


```

MM      MM  WW      WW  TTTTTTTTTT  UU      UU  RRRRRRRR      11
MM      MM  WW      WW  TTTTTTTTTT  UU      UU  RRRRRRRR      11
MMMM    MMMM WW      WW  TT          UU      UU  RR      RR  1111
MMMM    MMMM WW      WW  TT          UU      UU  RR      RR  1111
MM  MM  MM  WW      WW  TT          UU      UU  RR      RR  11
MM  MM  MM  WW      WW  TT          UU      UU  RR      RR  11
MM  MM  MM  WW      WW  TT          UU      UU  RRRRRRRR      11
MM  MM  MM  WW      WW  TT          UU      UU  RRRRRRRR      11
MM  MM  MM  WW  WW  WW  TT          UU      UU  RR  RR      11
MM  MM  MM  WW  WW  WW  TT          UU      UU  RR  RR      11
MM  MM  MM  WWW  WWW  TT          UU      UU  RR      RR  11
MM  MM  MM  WWW  WWW  TT          UU      UU  RR      RR  11
MM  MM  MM  WW      WW  TT          UU      UU  RR      RR  111111
MM  MM  MM  WW      WW  TT          UU      UU  RR      RR  111111

```

```

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
LLLLLLLLLLLL IIIIII  SSSSSSSS
LLLLLLLLLLLL IIIIII  SSSSSSSS

```

.....

```

1 0001 0 MODULE MWTUR1 (
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 *  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: MOUNT Utility Structure Level 1
34 0034 1
35 0035 1 ABSTRACT:
36 0036 1
37 0037 1     This module generates a window mapping the desired VBN from
38 0038 1     the supplied file header. This module is a direct crib from the
39 0039 1     FCP module WITURN. When we get conditional compilation with different
40 0040 1     parameter files worked out, they should really be the same source.
41 0041 1
42 0042 1 ENVIRONMENT:
43 0043 1
44 0044 1     STARLET operating system, including privileged system services
45 0045 1     and internal exec routines.
46 0046 1
47 0047 1 --
48 0048 1
49 0049 1
50 0050 1 AUTHOR: Andrew C. Goldstein, CREATION DATE: 7-Dec-1976 14:38
51 0051 1
52 0052 1 MODIFIED BY:
53 0053 1
54 0054 1     V03-001 HH0041 Hai Huang 24-Jul-1984
55 0055 1     Remove REQUIRE 'LIBD$:[VMSLIB.OBJ]MOUNTMSG.B32'.
56 0056 1
57 0057 1     V02-000 ACG0167 Andrew C. Goldstein, 18-Apr-1980 13:39

```

MWTUR1
V04-000

M 2
16-Sep-1984 01:26:30
14-Sep-1984 12:45:33

VAX-11 Bliss-32 V4.0-742 Page 2
DISK\$VMSMASTER:[MOUNT.SRC]MWTUR1.B32;1 (1)

