H 5

```
IIIIIIIII     MMM         MMM     GGGGGGGGGGGG  DDDDDDDDDDDD     MMM         MMM  PPPPPPPPPPPP
IIIIIIIII     MMM         MMM     GGGGGGGGGGGG  DDDDDDDDDDDD     MMM         MMM  PPPPPPPPPPPP
IIIIIIIII     MMM         MMM     GGGGGGGGGGGG  DDDDDDDDDDDD     MMM         MMM  PPPPPPPPPPPP
   III        MMMMMM   MMMMMM  GGG              DDD         DDD  MMMMMM   MMMMMM  PPP         PPP
   III        MMMMMM   MMMMMM  GGG              DDD         DDD  MMMMMM   MMMMMM  PPP         PPP
   III        MMMMMM   MMMMMM  GGG              DDD         DDD  MMMMMM   MMMMMM  PPP         PPP
   III        MMM   MMM   MMM  GGG              DDD         DDD  MMM   MMM   MMM  PPP         PPP
   III        MMM   MMM   MMM  GGG              DDD         DDD  MMM   MMM   MMM  PPP         PPP
   III        MMM   MMM   MMM  GGG              DDD         DDD  MMM   MMM   MMM  PPP         PPP
   III        MMM         MMM  GGG              DDD         DDD  MMM         MMM  PPPPPPPPPPPP
   III        MMM         MMM  GGG              DDD         DDD  MMM         MMM  PPPPPPPPPPPP
   III        MMM         MMM  GGG              DDD         DDD  MMM         MMM  PPPPPPPPPPPP
   III        MMM         MMM  GGG   GGGGGGGGG  DDD         DDD  MMM         MMM  PPP
   III        MMM         MMM  GGG   GGGGGGGGG  DDD         DDD  MMM         MMM  PPP
   III        MMM         MMM  GGG   GGGGGGGGG  DDD         DDD  MMM         MMM  PPP
   III        MMM         MMM  GGG         GGG  DDD         DDD  MMM         MMM  PPP
   III        MMM         MMM  GGG         GGG  DDD         DDD  MMM         MMM  PPP
   III        MMM         MMM  GGG         GGG  DDD         DDD  MMM         MMM  PPP
IIIIIIIII     MMM         MMM     GGGGGGGGG     DDDDDDDDDDDD     MMM         MMM  PPP
IIIIIIIII     MMM         MMM     GGGGGGGGG     DDDDDDDDDDDD     MMM         MMM  PPP
IIIIIIIII     MMM         MMM     GGGGGGGGG     DDDDDDDDDDDD     MMM         MMM  PPP
```

```
**FILE**ID**ANALIMDMP                                   G  7




  AAAAAA    NN      NN      AAAAAA    LL            IIIIII    MM      MM  DDDDDDDD    MM      MM  PPPPPPPP
  AAAAAA    NN      NN      AAAAAA    LL            IIIIII    MM      MM  DDDDDDDD    MM      MM  PPPPPPPP
AA      AA  NN      NN  AA      AA  LL              II      MMMM  MMMM  DD      DD  MMMM  MMMM  PP      PP
AA      AA  NN      NN  AA      AA  LL              II      MMMM  MMMM  DD      DD  MMMM  MMMM  PP      PP
AA      AA  NNNN    NN  AA      AA  LL              II      MM  MM  MM  DD      DD  MM  MM  MM  PP      PP
AA      AA  NNNN    NN  AA      AA  LL              II      MM  MM  MM  DD      DD  MM  MM  MM  PP      PP
AA      AA  NN  NN  NN  AA      AA  LL              II      MM      MM  DD      DD  MM      MM  PPPPPPPP
AA      AA  NN  NN  NN  AA      AA  LL              II      MM      MM  DD      DD  MM      MM  PPPPPPPP
AAAAAAAAAA  NN    NNNN  AAAAAAAAAA  LL              II      MM      MM  DD      DD  MM      MM  PP
AAAAAAAAAA  NN    NNNN  AAAAAAAAAA  LL              II      MM      MM  DD      DD  MM      MM  PP
AA      AA  NN      NN  AA      AA  LL              II      MM      MM  DD      DD  MM      MM  PP               ....
AA      AA  NN      NN  AA      AA  LL              II      MM      MM  DD      DD  MM      MM  PP               ....
AA      AA  NN      NN  AA      AA  LLLLLLLLLL    IIIIII    MM      MM  DDDDDDDD    MM      MM  PP               ....
AA      AA  NN      NN  AA      AA  LLLLLLLLLL    IIIIII    MM      MM  DDDDDDDD    MM      MM  PP               ....

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
```

```
0000            .TITLE  ANALIMDMP
0000            .IDENT  /V04-000/
0000       ;
0000       ;****************************************************************************
0000       ;*                                                                          *
0000       ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
0000       ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
0000       ;*  ALL RIGHTS RESERVED.                                                    *
0000       ;*                                                                          *
0000       ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
0000       ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0000       ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
0000       ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
0000       ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
0000       ;*  TRANSFERRED.                                                            *
0000       ;*                                                                          *
0000       ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
0000       ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0000       ;*  CORPORATION.                                                            *
0000       ;*                                                                          *
0000       ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
0000       ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
0000       ;*                                                                          *
0000       ;*                                                                          *
0000       ;****************************************************************************
0000       ;
0000
0000       ;++
0000       ; FACILITY:     IMAGE DUMP
0000       ;
0000       ; ABSTRACT:     Analyze an image dump and transfer control to debugger.
0000       ;
0000       ;
0000       ; ENVIRONMENT:  User mode
0000       ;
0000       ; AUTHOR:       Wayne Cardoza
0000       ;
0000       ; CREATION DATE: 14-Feb-1983
0000       ;
0000       ; MODIFIED BY:
0000       ;
0000       ;       V03-017 WMC0016         Wayne Cardoza   06-Aug-1984
0000       ;               Fix an improperly restored register.
0000       ;
0000       ;       V03-016 WMC0015         Wayne Cardoza   09-Jul-1984
0000       ;               Save and restore CTL$GL_IMGHDRBF.
0000       ;
0000       ;       V03-015 WMC0014         Wayne Cardoza   27-Jun-1984
0000       ;               Add control-Y handler to kill subprocess.
0000       ;
0000       ;       V03-014 WMC0014         Wayne Cardoza   23-May-1984
0000       ;               Fix several minor bugs relating to error checks and reporting.
0000       ;
0000       ;       V03-013 WMC0013         Wayne Cardoza   08-May-1984
0000       ;               Don't open dump file for write.
0000       ;
0000       ;       V03-012 WMC0012         Wayne Cardoza   22-Mar-1984
```

```
0000    ;               Don't let privilege be removed by image activation.
0000    ;
0000    ;       V03-011 WMC0011         Wayne Cardoza   29-Jan-1984
0000    ;               Fix defaults for /IMAGE.
0000    ;
0000    ;       V03-010 WMC0010         Wayne Cardoza   27-Dec-1983
0000    ;               Display the condition from the stack.
0000    ;
0000    ;       V03-009 WMC0009         Wayne Cardoza   13-Nov-1983
0000    ;               Don't let image activator remove privileges.
0000    ;
0000    ;       V03-008 WMC0008         Wayne Cardoza   26-Sep-1983
0000    ;               Vectors must also be reset after IMGACT.
0000    ;
0000    ;       V03-007 WMC0007         Wayne Cardoza   15-Sep-1983
0000    ;               Reset privileged library vectors when deleting P0.
0000    ;
0000    ;       V03-006 WMC0006         Wayne Cardoza   26-Aug-1983
0000    ;               Phony DEBUG frame had bad PUSH.
0000    ;
0000    ;       V03-005 WMC0005         Wayne Cardoza   14-Aug-1983
0000    ;               Fix priority of created subprocess.
0000    ;
0000    ;       V03-004 WMC0004         Wayne Cardoza   01-Jul-1983
0000    ;               SYS$IMGACT has been redesigned.
0000    ;
0000    ;       V03-003 WMC0003         Wayne Cardoza   25-May-1983
0000    ;               Fix a privilege problem.
0000    ;
0000    ;       V03-002 WMC0002         Wayne Cardoza   20-Apr-1983
0000    ;               Reset privileges before calling DEBUG.
0000    ;
0000    ;       V03-001 WMC0001         Wayne Cardoza   20-Apr-1983
0000    ;               Check dump version number for consistency.
0000    ;
```

```
00000000     94          .PSECT  ANALIMDMP,LONG
    0000     95          .DEFAULT DISPLACEMENT,WORD
    0000     96  ;
    0000     97  ; Data Structure Definitions
    0000     98  ;
    0000     99  ;
    0000    100          $IMGDMPDEF
    0000    101          $IMGMOVDEF
    0000    102          $IODEF
    0000    103          $DIBDEF
    0000    104          $IACDEF
    0000    105          $IHIDEF
    0000    106          $IHDDEF
    0000    107          $NAMDEF
    0000    108          $PHDDEF
    0000    109          $PRVDEF
```

```
                           0000  ;++
                           0000  ;
                           0000  ; Functional Description:
                           0000  ;       This is the main routine for analyzing an image dump.  It will display
                           0000  ;       some useful data and then transfer control to the debugger.
