```
EEEEEEEEEEEEEEE  VVV         VVV  LLL
EEEEEEEEEEEEEEE  VVV         VVV  LLL
EEEEEEEEEEEEEEE  VVV         VVV  LLL
EEE              VVV         VVV  LLL
EEE              VVV         VVV  LLL
EEE              VVV         VVV  LLL
EEE              VVV         VVV  LLL
EEE              VVV         VVV  LLL
EEE              VVV         VVV  LLL
EEEEEEEEEEEE     VVV         VVV  LLL
EEEEEEEEEEEE     VVV         VVV  LLL
EEEEEEEEEEEE     VVV         VVV  LLL
EEE              VVV         VVV  LLL
EEE              VVV         VVV  LLL
EEE              VVV         VVV  LLL
EEE                 VVV   VVV     LLL
EEE                 VVV   VVV     LLL
EEE                 VVV   VVV     LLL
EEEEEEEEEEEEEEE        VVV        LLLLLLLLLLLLLLL
EEEEEEEEEEEEEEE        VVV        LLLLLLLLLLLLLLL
EEEEEEEEEEEEEEE        VVV        LLLLLLLLLLLLLLL
```

```
_$25

Valu
----
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
```

FILEID**EVLMAIN

```
EEEEEEEEEE  VV      VV  LL          MM      MM    AAAAAA     IIIIII    NN      NN
EEEEEEEEEE  VV      VV  LL          MM      MM    AAAAAA     IIIIII    NN      NN
EE          VV      VV  LL          MMMM  MMMM   AA    AA      II      NN      NN
EE          VV      VV  LL          MMMM  MMMM   AA    AA      II      NN      NN
EE          VV      VV  LL          MM  MM  MM   AA    AA      II      NNNN    NN
EE          VV      VV  LL          MM  MM  MM   AA    AA      II      NNNN    NN
EEEEEEEE    VV      VV  LL          MM      MM   AA    AA      II      NN  NN  NN
EEEEEEEE    VV      VV  LL          MM      MM   AA    AA      II      NN  NN  NN
EE          VV      VV  LL          MM      MM   AAAAAAAAAA    II      NN    NNNN
EE          VV      VV  LL          MM      MM   AAAAAAAAAA    II      NN    NNNN
EE            VV  VV    LL          MM      MM   AA    AA      II      NN      NN      ....
EE            VV  VV    LL          MM      MM   AA    AA      II      NN      NN      ....
EEEEEEEEEE      VV      LLLLLLLLLL  MM      MM   AA    AA    IIIIII    NN      NN      ....
EEEEEEEEEE      VV      LLLLLLLLLL  MM      MM   AA    AA    IIIIII    NN      NN      ....

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 %TITLE  'DECnet Event Logger Main Module'
;  2     0002  0 MODULE EVLMAIN  (
;  3     0003  0                 LANGUAGE (BLISS32),
;  4     0004  0                 IDENT = 'V04-000',
;  5     0005  0                 MAIN = EVL$MAIN
;  6     0006  0                 ) =
;  7     0007  1 BEGIN
;  8     0008  1
;  9     0009  1 !
; 10     0010  1 !*****************************************************************************
; 11     0011  1 !*                                                                           *
; 12     0012  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
; 13     0013  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                    *
; 14     0014  1 !*  ALL RIGHTS RESERVED.                                                     *
; 15     0015  1 !*                                                                           *
; 16     0016  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
; 17     0017  1 !*  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE          *
; 18     0018  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER     *
; 19     0019  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
; 20     0020  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY      *
; 21     0021  1 !*  TRANSFERRED.                                                             *
; 22     0022  1 !*                                                                           *
; 23     0023  1 !*  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE    *
; 24     0024  1 !*  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT       *
; 25     0025  1 !*  CORPORATION.                                                             *
; 26     0026  1 !*                                                                           *
; 27     0027  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS     *
; 28     0028  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
; 29     0029  1 !*                                                                           *
; 30     0030  1 !*                                                                           *
; 31     0031  1 !*****************************************************************************
; 32     0032  1 !
; 33     0033  1
; 34     0034  1 !++
; 35     0035  1 ! FACILITY:     DECnet Event Logging (EVL)
; 36     0036  1 !
; 37     0037  1 ! ABSTRACT:
; 38     0038  1 !
; 39     0039  1 !       This module contains the main entry for the event logger.
