C 8

```
    000000000     PPPPPPPPPPPP        CCCCCCCCCCC     000000000     MMM         MMM
    000000000     PPPPPPPPPPPP        CCCCCCCCCCC     000000000     MMM         MMM
    000000000     PPPPPPPPPPPP        CCCCCCCCCCC     000000000     MMM         MMM
000         000   PPP         PPP  CCC            000         000   MMMMMM   MMMMMM
000         000   PPP         PPP  CCC            000         000   MMMMMM   MMMMMM
000         000   PPP         PPP  CCC            000         000   MMMMMM   MMMMMM
000         000   PPP         PPP  CCC            000         000   MMM   MMM   MMM
000         000   PPP         PPP  CCC            000         000   MMM   MMM   MMM
000         000   PPP         PPP  CCC            000         000   MMM   MMM   MMM
000         000   PPPPPPPPPPPP     CCC            000         000   MMM         MMM
000         000   PPPPPPPPPPPP     CCC            000         000   MMM         MMM
000         000   PPPPPPPPPPPP     CCC            000         000   MMM         MMM
000         000   PPP              CCC            000         000   MMM         MMM
000         000   PPP              CCC            000         000   MMM         MMM
000         000   PPP              CCC            000         000   MMM         MMM
000         000   PPP              CCC            000         000   MMM         MMM
000         000   PPP              CCC            000         000   MMM         MMM
000         000   PPP              CCC            000         000   MMM         MMM
    000000000     PPP                 CCCCCCCCCCC     000000000     MMM         MMM
    000000000     PPP                 CCCCCCCCCCC     000000000     MMM         MMM
    000000000     PPP                 CCCCCCCCCCC     000000000     MMM         MMM
```

```
_$2


Sym
---
ALL


ASC


BOD
BOD
BOD
BOD
BOD
BOD
BOD
BOD
BUG
BYP
CAN
CAN
CAN
CHE
CHE


CLU
CLU
CLU
CLU
CLU
CLU
CLU

CLU
CLU
CLU
CLU
CLU
CLU
CLU


CLU
CLU
```

```
**FILE**ID**LOGEVENT                                        G 16
```

```
LL            OOOOOO      GGGGGGGG  EEEEEEEEEE  VV      VV  EEEEEEEEEE  NN      NN  TTTTTTTTTT
LL            OOOOOO      GGGGGGGG  EEEEEEEEEE  VV      VV  EEEEEEEEEE  NN      NN  TTTTTTTTTT
LL          OO      OO  GG          EE          VV      VV  EE          NN      NN      TT
LL          OO      OO  GG          EE          VV      VV  EE          NN      NN      TT
LL          OO      OO  GG          EE          VV      VV  EE          NNNN    NN      TT
LL          OO      OO  GG          EE          VV      VV  EE          NNNN    NN      TT
LL          OO      OO  GG          EEEEEEEE    VV      VV  EEEEEEEE    NN  NN  NN      TT
LL          OO      OO  GG          EEEEEEEE    VV      VV  EEEEEEEE    NN  NN  NN      TT
LL          OO      OO  GG  GGGGGG  EE          VV      VV  EE          NN    NNNN      TT
LL          OO      OO  GG  GGGGGG  EE          VV      VV  EE          NN    NNNN      TT
LL          OO      OO  GG      GG  EE            VV  VV    EE          NN      NN      TT          ....
LL          OO      OO  GG      GG  EE            VV  VV    EE          NN      NN      TT          ....
LLLLLLLLLL    OOOOOO      GGGGGG    EEEEEEEEEE      VV      EEEEEEEEEE  NN      NN      TT          ....
LLLLLLLLLL    OOOOOO      GGGGGG    EEEEEEEEEE      VV      EEEEEEEEEE  NN      NN      TT          ....

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
```

```
0 MODULE  OPC$LOGEVENT    (
0                           LANGUAGE (BLISS32),
0                           IDENT = 'V04-000'
0                          ) =
0
0 !****************************************************************************
0 !*                                                                          *
0 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
0 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
0 !*  ALL RIGHTS RESERVED.                                                    *
0 !*                                                                          *
0 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
0 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
0 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
0 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
0 !*  TRANSFERRED.                                                            *
0 !*                                                                          *
0 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
0 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0 !*  CORPORATION.                                                            *
0 !*                                                                          *
0 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
0 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
0 !*                                                                          *
0 !*                                                                          *
0 !****************************************************************************
0
0 !++
0 ! FACILITY:
0 !
0 !       OPCOM
0 !
0 ! ABSTRACT:
0 !
0 !       This module contains all the various and sundry general
0 !       purpose utility routines used by OPCOM's request handlers.
0 !
0 ! Environment:
0 !
0 !       VAX/VMS operating system.
0 !
0 ! Author:
0 !
0 !       Steven T. Jeffreys
0 !
0 ! Creation date:
0 !
0 !       March 10, 1981
0 !
0 ! Revision history:
0 !
0 !       V03-005 CWH3169         CW Hobbs                        5-May-1984
0 !               Second pass for cluster-wide OPCOM:
0 !                - Add an explanation to DUMP_LOG_FILE and WRITE_LOG_FILE
0 !                  messages so that users won't bother us unless something
0 !                  is really wrong.
```

```
  0 !
  0 !        V03-004 RSH0117        R. Scott Hanna                   14-Mar-1983
  0 !                LOG_MESSAGE / Increase local buffer size to OPC$K_MAXMESSAGE.
  0 !
  0 !        V03-003 CWH3003        CW Hobbs                         16-Sep-1983
  0 !                Increase size of local buffer, print blank line as separate
  0 !                records instead of using <CR><LF>.
  0 !
  0 !        V03-002 CWH3001        CW Hobbs                         30-Jul-1983
  0 !                Various and sundry things to make OPCOM distributed
  0 !                across the cluster.
  0 !
  0 ! V03-001        STJ3032        Steven T. Jeffreys,     05-Oct-1982
  0 !                Set GBLSTS_K_FLUSH_PENDIND when writing to the logfile.
  0 !
  0 ! V02-002        STJ0160        Steven T. Jeffreys,     08-Feb-1982
  0 !                Jiggle the message size and pointer so that the 'bell'
  0 !                character on front of each message is not sent to the logfile.
  0 !
