


```
CCCCCCCC LL      000000 SSSSSSSS FFFFFFFF LL
CCCCCCCC LL      000000 SSSSSSSS FFFFFFFF LL
CC        LL      00      00 SS      FF      LL
CC        LL      00      00 SS      FF      LL
CC        LL      00      00 SS      FF      LL
CC        LL      00      00 SS      FF      LL
CC        LL      00      00 SS      FF      LL
CC        LL      00      00 SS      FF      LL
CC        LL      00      00 SS      FF      LL
CC        LL      00      00 SS      FF      LL
CC        LL      00      00 SS      FF      LL
CC        LL      00      00 SS      FF      LL
CCCCCCCC LLLLLLLLLL 000000 SSSSSSSS FFFFFFFF LL
CCCCCCCC LLLLLLLLLL 000000 SSSSSSSS FFFFFFFF LL
LLLLLLLLLL .....
```

```
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 CLOSFL (LANGUAGE (BLISS32) ,
2 0002 0 IDENT = 'V04-000' ,
3 0003 0 ) =
4 0004 1 BEGIN
5 0005 1
6 0006 1 *****
7 0007 1 *
8 0008 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY *
9 0009 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. *
10 0010 1 * ALL RIGHTS RESERVED. *
11 0011 1 *
12 0012 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
13 0013 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE *
14 0014 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER *
15 0015 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
16 0016 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY *
17 0017 1 * TRANSFERRED. *
18 0018 1 *
19 0019 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE *
20 0020 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT *
21 0021 1 * CORPORATION. *
22 0022 1 *
23 0023 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS *
24 0024 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL. *
25 0025 1 *
26 0026 1 *
27 0027 1 *****
28 0028 1
29 0029 1 ++
30 0030 1
31 0031 1 FACILITY: MTAACP
32 0032 1
33 0033 1 ABSTRACT:
34 0034 1 Close out current file
35 0035 1
36 0036 1 ENVIRONMENT:
37 0037 1
38 0038 1 Starlet operating system, including privileged system services
39 0039 1 and internal exec routines
40 0040 1
41 0041 1 --
42 0042 1
43 0043 1
44 0044 1
45 0045 1 AUTHOR: D. H. GILLESPIE, CREATION DATE: 19-MAY-77 12:00
46 0046 1
47 0047 1 MODIFIED BY:
48 0048 1
49 0049 1 V03-006 MMD0308 Meg Dumont, 5-Jul-1984 13:56
50 0050 1 Fix to ensure the number of headers gotten in the open
51 0051 1 is the number written on the close.
52 0052 1
53 0053 1 V03-005 MMD0276 Meg Dumont, 23-Mar-1984 9:54
54 0054 1 Clear the VCBSV_FIL_ACCESS flag on output
55 0055 1
56 0056 1 V03-004 ROW0258 Ralph O. Weber 17-NOV-1983
57 0057 1 The Paul Painter Memorial Enhancement