; 40     0040  1 !       A few routines of general utility are in the module too.
; 41     0041  1 !
; 42     0042  1 ! ENVIRONMENT:  VAX/VMS Operating System
; 43     0043  1 !
; 44     0044  1 ! AUTHOR:       Darrell Duffy   , CREATION DATE:  16-June-1980
; 45     0045  1 !
; 46     0046  1 ! MODIFIED BY:
; 47     0047  1 !
; 48     0048  1 !       V001    TMH0001         Tim Halvorsen   25-Jun-1981
; 49     0049  1 !                               Purge working set to minimum before hibernating,
; 50     0050  1 !                               to limit system physical memory usage while not
; 51     0051  1 !                               doing anything.
; 52     0052  1 !                               Change references to ASCID macro to use %ASCID.
; 53     0053  1 !--
```

```
;    55          0054  1 %SBTTL  'Definitions'
;    56          0055  1
;    57          0056  1 !
;    58          0057  1 ! TABLE OF CONTENTS:
;    59          0058  1 !
;    60          0059  1
;    61          0060  1 FORWARD ROUTINE
;    62          0061  1         EVL$MAIN            :,              ! Main entry
;    63          0062  1         EVL$INITLOG         : NOVALUE,      ! Initialize for debug logging
;    64          0063  1         EVL$PRINTLOG        : NOVALUE       ! Print data messages to log file
;    65          0064  1         ;
;    66          0065  1
;    67          0066  1 !
;    68          0067  1 ! INCLUDE FILES:
;    69          0068  1 !
;    70          0069  1
;    71          0070  1 LIBRARY 'LIB$:EVLIBRARY';               ! EVL definitions
;    72          0071  1 LIBRARY 'SYS$LIBRARY:STARLET';          ! VMS common definitions
;    73          0072  1
;    74          0073  1 !
;    75          0074  1 ! OWN STORAGE:
;    76          0075  1 !
;    77          0076  1
;    78          0077  1 GLOBAL
;    79          0078  1         EVL$GL_LOGMASK  : BBLOCK [8]            ! Logging control mask
;    80          0079  1         ;
;    81          0080  1
;    82          0081  1 !
;    83          0082  1 ! EXTERNAL REFERENCES:
;    84          0083  1 !
;    85          0084  1
;    86          0085  1 EXTERNAL
;    87          0086  1         EVL$GW_NETSHOCHAN       : WORD,         ! Channel to net for show
;    88          0087  1         EVL$B_RCVDONE           : BYTE,         ! True if receive inactive
;    89          0088  1         EVL$GB_TRANSDONE        : BYTE;         ! True if transmitter inactive
;    90          0089  1
;    91          0090  1 EXTERNAL ROUTINE
;    92          0091  1         EVL$RECEIVE             : NOVALUE,      ! Activate the receiver
;    93          0092  1         EVL$TRANSMIT            : NOVALUE,      ! Activate the transmitter
;    94          0093  1         WKQ$DO_WORK_ITEM        :,              ! Perform a work item in queue
;    95          0094  1         EVL$OBTAINNETCHAN       : NOVALUE       ! Obtain a channel to net
;    96          0095  1         ;
;    97          0096  1
;    98          0097  1 EXTERNAL ROUTINE
;    99          0098  1
;   100          0099  1         LIB$PUT_OUTPUT          : ADDRESSING_MODE (GENERAL),
;   101          0100  1         LIB$CVT_HTB             : ADDRESSING_MODE (GENERAL)
;   102          0101  1         ;
;   103          0102  1
```

```
;  105          0103  1 %SBTTL  'EVL$MAIN  Main Entry'
;  106          0104  1 GLOBAL ROUTINE EVL$MAIN =
;  107          0105  1
;  108          0106  1 !++
;  109          0107  1 ! FUNCTIONAL DESCRIPTION:
;  110          0108  1 !
;  111          0109  1 !       This is the main entry point for the event logger.
;  112          0110  1 !       It calls the initialization routines
;  113          0111  1 !       and sits in a loop performing work from the work queue.
;  114          0112  1 !
;  115          0113  1 ! FORMAL PARAMETERS:
;  116          0114  1 !
;  117          0115  1 !       NONE
;  118          0116  1 !
;  119          0117  1 ! IMPLICIT INPUTS:
;  120          0118  1 !