  0 !--
  0
  1 BEGIN                                                  ! Start of LOGEVENT
  1
  1 LIBRARY 'SYS$LIBRARY:LIB.L32';
  1 LIBRARY 'LIB$:OPCOMLIB';
  1
  1 FORWARD ROUTINE
  1         DUMP_LOG_FILE,                      ! Print a formatted dump of a buffer
  1         LOG_MESSAGE,                        ! Write a message to a log file
  1         LOG_MESSAGE_PUT,                    ! $PUT a single record of a message
  1         LOGEVENT_FAO_BUFFER,                ! Local fao buffer routine
  1         WRITE_LOG_FILE;                     ! Write a string desc to a log file
  1
  1 BIND
  1         EXP1 = %ASCID 'OPCOM has noticed a condition which might be due to an internal error.',
  1         EXP2 = %ASCID 'It might also be explained by normal events, especially if nodes have just',
  1         EXP3 = %ASCID 'crashed or rebooted in a VAXcluster.  Please bring this message to Digital''s',
  1         EXP4 = %ASCID 'attention only if you are having problems with operator communications.';
  1
  1 MACRO
M 1         PUT (DESC) =    BEGIN
M 1                         BIND DSC = (DESC) : $bblock;
M 1                         LOGFILE_RAB [RAB$W_RSZ] = .DSC [DSC$W_LENGTH];
M 1                         LOGFILE_RAB [RAB$L_RBF] = .DSC [DSC$A_POINTER];
M 1                         $PUT (RAB = LOGFILE_RAB)                       ! Status is value of the block
  1                         END %,
M 1         PUT_EXPLANATION =
M 1                         BEGIN PUT (EXP1); PUT (EXP2); PUT (EXP3); PUT (EXP4); END
  1                         %;
```

```
1 GLOBAL ROUTINE DUMP_LOG_FILE (BUFF_DESC : $ref_bblock, ID_DESC : $ref_bblock) =
1
1 !++
1 ! Functional description:
1 !
1 !       This routine will write a formatted hex dump to the operator log file.
1 ! **
1 ! ** This routine will be enhanced later to support mutliple log files.
1 ! **
1 !
1 ! Input:
1 !
1 !       BUFF_DESC       : Address of a string desc for buffer to be dumped
1 !       ID_DESC         : Explanatory text for dump
1 !
1 ! Implicit Input:
1 !
1 !       None.
1 !
1 ! Output:
1 !
1 !       None.
1 !
1 ! Implict output:
1 !
1 !       None.
1 !
1 ! Side effects:
1 !
1 !       None.
1 !
1 ! Routine value:
1 !
1 !       TRUE            : If success
1 !       <anything else> : If the log attempt failed
1 !--
1
2 BEGIN                                           ! Start of DUMP_LOG_FILE
2
2 EXTERNAL
2       GLOBAL_STATUS   : BITVECTOR,            ! OPCOM global status flags
2       LOGFILE_RAB     : $bblock,              ! RMS control structure
2       LOGFILE_FAB     : $bblock;              ! RMS control structure
2
2 LOCAL
2       BASE,
2       LEFT,
2       PTR,
2       LCL_DESC : $ref_bblock,
2       INTER : VECTOR [8, LONG],
2       STATUS          : LONG;
2 !
2 ! See if logging is enabled.
2 !
3 IF  (NOT .GLOBAL_STATUS [GBLSTS_K_LOGGING_ENABLED])
3 OR  (.GLOBAL_STATUS [GBLSTS_K_LOGFILE_CLOSED])
2 THEN
```

```
;  165          0164  2     RETURN (TRUE);
;  166          0165  2  !
;  167          0166  2  ! Format and print the message header
;  168          0167  2  !
;  169          0168  2  INTER [0] = 0;                        ! A blank line
;  170          0169  2  PUT (INTER);
;  171          0170  2  PUT (LOGEVENT_FAO_BUFFER (%ASCID '%%%%%%%%%%%  OPCOM  !%D  %%%%%%%%%%%  DUMP_LOG_FILE', 0));
;  172          0171  2  PUT_EXPLANATION;
;  173        P 0172  2  PUT (LOGEVENT_FAO_BUFFER (%ASCID 'Buffer is !5UL (%X!XW) bytes -- "!AS"',
;  174          0173  2                          .BUFF_DESC [DSC$W_LENGTH], .BUFF_DESC [DSC$W_LENGTH], .ID_DESC));
;  175          0174  2  !
;  176          0175  2  ! Format the buffer, 32 bytes at a time
;  177          0176  2  !
;  178          0177  2  LEFT = .BUFF_DESC [DSC$W_LENGTH];
;  179          0178  2  PTR = .BUFF_DESC [DSC$A_POINTER];
;  180          0179  2  BASE = 0;
;  181          0180  2  WHILE .LEFT GTR 0
;  182          0181  2  DO
;  183          0182  3      BEGIN
;  184          0183  3      !
;  185          0184  3      ! Move the next chunk of data to the intermediate buffer
;  186          0185  3      !
;  187          0186  3      CH$COPY (MINU (.LEFT, 32), .PTR, 0, 32, INTER [0]);
;  188        P 0187  3      PUT (LOGEVENT_FAO_BUFFER (%ASCID '!8(9XL) !32AF !XW',
;  189        P 0188  3                              .INTER [7], .INTER [6], .INTER [5], .INTER [4],
;  190        P 0189  3                              .INTER [3], .INTER [2], .INTER [1], .INTER [0],
;  191          0190  3                              32, INTER [0], .BASE));
;  192          0191  3      !
;  193          0192  3      ! Move to the next chunk
;  194          0193  3      !
;  195          0194  3      BASE = .BASE + 32;
;  196          0195  3      PTR  = .PTR  + 32;
;  197          0196  3      LEFT = .LEFT - 32;
;  198          0197  3      END;
;  199          0198  2
;  200          0199  2  $FLUSH (RAB = LOGFILE_RAB);
;  201          0200  2
;  202          0201  2  RETURN (TRUE);
;  203          0202  1  END;                                             ! End of DUMP_LOG_FILE


                                                                           .TITLE  OPC$LOGEVENT
                                                                           .IDENT  \V04-000\

                                                                           .PSECT  $PLIT$,NOWRT,NOEXE,2

63 69 74 6F 6E 20 73 61 68 20 4D 4F 43 50 4F  00000 P.AAB:  .ASCII  \OPCOM has noticed a condition which migh\  ;
20 6E 6F 69 74 69 64 6E 6F 63 20 61 20 64 65  0000F                                                            :
               68 67 69 6D 20 68 63 69 68 77  0001E                                                            :
20 6E 61 20 6F 74 20 65 75 64 20 65 62 20 74  00028         .ASCII  \t be due to an internal error.\<0><0>     :
2E 72 6F 72 72 65 20 6C 61 6E 72 65 74 6E 69  00037                                                            :
                                       00 00  00046                                                            :
                                    010E0046  00048 P.AAA:  .LONG   17694790                                   :
                                   00000000'  0004C         .ADDRESS P.AAB                                     :
62 20 6F 73 6C 61 20 74 68 67 69 6D 20 74 49  00050 P.AAD:  .ASCII  \It might also be explained by normal eve\  ;
20 79 62 20 64 65 6E 69 61 6C 70 78 65 20 65  0005F                                                            :
               65 76 65 20 6C 61 6D 72 6F 6E  0006E                                                            ;
```

```
79 6C 6C 61 69 63 65 70 73 65 20 2C 73 74 6E 00078          .ASCII  \nts, especially if nodes have just\<0>          ;
20 65 76 61 68 20 73 65 64 6F 6E 20 66 69 20 00087                                                                  ;
                              00 74 73 75 6A 00096                                                                  ;
                                          00 0009B          .ASCII  <0>                                             ;
                                   010E004A 0009C P.AAC:    .LONG   17694794                                        ;
                                  00000000' 000A0           .ADDRESS P.AAD                                          ;
6F 62 65 72 20 72 6F 20 64 65 68 73 61 72 63 000A4 P.AAF:   .ASCII  \crashed or rebooted in a VAXcluster.  Pl\      ;
6C 63 58 41 56 20 61 20 6E 69 20 64 65 74 6F 000B3                                                                  ;
               6C 50 20 20 2E 72 65 74 73 75 000C2                                                                  ;
73 69 68 74 20 67 6E 69 72 62 20 65 73 61 65 000CC          .ASCII  \ease bring this message to Digital's\          ;
67 69 44 20 6F 74 20 65 67 61 73 73 65 6D 20 000DB                                                                  ;