MWT
V04

```
: 58          0058 1  !           Previous revision history moved to MOUNT.REV
: 59          0059 1  !**
: 60          0060 1
: 61          0061 1
: 62          0062 1 LIBRARY 'SYSS$LIBRARY:LIB.L32';
: 63          0063 1 REQUIRE 'SRC$:MOUDEF.B32';
```

```

: 65 0595 1 GLOBAL ROUTINE TURN_WINDOW1 (WINDOW, HEADER, DESIRED_VBN, START_VBN) =
: 66 0596 1
: 67 0597 1 ++
: 68 0598 1
: 69 0599 1 FUNCTIONAL DESCRIPTION:
: 70 0600 1
: 71 0601 1 This routine scans the map area of the supplied file header
: 72 0602 1 and builds retrieval pointers in the window until
: 73 0603 1 (1) the entire header has been scanned, or
: 74 0604 1 (2) the first retrieval pointer in the window maps the desired VBN
: 75 0605 1
: 76 0606 1 CALLING SEQUENCE:
: 77 0607 1 TURN_WINDOW (ARG1, ARG2, ARG3, ARG4)
: 78 0608 1
: 79 0609 1 INPUT PARAMETERS:
: 80 0610 1 ARG1: address of window block
: 81 0611 1 ARG2: address of file header
: 82 0612 1 ARG3: desired VBN
: 83 0613 1 ARG4: starting VBN of file header
: 84 0614 1
: 85 0615 1 IMPLICIT INPUTS:
: 86 0616 1 NONE
: 87 0617 1
: 88 0618 1 OUTPUT PARAMETERS:
: 89 0619 1 updated window
: 90 0620 1
: 91 0621 1 IMPLICIT OUTPUTS:
: 92 0622 1 NONE
: 93 0623 1
: 94 0624 1 ROUTINE VALUE:
: 95 0625 1 1
: 96 0626 1
: 97 0627 1 SIDE EFFECTS:
: 98 0628 1 NONE
: 99 0629 1
: 100 0630 1 --
: 101 0631 1
: 102 0632 2 BEGIN
: 103 0633 2
: 104 0634 2 MAP
: 105 0635 2 WINDOW : REF BBLOCK, ! pointer to window
: 106 0636 2 HEADER : REF BBLOCK; ! pointer to file header
: 107 0637 2
: 108 0638 2 LOCAL
: 109 0639 2 VBN, ! VBN in scanning window
: 110 0640 2 COUNT, ! retrieval pointer count
: 111 0641 2 COUNTER, ! loop counter
: 112 0642 2 LBN, ! retrieval pointer start LBN
: 113 0643 2 W_POINTER : REF BBLOCK, ! pointer to scan window
: 114 0644 2 M_POINTER : REF BBLOCK, ! pointer to scan header map area
: 115 0645 2 MAP_AREA : REF BBLOCK; ! pointer to start of header map area
: 116 0646 2
: 117 0647 2 MACRO
: 118 0648 2 WINDOW_MAP = (.WINDOW+WCB$C_MAP)%; !start of window map area
: 119 0649 2
: 120 0650 2 ! Scan the window looking for the
: 121 0651 2 ! starting VBN of the header. If it is contained within the window, truncate

```

```

122 0652 2 ! the window so that it maps exactly up to the start of the header.
123 0653 2 ! If the starting VBN is not contained in the window, the entire window must be
124 0654 2 ! discarded. However, if the desired VBN precedes the header start VBN, we
125 0655 2 ! do nothing since the window is already best effort.
126 0656 2 !
127 0657 2 !
128 0658 2 W_POINTER = WINDOW_MAP; ! point to first retrieval pointer
129 0659 2 VBN = .WINDOW[WCBSL_STVBN]; ! get starting VBN of window
130 0660 2
131 0661 2 IF
132 0662 3 BEGIN
133 0663 3 INCR J FROM 1 TO .WINDOW[WCBSW_NMAP]
134 0664 3 DO
135 0665 3 IF .VBN + .W_POINTER[WCBSW_COUNT] LSSU .START_VBN
136 0666 3 THEN
137 0667 4 BEGIN
138 0668 4 VBN = .VBN + .W_POINTER[WCBSW_COUNT]; ! accumulate running VBN
139 0669 4 W_POINTER = .W_POINTER + 6; ! and move to next pointer
140 0670 4 END
141 0671 3 ELSE ! VBN is in or precedes current pointer
142 0672 4 BEGIN
143 0673 4 IF .VBN LSSU .START_VBN
144 0674 4 THEN
145 0675 5 BEGIN
146 0676 5 W_POINTER[WCBSW_COUNT] = .START_VBN - .VBN;
147 0677 5 W_POINTER = .W_POINTER + 6; ! bump the pointer
148 0678 5 WINDOW[WCBSW_NMAP] = .J; ! truncate the window
149 0679 5 EXITLOOP 0;
150 0680 5 END
151 0681 4 ELSE
152 0682 4 EXITLOOP -1; ! precedes the window
153 0683 4 END
154 0684 3 END
155 0685 3
156 0686 2 THEN ! header VBN is not in window
157 0687 3 BEGIN
158 0688 3 IF .DESIRED_VBN LSSU .START_VBN
159 0689 3 AND .START_VBN GTRU 1
160 0690 3 THEN
161 0691 3 RETURN 1 ! leave it alone
162 0692 3 ELSE
163 0693 4 BEGIN
164 0694 4 WINDOW[WCBSW_NMAP] = 0; ! flush the window
165 0695 4 WINDOW[WCBSL_STVBN] = .START_VBN;
166 0696 4 W_POINTER = WINDOW_MAP; ! point to first pointer
167 0697 4 END;
168 0698 3 END;
169 0699 2
170 0700 2 ! The window is now suitably initialized. Set up necessary pointers.
171 0701 2 ! Now scan the map area, extracting retrieval pointers.
172 0702 2 !
173 0703 2 !
174 0704 2 MAP_AREA = .HEADER + .HEADER[FH1$B_MPOFFSET]*2; ! point to map area
175 0705 2 M_POINTER = .MAP_AREA + FM1$C_POINTERS; ! point to start of pointers
176 0706 2
177 0707 2 DECR COUNTER FROM .MAP_AREA[FM1$B_INUSE]/2 TO 1 DO
178 0708 3 BEGIN

