C 6

```
MMM         MMM     000000000     UUU         UUU  NNN         NNN  TTTTTTTTTTTTTTT
MMM         MMM     000000000     UUU         UUU  NNN         NNN  TTTTTTTTTTTTTTT
MMM         MMM     000000000     UUU         UUU  NNN         NNN  TTTTTTTTTTTTTTT
MMMMMM   MMMMMM  000         000  UUU         UUU  NNN         NNN        TTT
MMMMMM   MMMMMM  000         000  UUU         UUU  NNN         NNN        TTT
MMMMMM   MMMMMM  000         000  UUU         UUU  NNN         NNN        TTT
MMM   MMM   MMM  000         000  UUU         UUU  NNNNNN      NNN        TTT
MMM   MMM   MMM  000         000  UUU         UUU  NNNNNN      NNN        TTT
MMM   MMM   MMM  000         000  UUU         UUU  NNNNNN      NNN        TTT
MMM         MMM  000         000  UUU         UUU  NNN   NNN   NNN        TTT
MMM         MMM  000         000  UUU         UUU  NNN   NNN   NNN        TTT
MMM         MMM  000         000  UUU         UUU  NNN   NNN   NNN        TTT
MMM         MMM  000         000  UUU         UUU  NNN      NNNNNN        TTT
MMM         MMM  000         000  UUU         UUU  NNN      NNNNNN        TTT
MMM         MMM  000         000  UUU         UUU  NNN      NNNNNN        TTT
MMM         MMM  000         000  UUU         UUU  NNN         NNN        TTT
MMM         MMM  000         000  UUU         UUU  NNN         NNN        TTT
MMM         MMM  000         000  UUU         UUU  NNN         NNN        TTT
MMM         MMM     000000000     UUUUUUUUUUUUUUU  NNN         NNN        TTT
MMM         MMM     000000000     UUUUUUUUUUUUUUU  NNN         NNN        TTT
MMM         MMM     000000000     UUUUUUUUUUUUUUU  NNN         NNN        TTT
```

```
_$

Sy
--
LI
LI
LI
LI
LI
LI
LI
LI
LI
LI
LI
LI

LI
LI


LI
LI
LI
LI
LI
LI
LI
LI
LN
LN
LO
LO

LO
LO
LO
LO
MA

MO
MO
MO
MO
MO
MO
MO


MO
MO
```

FILEID**MWTUR2

```
MM      MM  WW      WW  TTTTTTTTTT  UU      UU  RRRRRRRR      222222
MM      MM  WW      WW  TTTTTTTTTT  UU      UU  RRRRRRRR      222222
MMMM  MMMM  WW      WW      TT      UU      UU  RR      RR  22      22
MMMM  MMMM  WW      WW      TT      UU      UU  RR      RR  22      22
MM  MM  MM  WW      WW      TT      UU      UU  RR      RR          22
MM  MM  MM  WW      WW      TT      UU      UU  RR      RR          22
MM      MM  WW      WW      TT      UU      UU  RRRRRRRR          22
MM      MM  WW      WW      TT      UU      UU  RRRRRRRR          22
MM      MM  WW  WW  WW      TT      UU      UU  RR  RR          22
MM      MM  WW  WW  WW      TT      UU      UU  RR  RR          22
MM      MM  WWWW  WWWW      TT      UU      UU  RR    RR      22            ....
MM      MM  WWWW  WWWW      TT      UU      UU  RR    RR      22            ....
MM      MM  WW      WW      TT      UUUUUUUUUU  RR      RR  2222222222      ....
MM      MM  WW      WW      TT      UUUUUUUUUU  RR      RR  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
```