                           73 27 6C 61 74 69 000EA                                                                  ;
                                   010E004C 000F0 P.AAE:    .LONG   17694796                                        ;
                                  00000000' 000F4           .ADDRESS P.AAF                                          ;
20 79 6C 6E 6F 20 6E 6F 69 74 6E 65 74 74 61 000F8 P.AAH:   .ASCII  \attention only if you are having problem\      ;
69 76 61 68 20 65 72 61 20 75 6F 79 20 66 69 00107                                                                  ;
               6D 65 6C 62 6F 72 7C 20 67 6E 00116                                                                  ;
72 6F 74 61 72 65 70 6F 20 68 74 69 77 20 73 00120          .ASCII  \- with operator communications.\<0>            ;
73 6E 6F 69 74 61 63 69 6E 75 6D 6D 6F 63 20 0012F                                                                  ;
                                       00 2E 0013E                                                                  ;
                                   010E0047 00140 P.AAG:    .LONG   17694791                                        ;
                                  00000000' 00144           .ADDRESS P.AAH                                          ;
50 4F 20 20 25 25 25 25 25 25 25 25 25 25 25 00148 P.AAJ:   .ASCII  \%%%%%%%%%%% OPCOM !%D  %%%%%%%%%%%  DU\        ;
25 25 25 25 25 20 20 44 25 21 20 20 4D 4F 43 00157                                                                  ;
               55 44 20 20 25 25 25 25 25 25 00166                                                                  ;
         00 45 4C 49 46 5F 47 4F 4C 5F 50 4D 00170          .ASCII  \MP_LOG_FILE\<0>                                ;
                                   010E0033 0017C P.AAI:    .LONG   17694771                                        ;
                                  00000000' 00180           .ADDRESS P.AAJ                                          ;
20 4C 55 35 21 20 73 69 20 72 65 66 66 75 42 00184 P.AAL:   .ASCII  \Buffer is !5UL (%X!XW) bytes -- "!AS"\<0>      ;
2D 20 73 65 74 79 62 20 29 57 58 21 58 25 28 00193                                                                  ;
                        00 22 53 41 21 22 20 2D 001A2                                                               ;
                                       00 00 001AA          .ASCII  <0><0>                                          ;
                                   010E0025 001AC P.AAK:    .LONG   17694757                                        ;
                                  00000000' 001B0           .ADDRESS P.AAL                                          ;
21 20 46 41 32 33 21 20 29 4C 58 39 28 38 21 001B4 P.AAN:   .ASCII  \!8(9XL) !32AF !XW\<0><0><0>                    ;
                              00 00 00 57 58 001C3                                                                  ;
                                   010E0011 001C8 P.AAM:    .LONG   17694737                                        ;
                                  00000000' 001CC           .ADDRESS P.AAN                                          ;

                                                  EXP1=                 P.AAA
                                                  EXP2=                 P.AAC
                                                  EXP3=                 P.AAE
                                                  EXP4=                 P.AAG
                                                  DSC=                  P.AAA
                                                  DSC=                  P.AAC
                                                  DSC=                  P.AAE
                                                  DSC=                  P.AAG
                                                            .EXTRN  GLOBAL_STATUS, LOGFILE_RAB
                                                            .EXTRN  LOGFILE_FAB, SYS$PUT
                                                            .EXTRN  SYS$FLUSH

                                                            .PSECT  $CODE$,NOWRT,2

                                       0FFC 00000           .ENTRY  DUMP_LOG_FILE, Save R2,R3,R4,R5,R6,R7,R8,-   ; 0107
                                                                    R9,R10,R11
                       5B     0000'  CF  9E 00002           MOVAB   P.AAI, R11                                      ;
                       5A 00000000G  00  9E 00007           MOVAB   SYS$PUT, R10                                    ;
```

```
                            59    0000G  CF   9E 0000E         MOVAB    LOGFILE_RAB, R9                          ;
                            5E           20   C2 00013         SUBL2    #32, SP                                  ;
                 03  0000G  CF           01   E0 00016         BBS      #1, GLOBAL_STATUS, 2$                    ; 016
                                       00F9   31 0001C 1$:     BRW      6$                                       ;
                 F7  0000G  CF           03   E0 0001F 2$:     BBS      #3, GLOBAL_STATUS, 1$                    ; 0162
                                         6E   D4 00025         CLRL     INTER                                    ; 0168
                       22   A9           6E   B0 00027         MOVW     DSC, LOGFILE_RAB+34                      ; 0169
                       28   A9     04    AE   D0 0002B         MOVL     DSC+4, LOGFILE_RAB+40                    ;
                                         59   DD 00030         PUSHL    R9                                       ;
                            6A           01   FB 00032         CALLS    #1, SYS$PUT                              ;
                                         7E   D4 00035         CLRL     -(SP)                                    ; 0170
                                         5B   DD 00037         PUSHL    R11                                      ;
                    0000V   CF           02   FB 00039         CALLS    #2, LOGEVENT_FAO_BUFFER                  ;
                       22   A9           60   B0 0003E         MOVW     (R0), LOGFILE_RAB+34                     ;
                       28   A9     04    A0   D0 00042         MOVL     4(R0), LOGFILE_RAB+40                    ;
                                         59   DD 00047         PUSHL    R9                                       ;
                            6A           01   FB 00049         CALLS    #1, SYS$PUT                              ;
                       22   A9   FECC    CB   B0 0004C         MOVW     DSC, LOGFILE_RAB+34                      ;
                       28   A9   FED0    CB   D0 00052         MOVL     DSC+4, LOGFILE_RAB+40                    ;
                                         59   DD 00058         PUSHL    R9                                       ;
                            6A           01   FB 0005A         CALLS    #1, SYS$PUT                              ;
                       22   A9   FF20    CB   B0 0005D         MOVW     DSC, LOGFILE_RAB+34                      ;
                       28   A9   FF24    CB   D0 00063         MOVL     DSC+4, LOGFILE_RAB+40                    ;
                                         59   DD 00069         PUSHL    R9                                       ;
                            6A           01   FB 0006B         CALLS    #1, SYS$PUT                              ;
                       22   A9   FF74    CB   B0 0006E         MOVW     DSC, LOGFILE_RAB+34                      ;
                       28   A9   FF78    CB   D0 00074         MOVL     DSC+4, LOGFILE_RAB+40                    ;
                                         59   DD 0007A         PUSHL    R9                                       ;
                            6A           01   FB 0007C         CALLS    #1, SYS$PUT                              ;
