFFFF         111              111         XXX         XXX
FFFFFFFFFFFFFFF         111              111         XXX         XXX
FFF                  111111           111111         XXX         XXX
FFF                  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
FFFFFFFFFFFF            111              111               XXX
FFFFFFFFFFFF            111              111               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
```

```
_$25
Symb
----
IOC1
IO_C
IO_C
IO_D
IO_F
IO_S
KILL

KILL
KILL
LB_E
LB_D
LB_F
LB_H
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**CREWIN



```
;     1          0001  0 MODULE CREWIN (
;     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 creates and initializes a file window.
;    38          0038  1 ! ENVIRONMENT:
;    39          0039  1 !
;    40          0040  1 !       STARLET operating system, including privileged system services
;    41          0041  1 !       and internal exec routines. This routine must be called
;    42          0042  1 !       in kernel mode.
;    43          0043  1 !
;    44          0044  1 !--
;    45          0045  1 !
;    46          0046  1 !
;    47          0047  1 ! AUTHOR:  Andrew C. Goldstein,  CREATION DATE:  14-Dec-1976  17:10
;    48          0048  1 !
;    49          0049  1 ! MODIFIED BY:
;    50          0050  1 !
;    51          0051  1 !       V03-003 CDS0002         Christian D. Saether    18-Apr-1984
;    52          0052  1 !               Mask off unused bits when copying access control flags
;    53          0053  1 !               to wcb$w_acon, and copy FIB$V_NOLOCK state to WCB$V_NOACCLOCK.
;    54          0054  1 !
;    55          0055  1 !       V03-002 CDS0001         Christian D. Saether    29-Dec-1983
;    56          0056  1 !               Use L_NORM linkage and BIND_COMMON macro.
;    57          0057  1 !
```

```
;    58          0058  1 !       V03-001 LMP0016         L. Mark Pilant,         25-Mar-1982  13:20
;    59          0059  1 !               Remove diddling of the COMPLETE bit in the window segments.
;    60          0060  1 !
;    61          0061  1 !       V02-002 LMP0003         L. Mark Pilant,         17-Nov-1981  14:35
;    62          0062  1 !               Add support for segmented windows.
;    63          0063  1 !
;    64          0064  1 !       V02-001 ACG0167         Andrew C. Goldstein,    16-Apr-1980  19:25
;    65          0065  1 !               Previous revision history moved to F11B.REV
;    66          0066  1 !**
;    67          0067  1
;    68          0068  1
;    69          0069  1 LIBRARY 'SYS$LIBRARY:LIB.L32';
;    70          0070  1 REQUIRE 'SRC$:FCPDEF.B32';
```

```
;    72          1061  1 GLOBAL ROUTINE CREATE_WINDOW (ACCTL, SIZE, HEADER, PID, FCB) : L_NORM =
;    73          1062  1
;    74          1063  1 !++
;    75          1064  1 !
;    76          1065  1 ! FUNCTIONAL DESCRIPTION:
;    77          1066  1 !
;    78          1067  1 !       This routine creates a file access window.
;    79          1068  1 !
;    80          1069  1 ! CALLING SEQUENCE:
;    81          1070  1 !       CREATE_WINDOW (ARG1, ARG2, ARG3, ARG4, ARG5)
;    82          1071  1 !
;    83          1072  1 ! INPUT PARAMETERS:
;    84          1073  1 !       ARG1: access control word (from FIB, usually)
;    85          1074  1 !       ARG2: size of window in # of pointers
;    86          1075  1 !       ARG3: address of file header
;    87          1076  1 !       ARG4: PID of accessor
;    88          1077  1 !       ARG5: address of file FCB
;    89          1078  1 !
;    90          1079  1 ! IMPLICIT INPUTS:
;    91          1080  1 !       CURRENT_VCB: address of VCB of volume in process
;    92          1081  1 !       CURRENT_UCB: address of UCB of disk in process
;    93          1082  1 !
;    94          1083  1 ! OUTPUT PARAMETERS:
;    95          1084  1 !       NONE
;    96          1085  1 !
;    97          1086  1 ! IMPLICIT OUTPUTS:
;    98          1087  1 !       NONE
;    99          1088  1 !
;   100          1089  1 ! ROUTINE VALUE:
;   101          1090  1 !       address of window
;   102          1091  1 !
;   103          1092  1 ! SIDE EFFECTS:
;   104          1093  1 !       window block created
;   105          1094  1 !
;   106          1095  1 !--
;   107          1096  1
;   108          1097  2 BEGIN
;   109          1098  2
;   110          1099  2 MAP
;   111          1100  2         ACCTL           : BBLOCK,       ! access control flags
;   112          1101  2         HEADER          : REF BBLOCK,   ! file header arg
;   113          1102  2         FCB             : REF BBLOCK;   ! FCB arg
;   114          1103  2
;   115          1104  2 LOCAL
;   116          1105  2         WINDOW_SIZE,                    ! actual size of window
;   117          1106  2         WINDOW          : REF BBLOCK,   ! window created
;   118          1107  2         PRIMARY_WINDOW  : REF BBLOCK;   ! address of the primary window
;   119          1108  2
;   120          1109  2 BIND_COMMON;
;   121          1110  2
;   122          1111  2 EXTERNAL ROUTINE
;   123          1112  2         ALLOCATE        : L_NORM,       ! allocate dynamic memory
;   124          1113  2         TURN_WINDOW     : L_NORM;       ! window turner routine
;   125          1114  2
;   126          1115  2 ! Compute the size of the window. If fixed, allocate it and turn it to
;   127          1116  2 ! map VBN 1. If a maximal window is requested (indicated by a size of -1),
;   128          1117  2 ! the window turner will allocate the window.
```

```
;  129          1118  2 !
;  130          1119  2
;  131          1120  2 WINDOW_SIZE = .SIZE;
;  132          1121  2 IF .WINDOW_SIZE EQL 0
;  133          1122  2 THEN WINDOW_SIZE = .CURRENT_VCB[VCB$B_WINDOW];
;  134          1123  2
;  135          1124  2 IF .WINDOW_SIZE NEQ -1
;  136          1125  2 THEN
;  137          1126  3     BEGIN
;  138          1127  3     IF .WINDOW_SIZE GTRU MAX_WINDOW
;  139          1128  3     THEN WINDOW_SIZE = MAX_WINDOW;
;  140          1129  3     IF .WINDOW_SIZE LSSU MIN_WINDOW
;  141          1130  3     THEN WINDOW_SIZE = MIN_WINDOW;
;  142          1131  3     WINDOW = ALLOCATE (.WINDOW_SIZE * 6 + WCB$C_LENGTH, WCB_TYPE);
;  143          1132  3     IF .WINDOW NEQ 0
;  144          1133  3     THEN TURN_WINDOW (.WINDOW, .HEADER, 1, 1);
;  145          1134  3     END
;  146          1135  2 ELSE
;  147          1136  2     WINDOW = TURN_WINDOW (0, .HEADER, 1, 1);
;  148          1137  2
;  149          1138  2 ! Init cells within the window
;  150          1139  2 !
;  151          1140  2
;  152          1141  2 PRIMARY_WINDOW = .WINDOW;
;  153          1142  2 UNTIL .WINDOW EQL 0
;  154          1143  2 DO
;  155          1144  3     BEGIN
;  156          1145  3     WINDOW[WCB$L_PID]    = .PID;                    ! accessor PID
;  157          1146  3     WINDOW[WCB$L_ORGUCB]        = .CURRENT_UCB;     ! original device UCB
;  158          1147  3     WINDOW[WCB$W_ACON]   = .ACCTL [0,0,16,0];       ! access control bits
;  159          1148  3     WINDOW[WCB$W_ACON] = .WINDOW[WCB$W_ACON]        ! mask off unused flags
;  160          1149  4         AND (FIB$M_NOWRITE+FIB$M_DLOCK+FIB$M_SPOOL+FIB$M_WRITECK+FIB$M_SEQONLY
;  161          1150  3                 +FIB$M_WRITE+FIB$M_READCK+FIB$M_NOREAD+FIB$M_NOTRUNC);
;  162          1151  4     IF (.ACCTL [FIB$V_NOLOCK]                       ! This assumes FIB$L_ACCTL
;  163          1152  4         AND .CLEANUP_FLAGS [CLF_SYSPRV])
;  164          1153  3     THEN WINDOW [WCB$V_NOACCLOCK] = 1;              ! is first longword in FIB.
;  165          1154  3     WINDOW[WCB$L_FCB]    = .FCB;                    ! FCB address
;  166          1155  3     WINDOW[WCB$V_READ]   = 1;                       ! read access always allowed
;  167          1156  3     WINDOW[WCB$V_WRITE] = .WINDOW[WCB$V_WRITEAC];   ! write access sometimes
;  168          1157  3     IF .CURRENT_RVN NEQ 0 THEN WINDOW[WCB$L_RVT] = .CURRENT_VCB[VCB$L_RVT];
;  169          1158  3     IF .HEADER[FH2$V_READCHECK] THEN WINDOW[WCB$V_READCK] = 1;
;  170          1159  3     IF .HEADER[FH2$V_WRITCHECK] THEN WINDOW[WCB$V_WRITECK] = 1;
