

```
FFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFF
FFF
FFF
FFF
FFF
FFF
FFF
FFFFFFFFFFFF
FFFFFFFFFFFF
FFFFFFFFFFFF
FFF
FFF
FFF
FFF
FFF
FFF
FFF
FFF
FFF
FFF
```

```
111
111
111
111111
111111
111111
111
111
111
111
111
111
111
111
111
111
111
111
111111111
111111111
111111111
```

```
111
111
111
111111
111111
111111
111
111
111
111
111
111
111
111
111
111
111
111
111111111
111111111
111111111
```

```
AAAAAAAAAA
AAAAAAAAAA
AAAAAAAAAA
AAA
AAA
AAA
AAA
AAA
AAA
AAA
AAA
AAA
AAA
AAA
AAA
AAAAA
AAAAA
AAAAA
AAA
AAA
AAA
AAA
AAA
AAA
AAA
AAA
AAA
```

```

SSSSSSSS NN NN DDDDDDD EEEEEEEEE RRRRRRR LL
SSSSSSSS NN NN DDDDDDD EEEEEEEEE RRRRRRR LL
SS NN NN DD DD EE RR RR LL
SS NN NN DD DD EE RR RR LL
SS NNNN NN DD DD EE RR RR LL
SS NNNN NN DD DD EE RR RR LL
SSSSSS NN NN DD DD EEEEEEE RRRRRRR LL
SSSSSS NN NN DD DD EEEEEEE RRRRRRR LL
SS NN NNNN DD DD EE RR RR LL
SS NN NNNN DD DD EE RR RR LL
SS NN NN DD DD EE RR RR LL
SSSSSS NN NN DDDDDDD EEEEEEEEE RR RR LLLLLLLLL
SSSSSS NN NN DDDDDDD EEEEEEEEE RR RR LLLLLLLLL

```

```

LL IIIIII SSSSSSS
LL IIIIII SSSSSSS
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
LLLLLLLLLL IIIIII SSSSSSS
LLLLLLLLLL IIIIII SSSSSSS

```



1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57

```

0001 0 MODULE SNDERL (
0002 0
0003 0     LANGUAGE (BLISS32),
0004 0     IDENT = 'V04-000'
0005 1 BEGIN
0006 1
0007 1
0008 1 *****
0009 1 *
0010 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0011 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0012 1 * ALL RIGHTS RESERVED.
0013 1 *
0014 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0015 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0016 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0017 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0018 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0019 1 * TRANSFERRED.
0020 1 *
0021 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0022 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0023 1 * CORPORATION.
0024 1 *
0025 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0026 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0027 1 *
0028 1 *
0029 1 *****
0030 1
0031 1 ++
0032 1
0033 1 FACILITY: F11ACP Structure Level 1
0034 1
0035 1 ABSTRACT:
0036 1
0037 1     This routine sends a message to the error logger to inform it of a
0038 1     volume mount or dismount.
0039 1
0040 1 ENVIRONMENT:
0041 1
0042 1     STARLET operating system, including privileged system services
0043 1     and internal exec routines.
0044 1
0045 1 --
0046 1
0047 1
0048 1 AUTHOR: Andrew C. Goldstein, CREATION DATE: 23-Jun-1978 18:47
0049 1
0050 1 MODIFIED BY:
0051 1
0052 1     V03-001 LMP0221 L. Mark Pilant, 27-Mar-1984 13:21
0053 1     Change UCBSL_OWNUIC to ORBSL_OWNER and UCBSW_VPROT to
0054 1     ORBSW_PROT.
0055 1
0056 1     A0100 ACG0001 Andrew C. Goldstein, 10-Oct-1978 20:01
0057 1     Previous revision history moved to F11A.REV