                       22   A9     C4    AB   B0 0007F         MOVW     DSC, LOGFILE_RAB+34                      ;
                       28   A9     C8    AB   D0 00084         MOVL     DSC+4, LOGFILE_RAB+40                    ;
                                         59   DD 00089         PUSHL    R9                                       ;
                            6A           01   FB 0008B         CALLS    #1, SYS$PUT                              ;
                                   08    AC   DD 0008E         PUSHL    ID_DESC                                  ; 0173
                            52     04    AC   D0 00091         MOVL     BUFF_DESC, R2                            ;
                            7E           62   3C 00095         MOVZWL   (R2), -(SP)                              ;
                            7E           62   3C 00098         MOVZWL   (R2), -(SP)                              ;
                                   30    AB   9F 0009B         PUSHAB   P.AAK                                    ;
                    0000V   CF           04   FB 0009E         CALLS    #4, LOGEVENT_FAO_BUFFER                  ;
                       22   A9           60   B0 000A3         MOVW     (R0), LOGFILE_RA  34                     ;
                       28   A9     04    A0   D0 000A7         MOVL     4(R0), LOGFILE_RAB+40                    ;
                                         59   DD 000AC         PUSHL    R9                                       ;
                            6A           01   FB 000AE         CALLS    #1, SYS$PUT                              ;
                            56           62   3C 000B1         MOVZWL   (R2), LEFT                               ; 0177
                            57     04    A2   D0 000B4         MOVL     4(R2), PTR                               ; 0178
                                         58   D4 000B8         CLRL     BASE                                     ; 0179
                                         56   D5 000BA 3$:     TSTL     LEFT                                     ; 0180
                                         51   15 000BC         BLEQ     5$                                       ;
                            50           56   D0 000BE         MOVL     LEFT, R0                                 ; 0186
                            20           50   D1 000C1         CMPL     R0, #32                                  ;
                                         03   1B 000C4         BLEQU    4$                                       ;
                            50           20   D0 000C6         MOVL     #32, R0                                  ;
     20          00         67           50   2C 000C9 4$:     MOVC5    R0, (PTR), #0, #32, INTER                ;
                                         6E      000CE                                                           ;
                                         58   DD 000CF         PUSHL    BASE                                     ; 0190
                                   04    AE   9F 000D1         PUSHAB   INTER                                    ;
                                         20   DD 000D4         PUSHL    #32                                      ;
                                   0C    AE   DD 000D6          PUSHL    INTER                            ;
                                   14    AE   DD 000D9          PUSHL    INTER+4                          ;
                                   1C    AE   DD 000DC          PUSHL    INTER+8                          ;
                                   24    AE   DD 000DF          PUSHL    INTER+12                         ;
                                   2C    AE   DD 000E2          PUSHL    INTER+16                         ;
                                   34    AE   DD 000E5          PUSHL    INTER+20                         ;
                                   3C    AE   DD 000E8          PUSHL    INTER+24                         ;
                                   44    AE   DD 000EB          PUSHL    INTER+28                         ;
                                   4C    AB   9F 000EE          PUSHAB   P.AAM                            ;
                  0000V   CF             0C   FB 000F1          CALLS    #12, LOGEVENT_FAO_BUFFER         ;
                     22   A9             60   B0 000F6          MOVW     (R0), LOGFILE_RAB+34             ;
                     28   A9       04    A0   D0 000FA          MOVL     4(R0), LOGFILE_RAB+40            ;
                                         59   DD 000FF          PUSHL    R9                               ;
                          6A             01   FB 00101          CALLS    #1, SYS$PUT                      ;
                          58             20   C0 00104          ADDL2    #32, BASE                        ; 0194
                          57             20   C0 00107          ADDL2    #32, PTR                         ; 0195
                          56             20   C2 0010A          SUBL2    #32, LEFT                        ; 0196
                                         AB   11 0010D          BRB      3$                               ; 0180
                                         59   DD 0010F 5$:      PUSHL    R9                               ; 0199
              00000000G   00             01   FB 00111          CALLS    #1, SYS$FLUSH                    ;
                          50             01   D0 00118 6$:      MOVL     #1, R0                           ; 0201
                                              04 0011B          RET                                       ; 0202

; Routine Size:  284 bytes,    Routine Base:  $CODE$ + 0000
```

```
1 GLOBAL ROUTINE LOG_MESSAGE (RQCB) =
1
1 !++
1 ! Functional description:
1 !
1 !       This routine will write a message described by an MCB
1 !       to the operator log file.
1 ! **
1 ! ** This routine will be enhanced later to support mutliple log files.
1 ! **
1 !
1 ! Input:
1 !
1 !       RQCB            : Address of an RQCB data structure
1 !
1 ! Implicit Input:
1 !
1 !       RQCB [RQCB_L_MCB] points to a valid MCB.
1 !
1 ! Output:
1 !
1 !       None.
1 !
1 ! Implict output:
1 !
1 !       None.
1 !
1 ! Side effects:
1 !
1 !       None.
1 !
1 ! Routine value:
1 !
1 !       TRUE            : If success
1 !       <anything else> : If the log attempt failed
1 !--
1
2 BEGIN                                         ! Start of LOG_MESSAGE
2
2 MAP
2       RQCB            : $ref_bblock;
2
2 EXTERNAL LITERAL
2       MCB_K_TYPE;                             ! MCB structure type
2
2 EXTERNAL
2       GLOBAL_STATUS   : BITVECTOR;            ! OPCOM global status flags
2
2 LOCAL
2       ADR             : REF VECTOR [, BYTE],  ! Adjusted address of string
2       LEN             : LONG,                 ! Adjusted length of string
2       RECLEN          : LONG,                 ! Adjusted length of single record
2       CHAR            : BYTE,
2       BUF             : VECTOR [OPC$K_MAXMESSAGE, BYTE],
2       BUFP            : REF VECTOR [, BYTE],
2       MCB             : $ref_bblock,          ! MCB data structure
2       STATUS          : LONG;
```

```
;  262        0260  2
;  263        0261  2 !
;  264        0262  2 ! Check for a valid MCB.
;  265        0263  2 !
;  266        0264  2 MCB = .RQCB [RQCB_L_MCB];
;  267        0265  3 IF (.MCB EQL 0) OR (.MCB [MCB_B_TYPE] NEQ MCB_K_TYPE)
;  268        0266  2 THEN
;  269        0267  2     RETURN (FALSE);
;  270        0268  2 !
;  271        0269  2 ! See if logging is enabled.
;  272        0270  2 !
;  273        0271  3 IF  (NOT .GLOBAL_STATUS [GBLSTS_K_LOGGING_ENABLED])
;  274        0272  3 OR  (.GLOBAL_STATUS [GBLSTS_K_LOGFILE_CLOSED])
;  275        0273  3 OR  ((.$bblock [RQCB [RQCB_L_OPTIONS], OPC$V_NOLOG]) AND
;  276        0274  3      (.$bblock [RQCB [RQCB_L_PRIVMASK1], PRV$V_OPER]))
;  277        0275  2 THEN
;  278        0276  2     RETURN (TRUE);
;  279        0277  2 !