;  121          0119  1 !       NONE
;  122          0120  1 !
;  123          0121  1 ! IMPLICIT OUTPUTS:
;  124          0122  1 !
;  125          0123  1 !       NONE
;  126          0124  1 !
;  127          0125  1 ! ROUTINE VALUE:
;  128          0126  1 ! COMPLETION CODES:
;  129          0127  1 !
;  130          0128  1 !       NONE
;  131          0129  1 !
;  132          0130  1 ! SIDE EFFECTS:
;  133          0131  1 !
;  134          0132  1 !       NONE
;  135          0133  1 !
;  136          0134  1 !--
;  137          0135  1
;  138          0136  2     BEGIN
;  139          0137  2
;  140          0138  2     EVL$INITLOG ();                         ! Initialize for debug logging
;  141          0139  2     EVL$OBTAINNETCHAN (EVL$GW_NETSHOCHAN); ! Obtain a channel to net
;  142          0140  2     EVL$RECEIVE ();                         ! Initialize receiver portion
;  143          0141  2     EVL$TRANSMIT ();                        ! Initialize transmitter portion
;  144          0142  2
;  145          0143  3     WHILE NOT (.EVL$B_RCVDONE AND .EVL$GB_TRANSDONE) ! Until both sides finish,
;  146          0144  2     DO
;  147          0145  3         BEGIN
;  148          0146  3         $PURGWS(INADR=PLIT(0,%X'7FFFFFFF')); ! Purge entire working set
;  149          0147  3         $HIBER;                             ! Wait on some work to do
;  150          0148  3         WHILE WKQ$DO_WORK_ITEM () DO;       ! Do work til empty queue
;  151          0149  2         END;
;  152          0150  2
;  153          0151  2     RETURN TRUE;                            ! Exit suces fully
;  154          0152  1     END;


                                                        .TITLE  EVLMAIN DECnet Event Logger Main Module
                                                        .IDENT  \V04-000\

                                                        .PSECT  $PLIT$,NOWRT,NOEXE,2
```

```
                                00000002  00000                 .LONG   2                                                       ;
                        7FFFFFFF  00000000  00004 P.AAA:        .LONG   0, 2147483647                                           ;

                                                                .PSECT  $GLOBAL$,NOEXE,2

                                          00000 EVL$GL_LOGMASK::
                                                                .BLKB   8

                                                                .EXTRN  EVL$GW_NETSHOCHAN
                                                                .EXTRN  EVL$B_RCVDONE, EVL$GB_TRANSDONE
                                                                .EXTRN  EVL$RECEIVE, EVL$TRANSMIT
                                                                .EXTRN  WKQ$DO_WORK_ITEM
                                                                .EXTRN  EVL$OBTAINNETCHAN
                                                                .EXTRN  LIB$PUT_OUTPUT, LIB$CVT_HTB
                                                                .EXTRN  SYS$PURGWS, SYS$HIBER

                                                                .PSECT  $CODE$,NOWRT,2

                                     0000 00000                 .ENTRY  EVL$MAIN, Save nothing                                  ; 0104
                0000V   CF                00   FB 00002         CALLS   #0, EVL$INITLOG                                         ; 0138
                                  0000G   CF   9F 00007         PUSHAB  EVL$GW_NETSHOCHAN                                       ; 0139
                0000G   CF                01   FB 0000B         CALLS   #1, EVL$OBTAINNETCHAN                                   ;
                0000G   CF                00   FB 00010         CALLS   #0, EVL$RECEIVE                                         ; 0140
                0000G   CF                00   FB 00015         CALLS   #0, EVL$TRANSMIT                                        ; 0141
                        05        0000G   CF   E9 0001A 1$:      BLBC    EVL$B_RCVDONE, 2$                                       ; 0143
                        1C        0000G   CF   E8 0001F         BLBS    EVL$GB_TRANSDONE, 4$                                    ;
                                  0000'   CF   9F 00024 2$:      PUSHAB  P.AAA                                                   ; 0146
          00000000G     00                01   FB 00028         CALLS   #1, SYS$PURGWS                                          ;
          00000000G     00                00   FB 0002F         CALLS   #0, SYS$HIBER                                           ;
              0000G     CF                00   FB 00036 3$:      CALLS   #0, WKQ$DO_WORK_ITEM                                    ; 0148
                        DC                50   E9 0003B         BLBC    R0, 1$                                                  ;
                                          F6   11 0003E         BRB     3$                                                      ;
                        50                01   D0 00040 4$:      MOVL    #1, R0                                                  ; 0151
                                               04 00043         RET                                                             ; 0152

; Routine Size:  68 bytes,    Routine Base:  $CODE$ + 0000
```

```
;  156          0153  1 %SBTTL 'EVL$INITLOG     Initialization debug logging'
;  157          0154  1 ROUTINE EVL$INITLOG : NOVALUE =
;  158          0155  1
;  159          0156  1 !++
;  160          0157  1 ! FUNCTIONAL DESCRIPTION:
;  161          0158  1 !