RDH
V04

```
0 MODULE MWTUR2 (
0                 LANGUAGE (BLISS32),
0                 IDENT = 'V04-000'
0                 ) =
1 BEGIN
1
1 !
1 !****************************************************************************
1 !*                                                                          *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
1 !*  ALL RIGHTS RESERVED.                                                    *
1 !*                                                                          *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
1 !*  TRANSFERRED.                                                            *
1 !*                                                                          *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
1 !*  CORPORATION.                                                            *
1 !*                                                                          *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
1 !*                                                                          *
1 !*                                                                          *
1 !****************************************************************************
1
1 !++
1 !
1 ! FACILITY:  MOUNT Utility Structure Level 2
1 !
1 ! ABSTRACT:
1 !
1 !       This module generates a window mapping the desired VBN from
1 !       the supplied file header. This module is a direct crib from the
1 !       FCP module WITURN. When we get conditional compilation with different
1 !       parameter files worked out, they should really be the same source.
1 !
1 ! ENVIRONMENT:
1 !
1 !       STARLET operating system, including privileged system services
1 !       and internal exec routines.
1 !
1 !--
1 !
1 !
1 ! AUTHOR:  Andrew C. Goldstein, CREATION DATE:  7-Dec-1976  14:38
1 !
1 ! MODIFIED BY:
1 !
1 !       V03-001 HH0041          Hai Huang               24-Jul-1984
1 !               Remove REQUIRE 'LIBD$:[VMSLIB.OBJ]MOUNTMSG.B32'.
1 !
1 !       V02-001 ACG0167         Andrew C. Goldstein,    18-Apr-1980  13:39
```

```
0058  1 !                     Previous revision history moved to MOUNT.REV
0059  1 !**
0060  1
0061  1
0062  1 LIBRARY 'SYS$LIBRARY:LIB.L32';
0063  1 REQUIRE 'SRC$:MOUDEF.B32';
```

```
1 GLOBAL ROUTINE TURN_WINDOW2 (WINDOW, HEADER, DESIRED_VBN, START_VBN, RVN) =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine scans the map area of the supplied file header
1 !       and builds retrieval pointers in the window until
1 !       (1) the entire header has been scanned, or
1 !       (2) the first retrieval pointer in the window maps the desired VBN
1 !
1 ! CALLING SEQUENCE:
1 !       TURN_WINDOW (ARG1, ARG2, ARG3, ARG4, ARG5)
1 !
1 ! INPUT PARAMETERS:
1 !       ARG1: address of window block
1 !       ARG2: address of file header
1 !       ARG3: desired VBN
1 !       ARG4: starting VBN of file header
1 !       ARG5: RVN of file header
1 !
1 ! IMPLICIT INPUTS:
1 !       NONE
1 !
1 ! OUTPUT PARAMETERS:
1 !       updated window
1 !
1 ! IMPLICIT OUTPUTS:
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 !       1
1 !
1 ! SIDE EFFECTS:
1 !       NONE
1 !
1 !--
1
2 BEGIN
2
2 MAP
2       WINDOW          : REF BBLOCK,   ! pointer to window
2       HEADER          : REF BBLOCK;   ! pointer to file header
2
2 LINKAGE
2       L_MAP_POINTER   = JSB :
2                         GLOBAL (COUNT = 6, LBN = 7, MAP_POINTER = 8);
2
2 GLOBAL REGISTER
2       COUNT           = 6,            ! retrieval pointer count
2       LBN             = 7,            ! retrieval pointer start LBN
2       MAP_POINTER     = 8 : REF BBLOCK; ! pointer to scan header map area
2
2 LABEL
2       MAP_BUILD;                      ! loop to build window map
2
2 LOCAL
```

```
2           VBN,                                ! VBN in scanning window
2           COUNTER,                            ! loop counter
2           W_POINTER       : REF BBLOCK;       ! pointer to scan window
2
2 MACRO
2           WINDOW_MAP      = (.WINDOW+WCB$C_MAP)%; !start of window map area
2
2 EXTERNAL ROUTINE
2           GET_MAP_POINTER : L_MAP_POINTER; ! get value of next header map pointer
2
2 ! Scan the window looking for the starting VBN of the header. If it is
2 ! contained within the window, truncate the window so that it maps exactly
2 ! up to the start of the header. If the starting VBN is not contained in the
2 ! window, the entire window must be discarded. However, if the desired VBN
2 ! precedes the header start VBN, we do nothing since the window is already
2 ! best effort.
2 !
2
2 W_POINTER = WINDOW_MAP;                               ! point to first retrieval pointer
2 VBN = .WINDOW[WCB$L_STVBN];                           ! get starting VBN of window
2
2 IF .START_VBN LEQU .VBN
2 OR
3     BEGIN
3     INCR J FROM 1 TO .WINDOW[WCB$W_NMAP]
3     DO
4         BEGIN
4         VBN = .VBN + .W_POINTER[WCB$W_COUNT];     ! VBN at end of this pointer
4         W_POINTER = .W_POINTER + 6;
4         IF.START_VBN EQL .VBN
4         THEN
5             BEGIN
5             WINDOW[WCB$W_NMAP] = .J;              ! truncate the window
5             EXITLOOP 0;
4             END;
4         END
3     END
3
2 THEN                                                  ! header VBN is not in window
3     BEGIN
3     IF .DESIRED_VBN LSSU .START_VBN
3     AND .START_VBN GTRU 1
3     THEN
3         RETURN 1                                  ! leave it alone
3     ELSE
4         BEGIN
4         WINDOW[WCB$W_NMAP] = 0;                   ! flush the window
4         WINDOW[WCB$L_STVBN] = .START_VBN;
4         W_POINTER = WINDOW_MAP;                   ! point to first pointer
3         END;
2     END;
2
2 ! The window is now suitably initialized. Set up necessary pointers.
2 ! Now scan the map area, extracting retrieval pointers.
2 !
2
2 MAP_POINTER = .HEADER + .HEADER[FH2$B_MPOFFSET]*2;      ! point to map area
```