                           0000  ;
                           0000  ; Calling Sequence:
                           0000  ;       standard
                           0000  ;
                           0000  ; Input Parameters:
                           0000  ;       standard image argument list
                           0000  ;
                           0000  ; Implicit Inputs:
                           0000  ;       the command line
                           0000  ;
                           0000  ; Output Parameters:
                           0000  ;       none
                           0000  ;
                           0000  ; Implicit Outputs:
                           0000  ;       none
                           0000  ;
                           0000  ; Routine Value:
                           0000  ;       none
                           0000  ;
                           0000  ; Signals:
                           0000  ;       none
                           0000  ;
                           0000  ; Side Effects:
                           0000  ;       many
                           0000  ;
                           0000  ;--
                           0000  ;
                           0000  ;
                           0000  ANALIMDMP:
                      0000 0000          .WORD 0
                           0002
     17CB'CF    0C AC   D0 0002          MOVL    12(AP),THIS_HDR         ; Save this images pointers
     17CF'CF    10 AC   D0 0008          MOVL    16(AP),THIS_HDR+4       ; in case we don't load an image
                           000E  ;
              1908'CF   9F 000E          PUSHAB  SYS$INPUT               ; Translate sys$input
              193C'CF   9F 0012          PUSHAB  INPUT_TRN
                 04E2   30 0016          BSBW    GET_TRAN
     1B1B 8F  1944'CF   B1 0019          CMPW    SYS$INPUT_TRN,#<27+<27@8>> ; ESC-ESC means subprocess
                   08   12 0020          BNEQ    5$
                 03FA   30 0022          BSBW    INIT_SUBP               ; Do subprocess initialization
                45 50   E9 0025          BLBC    R0,10$
                   12   11 0028          BRB     7$
                           002A
              13BC'CF   9F 002A  5$:     PUSHAB  DUMP_NAME               ; Get the dump file name
              18E2'CF   9F 002E          PUSHAB  CLI_PARAMETER
 00000000'GF       02   FB 0032          CALLS   #2,G^CLI$GET_VALUE
                31 50   E9 0039          BLBC    R0,10$
                           003C  ;
                           003C  7$:     $CREATE FAB = OUTFAB            ; We will eventually need this
                23 50   E9 0047          BLBC    R0,10$
                           004A          $CONNECT RAB = OUTRAB
                15 50   E9 0055          BLBC    R0,10$
```

```
                                0058       ;
   1358'CF    13BC'CF       90  0058              MOVB    DUMP_NAME,DMP_FAB+FAB$B_FNS
                                005F              $OPEN FAB = DMP_FAB             ; Open the dump file
                  01 50     E8  006A              BLBS    R0,20$
                            04  006D       10$:   RET
                                006E       20$:   $CONNECT RAB = DMP_RAB
                  F1 50     E9  0079              BLBC    R0,10$
                                007C       ;
      13AC'CF      01       D0  007C              MOVL    #1,DMP_RAB+RAB$L_BKT    ; Read the image header block
   1394'CF    0200 8F       B0  0081              MOVW    #512,DMP_RAB+RAB$W_USZ  ; One block
   1398'CF    1124'CF       9E  0088              MOVAB   IMGHDR,DMP_RAB+RAB$L_UBF
                                008F              $READ RAB = DMP_RAB
                  D0 50     E9  009A              BLBC    R0,10$
   1B1B 8F    1944'CF       B1  009D              CMPW    SYS$INPUT_TRN,#<27+<27@8>> ; ESC-ESC means subprocess
                     03     12  00A4              BNEQ    25$
                   00A4     31  00A6              BRW     40$                     ; Skip all the image name stuff
         52   1124'CF       9E  00A9       25$:   MOVAB   IMGHDR,R2
           50   06 A2       3C  00AE              MOVZWL  IHD$W_IMGIDOFF(R2),R0   ; Get to image name
              52   50       C0  00B2              ADDL    R0,R2
              50   62       9A  00B5              MOVZBL  IHI$T_IMGNAM(R2),R0     ; Image name length
   1C64'CF    140C'CF       9E  00B8              MOVAB   MISC,IMGFAB+FAB$L_DNA   ; Image name will be here
   1C69'CF    50   04       81  00BF              ADDB3   #4,R0,IMGFAB+FAB$B_DNS  ; Allow for .EXE in image name
140C'CF   01 A2    50       28  00C5              MOVC3   R0,IHI$T_IMGNAM+1(R2),MISC ; Save image name for use as default
   63    4558452E 8F        D0  00CC              MOVL    #^A/.EXE7,(R3)          ; Add default extension to end of MOVC3
                                00D3       ;
              18EC'CF       9F  00D3              PUSHAB  CLI_IMAGE               ; See if image qualifier is there
    00000000'GF    01       FB  00D7              CALLS   #1,G^CLI$PRESENT
                  20 50     E9  00DE              BLBC    R0,30$
              1614'CF       9F  00E1              PUSHAB  IMAGE_DESC              ; Get image name
              18EC'CF       9F  00E5              PUSHAB  CLI_IMAGE
    00000000'GF    02       FB  00E9              CALLS   #2,G^CLI$GET_VALUE
                  0E 50     E9  00F0              BLBC    R0,30$
   1C68'CF    1614'CF       90  00F3              MOVB    IMAGE_DESC,IMGFAB+FAB$B_FNS
   1C60'CF    1618'CF       D0  00FA              MOVL    IMAGE_DESC+4,IMGFAB+FAB$L_FNA
                                0101       30$:
                                0101              $OPEN   FAB = IMGFAB
              52   50       D0  010C              MOVL    R0,R2                   ; Save status
   0AF3'CF    1C8F'CF       9A  010F              MOVZBL  IMGNAM+NAM$B_ESL,IMAGE  ; Expanded file name length
   0AF7'CF    1C90'CF       D0  0116              MOVL    IMGNAM+NAM$L_ESA,IMAGE+4
                                011D              $CLOSE  FAB = IMGFAB            ; We don't really need the file
              50   52       D0  0128              MOVL    R2,R0                   ; Get back the status
                 0F 50      E8  012B              BLBS    R0,35$                  ; The image file is there
    00000000'8F    50       D1  012E              CMPL    R0,#RMS$_FNF
                     79     12  0135              BNEQ    50$                     ; A real error
              0AF3'CF       D4  0137              CLRL    IMAGE                   ; Treat like /NOIMAGE
                     10     11  013B              BRB     40$
                                013D       ;
              18F9'CF       9F  013D       35$:   PUSHAB  CLI_NOIMAGE             ; Is noimage qualifier there
         0000'CF   01       FB  0141              CALLS   #1,CLI$PRESENT
                  04 50     E9  0146              BLBC    R0,40$
              0AF3'CF       D4  0149              CLRL    IMAGE                   ; Indicate no image to be loaded
                                014D       ;
      13AC'CF      02       D0  014D       40$:   MOVL    #2,DMP_RAB+RAB$L_BKT    ; Read misc data block
   1394'CF    0200 8F       B0  0152              MOVW    #512,DMP_RAB+RAB$W_USZ  ; One block
   1398'CF    140C'CF       9E  0159              MOVAB   MISC,DMP_RAB+RAB$L_UBF
                                0160              $READ RAB = DMP_RAB
                  42 50     E9  016B              BLBC    R0,50$
```

```
                                016E       ;
          03     1450'CF   D1   016E               CMPL    MISC+IMGDMP$L_VERSION,#IMGDMP$C_VERSION
                      09   13   0173               BEQL    43$                    ; Versions of dump and program match
      50   00000000'8F     D0   0175               MOVL    #SS$_BADFILEVER,R0
                      32   11   017C               BRB     50$
                                017E       ;
                    0845   30   017E       43$:    BSBW    BLD_MISC_VA            ; Build table of misc VA's
                                0181       ;
    1B1B 8F      1944'CF   B1   0181               CMPW    SYS$INPUT_TRN,#<27+<27@8>> ; ESC-ESC means subprocess
                      05   13   0188               BEQL    45$                    ; We already did the display
        0000'CF       00   FB   018A               CALLS   #0,DISPLAY_DUMP        ; Display the dump data
                                018F       ;
           13AC'CF    03   D0   018F       45$:    MOVL    #3,DMP_RAB+RAB$L_BKT   ; Read first map block
   1394'CF     0200 8F     B0   0194               MOVW    #512,DMP_RAB+RAB$W_USZ ; One block
   1398'CF     0F24'CF     9E   019B               MOVAB   MAP,DMP_RAB+RAB$L_UBF
                                01A2               $READ RAB = DMP_RAB
                   01 50   E8   01AD               BLBS    R0,60$
                      04        01B0       50$:    RET
                                01B1       60$:    $DISCONNECT RAB = DMP_RAB
                   F1 50   E9   01BC               BLBC    R0,50$
                                01BF               $CLOSE FAB = DMP_FAB           ; Close file before image activation
                   E3 50   E9   01CA               BLBC    R0,50$
                                01CD       ;
                                01CD       ; Decide if we can do the job in this process or if a subprocess is needed to
                                01CD       ; make room for the saved stack.
                                01CD       ;
  0000000C'GF    144C'CF   D1   01CD               CMPL    MISC+IMGDMP$L_USRSTK,G^CTL$AL_STACK+12
                      0D   1B   01D6               BLEQU   70$                    ; No problem
    1B1B 8F      1944'CF   B1   01D8               CMPW    SYS$INPUT_TRN,#<27+<27@8>> ; ESC-ESC means subprocess
                      31   13   01DF               BEQL    80$                    ; Already a subprocess
                    0036   30   01E1               BSBW    CREATE_SUBP            ; Go create a subprocess
                      04        01E4               RET
                                01E5       ;
    1C18'CF      1D09'CF   D0   01E5       70$:    MOVL    COND_MSG+4,OUTRAB+RAB$L_RBF ; First half of condition message
    1C12'CF      1D05'CF   B0   01EC               MOVW    COND_MSG,OUTRAB+RAB$W_RSZ   ;  it is easier to do unrelocated
                                01F3               $PUT    RAB = OUTRAB
                   AF 50   E9   01FE               BLBC    R0,50$
      5E   00001381'8F     C2   0201               SUBL    #MOVE_END-MOVE_BEG,SP
  6E    0561'CF  1381'8F   28   0208               MOVC3   #MOVE_END-MOVE_BEG,MOVE_BEG,(SP) ; Move the code
                      6E   17   0210               JMP     (SP)    ; Relocate execution
                                0212       ;
                                0212       ; No hope of analyzing this dump.
                                0212       ;
      50   00000000'8F     D0   0212       80$:    MOVL    #SS$_VASFULL,R0
                      04        0219               RET
```

```
                              021A   ;
                              021A   ; Create a subprocess to execute this image so we can fix the user stack in
                              021A   ; its old position.