;  162          0159  1 !       This routine initializes the internal logging flags for EVL debugging.
;  163          0160  1 !       The logical name EVL$LOG is translated to obtain a hex number which is
;  164          0161  1 !       converted to binary.  Each bit of the mask controls information to
;  165          0162  1 !       be logged.
;  166          0163  1 !
;  167          0164  1 ! FORMAL PARAMETERS:
;  168          0165  1 !
;  169          0166  1 !       NONE
;  170          0167  1 !
;  171          0168  1 ! IMPLICIT INPUTS:
;  172          0169  1 !
;  173          0170  1 !       EVL$LOG logical name
;  174          0171  1 !
;  175          0172  1 ! IMPLICIT OUTPUTS:
;  176          0173  1 !
;  177          0174  1 !       EVL$GL_LOGMASK
;  178          0175  1 !
;  179          0176  1 ! ROUTINE VALUE:
;  180          0177  1 ! COMPLETION CODES:
;  181          0178  1 !
;  182          0179  1 !       NONE
;  183          0180  1 !
;  184          0181  1 ! SIDE EFFECTS:
;  185          0182  1 !
;  186          0183  1 !       NONE
;  187          0184  1 !
;  188          0185  1 !--
;  189          0186  1
;  190          0187  2     BEGIN
;  191          0188  2
;  192          0189  2     LITERAL
;  193          0190  2         RSLSIZE = 10                        ! Size of the result buffer
;  194          0191  2         ;
;  195          0192  2
;  196          0193  2     LOCAL
;  197          0194  2         RSLBFR : VECTOR [RSLSIZE, BYTE],    ! Buffer for the translation
;  198          0195  2         RSLDSC : VECTOR [2]                 ! Descriptor for the buffer
;  199          0196  2         ;
;  200          0197  2
;  201          0198  2     EVL$GL_LOGMASK = 0;                     ! Initialize the logging mask
;  202          0199  2     RSLDSC [0] = RSLSIZE;                   ! Setup the descriptor
;  203          0200  2     RSLDSC [1] = RSLBFR;
;  204          0201  2
;  205          0202  2     IF                                      ! We must get a translation
;  206          0203  3         (
;  207        P 0204  3         $TRNLOG                             ! Translate the name once
;  208        P 0205  3             (
;  209        P 0206  3             LOGNAM = %ASCID 'EVL$LOG',      ! This is the logical name
;  210        P 0207  3             RSLLEN = RSLDSC [0],            ! Place the length here
;  211        P 0208  3             RSLBUF = RSLDSC                 ! Place the translation here
;  212          0209  4             )
```

```
;  213     0210  3          )
;  214     0211  2          EQL
;  215     0212  2          SS$_NORMAL                          ! If a successful translation
;  216     0213  2      THEN                                    ! Then convert the result
;  217     0214  2          LIB$CVT_HTB                         ! Convert hex to binary
;  218     0215  2              (
;  219     0216  2              .RSLDSC [0],                    ! Size of string
;  220     0217  2              .RSLDSC [1],                    ! Address of string
;  221     0218  2              EVL$GL_LOGMASK                  ! Address of longword result
;  222     0219  2              )
;  223     0220  2      ;
;  224     0221  2
;  225     0222  1      END;


                                                    .PSECT  $PLIT$,NOWRT,NOEXE,2

                     00  47  4F  4C  24  4C  56  45  0000C P.AAC:  .ASCII  \EVL$LOG\<0>                    ;
                                       010E0007  00014 P.AAB:  .LONG   17694727                          ;
                                       00000000' 00018         .ADDRESS P.AAC                            ;

                                                    .EXTRN  SYS$TRNLOG

                                                    .PSECT  $CODE$,NOWRT,2

                                         0000 00000 EVL$INITLOG:
                                                    .WORD   Save nothing                              ; 0154
                           5E                 10  C2 00002          SUBL2   #16, SP                   ;
                                        0000'  CF  D4 00005          CLRL    EVL$GL_LOGMASK            ; 0198
                                               0A  DD 00009          PUSHL   #10                       ; 0199
                        04    AE         08    AE  9E 0000B          MOVAB   RSLBFR, RSLDSC+4          ; 0200
                                               7E  7C 00010          CLRQ    -(SP)                     ; 0209
