```
EEEEEEEEEEEEEEE  DDDDDDDDDDDD     TTTTTTTTTTTTTTT
EEEEEEEEEEEEEEE  DDDDDDDDDDDD     TTTTTTTTTTTTTTT
EEEEEEEEEEEEEEE  DDDDDDDDDDDD     TTTTTTTTTTTTTTT
EEE              DDD         DDD        TTT
EEE              DDD         DDD        TTT
EEE              DDD         DDD        TTT
EEE              DDD         DDD        TTT
EEE              DDD         DDD        TTT
EEE              DDD         DDD        TTT
EEEEEEEEEEEE     DDD         DDD        TTT
EEEEEEEEEEEE     DDD         DDD        TTT
EEEEEEEEEEEE     DDD         DDD        TTT
EEE              DDD         DDD        TTT
EEE              DDD         DDD        TTT
EEE              DDD         DDD        TTT
EEE              DDD         DDD        TTT
EEE              DDD         DDD        TTT
EEE              DDD         DDD        TTT
EEEEEEEEEEEEEEE  DDDDDDDDDDDD           TTT
EEEEEEEEEEEEEEE  DDDDDDDDDDDD           TTT
EEEEEEEEEEEEEEE  DDDDDDDDDDDD           TTT
```

```
EX

Mo
--
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
SY
LB
LI
```

```
**FILE**ID**WRIEDTMSG                                        M  2
```

```
WW      WW  RRRRRRRR      IIIIII    EEEEEEEEEE  DDDDDDDD    TTTTTTTTTT  MM      MM    SSSSSSSS    GGGGGGGG
WW      WW  RRRRRRRR      IIIIII    EEEEEEEEEE  DDDDDDDD    TTTTTTTTTT  MM      MM    SSSSSSSS    GGGGGGGG
WW      WW  RR      RR      II      EE          DD      DD      TT      MMMM  MMMM  SS          GG
WW      WW  RR      RR      II      EE          DD      DD      TT      MMMM  MMMM  SS          GG
WW      WW  RR      RR      II      EE          DD      DD      TT      MM  MM  MM  SS          GG
WW      WW  RR      RR      II      EE          DD      DD      TT      MM  MM  MM  SS          GG
WW      WW  RRRRRRRR        II      EEEEEEEE    DD      DD      TT      MM      MM    SSSSSS    GG
WW      WW  RRRRRRRR        II      EEEEEEEE    DD      DD      TT      MM      MM    SSSSSS    GG
WW  WW  WW  RR  RR          II      EE          DD      DD      TT      MM      MM          SS  GG  GGGGGG
WW  WW  WW  RR  RR          II      EE          DD      DD      TT      MM      MM          SS  GG  GGGGGG
WWWW  WWWW  RR    RR        II      EE          DD      DD      TT      MM      MM          SS  GG      GG      ....
WWWW  WWWW  RR    RR        II      EE          DD      DD      TT      MM      MM          SS  GG      GG      ....
WW      WW  RR      RR    IIIIII    EEEEEEEEEE  DDDDDDDD        TT      MM      MM  SSSSSSSS      GGGGGG        ....
WW      WW  RR      RR    IIIIII    EEEEEEEEEE  DDDDDDDD        TT      MM      MM  SSSSSSSS      GGGGGG        ....


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
```

```
EDT
V04-
56
74
74
72
73
53
20
6E
2E
20
27
45
00
```

```
0 %TITLE 'EDT$WRIEDTMSG - write VMSMSG.MSG'
0 MODULE EDT$WRIEDTMSG (                                 ! Write VMSMSG.MSG
0                   IDENT = 'V04-000',                   ! File: WRIEDTMSG.B32 Edit: JBS1007
0                   MAIN = EDT$WRIEDTMSG
0                   ) =
1 BEGIN
1 !
1 !****************************************************************************
1 !*                                                                          *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
1 !*  ALL RIGHTS RESERVED.                                                    *
1 !*                                                                          *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
1 !*  TRANSFERRED.                                                            *
1 !*                                                                          *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
1 !*  CORPORATION.                                                            *
1 !*                                                                          *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
1 !*                                                                          *
1 !*                                                                          *
1 !****************************************************************************
1 !
1
1 !++
1 ! FACILITY:     EDT -- The DEC Standard Editor
1 !
1 ! ABSTRACT:
1 !
1 !       This module, WRIEDTMSG.FOR, is a FORTRAN program that writes
1 !       the file VMSMSG.MSG, which is read by the message compiler
1 !       to produce EDT's run-time messages.
1 !
1 ! ENVIRONMENT:  Runs at any access mode - AST reentrant
1 !
1 ! AUTHOR: John Sauter, CREATION DATE: 23-Jul-1981
1 !
1 ! MODIFIED BY:
1 !
1 ! 1-001 - Original, from BASMSG.FOR, created November 3, 1978, last
1 !         revised September 24, 1979 (version 1-015).  JBS 28-Jul-1981
1 ! 1-002 - Don't omit the first message.  JBS 03-Aug-1981
1 ! 1-003 - Change output file name to VMSMSG.MSG.  JBS 03-Aug-1981
1 ! 1-004 - Fix output file's module name   JBS 06-Aug-1981
1 ! 1-005 - Recoded in BLISS since VMS doesn't like its components to be
1 !         dependent upon Fortran.  JBS 22-Oct-1981
1 ! 1-006 - Change output file name to VMSMSG.TMP. BLS 6-May-1983
1 ! 1-007 - Correct the module header and trailer.  JBS 09-May-1983
1 !--
1
```

```
1 %SBTTL 'Declarations'
1 !
1 ! SWITCHES:
1 !
1
1 SWITCHES ADDRESSING_MODE (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);
1
1 !
1 ! LINKAGES:
1 !
1 !      NONE
1 !
1 ! TABLE OF CONTENTS:
1 !
1
1 FORWARD ROUTINE
1     EDT$WRIEDTMSG,                              ! Write VMSMSG.TMP
1     WRITE_FILE,                                 ! Actually write the text
1     PRINT,                                      ! Print a line of text
1     HEX_TEXT,                                   ! Convert binary to hexadecimal
1     PRINTABLE_TEXT;                             ! Convert binary to ASCII, printable
1
1 !
1 ! INCLUDE FILES:
1 !
1
1 REQUIRE 'EDTSRC:PSECTS.REQ';                    ! Define PSECT declaration macros
1
1 REQUIRE 'EDTSRC:SYSSYM.REQ';                    ! Define system symbols
1
1 !
1 ! MACROS:
1 !
1 !      NONE
1 !
1 ! EQUATED SYMBOLS:
1 !
1
1 LITERAL
1     EDT$K_FAC_NO = 133;                         ! Facility number, for signaling.
1
1 !
1 ! FIELDS:
1 !
1 !      NONE
1 !
1 ! STRUCTURES:
1 !
1 !      NONE
1 !
1 ! PSECTS:
1 !
1 DECLARE_PSECTS (EDT);                           ! Declare PSECTs for EDT$ facility
1 !
1 ! OWN STORAGE:
1 !
1 !      NONE
```

```
;               0248  1 !
;               0249  1 ! EXTERNAL REFERENCES:
;               0250  1 !
;               0251  1
;               0252  1 EXTERNAL ROUTINE
;               0253  1     STR$COPY_DX,                        ! Copy a string, by descriptor
;               0254  1     STR$CONCAT,                         ! Concatenate strings
;               0255  1     LIB$GET_INPUT,                      ! Get a line from SYS$INPUT
;               0256  1     STR$COPY_R,                         ! Copy a string, by reference
;               0257  1     STR$FREE1_DX,                       ! Free a dynamic string
;               0258  1     EDT$MSGTXT;                         ! Return the text of a message
;               0259  1
```

```
   1 %SBTTL 'Package of macros for string processing'
   1 !+
   1 ! Macro to initialize a dynamic descriptor.
   1 !-
   1
   1 MACRO
M  1     INIT_DESCRIPTOR (DESCR) =
M  1         DESCR [DSC$W_LENGTH] = 0;
M  1         DESCR [DSC$B_DTYPE] = DSC$K_DTYPE_T;
M  1         DESCR [DSC$B_CLASS] = DSC$K_CLASS_D;
M  1         DESCR [DSC$A_POINTER] = 0;
   1 %,
   1 !<BLF/MACRO>
   1 !+
   1 ! Macro to discard a dynamic descriptor.
   1 !-
M  1     DISCARD_DESCRIPTOR (DESCR) =
M  1         BEGIN
M  1
M  1         LOCAL
M  1             FREE_STATUS;
M  1
M  1         FREE_STATUS = STR$FREE1_DX (DESCR);
M  1
M  1         IF ( NOT .FREE_STATUS) THEN SIGNAL_STOP (.FREE_STATUS);
M  1
M  1         END;
   1     %,
   1 !+
   1 ! Macro to build a text line using FAO.  This is a convenience macro.
   1 !-
M  1     BUILD_TEXT_LINE (DESCR, CTL_STRING, FAO_ARGS) =
M  1         BEGIN
M  1
M  1         LOCAL
M  1             FAO_STATUS,
M  1             COPY_STATUS;
M  1
M  1         CTL_STR_DSC [DSC$W_LENGTH] = %CHARCOUNT (CTL_STRING);
M  1         CTL_STR_DSC [DSC$B_DTYPE] = DSC$K_DTYPE_T;
M  1         CTL_STR_DSC [DSC$B_CLASS] = DSC$K_CLASS_S;
M  1         CTL_STR_DSC [DSC$A_POINTER] = CH$PTR (UPLIT (CTL_STRING));
M  1         FAO_STATUS = $FAO (                             !
M  1             CTL_STR_DSC,                                !
M  1             OUT_LENGTH,                                 !
M  1             TEMP_STR_DSC,                               !
M  1             %REMOVE (FAO_ARGS));
M  1
M  1         IF ( NOT .FAO_STATUS) THEN SIGNAL_STOP (.FAO_STATUS);
M  1
M  1         COPY_STATUS = STR$COPY_R (DESCR, OUT_LENGTH, .TEMP_STR_DSC [DSC$A_POINTER]);
M  1         .COPY_STATUS
M  1         END
   1     %,
   1 !+
   1 ! Macro to format and print a line.  Errors are returned to the caller.