;  280        0278  2 ! Adjust the string.  Remove any trailing control and space characters
;  281        0279  2 !
;  282        0280  2 LEN = .MCB [MCB_L_TEXTLEN];
;  283        0281  2 ADR = .MCB [MCB_L_TEXTPTR];
;  284        0282  2 IF .LEN LEQ 0
;  285        0283  2 THEN
;  286        0284  2     RETURN (TRUE);
;  287        0285  2 DECR I FROM .LEN-1 TO 0                       ! Remove all trailing control and space characters
;  288        0286  2 DO
;  289        0287  2     SELECTONE .ADR [.I] OF
;  290        0288  2     SET
;  291        0289  2         [0 TO 32] :     LEN = .LEN - 1;         ! Bad one, remove it
;  292        0290  2         [OTHERWISE] :   EXITLOOP;               ! Good one, we are done looking
;  293        0291  2     TES;
;  294        0292  2 IF .LEN LEQ 0                                 ! If we got rid of the whole string, we are done
;  295        0293  2 THEN
;  296        0294  2     RETURN (TRUE);
;  297        0295  2 !
;  298        0296  2 ! Now get ready to filter the string.  We will copy it to a local buffer, making some changes.
;  299        0297  2 ! Ignore all control characters except tab and line-feed.  Pass tab through, if see a line-feed
;  300        0298  2 ! then write the record found up to the line-feed.  Do not print zero-length lines.
;  301        0299  2 !
;  302        0300  2 STATUS = LOG_MESSAGE_PUT (0, BUF);            ! Put a single blank line before the message
;  303        0301  2 RECLEN = 0;
;  304        0302  2 BUFP = BUF;
;  305        0303  2 WHILE .LEN GTR 0
;  306        0304  2 DO
;  307        0305  3     BEGIN
;  308        0306  3     CHAR = CH$RCHAR_A (ADR);
;  309        0307  3     SELECTONE .CHAR OF
;  310        0308  3     SET
;  311        0309  3         !
;  312        0310  3         ! Line feed, print the old record and start a new one
;  313        0311  3         !
;  314        0312  4         [10] :                  BEGIN
;  315        0313  4                                 IF .RECLEN GTR 0
;  316        0314  4                                 THEN
;  317        0315  4                                     STATUS = LOG_MESSAGE_PUT (.RECLEN, BUF);
;  318        0316  4                                 RECLEN = 0;
;  319          0317  4                                  BUFP = BUF;
;  320          0318  3                                  END;
;  321          0319  3              !
;  322          0320  3              ! Misc control char, ignore
;  323          0321  3              !
;  324          0322  3              [0 TO 8, 11 TO 31] :      ;
;  325          0323  3              !
;  326          0324  3              ! Valid char, pass it to the new buffer
;  327          0325  3              !
;  328          0326  4              [OTHERWISE] :               BEGIN
;  329          0327  4                                          RECLEN = .RECLEN + 1;
;  330          0328  4                                          CH$WCHAR_A (.CHAR, BUFP);
;  331          0329  3                                          END;
;  332          0330  3          TES;
;  333          0331  3          LEN = .LEN - 1;
;  334          0332  2          END;
;  335          0333  2   !
;  336          0334  2   ! If we have some data in the new buffer, print it
;  337          0335  2   !
;  338          0336  2   IF .RECLEN GTR 0
;  339          0337  2   THEN
;  340          0338  2       STATUS = LOG_MESSAGE_PUT (.RECLEN, BUF);
;  341          0339  2
;  342          0340  2   RETURN (.STATUS);
;  343          0341  1   END;                                                  ! End of LOG_MESSAGE


                                                                     .EXTRN  MCB_K_TYPE

                                                  00FC 00000         .ENTRY  LOG_MESSAGE, Save R2,R3,R4,R5,R6,R7        ; 0203
                                 57      0000V  CF  9E 00002         MOVAB   LOG_MESSAGE_PUT, R7                        ;
                                 5E      F800   CE  9E 00007         MOVAB   -2048(SP), SP                              ;
                                 51        04   AC  D0 0000C         MOVL    RQCB, R1                                   ; 0264
                                 50        6C   A1  D0 00010         MOVL    108(R1), MCB                               ;
                                                03  12 00014         BNEQ    2$                                         ; 0265
                                              0096  31 00016 1$:     BRW     15$                                        ;
00000000G  8F       0A   A0      08             00  ED 00019 2$:     CMPZV   #0, #8, 10(MCB), #MCB_K_TYPE               ;
                                                F1  12 00023         BNEQ    1$                                         ;
                      32       0000G  CF        01  E1 00025         BBC     #1, GLOBAL_STATUS, 7$                      ; 0271
                      2C       0000G  CF        03  E0 0002B         BBS     #3, GLOBAL_STATUS, 7$                      ; 0272
                      05          54  A1        E9 00031             BLBC    84(R1), 3$                                 ; 0273
                      23          32  A1        02  E0 00035         BBS     #2, 50(R1), 7$                             ; 0274
                                 52        30   A0  D0 0003A 3$:     MOVL    48(MCB), LEN                               ; 0280
                                 56        34   A0  D0 0003E         MOVL    52(MCB), ADR                               ; 0281
                                                52  D5 00042         TSTL    LEN                                        ; 0282
                                                17  15 00044         BLEQ    7$                                         ;
                                 50             52  D0 00046         MOVL    LEN, I                                     ; 0287
                                                0B  11 00049         BRB     5$                                         ;
                                 51           6046  9A 0004B 4$:     MOVZBL  (I)[ADR], R1                               ;
                                 20             51  91 0004F         CMPB    R1, #32                                    ; 0289
                                                05  1A 00052         BGTRU   6$                                         ;
                                                52  D7 00054         DECL    LEN                                        ;
                                 F2             50  F4 00056 5$:     SOBGEQ  I, 4$                                      ; 0287
                                                52  D5 00059 6$:     TSTL    LEN                                        ; 0292
                                                04  14 0005B         BGTR    8$                                         ;
                                 50             01  D0 0005D 7$:     MOVL    #1, R0                                     ; 0294
```

```
                                     04 00060         RET                                  ;
                                 5E  DD 00061  8$:    PUSHL     SP                         ; 0300
                                 7E  D4 00063         CLRL      -(SP)                      ;
                  67             02  FB 00065         CALLS     #2, LOG_MESSAGE_PUT        ;
                                 53  D4 00068         CLRL      RECLEN                     ; 0301
                  55             6E  9E 0006A         MOVAB     BUF, BUFP                  ; 0302
                                 52  D5 0006D  9$:    TSTL      LEN                        ; 0303
                                 32  15 0006F         BLEQ      14$                        ;
                  54             86  90 00071         MOVB      (ADR)+, CHAR               ; 0306
                  0A             54  91 00074         CMPB      CHAR, #10                  ; 0312
                                 12  12 00077         BNEQ      11$                        ;
                                 53  D5 00079         TSTL      RECLEN                     ; 0313
                                 07  15 0007B         BLEQ      10$                        ;
                        4008     8F  BB 0007D         PUSHR     #^M<R3,SP>                 ; 0315
                  67             02  FB 00081         CALLS     #2, LOG_MESSAGE_PUT        ;
                                 53  D4 00084 10$:    CLRL      RECLEN                     ; 0316
                  55             6E  9E 00086         MOVAB     BUF, BUFP                  ; 0317
                                 14  11 00089         BRB       13$                        ; 0307
                  08             54  91 0008B 11$:    CMPB      CHAR, #8                   ; 0322
                                 0F  1B 0008E         BLEQU     13$                        ;
                  0B             54  91 00090         CMPB      CHAR, #11                  ;
                                 05  1F 00093         BLSSU     12$                        ;
                  1F             54  91 00095         CMPB      CHAR, #31                  ;
                                 05  1B 00098         BLEQU     13$                        ;
                                 53  D6 0009A 12$:    INCL      RECLEN                     ; 0327
                  85             54  90 0009C         MOVB      CHAR, (BUFP)+              ; 0328
                                 52  D7 0009F 13$:    DECL      LEN                        ; 0331
                                 CA  11 000A1         BRB       9$                         ; 0303
                                 53  D5 000A3 14$:    TSTL      RECLEN                     ; 0336
                                 0A  15 000A5         BLEQ      16$                        ;
                        4008     8F  BB 000A7         PUSHR     #^M<R3,SP>                 ; 0338
                  67             02  FB 000AB         CALLS     #2, LOG_MESSAGE_PUT        ;
                                     04 000AE         RET                                  ; 0340
                                 50  D4 000AF 15$:    CLRL      R0                         ; 0341
                                     04 000B1 16$:    RET                                  ;

; Routine Size:  178 bytes,    Routine Base:  $CODE$ + 011C
```

```
1 GLOBAL ROUTINE LOG_MESSAGE_PUT (LEN, ADR) =
1
1 !++
1 ! Functional description:
1 !
