


```

LL          000000  GGGGGGGG  FFFFFFFFFF  IIIIII  LL          EEEEEEEEEE
LL          000000  GGGGGGGG  FFFFFFFFFF  IIIIII  LL          EEEEEEEEEE
LL          00      00  GG          FF          II          LL          EE
LL          00      00  GG          FF          II          LL          EE
LL          00      00  GG          FF          II          LL          EE
LL          00      00  GG          FF          II          LL          EE
LL          00      00  GG          FF          II          LL          EE
LL          00      00  GG          FFFFFFFF  II          LL          EEEEEEEEE
LL          00      00  GG          FFFFFFFF  II          LL          EEEEEEEEE
LL          00      00  GG  GGGGGG  FF          II          LL          EE
LL          00      00  GG  GGGGGG  FF          II          LL          EE
LL          00      00  GG          GG  FF          II          LL          EE
LL          00      00  GG          GG  FF          II          LL          EE
LLLLLLLLLLL 000000  GGGGGG  FF          IIIIII  LLLLLLLLLL  EEEEEEEEE
LLLLLLLLLLL 000000  GGGGGG  FF          IIIIII  LLLLLLLLLL  EEEEEEEEE

```

65
74

4E

45

4C

4F

45

```

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
LLLLLLLLLLL IIIIII  SSSSSSSS
LLLLLLLLLLL IIIIII  SSSSSSSS

```

```

1 0001 0 MODULE OPC$LOGFILE (
2 0002 0 LANGUAGE (BLISS32),
3 0003 0 IDENT = 'V04-000'
4 0004 0 ) =
5 0005 0
6 0006 0 *****
7 0007 0 *
8 0008 0 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY *
9 0009 0 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. *
10 0010 0 * ALL RIGHTS RESERVED. *
11 0011 0 *
12 0012 0 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
13 0013 0 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE *
14 0014 0 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER *
15 0015 0 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
16 0016 0 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY *
17 0017 0 * TRANSFERRED. *
18 0018 0 *
19 0019 0 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE *
20 0020 0 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT *
21 0021 0 * CORPORATION. *
22 0022 0 *
23 0023 0 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS *
24 0024 0 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL. *
25 0025 0 *
26 0026 0 *
27 0027 0 *****
28 0028 0
29 0029 0 ++
30 0030 0 FACILITY:
31 0031 0
32 0032 0 OPCOM
33 0033 0
34 0034 0 ABSTRACT:
35 0035 0
36 0036 0 This module contains the specialized logic to service
37 0037 0 a particular type of request sent by a user to OPCOM.
38 0038 0
39 0039 0 Environment:
40 0040 0
41 0041 0 VAX/VMS operating system.
42 0042 0
43 0043 0 Author:
44 0044 0
45 0045 0 Steven T. Jeffreys
46 0046 0
47 0047 0 Creation date:
48 0048 0
49 0049 0 March 10, 1981
50 0050 0
51 0051 0 Revision history:
52 0052 0
53 0053 0 V03-001 CWH3001 CW Hobbs 30-Jul-1983
54 0054 0 Various and sundry things to make OPCOM distributed
55 0055 0 across the cluster.
56 0056 0
57 0057 0 V02-002 STJ0082 Steven T. Jeffreys, 02-Aug-1981