```

```
.. 58      0058 1  |  
.. 59      0059 1  |**  
.. 60      0060 1  |  
.. 61      0061 1  |  
.. 62      0062 1  | LIBRARY 'SYS$LIBRARY:LIB,L32';  
.. 63      0063 1  | REQUIRE 'SRC$FCPDEF.B32';  
.. 64      0378 1  |  
.. 65      0379 1  |  
.. 66      0380 1  | This routine is called at raised IPL and must be locked into the working set.  
.. 67      0381 1  |  
.. 68      0382 1  |  
.. 69      0383 1  | LOCK_CODE;
```

```

71 0384 1 GLOBAL ROUTINE SEND_ERRLOG (MODE, UCB) =
72 0385 1
73 0386 1 !++
74 0387 1
75 0388 1 FUNCTIONAL DESCRIPTION:
76 0389 1
77 0390 1     This routine sends a message to the error logger to inform it of a
78 0391 1     volume mount or dismount.
79 0392 1
80 0393 1
81 0394 1 CALLING SEQUENCE:
82 0395 1     SEND_ERRLOG (ARG1, ARG2)
83 0396 1
84 0397 1 INPUT PARAMETERS:
85 0398 1     ARG1: 1 to signal mount
86 0399 1           0 to signal dismount
87 0400 1     ARG3: address of UCB
88 0401 1
89 0402 1 IMPLICIT INPUTS:
90 0403 1     NONE
91 0404 1
92 0405 1 OUTPUT PARAMETERS:
93 0406 1     NONE
94 0407 1
95 0408 1 IMPLICIT OUTPUTS:
96 0409 1     NONE
97 0410 1
98 0411 1 ROUTINE VALUE:
99 0412 1     1
100 0413 1
101 0414 1 SIDE EFFECTS:
102 0415 1     Message sent to error logger
103 0416 1
104 0417 1 !--
105 0418 1
106 0419 2 BEGIN
107 0420 2
108 0421 2 MAP
109 0422 2     UCB          : REF BBLOCK;    ! UCB argument
110 0423 2
111 0424 2 LINKAGE
112 0425 2     L_ERL_ALLOC = JSB (REGISTER = 1) :
113 0426 2                   GLOBAL (ADDRESS = 2)
114 0427 2                   NOTUSED (3, 4, 5, 6, 7, 8, 9, 10, 11),
115 0428 2
116 0429 2     L_ERL_RELEASE = JSB (REGISTER = 2) :
117 0430 2                   NOTUSED (3, 4, 5, 6, 7, 8, 9, 10, 11);
118 0431 2
119 0432 2 LOCAL
120 0433 2     ORB          : REF BBLOCK,    ! local address of ORB
121 0434 2     MSG_BUFFER  : REF BBLOCK;    ! other buffer pointer to dodge MOVC
122 0435 2
123 0436 2 EXTERNAL ROUTINE
124 0437 2     ERL$ALLOCEMB : L_ERL_ALLOC ADDRESSING_MODE (GENERAL),
125 0438 2                   ! allocate error log buffer
126 0439 2     ERL$RELEASEMB : L_ERL_RELEASE ADDRESSING_MODE (GENERAL);
127 0440 2                   ! release error log buffer

```

```

128 0441
129 0442
130 0443 ! Allocate an error log buffer. If this fails, forget it.
131 0444
132 0445
133 0446 BEGIN
134 0447 GLOBAL REGISTER
135 0448 ADDRESS = 2 : REF BBLOCK; ! pointer to error log buffer
136 0449
137 0450 IF NOT ERL$ALLOCEMB (EMBSK_VM_LENGTH)
138 0451 THEN RETURN 1;
139 0452 MSG_BUFFER = .ADDRESS;
140 0453 END;
141 0454
142 0455 ! Now fill in the message buffer.
143 0456
144 0457
145 0458 IF .MODE
146 0459 THEN MSG_BUFFER[EMBSW_VM_ENTRY] = EMBSK_VM
147 0460 ELSE MSG_BUFFER[EMBSW_VM_ENTRY] = EMBSK_VD; ! Log entry type
148 0461
149 0462 ORB = .UCB[UCBSL_ORB];
150 0463 MSG_BUFFER[EMBSL_VM_OWNUIC] = .ORB[ORBSL_OWNER];
151 0464 MSG_BUFFER[EMBSL_VM_ERRCNT] = .UCB[UCBSW_ERRCNT];
152 0465 MSG_BUFFER[EMBSL_VM_OPRCNT] = .UCB[UCBSL_OPCNT];
153 0466 MSG_BUFFER[EMBSW_VM_UNIT] = .UCB[UCBSW_UNIT];
154 0467
155 0468 MSG_BUFFER[EMBSW_VM_VOLNUM] = 0;
156 0469 MSG_BUFFER[EMBSW_VM_NUMSET] = 0;
157 0470
158 0471 CH$MOVE (. (BBLOCK [.UCB[UCBSL_DDB], DDB$T_NAME])<0,8> + 1,
159 0472 BBLOCK [.UCB[UCBSL_DDB], DDB$T_NAME],
160 0473 MSG_BUFFER[EMBSB_VM_NAMLNG]);
161 0474
162 0475 IF .BBLOCK[UCB[UCBSL_DEVCHAR], DEV$V_FOR]
163 0476 OR NOT .BBLOCK[UCB[UCBSL_DEVCHAR], DEV$V_SQD]
164 0477 THEN
165 0478 CH$MOVE (VCBS$ VOLNAME,
166 0479 BBLOCK [.UCB[UCBSL_VCB], VCB$T_VOLNAME],
167 0480 MSG_BUFFER[EMBS$T_VM_LABEL])
168 0481 ELSE
169 0482 BEGIN
170 0483 LOCAL
171 0484 MVL : REF BBLOCK, ! magtape volume labels
172 0485 MVL_ENTRY : REF BBLOCK, ! address of label entry
173 0486 RUN, ! relative unit number
174 0487 RVT : REF BBLOCK, ! relative volume table
175 0488 UCBLIST : REF VECTOR, ! address of UCB list
176 0489 VCB : REF BBLOCK; ! volume control block
177 0490 VCB = .UCB[UCBSL_VCB];
178 0491 RVT = .VCB[VCBSL_RVT];
179 0492 UCBLIST = RVT[RVT$L_UCBLST];
180 0493 MVL = .VCB[VCBSL_MVC];
181 0494 MSG_BUFFER[EMBSW_VM_NUMSET] = .MVL[MVL$B_NVOLS]; ! no of volumes in vol set known
182 0495 CH$FILL(' ',VCBS$ VOLNAME,MSG_BUFFER[EMBS$T_VM_LABEL]);
183 0496 INCR I FROM 0 TO .RVT[RVT$B_NVOLS] - 1 DO
184 0497 BEGIN