1 !       Place the record in the log file.
1 ! **
1 ! ** This routine will be enhanced later to support multiple log files.
1 ! **
1 !
1 ! Input:
1 !
1 !       LEN - Length of record
1 !       ADR - Address of record
1 !
1 ! Implicit Input:
1 !
1 !       None.
1 !
1 ! Output:
1 !
1 !       None.
1 !
1 ! Implicit output:
1 !
1 !       None.
1 !
1 ! Side effects:
1 !
1 !       None.
1 !
1 ! Routine value:
1 !
1 !       TRUE            : If success
1 !       <anything else> : If the log attempt failed
1 !--
1
2 BEGIN                                         ! Start of LOG_MESSAGE_PUT
2
2 EXTERNAL
2       GLOBAL_STATUS   : BITVECTOR,            ! OPCOM global status flags
2       LOGFILE_RAB     : $bblock,              ! RMS control structure
2       LOGFILE_FAB     : $bblock;              ! RMS control structure
2
2 LOCAL
2       MESSAGE         : LONG,                 ! Error message code
2       STATUS          : LONG;
2 !
2 ! Write the message to the logfile.
2 !
2 LOGFILE_RAB [RAB$W_RSZ] = .LEN;
2 LOGFILE_RAB [RAB$L_RBF] = .ADR;
2 GLOBAL_STATUS [GBLSTS_K_FLUSH_PENDING] = TRUE;
3 IF NOT (STATUS = $PUT (RAB = LOGFILE_RAB))
2 THEN
3     BEGIN
3     !
```

```
:  402          0399  3     ! The log attempt failed.  Complain if appropriate.
:  403          0400  3     !
:  404          0401  3     IF NOT .GLOBAL_STATUS [GBLSTS_K_LAST_LOG_FAILED]
:  405          0402  3     THEN
:  406          0403  4         BEGIN
:  407          0404  4         !
:  408          0405  4         ! Complain to the appropriate operators.
:  409          0406  4         !
:  410          0407  4         MESSAGE = OPC$_LOGFAIL;
:  411          0408  4         ! *** the remainder will be supplied later ***
:  412          0409  3         END;
:  413          0410  3      GLOBAL_STATUS [GBLSTS_K_LAST_LOG_FAILED] = TRUE;
:  414          0411  2      END;
:  415          0412  2
:  416          0413  2 RETURN (.STATUS);
:  417          0414  2
:  418          0415  1 END;                                      ! End of LOG_MESSAGE_PUT


                                      0004 00000          .ENTRY  LOG_MESSAGE_PUT, Save R2                   ; 0342
                    52      0000G  CF   9E 00002          MOVAB   GLOBAL_STATUS, R2                          ;
             0000G  CF         04  AC   B0 00007          MOVW    LEN, LOGFILE_RAB+34                        ; 0392
             0000G  CF         08  AC   D0 0000D          MOVL    ADR, LOGFILE_RAB+40                        ; 0393
                    62         80  8F   88 00013          BISB2   #128, GLOBAL_STATUS                        ; 0394
                            0000G  CF   9F 00017          PUSHAB  LOGFILE_RAB                                ; 0395
         00000000G  00             01   FB 0001B          CALLS   #1, SYS$PUT                                ;
                    0E             50   E8 00022          BLBS    STATUS, 2$                                 ;
        07          62             02   E0 00025          BBS     #2, GLOBAL_STATUS, 1$                      ; 0401
                    51 00058034    8F   D0 00029          MOVL    #360500, MESSAGE                           ; 0407
                    62             04   88 00030 1$:      BISB2   #4, GLOBAL_STATUS                          ; 0410
                                        04 00033 2$:      RET                                                ; 0415

; Routine Size:  52 bytes,    Routine Base:  $CODE$ + 01CE
```

```
1 ROUTINE LOGEVENT_FAO_BUFFER (CTRSTR : REF VECTOR[2], ARGS : VECTOR [4]) =        %SBTTL 'LOGEVENT_FAO_BUFFER'
2 BEGIN
2 !++
2 !
2 ! FUNCTIONAL DESCRIPTION:
2 !
2 !       This routine passes an ascii string through the FAO system service with any number of specified para
2 !
2 ! INPUTS:
2 !
2 !       ctrstr  Address of FAO control string descriptor
2 !       args    Any number of additional arguments
2 !
2 ! IMPLICIT INPUTS:
2 !
2 !       none
2 !
2 ! OUTPUTS:
2 !
2 !       none
2 !
2 ! IMPLICIT OUTPUTS:
2 !
2 !       none
2 !
2 ! ROUTINE VALUE:
2 !
2 !       Address of formatted descriptor
2 !
2 ! SIDE EFFECTS:
2 !