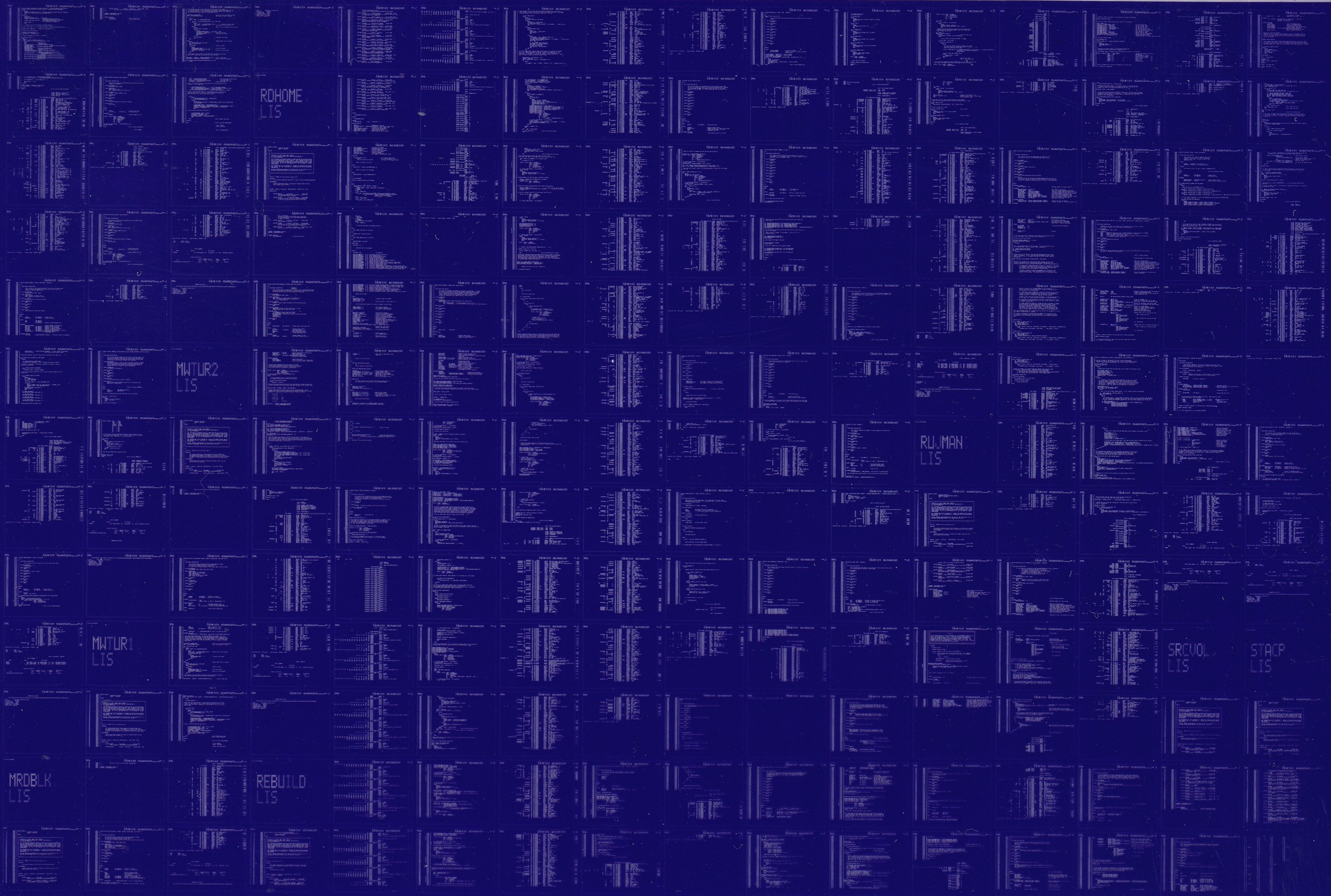
```
2
3 MAP_BUILD: BEGIN
3 UNTIL .MAP_POINTER GEQA .HEADER + (.HEADER[FH2$B_MPOFFSET] + .HEADER[FH2$B_MAP_INUSE]) * 2
3 DO
4     BEGIN
4
4     GET_MAP_POINTER ();
4
4 ! Build new retrieval pointers, using as many as needed to run out the
4 ! count. If the window is full, shuffle the entries up by one. If this
4 ! would cause the pointer mapping the  desired VBN to fall off the top,
4 ! we are done.
4 !
4
4     IF .COUNT NEQ 0
4     THEN
4         WHILE 1 DO
5         BEGIN
5         IF (.WINDOW[WCB$W_NMAP]+1)*6 + WCB$C_LENGTH
5             GTRU .WINDOW[WCB$W_SIZE]
5         THEN
6             BEGIN
6             IF .WINDOW[WCB$L_STVBN] + .WINDOW[WCB$W_P1_COUNT] GTRU .DESIRED_VBN
6             THEN LEAVE MAP_BUILD;
6
6             WINDOW[WCB$W_NMAP] = .WINDOW[WCB$W_NMAP] - 1;
6             WINDOW[WCB$L_STVBN] = .WINDOW[WCB$L_STVBN] + .WINDOW[WCB$W_P1_COUNT];
6             CH$MOVE (.WINDOW[WCB$W_NMAP]*6, WINDOW_MAP+6, WINDOW_MAP);
6             W_POINTER = .W_POINTER - 6;
5             END;
5
5 ! Finally build the pointer and count it.
5 !
5
5         W_POINTER[WCB$W_COUNT] = MINU (.COUNT, 65535);
5         W_POINTER[WCB$L_LBN] = .LBN;
5         (W_POINTER[WCB$L_LBN])<24,8> = .RVN;
5         W_POINTER = .W_POINTER + 6;
5         WINDOW[WCB$W_NMAP] = .WINDOW[WCB$W_NMAP] + 1;
5         LBN = .LBN + 65535;
5         COUNT = .COUNT - MINU (.COUNT, 65535);
5         IF .COUNT EQL 0 THEN EXITLOOP;
4         END;
4
3     END;                                          ! end of header scan loop
2 END;                                              ! end of block MAP_BUILD
2
2 RETURN 1;
2
1 END;                                              ! end of routine TURN_WINDOW2

                                                    .TITLE  MWTUR2
                                                    .IDENT  \V04-000\

                                                    .EXTRN  GET_MAP_POINTER
```

```
                                                              M  3
MWTUR2                                                  16-Sep-1984 01:27:16    VAX-11 Bliss-32 V4.0-742                 Page   6
V04-000                                                 14-Sep-1984 12:45:33    DISK$VMSMASTER:[MOUNT.SRC]MWTUR2.B32;1    (2)

                                                                                .PSECT  $CODE$,NOWRT,2

                                                        0FFC 00000              .ENTRY  TURN_WINDOW2, Save R2,R3,R4,R5,R6,R7,R8,R9,-; 0595
                                                                                        R10,R11
                                    5E              04  C2 00002                SUBL2   #4, SP                                      ;
                                    59         04   AC  D0 00005                MOVL    WINDOW, R9                                  ; 0670
                                               30   A9  9F 00009                PUSHAB  48(R9)                                      ;
                                                    6E  DD 0000C                PUSHL   (SP)                                        ;
                                    52         2C   A9  D0 0000E                MOVL    44(R9), VBN                                 ; 0671