```

```

58 0058 1 |
59 0059 1 |
60 0060 1 |
61 0061 1 |
62 0062 1 |
63 0063 1 |
64 0064 1 |
65 0065 1 |
66 0066 1 |
67 0067 1 |
68 0068 1 |
69 0069 1 |
70 0070 1 |
71 0071 1 |
72 0072 1 |
73 0073 1 |
74 0074 1 |
75 0075 1 |
76 0076 1 |
77 0077 1 |
78 0078 1 |
79 0079 1 |
80 0080 1 |
81 0081 1 |
82 0082 1 |
83 0083 1 |
84 0084 1 |
85 0085 1 |
86 0086 1 |
87 0470 1 |
88 0471 1 |
89 0472 1 |
90 0473 1 |
91 0474 1 |
92 0475 1 |
93 0476 1 |
94 0477 1 |
95 0478 1 |
96 0479 1 |
97 0480 1 |
98 0481 1 |
99 0482 1 |
100 0483 1 |
101 0484 1 |
102 0485 1 |
103 0486 1 |
104 0487 1 |
105 0488 1 |

Named for one of the unfortunate customers who suffered much
to determine the great UCBSL_MT_RECORD secret while trying to
create a user-written magtape driver, this change eliminates
use of the device dependent field, UCBSL_MT_RECORD in favor of
the device independent field, UCBSL_RECORD.

V03-003 MMD0154 Meg Dumont, 26-Apr-1983 9:12
Change reference to 80 to the symbol ANSI_LBLSZ. Change
the routine WRITE_TRAILERS to write an EOF4 if the mount
switch and number of labels originally written (i.e. if
we appending to the file) allows.

V03-002 MMD0002 Meg Dumont, 3-Jan-1983 15:27
Add code to check to ensure the user is not trying to write over
the trailer labels of the file.

V03-001 MMD0001 Meg Dumont, 5-Nov-1982 16:01
Clear user handling of EOT on deaccess of file

V02-005 REFORMAT Maria del C. Nasr 30-Jun-1980

V02-004 MCN0003 Maria del C. Nasr 10-Jun-1980
Add EOF3 label as part of HDR3 processing

**
LIBRARY 'SYSS$LIBRARY:LIB.L32';
REQUIRE 'SRC$:MTADEF.B32';

FORWARD ROUTINE
CLOSE_FILE : L$CLOSE_FILE NOVALUE, : CLOSE A FILE
UPDVCB_STATUS : COMMON_CALL NOVALUE, : UPDATE VCB STATUS BITS
WRITE_TRAILERS : L$WRITE_TRAILER NOVALUE; : WRITE TRAILER LABEL SET

EXTERNAL
CURRENT_UCB : REF BBLOCK, : ADDRESS OF CURRENT UNIT CONTROL BLOCK
HDR1 : REF BBLOCK, : ADDRESS OF HDR1(EOF1) LABEL
HDR2 : REF BBLOCK, : ADDRESS OF HDR2(EOF2) LABEL
HDR3 : REF BBLOCK, : ADDRESS OF HDR3(EOF3) LABEL
HDR4 : REF BBLOCK; : ADDRESS OF HDR4(EOF4) LABEL

EXTERNAL ROUTINE
SPACE_TM : COMMON_CALL NOVALUE, : Space a # of TM's
SYSS$FAO : ADDRESSING_MODE (ABSOLUTE), : FORMAT ASCII OUTPUT
WRITE_TM : L$WRITE_TM, : WRITE ONE TM
WRITE_BLOCK : COMMON_CALL; : WRITE BLOCK

```

```

107 0489 1 GLOBAL ROUTINE CLOSE_FILE : L$CLOSE_FILE NOVALUE =
108 0490 1
109 0491 1 ++
110 0492 1
111 0493 1 FUNCTIONAL DESCRIPTION:
112 0494 1     this routine closes out current file
113 0495 1
114 0496 1 CALLING SEQUENCE:
115 0497 1     close_file()
116 0498 1
117 0499 1 INPUT PARAMETERS:
118 0500 1     none
119 0501 1
120 0502 1 IMPLICIT INPUTS:
121 0503 1     current_ucb     - address of current ucb
122 0504 1     current_vcb     - address of current vcb
123 0505 1
124 0506 1 OUTPUT PARAMETERS:
125 0507 1     none
126 0508 1
127 0509 1 IMPLICIT OUTPUTS:
128 0510 1     none
129 0511 1
130 0512 1 ROUTINE VALUE:
131 0513 1     none
132 0514 1
133 0515 1 SIDE EFFECTS:
134 0516 1     file closed on magnetic tape; logical end_of_tape set;
135 0517 1     partial file cleared
136 0518 1
137 0519 1 Note: Logical end_of_tape will be set unless the
138 0520 1     user has tried to write over the EOF labels. This can happen
139 0521 1     if a user reads pass the TM after the data then does a write
140 0522 1     at that point. The write will fail at FDT processing time, and
141 0523 1     the user will get a status of SSS ENDOFFILE. However the WCB now looks
142 0524 1     like it was writing and when the MTAACP tries to close the
143 0525 1     file, the MTAACP must be sure that the user hasn't tried to do this.
144 0526 1
145 0527 1 --
146 0528 1
147 0529 2 BEGIN
148 0530 2
149 0531 2 EXTERNAL REGISTER
150 0532 2     COMMON_REG;
151 0533 2
152 0534 2 ! If the user has tried to write over the trailer labels then the
153 0535 2 ! MTAACP will be postioned after the first EOF. Therefore close
154 0536 2 ! the file as if nothing had been changed on the tape.
155 0537 2
156 0538 2 IF .CURRENT_VCB[VCBSB_TM] EQL 2
157 0539 2 THEN
158 0540 2     BEGIN
159 0541 2     SPACE_TM(1);           ! Go to TM after trailer labels
160 0542 2     KERNEC_CALL(UPDVCB_STATUS,0) ! Set not at logical EOv
161 0543 2     END
162 0544 2
163 0545 2     ! if only the headers have been written without the tm before the data