;  171          1160  3     WINDOW = .WINDOW[WCB$L_LINK];                   ! set address of next segment
;  172          1161  2     END;
;  173          1162  2
;  174          1163  2 RETURN .PRIMARY_WINDOW;
;  175          1164  2
;  176          1165  1 END;                                    ! end of routine CREATE_WINDOW


                                                                .TITLE  CREWIN
                                                                .IDENT  \V04-000\

                                                                .EXTRN  ALLOCATE, TURN_WINDOW

                                                                .PSECT  $CODE$,NOWRT,2
```

```
                                 0004 00000          .ENTRY  CREATE_WINDOW, Save R2                          ; 1061
                          50     08   AC  D0 00002          MOVL    SIZE, WINDOW_SIZE                       ; 1120
                                      08  12 00006          BNEQ    1$                                      ; 1121
                          51     98   AA  D0 00008          MOVL    -104(BASE), R1                          ; 1122
                          50     48   A1  9A 0000C          MOVZBL  72(R1), WINDOW_SIZE                     ;
                FFFFFFFF  8F           50  D1 00010 1$:     CMPL    WINDOW_SIZE, #-1                        ; 1124
                                       36  13 00017          BEQL    4$                                      ;
                00000050  8F           50  D1 00019          CMPL    WINDOW_SIZE, #80                        ; 1127
                                       04  1B 00020          BLEQU   2$                                      ;
                          50     50   8F  9A 00022          MOVZBL  #80, WINDOW_SIZE                        ; 1128
                                       50  D5 00026 2$:      TSTL    WINDOW_SIZE                             ; 1129
                                       03  12 00028          BNEQ    3$                                      ;
                          50           01  D0 0002A          MOVL    #1, WINDOW_SIZE                         ; 1130
                                       01  DD 0002D 3$:      PUSHL   #1                                      ; 1131
                          50           06  C4 0002F          MULL2   #6, R0                                  ;
                                 30    A0  9F 00032          PUSHAB  48(R0)                                  ;
                     0000G CF           02  FB 00035          CALLS   #2, ALLOCATE                            ;
                          52           50  D0 0003A          MOVL    R0, WINDOW                              ;
                                       21  13 0003D          BEQL    5$                                      ; 1132
                                       01  DD 0003F          PUSHL   #1                                      ; 1133
                                       01  DD 00041          PUSHL   #1                                      ;
                                 0C    AC  DD 00043          PUSHL   HEADER                                  ;
                                       52  DD 00046          PUSHL   WINDOW                                  ;
                     0000G CF           04  FB 00048          CALLS   #4, TURN_WINDOW                         ;
                                       11  11 0004D          BRB     5$                                      ; 1124
                                       01  DD 00049 4$:      PUSHL   #1                                      ; 1136
                                       01  DD 0005> 1         PUSHL   #1                                      ;
                                 0C    AC  DD 00053          PUSHL   HEADER                                  ;
                                       7E  D4 00056          CLRL    -(SP)                                   ;
                           0000G CF          04  FB 00058          CALLS   #4, TURN_WINDOW                     ;