                              021A   ;
                              021A   CREATE_SUBP:
                              021A   ;
                              021A   ; Announce what we are doing
                              021A   ;
1C18'CF   1CE8'CF     D0      021A           MOVL    CRE_SUB_MSG+4,OUTRAB+RAB$L_RBF
1C12'CF   1CE4'CF     B0      0221           MOVW    CRE_SUB_MSG,OUTRAB+RAB$W_RSZ
                              0228           $PUT    RAB = OUTRAB
              1A 50   E9      0233           BLBC    R0,10$
                              0236   ;
                              0236           $CREMBX_S CHAN = INP_MBX,-        ; Mailbox for new process SYS$INPUT
                              0236                     MAXMSG = #256
              01 50   E8      024D           BLBS    R0,20$
                      05      0250   10$:    RSB
                              0251   20$:    $CREMBX_S CHAN = TERM_MBX         ; Termination mailbox for the created proces
              E9 50   E9      0264           BLBC    R0,10$
            1908'CF   9F      0267           PUSHAB  SYS$INPUT                 ; Get recursive translation of SYS$INPUT
            1984'CF   9F      026B           PUSHAB  INPUT
               0289   30      026F           BSBW    GET_TRAN
            5E   08   C0      0272           ADDL    #8,SP
              D8 50   E9      0275           BLBC    R0,10$
            1919'CF   9F      0278           PUSHAB  SYS$OUTPUT                ; Get recursive translation of SYS$OUTPUT
            19CC'CF   9F      027C           PUSHAB  OUTPUT
               0278   30      0280           BSBW    GET_TRAN
            5E   08   C0      0283           ADDL    #8,SP
              C7 50   E9      0286           BLBC    R0,10$
            192B'CF   9F      0289           PUSHAB  SYS$ERROR                 ; Get recursive translation of SYS$ERROR
            1A14'CF   9F      028D           PUSHAB  ERROR
               0267   30      0291           BSBW    GET_TRAN
            5E   08   C0      0294           ADDL    #8,SP
              B6 50   E9      0297           BLBC    R0,10$
            1A5C'CF   DD      029A           PUSHL   INP_MBX                   ; Get unit number of mailbox
            1A64'CF   9F      029E           PUSHAB  INP_MBX_UNIT
               029D   30      02A2           BSBW    MBX_UNIT
            5E   08   C0      02A5           ADDL    #8,SP
              A5 50   E9      02A8           BLBC    R0,10$
            1A60'CF   DD      02AB           PUSHL   TERM_MBX                  ; Get unit number of mailbox
            1A68'CF   9F      02AF           PUSHAB  TERM_MBX_UNIT
               028C   30      02B3           BSBW    MBX_UNIT
            5E   08   C0      02B6           ADDL    #8,SP
              94 50   E9      02B9           BLBC    R0,10$
                              02BC           $QIOW_S CHAN = INP_MBX,-          ; Data for the new process
                              02BC                   FUNC = #IO$_WRITEVBLK!IO$M_NOW,-
                              02BC                   P1 = @INPUT+4,-
                              02BC                   P2 = INPUT
              4C 50   E9      02DF           BLBC    R0,30$
                              02E2           $QIOW_S CHAN = INP_MBX,-
                              02E2                   FUNC = #IO$_WRITEVBLK!IO$M_NOW,-
                              02E2                   P1 = @IMAGE+4,-
                              02E2                   P2 = IMAGE
              26 50   E9      0305           BLBC    R0,30$
                              0308           $QIOW_S CHAN = INP_MBX,-
                              0308                   FUNC = #IO$_WRITEVBLK!IO$M_NOW,-
                              0308                   P1 = NAME_BUFFER,-
```

```
                         0308     328                          P2 = DUMP_NAME
          01 50      E8  032B     329               BLBS       R0,40$
                     05  032E     330 30$:          RSB
                         032F     331 40$:          $FAO_S     CTRSTR = INPFAO,-           ; Get mailbox unit for SYS$INPUT
                         032F     332                          OUTLEN = INP_MBX_NAM,-
                         032F     333                          OUTBUF = INP_MBX_NAM,-
                         032F     334                          P1 = INP_MBX_UNIT
        18D6'CF      9F  0346     335               PUSHAB     OLD_CTRL
        18D2'CF      9F  034A     336               PUSHAB     CTRY_DISABL
 00000000'GF    02   FB  034E     337               CALLS      #2,G^LIB$DISABLE_CTRL       ; Disable DCL use of control-Y
                         0355     338               $ASSIGN_S DEVNAM = SYS$INPUT,-         ; Get channel for control-Y
                         0355     339                          CHAN    = INP_CHAN
                         0366     340               $QIO_S     CHAN = INP_CHAN,-           ; Request AST on control-Y
                         0366     341                          FUNC = #IO$_SETMODE!IO$M_CTRLYAST,-
                         0366     342                          P1    = CNTRLY_AST
                         0387     343               $CREPRC_S IMAGE = ANAL_IMG,-           ; Create the process
                         0387     344                            INPUT = INP_MBX_NAM,-
                         0387     345                            OUTPUT = OUTPUT,-
                         0387     346                            ERROR = ERROR,-
                         0387     347                            MBXUNT = TERM_MBX_UNIT,-
                         0387     348                            BASPRI = #4,-
                         0387     349                            PRVADR = L^CTL$GQ_PROCPRIV,-
                         0387     350                            PIDADR = SUBP_PID
          27 50      E9  03B9     351               BLBC       R0,45$
                         03BC     352               $QIOW_S CHAN = TERM_MBX,-
                         03BC     353                          FUNC = #IO$_READVBLK,-
                         03BC     354                          P1 = TERM_MSG,-
                         03BC     355                          P2 = #10
          05 50      E9  03DB     356               BLBC       R0,45$
     50   1ABB'CF    D0  03DE     357               MOVL       TERM_MSG+4,R0               ; Get process exit status
                50   DD  03E3     358 45$:          PUSHL      R0                          ; Save exit status
                7E   D4  03E5     359               CLRL       -(SP)
        18D6'CF      9F  03E7     360               PUSHAB     OLD_CTRL                    ; Original control-Y status
 00000000'GF    02   FB  03EB     361               CALLS      #2,G^LIB$ENABLE_CTRL        ; Restore DCL use of control-Y
                50   BA  03F2     362               POPR       R0
                     05  03F4     363 50$:          RSB
```

```
                        03F5    365 ;
                        03F5    366 ; AST routine for control-Y when a subprocess is active
                        03F5    367 ;
                        03F5    368 CNTRLY_AST:
                  0000  03F5    369         .WORD   0
                        03F7    370         $DELPRC_S PIDADR = SUBP_PID     ; Get rid of the subprocess
             7E     D4  0404    371         CLRL    -(SP)
        18D6'CF     9F  0406    372         PUSHAB  OLD_CTRL                ; Original control-Y status
00000000'GF  02     FB  040A    373         CALLS   #2,G^LIB$ENABLE_CTRL    ; Restore DCL use of control-Y
                        0411    374         $EXIT_S CODE = #SS$_NORMAL      ; Exit this image
                    04  041E    375         RET
```

```
                            041F    ;
                            041F    ; Initialization routines for running in subprocess
                            041F    ;
                            041F    INIT_SUBP:
     193C'CF    02    C2    041F            SUBL2   #2,INPUT_TRN     ; Get rid of the ESC-ESC
     1940'CF    02    C0    0424            ADDL    #2,INPUT_TRN+4
                            0429            $ASSIGN_S DEVNAM = INPUT_TRN,-
                            0429                      CHAN = INP_MBX
             01 50    E8    043A            BLBS    R0,20$
                      05    043D    10$:    PSB
                            043E    20$:    $QIOW_S CHAN = INP_MBX,-
                            043E                    FUNC = #IO$_READVBLK,-
                            043E                    IOSB = IOSB,-
                            043E                    P1 = @REAL_INPUT+4,-
                            043E                    P2 = REAL_INPUT
             D9 50    E9    0461            BLBC    R0,10$
        50   1B4D'CF  3C    0464            MOVZWL  IOSB,R0
             D1 50    E9    0469            BLBC    R0,10$
   1B55'CF   1B4F'CF  B0    046C            MOVW    IOSB+2,REAL_INPUT           ; Length
                            0473            $CRELOG_S LOGNAM = SYS$INPUT,-   ; Make SYS$INPUT correct
                            0473                      EQLNAM = REAL_INPUT,-
                            0473                      TBLFLG = #2
             B4 50    E9    0486            BLBC    R0,10$
                            0489            $QIOW_S CHAN = INP_MBX,-
                            0489                    FUNC = #IO$_READVBLK,-
                            0489                    IOSB = IOSB,-
                            0489                    P1 = @IMAGE_DESC+4,-
                            0489                    P2 = IMAGE_DESC
             4B 50    E9    04AC            BLBC    R0,30$
        50   1B4D'CF  3C    04AF            MOVZWL  IOSB,R0
             43 50    E9    04B4            BLBC    R0,30$
   0AF3'CF   1B4F'CF  B0    04B7            MOVW    IOSB+2,IMAGE     ; Length
   0AF7'CF   1618'CF  D0    04BE            MOVL    IMAGE_DESC+4,IMAGE+4
                            04C5            $QIOW_S CHAN = INP_MBX,-
                            04C5                    FUNC = #IO$_READVBLK,-
                            04C5                    IOSB = IOSB,-
                            04C5                    P1 = @DUMP_NAME+4,-
                            04C5                    P2 = DUMP_NAME
             0F 50    E9    04E8            BLBC    R0,30$
        50   1B4D'CF  3C    04EB            MOVZWL  IOSB,R0
             07 50    E9    04F0            BLBC    R0,30$
   13BC'CF   1B4F'CF  B0    04F3            MOVW    IOSB+2,DUMP_NAME            ; Length
                      05    04FA    30$:    RSB
```

```
                       04FB       ;
                       04FB       ; Misc routines for subprocess creation
                       04FB       ;
                       04FB       ;
                       04FB       ; Recursively translate a logical name
                       04FB       ;       Inputs
                       04FB       ;        address of descriptor of output buffer
                       04FB       ;        address of descriptor of input name
                       04FB       ;
                       04FB       GET_TRAN:
      52   04 AE   D0  04FB               MOVL    4(SP),R2        ; Output descriptor
    1AC1'CF   62   7D  04FF               MOVQ    (R2),LOG_OUT    ; Output descriptor
 1AC9'CF   08 BE   7D  0504               MOVQ    @8(SP),LOG_IN   ; Name to be translated
                       050A       10$:    $TRNLOG_S LOGNAM = LOG_IN,-
                       050A                         RSLLEN = (R2),-
                       050A                         RSLBUF = LOG_OUT
           10 50   E9  0521               BLBC    R0,20$
00000000'8F   50   D1  0524               CMPL    R0,#SS$_NOTRAN
              07   13  052B               BEQL    20$             ; All done
    1AC9'CF   62   7D  052D               MOVQ    (R2),LOG_IN     ; Result of the last try
              D6   11  0532               BRB     10$
      1B   04 B2   B1  0534       20$:    CMPW    @4(R2),#27      ; ESC-0 means PPF
              07   12  0538               BNEQ    30$
         62   04   C2  053A               SUBL    #4,(R2)         ; Get rid of PPF header
      04 A2   04   C0  053D               ADDL    #4,4(R2)
                   05  0541       30$:    RSB
                       0542       ;
                       0542       ;
                       0542       ; Get a mailbox unit number