2 !       none
2 !--
2
2 OWN
2     desc : VECTOR [2, LONG],
2     faobuf : VECTOR [512, BYTE]
2     ;
2
2 desc [0] = 512;                              ! Set up result descriptor
2 desc [1] = faobuf;
2
2 $faol (ctrstr=.ctrstr, outlen=desc, outbuf=desc, prmlst=args);
2
2 RETURN desc;
1 END;


                                    .PSECT  $OWN$,NOEXE,2

                  00000 DESC:       .BLKB   8
                  00008 FAOBUF:     .BLKB   512

                                    .EXTRN  SYS$FAOL

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

```
                                      0004 0000C LOGEVENT_FAO_BUFFER:
                                                        .WORD    Save R2                              ; 0416
                        52     0000'   CF   9E 00002    MOVAB    DESC, R2                             ;
                        62     0200    8F   3C 00007    MOVZWL   #512, DESC                           ; 0455
                   04   A2       08    A2   9E 0000C    MOVAB    FAOBUF, DESC+4                       ; 0456
                                 08    AC   9F 00011    PUSHAB   ARGS                                 ; 0458
                                       52   DD 00014    PUSHL    R2                                   ;
                                       52   DD 00016    PUSHL    R2                                   ;
                                 04    AC   DD 00018    PUSHL    CTRSTR                               ;
            00000000G   00             04   FB 0001B    CALLS    #4, SYS$FAOL                         ;
                        50             62   9E 00022    MOVAB    DESC, R0                             ; 0460
                                            04 00025    RET                                           ; 0461

; Routine Size:  38 bytes,    Routine Base:  $CODE$ + 0202
```

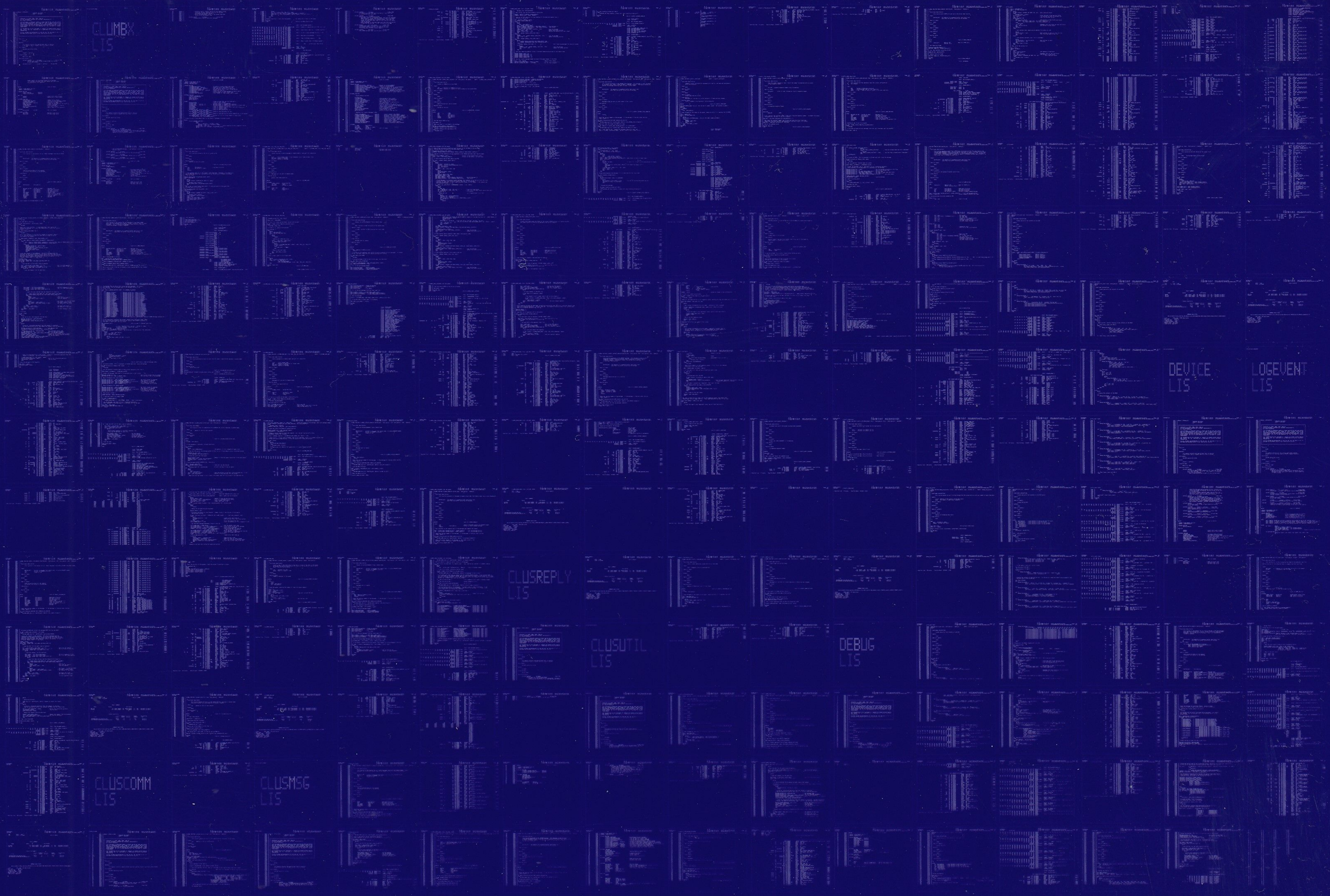
```
1 GLOBAL ROUTINE WRITE_LOG_FILE (DESC : $ref_bblock) =
1
1 !++
1 ! Functional description:
1 !
1 !       This routine will write a message described by simple string desc
1 !       to the operator log file.
1 ! **
1 ! ** This routine will be enhanced later to support mutliple log files.
1 ! **
1 !
1 ! Input:
1 !
1 !       DESC             : Address of a string desc
1 !
1 ! Implicit Input:
1 !
1 !       RQCB [RQCB_L_MCB] points to a valid MCB.
1 !
1 ! Output:
1 !
1 !       None.
1 !
1 ! Implicit output:
1 !
1 !       None.
1 !
1 ! Side effects:
1 !
1 !       None.
1 !
1 ! Routine value:
1 !
1 !       TRUE             : If success
1 !       <anything else>  : If the log attempt failed
1 !--
1
2 BEGIN                                         ! Start of WRITE_LOG_FILE
2
2 EXTERNAL
2       GLOBAL_STATUS    : BITVECTOR,           ! OPCOM global status flags
2       LOGFILE_RAB      : $bblock,             ! RMS control structure
2       LOGFILE_FAB      : $bblock;             . RMS control structure
2
2 LOCAL
2       NULLDESC         : LONG;                ! Only need length word
2
2 !
2 ! See if logging is enabled.
2 !
3 IF  (NOT .GLOBAL_STATUS [GBLSTS_K_LOGGING_ENABLED])
3 OR  (.GLOBAL_STATUS [GBLSTS_K_LOGFILE_CLOSED])
2 THEN
2    RETURN (TRUE);
2 !
2 ! Format and print the message header
2 !