                                               7E  D4 00012          CLRL    -(SP)                     ;
                                         0C    AE  9F 00014          PUSHAB  RSLDSC                    ;
                                         10    AE  9F 00017          PUSHAB  RSLDSC                    ;
                                        0000'  CF  9F 0001A          PUSHAB  P.AAB                     ;
                              00000000G  00        06  FB 0001E      CALLS   #6, SYS$TRNLOG            ;
                                         01        50  D1 00025      CMPL    R0, #1                    ; 0211
                                                   11  12 00028      BNEQ    1$                        ;
                                        0000'  CF  9F 0002A          PUSHAB  EVL$GL_LOGMASK            ; 0215
                                         08    AE  DD 0002E          PUSHL   RSLDSC+4                  ; 0217
                                         08    AE  DD 00031          PUSHL   RSLDSC                    ; 0216
                              00000000G  00        03  FB 00034      CALLS   #3, LIB$CVT_HTB           ;
                                                   04 0003B 1$:     RET                               ; 0222

; Routine Size:  60 bytes,    Routine Base:  $CODE$ + 0044
```

```
;   227         0223  1 %SBTTL  'EVL$PRINTLOG  Print a Data Message to the Log'
;   228         0224  1 GLOBAL ROUTINE EVL$PRINTLOG (LOGBIT, HEADDSC, EXTRADSC, DATADSC) : NOVALUE =
;   229         0225  1
;   230         0226  1 !++
;   231         0227  1 ! FUNCTIONAL DESCRIPTION:
;   232         0228  1 !
;   233         0229  1 !       Check the logging control mask and if the corresponding bit is set
;   234         0230  1 !       then print the special message to the log file.  The message
;   235         0231  1 !       has a header and optionally some extra text which explains the
;   236         0232  1 !       logged message.
;   237         0233  1 !
;   238         0234  1 ! FORMAL PARAMETERS:
;   239         0235  1 !
;   240         0236  1 !       LOGBIT          Value of the logging bit to control the logging
;   241         0237  1 !                       as a bit number
;   242         0238  1 !       HEADDSC         Address of a descriptor of the header text
;   243         0239  1 !       EXTRADSC        Address of a descriptor of the extra text (optional)
;   244         0240  1 !       DATADSC         Address of a descriptor of the data to be converted
;   245         0241  1 !                       and printed
;   246         0242  1 !
;   247         0243  1 ! IMPLICIT INPUTS:
;   248         0244  1 !
;   249         0245  1 !       EVL$GL_LOGCONTROL
;   250         0246  1 !
;   251         0247  1 ! IMPLICIT OUTPUTS:
;   252         0248  1 !
;   253         0249  1 !       NONE
;   254         0250  1 !
;   255         0251  1 ! ROUTINE VALUE:
;   256         0252  1 ! COMPLETION CODES:
;   257         0253  1 !
;   258         0254  1 !       NONE
;   259         0255  1 !
;   260         0256  1 ! SIDE EFFECTS:
;   261         0257  1 !
;   262         0258  1 !       NONE
;   263         0259  1 !
;   264         0260  1 !--
;   265         0261  1
;   266         0262  2     BEGIN
;   267         0263  2
;   268         0264  2     MAP
;   269         0265  2         HEADDSC         : REF BBLOCK,
;   270         0266  2         EXTRADSC        : REF BBLOCK,
;   271         0267  2         DATADSC         : REF BBLOCK
;   272         0268  2         ;
;   273         0269  2
;   274         0270  2     LITERAL
;   275         0271  2         FAOSIZ = 256                    ! The print buffer
;   276         0272  2         ;
;   277         0273  2
;   278         0274  2     LOCAL
;   279         0275  2         FAOBUF : VECTOR [FAOSIZ, BYTE], ! Print buffer
;   280         0276  2         FAOLST : VECTOR [100],          ! List of args to $FAOL
;   281         0277  2         OUTDSC : VECTOR [2],            ! Descriptor of the output line
;   282         0278  2         BYTES,                          ! Counter for bytes written
;   283         0279  2         CTR : SIGNED,                   ! A random counter
```

```
;   284          0280  2          PTR,                                    ! A random pointer
;   285          0281  2          ITR_CNT                                 ! Temporary iteration count
;   286          0282  2          ;
;   287          0283  2
;   288          0284  2      !