                                    50         10   AC  D0 00012                MOVL    START_VBN, R0                               ; 0673
                                    52              50  D1 00016                CMPL    R0, VBN                                     ;
                                                    21  1B 00019                BLEQU   3$                                          ;
                                    53         16   A9  3C 0001B                MOVZWL  22(R9), R3                                  ; 0676
                                                    51  D4 0001F                CLRL    J                                           ;
                                                    15  11 00021                BRB     2$                                          ;
                                    54         00   BE  3C 00023 1$:            MOVZWL  @W_POINTER, R4                              ; 0679
                                    52              54  C0 00027                ADDL2   R4, VBN                                     ;
                                    6E              06  C0 0002A                ADDL2   #6, W_POINTER                               ; 0680
                                    52              50  D1 0002D                CMPL    R0, VBN                                     ; 0681
                                                    06  12 00030                BNEQ    2$                                          ;
                              16    A9              51  B0 00032                MOVW    J, 22(R9)                                   ; 0684
                                                    1A  11 00036                BRB     5$                                          ; 0685
                      E7            51              53  F3 00038 2$:            AOBLEQ  R3, J, 1$                                   ; 0676
                                    50         0C   AC  D1 0003C 3$:            CMPL    DESIRED_VBN, R0                             ; 0692
                                                    05  1E 00040                BGEQU   4$                                          ;
                                    01              50  D1 00042                CMPL    R0, #1                                      ; 0693
                                                    55  1A 00045                BGTRU   8$                                          ;
                                               16   A9  B4 00047 4$:            CLRW    22(R9)                                      ; 0698
                              2C    A9              50  D0 0004A                MOVL    R0, 44(R9)                                  ; 0699
                                    6E         04   AE  D0 0004E                MOVL    4(SP), W_POINTER                            ; 0700
                                    5B         08   AC  D0 00052 5$:            MOVL    HEADER, R11                                 ; 0708
                              08    AE         01   AB  9A 00056                MOVZBL  1(R11), 8(SP)                               ;
                                    50         08   AE  D0 0005B                MOVL    8(SP), R0                                   ;
                                    58            6B40  3E 0005F                MOVAW   (R11)[R0], MAP_POINTER                      ;