   1 ! This is a convenience macro.
```

```
;  186           0317  1 !-
;  187         M 0318  1      PRINT_LINE (TEXT, VARS) =
;  188         M 0319  1          BEGIN
;  189         M 0320  1
;  190         M 0321  1          LOCAL
;  191         M 0322  1              BUILD_STATUS,
;  192         M 0323  1              PRINT_STATUS;
;  193         M 0324  1
;  194         M 0325  1          BUILD_STATUS = BUILD_TEXT_LINE (LINE_DESC, %STRING (%REMOVE (TEXT)), VARS);
;  195         M 0326  1
;  196         M 0327  1          IF ( NOT .BUILD_STATUS) THEN RETURN (.BUILD_STATUS);
;  197         M 0328  1
;  198         M 0329  1          PRINT_STATUS = PRINT (.OUTPUT_RAB, LINE_DESC);
;  199         M 0330  1
;  200         M 0331  1          IF ( NOT .PRINT_STATUS) THEN RETURN (.PRINT_STATUS);
;  201         M 0332  1
;  202         M 0333  1          END
;  203           0334  1      %;
;  204           0335  1
```

```
  1 %SBTTL 'EDT$WRIEDTMSG - Write VMSMSG.TMP'
  1 ROUTINE EDT$WRIEDTMSG                               ! Write VMSMSG.TMP
  1     =
  1
  1 !++
  1 ! FUNCTIONAL DESCRIPTION:
  1 !
  1 !       This routine writes the file VMSMSG.TMP.
  1 !
  1 ! CALLING SEQUENCE:
  1 !
  1 !       ret_status.wlc.v = EDT$WRIEDTMSG ()
  1 !
  1 ! FORMAL PARAMETERS:
  1 !
  1 !       NONE
  1 !
  1 ! IMPLICIT INPUTS:
  1 !
  1 !       NONE
  1 !
  1 ! IMPLICIT OUTPUTS:
  1 !
  1 !       NONE
  1 !
  1 ! COMPLETION STATUS:
  1 !
  1 !       SS$_NORMAL      Normal successful completion
  1 !       Any error from LIB$GET_INPUT or STR$FREE1_DX
  1 !
  1 ! SIDE EFFECTS:
  1 !
  1 !       Writes a file.
  1 !       Any errors from RMS$CREATE, RMS$OPEN, RMS$CONNECT or RMS$CLOSE
  1 !        are signalled.
  1 !
  1 !--
  1
  2     BEGIN
  2
  2     LOCAL
  2         OUTPUT_BUFFER : BLOCK [132, BYTE],       ! output buffer, for RMS
  2         OUTPUT_FAB : $FAB_DECL,                  ! RMS FAB for the output file
  2         OUTPUT_NAM : $NAM_DECL,                  ! RMS NAM for the output file
  2         OUTPUT_RAB : $RAB_DECL,                  ! RMS RAB for the output file
  2         OUTPUT_FILE_NAME_DESC : BLOCK [8, BYTE],         ! Name of output file
  2         OUTPUT_RESULT_NAME : BLOCK [NAM$C_MAXRSS, BYTE];         ! Place to store output file name
  2
  2 !
  2     OUTPUT_FILE_NAME_DESC [DSC$W_LENGTH] = %CHARCOUNT ('VMSMSG');
  2     OUTPUT_FILE_NAME_DESC [DSC$B_DTYPE] = DSC$K_DTYPE_T;
  2     OUTPUT_FILE_NAME_DESC [DSC$B_CLASS] = DSC$K_CLASS_S;
  2     OUTPUT_FILE_NAME_DESC [DSC$A_POINTER] = UPLIT ('VMSMSG');
  2 !+
  2 ! Initialize the FAB, NAM and RAB for the output file
  2 !-
P 2     $FAB_INIT (FAB = OUTPUT_FAB,                 !
            FAC = (PUT),
            FOP = (OFP, SQO, DFW),                    !
            ORG = SEQ,                                !
            SHR = NIL,
            MRS = 132,                                !
            RAT = CR,                                 !
            RFM = VAR,                                !
            FNA = .OUTPUT_FILE_NAME_DESC [DSC$A_POINTER],   !
            FNS = .OUTPUT_FILE_NAME_DESC [DSC$W_LENGTH],    !
            DNA = UPLIT ('EDTSRC:.TMP'),              !
            DNS = %CHARCOUNT ('EDTSRC:.TMP'),         !
            NAM = OUTPUT_NAM);
        $NAM_INIT (NAM = OUTPUT_NAM,                  !
            RSA = OUTPUT_RESULT_NAME,                 !
            RSS = NAM$C_MAXRSS);
        $RAB_INIT (RAB = OUTPUT_RAB,                  !
            RAC = SEQ,                                !
            ROP = WBH,                                !
            USZ = 132,                                !
            UBF = OUTPUT_BUFFER,                      !
            FAB = OUTPUT_FAB);
    !+
    ! Create the output file, and do the $CONNECT.
    !-
        BEGIN

        LOCAL
            [illegible]_STATUS,
            [illegible]T_STATUS;

        CREATE_STATUS = $CREATE (FAB = OUTPUT_FAB);

        IF ( NOT .CREATE_STATUS)
        THEN
            SIGNAL_STOP (                             !
                SHR$_OPENOUT + (EDT$K_FAC_NO*65536) + STS$K_SEVERE,       !
                1,
                OUTPUT_FILE_NAME_DESC,                !
                .OUTPUT_FAB [FAB$L_STS], .OUTPUT_FAB [FAB$L_STV]);

        CONNECT_STATUS = $CONNECT (RAB = OUTPUT_RAB);

        IF ( NOT .CONNECT_STATUS)
        THEN
            SIGNAL_STOP (                             !
                SHR$_OPENOUT + (EDT$K_FAC_NO*65536) + STS$K_SEVERE,       !
                1,
                OUTPUT_FILE_NAME_DESC,                !
                .OUTPUT_RAB [RAB$L_STS], .OUTPUT_RAB [RAB$L_STV]);

        END;
    !+
    ! Point the file name descriptor to the resultant name string.
    !-
        OUTPUT_FILE_NAME_DESC [DSC$W_LENGTH] = .OUTPUT_NAM [NAM$B_RSL];
        OUTPUT_FILE_NAME_DESC [DSC$B_DTYPE] = DSC$K_DTYPE_T;
        OUTPUT_FILE_NAME_DESC [DSC$B_CLASS] = DSC$K_CLASS_S;
```

```
;   320        0450  2      OUTPUT_FILE_NAME_DESC [DSC$A_POINTER] = .OUTPUT_NAM [NAM$L_RSA];
;   321        0451  2
;   322        0452  3      IF ( NOT WRITE_FILE (OUTPUT_RAB))
;   323        0453  2      THEN
;   324        0454  2          SIGNAL_STOP (                              !
;   325        0455  2              SHR$_WRITEERR + (EDT$K_FAC_NO*65536) + STS$K_SEVERE,          !
;   326        0456  2              1,                                     !
;   327        0457  2              OUTPUT_FILE_NAME_DESC,                 !
;   328        0458  2              .OUTPUT_RAB [RAB$L_STS], .OUTPUT_RAB [RAB$L_STV]);
;   329        0459  2
;   330        0460  3      BEGIN
;   331        0461  3  !+
;   332        0462  3  ! Close the output file.
;   333        0463  3  !-
;   334        0464  3
;   335        0465  3      LOCAL
;   336        0466  3          CLOSE_STATUS;
;   337        0467  3
;   338        0468  3      CLOSE_STATUS = $CLOSE (FAB = OUTPUT_FAB);
;   339        0469  3
;   340        0470  4      IF ( NOT .CLOSE_STATUS)
;   341        0471  3      THEN
;   342        0472  3          SIGNAL_STOP (                              !
;   343        0473  3              SHR$_CLOSEOUT + (EDT$K_FAC_NO*65536) + STS$K_SEVERE,          !
;   344        0474  3              1,                                     !
;   345        0475  3              OUTPUT_FILE_NAME_DESC,                 !