;   289          0285  2      !   See if this text should be logged, and if not then return
;   290          0286  2      !
;   291          0287  2
;   292          0288  2          IF NOT .EVL$GL_LOGMASK [0, .LOGBIT, 1, 0]
;   293          0289  2          THEN
;   294          0290  2              RETURN
;   295          0291  2          ;
;   296          0292  2
;   297          0293  2          OUTDSC [0] = FAOSIZ;                    ! Initialize the output buffer dsc
;   298          0294  2          OUTDSC [1] = FAOBUF;
;   299          0295  2          FAOLST [0] = .HEADDSC;                  ! Header text
;   300          0296  2          FAOLST [1] = .DATADSC [DSC$W_LENGTH]; ! Data length
;   301          0297  2          FAOLST [2] =                            ! Extra text dsc
;   302          0298  3              (
;   303          0299  3              IF .EXTRADSC EQL 0
;   304          0300  3              THEN
;   305          0301  3                  %ASCID ''
;   306          0302  3              ELSE
;   307          0303  3                  .EXTRADSC
;   308          0304  2              );
;   309        P 0305  2          $FAOL                                   ! Write the header out
;   310        P 0306  2              (
;   311        P 0307  2              CTRSTR = %ASCID '!/  !AS  (length = !UL bytes)!/  !AS!/',
;   312        P 0308  2              OUTLEN = OUTDSC [0],
;   313        P 0309  2              OUTBUF = OUTDSC,
;   314        P 0310  2              PRMLST = FAOLST
;   315          0311  2              );
;   316          0312  2          LIB$PUT_OUTPUT (OUTDSC);
;   317          0313  2
;   318          0314  2          CTR = .DATADSC [DSC$W_LENGTH];          ! Size of message
;   319          0315  2          PTR = .DATADSC [DSC$A_POINTER];         ! Its address
;   320          0316  2          WHILE .CTR GTR 0                        ! Process it all
;   321          0317  2          DO
;   322          0318  3              BEGIN
;   323          0319  3              OUTDSC [0] = FAOSIZ;                ! Set a descriptor
;   324          0320  3              OUTDSC [1] = FAOBUF;
;   325          0321  3              ITR_CNT = MIN (.CTR, 20);           ! Get byte count
;   326          0322  3              FAOLST [0] = .ITR_CNT;              ! Add count parameter
;   327          0323  3              FAOLST [.ITR_CNT+1] = .ITR_CNT;     !
;   328          0324  3              FAOLST [(.ITR_CNT+1)*2] = .ITR_CNT; !