                                    50         3A   AB  9A 00063 6$:            MOVZBL  58(R11), R0                                 ; 0711
                                    50         08   AE  C0 00067                ADDL2   8(SP), R0                                   ;
                                    50            6B40  3E 0006B                MOVAW   (R11)[R0], R0                               ;
                                    50              58  D1 0006F                CMPL    MAP_POINTER, R0                             ;
                                                    7B  1E 00072                BGEQU   11$                                         ;
                                                 0000G  30 00074                BSBW    GET_MAP_POINTER                             ; 0715
                                                    56  D5 00077                TSTL    COUNT                                       ; 0723
                                                    E8  13 00079                BEQL    6$                                          ;
                                    5A         16   A9  9E 0007B                MOVAB   22(R9), R10                                 ; 0727
                                    50              6A  3C 0007F 7$:            MOVZWL  (R10), R0                                   ;
                                    50              06  C4 00082                MULL2   #6, R0                                      ;
                                    50              36  C0 00085                ADDL2   #54, R0                                     ;
          50        08        A9    10              00  ED 00088                CMPZV   #0, #16, 8(R9), R0                          ; 0728
                                                    27  1E 0008E                BGEQU   9$                                          ;
                                    50         04   BE  3C 00090                MOVZWL  @4(SP), R0                                  ; 0731
                                    50         2C   A9  C0 00094                ADDL2   44(R9), R0                                  ;
                              0C    AC              50  D1 00098                CMPL    R0, DESIRED_VBN                             ;
                                                    51  1A 0009C 8$:            BGTRU   11$                                         ;
                                                    6A  B7 0009E                DECW    (R10)                                       ; 0734
                                    50         04   BE  3C 000A0                MOVZWL  @4(SP), R0                                  ; 0735
                              2C    A9              50  C0 000A4                ADDL2   R0, 44(R9)                                  ;
                                    50              6A  3C 000A8                MOVZWL  (R10), R0                                   ; 0736
                                    50              06  C4 000AB                MULL2   #6, R0                                      ;
```

```
                    04    BE       36    A9             50  28 000AE             MOVC3    R0, 54(R9), @4(SP)                      ;