```

164 0546
165 0547
166 0548
167 0549
168 0550
169 0551
170 0552
171 0553
172 0554
173 0555
174 0556
175 0557
176 0558
177 0559
178 0560
179 0561
180 0562

```
! then write this tm first
ELSE
BEGIN
IF .CURRENT_VCB[VCBSB_TM] EQL 0
THEN
WRITE_TM();

IF .CURRENT_VCB[VCBSB_TM] EQL 1
THEN
WRITE_TRAILERS('F');

WRITE_TM();
WRITE_TM();
KERNEL_CALL(UPDVCB_STATUS,1);
END;
! Set Logical EOV
END;
```

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

.EXTRN CURRENT_UCB, HDR1
.EXTRN HDR2, HDR3, HDR4
.EXTRN SPACE_TM, SYSSFAO
.EXTRN WRITE_TM, WRITE_BLOCK
.EXTRN SYSSCMKRNL
```

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

02	2E	AB	91	00000	CLOSE_FILE::			
					CMPB	46(CURRENT_VCB), #2		0538
					BNEQ	1\$		
0000G	CF				PUSHL	#1		0541
					CALLS	#1, SPACE_TM		
					CLRL	-(SP)		0542
					BRB	4\$		
					TSTB	46(CURRENT_VCB)		0550
					BNEQ	2\$		
		0000G			BSBW	WRITE_TM		0552
01	2E	AB	91	00019	2\$:	46(CURRENT_VCB), #1		0554
					BNEQ	3\$		
7E	46	8F	9A	0001F	MOVZBL	#70, -(SP)		0556
					BSBW	WRITE_TRAILERS		
5E					ADDL2	#4, SP		
					BSBW	WRITE_TM		0558
					BSBW	WRITE_TM		0559
					PUSHL	#1		0560
					PUSHL	#1		
					PUSHL	SP		
00000000G	9F	0000V	CF	9F	00035	PUSHAB	UPDVCB_STATUS	
					04	FB	#4, @#SYSSCMKRNL	
					05	00040	RSB	0562

: Routine Size: 65 bytes, Routine Base: \$CODE\$ + 0000

CLOSFL
V04-000

: 181

0563 1

^{E 1}
16-Sep-1984 02:11:05
14-Sep-1984 12:46:36

VAX-11 Bliss-32 V4.0-742
[MTAACP.SRC]CLOSFL.B32;1

Page 5
(2)

CN
VO

```

183 0564 1 GLOBAL ROUTINE WRITE_TRAILERS (END_FLAG) : L$WRITE_TRAILER NOVALUE =
184 0565 1
185 0566 1 !++
186 0567 1
187 0568 1 FUNCTIONAL DESCRIPTION:
188 0569 1 This routine writes out eof(v)1, eof(v)2, eof(v)3 and eof(v)4
189 0570 1 system labels.
190 0571 1
191 0572 1 CALLING SEQUENCE:
192 0573 1 write_trailer(arg1)
193 0574 1
194 0575 1 INPUT PARAMETERS:
195 0576 1 arg1 - 'v' for end of volume
196 0577 1 'f' for end of file
197 0578 1
198 0579 1 IMPLICIT INPUTS:
199 0580 1 none
200 0581 1
201 0582 1 OUTPUT PARAMETERS:
202 0583 1 none
203 0584 1
204 0585 1 IMPLICIT OUTPUTS:
205 0586 1 system trailer label written to tape
206 0587 1
207 0588 1 ROUTINE VALUE:
208 0589 1 none
209 0590 1
210 0591 1 SIDE EFFECTS:
211 0592 1 none
212 0593 1
213 0594 1 --
214 0595 1
215 0596 2 BEGIN
216 0597 2
217 0598 2 EXTERNAL REGISTER
218 0599 2 COMMON_REG;
219 0600 2
220 0601 2 LOCAL
221 0602 2 BLCNT_DESC : VECTOR [2];
222 0603 2
223 0604 2 ! calculate block count by subtracting the block count after the tape mark
224 0605 2 ! at the start of the data area from the current block count
225 0606 2
226 0607 2 BLCNT_DESC[0] = E01$$ BLOCKCNT;
227 0608 2 BLCNT_DESC[1] = HDR1[E01$T_BLOCKCNT];
228 0609 2 SYSS$AO(DESCRIPTOR('!6ZL'),
229 0610 2 0,
230 0611 2 BLCNT_DESC,
231 0612 2 .CURRENT_UCB[UCB$SL_RECORD] - .CURRENT_VCB[VCB$SL_ST_RECORD]);
232 0613 2
233 0614 2 ! format eof1 trailer
234 0615 2
235 0616 2 HDR1[E01$E01LID] = 'EOF1';
236 0617 2 (.HDR1)<16, 8> = .END_FLAG<0, 8> ! either f or v
237 0618 2
238 0619 2 ! write tm plus eof1
239 0620 2

```