```

```

185 0498 4      RUN = .I;
186 0499 4      IF .UCBLISTC.I] EQL .UCB THEN EXITLOOP;
187 0500 3      END;
188 0501 3      MVL_ENTRY = .MVL + MVL$K_FIXLEN;
189 0502 3      INCR I FROM 0 TO .MVL[MVLSB_NVOLS] -1 DO
190 0503 4      BEGIN
191 0504 4      IF .MVL_ENTRY[MVLSB_RVN] EQL .RUN
192 0505 4      AND .MVL_ENTRY[MVLSV_MOUNTED]
193 0506 4      THEN
194 0507 5      BEGIN
195 0508 5      MSG_BUFFER[EMB$W_VM_VOLNUM] = .I + 1;
196 0509 5      CH$COPY(MVL$$ VOCLBC, MVL_ENTRY[MVL$T_VOLLBL], ' ',
197 0510 5      VCB$$_VOLNAME, MSG_BUFFER[EMB$T_VM_LABEL]);
198 0511 5      EXITLOOP;
199 0512 4      END;
200 0513 4      MVL_ENTRY = .MVL_ENTRY + MVL$K_LENGTH;
201 0514 3      END;
202 0515 2      END;
203 0516 2
204 0517 2      ! Finally release the buffer and make the entry.
205 0518 2      !
206 0519 2
207 0520 2      ERL$RELEASEMB (.MSG_BUFFER);
208 0521 2
209 0522 2      RETURN 1;
210 0523 2
211 0524 1      END;
! end of routine SEND_FRRLOG

```

```

.TITLE SNDRERL
.IDENT \V04-000\

.EXTRN ERL$ALLOCEMB, ERL$RELEASEMB

.PSECT $LOCKEDC1$,NOWRT,2

.ENTRY SEND_ERRLOG, Save R2,R3,R4,R5,R6,R7,R8,R9,- ; 0384
R10
MOVL #62, R1 ; 0450
JSB ERL$ALLOCEMB
BLBS R0, 1$
BRW 13$
MOVL ADDRESS, MSG_BUFFER ; 0452
BLBC MODE, 2$ ; 0458
MOVZBW #64, 4(MSG_BUFFER) ; 0459
BRB 3$
MOVZBW #65, 4(MSG_BUFFER) ; 0460
MOVL UCB, R9 ; 0462
MOVL 28(R9), ORB
MOVL (ORB), 16(MSG_BUFFER) ; 0463
MOVZWL 130(R9), 20(MSG_BUFFER) ; 0464
MOVL 112(R9), 24(MSG_BUFFER) ; 0465
MOVW 84(R9), 28(MSG_BUFFER) ; 0466
CLRL 46(MSG_BUFFER) ; 0468
MOVL 40(R9), R0 ; 0471
MOVZBL 20(R0), R1
INCL R1

```

```

07FC 00000
51 00000000G 3E D0 00002
03 50 E8 0000B
00C0 31 0000E
5A 52 D0 00011 1$:
07 04 AC E9 00014
04 AA 40 8F 9B 00018
05 11 0001D
04 AA 41 8F 9B 0001F 2$:
59 08 AC D0 00024 3$:
50 1C A9 D0 00028
10 AA 60 D0 0002C
14 AA 0082 C9 3C 00030
18 AA 70 A9 D0 00036
1C AA 54 A9 B0 0003B
50 2E AA D4 00040
51 28 A9 D0 00043
51 14 A0 9A 00047
51 D6 0004B