;   329          0325  3              INCRU IDX FROM 1 TO .ITR_CNT        ! A few bytes at a time
;   330          0326  3              DO
;   331          0327  4                  BEGIN
;   332          0328  4                  FAOLST [.IDX] = .(.PTR) <0, 8, 0>; ! One for the hex
;   333          0329  4                  FAOLST [.IDX + .ITR_CNT+1] = .(.PTR) <0, 8, 0>; ! Decimal
;   334          0330  4                  FAOLST [2*(.IDX + .ITR_CNT)+1] = 1;     ! One for character
;   335          0331  4                  FAOLST [2*(.IDX + .ITR_CNT)+1 + 1] = .PTR;
;   336          0332  4                  PTR = .PTR + 1;             ! Next one
;   337          0333  4                  CTR = .CTR - 1;             ! One less
;   338          0334  4                  END
;   339          0335  3              ;
;   340          0336  3
```

```
;  341          P 0337  3              $FAOL                           ! Saviour of bored programmers
;  342          P 0338  3                  (
;  343          P 0339  3                  CTRSTR = %ASCID '!#(4XB)!/!#(4UB)!/   !#(4AF)!/',
;  344          P 0340  3                  OUTLEN = OUTDSC [0],
;  345          P 0341  3                  OUTBUF = OUTDSC,
;  346          P 0342  3                  PRMLST = FAOLST
;  347            0343  3                  );
;  348            0344  3
;  349            0345  3              LIB$PUT_OUTPUT (OUTDSC);    ! Write to SYS$OUTPUT
;  350            0346  2              END;
;  351            0347  1      END;


                                                                    .PSECT  $PLIT$,NOWRT,NOEXE,2

                                                             0001C P.AAE:   .BLKB   0                                                    ;
                                                  010E0000   0001C P.AAD:   .LONG   17694720                                             ;
                                                  00000000'  00020          .ADDRESS P.AAE                                               ;
74  67  6E  65  6C  28  20  20  53  41  21  20  20  2F  21   00024 P.AAG:   .ASCII  \!/  !AS  (length = !UL bytes)!/  !AS!/-              ; R
21  29  73  65  74  79  62  20  4C  55  21  20  3D  20  68   00033                  \<0>
                    00  2F  21  53  41  21  20  20  2F       00042
                                                  00         0004B          .ASCII  <0>
                                                  010E0026   0004C P.AAF:   .LONG   17694758                                             ;
                                                  00000000'  00050          .ADDRESS P.AAG                                               ;
42  55  34  28  23  21  2F  21  29  42  58  34  28  23  21   00054 P.AAI:   .ASCII  \!#(4XB)!/!#(4UB)!/   !#(4AF)!/\<0><0>               ;
2F  21  29  46  41  34  28  23  21  20  20  20  2F  21  29   00063
                                                  00  00     00072                                                                        ;
                                                  010E001E   00074 P.AAH:   .LONG   17694750                                             ;
                                                  00000000'  00078          .ADDRESS P.AAI                                               ;

                                                                    .EXTRN  SYS$FAOL

                                                                    .PSECT  $CODE$,NOWRT,2

                                                      00FC  00000           .ENTRY  EVL$PRINTLOG, Save R2,R3,R4,R5,R6,R7                ; 0224
                    57 00000000G   00   9E  00002           MOVAB   LIB$PUT_OUTPUT, R7                                                  ;
                    56 00000000G   00   9E  00009           MOVAB   SYS$FAOL, R6                                                        ;
                    5E     FD68    CE   9E  00010           MOVAB   -664(SP), SP                                                        ;
01        0000'     CF       04    AC   E0  00015           BBS     LOGBIT, EVL$GL_LOGMASK, 1$                                          ; 0288
                                    04  0001C               RET                                                                         ;
                    6E     0100    8F   3C  0001D 1$:       MOVZWL  #256, OUTDSC                                                        ; 0293
              04    AE     FF00    CD   9E  00022           MOVAB   FAOBUF, OUTDSC+4                                                    ; 0294
              08    AE       08    AC   D0  00028           MOVL    HEADDSC, FAOLST                                                     ; 0295
                    52       10    AC   D0  0002D           MOVL    DATADSC, R2                                                         ; 0296
              0C    AE             62   3C  00031           MOVZWL  (R2), FAOLST+4                                                      ;
                             0C    AC   D5  00035           TSTL    EXTRADSC                                                            ; 0299
                                   07   12  00038           BNEQ    2$                                                                  ;
                    50     0000'   CF   9E  0003A           MOVAB   P.AAD, R0                                                           ; 0300
                                   04   11  0003F           BRB     3$                                                                  ;
                    50       0C    AC   D0  00041 2$:       MOVL    EXTRADSC, R0                                                        ; 0303
              10    AE             50   D0  00045 3$:       MOVL    R0, FAOLST+8                                                        ; 0298
                             08    AE   9F  00049           PUSHAB  FAOLST                                                              ; 0311
                             04    AE   9F  0004C           PUSHAB  OUTDSC                                                              ;
                             08    AE   9F  0004F           PUSHAB  OUTDSC                                                              ;
                           0000'   CF   9F  00052           PUSHAB  P.AAF                                                               ;
                    66             04   FB  00056           CALLS   #4, SYS$FAOL                                                        ;
```

```
                                 5E   DD 00059          PUSHL   SP                                          ; 0312
                             67      01   FB 0005B      CALLS   #1, LIB$PUT_OUTPUT
                             53      62   3C 0005E      MOVZWL  (R2), CTR                                   ; 0314
                             55  04  A2   D0 00061      MOVL    4(R2), PTR                                  ; 0315
                                     53   D5 00065 4$:   TSTL    CTR                                         ; 0316
                                     72   15 00067      BLEQ    8$
                             6E  0100    8F   3C 00069  MOVZWL  #256, OUTDSC                                ; 0319
                         04  AE  FF00    CD   9E 0006E  MOVAB   FAOBUF, OUTDSC+4                            ; 0320
                             50      53   D0 00074      MOVL    CTR, R0                                     ; 0321
                             14      50   D1 00077      CMPL    R0, #20
                                     03   15 0007A      BLEQ    5$
                             50      14   D0 0007C      MOVL    #20, R0
                             52      50   D0 0007F 5$:   MOVL    R0, ITR_CNT
                         08  AE      52   D0 00082      MOVL    ITR_CNT, FAOLST                             ; 0322
                         0C AE42     52   D0 00086      MOVL    ITR_CNT, FAOLST+4[ITR_CNT]                  ; 0323
                 50          52      01   78 0008B      ASHL    #1, ITR_CNT, R0                             ; 0324
                         10 AE40     52   D0 0008F      MOVL    ITR_CNT, FAOLST+8[R0]
                             51      01   D0 00094      MOVL    #1, IDX                                     ; 0325
                                     26   11 00097      BRB     7$
                         08 AE41     65   9A 00099 6$:   MOVZBL  (PTR), FAOLST[IDX]                        ; 0328
                 50          51      52   C1 0009E      ADDL3   ITR_CNT, IDX, R0                            ; 0329
                         0C AE40     65   9A 000A2      MOVZBL  (PTR), FAOLST+4[R0]
                             54  8142    9E 00CA7      MOVAB   (IDX)+[ITR_CNT], R4                         ; 0330
                 50          54      01   78 000AB      ASHL    #1, R4, R0
                         0C AE40     01   D0 000AF      MOVL    #1, FAOLST+4[R0]
                 50          54      01   78 000B4      ASHL    #1, R4, R0                                  ; 0331
                         10 AE40     85   9E 000B8      MOVAB   (PTR)+, FAOLST+8[R0]
                                     53   D7 000BD      DECL    CTR                                         ; 0333
                             52      51   D1 000BF 7$:   CMPL    IDX, ITR_CNT                                ; 0325
                                     D5   1B 000C2      BLEQU   6$
                                 08  AE   9F 000C4      PUSHAB  FAOLST                                      ; 0343
                                 04  AE   9F 000C7      PUSHAB  OUTDSC
                                 08  AE   9F 000CA      PUSHAB  OUTDSC
                               0000' CF   9F 000CD      PUSHAB  P.AAH
                         66              04   FB 000D1      CALLS   #4, SYS$FAOL
                                         5E   DD 000D4      PUSHL   SP                                      ; 0345
                         67              01   FB 000D6      CALLS   #1, LIB$PUT_OUTPUT
                                         8A   11 000D9      BRB     4$                                      ; 0316
                                              04 000DB 8$:  RET                                             ; 0347

; Routine Size:  220 bytes,    Routine Base:  $CODE$ + 0080
```

```
;  353          0348  1 END                                !End of module
;  354          0349  0 ELUDOM
```

```
;                                  PSECT SUMMARY
;
;       Name                    Bytes                           Attributes
;
;  $GLOBAL$                       8  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $PLIT$                       124  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $CODE$                       348  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
```

```
;                               Library Statistics
;                                        --------  Symbols  --------      Pages       Processing
;       File                             Total     Loaded   Percent      Mapped      Time
;
;  _$255$DUA28:[EVL.OBJ]EVLIBRARY.L32;1    191          2         1          14        00:00.1
;  _$255$DUA28:[SYSLIB]STARLET.L32;1      9776          9         0         581        00:01.3
```

```
;                                    COMMAND QUALIFIERS

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

; Size:          348 code + 132 data bytes
; Run Time:          00:08.7
; Elapsed Time:      00:20.2
; Lines/CPU Min:      2398
; Lexemes/CPU-Min:  18522
; Memory Used:   86 pages
; Compilation Complete
```

**F

EVLJULIAN
LIS

EVCDEF
LIS

ERRMSG
LIS

EVLIBRARY
LIS

EVLTRANS
LIS

EVLSHOW
LIS

EVLMAIN
LIS

RECEIVER
LIS