```

```

179 0709
180 0710 COUNT = .M_POINTER[FM1$B_COUNT] + 1; ! get retrieval pointer count
181 0711 LBN = .M_POINTER[FM1$W_LOWLBN]; ! low order LBN
182 0712 LBN<16,85 = .M_POINTER[FM1$B_HIGHLBN]; ! and high order LBN
183 0713 M_POINTER = .M_POINTER + 4; ! update map pointer
184 0714
185 0715 ! See if this set of blocks is contiguous with the preceding (in the
186 0716 window). If so, merge them.
187 0717
188 0718
189 0719 IF .WINDOW[WCBSW_NMAP] NEQ 0 ! only if non-empty window
190 0720 AND .W_POINTER[WCBSW_PREVCOUNT] + .W_POINTER[WCBSL_PREVLBN] EQL .LBN
191 0721 AND .W_POINTER[WCBSW_PREVCOUNT] + .COUNT LSSU 65536
192 0722 THEN W_POINTER[WCBSW_PREVCOUNT] = .W_POINTER[WCBSW_PREVCOUNT] + .COUNT
193 0723
194 0724 ! Build a new retrieval pointer. If the window is full, shuffle the
195 0725 entries up by one. If this would cause the pointer mapping the
196 0726 desired VBN to fall off the top, we are done.
197 0727
198 0728
199 0729 ELSE
200 0730 BEGIN
201 0731 IF (.WINDOW[WCBSW_NMAP]+1)*6 + WCB$C_LENGTH
202 0732 GTRU .WINDOW[WCBSW_SIZE]
203 0733 THEN
204 0734 BEGIN
205 0735 IF .WINDOW[WCBSL_STVBN] + .WINDOW[WCBSW_P1_COUNT] GTRU .DESIRED_VBN
206 0736 THEN EXITLOOP;
207 0737
208 0738 WINDOW[WCBSW_NMAP] = .WINDOW[WCBSW_NMAP] - 1;
209 0739 WINDOW[WCBSL_STVBN] = .WINDOW[WCBSL_STVBN] + .WINDOW[WCBSW_P1_COUNT];
210 0740 CHSMOVE (.WINDOW[WCBSW_NMAP]*6, WINDOW_MAP+6, WINDOW_MAP);
211 0741 W_POINTER = .W_POINTER - 6;
212 0742 END;
213 0743
214 0744 ! Finally build the pointer and count it.
215 0745
216 0746
217 0747 W_POINTER[WCBSW_COUNT] = .COUNT;
218 0748 W_POINTER[WCBSL_LBN] = .LBN;
219 0749 W_POINTER = .W_POINTER + 6;
220 0750 WINDOW[WCBSW_NMAP] = .WINDOW[WCBSW_NMAP] + 1;
221 0751 END;
222 0752
223 0753 END; ! end of header scan loop
224 0754
225 0755 RETURN 1;
226 0756
227 0757 END; ! end of routine TURN_WINDOW1

```

.TITLE MWTUR1
.IDENT \V04-000\

.PSECT \$CODE\$,NOWRT,2

50	08	A8	FA	A6	58	A0	000AF	ADDW2	COUNT, -6(W_POINTER)	:	0722	
				50	40	11	000B3	BRB	11\$:		
				50	69	3C	000B5	MOVZWL	(R9), R0	:	0731	
				50	06	C4	000B8	MULL2	#6, R0	:		
				50	36	C0	000BB	ADDL2	#54, R0	:		
				10	00	ED	000BE	CMPZV	#0, #16, 8(R8), R0	:	0732	
				50	27	1E	000C4	BGEQU	10\$:		
				50	00	BE	3C	000C6	MOVZWL	@0(SP), R0	:	0735
				50	2C	A8	C0	000CA	ADDL2	44(R8), R0	:	
			0C	AC	50	D1	000CE	CMPL	R0, DESIRED_VBN	:		
					25	1A	000D2	BGTRU	12\$:		
					69	B7	000D4	DECW	(R9)	:	0738	
				50	00	BE	3C	000D6	MOVZWL	@0(SP), R0	:	0739
			2C	A8	50	C0	000DA	ADDL2	R0, 44(R8)	:		
				50	69	3C	000DE	MOVZWL	(R9), R0	:	0740	
				50	06	C4	000E1	MULL2	#6, R0	:		
	00	BE		36	50	28	000E4	MOVCS	R0, 54(R8), @0(SP)	:		
				56	06	C2	000EA	SUBL2	#6, W_POINTER	:	0741	
				86	5B	B0	000ED	MOVW	COUNT, (W_POINTER)+	:	0747	
				86	5A	D0	000F0	MOVL	LBN, (W_POINTER)+	:	0748	
				86	69	B6	000F3	INCW	(R9)	:	0750	
				86	04	AE	F5	000F5	SOBGTR	COUNTER, 7\$:	0707
				50	01	D0	000F9	MOVL	#1, R0	:	0755	
					04	000FC		RET		:	0757	