```

1E	AA	14	A0	51	28	0004D	MOVCS	R1, 20(R0), 30(MSG_BUFFER)	0473
			05	3B	A9	E8 00053	BLBS	59(R9), 4\$	0475
	OC	38	A9	05	E0	00057	BBS	#5, 56(R9), 5\$	0476
			50	34	A9	D0 0005C 4\$:	MOVL	52(R9), R0	0479
32	AA	14	A0	0C	28	00060	MOVCS	#12, 20(R0), 50(MSG_BUFFER)	0480
				60	11	00066	BRB	12\$	
			50	34	A9	D0 00068 5\$:	MOVL	52(R9), VCB	0490
			56	20	A0	D0 0006C	MOVL	32(VCB), RVT	0491
			58	44	A6	9E 00070	MOVAB	68(R6), UCBLIST	0492
			57	34	A0	D0 00074	MOVL	52(VCB), MVL	0493
		30	AA	0B	A7	9B 00078	MOVZBW	11(MVL), 48(MSG_BUFFER)	0494
OC		20	6E	00	2C	0007D	MOVCS	#0, (SP), #32, #12, 50(MSG_BUFFER)	0495
				32	AA	00082			
			51	0B	A6	9A 00084	MOVZBL	11(RVT), R1	0496
			50		01	CE 00088	MNEGL	#1, I	0499
					09	11 0008B	BRB	7\$	
			56		50	D0 0008D 6\$:	MOVL	I, RUN	0498
			59		6840	D1 00090	CPL	(UCBLIST)[I], R9	0499
					04	13 00094	BEQL	8\$	
	F3		50		51	F2 00096 7\$:	AOBLSS	R1, I, 6\$	0496
			58	24	A7	9E 0009A 8\$:	MOVAB	36(R7), MVL_ENTRY	0501
			57	0B	A7	9A 0009E	MOVZBL	11(MVL), R7	0502
			59		01	CE 000A2	MNEGL	#1, I	0504
					1D	11 000A5	BRB	11\$	
56	06	A8	08		00	ED 000A7 9\$:	CMPZV	#0, #8, 6(MVL_ENTRY), RUN	
					12	12 000AD	BNEQ	10\$	
			0E	07	A8	E9 000AF	BLBC	7(MVL_ENTRY), 10\$	0505
			59		01	A1 000B3	ADDW3	#1, I, 46(MSG_BUFFER)	0508
OC	2E	AA	68		06	2C 000B8	MOVCS	#6, (MVL_ENTRY), #32, #12, 50(MSG_BUFFER)	0510
				32	AA	000BD			
				07	11	000BF	BRB	12\$	0507
			58		08	CO 000C1 10\$:	ADDL2	#8, MVL_ENTRY	0513
			59		57	F2 000C4 11\$:	AOBLSS	R7, I, 9\$	0502
			52		5A	D0 000C8 12\$:	MOVL	MSG_BUFFER, R2	0520
				00000000G	00	16 0J0CB	JSB	ERL\$RELEASEMB	
			50		01	D0 000D1 13\$:	MOVL	#1, R0	0522
					04	000D4	RET		0524

: Routine Size: 213 bytes, Routine Base: \$LOCKEDC1\$ + 0000

: 212 0525 1
: 213 0526 1 END
: 214 0527 0 ELUDOM

PSECT SUMMARY

Name	Bytes	Attributes
\$LOCKEDC1\$	213	NOVEC, NOWRT, RD, EXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)

Library Statistics

File	Symbols		Pages Mapped	Processing Time
	Total	Loaded Percent		
:_\$255\$DUA28:[SYSLIB]LIB.L32;1	18619	37 0	1000	00:01.9

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,IN'TIAL,OPTIMIZE)/LIS=LIS\$:SNDRL/OBJ=OBJ\$:SNDRL MSRC\$:SNDRL/UPDATE=(ENH\$:SNDRL)

: Size: 213 code + 0 data bytes
: Run Time: 00:09.3
: Elapsed Time: 00:25.7
: Lines/CPU Min: 3411
: Lexemes/CPU-Min: 17061
: Memory Used: 128 pages
: Compilation Complete

FCPDEF B32	ACL CNTRL LIS	ACL SUBR LIS	ACPCNTRL LIS
SNDR LIS	FLIX	FLIBXQP MAP	ACCESS LIS
TRUNC LIS	WTRN LIS	FILESERV MAP	SNDSMB LIS