                       0542       ;       Inputs
                       0542       ;         address to return unit number
                       0542       ;         channel number
                       0542       ;
                       0542       MBX_UNIT:
      51   08 AE   D0  0542               MOVL    8(SP),R1        ; Channel
                       0546               $GETCHN_S CHAN = R1,-
                       0546                         PRIBUF = MBXCHAR
 04 BE   1AE5'CF   3C  055A               MOVZWL  MBXCHARBUF+DIB$W_UNIT,@4(SP)
                   05  0560               RSB
                       0561
```

```
                                 0561      ;
                                 0561      ; The relocateable portion of the code begins here
                                 0561      ;
                                 0561      MOVE_BEG:
 57   5E   00000561'8F      C3   0561              SUBL3   #MOVE_BEG,SP,R7          ; Relocation constant
                                 0569              $CMKRNL_S ROUTIN = DELETE        ; Go delete P0 before image activation
                  01 50     E8   0576              BLBS    R0,10$
                            04   0579              RET
 0A98'CF    0000000C'9F     D0   057A      10$:    MOVL    @#CTL$AL_STACK+12,STACK_INI ; Save current stack base
 0A9C'CF    00000000'9F     D0   0583              MOVL    @#CTL$GL_IMGHDRBF,IMGHDRBF_INI ; Save pointer for analimdmp
                 0AF3'CF    D5   058C              TSTL    IMAGE                    ; Should we load an image
                      2B    13   0590              BEQL    40$                      ; No
                                 0592      ;
           0AF7'CF    57    C0   0592              ADDL    R7,IMAGE+4               ; Relocate image name
           0AC8'CF    57    C0   0597              ADDL    R7,IMG_DEFAULT+4
                                 059C              $CMEXEC_S ROUTIN = IMGACT        ; IMGACT it in EXEC mode so we are
                  0A 50     E8   05A9              BLBS    R0,30$                   ; able to restore privileges
       00000000'8F    50    D1   05AC              CMPL    R0,#RMS$_FNF
                      08    13   05B3              BEQL    40$                      ; Treat FNF like /NOIMAGE
                            04   05B5              RET
                                 05B6      30$:    $IMGFIX_S                        ; Address fixups
                                 05BD      ;
                                 05BD      40$:    $CMKRNL_S ROUTIN = RESET_VEC     ; Reset any privileged library vectors
                  01 50     E8   05CA              BLBS    R0,45$
                            04   05CD              RET
           0D0F'CF    57    C0   05CE      45$:    ADDL    R7,DEBUG+4               ; Relocate DEBUG name
           0ADF'CF    57    C0   05D3              ADDL    R7,DBG_DEFAULT+4
     0D03'CF    171B'CF     D0   05D8              MOVL    DEBUG_VA,DBG_RANGE       ; First unallocated P0
                                 05DF              $IMGACT_S  NAME = DEBUG,-        ; Merge in DEBUG
                                 05DF                         DFLNAM = DBG_DEFAULT,-
                                 05DF                         IMGCTL = #IAC$M_MERGE,-
                                 05DF                         INADR = DBG_RANGE,-
                                 05DF                         RETADR = DBG_RETADR,-
                                 05DF                         HDRBUF = DBG_HDRBUF
                  32 50     E9   0600              BLBC    R0,50$
                                 0603              $IMGFIX_S
                                 060A      ;
           13B0'CF    57    C0   060A              ADDL    R7,DMP_RAB+RAB$L_FAB     ; Relocate file name stuff
           1350'CF    57    C0   060F              ADDL    R7,DMP_FAB+FAB$L_FNA
           1354'CF    57    C0   0614              ADDL    R7,DMP_FAB+FAB$L_DNA
                                 0619              $OPEN FAB = DMP_FAB              ; Open the dump file again
                  0E 50     E9   0624              BLBC    R0,50$
                                 0627              $CONNECT RAB = DMP_RAB
                  01 50     E8   0632              BLBS    R0,60$
                            04   0635      50$:    RET
                    03C2    30   0636      60$:    BSBW    RESTORE_MISC_VA          ; Restore the misc VA
                                 0639      ;
                                 0639      ; Do original process address space
                                 0639      ;
           13AC'CF    04    D0   0639              MOVL    #4,DMP_RAB+RAB$L_BKT     ; First data block
           55   0F24'CF     9E   063E              MOVAB   MAP,R5                   ; Map pointer
                52    85    D0   0643      70$:    MOVL    (R5)+,R2                 ; Page count
                      03    18   0646              BGEQ    80$
                    0283    30   0648              BSBW    NEXT_MAP                 ; Nothing left in this map block
                      03    12   064B      80$:    BNEQ    90$
                    016B    31   064D              BRW     GET_DEBUG                ; All done - go to DEBUG
             03 65    1E    E1   0650      90$:    BBC     #30,(R5),95$
```

```
                     0084   31  0654   519          BRW     GET_P1                      ; Go restore P1
            1398'CF    85   D0  0657   520 95$:     MOVL    (R5)+,DMP_RAB+RAB$L_UBF     ; Starting VA
    53   52  00000200 8F   C5  065C   521 100$:    MULL3   #512,R2,R3                  ; Byte count
        0000FE00 8F    53   D1  0664   522          CMPL    R3,#<127*512>               ; Is it greater than maximum
                       07   1B  066B   523          BLEQU   101$                        ; No
        53   0000FE00 8F   D0  066D   524          MOVL    #<127*512>,R3
          160C'CF  1398'CF  D0  0674   525 101$:    MOVL    DMP_RAB+RAB$L_UBF,CREATE_PAGE ; Set up to create the page
   1610'CF   1398'CF   53   C1  067B   526          ADDL3   R3,DMP_RAB+RAB$L_UBF,CREATE_PAGE+4
                  1610'CF   D7  0683   527          DECL    CREATE_PAGE+4               ; Top of range
                                0687   528          $CRETVA_S INADR = CREATE_PAGE,-
                                0687   529                  RETADR = CREATE_PAGE
                    15 50   E8  0698   530          BLBS    R0,110$                     ; Were there any problems
                                069B   531          ; Not owner of page
    53   1610'CF  160C'CF   C3  069B   532          SUBL3   CREATE_PAGE,CREATE_PAGE+4,R3 ; Created byte count - 1
                       09   12  06A3   533          BNEQ    105$                        ; Any pages created?
         53   00000200 8F   D0  06A5   534          MOVL    #512,R3                     ; Skip one page
                       15   11  06AC   535          BRB     115$                        ; Avoid the read
                       53   D6  06AE   536 105$:    INCL    R3                          ; Make it byte count
            1394'CF    53   B0  06B0   537 110$:    MOVW    R3,DMP_RAB+RAB$W_USZ
                                06B5   538          $READ RAB = DMP_RAB                 ; Read the dump page
                    17 50   E9  06C0   539          BLBC    R0,120$
            1398'CF    53   C0  06C3   540 115$:    ADDL    R3,DMP_RAB+RAB$L_UBF
          53   53    F7 8F   78  06C8   541          ASHL    #-9,R3,R3                   ; Page count
            13AC'CF    53   C0  06CD   542          ADDL    R3,DMP_RAB+RAB$L_BKT
                 52    53   C2  06D2   543          SUBL    R3,R2                       ; Remaining page count
                       85   14  06D5   544          BGTR    100$
                     FF69   31  06D7   545          BRW     70$
                            04  06DA   546 120$:    RET
                                06DB   547 ;
                                06DB   548 GET_P1:
            50   0B03'CF   9E  06DB   549          MOVAB   IMG_HDRBUF,R0               ; Normalize image header
            0B03'CF    50   C2  06E0   550          SUBL    R0,IMG_HDRBUF
            0B07'CF    50   C2  06E5   551          SUBL    R0,IMG_HDRBUF+4
            13B0'CF    57   C2  06EA   552          SUBL    R7,DMP_RAB+RAB$L_FAB        ; and the RAB
                 55    04   C2  06EF   553          SUBL    #4,R5                       ; Reset map pointer
                                06F2   554 ;
                                06F2   555 ; Space is created and the code is relocated to the top of P0
                                06F2   556 ; A P0 stack (one page) is also created
                                06F2   557 ;
                                06F2   558          $EXPREG_S PAGCNT = #<<MOVE_END-MOVE_BEG>+511/512>+1,-
                                06F2   559                  RETADR = NEW_P0
                       55   DD  0707   560          PUSHL   R5
 1404'DF    FE51 CF   1381'8F   28  0709   561          MOVC3   #MOVE_END-MOVE_BEG,MOVE_BEG,@NEW_P0
                 55    8E   D0  0713   562          MOVL    (SP)+,R5
            50   0F24'CF   9E  0716   563          MOVAB   MAP,R0
                 55    50   C2  071B   564          SUBL    R0,R5                       ; Normalize map pointer
            50   1404'CF   D0  071E   565          MOVL    NEW_P0,R0                   ; Address code was moved to
             000001C8'E0   17  0723   566          JMP     <10$-MOVE_BEG>(R0)          ; Relocate execution
                                0729   567 ;
     57    50   00000561'8F   C3  0729   568 10$:     SUBL3   #MOVE_BEG,R0,R7             ; Relocation constant
            13B0'CF    57   C0  0731   569          ADDL    R7,DMP_RAB+RAB$L_FAB
            50   0F24'CF   9E  0736   570          MOVAB   MAP,R0
                 55    50   C0  073B   571          ADDL    R0,R5                       ; Relocate map pointer
            50   0B03'CF   9E  073E   572          MOVAB   IMG_HDRBUF,R0               ; and image header
            0B03'CF    50   C0  0743   573          ADDL    R0,IMG_HDRBUF
            0B07'CF    50   C0  0748   574          ADDL    R0,IMG_HDRBUF+4
            5E   1408'CF   D0  074D   575          MOVL    NEW_P0+4,SP                 ; Get stack out of the way of P1
        0760'CF    00    FB  0752   576         CALLS   #0,15$                     ; Create top frame for new stack
                             0757   577         $EXIT_S CODE = R0                  ; Get out with correct status
                             0760   578 ;
                     0000    0760   579 15$:    .WORD   0
                    6D   D4  0762   580         CLRL    (FP)                       ; Terminate frames
                             0764   581 ;
              52   85    D0  0764   582 20$:    MOVL    (R5)+,R2                   ; Page count
                   03    18  0767   583         BGEQ    30$
                 0162    30  0769   584         BSBW    NEXT_MAP                   ; We have finished this map block
                   4D    13  076C   585 30$:    BEQL    GET_DEBUG                  ; All done - go start DEBUG
        1398'CF    85    D0  076E   586         MOVL    (R5)+,DMP_RAB+RAB$L_UBF    ; Starting VA
   160C'CF    1398'CF    D0  0773   587 40$:    MOVL    DMP_RAB+RAB$L_UBF,CREATE_PAGE ; Set up to create the page
   1610'CF    1398'CF    D0  077A   588         MOVL    DMP_RAB+RAB$L_UBF,CREATE_PAGE+4
                             0781   589         $CRETVA_S INADR = CREATE_PAGE
                27 50    E9  0790   590         BLBC    R0,50$                     ; Not owner of page
   1394'CF    0200 8F    B0  0793   591         MOVW    #512,DMP_RAB+RAB$W_USZ     ; One block
                             079A   592         $READ RAB = DMP_RAB
                12 50    E9  07A5   593         BLBC    R0,50$
              13AC'CF    D6  07A8   594         INCL    DMP_RAB+RAB$L_BKT          ; Next file block
 1398'CF  00000200 8F    C0  07AC   595         ADDL    #512,DMP_RAB+RAB$L_UBF     ; Next page
                BB 52    F5  07B5   596         SOBGTR  R2,40$
                   AA    11  07B8   597         BRB     20$                        ; Go do next run of pages
                             07BA   598 ;
                   04        07BA   599 50$:    RET
                             07BB   600 ;