```

240 0621 2 WRITE_TM();
241 0622 2 WRITE_BLOCK(.HDR1, ANSI_LBLSZ);
242 0623 2
243 0624 2 ! Check the label count sure that we aren't writing extra headers onto the
244 0625 2 ! the tape. This means that whatever number of labels with which the file
245 0626 2 ! was opened we must ensure that only that number gets written on the file
246 0627 2 ! close, else we violate the ANSI standard.
247 0628 2
248 0629 2 IF .CURRENT_VCB[VCBSB_LBLCNT] GEQ 2
249 0630 2 THEN
250 0631 2 BEGIN
251 0632 2 ! format and write eof2
252 0633 2
253 0634 2 HDR2[E02$L_E02LID] = 'EOF2';
254 0635 2 (.HDR2)<16, 8> = .END_FLAG<0, 8>;
255 0636 2 WRITE_BLOCK(.HDR2, ANSI_LBLSZ);
256 0637 2
257 0638 2 ! Format and write eof3 and eof4, if mount switch (/NOHDR3 or /INTERCHANGE)
258 0639 2 ! allows it.
259 0640 2
260 0641 2 IF (NOT .CURRENT_VCB[VCBSV_NOHDR3])
261 0642 2 THEN
262 0643 2 BEGIN
263 0644 2 HDR3[E03$L_E03LID] = 'EOF3';
264 0645 2 (.HDR3)<16, 8> = .END_FLAG<0, 8>;
265 0646 2 WRITE_BLOCK(.HDR3, ANSI_LBLSZ);
266 0647 2
267 0648 2 IF .CURRENT_VCB[VCBSB_LBLCNT] GTR 3
268 0649 2 THEN
269 0650 2 BEGIN
270 0651 2 HDR4[E04$L_E04LID] = 'EOF4';
271 0652 2 (.HDR4)<16, 8> = .END_FLAG<0, 8>;
272 0653 2 WRITE_BLOCK(.HDR4, ANSI_LBLSZ);
273 0654 2
274 0655 2 END;
275 0656 2 END;
276 0657 2 END;
277 0658 1 ! END OF ROUTINE

```

```

4C 5A 36 21 00041 P.AAB: .ASCII \!6ZL\
          00045 .BLKB 3
          00000004, 00048 P.AAA: .LONG 4
          00000000' 0004C .ADDRESS P.AAB

```

```

SE 04 C2 00000 WRITE_TRAILERS::
          06 DD 00003 SUBL2 #4, SP
          36 C1 00005 PUSHL #6
04 AE 0000G CF 0000G CF D0 0000C ADDL3 #54, HDR1, BLCNT_DESC+4
          30 AB C3 00011 MOVL CURRENT_UCB, R0
7E 00B0 C0 04 AE 9F 0001B SUBL3 48(CURRENT_VCB), 176(R0), -(SP)
          04 AE 9F 0001B PUSHAB BLCNT_DESC
          7E D4 0001B CLRL -(SP)
          D8 AF 9F 0001D PUSHAB P.AAA

```

```

: 0564
: 0607
: 0608
: 0612
: 0609
:

```

0000G	DF	08	00000000G	9F	04	FB	00C20	CALLS	#4, @MSYSS\$FA0		
			0000G	DF	8F	DO	00027	MOVL	#826691397, @HDR1		0616
				10	AE	FO	00030	INSV	END_FLAG, #16, #8, @HDR1		0617
					0000G	30	00038	BSBW	WRITE_TM		0621
				7E	50	8F	9A	MOVZBL	#80, -(SP)		0622
					0000G	CF	DD	PUSHL	HDR1		
			0000G	CF	02	FB	00043	CALLS	#2, WRITE_BLOCK		
				02	48	AB	91	CMPB	72(CURRENT_VCB), #2		0629
						5F	1F	BLSSU	1\$		
				50	0000G	CF	DO	MOVL	HDR2, R0		0635
				60	32464F45	8F	DO	MOVL	#843468613, (R0)		
			02	A0	0C	AE	90	MOVB	END_FLAG, 2(R0)		0636
				7E	50	8F	9A	MOVZBL	#80, -(SP)		0637
						50	DD	PUSHL	R0		
			0000G	CF	02	FB	00065	CALLS	#2, WRITE_BLOCK		
					2C	AB	95	TSTB	44(CURRENT_VCB)		0642
						3E	19	BLSS	1\$		
				50	0000G	CF	DO	MOVL	HDR3, R0		0645
				60	33464F45	8F	DO	MOVL	#860245829, (R0)		
			02	A0	0C	AE	90	MOVB	END_FLAG, 2(R0)		0646
				7E	50	8F	9A	MOVZBL	#80, -(SP)		0647
						50	DD	PUSHL	R0		
			0000G	CF	02	FB	00086	CALLS	#2, WRITE_BLOCK		
				03	48	AB	91	CMPB	72(CURRENT_VCB), #3		0649
						1C	1B	BLEQU	1\$		
				50	0000G	CF	DO	MOVL	HDR4, R0		0652
				60	34464F45	8F	DO	MOVL	#877023045, (R0)		
			02	A0	0C	AE	90	MOVB	END_FLAG, 2(R0)		0653
				7E	50	8F	9A	MOVZBL	#80, -(SP)		0654
						50	DD	PUSHL	R0		
			0000G	CF	02	FB	000A8	CALLS	#2, WRITE_BLOCK		
				5E		08	CO	ADDL2	#8, SP		0658
						05	000B0	RSB			

; Routine Size: 177 bytes, Routine Base: \$CODE\$ + 0050

; 278 0659 1