;   346        0476  3              .OUTPUT_FAB [FAB$L_STS], .OUTPUT_FAB [FAB$L_STV]);
;   347        0477  3
;   348        0478  2      END;
;   349        0479  2      RETURN (SS$_NORMAL);
;   350        0480  1      END;                                        ! End of routine EDT$WRIEDTMSG


                                                                        .TITLE  EDT$WRIEDTMSG EDT$WRIEDTMSG - write VMSMSG.MSG
                                                                        .IDENT  \V04-000\

                                                                        .PSECT  _EDT$CODE,NOWRT,  SHR,  PIC,2

                              00  00  47  53  4D  53  4D  56  00000 P.AAA:  .ASCII  \VMSMSG\<0><0>                                    ;
         00  50  4D  54  2E  3A  43  52  53  54  44  45  00008 P.AAB:  .ASCII  \EDTSRC:.TMP\<0>                                  ;

                                                                        .EXTRN  STR$COPY_DX, STR$CONCAT
                                                                        .EXTRN  LIB$GET_INPUT, STR$COPY_R
                                                                        .EXTRN  STR$FREE1_DX, EDT$MSGTXT
                                                                        .EXTRN  SYS$CREATE, SYS$CONNECT
                                                                        .EXTRN  SYS$CLOSE

                                                          007C 00000 EDT$WRIEDTMSG:
                                                                        .WORD   Save R2,R3,R4,R5,R6                              ; 0337
                                    56 00000000G  00  9E 00002          MOVAB   LIB$STOP, R6                                     ;
                                    5E      FD80  CE  9E 00009          MOVAB   -640(SP), SP                                     ;
                             0100   CE 010E0006   8F  D0 0000E          MOVL    #17694726, OUTPUT_FILE_NAME_DESC                 ; 0385
                             0104   CE        D2  AF  9E 00017          MOVAB   P.AAA, OUTPUT_FILE_NAME_DESC+4                   ; 0388
   0050    8F             00        6E            00  2C 0001D          MOVC5   #0, (SP), #0, #80, $RMS_PTR                      ; 0404
                                          FF2C    CD     00024                                                                   ;
                             FF2C   CD      5003  8F  B0 00027          MOVW    #20483, $RMS_PTR                                 ;
                             FF30   CD 20000060   8F  D0 0002E          MOVL    #536871008, $RMS_PTR+4                           ;
```

```
                         FF42   CD      2001   8F  B0 00037          MOVW     #8193, $RMS_PTR+22                            :
                         FF49   CD      0200   8F  B0 0003E          MOVW     #512, $RMS_PTR+29                             :
                         FF4B   CD             02  90 00045          MOVB     #2, $RMS_PTR+31                               :
                         FF54   CD      FECC   CD  9E 0004A          MOVAB    OUTPUT_NAM, $RMS_PTR+40                       :
                         FF58   CD      0104   CE  D0 00051          MOVL     OUTPUT_FILE_NAME_DESC+4, $RMS_PTR+44          :
                         FF5C   CD        99   AF  9E 00058          MOVAB    P.AAB, $RMS_PTR+48                            :
                         FF60   CD      0100   CE  90 0005E          MOVB     OUTPUT_FILE_NAME_DESC, $RMS_PTR+52            :
                         FF61   CD             0B  90 00065          MOVB     #11, $RMS_PTR+53                              :
                         FF62   CD        84   8F  9B 0006A          MOVZBW   #132, $RMS_PTR+54                             :
    0060   8F         00        6E             00  2C 00070          MOVC5    #0, (SP), #0, #96, $RMS_PTR                   : 0407
                                        FECC   CD     00077                                                             :
                         FECC   CD      6002   8F  B0 0007A          MOVW     #24578, $RMS_PTR                              :
                         FECE   CD             01  8E 00081          MNEGB    #1, $RMS_PTR+2                                :
                         FED0   CD             6E  9E 00086          MOVAB    OUTPUT_RESULT_NAME, $RMS_PTR+4                :
    0044   8F         00        6E             00  2C 0008B          MOVC5    #0, (SP), #0, #68, $RMS_PTR                   : 0413
                                        0108   CE     00092                                                             :
                         0108   CE      4401   8F  B0 00095          MOVW     #17409, $RMS_PTR                              :
                         010C   CE      0400   8F  3C 0009C          MOVZWL   #1024, $RMS_PTR+4                             :
                                        0126   CE  94 000A3          CLRB     $RMS_PTR+30                                   :
                         0128   CE        84   8F  9B 000A7          MOVZBW   #132, $RMS_PTR+32                             :
                         012C   CE      FF7C   CD  9E 000AD          MOVAB    OUTPUT_BUFFER, $RMS_PTR+36                    :
                         FEC4   CD      FF2C   CD  9E 000B4          MOVAB    OUTPUT_FAB, $RMS_PTR+60                       :
                                        FF2C   CD  9F 000BB          PUSHAB   OUTPUT_FAB                                    : 0423
                    00000000G   00             01  FB 000BF          CALLS    #1, SYS$CREATE                                :
                                14             50  E8 000C6          BLBS     CREATE_STATUS, 1$                             : 0425
                                7E      FF34   CD  7D 000C9          MOVQ     OUTPUT_FAB+8, -(SP)                           : 0431
                                        0108   CE  9F 000CE          PUSHAB   OUTPUT_FILE_NAME_DESC                         : 0427
                                               01  DD 000D2          PUSHL    #1                                            :
                                    008510A4   8F  DD 000D4          PUSHL    #8720548                                      : 0428
                                66             05  FB 000DA          CALLS    #5, LIB$STOP                                  :
                                        0108   CE  9F 000DD 1$:      PUSHAB   OUTPUT_RAB                                    : 0433
                    00000000G   00             01  FB 000E1          CALLS    #1, SYS$CONNECT                               :
                                17             50  E8 000E8          BLBS     CONNECT_STATUS, 2$                            : 0435
                                        0114   CE  DD 000EB          PUSHL    OUTPUT_RAB+12                                 : 0441
                                        0114   CE  DD 000EF          PUSHL    OUTPUT_RAB+8                                  :
                                        0108   CE  9F 000F3          PUSHAB   OUTPUT_FILE_NAME_DESC                         : 0437
                                               01  DD 000F7          PUSHL    #1                                            :
                                    00851CA4   8F  DD 000F9          PUSHL    #8720548                                      : 0438
                                66             05  FB 000FF          CALLS    #5, LIB$STOP                                  :
                         0100   CE      FECF   CD  9B 00102 2$:      MOVZBW   OUTPUT_NAM+3, OUTPUT_FILE_NAME_DESC           : 0447
                         0102   CE      010E   8F  B0 00109          MOVW     #270, OUTPUT_FILE_NAME_DESC+2                 : 0448
                         0104   CE      FED0   CD  D0 00110          MOVL     OUTPUT_NAM+4, OUTPUT_FILE_NAME_DESC+4         : 0450
                                        0108   CE  9F 00117          PUSHAB   OUTPUT_RAB                                    : 0452
                         0000V  CF             01  FB 0011B          CALLS    #1, WRITE_FILE                                :
                                17             50  E8 00120          BLBS     R0, 3$                                        :
                                        0114   CE  DD 00123          PUSHL    OUTPUT_RAB+12                                 : 0458
                                        0114   CE  DD 00127          PUSHL    OUTPUT_RAB+8                                  :
                                        0108   CE  9F 0012B          PUSHAB   OUTPUT_FILE_NAME_DESC                         : 0454
                                               01  DD 0012F          PUSHL    #1                                            :
                                    008510D4   8F  DD 00131          PUSHL    #8720596                                      : 0455
                                66             05  FB 00137          CALLS    #5, LIB$STOP                                  :
                                        FF2C   CD  9F 0013A 3$:      PUSHAB   OUTPUT_FAB                                    : 0468
                    00000000G   00             01  FB 0013E          CALLS    #1, SYS$CLOSE                                 :
                                14             50  E8 00145          BLBS     CLOSE_STATUS, 4$                              : 0470
                                7E      FF34   CD  7D 00148          MOVQ     OUTPUT_FAB+8, -(SP)                           : 0476
                                        0108   CE  9F 0014D          PUSHAB   OUTPUT_FILE_NAME_DESC                         : 0472
                                               01  DD 00151          PUSHL    #1                                            :
```

```
                        0085105C    8F  DD 00153          PUSHL   #8720476                            ; 0473
                  66                05  FB 00159          CALLS   #5, LIB$STOP                        ;
                  50                01  D0 0015C 4$:      MOVL    #1, R0                              ; 0479
                                        04 0015F          RET                                         ; 0480

; Routine Size:  352 bytes,    Routine Base:  _EDT$CODE + 0014
```

```
1 %SBTTL 'WRITE_FILE - Actually write the file'
1 ROUTINE WRITE_FILE (                                  ! Actually write the file
1     OUTPUT_RAB                                        ! Where to write each record
1     ) =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine writes each record on the specified RAB.
1 !
1 ! CALLING SEQUENCE:
1 !
1 !       ret_status.wlc.v = WRITE_FILE (OUTPUT_RAB.mz.r)
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       OUTPUT_RAB              RAB onto which to write the text
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       None
1 !
1 ! COMPLETION STATUS:
1 !
1 !       SS$_NORMAL      Normal successful completion
1 !       Any errors from RMS $PUT
1 !
1 ! SIDE EFFECTS:
1 !
1 !       Writes on the file connected to OUTPUT_RAB
1 !
1 !--
1
2     BEGIN
2
2     MAP
2       OUTPUT_RAB : REF $RAB_DECL;
2
2     LOCAL
2 !+
2 ! Stuff for BUILD_TEXT_LINE
2 !-
2       CTL_STR_DSC : BLOCK [8, BYTE],
2       TEMP_STR_DSC : BLOCK [8, BYTE],
2       TEMP_STRING : VECTOR [132, BYTE],
2       OUT_LENGTH,
2 !+
2 ! Stuff for PRINT_LINE
2 !-
2       LINE_DESC : BLOCK [8, BYTE],
2 !+
2 ! End of stuff for PRINT_LINE
2 !-
```

```
2               PRINTABLE_DESC : BLOCK [8, BYTE],
2               HEX_DESC : BLOCK [8, BYTE],
2               TEXT_DESC : BLOCK [8, BYTE],
2               NAME_DESC : BLOCK [8, BYTE],
2               SEVERITY_DESC : BLOCK [8, BYTE],
2               NAME_LENGTH,
2               TEXT_LENGTH,
2               SEVERITY_LENGTH,
2               SEVERITY_ADDR : REF VECTOR [, BYTE];
2
2   !+
2   ! Set up TEMP_STR_DSC for BUILD_TEXT_LINE
2   !-
2           TEMP_STR_DSC [DSC$W_LENGTH] = 132;
2           TEMP_STR_DSC [DSC$B_DTYPE] = DSC$K_DTYPE_T;
2           TEMP_STR_DSC [DSC$B_CLASS] = DSC$K_CLASS_S;
2           TEMP_STR_DSC [DSC$A_POINTER] = CH$PTR (TEMP_STRING);
2   !+
2   ! Set up LINE_DESC for PRINT_LINE, etc.
2   !-
2           INIT_DESCRIPTOR (LINE_DESC);
2           INIT_DESCRIPTOR (PRINTABLE_DESC);
2           INIT_DESCRIPTOR (HEX_DESC);
2           INIT_DESCRIPTOR (TEXT_DESC);
2           INIT_DESCRIPTOR (NAME_DESC);
2           INIT_DESCRIPTOR (SEVERITY_DESC);
2   !+
2   ! Put out the initial information.
2   !-
2           PRINT_LINE (<'!! This file, VMSMSG.TMP, contains the definitions of the EDT'>, <' '>);
2           PRINT_LINE (<'!! messages for VAX/VMS.  This file is read by the MESSAGE compiler'>, <' '>);
2           PRINT_LINE (<'!! to build an object file containing the EDT messages.'>, <' '>);
2           PRINT_LINE (<'!!'>, <' '>);
2           PRINT_LINE (<'        .TITLE EDT$VMSMSG EDT''s message text'>, <' '>);
2           PRINT_LINE (<'!!'>, <' '>);
2           PRINT_LINE (<'.FACILITY/SYSTEM EDT, !SL'>, <EDT$K_FAC_NO>);
2   !+
2   ! Write a line for each message
2   !-
2
2           INCR CODE FROM 0 TO 4095 DO
3               BEGIN
3               EDT$MSGTXT (CODE, SEVERITY_DESC, NAME_LENGTH, NAME_DESC, TEXT_LENGTH, TEXT_DESC);
3   !+
3   ! If the severity field is blank we are done.
3   !-
3               SEVERITY_ADDR = .SEVERITY_DESC [DSC$A_POINTER];
3
3               IF (.SEVERITY_ADDR [0] EQL ' ') THEN EXITLOOP;
3
3               PRINT_LINE (<'!AS/!AS <!AS>'>, <NAME_DESC, SEVERITY_DESC, TEXT_DESC>);
2               END;
2
2   !+
2   ! Write out the trailer line
2   !-
2           PRINT_LINE (<'.END'>, <' '>);
```

```
:  466          0595  2 !+
:  467          0596  2 ! All done.
:  468          0597  2 !-
:  469          0598  2     DISCARD_DESCRIPTOR (LINE_DESC);
:  470          0599  2     DISCARD_DESCRIPTOR (PRINTABLE_DESC);
:  471          0600  2     DISCARD_DESCRIPTOR (HEX_DESC);
:  472          0601  2     DISCARD_DESCRIPTOR (TEXT_DESC);
:  473          0602  2     DISCARD_DESCRIPTOR (NAME_DESC);
:  474          0603  2     DISCARD_DESCRIPTOR (SEVERITY_DESC);
:  475          0604  2     RETURN (SS$_NORMAL);
:  476          0605  1     END;                                  ! End of routine WRITE_FILE


56 20 2C 65 6C 69 66 20 73 69 68 54 20 21 21 00174 P.AAC:  .ASCII  \!! This file, VMSMSG.TMP, contains the d\  ;
74 6E 6F 63 20 2C 50 4D 54 2E 47 53 4D 53 4D 00183                                                            ;
               64 20 65 68 74 20 73 6E 69 61 00192                                                            ;
74 20 66 6F 20 73 6E 6F 69 74 69 6E 69 66 65 0019C         .ASCII  \efinitions of the EDT\<0><0><0>           ;
                  00 00 00 54 44 45 20 65 68 001AB                                                            ;
72 6F 66 20 73 65 67 61 73 73 65 6D 20 21 21 001B4 P.AAD:  .ASCII  \!! messages for VAX/VMS.  This file is r\  ;
73 69 68 54 20 20 2E 53 4D 56 2F 58 41 56 20 001C3                                                            ;
               72 20 73 69 20 65 6C 69 66 20 001D2                                                            ;
53 53 45 4D 20 65 68 74 20 79 62 20 64 61 65 001DC         .ASCII  \ead by the MESSAGE compiler\<0>           ;
      00 72 65 6C 69 70 6D 6F 63 20 45 47 41 001EB                                                            ;
20 6E 61 20 64 6C 69 75 62 20 6F 74 20 21 21 001F8 P.AAE:  .ASCII  \!! to build an object file containing th\  ;
6E 6F 63 20 65 6C 69 66 20 74 63 65 6A 62 6F 00207                                                            ;
               68 74 20 67 6E 69 6E 69 61 74 00216                                                            ;
2E 73 65 67 61 73 73 65 6D 20 54 44 45 20 65 00220         .ASCII  \e EDT messages.\<0>                       ;
                                             00 0022F                                                            ;
                                    00 00 21 21 00230 P.AAF:  .ASCII  \!!\<0><0>                                 ;
20 45 4C 54 49 54 2E 20 20 20 20 20 20 20 20 00234 P.AAG:  .ASCII  \        .TITLE EDT$VMSMSG EDT's message \  ;
27 54 44 45 20 47 53 4D 53 4D 56 24 54 44 45 00243                                                            ;
               20 65 67 61 73 73 65 6D 20 73 00252                                                            ;
                                    74 78 65 74 0025C         .ASCII  \text\                                      ;
                                    00 00 21 21 00260 P.AAH:  .ASCII  \!!\<0><0>                                 ;
45 54 53 59 53 2F 59 54 49 4C 49 43 41 46 2E 00264 P.AAI:  .ASCII  \.FACILITY/SYSTEM EDT, !SL\<0><0><0>        ;
      00 00 00 4C 53 21 20 2C 54 44 45 20 4D 00273                                                            ;
00 00 3E 53 41 21 3C 20 53 41 21 2F 53 41 21 00280 P.AAJ:  .ASCII  \!AS/!AS <!AS>\<0><0><0>                    ;
                                             00 0028F                                                            ;
                                    44 4E 45 2E 00290 P.AAK:  .ASCII  \.END\                                      ;

                                                            .EXTRN  SYS$FAO

                                       01FC 00000 WRITE_FILE:
                                                            .WORD   Save R2,R3,R4,R5,R6,R7,R8                  ; 0482
                    58      0000V  CF  9E 00002             MOVAB   PRINT, R8                                  ;
                    57 00000000G   00  9E 00007             MOVAB   STR$FREE1_DX, R7                           ;
                    56 00000000G   00  9E 0000E             MOVAB   STR$COPY_R, R6                             ;
                    55 00000000G   00  9E 00015             MOVAB   SYS$FAO, R5                                ;
                    54 00000000G   00  9E 0001C             MOVAB   LIB$STOP, R4                               ;
                    5E      FF2C   CE  9E 00023             MOVAB   -212(SP), SP                               ;
              F0    AD 010E0084    8F  D0 00028             MOVL    #17694852, TEMP_STR_DSC                    ; 0551
              F4    AD        40   AE  9E 00030             MOVAB   TEMP_STRING, TEMP_STR_DSC+4                ; 0554
              38    AE 020E0000    8F  D0 00035             MOVL    #34471936, LINE_DESC                       ; 0558
                              3C   AE  D4 0003D             CLRL    LINE_DESC+4                                ;
              30    AE 020E0000    8F  D0 00040             MOVL    #34471936, PRINTABLE_DESC                  ; 0559
                              34   AE  D4 00048             CLRL    PRINTABLE_DESC+4                           ;
```

```
       28    AE 020E0000   8F   D0 0004B          MOVL     #34471936, HEX_DESC                      ; 0560
                      2C   AE   D4 00053          CLRL     HEX_DESC+4                               ;
       20    AE 020E0000   8F   D0 00056          MOVL     #34471936, TEXT_DESC                     ; 0561
                      24   AE   D4 0005E          CLRL     TEXT_DESC+4                              ;
       18    .E 020E0000   8F   D0 00061          MOVL     #34471936, NAME_DESC                     ; 0562
                      1C   AE   D4 00069          CLRL     NAME_DESC+4                              ;
       10    AE 020E0000   8F   D0 0006C          MOVL     #34471936, SEVERITY_DESC                 ; 0563
                      14   AE   D4 00074          CLRL     SEVERITY_DESC+4                          ;
       F8    AD 010E003D   8F   D0 00077          MOVL     #17694781, CTL_STR_DSC                   ; 0567
       FC    AD     FE5D   CF   9E 0007F          MOVAB    P.AAC, CTL_STR_DSC+4                     ;
                           20   DD 00085          PUSHL    #32                                      ;
                      F0   AD   9F 00087          PUSHAB   TEMP_STR_DSC                             ;
                      14   AE   9F 0008A          PUSHAB   OUT_LENGTH                               ;
                      F8   AD   9F 0008D          PUSHAB   CTL_STR_DSC                              ;
             65            04   FB 00090          CALLS    #4, SYS$FAO                              ;
             05            50   E8 00093          BLBS     FAO_STATUS, 1$                           ;
                           50   DD 00096          PUSHL    FAO_STATUS                               ;
             64            01   FB 00098          CALLS    #1, LIB$STOP                             ;
                      F4   AD   DD 0009B 1$:      PUSHL    TEMP_STR_DSC+4                           ;
                      10   AE   9F 0009E          PUSHAB   OUT_LENGTH                               ;
                      40   AE   9F 000A1          PUSHAB   LINE_DESC                                ;
             66            03   FB 000A4          CALLS    #3, STR$COPY_R                           ;
             7D            50   E9 000A7          BLBC     BUILD_STATUS, 4$                         ;
                      38   AE   9F 000AA          PUSHAB   LINE_DESC                                ;
             52       04   AC   D0 000AD          MOVL     OUTPUT_RAB, R2                           ;
                           52   DD 000B1          PUSHL    R2                                       ;
             68            02   FB 000B3          CALLS    #2, PRINT                                ;
             79            50   E9 000B6          BLBC     PRINT_STATUS, 5$                         ;
       F8    AD 010E0043   8F   D0 000B9          MOVL     #17694787, CTL_STR_DSC                   ; 0568
       FC    AD     FE5B   CF   9E 000C1          MOVAB    P.AAD, CTL_STR_DSC+4                     ;
                           20   DD 000C7          PUSHL    #32                                      ;
                      F0   AD   9F 000C9          PUSHAB   TEMP_STR_DSC                             ;
                      14   AE   9F 000CC          PUSHAB   OUT_LENGTH                               ;
                      F8   AD   9F 000CF          PUSHAB   CTL_STR_DSC                              ;
             65            04   FB 000D2          CALLS    #4, SYS$FAO                              ;
             05            50   E8 000D5          BLBS     FAO_STATUS, 2$                           ;
                           50   DD 000D8          PUSHL    FAO_STATUS                               ;
             64            01   FB 000DA          CALLS    #1, LIB$STOP                             ;
                      F4   AD   DD 000DD 2$:      PUSHL    TEMP_STR_DSC+4                           ;
                      10   AE   9F 000E0          PUSHAB   OUT_LENGTH                               ;
                      40   AE   9F 000E3          PUSHAB   LINE_DESC                                ;
             66            03   FB 000E6          CALLS    #3, STR$COPY_R                           ;
             79            50   E9 000E9          BLBC     BUILD_STATUS, 7$                         ;
                      38   AE   9F 000EC          PUSHAB   LINE_DESC                                ;
                           52   DD 000EF          PUSHL    R2                                       ;
             68            02   FB 000F1          CALLS    #2, PRINT                                ;
             79            50   E9 000F4          BLBC     PRINT_STATUS, 8$                         ;
       F8    AD 010E0037   8F   D0 000F7          MOVL     #17694775, CTL_STR_DSC                   ; 0569
       FC    AD     FE61   CF   9E 000FF          MOVAB    P.AAE, CTL_STR_DSC+4                     ;
                           20   DD 00105          PUSHL    #32                                      ;
                      F0   AD   9F 00107          PUSHAB   TEMP_STR_DSC                             ;
                      14   AE   9F 0010A          PUSHAB   OUT_LENGTH                               ;
                      F8   AD   9F 0010D          PUSHAB   CTL_STR_DSC                              ;
             65            04   FB 00110          CALLS    #4, SYS$FAO                              ;
             05            50   E8 00113          BLBS     FAO_STATUS, 3$                           ;
                           50   DD 00116          PUSHL    FAO_STATUS                               ;
             64            01   FB 00118          CALLS    #1, LIB$STOP                             ;
```

```
                        F4    AD  DD 0011B 3$:     PUSHL   TEMP_STR_DSC+4                  ;
                        10    AE  9F 0011E         PUSHAB  OUT_LENGTH                      ;
                        40    AE  9F 00121         PUSHAB  LINE_DESC                       ;
               66             03  FB 00124         CALLS   #3, STR$COPY_R                  ;
               79             50  E9 00127 4$:     BLBC    BUILD_STATUS, 10$               ;
                        38    AE  9F 0012A         PUSHAB  LINE_DESC                       ;
                              52  DD 0012D         PUSHL   R2                              ;
               68             02  FB 0012F         CALLS   #2, PRINT                       ;
               79             50  E9 00132 5$:     BLBC    PRINT_STATUS, 11$               ;
         F8    AD 010E0002    8F  D0 00135         MOVL    #17694722, CTL_STR_DSC          ; 0570
         FC    AD      FE5B   CF  9E 0013D         MOVAB   P.AAF, CTL_STR_DSC+4            ;
                              20  DD 00143         PUSHL   #32                             ;
                        F0    AD  9F 00145         PUSHAB  TEMP_STR_DSC                    ;
                        14    AE  9F 00148         PUSHAB  OUT_LENGTH                      ;
                        F8    AD  9F 0014B         PUSHAB  CTL_STR_DSC                     ;
               65             04  FB 0014E         CALLS   #4, SYS$FAO                     ;
               05             50  E8 00151         BLBS    FAO_STATUS, 6$                  ;
                              50  DD 00154         PUSHL   FAO_STATUS                      ;
               64             01  FB 00156         CALLS   #1, LIB$STOP                    ;
                        F4    AD  DD 00159 6$:     PUSHL   TEMP_STR_DSC+4                  ;
                        10    AE  9F 0015C         PUSHAB  OUT_LENGTH                      ;
                        40    AE  9F 0015F         PUSHAB  LINE_DESC                       ;
               66             03  FB 00162         CALLS   #3, STR$COPY_R                  ;
               79             50  E9 00165 7$:     BLBC    BUILD_STATUS, 13$               ;
                        38    AE  9F 00168         PUSHAB  LINE_DESC                       ;
                              52  DD 0016B         PUSHL   R2                              ;
               68             02  FB 0016D         CALLS   #2, PRINT                       ;
               79             50  E9 00170 8$:     BLBC    PRINT_STATUS, 14$               ;
         F8    AD 010E002C    8F  D0 00173         MOVL    #17694764, CTL_STR_DSC          ; 0571
         FC    AD      FE21   CF  9E 0017B         MOVAB   P.AAG, CTL_STR_DSC+4            ;
                              20  DD 00181         PUSHL   #32                             ;
                        F0    AD  9F 00183         PUSHAB  TEMP_STR_DSC                    ;
                        14    AE  9F 00186         PUSHAB  OUT_LENGTH                      ;
                        F8    AD  9F 00189         PUSHAB  CTL_STR_DSC                     ;
               65             04  FB 0018C         CALLS   #4, SYS$FAO                     ;
               05             50  E8 0018F         BLBS    FAO_STATUS, 9$                  ;
                              50  DD 00192         PUSHL   FAO_STATUS                      ;
               64             01  FB 00194         CALLS   #1, LIB$STOP                    ;
                        F4    AD  DD 00197 9$:     PUSHL   TEMP_STR_DSC+4                  ;
                        10    AE  9F 0019A         PUSHAB  OUT_LENGTH                      ;
                        40    AE  9F 0019D         PUSHAB  LINE_DESC                       ;
               66             03  FB 001A0         CALLS   #3, STR$COPY_R                  ;
               7B             50  E9 001A3 10$:    BLBC    BUILD_STATUS, 16$               ;
                        38    AE  9F 001A6         PUSHAB  LINE_DESC                       ;
                              52  DD 001A9         PUSHL   R2                              ;
               68             02  FB 001AB         CALLS   #2, PRINT                       ;
               7B             50  E9 001AE 11$:    BLBC    PRINT_STATUS, 17$               ;
         F8    AD 010E0002    8F  D0 001B1         MOVL    #17694722, CTL_STR_DSC          ; 0572
         FC    AD      FE0F   CF  9E 001B9         MOVAB   P.AAH, CTL_STR_DSC+4            ;
                              20  DD 001BF         PUSHL   #32                             ;
                        F0    AD  9F 001C1         PUSHAB  TEMP_STR_DSC                    ;
                        14    AE  9F 001C4         PUSHAB  OUT_LENGTH                      ;
                        F8    AD  9F 001C7         PUSHAB  CTL_STR_DSC                     ;
               65             04  FB 001CA         CALLS   #4, SYS$FAO                     ;
               05             50  E8 001CD         BLBS    FAO_STATUS, 12$                 ;
                              50  DD 001D0         PUSHL   FAO_STATUS                      ;
               64             01  FB 001D2         CALLS   #1, LIB$STOP                    ;
```

```
                          F4  AD  DD 001D5 12$:    PUSHL   TEMP_STR_DSC+4                              ;
                          10  AE  9F 001D8         PUSHAB  OUT_LENGTH                                  ;
                          40  AE  9F 001DB         PUSHAB  LINE_DESC                                   ;
                 66           03  FB 001DE         CALLS   #3, STR$COPY_R                              ;
                 48           50  E9 001E1 13$:    BLBC    BUILD_STATUS, 17$                           ;
                          38  AE  9F 001E4         PUSHAB  LINE_DESC                                   ;
                              52  DD 001E7         PUSHL   R2                                          ;
                 68           02  FB 001E9         CALLS   #2, PRINT                                   ;
                 3D           50  E9 001EC 14$:    BLBC    PRINT_STATUS, 17$                           ;
            F8   AD 010E0019  8F  D0 001EF         MOVL    #17694745, CTL_STR_DSC                      ; 0573
            FC   AD     FDD5  CF  9E 001F7         MOVAB   P.AAI, CTL_STR_DSC+4                        ;
                 7E       85  8F  9A 001FD         MOVZBL  #133, -(SP)                                 ;
                          F0  AD  9F 00201         PUSHAB  TEMP_STR_DSC                                ;
                          14  AE  9F 00204         PUSHAB  OUT_LENGTH                                  ;
                          F8  AD  9F 00207         PUSHAB  CTL_STR_DSC                                 ;
                 65           04  FB 0020A         CALLS   #4, SYS$FAO                                 ;
                 05           50  E8 0020D         BLBS    FAO_STATUS, 15$                             ;
                              50  DD 00210         PUSHL   FAO_STATUS                                  ;
                 64           01  FB 00212         CALLS   #1, LIB$STOP                                ;
                          F4  AD  DD 00215 15$:    PUSHL   TEMP_STR_DSC+4                              ;
                          10  AE  9F 00218         PUSHAB  OUT_LENGTH                                  ;
                          40  AE  9F 0021B         PUSHAB  LINE_DESC                                   ;
                 66           03  FB 0021E         CALLS   #3, STR$COPY_R                              ;
                 72           50  E9 00221 16$:    BLBC    BUILD_STATUS, 20$                           ;
                          38  AE  9F 00224         PUSHAB  LINE_DESC                                   ;
                              52  DD 00227         PUSHL   R2                                          ;
                 68           02  FB 00229         CALLS   #2, PRINT                                   ;
                 67           50  E9 0022C 17$:    BLBC    PRINT_STATUS, 20$                           ;
                          08  AE  D4 0022F         CLRL    CODE                                        ; 0578
                          20  AE  9F 00232 18$:    PUSHAB  TEXT_DESC                                   ; 0580
                          04  AE  9F 00235         PUSHAB  TEXT_LENGTH                                 ;
                          20  AE  9F 00238         PUSHAB  NAME_DESC                                   ;
                          10  AE  9F 0023B         PUSHAB  NAME_LENGTH                                 ;
                          20  AE  9F 0023E         PUSHAB  SEVERITY_DESC                               ;
                          1C  AE  9F 00241         PUSHAB  CODE                                        ;
      00000000G  00           06  FB 00244         CALLS   #6, EDT$MSGTXT                              ;
                 53       14  AE  D0 0024B         MOVL    SEVERITY_DESC+4, SEVERITY_ADDR              ; 0584
                 20           63  91 0024F         CMPB    (SEVERITY_ADDR), #32                        ; 0586
                              4E  13 00252         BEQL    21$                                         ;
            F8   AD 010E000D  8F  D0 00254         MOVL    #17694733, CTL_STR_DSC                      ; 0588
            FC   AD     FD8C  CF  9E 0025C         MOVAB   P.AAJ, CTL_STR_DSC+4                        ;
                          20  AE  9F 00262         PUSHAB  TEXT_DESC                                   ;
                          14  AE  9F 00265         PUSHAB  SEVERITY_DESC                               ;
                          20  AE  9F 00268         PUSHAB  NAME_DESC                                   ;
                          F0  AD  9F 0026B         PUSHAB  TEMP_STR_DSC                                ;
                          1C  AE  9F 0026E         PUSHAB  OUT_LENGTH                                  ;
                          F8  AD  9F 00271         PUSHAB  CTL_STR_DSC                                 ;
                 65           06  FB 00274         CALLS   #6, SYS$FAO                                 ;
                 05           50  E8 00277         BLBS    FAO_STATUS, 19$                             ;
                              50  DD 0027A         PUSHL   FAO_STATUS                                  ;
                 64           01  FB 0027C         CALLS   #1, LIB$STOP                                ;
                          F4  AD  DD 0027F 19$:    PUSHL   TEMP_STR_DSC+4                              ;
                          10  AE  9F 00282         PUSHAB  OUT_LENGTH                                  ;
                          40  AE  9F 00285         PUSHAB  LINE_DESC                                   ;
                 66           03  FB 00288         CALLS   #3, STR$COPY_R                              ;
                 4F           50  E9 0028B         BLBC    BUILD_STATUS, 23$                           ;
                          38  AE  9F 0028E         PUSHAB  LINE_DESC                                   ;
```

```
                                      52  DD 00291         PUSHL   R2                                         ;
                         68           02  FB 00293         CALLS   #2, PRINT                                  ;
                         44           50  E9 00296  20$:   BLBC    PRINT_STATUS, 23$                          ;
          90      08     AE 00000FFF  8F  F3 00299         AOBLEQ  #4095, CODE, 18$                           ; 0578
                  F8     AD 010E0004  8F  D0 002A2  21$:   MOVL    #17694724, CTL_STR_DSC                     ; 0594
                  FC     AD     FD4E  CF  9E 002AA         MOVAB   P.AAK, CTL_STR_DSC+4                       ;
                                      20  DD 002B0         PUSHL   #32                                        ;
                              F0      AD  9F 002B2         PUSHAB  TEMP_STR_DSC                               ;
                              14      AE  9F 002B5         PUSHAB  OUT_LENGTH                                 ;
                              F8      AD  9F 002B8         PUSHAB  CTL_STR_DSC                                ;
                         65           04  FB 002BB         CALLS   #4, SYS$FAO                                ;
                         05           50  E8 002BE         BLBS    FAO_STATUS, 22$                            ;
                                      50  DD 002C1         PUSHL   FAO_STATUS                                 ;
                         64           01  FB 002C3         CALLS   #1, LIB$STOP                               ;
                              F4      AD  DD 002C6  22$:   PUSHL   TEMP_STR_DSC+4                             ;
                              10      AE  9F 002C9         PUSHAB  OUT_LENGTH                                 ;
                              40      AE  9F 002CC         PUSHAB  LINE_DESC                                  ;
                         66           03  FB 002CF         CALLS   #3, STR$COPY_R                             ;
                         62           50  E9 002D2         BLBC    BUILD_STATUS, 30$                          ;
                              38      AE  9F 002D5         PUSHAB  LINE_DESC                                  ;
                                      52  DD 002D8         PUSHL   R2                                         ;
                         68           02  FB 002DA         CALLS   #2, PRINT                                  ;
                         57           50  E9 002DD  23$:   BLBC    PRINT_STATUS, 30$                          ;
                              38      AE  9F 002E0         PUSHAB  LINE_DESC                                  ; 0598
                         67           01  FB 002E3         CALLS   #1, STR$FREE1_DX                           ;
                         05           50  E8 002E6         BLBS    FREE_STATUS, 24$                           ;
                                      50  DD 002E9         PUSHL   FREE_STATUS                                ;
                         64           01  FB 002EB         CALLS   #1, LIB$STOP                               ;
                              30      AE  9F 002EE  24$:   PUSHAB  PRINTABLE_DESC                             ; 0599
                         67           01  FB 002F1         CALLS   #1, STR$FREE1_DX                           ;
                         05           50  E8 002F4         BLBS    FREE_STATUS, 25$                           ;
                                      50  DD 002F7         PUSHL   FREE_STATUS                                ;
                         64           01  FB 002F9         CALLS   #1, LIB$STOP                               ;
                              28      AE  9F 002FC  25$:   PUSHAB  HEX_DESC                                   ; 0600
                         67           01  FB 002FF         CALLS   #1, STR$FREE1_DX                           ;
                         05           50  E8 00302         BLE.    FREE_STATUS, 26$                           ;
                                      50  DD 00305         PUSHL   FREE_STATUS                                ;
                         64           01  FB 00307         CALLS   #1, LIB$STOP                               ;
                              20      AE  9F 0030A  26$:   PUSHAB  TEXT_DESC                                  ; 0601
                         67           01  FB 0030D         CALLS   #1, STR$FREE1_DX                           ;
                         05           50  E8 00310         BLBS    FREE_STATUS, 27$                           ;
                                      50  DD 00313         PUSHL   FREE_STATUS                                ;
                         64           01  FB 00315         CALLS   #1, LIB$STOP                               ;
                              18      AE  9F 00318  27$:   PUSHAB  NAME_DESC                                  ; 0602
                         67           01  FB 0031B         CALLS   #1, STR$FREE1_DX                           ;
                         05           50  E8 0031E         BLBS    FREE_STATUS, 28$                           ;
                                      50  DD 00321         PUSHL   FREE_STATUS                                ;
                         64           01  FB 00323         CALLS   #1, LIB$STOP                               ;
                              10      AE  9F 00326  28$:   PUSHAB  SEVERITY_DESC                              ; 0603
                         67           01  FB 00329         CALLS   #1, STR$FREE1_DX                           ;
                         05           50  E8 0032C         BLBS    FREE_STATUS, 29$                           ;
                                      50  DD 0032F         PUSHL   FREE_STATUS                                ;
                         64           01  FB 00331         CALLS   #1, LIB$STOP                               ;
                         50           01  D0 00334  29$:   MOVL    #1, R0                                     ; 0604
                                          04 00337  30$:   RET                                                ; 0605

; Routine Size:  824 bytes,    Routine Base:  _EDT$CODE + 0294
```

```
1 %SBTTL 'PRINT - print a text line on a file'
1 ROUTINE PRINT (                                       ! Print a text line on a file
1       RAB_ADDR,                                       ! RAB onto which to print the line
1       TEXT_LINE                                       ! Descriptor of the line to print
1       ) =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine interfaces to RMS to print a line of text.
1 !
1 ! CALLING SEQUENCE:
1 !
1 !       ret_status.wlc.v = PRINT (RAB_ADDR.mz.r, TEXT_LINE.rt.dx)
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       RAB_ADDR                Pointer to the RAB onto which to print the line of text
1 !       TEXT_LINE               Descriptor for the line of text to be printed.
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! COMPLETION STATUS:
1 !
1 !       SS$_NORMAL      Normal successful completion
1 !       All RMS errors are returned to the caller, so that they can be
1 !       signalled with the file name.
1 !
1 ! SIDE EFFECTS:
1 !
1 !       Does a $PUT to the RAB.
1 !
1 !--
1
2     BEGIN
2
2     MAP
2       RAB_ADDR : REF $RAB_DECL,
2       TEXT_LINE : REF BLOCK [8, BYTE];
2
2     LOCAL
2       PUT_STATUS;
2
2 !+
2 ! Fill in the RAB fields.
2 !-
2     RAB_ADDR [RAB$W_RSZ] = .TEXT_LINE [DSC$W_LENGTH];
2     RAB_ADDR [RAB$L_RBF] = .TEXT_LINE [DSC$A_POINTER];
2 !+
2 ! Now do the $PUT
2 !-
```

```
;  535          0663  2       PUT_STATUS = $PUT (RAB = .RAB_ADDR);
;  536          0664  2
;  537          0665  2       IF ( NOT .PUT_STATUS) THEN RETURN (.PUT_STATUS);
;  538          0666  2
;  539          0667  2       RETURN (SS$_NORMAL);
;  540          0668  1       END;                                      ! End of routine PRINT


                                                               .EXTRN  SYS$PUT

                                             0000 00000 PRINT:  .WORD   Save nothing                          ; 0607
                          51       04   AC   D0 00002           MOVL    RAB_ADDR, R1                          ; 0658
                          50       08   AC   D0 00006           MOVL    TEXT_LINE, R0                         ;
                     22   A1            60   B0 0000A           MOVW    (R0), 34(R1)                          ;
                     28   A1       04   A0   D0 0000E           MOVL    4(R0), 40(R1)                         ; 0659
                                        51   DD 00013           PUSHL   R1                                    ; 0663
              00000000G   00            01   FB 00015           CALLS   #1, SYS$PUT                           ;
                          03            50   E9 0001C           BLBC    PUT_STATUS, 1$                        ; 0665
                          50            01   D0 0001F           MOVL    #1, R0                                ; 0667
                                             04 00022 1$:       RET                                           ; 0668

; Routine Size:  35 bytes,    Routine Base:  _EDT$CODE + 05CC
```

```
1 %SBTTL 'HEX_TEXT - Return a binary string in hexadecimal'
1 ROUTINE HEX_TEXT (                                          ! Return a binary string in hex
1     OUTPUT_DESC,                                            ! Descriptor to receive the hex
1     INPUT_LEN,                                              ! Number of input bytes
1     INPUT_ADDR                                              ! Address of start of input
1     ) =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine converts an arbitrary string of bytes into hex, so it
1 !       can be printed.  Early bytes are put to the right of later bytes.
1 !
1 ! CALLING SEQUENCE:
1 !
1 !       status.wlc.v = HEX_TEXT (OUTPUT_desc.wt.dx, INPUT_LEN.rl.v, INPUT_ADDR,ra.v)
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       output_desc      Where the result text is stored.
1 !       input_len        Number of bytes of input
1 !       input_addr       Address of first input byte
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! COMPLETION STATUS:
1 !
1 !       SS$_NORMAL       Normal successful completion
1 !       Any errors from STR$CONCAT
1 !       Any errors from STR$COPY_DX
1 !
1 ! SIDE EFFECTS:
1 !
1 !       Calls STR$CONCAT and STR$COPY_DX, thus manipulating string storage.
1 !
1 !--
1
2     BEGIN
2
2     MAP
2         INPUT_ADDR : REF VECTOR [, BYTE],
2         OUTPUT_DESC : REF BLOCK [8, BYTE];
2
2     LOCAL
2         INTER_DESC : BLOCK [8, BYTE],
2         DIGIT_DESC : BLOCK [8, BYTE],
2         DIGIT,
2         STATUS;
2
2     INIT_DESCRIPTOR (INTER_DESC);
2     DIGIT_DESC [DSC$W_LENGTH] = 1;
```

```
;  599          0726  2         DIGIT_DESC [DSC$B_DTYPE] = DSC$K_DTYPE_T;
;  600          0727  2         DIGIT_DESC [DSC$B_CLASS] = DSC$K_CLASS_S;
;  601          0728  2         DIGIT_DESC [DSC$A_POINTER] = DIGIT;
;  602          0729  2
;  603          0730  2         INCR CHAR_NO FROM 1 TO .INPUT_LEN DO
;  604          0731  3             BEGIN
;  605          0732  3
;  606          0733  3             LOCAL
;  607          0734  3                 CHAR;
;  608          0735  3
;  609          0736  3             CHAR = .INPUT_ADDR [.CHAR_NO - 1];
;  610          0737  3             DIGIT = (.CHAR AND 15) + '0';
;  611          0738  3
;  612          0739  3             IF (.DIGIT GTR '9') THEN DIGIT = .DIGIT - 10 - '0' + 'A';
;  613          0740  3
;  614          0741  3             STATUS = STR$CONCAT (INTER_DESC, DIGIT_DESC, INTER_DESC);
;  615          0742  3
;  616          0743  3             IF ( NOT .STATUS) THEN RETURN (.STATUS);
;  617          0744  3
;  618          0745  3             DIGIT = (.CHAR^-4) + '0';
;  619          0746  3
;  620          0747  3             IF (.DIGIT GTR '9') THEN DIGIT = .DIGIT - 10 - '0' + 'A';
;  621          0748  3
;  622          0749  3             STATUS = STR$CONCAT (INTER_DESC, DIGIT_DESC, INTER_DESC);
;  623          0750  3
;  624          0751  3             IF ( NOT .STATUS) THEN RETURN (.STATUS);
;  625          0752  3
;  626          0753  2             END;
;  627          0754  2
;  628          0755  2         STATUS = STR$COPY_DX (.OUTPUT_DESC, INTER_DESC);
;  629          0756  2         DISCARD_DESCRIPTOR (INTER_DESC);
;  630          0757  2         RETURN (.STATUS);
;  631          0758  1         END;                                        ! End of routine HEX_TEXT



                                                  003C 00000 HEX_TEXT:
                                                                    .WORD   Save R2,R3,R4,R5                           ; 0670
                             55 00000000G   00   9E 00002           MOVAB   STR$CONCAT, R5                             ;
                             5E             14   C2 00009           SUBL2   #20, SP                                    ;
                        0C   AE 020E0000    8F   D0 0000C           MOVL    #34471936, INTER_DESC                      ; 0724
                                       10   AE   D4 00014           CLRL    INTER_DESC+4                               ;
                        04   AE 010E0001    8F   D0 00017           MOVL    #17694721, DIGIT_DESC                      ; 0725
                        08   AE             6E   9E 0001F           MOVAB   DIGIT, DIGIT_DESC+4                        ; 0728
                                            53   D4 00023           CLRL    CHAR_NO                                    ; 0736
                                            4E   11 00025           BRB     4$                                         ;
                 50          53        0C   AC   C1 00027 1$:       ADDL3   INPUT_ADDR, CHAR_NO, R0                    ;
                             52        FF   A0   9A 0002C           MOVZBL  -1(R0), CHAR                               ;
       6E        52          04             00   EF 00030           EXTZV   #0, #4, CHAR, DIGIT                        ; 0737
                             6E             30   C0 00035           ADDL2   #48, DIGIT                                 ;
                             39             6E   D1 00038           CMPL    DIGIT, #57                                 ; 0739
                                            03   15 0003B           BLEQ    2$                                         ;
                             6E             07   C0 0003D           ADDL2   #7, DIGIT                                  ;
                                       0C   AE   9F 00040 2$:       PUSHAB  INTER_DESC                                 ; 0741
                                       08   AE   9F 00043           PUSHAB  DIGIT_DESC                                 ;
```

```
                               14    AE   9F 00046         PUSHAB   INTER_DESC                           ;
                      65             03   FB 00049         CALLS    #3, STR$CONCAT                       ;
                      54             50   D0 0004C         MOVL     R0, STATUS                           ;
                      4E             54   E9 0004F         BLBC     STATUS, 5$                           ; 0743
          52          52       FC    8F   78 00052         ASHL     #-4, CHAR, R2                        ; 0745
                      6E       30    A2   9E 00057         MOVAB    48(R2), DIGIT                        ;
                      39             6E   D1 0005B         CMPL     DIGIT, #57                           ; 0747
                                     03   15 0005E         BLEQ     3$                                   ;
                      6E             07   C0 00060         ADDL2    #7, DIGIT                            ;
                               0C    AE   9F 00063 3$:     PUSHAB   INTER_DESC                           ; 0749
                               08    AE   9F 00066         PUSHAB   DIGIT_DESC                           ;
                               14    AE   9F 00069         PUSHAB   INTER_DESC                           ;
                      65             03   FB 0006C         CALLS    #3, STR$CONCAT                       ;
                      54             50   D0 0006F         MOVL     R0, STATUS                           ;
                      2B             54   E9 00072         BLBC     STATUS, 5$                           ; 0751
          AD          53       08    AC   F3 00075 4$:     AOBLEQ   INPUT_LEN, CHAR_NO, 1$               ; 0730
                               0C    AE   9F 0007A         PUSHAB   INTER_DESC                           ; 0755
                               04    AC   DD 0007D         PUSHL    OUTPUT_DESC                          ;
             00000000G  00           02   FB 00080         CALLS    #2, STR$COPY_DX                      ;
                      54             50   D0 00087         MOVL     R0, STATUS                           ;
                               0C    AE   9F 0008A         PUSHAB   INTER_DESC                           ; 0756
             00000000G  00           01   FB 0008D         CALLS    #1, STR$FREE1_DX                     ;
                      09             50   E8 00094         BLBS     FREE_STATUS, 5$                      ;
                                     50   DD 00097         PUSHL    FREE_STATUS                          ;
             00000000G  00           01   FB 00099         CALLS    #1, LIB$STOP                         ;
                      50             54   D0 000A0 5$:     MOVL     STATUS, R0                           ; 0757
                                          04 000A3         RET                                           ; 0758

; Routine Size:  164 bytes,    Routine Base:  _EDT$CODE + 05EF
```

```
1 %SBTTL 'PRINTABLE_TEXT - Return a binary string in ASCII, printable'
1 ROUTINE PRINTABLE_TEXT (                              ! Return a binary string in printable ASCII
1     OUTPUT_DESC,                                      ! Descriptor to receive the text
1     INPUT_LEN,                                        ! Number of input bytes
1     INPUT_ADDR                                        ! Address of start of input
1     ) =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine converts an arbitrary string of bytes into ASCII, representing
1 !       unprintable characters in hexadecimal so the result can be printed.
1 !
1 ! CALLING SEQUENCE:
1 !
1 !       status.wlc.v = PRINTABLE_TEXT (OUTPUT_desc.wt.dx, INPUT_LEN.rl.v, INPUT_ADDR,ra.v)
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       OUTPUT_DESC     Where the result text is stored.
1 !       INPUT_LEN       Number of bytes of input
1 !       INPUT_ADDR      Address of first input byte
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! COMPLETION STATUS:
1 !
1 !       SS$_NORMAL      Normal successful completion
1 !       Any errors from STR$CONCAT
1 !       Any errors from STR$COPY_DX
1 !
1 ! SIDE EFFECTS:
1 !
1 !       Calls STR$CONCAT and STR$COPY_DX, thus manipulating string storage.
1 !
1 !--
1
2     BEGIN
2
2     MAP
2         INPUT_ADDR : REF VECTOR [, BYTE],
2         OUTPUT_DESC : REF BLOCK [8, BYTE];
2
2     LOCAL
2         INTER_DESC : BLOCK [8, BYTE],
2         CHAR_DESC : BLOCK [8, BYTE],
2         CHAR_REP : VECTOR [4, BYTE],
2         STATUS;
2
2     INIT_DESCRIPTOR (INTER_DESC);
2     CHAR_DESC [DSC$W_LENGTH] = 1;
```

```
2          CHAR_DESC [DSC$B_DTYPE] = DSC$K_DTYPE_T;
2          CHAR_DESC [DSC$B_CLASS] = DSC$K_CLASS_S;
2          CHAR_DESC [DSC$A_POINTER] = CHAR_REP [0];
2
2          INCR CHAR_NO FROM 1 TO .INPUT_LEN DO
3              BEGIN
3
3              LOCAL
3                  CHAR;
3
3              CHAR = .INPUT_ADDR [.CHAR_NO - 1];
3
4              IF ((.CHAR GEQ %X'20') AND                    !
4                  (.CHAR LSS %X'7F') AND                    !
4                  (.CHAR NEQ '<') AND                       !
4                  (.CHAR NEQ '"'))
3              THEN
4                  BEGIN
4  !+
4  ! Show character as itself.
4  !-
4                  CHAR_REP [0] = .CHAR;
4                  CHAR_DESC [DSC$W_LENGTH] = 1;
4                  END
3              ELSE
4                  BEGIN
4  !+
4  ! The character is not printable.  Represent it by <>.  To avoid
4  ! ambiguity, "<" and '"' are also represented this way.  Control characters
4  ! SOH through SUB are represented by <^letter>; others characters by <hex>.
4  !-
4                  CHAR_REP [0] = '<';
4
5                  IF (((.CHAR + %X'40') GEQ 'A') AND ((.CHAR + %X'40') LEQ 'Z'))
4                  THEN
5                      BEGIN
5                      CHAR_REP [1] = '^';
5                      CHAR_REP [2] = .CHAR + %X'40';
5                      END
4                  ELSE
5                      BEGIN
5
5                      LOCAL
5                          DIGIT;
5
5                      DIGIT = (.CHAR^-4) + '0';
5
5                      IF (.DIGIT GTR '9') THEN DIGIT = .DIGIT - 10 - '0' + 'A';
5
5                      CHAR_REP [1] = .DIGIT;
5                      DIGIT = (.CHAR AND 15) + '0';
5
5                      IF (.DIGIT GTR '9') THEN DIGIT = .DIGIT - 10 - '0' + 'A';
5
5                      CHAR_REP [2] = .DIGIT;
4                      END;
4
```

```
                                                                               M  4
EDT$WRIEDTMSG     EDT$WRIEDTMSG - write VMSMSG.MSG               16-Sep-1984 02:18:31    VAX-11 Bliss-32 V4.0-742              Page  26
V04-000           PRINTABLE_TEXT - Return a binary string in ASCI 14-Sep-1984 12:25:55    [EDT.SRC]WRIEDTMSG.B32;1                   (8)

:  747            0873  4                   CHAR_REP [3] = '>';
:  748            0874  4                   CHAR_DESC [DSC$W_LENGTH] = 4;
:  749            0875  3                   END;
:  750            0876  3
:  751            0877  3               STATUS = STR$CONCAT (INTER_DESC, INTER_DESC, CHAR_DESC);
:  752            0878  3
:  753            0879  3               IF ( NOT .STATUS) THEN RETURN (.STATUS);
:  754            0880  3
:  755            0881  2               END;
:  756            0882  2
:  757            0883  2           STATUS = STR$COPY_DX (.OUTPUT_DESC, INTER_DESC);
:  758            0884  2           DISCARD_DESCRIPTOR (INTER_DESC);
:  759            0885  2           RETURN (.STATUS);
:  760            0886  1           END;                                        ! End of routine PRINTABLE_TEXT



                                                    000C 00000 PRINTABLE_TEXT:
                                                                      .WORD     Save R2,R3                                 ; 0760
                        5E              14  C2 00002                  SUBL2     #20, SP                                    ;
                0C      AE 020E0000     8F  D0 00005                  MOVL      #34471936, INTER_DESC                      ; 0814
                                 10     AE  D4 0000D                  CLRL      INTER_DESC+4                               ;
                04      AE 010E0001     8F  D0 00010                  MOVL      #17694721, CHAR_DESC                       ; 0815
                08      AE              6E  9E 00018                  MOVAB     CHAR_REP, CHAR_DESC+4                      ; 0818
                                        52  D4 0001C                  CLRL      CHAR_NO                                    ; 0826
                                      0090  31 0001E                  BRW       7$                                         ;
          50            52          0C  AC  C1 00021 1$:              ADDL3     INPUT_ADDR, CHAR_NO, R0                    ;
                        51          FF  A0  9A 00026                  MOVZBL    -1(R0), CHAR                               ;
                        20              51  D1 0002A                  CMPL      CHAR, #32                                  ; 0828
                                        1C  19 0002D                  BLSS      2$                                         ;
            0000007F    8F              51  D1 0002F                  CMPL      CHAR, #127                                 ; 0829
                                        13  18 00036                  BGEQ      2$                                         ;
                        3C              51  D1 00038                  CMPL      CHAR, #60                                  ; 0830
                                        0E  13 0003B                  BEQL      2$                                         ;
                        22              51  D1 0003D                  CMPL      CHAR, #34                                  ; 0831
                                        09  13 00040                  BEQL      2$                                         ;
                        6E              51  90 00042                  MOVB      CHAR, CHAR_REP                             ; 0837
                04      AE              01  B0 00045                  MOVW      #1, CHAR_DESC                              ; 0838
                                        50  11 00049                  BRB       6$                                         ; 0828
                        6E              3C  90 0004B 2$:              MOVB      #60, CHAR_REP                              ; 0847
                        50          40  A1  9E 0004E                  MOVAB     64(R1), R0                                 ; 0849
            00000041    8F              50  D1 00052                  CMPL      R0, #65                                    ;
                                        10  19 00059                  BLSS      3$                                         ;
            0000005A    8F              50  D1 0005B                  CMPL      R0, #90                                    ;
                                        07  14 00062                  BGTR      3$                                         ;
                01      AE          5E  8F  90 00064                  MOVB      #94, CHAR_REP+1                            ; 0852
                                        24  11 00069                  BRB       5$                                         ; 0853
          50            51          FC  8F  78 0006B 3$:              ASHL      #-4, CHAR, R0                              ; 0861
                        50              30  C0 00070                  ADDL2     #48, DIGIT                                 ;
                        39              50  D1 00073                  CMPL      DIGIT, #57                                 ; 0863
                                        03  15 00076                  BLEQ      4$                                         ;
                        50              07  C0 00078                  ADDL2     #7, DIGIT                                  ;
                01      AE              50  90 0007B 4$:              MOVB      DIGIT, CHAR_REP+1                          ; 0865
  50            51      04              00  EF 0007F                  EXTZV     #0, #4, CHAR, DIGIT                        ; 0866
                        50              30  C0 00084                  ADDL2     #48, DIGIT                                 ;
```

```
_$2

Syml
---
CTL'
CTL'
EXE'
EXE'
EXE'
EXE'
EXE'
EXE'
FP$I
FP$I
FP$I
FP$I
FP$
FP_
MMG'
MMG'
PR$
SYS'
VAX'
VAX'
```

```
                                   39            50  D1 00087          CMPL     DIGIT, #57                                  ; 0868
                                                 03  15 0008A          BLEQ     5$                                          ;
                                   50            07  C0 0008C          ADDL2    #7, DIGIT                                   ;
                              02   AE            50  90 0008F 5$:      MOVB     DIGIT, CHAR_REP+2                           ; 0870
                              03   AE            3E  90 00093          MOVB     #62, CHAR_REP+3                             ; 0873
                              04   AE            04  B0 00097          MOVW     #4, CHAR_DESC                               ; 0874
                                          04     AE  9F 0009B 6$:      PUSHAB   CHAR_DESC                                   ; 0877
                                          10     AE  9F 0009E          PUSHAB   INTER_DESC                                  ;
                                          14     AE  9F 000A1          PUSHAB   INTER_DESC                                  ;
                        00000000G  00            03  FB 000A4          CALLS    #3, STR$CONCAT                              ;
                                   53            50  D0 000AB          MOVL     R0, STATUS                                  ;
                                   2D            53  E9 000AE          BLBC     STATUS, 8$                                  ; 0879
          FF69          52         01     08     AC  F1 000B1 7$:      ACBL     INPUT_LEN, #1, CHAR_NO, 1$                  ; 0820
                                          0C     AE  9F 000B8          PUSHAB   INTER_DESC                                  ; 0883
                                          04     AC  DD 000BB          PUSHL    OUTPUT_DESC                                 ;
                        00000000G  00            02  FB 000BE          CALLS    #2, STR$COPY_DX                             ;
                                   53            50  D0 000C5          MOVL     R0, STATUS                                  ;
                                          0C     AE  9F 000C8          PUSHAB   INTER_DESC                                  ; 0884
                        00000000G  00            01  FB 000CB          CALLS    #1, STR$FREE1_DX                            ;
                                   09            50  E8 000D2          BLBS     FREE_STATUS, 8$                             ;
                                                 50  DD 000D5          PUSHL    FREE_STATUS                                 ;
                        00000000G  00            01  FB 000D7          CALLS    #1, LIB$STOP                                ;
                                   50            53  D0 000DE 8$:      MOVL     STATUS, R0                                  ; 0885
                                                     04 000E1          RET                                                  ; 0886

; Routine Size:  226 bytes,    Routine Base:  _EDT$CODE + 0693


;  761          0887  1 !<BLF/PAGE>
```

```
_$2
Valu
----
000
000
000
000
000
000
000
000
4851
7FFE
7FFE
800
800
800
800
800
800
8001
8001
```

```
;  763         0888  1 END                                              ! End of module EDT$WRIEDTMSG
;  764         0889  1
;  765         0890  0 ELUDOM



                                                                    .EXTRN  LIB$STOP

;                                        PSECT SUMMARY
;
;       Name                          Bytes                        Attributes
;
;  _EDT$CODE                              1909  NOVEC,NOWRT,  RD ,  EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(2)



;                                Library Statistics
;
;                                                  -------- Symbols --------      Pages        Processing
;       File                                       Total    Loaded   Percent      Mapped       Time
;
;  _$255$DUA28:[SYSLIB]STARLET.L32;1                9776      106         1         581         00:02.4




;                                        COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACEBACK/LIS=LIS$:WRIEDTMSG/OBJ=OBJ$:WRIEDTMSG MSRC$:WRIEDTMSG.B32/UPDATE=(ENH$:W
;       RIEDTMSG)

; Size:          1601 code + 308 data bytes
; Run Time:         01:14.7
; Elapsed Time:     02:10.7
; Lines/CPU Min:      714
; Lexemes/CPU-Min: 11820
; Memory Used:  334 pages
; Compilation Complete
```

```
_$2

Vir
Sta
Ima
Ima
Ima
Num
Num
Num
Num
Num
Num
Ima
Map
Est

Per
---

Tot

Usi

Tot

Num

O l

A t

LIN
OAD
```

0142 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

XLATE
LIS

VAXEMUL
MAP

WIOHNDLR
LIS

WORKIO
LIS

BOOSTRING
LIS

EMULAT

FPEMUL
MAP

BOOEMULAT
LIS

WORDWRAP
LIS

VAXMACROS
MAR

WRIEDTMSG
LIS

FPEMULATE
LIS

VAXREGDEF
SDL