                             07BB   601 GET_DEBUG:
                             07BB   602         $CMKRNL_S ROUTIN = FIX_STACK       ; Put stack limits back
                             07C8   603         $CMKRNL_S ROUTIN = FIX_IMGHDRBF    ; Restore CTL$GL_IMGHDRBF
                             07D5   604         $CMKRNL_S ROUTIN = RESET_PRIV      ; Eliminate the image privileges
              0AF3'CF    D5  07E2   605         TSTL    IMAGE                      ; Was an image loaded
                   10    12  07E6   606         BNEQ    10$
   0AB4'CF    17CB'CF    D0  07E8   607         MOVL    THIS_HDR,DBG_ARG+12        ; Let DEBUG try to look at this image
   0AB8'CF    17CF'CF    D0  07EF   608         MOVL    THIS_HDR+4,DBG_ARG+16      ; just to keep it happy
                   0E    11  07F6   609         BRB     20$
   0AB4'CF    0B03'CF    D0  07F8   610 10$:    MOVL    IMG_HDRBUF,DBG_ARG+12      ; Arguments to start DEBUG
   0AB8'CF    0B07'CF    D0  07FF   611         MOVL    IMG_HDRBUF+4,DBG_ARG+16
        50    0D18'CF    D0  0806   612 20$:    MOVL    DBG_RETADR,R0
   0F20'CF    08 B040    9E  080B   613         MOVAB   @8(R0)[R0],DEBUG_BEG       ; Find debug transfer address
                             0812   614 ;
                             0812   615 ; Display the original cause of the dump
                             0812   616 ;
        52    1440'CF    D0  0812   617         MOVL    MISC+IMGDMP$L_AP,R2
  04 B2    000F0000 8F   C8  0817   618         BISL    #^XF@16,@4(R2)              ; Add flags to display all message fields
                             081F   619         $PUTMSG_S MSGVEC = @4(R2)           ; Ignore errors
  04 B2    000F0000 8F   CA  082F   620         BICL    #^XF@16,@4(R2)              ; Clear the flags
                             0837   621
        5E    143C'CF    D0  0837   622         MOVL    MISC+IMGDMP$L_SP,SP        ; Restore registers
        5C    1440'CF    D0  083C   623         MOVL    MISC+IMGDMP$L_AP,AP
        50    140C'CF    7D  0841   624         MOVQ    MISC+IMGDMP$L_R0,R0
        52    1414'CF    7D  0846   625         MOVQ    MISC+IMGDMP$L_R2,R2
        54    141C'CF    7D  084B   626         MOVQ    MISC+IMGDMP$L_R4,R4
        56    1424'CF    7D  0850   627         MOVQ    MISC+IMGDMP$L_R6,R6
        58    142C'CF    7D  0855   628         MOVQ    MISC+IMGDMP$L_R8,R8
        5A    1434'CF    7D  085A   629         MOVQ    MISC+IMGDMP$L_R10,R10
                             085F   630 ;
                             085F   631 ; Build a phony SS$_DEBUG frame
                             085F   632 ;
```

```
         50   04 AC   D0  085F   633          MOVL    4(AP),R0                   ; Real signal array
           51    60   D0  0863   634          MOVL    (R0),R1
               6041   DD  0866   635          PUSHL   (R0)[R1]                   ; PC, PSL
            FC A041   DD  0869   636          PUSHL   -4(R0)[R1]
        00000000'8F   DD  086D   637          PUSHL   #SS$_DEBUG
                 03   DD  0873   638          PUSHL   #3
            51   5E   D0  0875   639          MOVL    SP,R1
            1410'CF   DD  0878   640          PUSHL   MISC+IMGDMP$L_R1
            140C'CF   DD  087C   641          PUSHL   MISC+IMGDMP$L_R0
                 7E   D4  0880   642          CLRL    -(SP)
              0C AD   D4  0882   643          CLRL    12(FP)                     ; Make this final frame in case we need it
1444'DF    04    00   0C  0885   644          PROBER  #0,#4,@MISC+IMGDMP$L_FP
                 05   13  088B   645          BEQL    25$                        ; The FP is no good
     5D    1444'CF    D0  088D   646          MOVL    MISC+IMGDMP$L_FP,FP        ; Real FP
                 5D   DD  0892   647 25$:     PUSHL   FP
            50   5D   D0  0894   648          MOVL    FP,R0
   0C A0   04    00   0C  0897   649 30$:     PROBER  #0,#4,12(R0)               ; In case of a corrupted stack
                 0B   13  089C   650          BEQL    40$
         50   0C A0   D0  089E   651          MOVL    12(R0),R0                  ; Trace back FP
                 05   13  08A2   652          BEQL    40$                        ; End of the chain
            6E   50   D0  08A4   653          MOVL    R0,(SP)                    ; This is a good one
                 EE   11  08A7   654          BRB     30$
                 04   DD  08A9   655 40$:     PUSHL   #4
                 5E   DD  08AB   656          PUSHL   SP                         , Mechanism array
                 51   DD  08AD   657          PUSHL   R1                         ; Signal array
                 02   DD  08AF   658          PUSHL   #2
            5C   5E   D0  08B1   659          MOVL    SP,AP                      ; Phony SS$_DEBUG frame
    08B9'CF      6C   FA  08B4   660          CALLG   (AP),50$                   ; DEBUG likes to see the resulting frame
                    0FFC  08B9   661 50$:     .WORD   ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
    0AAC'CF      5C   D0  08BB   662          MOVL    AP,DBG_ARG+4
     5C    0AA8'CF    9E  08C0   663          MOVAB   DBG_ARG,AP
     50    0000'CF    D0  08C5   664          MOVL    SS$_NORMAL,R0              ; DEBUG is used to this
           0F20'DF    17  08CA   665          JMP     @DEBUG_BEG
                          08CE   666 ;
                          08CE   667 ;
                          08CE   668 ;
                          08CE   669 ; Get a new map block
                          08CE   670 ;
                          08CE   671 NEXT_MAP:
            1398'CF   DD  08CE   672          PUSHL   DMP_RAB+RAB$L_UBF          ; Save current VA
1394'CF    0200 8F    B0  08D2   673          MOVW    #512,DMP_RAB+RAB$W_USZ     ; One block
1398'CF    0F24'CF    9E  08D9   674          MOVAB   MAP,DMP_RAB+RAB$L_UBF
                          08E0   675          $READ   RAB = DMP_RAB
              12 50   E9  08EB   676          BLBC    R0,10$
            13AC'CF   D6  08EE   677          INCL    DMP_RAB+RAB$L_BKT          ; Next block
     1398'CF     8E   D0  08F2   678          MOVL    (SP)+,DMP_RAB+RAB$L_UBF    ; Restore VA
     55    0F24'CF    9E  08F7   679          MOVAB   MAP,R5
            52   85   D0  08FC   680          MOVL    (R5)+,R2                   ; Get next count
                      05  08FF   681          RSB
                          0900   682 ;
                      04  0900   683 10$:     RET                                ; Error - give up
```

```
                                  0901   ;
                                  0901   ; Misc privileged routines
                                  0901   ;
                                  0901   ;
                                  0901   ;
                                  0901   ; Delete all of P0
                                  0901   ;
                                  0901   DELETE:
                          0044    0901           .WORD   ^M<R2,R6>
             00000000'9F    16    0903           JSB     @#EXE$RESETVEC          ; Reset privileged library stuff
                                  0909           $DELTVA_S INADR = ALL_P0
                                  0918   ;
                                  0918   ; Deallocate all image control blocks that describe currently activated images
                                  0918   ;
      52     00000000'GF    7E    0918           MOVAQ   G^IAC$GL_ICBFL,R2       ; R2 = address of free list
                                  091F
      51     00000000'GF    7E    091F           MOVAQ   G^IAC$GL_IMAGE_LIST,R1  ; R1 = listhead of ICBS
               56    00 B1  0F    0926   5$:     REMQUE  @(R1),R6                ; Remove next entry
                         06  1D   092A           BVS     10$                     ; List empty - all done
             04 B2       66  0E   092C           INSQUE  (R6),@4(R2)             ; Insert at end of free list
                         F4  11   0930           BRB     5$                      ; Go back for more
                                  0932
                             04   0932   10$:    RET
                                  0933   ;
                                  0933   ; Fix up the stack limit
                                  0933   ;
                                  0933   FIX_STACK:
                          0000    0933           .WORD   0
  0000000C'9F       0A98'CF  D0   0935           MOVL    STACK_INI,@#CTL$AL_STACK+12
                             04   093E           RET
                                  093F   ;
                                  093F   ; Fix up CTL$GL_IMGHDRBF
                                  093F   ;
                                  093F   FIX_IMGHDRBF:
                          0000    093F           .WORD   0
  00000000'9F       0A9C'CF  D0   0941           MOVL    IMGHDRBF_INI,@#CTL$GL_IMGHDRBF
                             04   094A           RET
                                  094B   ;
                                  094B   ; Reset privileges to get rid of image privileges
                                  094B   ;
                                  094B   RESET_PRIV:
                          0000    094B           .WORD   0
      50     00000000'9F     D0   094D           MOVL    @#CTL$GL_PHD,R0
                  00E8 C0    7C   0954           CLRQ    PHD$Q_IMAGPRIV(R0)       ; No more authorized image provs
      7E     00000004'9F     D2   0958           MCOML   @#CTL$GQ_PROCPRIV+4,-(SP) ; Complement of the permanent privileges
      7E     00000000'9F     D2   095F           MCOML   @#CTL$GQ_PROCPRIV,-(SP)
                    51   5E  D0   0966           MOVL    SP,R1
                                  0969           $SETPRV_S ENBFLG = #0,-
                                  0969                   PRVADR = (R1)
                             04   0978           RET
                                  0979   ;
                                  0979   ; Reset privileged library vectors
                                  0979   ;
                                  0979   RESET_VEC:
                          0044    0979           .WORD   ^M<R2,R6>
             00000000'9F    16    097B           JSB     @#EXE$RESETVEC          ; Reset privileged library stuff
      50     00000000'8F     D0   0981           MOVL    #SS$_NORMAL,R0
```

```
              04  0988   742          RET
                  0989   743 ;
                  0989   744 ; IMGACT the original image and then reset the image privileges
                  0989   745 ;
                  0989   746 IMGACT:
            0000  0989   747          .WORD   0
                  098B   748          $IMGACT_S  NAME = IMAGE,-        ; Activate original image
                  098B   749                     DFLNAM = IMG_DEFAULT,-
                  098B   750                     RETADR = IMG_RETADR,-
                  098B   751                     HDRBUF = IMG_HDRBUF
       50     DD  09A8   752          PUSHL   R0
  7E   03     7D  09AA   753          MOVQ    #<1@PRV$V_CMKRNL>!<1@PRV$V_CMEXEC>,-(SP) ; Restore privileges
  51   5E     D0  09AD   754          MOVL    SP,R1
                  09B0   755          $SETPRV_S  ENBFLG = #1,-
                  09B0   756                     PRVADR = (R1)
  50   8E     7D  09BF   757          MOVQ    (SP)+,R0                  ; Clean up the stack
       50   8ED0  09C2   758          POPL    R0
              04  09C5   759          RET
```

```
                              09C6      ;
                              09C6      ; Routines to handle misc address space
                              09C6      ;
                              09C6      ;
                              09C6      ; A table is used
                              09C6      ;       count of table entries
                              09C6      ;       longword offset of size in MISC
                              09C6      ;       longword offset of file block in MISC
                              09C6      ;       longword offset of VA in MISC_VA
                              09C6      ;
                              09C6      ; Build table of VA's of misc pieces of address space
                              09C6      ;
                              09C6      BLD_MISC_VA:
      50   1454'CF      D0    09C6              MOVL    MISC+IMGDMP$L_FREE_P0,R0 ; Start here
      51   177F'CF      9E    09CB              MOVAB   MISC_CONTROL,R1         ; Table address
           52   81      D0    09D0              MOVL    (R1)+,R2                ; Number of entries