                                         6E             06  C2 000B4             SUBL2    #6, W_POINTER                           ; 0737
                                         50             56  D0 000B7 9$:         MOVL     COUNT, R0                               ; 0743
                          0000FFFF       8F             50  D1 000BA             CMPL     R0, #65535                              ;
                                                        05  1B 000C1             BLEQU    10$                                     ;
                                         50      FFFF   8F  3C 000C3             MOVZWL   #65535, R0                              ;
                                 00      BE             50  B0 000C8 10$:        MOVW     R0, @W_POINTER                          ;
                          51             6E             02  C1 000CC             ADDL3    #2, W_POINTER, R1                       ; 0744
                                         61             87  7E 000D0             MOVAQ    (LBN)+, (R1)                            ;
                          51             6E             05  C1 000D3             ADDL3    #5, W_POINTER, R1                       ; 0745
                                         61        14   AC  90 000D7             MOVB     RVN, (R1)                               ;
                                         6E             06  C0 000DB             ADDL2    #6, W_POINTER                           ; 0746
                                                        6A  B6 000DE             INCW     (R10)                                   ; 0747
                                         57 0000FFF7    E7  9E 000E0             MOVAB    65527(R7), LBN                          ; 0748
                                         56             50  C2 000E7             SUBL2    R0, COUNT                               ; 0749
                                                        93  12 000EA             BNEQ     7$                                      ; 0750
                                                      FF74  31 000EC             BRW      6$                                      ;
                                         50             01  D0 000EF 11$:        MOVL     #1, R0                                  ; 0756
                                                            04 000F2             RET                                              ; 0758

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


;  229          0759  1
;  230          0760  1 END
;  231          0761  0 ELUDOM




;                                        PSECT SUMMARY
;
;       Name                        Bytes                             Attributes
;
;    $CODE$                               243  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        18         0       1000           00:02.0




;                                        COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:MWTUR2/OBJ=OBJ$:MWTUR2 MSRC$:MWTUR2/UPDATE=(ENH$:MWTUR2)
; Size:         243 code + 0 data bytes
; Run Time:        00:14.4
; Elapsed Time:    00:30.0
; Lines/CPU Min.    3179
; Lexemes/CPU-Min: 29594
; Memory Used:  144 pages
; Compilation Complete
```

0246 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY


RDHOME
LIS

MWTUR2
LIS

RUJMAN
LIS

MWTUR1
LIS

SRCVOL
LIS

STACP
LIS

MRDBLK
LIS

REBUILD
LIS