                                 52          50  D0 0005D          MOVL    R0, WINDOW                          ;
                                 51          52  D0 00060 5$:      MOVL    WINDOW, PRIMARY_WINDOW              ; 1141
                                             60  13 00063 6$:      BEQL    11$                                 ; 1142
                              0C    A2   10   AC  D0 00065          MOVL    PID, 12(WINDOW)                     ; 1145
                              10    A2   94   AA  D0 0006A          MOVL    -108(BASE), 16(WINDOW)              ; 1146
                              14    A2   04   AC  B0 0006F          MOVW    ACCTL, 20(WINDOW)                   ; 1147
                              14    A2   F08C 8F  AA 00074          BICW2   #61580, 20(WINDOW)                  ; 1149
                        08    06    AC         04  E1 0007A          BBC     #4, ACCTL+2, 7$                     ; 1151
                              04    01   AA    E9 0007F          BLBC    1(BASE), 7$                         ; 1152
                              14    A2         04  88 00083          BISB2   #4, 20(WINDOW)                      ; 1153
                              18    A2   14   AC  D0 00087 7$:      MOVL    FCB, 24(WINDOW)                     ; 1154
                              0B    A2         01  88 0008C          BISB2   #1, 11(WINDOW)                      ; 1155
                0B    A2        01           01  15  A2  F0 00090          INSV    21(WINDOW), #1, #1, 11(WINDOW)  ; 1156
                                             A0  AA  D5 00097          TSTL    -96(BASE)                       ; 1157
                                             09  13 0009A          BEQL    8$                                  ;
                                       50    98  AA  D0 0009C          MOVL    -104(BASE), R0                  ;
                                       1C    A2  20  A0  D0 000A0          MOVL    32(R0), 28(WINDOW)          ;
                                       50    0C  AC  D0 000A5 8$:      MOVL    HEADER, R0                      ; 1158
                                 04    34    A0        03  E1 000A9          BBC     #3, 52(R0), 9$              ;
                                       15    A2        02  88 000AE          BISB2   #2, 21(WINDOW)              ;
                                       50    0C  AC  D0 000B2 9$:      MOVL    HEADER, R0                      ; 1159
                                 04    34    A0        04  E1 000B6          BBC     #4, 52(R0), 10$             ;
                                       14    A2        20  88 000BB          BISB2   #32, 20(WINDOW)             ;
                                       52    20  A2  D0 000BF 10$:      MOVL    32(WINDOW), WINDOW              ; 1160
                                             9E  11 000C3          BRB     6$                                  ; 1142
                                       50          51  D0 000C5 11$:      MOVL    PRIMARY_WINDOW, R0              ; 1163
                                                   04 000C8          RET                                     ; 1165
```

```
; Routine Size:  201 bytes,    Routine Base:  $CODE$ + 0000

;  177          1166  1
;  178          1167  1 END
;  179          1168  0 ELUDOM


;                                  PSECT SUMMARY
;
;      Name                      Bytes                          Attributes
;  $CODE$                          201  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        44         0        1000        00:01.9


;                                        COMMAND QUALIFIERS
;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:CREWIN/OBJ=OBJ$:CREWIN MSRC$:CREWIN/UPDATE=(ENH$:CREWIN)

; Size:          201 code + 0 data bytes
; Run Time:          00:17.3
; Elapsed Time:      00:37.2
; Lines/CPU Min:      4048
; Lexemes/CPU-Min: 50253
; Memory Used:  221 pages
; Compilation Complete
```

DEACCS
LIS

DELETE
LIS

DIRSCN
LIS

DISPAT
LIS

CREHDR
LIS

DIRACC
LIS

DELBAD
LIS

CREFCB
LIS

CREWIN
LIS

DISPATCH
LIS

ENTER
LIS

DELFIL
LIS