           53   61      D0    09D3      10$:    MOVL    (R1),R3                 ; Offset for size
      53   140C'CF43    D0    09D6              MOVL    MISC[R3],R3             ; Size
                11      13    09DC              BEQL    20$                     ; Nothing saved for this one
        54   08 A1      D0    09DE              MOVL    8(R1),R4                ; Offset for VA
   171F'CF44    50      D0    09E2              MOVL    R0,MISC_VA[R4]          ; Save VA
      53   53   09      78    09E8              ASHL    #9,R3,R3                ; Page count -> bytes
           50   53      C0    09EC              ADDL    R3,R0                   ; New VA
           51   0C      C0    09EF      20$:    ADDL    #12,R1                  ; Next entry
           DE   52      F5    09F2              SOBGTR  R2,10$
      171B'CF   50      D0    09F5              MOVL    R0,DEBUG_VA             ; Start DEBUG at the end
                        05    09FA              RSB
                              09FB      ;
                              09FB      ; Restore misc VA
                              09FB      ;
                              09FB      RESTORE_MISC_VA:
      56   177F'CF      9E    09FB              MOVAB   MISC_CONTROL,R6         ; Table address
      57   171F'CF      9E    0A00              MOVAB   MISC_VA,R7
      58   140C'CF      9E    0A05              MOVAB   MISC,R8
           55   86      D0    0A0A              MOVL    (R6)+,R5                ; Number of entries
           52   66      D0    0A0D      10$:    MOVL    (R6),R2                 ; Offset of area size
        52   6842       D0    0A10              MOVL    (R8)[R2],R2             ; Size
                12      13    0A14              BEQL    20$                     ; Nothing there
        53   08 A6      D0    0A16              MOVL    8(R6),R3                ; Offset of VA
        53   6743       D0    0A1A              MOVL    (R7)[R3],R3             ; VA
        54   04 A6      D0    0A1E              MOVL    4(R6),R4                ; Offset of file block
        54   6844       D0    0A22              MOVL    (R8)[R4],R4             ; File block
                07      10    0A26              BSBB    READ_ONE_VA
           56   0C      C0    0A28      20$:    ADDL    #12,R6                  ; Next entry
           DF   55      F5    0A2B              SOBGTR  R5,10$
                        05    0A2E              RSB
                              0A2F      ;
                              0A2F      ;
                              0A2F      ; Read a piece of address space
                              0A2F      ;       R2 = page count
                              0A2F      ;       R3 = starting VA
                              0A2F      ;       R4 = starting file block
                              0A2F      ;
                              0A2F      READ_ONE_VA:
   13AC'CF      54      D0    0A2F              MOVL    R4,DMP_RAB+RAB$L_BKT
   1398'CF      53      D0    0A34      10$:    MOVL    R3,DMP_RAB+RAB$L_UBF    ; Address
   160C'CF      53      D0    0A39              MOVL    R3,CREATE_PAGE          ; Create the address range
```

```
                54   52   D0  0A3E    818          MOVL    R2,R4                    ; Remaining page count
       0000FE00 8F   54   D1  0A41    819          CMPL    R4,#<127*512>            ; Compare with maximum
                     07   1B  0A48    820          BLEQU   20$
       54   0000007F 8F   D0  0A4A    821          MOVL    #127,R4                  ; Use maximum
            54   54  09   78  0A51    822  20$:    ASHL    #9,R4,R4
            1394'CF  54   B0  0A55    823          MOVW    R4,DMP_RAB+RAB$W_USZ     ; Byte count
                     54   D7  0A5A    824          DECL    R4
     1610'CF    53   54   C1  0A5C    825          ADDL3   R4,R3,CREATE_PAGE+4
                              0A62    826          $CRETVA_S INADR = CREATE_PAGE
                23 50     E9  0A71    827          BLBC    R0,50$
                              0A74    828          $READ   RAB = DMP_RAB
                15 50     E9  0A7F    829          BLBC    R0,50$
                     54   D6  0A82    830          INCL R4
                53   54   C0  0A84    831          ADDL    R4,R3                    ; Update address
        54   54   F7 8F   78  0A87    832          ASHL    #-9,R4,R4                ; Get back the page count
           13AC'CF   54   C0  0A8C    833          ADDL    R4,DMP_RAB+RAB$L_BKT
                52   54   C2  0A91    834          SUBL    R4,R2                    ; Any pages left?
                     9E   14  0A94    835          BGTR    10$                      ; Continue
                          05  0A96    836          RSB
                          04  0A97    837  50$:    RET                              ; Error
```

```
                                       0A98    839 ;
                                       0A98    840 ; Data
                                       0A98    841 ;
                                       0A98    842 ;
                                       0A98    843 STACK_INI:                                      ; Stack limit before $IMGACT
                             00000000  0A98    844         .LONG   0
                                       0A9C    845 ;
                                       0A9C    846 IMGHDRBF_INI:                                    ; CTL$GL_IMGHDRBF before our IMGACTs
                             00000000  0A9C    847         .LONG   0
                                       0AA0    848 ;
                    3FFFFFFF 00000000  0AA0    849 ALL_P0: .LONG   0,^X3FFFFFFF                    ; Range to delete all of P0
                                       0AA8    850 ;
                                       0AA8    851 DBG_ARG:                                        ; Argument list to call DEBUG
00000000 00000000 00000000 00000006    0AA8    852         .LONG   6,0,0,0,0,0,CLI$M_DBGEXCP
         00000000'00000000 00000000    0AB8
                                       0AC4    853 ;
                                       0AC4    854 IMG_DEFAULT:                                    ; Default name for images
59 53 24 53 59 53 00000ACC'010E0000'   0AC4    855         .ASCID  /SYS$SYSTEM:.EXE/
         45 58 45 2E 3A 4D 45 54 53    0AD2
                                       0ADB    856 ;
                                       0ADB    857 DBG_DEFAULT:                                    ; Default name for DEBUG
49 4C 24 53 59 53 00000AE3'010E0000'   0ADB    858         .ASCID  /SYS$LIBRARY:.EXE/
      45 58 45 2E 3A 59 52 41 52 42    0AE9
                                       0AF3    859 ;
                    00000000 00000000  0AF3    860 IMAGE:  .LONG   0,0                             ; Descriptor for image name
                                       0AFB    861 ;
                                       0AFB    862 IMG_RETADR:                                     ; Address range of image
                             00000B03  0AFB    863         .BLKL   2
                                       0B03    864 ;
                                       0B03    865 IMG_HDRBUF:                                     ; IMGACT buffer for image
                             00000D03  0B03    866         .BLKB   512
                                       0D03    867 ;
                                       0D03    868 DBG_RANGE:                                      ; Range to merge in DEBUG
                    3FFFFFFF 00000000  0D03    869         .LONG   0,^X3FFFFFFF
                                       0D0B    870 ;
   47 55 42 45 44 00000D13'010E0000'   0D0B    871 DEBUG:  .ASCID  /DEBUG/                         ; Name of DEBUG
                                       0D18    872 ;
                                       0D18    873 DBG_RETADR:                                     ; Address range used by DEBUG
                             00000D20  0D18    874         .BLKL   2
                                       0D20    875 ;
                                       0D20    876 DBG_HDRBUF:                                     ; IMGACT buffer for DEBUG
                             00000F20  0D20    877         .BLKB   512
                                       0F20    878 ;
                                       0F20    879 DEBUG_BEG:                                      ; Transfer address for DEBUG
                             00000F24  0F20    880         .BLKL   1
                                       0F24    881 ;
                             00001124  0F24    882 MAP:    .BLKB   512                             ; Address map buffer
                                       1124    883 ;
                                       1124    884 IMGHDR::                                        ; First block of image header
                             00001324  1124    885         .BLKB   512
                                       1324    886 ;
                                       1324    887         .ALIGN  LONG
                                       1324    888 ;
                                       1324    889 DMP_FAB: $FAB   FAC = <BIO,GET>,-
                                       1324    890                 FNA = NAME_BUFFER,-
                                       1324    891                 DNA = DFLNAM,-
                                       1324    892                 DNS = 4
```

```
                     1374    893 ;
                     1374    894 DMP_RAB: $RAB    FAB = DMP_FAB,-
                     1374    895                  USZ = 512,-
                     1374    896                  ROP = BIO
         50 4D 44 2E 13B8    897 DFLNAM: .ASCII  /.DMP/                      ; Default name for dumps
                     13BC    898 ;
                     13BC    899 DUMP_NAME:                                  ; Descriptor for name of dump file
            00000040 13BC    900         .LONG   64
           000013C4' 13C0    901         .ADDRESS NAME_BUFFER
                     13C4    902 NAME_BUFFER:
            00001404 13C4    903         .BLKB   64
                     1404    904 ;
            0000140C 1404    905 NEW_P0: .BLKL   2                           ; P0 space for relocating code
                     140C    906 ;
            0000160C 140C    907 MISC::  .BLKB   512                         ; Buffer for misc data block
                     160C    908 ;
                     160C    909 CREATE_PAGE:                                ; Address range for a page to be created
            00001614 160C    910         .BLKL   2
                     1614    911 ;
                     1614    912 IMAGE_DESC:                                 ; Desriptor for image name
            000000FF 1614    913         .LONG   NAM$C_MAXRSS
           0000161C' 1618    914         .ADDRESS 1$
            0000171B 161C    915 1$:     .BLKB   NAM$C_MAXRSS
                     171B    916 ;
                     171B    917 DEBUG_VA:                                   ; VA for DEBUG
            00000000 171B    918         .LONG   0
                     171F    919 ;
                     171F    920 MISC_VA::                                   ; VA's of misc data areas
            0000177F 171F    921         .BLKL   IMGMOV$C_LENGTH
                     177F    922 ;
                     177F    923 MISC_CONTROL:                               ; Description of misc address space
           00000006' 177F    924         .LONG   <1$ - MISC_CONTROL>/12
            0000002E 1783    925         .LONG   IMGDMP$L_KSTK_SIZ/4         ; Kernel stack
            0000002C 1787    926         .LONG   IMGDMP$L_KSTK_BLK/4
            00000000 178B    927         .LONG   IMGMOV$L_KSTK74
            00000031 178F    928         .LONG   IMGDMP$L_ESTK_SIZ/4         ; Exec stack
            0000002F 1793    929         .LONG   IMGDMP$L_ESTK_BLK/4
            00000001 1797    930         .LONG   IMGMOV$L_ESTK74
            00000034 179B    931         .LONG   IMGDMP$L_VECPAG_SIZ/4       ; Vector page
            00000032 179F    932         .LONG   IMGDMP$L_VECPAG_BLK/4
            00000002 17A3    933         .LONG   IMGMOV$L_VECPAG74
            00000037 17A7    934         .LONG   IMGDMP$L_PIO_SIZ/4          ; PIO (RMS) area
            00000035 17AB    935         .LONG   IMGDMP$L_PIO_BLK/4
            00000003 17AF    936         .LONG   IMGMOV$L_PIO74