```

: 280 0660 1 ROUTINE UPDVCB_STATUS(LEOV) : COMMON_CALL NOVALUE =
: 281 0661 1
: 282 0662 1 ++
: 283 0663 1
: 284 0664 1 FUNCTIONAL DESCRIPTION:
: 285 0665 1     this routine updates the status bits in the current vcb
: 286 0666 1
: 287 0667 1 CALLING SEQUENCE:
: 288 0668 1     updvcb_status(LEOV), called in kernel mode
: 289 0669 1
: 290 0670 1 INPUT PARAMETERS:
: 291 0671 1     Value to set logical end of volume
: 292 0672 1         0 = clear bit not logical EOV
: 293 0673 1         1 = set bit at logical EOV
: 294 0674 1
: 295 0675 1 IMPLICIT INPUTS:
: 296 0676 1     current_vcb - address of current volume control block
: 297 0677 1
: 298 0678 1 OUTPUT PARAMETERS:
: 299 0679 1     none
: 300 0680 1
: 301 0681 1 IMPLICIT OUTPUTS:
: 302 0682 1     status set to at logical end of tape with no partial file
: 303 0683 1
: 304 0684 1 ROUTINE VALUE:
: 305 0685 1     none
: 306 0686 1
: 307 0687 1 SIDE EFFECTS:
: 308 0688 1     none
: 309 0689 1
: 310 0690 1 --
: 311 0691 1
: 312 0692 2 BEGIN
: 313 0693 2
: 314 0694 2 EXTERNAL REGISTER
: 315 0695 2 COMMON_REG;
: 316 0696 2
: 317 0697 2 CURRENT_VCB[VCBSV_LOGICEOVS] = .LEOV;
: 318 0698 2 CURRENT_VCB[VCBSV_MUSTCLOSE] = 0;
: 319 0699 2 CURRENT_VCB[VCBSV_PARTFILE] = 0;
: 320 0700 2 CURRENT_VCB[VCBSV_ENUSEREOT]= 0;           ! reset user enables EOT handling
: 321 0701 2 CURRENT_VCB[VCBSV_FIL_ACCESS]= 0;       ! reset users' accessibility to file
: 322 0702 2
: 323 0703 1 END;                                     ! end of routine

```

0000 0000 UPDVCB_STATUS:

OB	AB	01	0B	01	04	AC	FO	00002	.WORD	Save nothing	: 0660
			2D	AB	41	8F	8A	00009	INSV	LEOV, #1, #1, 11(CURRENT_VCB)	: 0697
					42	8F	8A	0000E	BICB2	#65, 11(CURRENT_VCB)	: 0699
							04	00013	BICB2	#66, 45(CURRENT_VCB)	: 0701
									RET		: 0703

: Routine Size: 20 bytes, Routine Base: \$CODE\$ + 0101

```

: 324      0704 1
: 325      0705 1 END
: 326      0706 1
: 327      0707 0 ELUDOM

```

PSECT SUMMARY

```

:
: Name          Bytes          Attributes
:
: $CODE$       277 NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPI,ALIGN(2)
:

```

Library Statistics

```

:
: File          Total  Symbols  Percent  Pages  Processing
:              -----  Loaded  -----  Mapped  Time
:
: _$255$DUA28:[SYSLIB]LIB.L32;1  18619      20      0      1000    00:01.9
:

```

COMMAND QUALIFIERS

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

```

: Size:          262 code + 15 data bytes
: Run Time:      00:09.2
: Elapsed Time: 00:31.5
: Lines/CPU Min: 4615
: Lexemes/CPU-Min: 19664
: Memory Used: 93 pages
: Compilation Complete

```