```



```

75 0074 1 GLOBAL ROUTINE LOGFILE_HANDLER (BUFFER_DESC) : NOVALUE =
76 0075 1
77 0076 1 !++
78 0077 1 | Functional description.
79 0078 1 |
80 0079 1 |     This routine is the handler for all LOGFILE messages received by OPCOM.
81 0080 1 |
82 0081 1 | Input:
83 0082 1 |
84 0083 1 |     BUFFER_DESC : The address of a quadword buffer descriptor that
85 0084 1 |     describes the buffer containing the message.
86 0085 1 |
87 0086 1 | Implicit Input:
88 0087 1 |
89 0088 1 |     None.
90 0089 1 |
91 0090 1 | Output:
92 0091 1 |
93 0092 1 |     None.
94 0093 1 |
95 0094 1 | Implicit output:
96 0095 1 |
97 0096 1 |     Some accounting data will be updated
98 0097 1 |     to reflect the receipt of the message.
99 0098 1 |
100 0099 1 |
101 0100 1 | Side effects:
102 0101 1 |
103 0102 1 |     None.
104 0103 1 |
105 0104 1 | Routine value:
106 0105 1 |
107 0106 1 |     None.
108 0107 1 | --
109 0108 1 |
110 0109 2 BEGIN                               ! Start of LOGFILE_HANDLER
111 0110 2
112 0111 2 MAP
113 0112 2     BUFFER_DESC      : $ref_bblock;
114 0113 2
115 0114 2 OWN
116 0115 2     DEFAULT_LOGFILE : $string_desc ('SYS$MANAGER:OPERATOR.LOG');
117 0116 2
118 0117 2 EXTERNAL
119 0118 2     LOGFILE_FAB      : $bblock,           ! System logfile FAB
120 0119 2     LOGFILE_NAM      : $bblock,           ! System logfile NAM
121 0120 2     LOGFILE_RAB      : $bblock,           ! System logfile RAB
122 0121 2     GLOBAL_STATUS   : BITVECTOR;
123 0122 2
124 0123 2 EXTERNAL ROUTINE
125 0124 2     CHECK_REQUEST,   ! Common sanity checks
126 0125 2     DEALLOCATE_RQCB : NOVALUE,           ! Dispose of an RQCB
127 0126 2     FIND_OPERATOR,   ! Find a given operator
128 0127 2     FORMAT_MESSAGE,  ! Format a message and build an MCB
129 0128 2     LOG_MESSAGE,     ! Write a message to a logfile
130 0129 2     NOTIFY_OPERATOR, ! Send a message to an operator
131 0130 2     NOTIFY_LISTED_OPERATORS, ! Send a msg to a list of operators

```

```

132 0131 2 VALID_OPERATOR; ! See if an operator device is valid
133 0132 3
134 0133 4 LOCAL
135 0134 5 MESSAGE_VECTOR : VECTOR [5, LONG], ! Message info
136 0135 6 FOUND : LONG, ! Boolean
137 0136 7 MESSAGE : LONG, ! Output message code
138 0137 8 OPER_RQCB : $ref_bblock, ! Known operator RQCB
139 0138 9 RQCB : $ref_bblock, ! Requestor operator RQCB
140 0139 10 DESC : VECTOR [2, LONG],
141 0140 11 STATUS : LONG;
142 0141 12
143 0142 13
144 0143 14
145 0144 15 ! If the request does not contain enough data,
146 0145 16 ! then it cannot be processed. Ignore it.
147 0146 17
148 0147 18 IF .BUFFER_DESC [DSC$W_LENGTH] LSS (OPC$K_LOGFILE_MIN_SIZE + OPC$K_COMHDRSIZ)
149 0148 19 THEN
150 0149 20 RETURN;
151 0150 21
152 0151 22 ! Call a special routine that does some sanity
153 0152 23 ! checking and startup code for all request handlers.
154 0153 24
155 0154 25 IF NOT (STATUS = CHECK_REQUEST (.BUFFER_DESC, RQCB))
156 0155 26 THEN
157 0156 27 RETURN;
158 0157 28
159 0158 29 ! Check the specified device and see if it is a
160 0159 30 ! valid operator device. This routine will also
161 0160 31 ! format the operator device name and create a
162 0161 32 ! device name descriptor within the RQCB.
163 0162 33
164 0163 34 IF NOT VALID_OPERATOR (.BUFFER_DESC, .RQCB)
165 0164 35 THEN
166 0165 36 BEGIN
167 0166 37 DEALLOCATE_RQCB (.RQCB);
168 0167 38 RETURN;
169 0168 39 END;
170 0169 40
171 0170 41 ! Determine if this is a known operator.
172 0171 42
173 0172 43 FOUND = FIND_OPERATOR (.RQCB, OPER_RQCB);
174 0173 44
175 0174 45 ! Do some sanity checking before granting the request.
176 0175 46 ! For the request to be granted, the following must be true:
177 0176 47 ! - The operator must be known to OPCOM.
178 0177 48 ! - The operator must have the proper privilege.
179 0178 49 ! - Both the INITLOG and CLOSELOG functions must not be requested.
180 0179 50
181 0180 51 IF (NOT .FOUND)
182 0181 52 OR ((.RQCB [RQCB_B_SCOPE] EQL OPC$K_SYSTEM) AND (NOT .Sbblock [RQCB [RQCB_L_PRIVMASK1], PRV$V_OPER]))
183 0182 53 OR ((.RQCB [RQCB_B_SCOPE] EQL OPC$K_GROUP) AND NOT ((.Sbblock [RQCB [RQCB_L_PRIVMASK1], PRV$V_OPER]) OR
184 0183 54 (.Sbblock [RQCB [RQCB_L_PRIVMASK1], PRV$V_GROUP])))
185 0184 55 OR ((.Sbblock [RQCB [RQCB_L_RQ_OPTIONS], OPC$V_INITLOG]) AND
186 0185 56 (.Sbblock [RQCB [RQCB_L_RQ_OPTIONS], OPC$V_CLOSELOG]))
187 0186 57 THEN
188 0187 58 BEGIN

```



```

: 246 0245 2 RQCB [HDR_V_BRD] = TRUE;
: 247 0246 2 RQCB [RQCB[_ATTN_MASK1]] = OPC$M_NM_CENTRL;
: 248 0247 2 RQCB [RQCB[_ATTN_MASK2]] = 0;
: 249 0248 2 NOTIFY_LISTED_OPERATORS (.RQCB);
: 250 0249 2 LOG_MESSAGE (.RQCB);
: 251 0250 2 DEALLOCATE_RQCB (.RQCB);
: 252 0251 2
: 253 0252 1 END;

```

! End of LOGFILE_HANDLER

```

.TITLE OPC$LOGFILE
.IDENT \V04-000\
.PSECT $PLI$,NOWRT,NOEXE,2

```

```

45 50 4F 3A 52 45 47 41 4E 41 4D 24 53 59 53 00000 P.AAA: .ASCII \SYS$MANAGER:OPERATOR.LOG\
47 4F 4C 2E 52 4F 54 41 52 0000F

```

.PSECT \$OWNS\$,NOEXE,2

```

0018 00000 DEFAULT_LOGFILE:
01 0E 00002 .WORD 24
00000000' 00004 .BYTE 14, 1
.ADDRESS P.AAA

```

```

.EXTRN LOGFILE_FAB, LOGFILE_NAM
.EXTRN LOGFILE_RAB, GLOBAL_STATUS
.EXTRN CHECK_REQUEST, DEALLOCATE_RQCB
.EXTRN FIND_OPERATOR, FORMAT_MESSAGE
.EXTRN LOG_MESSAGE, NOTIFY_OPERATOR
.EXTRN NOTIFY_LISTED_OPERATORS
.EXTRN VALID_OPERATOR, SYSS$CLOSE
.EXTRN SYSS$CREATE, SYSS$CONNECT

```

.PSECT \$CODE\$,NOWRT,2

```

007C 00000 .ENTRY LOGFILE_HANDLER, Save R2,R3,R4,R5,R6 : 0074
56 0000G CF 9E 00002 MOVAB LOGFILE_FAB, R6
55 0000G CF 9E 00007 MOVAB GLOBAL_STATUS, R5
54 0000G CF 9E 0000C MOVAB LOGFILE_NAM+3, R4
53 00000000G 00 9E 00011 MOVAB SYSS$CLOSE, R3
5E 24 C2 00018 SUBL2 #36, SP
0044 8F 04 BC B1 0001B CMPW @BUFFER_DESC, #68 : 0147
01 1E 00021 BGEQU 1$
04 04 00023 RET
5E DD 00024 1$: PUSHL SP : 0154
04 AC DD 00026 PUSHL BUFFER_DESC
0000G CF 02 FB 00029 CALLS #2, CHECK_REQUEST
01 50 E8 0002E BLBS STATUS, 2$
52 6E D0 00032 2$: MOVL RQCB, R2 : 0163
52 DD 00035 PUSHL R2
04 AC DD 00037 PUSHL BUFFER_DESC
0000G CF 02 FB 0003A CALLS #2, VACID_OPERATOR
4C 50 E9 0003F BLBC R0, 6$
04 AE 9F 00042 PUSHAB OPER_RQCB : 0172
52 DD 00045 PUSHL R2

```


	0000G	CF		02	FB	00047	CALLS	#2, FIND_OPERATOR	
		23		50	E9	0004C	BLBC	FOUND, 5\$	0180
		01	53	A2	91	0004F	CMPB	83(R2), #1	0181
				05	12	00053	BNEQ	3\$	
18	32	A2		02	E1	00055	BBC	#2, 50(R2), 5\$	
		02	53	A2	91	0005A	CMPB	83(R2), #2	0182
				09	12	0005E	BNEQ	4\$	
04	32	A2		02	E0	00060	BBS	#2, 50(R2), 4\$	
		09	31	A2	E9	00065	BLBC	49(R2), 5\$	0183
		79	58	A2	E9	00069	BLBC	88(R2), 8\$	0184
1F	58	A2		01	E1	0006D	BBC	#1, 88(R2), 7\$	0185
	10	AE	0005807C	8F	DO	00072	MOVL	#360572, MESSAGE_VECTOR	0188
			14	AE	D4	0007A	CLRL	MESSAGE_VECTOR+4	0189
			10	AE	9F	0007D	PUSHAB	MESSAGE_VECTOR	0190
				52	DD	00080	PUSHL	R2	
	0000G	CF		02	FB	00082	CALLS	#2, FORMAT_MESSAGE	
				52	DD	00087	PUSHL	R2	0191
	0000G	CF		01	FB	00089	CALLS	#1, NOTIFY_OPERATOR	
				00A0	31	0008E	BRW	10\$	0192
		51	58	A2	E9	00091	BLBC	88(R2), 8\$	0198
		65		04	88	00095	BISB2	#4, GLOBAL_STATUS	0204
	10	AE	00058034	8F	DO	00098	MOVL	#360500, MESSAGE_VECTOR	0205
			14	AE	D4	000A0	CLRL	MESSAGE_VECTOR+4	0206
	18	AE	7C	A2	9E	000A3	MOVAB	124(R2), MESSAGE_VECTOR+8	0207
	1C	AE	0000	CF	9E	000A8	MOVAB	DEFAULT_LOGFILE, MESSAGE_VECTOR+12	0208
				56	DD	000AE	PUSHL	R6	0209
		63		01	FB	000B0	CALLS	#1, SYSSCLOSE	
				56	DD	000B3	PUSHL	R6	0210
	00000000G	00		01	FB	000B5	CALLS	#1, SYSSCREATE	
		27		50	E9	000BC	BLBC	R0, 8\$	
			0000G	CF	9F	000BF	PUSHAB	LOGFILE_RAB	0212
	00000000G	00		01	FB	000C3	CALLS	#1, SYSSCONNECT	
		19		50	E9	000CA	BLBC	R0, 8\$	
		65		0C	8A	000CD	BICB2	#12, GLOBAL_STATUS	0216
	10	AE	00058009	8F	DO	000D0	MOVL	#360457, MESSAGE_VECTOR	0217
	08	AE		64	9A	000D8	MOVZBL	LOGFILE_NAM+3, DESC	0218
	0C	AE	01	A4	DO	000DC	MOVL	LOGFILE_NAM+4, DESC+4	0219
	1C	AE	08	AE	9E	000E1	MOVAB	DESC, MESSAGE_VECTOR+12	0220
26	58	A2		01	E1	000E6	BBC	#1, 88(R2), 9\$	0224
		65		08	88	000EB	BISB2	#8, GLOBAL_STATUS	0230
	10	AE	000580A4	8F	DO	000EE	MOVL	#360612, MESSAGE_VECTOR	0231
			14	AE	D4	000F6	CLRL	MESSAGE_VECTOR+4	0232
	18	AE	7C	A2	9E	000F9	MOVAB	124(R2), MESSAGE_VECTOR+8	0233
	08	AE		64	9A	000FE	MOVZBL	LOGFILE_NAM+3, DESC	0234
	0C	AE	01	A4	DO	00102	MOVL	LOGFILE_NAM+4, DESC+4	0235
	1C	AE	08	AE	9E	00107	MOVAB	DESC, MESSAGE_VECTOR+12	0236
				56	DD	0010C	PUSHL	R6	0237
		63		01	FB	0010E	CALLS	#1, SYSSCLOSE	
			10	AE	9F	00111	PUSHAB	MESSAGE_VECTOR	0244
				52	DD	00114	PUSHL	R2	
	0000G	CF		02	FB	00116	CALLS	#2, FORMAT_MESSAGE	
	28	A2		02	88	0011B	BISB2	#2, 40(R2)	0245
	5C	A2		01	7D	0011F	MOVQ	#1, 92(R2)	0246
				52	DD	00123	PUSHL	R2	0248
	0000G	CF		01	FB	00125	CALLS	#1, NOTIFY_LISTED_OPERATORS	
				52	DD	0012A	PUSHL	R2	0249
	0000G	CF		01	FB	0012C	CALLS	#1, LOG_MESSAGE	

0000G CF 52 DD 00131 10\$: PUSHL R2
01 FB 00133 CALLS #1, DEALLOCATE_RQCB
04 00138 RET

: 0250
:
: 0252

: Routine Size: 313 bytes, Routine Base: \$CODE\$ + 0000

: 254 0253 1
: 255 0254 1 END
: 256 0255 0 ELUDOM

! End of LOGFILE

PSECT SUMMARY

Name	Bytes	Attributes
\$SPLITS	24	NOVEC,NOWRT, RD,NOEXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)
\$OWNS	8	NOVEC, WRT, RD,NOEXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)
\$CODE\$	313	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	23 0	1000	00:01.8
_\$255\$DUA28:[OPCOM.OBJ]OPCOMLIB.L32;1	633	19 3	43	00:00.8

COMMAND QUALIFIERS

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

: Size: 313 code + 32 data bytes
: Run Time: 00:09.4
: Elapsed Time: 00:28.1
: Lines/CPU Min: 1636
: Lexemes/CPU-Min: 16414
: Memory Used: 122 pages
: Compilation Complete

0290

AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

This image shows a dense grid of approximately 150 small, overlapping window-like displays. Each display contains technical information, primarily in the form of lists and tables. The most prominent text within these displays includes:

- LOGFILE LIS** (top left)
- OPCOMDEF LIS** (top row, second from left)
- OPCOMDATA LIS** (top row, third from left)
- OPCOMINI LIS** (middle row, second from left)
- OPCOMLIB LIS** (middle row, third from left)
- OPCOMMAIN LIS** and **OPCOMOLD LIS** (middle row, fourth and fifth from left)
- OPCOMRPLY LIS** (middle row, sixth from left)
- OPCCrash LIS** (bottom row, first from left)
- OPCOMRST LIS** (bottom row, sixth from left)
- OPERUTIL LIS** (bottom right)

The displays also feature various symbols, including bar charts, small diagrams, and alphanumeric strings, all rendered in a monochrome, high-contrast style typical of early computer terminal output.