: Routine Size: 253 bytes. Routine Base: \$CODE\$ + 0000

: 228 0758 1 END
: 229 0759 0 ELUDOM

PSECT SUMMARY

Name	Bytes	Attributes
\$CODE\$	253	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

MWTUR1
V04-000

F 3
16-Sep-1984 01:26:30
14-Sep-1984 12:45:33

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[MOUNT.SRC]MWTUR1.B32;1 Page 8 (2)

RDH
V04

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS\$:MWTUR1/OBJ=OBJ\$:MWTUR1 MSRCS:MWTUR1/UPDATE=(ENH\$:MWTUR1)

: Size: 253 code + 0 data bytes
: Run Time: 00:14.7
: Elapsed Time: 00:34.9
: Lines/CPU Min: 3102
: Lexemes/CPU-Min: 30085
: Memory Used: 153 pages
: Compilation Complete

.....

Terminal window 1	Terminal window 2	Terminal window 3	Terminal window 4	Terminal window 5	Terminal window 6	Terminal window 7	Terminal window 8	Terminal window 9	Terminal window 10	Terminal window 11	Terminal window 12
Terminal window 13	Terminal window 14	Terminal window 15	RDHOME LIS	Terminal window 17	Terminal window 18	Terminal window 19	Terminal window 20	Terminal window 21	Terminal window 22	Terminal window 23	Terminal window 24
Terminal window 25	Terminal window 26	Terminal window 27	Terminal window 28	Terminal window 29	Terminal window 30	Terminal window 31	Terminal window 32	Terminal window 33	Terminal window 34	Terminal window 35	Terminal window 36
Terminal window 37	Terminal window 38	Terminal window 39	Terminal window 40	Terminal window 41	Terminal window 42	Terminal window 43	Terminal window 44	Terminal window 45	Terminal window 46	Terminal window 47	Terminal window 48
Terminal window 49	Terminal window 50	Terminal window 51	MTUR2 LIS	Terminal window 53	Terminal window 54	Terminal window 55	Terminal window 56	Terminal window 57	Terminal window 58	Terminal window 59	Terminal window 60
Terminal window 61	Terminal window 62	Terminal window 63	Terminal window 64	Terminal window 65	Terminal window 66	Terminal window 67	Terminal window 68	Terminal window 69	Terminal window 70	RUJMAN LIS	Terminal window 72
Terminal window 73	Terminal window 74	Terminal window 75	Terminal window 76	Terminal window 77	Terminal window 78	Terminal window 79	Terminal window 80	Terminal window 81	Terminal window 82	Terminal window 83	Terminal window 84
Terminal window 85	Terminal window 86	Terminal window 87	Terminal window 88	Terminal window 89	Terminal window 90	Terminal window 91	Terminal window 92	Terminal window 93	Terminal window 94	Terminal window 95	Terminal window 96
Terminal window 97	Terminal window 98	Terminal window 99	Terminal window 100	Terminal window 101	Terminal window 102	Terminal window 103	Terminal window 104	Terminal window 105	Terminal window 106	Terminal window 107	Terminal window 108
Terminal window 109	Terminal window 110	Terminal window 111	Terminal window 112	Terminal window 113	Terminal window 114	Terminal window 115	Terminal window 116	Terminal window 117	Terminal window 118	Terminal window 119	Terminal window 120
Terminal window 121	Terminal window 122	Terminal window 123	Terminal window 124	Terminal window 125	Terminal window 126	Terminal window 127	Terminal window 128	Terminal window 129	Terminal window 130	Terminal window 131	Terminal window 132
Terminal window 133	Terminal window 134	Terminal window 135	Terminal window 136	Terminal window 137	Terminal window 138	Terminal window 139	Terminal window 140	Terminal window 141	Terminal window 142	Terminal window 143	Terminal window 144