            0000003A 17B3    937         .LONG   IMGDMP$L_IMGCTX_SIZ/4       ; Image activator context pages
            00000038 17B7    938         .LONG   IMGDMP$L_IMGCTX_BLK/4
            00000004 17BB    939         .LONG   IMGMOV$L_IMGCTX74
            0000003D 17BF    940         .LONG   IMGDMP$L_USRCTX_SIZ/4       ; User writeable context pages
            0000003B 17C3    941         .LONG   IMGDMP$L_USRCTX_BLK/4
            00000005 17C7    942         .LONG   IMGMOV$L_USRCTX74
                     17CB    943 1$:
                     17CB    944 ;
                     17CB    945 THIS_HDR:                                   ; Pointers to this image's header buffer
            000017D3 17CB    946         .BLKL   2
                     17D3    947 ;
                     17D3    948 IMGNAMESS:                                  ; ESS from open of image
            000018D2 17D3    949         .BLKB   NAM$C_MAXRSS
```

```
                                  18D2  ;
                                  18D2  CTRY_DISABL:                            ; Mask to disable control-Y
                       00000000'  18D2          .LONG   LIB$M_CLI_CTRLY
                                  18D6  ;
                                  18D6  OLD_CTRL:                               ; Control-Y state
                       00000000   18D6          .LONG   0
                                  18DA  ;
                                  18DA  INP_CHAN:                               ; SYS$INPUT channel
                       00000000   18DA          .LONG   0
                                  18DE  ;
                                  18DE  SUBP_PID:                               ; PID of subprocess
                       00000000   18DE          .LONG   0
                                  18E2  ;
                                  18E2  MOVE_END:                               ; End of code to be relocated
                                  18E2  ;
                                  18E2  ;
                                  18E2  ; Data after this point is not relocated
                                  18E2  ;
                                  18E2  ;
                                  18E2  CLI_PARAMETER:                          ; Get the command line parameter
          31 50 000018EA'010E0000'  18E2          .ASCID  /P1/
                                  18EC  ;
                                  18EC  CLI_IMAGE:                              ; The /IMAGE qualifier
 45 47 41 4D 49 000018F4'010E0000'  18EC          .ASCID  /IMAGE/
                                  18F9  ;
                                  18F9  CLI_NOIMAGE:                            ; The /NOIMAGE qualifier
47 41 4D 49 4F 4E 00001901'010E0000'  18F9          .ASCID  /NOIMAGE/
                              45  1907
                                  1908  ;
                                  1908  SYS$INPUT:                              ; Stings for TRNLOG
4E 49 24 53 59 53 00001910'010E0000'  1908          .ASCID  /SYS$INPUT/
                        54 55 50  1916
                                  1919  SYS$OUTPUT:
55 4F 24 53 59 53 00001921'010E0000'  1919          .ASCID  /SYS$OUTPUT/
                     54 55 50 54  1927
                                  192B  SYS$ERROR:
52 45 24 53 59 53 00001933'010E0000'  192B          .ASCID  /SYS$ERROR/
                        52 4F 52  1939
                                  193C  ;
                                  193C  INPUT_TRN:                              ; Translation of initial SYS$INPUT
                       00000040   193C          .LONG   64
                       00001944'  1940          .ADDRESS SYS$INPUT_TRN
                                  1944  SYS$INPUT_TRN:
                       00001984   1944          .BLKB   64
                                  1984  ;
                       00000040   1984  INPUT:  .LONG   64                      ; Output strings from TRNLOG
                       0000198C'  1988          .ADDRESS 1$
                       000019CC   198C  1$:     .BLKB   64
                       00000040   19CC  OUTPUT: .LONG   64
                       000019D4'  19D0          .ADDRESS 1$
                       00001A14   19D4  1$:     .BLKB   64
                       00000040   1A14  ERROR:  .LONG   64
                       00001A1C'  1A18          .ADDRESS 1$
                       00001A5C   1A1C  1$:     .BLKB   64
                                  1A5C  ;
                                  1A5C  INP_MBX:                                ; Channel for communications mailbox
                       00000000   1A5C          .LONG   0
```

```
                                    1A60  ;
                                    1A60  TERM_MBX:                                ; Channel for termination mailbox
                          00000000  1A60          .LONG   0
                                    1A64  ;
                                    1A64  INP_MBX_UNIT:                            ; Unit number for communications mailbox
                          00000000  1A64          .LONG   0
                                    1A68  ;
                                    1A68  TERM_MBX_UNIT:                           ; Unit number for termination mailbox
                          00000000  1A68          .LONG   0
                                    1A6C  ;
41 42 4D 5F 1B 1B 00001A74'010E0000' 1A6C  INPFAO: .ASCID  <27><27>/_MBA!5ZW:/      ; FAO string for mailbox name + ESCs
                    3A 57 5A 35 21  1A7A
                                    1A7F  ;
                                    1A7F  INP_MBX_NAM:                             ; Communications mailbox name
                          0000000E  1A7F          .LONG   14
                         00001A87'  1A83          .ADDRESS 1$
                          00001A97  1A87  1$:     .BLKB   16
                                    1A97  ;
                                    1A97  ANAL_IMG:                                ; Name of this image for CREPRC
59 53 24 53 59 53 00001A9F'010E0000' 1A97          .ASCID  /SYS$SYSTEM:ANALIMDMP.EXE/
44 4D 49 4C 41 4E 41 3A 4D 45 54 53  1AA5
                 45 58 45 2E 50 4D  1AB1
                                    1AB7  ;
                                    1AB7  TERM_MSG:                                ; Termination mailbox message
                          00001AC1  1AB7          .BLKB   10
                                    1AC1  ;
                                    1AC1  LOG_OUT:                                 ; Descriptor for output logical name
                          00001AC9  1AC1          .BLKL   2
                                    1AC9  ;
                                    1AC9  LOG_IN:                                  ; Descriptor for input logical name
                          00001AD1  1AC9          .BLKL   2
                                    1AD1  ;
                                    1AD1  MBXCHAR:
                          00000074  1AD1          .LONG   DIB$K_LENGTH
                         00001AD9'  1AD5          .ADDRESS MBXCHARBUF
                                    1AD9  MBXCHARBUF:                              ; Buffer for malbox characteristics
                          00001B4D  1AD9          .BLKB   DIB$K_LENGTH
                                    1B4D  ;
                          00001B55  1B4D  IOSB:   .BLKL   2                        ; IOSB for mailbox use
                                    1B55  ;
                                    1B55  REAL_INPUT:                              ; Real SYS$INPUT when in subprocess
                          00000040  1B55          .LONG   64
                         00001B5D'  1B59          .ADDRESS 1$
                          00001B9D  1B5D  1$:     .BLKB   64
                                    1B9D  ;
                                    1B9D          .ALIGN  LONG
                                    1BA0  ;
                                    1BA0  OUTFAB: $FAB    FNM = <SYS$OUTPUT>,-     ; FAB for SYS$OUTPUT
                                    1BA0                  FAC = <GET,PUT>,-
                                    1BA0                  FOP = <CIF>
                                    1BF0  ;
                                    1BF0  OUTRAB: $RAB    FAB = OUTFAB
                                    1C34  ;
                                    1C34  IMGFAB: $FAB    DNM = <.EXE>,-
                                    1C34                  NAM = IMGNAM
                                    1C84  ;
                                    1C84  IMGNAM: $NAM    ESS = NAM$C_MAXRSS,-
```

```
                                              1C84  1057                  ESA = IMGNAMESS
                                              1CE4  1058 ;
72 43 0A 0D 0A 0D 00001CEC'010E0000'  1CE4  1059 CRE_SUB_MSG: .ASCID <13><10><13><10>/Creating a subprocess/
62 75 73 20 61 20 67 6E 69 74 61 65   1CF2
                  73 73 65 63 6F 72 70   1CFE
                                              1D05  1060 ;
64 6E 6F 43 0A 0D 00001D0D'010E0000'  1D05  1061 COND_MSG:    .ASCID <13><10>/Condition signalled to take dump:/
6C 61 6E 67 69 73 20 6E 6F 69 74 69   1D13
20 65 6B 61 74 20 6F 74 20 64 65 6C   1D1F
                           3A 70 6D 75 64   1D2B
                                              1D30  1062 ;
                                              1D30  1063          .END     ANALIMDMP
```

```
$$.TAB                         = 00001C84 R       01       FAB$V_BIO                      = 00000005
$$.TABEND                      = 00001CE4 R       01       FAB$V_CHAN_MODE                = 00000002
$$.TMP                         = 00000000                  FAB$V_CIF                      = 00000019
$$.TMP1                        = 00000001                  FAB$V_FILE_MODE                = 00000004
$$.TMP2                        = 000000CF                  FAB$V_GET                      = 00000001
$$.TMPX                        = 0000000A R       03       FAB$V_LNM_MODE                 = 00000000
$$.TMPX1                       = 00000004                  FAB$V_PUT                      = 00000000
$$T1                           = 00000001                  FAB$W_GBC                      = 00000048
$$T2                           = 00000004                  FIX_IMGHDRBF                     0000093F R       01
ALL_PO                           00000AA0 R       01       FIX_STACK                        00000933 R       01
ANALIMDMP                        00000000 R       01       GET_DEBUG                        000007BB R       01
ANAL_IMG                         00001A97 R       01       GET_P1                           000006DB R       01
BLD_MISC_VA                      000009C6 R       01       GET_TRAN                         000004FB R       01
CLI$GET_VALUE                    ********     X   01       IAC$GL_ICBFL                     ********     X   01
CLI$M_DBGEXCP                    ********     X   01       IAC$GL_IMAGE_LIST                ********     X   01
CLI$PRESENT                      ********     X   01       IAC$M_MERGE                    = 00000010
CLI_IMAGE                        000018EC R       01       IHD$W_IMGIDOFF                 = 00000006
CLI_NOIMAGE                      000018F9 R       01       IHI$T_IMGNAM                   = 00000000
CLI_PARAMETER                    000018E2 R       01       IMAGE                            00000AF3 R       01
CNTRLY_AST                       000003F5 R       01       IMAGE_DESC                       00001614 R       01
COND_MSG                         00001D05 R       01       IMGACT                           00000989 R       01
CREATE_PAGE                      0000160C R       01       IMGDMP                         = 00000000
CREATE_SUBP                      0000021A R       01       IMGDMP$C_LENGTH                = 000000F8
CRE_SUB_MSG                      00001CE4 R       01       IMGDMP$C_VERSION               = 00000003
CTL$AL_STACK                     ********     X   01       IMGDMP$L_AP                    = 00000034
CTL$GL_IMGHDRBF                  ********     X   01       IMGDMP$L_ASTACT                = 00000050
CTL$GL_PHD                       ********     X   01       IMGDMP$L_ASTCNT                = 000000A0
CTL$GQ_PROCPRIV                  ********     X   01       IMGDMP$L_ASTEN                 = 00000054
CTRY_DISABL                      000018D2 R       01       IMGDMP$L_ASTLM                 = 000000A4
DBG_ARG                          00000AA8 R       01       IMGDMP$L_BIOCNT                = 00000068