```

```
;  524              0519  2 NULLDESC = 0;                                          ! A blank line
;  525              0520  2 PUT (NULLDESC);
;  526              0521  2 PUT (LOGEVENT_FAO_BUFFER (%ASCID '%%%%%%%%%%%  OPCOM  !%D  %%%%%%%%%%%  WRITE_LOG_FILE', 0));
;  527              0522  2 PUT_EXPLANATION;
;  528              0523  2 PUT (LOGEVENT_FAO_BUFFER (%ASCID '"!AS"', .DFSC));
;  529              0524  2 $FLUSH (RAB = LOGFILE_RAB);
;  530              0525  2
;  531              0526  2 RETURN (TRUE);
;  532              0527  1 END;                                                   ! End of WRITE_LOG_FILE


                                                                       .PSECT  $PLIT$,NOWRT,NOEXE,2

50  4F  20  20  25  25  25  25  25  25  25  25  25  25  25  001D0 P.AAP:  .ASCII  \%%%%%%%%%%%  OPCOM  !%D  %%%%%%%%%%%  WR\  ;
25  25  25  25  25  20  20  44  25  21  20  20  4D  4F  43  001DF                                                          ;
                    52  57  20  20  25  25  25  25  25  25  001EE                                                          ;
            45  4C  49  46  5F  47  4F  4C  5F  45  54  49  001F8         .ASCII  \ITE_LOG_FILE\                           ;
                                             010E0034  00204 P.AAO:  .LONG   17694772                                  ;
                                             00000000' 00208         .ADDRESS P.AAP                                    ;
                        00  00  00  22  53  41  21  22  0020C P.AAR:  .ASCII  \"!AS"\<0><0><0>                         ;
                                             010E0005  00214 P.AAQ:  .LONG   17694725                                  ;
                                             00000000' 00218         .ADDRESS P.AAR                                    ;

                                                        DSC=                     P.AAA
                                                        DSC=                     P.AAC
                                                        DSC=                     P.AAE
                                                        DSC=                     P.AAG


                                                                       .PSECT  $CODE$,NOWRT,2

                                           000C  00000         .ENTRY  WRITE_LOG_FILE, Save R2,R3                ; 0462
                     53 00000000G  00  9E  00002         MOVAB   SYS$PUT, R3                                       ;
                     52     0000G  CF  9E  00009         MOVAB   LOGFILE_RAB, R2                                   ;
                     5E            04  C2  0000E         SUBL2   #4, SP                                            ;
           03        0000G   CF    01  E0  00011         BBS     #1, GLOBAL_STATUS, 2$                             ; 0512
                                 0094  31  00017 1$:     BRW     3$                                                ;
           F7        0000G   CF    03  E0  0001A 2$:     BBS     #3, GLOBAL_STATUS, 1$                             ; 0513
                                   6E  D4  00020         CLRL    NULLDESC                                          ; 0519
                22   A2            6E  B0  00022         MOVW    DSC, LOGFILE_RAB+34                               ; 0520
                28   A2            6D  D0  00026         MOVL    DSC+4, LOGFILE_RAB+40                             ;
                                   52  DD  0002A         PUSHL   R2                                                ;
                     63            01  FB  0002C         CALLS   #1, SYS$PUT                                       ;
                                   7E  D4  0002F         CLRL    -(SP)                                             ; 0521
                          0000'    CF  9F  00031         PUSHAB  P.AAO                                             ;
                A1   AF            02  FB  00035         CALLS   #2, LOGEVENT_FAO_BUFFER                           ;
                22   A2            60  B0  00039         MOVW    (R0), LOGFILE_RAB+34                              ;
                28   A2      04    A0  D0  0003D         MOVL    4(R0), LOGFILE_RAB+40                             ;
                                   52  DD  00042         PUSHL   R2                                                ;
                     63            01  FB  00044         CALLS   #1, SYS$PUT                                       ;
                22   A2   0000'    CF  B0  00047         MOVW    DSC, LOGFILE_RAB+34                               ;
                28   A2   0000'    CF  D0  0004D         MOVL    DSC+4, LOGFILE_RAB+40                             ;
                                   52  DD  00053         PUSHL   R2                                                ;
                     63            01  FB  00055         CALLS   #1, SYS$PUT                                       ;
                22   A2   0000'    CF  B0  00058         MOVW    DSC, LOGFILE_RAB+34                               ;
                28   A2   0000'    CF  D0  0005E         MOVL    DSC+4, LOGFILE_RAB+40                             ;
```

```
                                        52  DD 00064          PUSHL   R2                                          ;
                         63             01  FB 00066          CALLS   #1, SYS$PUT                                 ;
                    22   A2      0000'  CF  B0 00069          MOVW    DSC, LOGFILE_RAB+34                         ;
                    28   A2      0000'  CF  D0 0006F          MOVL    DSC+4, LOGFILE_RAB+40                       ;
                                        52  DD 00075          PUSHL   R2                                          ;
                         63             01  FB 00077          CALLS   #1, SYS$PUT                                 ;
                    22   A2      0000'  CF  B0 0007A          MOVW    DSC, LOGFILE_RAB+34                         ;
                    28   A2      0000'  CF  D0 00080          MOVL    DSC+4, LOGFILE_RAB+40                       ;
                                        52  DD 00086          PUSHL   R2                                          ;
                         63             01  FB 00088          CALLS   #1, SYS$PUT                                 ;
                                   04   AC  DD 0008B          PUSHL   DESC                                        ; 0523
                                 0000'  CF  9F 0008E          PUSHAB  P.AAQ                                       ;
                  FF43   CF             02  FB 00092          CALLS   #2, LOGEVENT_FAO_BUFFER                     ;
                    22   A2             60  B0 00097          MOVW    (R0), LOGFILE_RAB+34                        ;
                    28   A2        04   A0  D0 0009B          MOVL    4(R0), LOGFILE_RAB+40                       ;
                                        52  DD 000A0          PUSHL   R2                                          ;
                         63             01  FB 000A2          CALLS   #1, SYS$PUT                                 ;
                                        52  DD 000A5          PUSHL   R2                                          ; 0524
            00000000G    00             01  FB 000A7          CALLS   #1, SYS$FLUSH                               ;
                         50             01  D0 000AE 3$:      MOVL    #1, R0                                      ; 0526
                                            04 000B1          RET                                                 ; 0527

; Routine Size:  178 bytes,    Routine Base:  $CODE$ + 0228


;  533          0528  1
;  534          0529  1 END                                                       ! End of LOGEVENT
;  535          0530  0 ELUDOM




;                                        PSECT SUMMARY
;
;       Name                        Bytes                          Attributes
;
;  $PLIT$                                540  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $CODE$                                730  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $OWN$                                 520  NOVEC,  WRT,  RD ,NOEXE,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        13         0       1000        00:01.8
;  _$255$DUA28:[OPCOM.OBJ]OPCOMLIB.L32;1             633        15         2         43        00:00.8
;                                    COMMAND QUALIFIERS

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

; Size:          730 code + 1060 data bytes
; Run Time:         00:18.2
; Elapsed Time:     01:01.1
; Lines/CPU Min:     1751
; Lexemes/CPU-Min: 23511
; Memory Used:  135 pages
; Compilation Complete
```

0289 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



0290 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

LOGFILE
LIS

OPCOMDEF
LIS

OPCOMDATA
LIS

OPCOMINI
LIS

OPCOMUTIL
LIS

OPCOMLIB
LIS

OPCDEFTMP
LIS

OPCOMMAIN
LIS

OPCOMOLD
LIS

OPCOMRPLY
LIS

OPCCRASH
LIS

OPERUTIL
LIS

OPCOMRQST
LIS