DBG_DEFAULT                      00000ADB R       01       IMGDMP$L_BIOLM                 = 0000006C
DBG_HDRBUF                       00000D20 R       01       IMGDMP$L_BUFIO                 = 00000070
DBG_RANGE                        00000D03 R       01       IMGDMP$L_BYTCNT                = 00000074
DBG_RETADR                       00000D18 R       01       IMGDMP$L_BYTLM                 = 00000078
DEBUG                            00000D0B R       01       IMGDMP$L_DIOCNT                = 0000007C
DEBUG_BEG                        00000F20 R       01       IMGDMP$L_DIOLM                 = 00000080
DEBUG_VA                         0000171B R       01       IMGDMP$L_DIRIO                 = 00000084
DELETE                           00000901 R       01       IMGDMP$L_EFCS                  = 00000060
DFLNAM                           000013B8 R       01       IMGDMP$L_EFCU                  = 00000064
DIB$K_LENGTH                   = 00000074                  IMGDMP$L_ENQCNT                = 000000A8
DIB$W_UNIT                     = 0000000C                  IMGDMP$L_ENQLM                 = 000000AC
DISPLAY_DUMP                     ********     X   01       IMGDMP$L_ESTK_BLK              = 000000BC
DMP_FAB                          00001324 R       01       IMGDMP$L_ESTK_SIZ              = 000000C4
DMP_RAB                          00001374 R       01       IMGDMP$L_ESTK_VA               = 000000C0
DUMP_NAME                        000013BC R       01       IMGDMP$L_FILCNT                = 00000088
ERROR                            00001A14 R       01       IMGDMP$L_FILLM                 = 0000008C
EXE$RESETVEC                     ********     X   01       IMGDMP$L_FIRST_MAP             = 0000003C
FAB$B_DNS                      = 00000035                  IMGDMP$L_FP                    = 00000038
FAB$B_FNS                      = 00000034                  IMGDMP$L_FREE_P0               = 00000048
FAB$C_BID                      = 00000003                  IMGDMP$L_FREE_P1               = 0000004C
FAB$C_BLN                      = 00000050                  IMGDMP$L_IMGCTX_BLK            = 000000E0
FAB$C_SEQ                      = 00000000                  IMGDMP$L_IMGCTX_SIZ            = 000000E8
FAB$C_VAR                      = 00000002                  IMGDMP$L_IMGCTX_VA             = 000000E4
FAB$L_ALQ                      = 00000010                  IMGDMP$L_KSTK_BLK              = 000000B0
FAB$L_DNA                      = 00000030                  IMGDMP$L_KSTK_SIZ              = 000000B8
FAB$L_FNA                      = 0000002C                  IMGDMP$L_KSTK_VA               = 000000B4
FAB$L_FOP                      = 00000004                  IMGDMP$L_PIO_BLK               = 000000D4
```

```
IMGDMP$L_PIO_SIZ              = 000000DC            IO$_READVBLK        = 00000031
IMGDMP$L_PIO_VA               = 000000D8            IO$_SETMODE         = 00000023
IMGDMP$L_PRCCNT               = 00000090            IO$_WRITEVBLK       = 00000030
IMGDMP$L_PRCLM                = 00000094            IOSB                  00001B4D R        01
IMGDMP$L_R0                   = 00000000            LIB$DISABLE_CTRL      ********       X   01
IMGDMP$L_R1                   = 00000004            LIB$ENABLE_CTRL       ********       X   01
IMGDMP$L_R10                  = 00000028            LIB$M_CLI_CTRLY       ********       X   01
IMGDMP$L_R11                  = 0000002C            LOG_IN                00001AC9 R        01
IMGDMP$L_R2                   = 00000008            LOG_OUT               00001AC1 R        01
IMGDMP$L_R3                   = 0000000C            MAP                   00000F24 R        01
IMGDMP$L_R4                   = 00000010            MBXCHAR               00001AD1 R        01
IMGDMP$L_R5                   = 00000014            MBXCHARBUF            00001AD9 R        01
IMGDMP$L_R6                   = 00000018            MBX_UNIT              00000542 R        01
IMGDMP$L_R7                   = 0000001C            MISC                  0000140C RG       01
IMGDMP$L_R8                   = 00000020            MISC_CONTROL          0000177F R        01
IMGDMP$L_R9                   = 00000024            MISC_VA               0000171F RG       01
IMGDMP$L_SP                   = 00000030            MOVE_BEG              00000561 R        01
IMGDMP$L_TQCNT                = 00000098            MOVE_END              000018E2 R        01
IMGDMP$L_TQLM                 = 0000009C            NAM$B_ESL           = 0000000B
IMGDMP$L_USRCTX_BLK           = 000000EC            NAM$B_ESS           = 0000000A
IMGDMP$L_USRCTX_SIZ           = 000000F4            NAM$B_NOP           = 00000008
IMGDMP$L_USRCTX_VA            = 000000F0            NAM$B_RSS           = 00000002
IMGDMP$L_USRSTK               = 00000040            NAM$C_BID           = 00000002
IMGDMP$L_VECPAG_BLK           = 000000C8            NAM$C_BLN           = 00000060
IMGDMP$L_VECPAG_SIZ           = 000000D0            NAM$C_MAXRSS        = 000000FF
IMGDMP$L_VECPAG_VA            = 000000CC            NAM$L_ESA           = 0000000C
IMGDMP$L_VERSION              = 00000044            NAM$L_RSA           = 00000004
IMGDMP$Q_CURPRIV              = 00000058            NAME_BUFFER           000013C4 R        01
IMGDMP$S_CURPRIV              = 00000008            NEW_PO                00001404 R        01
IMGDMP$S_IMGDMP               = 000000F8            NEXT_MAP              000008CE R        01
IMGFAB                          00001C34 R        01  OLD_CTRL              000018D6 R        01
IMGHDR                          00001124 RG       01  OUTFAB                00001BA0 R        01
IMGHDRBF_INI                    00000A9C R        01  OUTPUT                000019CC R        01
IMGMOV                        = 00000000            OUTRAB                00001BF0 R        01
IMGMOV$C_LENGTH               = 00000018            PHD$Q_IMAGPRIV      = 000000E8
IMGMOV$L_ESTK                 = 00000004            PRV$V_CMEXEC        = 00000001
IMGMOV$L_IMGCTX               = 00000010            PRV$V_CMKRNL        = 00000000
IMGMOV$L_KSTK                 = 00000000            RAB$B_RAC           = 0000001E
IMGMOV$L_PIO                  = 0000000C            RAB$C_BID           = 00000001
IMGMOV$L_USRCTX               = 00000014            RAB$C_BLN           = 00000044
IMGMOV$L_VECPAG               = 00000008            RAB$C_SEQ           = 00000000
IMGMOV$S_IMGMOV               = 00000018            RAB$L_CTX           = 00000018
IMGNAM                          00001C84 R        01  RAB$L_ROP           = 00000004
IMGNAMESS                       000017D3 R        01  RAB$V_BIO           = 0000000B
IMG_DEFAULT                     00000AC4 R        01  READ_ONE_VA           00000A2F R        01
IMG_HDRBUF                      00000B03 R        01  REAL_INPUT            00001B55 R        01
IMG_RETADR                      00000AFB R        01  RESET_PRIV            0000094B R        01
INIT_SUBP                       0000041F R        01  RESET_VEC             00000979 R        01
INPFAO                          00001A6C R        01  RESTORE_MISC_VA       000009FB R        01
INPUT                           00001984 R        01  RMS$_FNF              ********       X   01
INPUT_TRN                       0000193C R        01  SS$_BADFILEVER        ********       X   01
INP_CHAN                        000018DA R        01  SS$_DEBUG             ********       X   01
INP_MBX                         00001A5C R        01  SS$_NORMAL            ********       X   01
INP_MBX_NAM                     00001A7F R        01  SS$_NOTRAN            ********       X   01
INP_MBX_UNIT                    00001A64 R        01  SS$_VASFULL           ********       X   01
IO$M_CTRLYAST                 = 00000080            STACK_INI             00000A98 R        01
IO$M_NOW                      = 00000040            SUBP_PID              000018DE R        01
```

```
SYS$ASSIGN                 ********  GX   01
SYS$CLOSE                  ********  GX   01
SYS$CMEXEC                 ********  GX   01
SYS$CMKRNL                 ********  GX   01
SYS$CONNECT                ********  GX   01
SYS$CREATE                 ********  GX   01
SYS$CRELOG                 ********  GX   01
SYS$CREMBX                 ********  GX   01
SYS$CREPRC                 ********  GX   01
SYS$CRETVA                 ********  GX   01
SYS$DELPRC                 ********  GX   01
SYS$DELTVA                 ********  GX   01
SYS$DISCONNECT             ********  GX   01
SYS$ERROR                  0000192B R     01
SYS$EXIT                   ********  GX   01
SYS$EXPREG                 ********  GX   01
SYS$FAO                    ********   X   01
SYS$GETCHN                 ********  GX   01
SYS$IMGACT                 ********  GX   01
SYS$IMGFIX                 ********  GX   01
SYS$INPUT                  00001908 R     01
SYS$INPUT_TRN              00001944 R     01
SYS$OPEN                   ********  GX   01
SYS$OUTPUT                 00001919 R     01
SYS$PUT                    ********  GX   01
SYS$PUTMSG                 ********  GX   01
SYS$QIO                    ********  GX   01
SYS$QIOW                   ********  GX   01
SYS$READ                   ********  GX   01
SYS$SETPRV                 ********  GX   01
SYS$TRNLOG                 ********  GX   01
TERM_MBX                   00001A60 R     01
TERM_MBX_UNIT              00001A68 R     01
TERM_MSG                   00001AB7 R     01
THIS_HDR                   000017CB R     01

                                          +----------------+
                                          ! Psect synopsis !
                                          +----------------+

PSECT name                Allocation          PSECT No.  Attributes
----------                ----------          ---------  ----------
.  ABS  .                 00000000  (     0.)  00 (  0.)  NOPIC   USR   CON   ABS   LCL NOSHR NOEXE NORD  NOWRT NOVEC BYTE
ANALIMDMP                 00001D30  (  7472.)  01 (  1.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD    WRT NOVEC LONG
$ABS$                     00000000  (     0.)  02 (  2.)  NOPIC   USR   CON   ABS   LCL NOSHR   EXE   RD    WRT NOVEC BYTE
$RMSNAM                   0000000E  (    14.)  03 (  3.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD    WRT NOVEC BYTE

                                        +------------------------+
                                        ! Performance indicators !
                                        +------------------------+

Phase                   Page faults    CPU Time       Elapsed Time
-----                   -----------    --------       ------------
Initialization                  36     00:00:00.09    00:00:01.23
Command processing             139     00:00:00.77    00:00:03.81
Pass 1                         488     00:00:19.26    00:00:42.53
Symbol table sort                0     00:00:02.09    00:00:04.22
```

```
Pass 2                          198     00:00:04.37     00:00:09.38
Symbol table output              34     00:00:00.22     00:00:00.49
Psect synopsis output             5     00:00:00.03     00:00:00.03
Cross-reference output            0     00:00:00.00     00:00:00.00
Assembler run totals            903     00:00:26.84     00:01:01.69

The working set limit was 1500 pages.
105212 bytes (206 pages) of virtual memory were used to buffer the intermediate code.
There were 80 pages of symbol table space allocated to hold 1373 non-local and 71 local symbols.
1063 source lines were read in Pass 1, producing 29 object records in Pass 2.
65 pages of virtual memory were used to define 55 macros.

                                                +----------------------------+
                                                ! Macro library statistics !
                                                +----------------------------+

Macro library nam                               Macros defined
-----------------                               --------------
_$255$DUA28:[IMGDMP.OBJ]IMGDMPLIB.MLB;1                  2
_$255$DUA28:[SYS.OBJ]LIB.MLB;1                           3
_$255$DUA28:[SYSLIB]STARLET.MLB;2                       47
TOTALS (all libraries)                                  52

1781 GETS were required to define 52 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:ANALIMDMP/OBJ=OBJ$:ANALIMDMP MSRC$:ANALIMDMP/UPDATE=(ENH$:ANALIMDMP)+EXECML$/LIB+LIB$:IMGDMPLIB/LIB
```

0186 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY