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
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**IMGDMP                                    L 11
```

```
  IIIIII    MM      MM    GGGGGGGG  DDDDDDDD    MM      MM  PPPPPPPP
  IIIIII    MM      MM    GGGGGGGG  DDDDDDDD    MM      MM  PPPPPPPP
    II      MMMM  MMMM  GG          DD      DD  MMMM  MMMM  PP      PP
    II      MMMM  MMMM  GG          DD      DD  MMMM  MMMM  PP      PP
    II      MM  MM  MM  GG          DD      DD  MM  MM  MM  PP      PP
    II      MM  MM  MM  GG          DD      DD  MM  MM  MM  PP      PP
    II      MM      MM  GG          DD      DD  MM      MM  PPPPPPPP
    II      MM      MM  GG          DD      DD  MM      MM  PPPPPPPP
    II      MM      MM  GG  GGGGGG  DD      DD  MM      MM  PP
    II      MM      MM  GG  GGGGGG  DD      DD  MM      MM  PP
    II      MM      MM  GG      GG  DD      DD  MM      MM  PP              ....
    II      MM      MM  GG      GG  DD      DD  MM      MM  PP              ....
  IIIIII    MM      MM    GGGGGG    DDDDDDDD    MM      MM  PP              ....
  IIIIII    MM      MM    GGGGGG    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  IMGDMP
0000          .IDENT  /V04-000/
0000
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:      Dump address space and other context of a process.
0000   ;
0000   ;
0000   ; ENVIRONMENT:  User mode
0000   ;
0000   ; AUTHOR:        Wayne Cardoza
0000   ;
0000   ; CREATION DATE: 14-Feb-1983
0000   ;
0000   ; MODIFIED BY:
0000   ;
0000   ;       V03-006 WMC0006         Wayne Cardoza   12-Jun-1984
0000   ;               Fix incorrect register after MOVC3
0000   ;               Improve performance.
0000   ;
0000   ;       V03-005 WMC0005         Wayne Cardoza   06-Sep-1983
0000   ;               Add .DMP to file name to keep logical name for image from
0000   ;               having an effect.
0000   ;
0000   ;       V03-004 WMC0004         Wayne Cardoza   29-Aug-1983
0000   ;               Properly save SP in EXEC mode.
0000   ;
0000   ;       V03-003 WMC0003         Wayne Cardoza   01-Jul-1983
0000   ;               SYS$IMGACT has been redesigned.
0000   ;
0000   ;       V03-002 WMC0002         Wayne Cardoza   30-Apr-1983
```

```
0000     58 ;                   More GETJPI items.
0000     59 ;
0000     60 ;           V03-001 WMC0001         Wayne Cardoza   20-Apr-1983
0000     61 ;                   Add SYS$PROCDMP translation.
0000     62 ;
```

```
                      0000  ;
                  00000000          .PSECT  $AAIMGDMP,LONG
                      0000          .DEFAULT DISPLACEMENT,WORD
                      0000  ;
                      0000  ; Data Structure Definitions
                      0000  ;
                      0000          $IMGDMPDEF
                      0000          $IHDDEF
                      0000          $IHIDEF
                      0000          $IFDDEF
                      0000          $JPIDEF
                      0000          $PSLDEF
                      0000          $SFDEF
                      0000  ;
                      0000  ; This is the standard header for debuggers.
                      0000  ; It must be the first thing in the image.
                      0000  ;
                      0000  BEGIN_DUMP:
 00000000 00000000    0000          .LONG   0,0
          0000000C'   0008          .LONG   BEGIN_CODE-BEGIN_DUMP
```

```
                                 000C   ;++
                                 000C   ;
                                 000C   ; Functional Description:
                                 000C   ;       This is the main routine for taking an image dump.  It is JMPed to
                                 000C   ;       after being merged in by a condition handler.
                                 000C   ;
                                 000C   ; Calling Sequence:
                                 000C   ;       JMP
                                 000C   ;
                                 000C   ; Input Parameters:
                                 000C   ;       signal and mechanism arrays
                                 000C   ;
                                 000C   ; Implicit Inputs:
                                 000C   ;       a great many
                                 000C   ;
                                 000C   ; Output Parameters:
                                 000C   ;       none
                                 000C   ;
                                 000C   ; Implicit Outputs:
                                 000C   ;       none
                                 000C   ;
                                 000C   ; Routines Called:
                                 000C   ;       none
                                 000C   ;
                                 000C   ; Routine Value:
                                 000C   ;       none
                                 000C   ;
                                 000C   ; Signals:
                                 000C   ;       none
                                 000C   ;
                                 000C   ; Side Effects:
                                 000C   ;       dump taken
                                 000C   ;
                                 000C   ;--
                                 000C   ;
                                 000C   BEGIN_CODE:
                                 000C   ;
                                 000C   ; Save registers, clean up frame, save misc process context
                                 000C   ;
               50    04 AC   D0  000C           MOVL    4(AP),R0                ; Address of signal array
               51    08 AC   D0  0010           MOVL    8(AP),R1                ; Address of mechanism array
          0836'CF       5C   D0  0014           MOVL    AP,MISC+IMGDMP$L_AP     ; AP for the condition handler
       0802'CF       0C A1   7D  0019           MOVQ    12(R1),MISC+IMGDMP$L_R0 ; Get R0 and R1 from mechanism array
       083A'CF       0C AD   D0  001F           MOVL    SF$L_SAVE_FP(FP),MISC+IMGDMP$L_FP
    080A'CF    14 AD    28   28  0025           MOVC3   #10*4,SF$L_SAVE_REGS(FP),MISC+IMGDMP$L_R2
       0832'CF       5D 3C   C1  002C           ADDL3   #SF$L_SAVE_REGS+<10*4>,FP,MISC+IMGDMP$L_SP
                                 0032   ;
                                 0032   ; Save the correct SP if we aren't in user mode
                                 0032   ;
                        50   DC  0032           MOVPSL  R0
       50    50    02   18   EF  0034           EXTZV   #PSL$V_CURMOD,#PSL$S_CURMOD,R0,R0
                   03   50   D1  0039           CMPL    R0,#PSL$C_USER
                        14   13  003C           BEQL    3$                      ; All OK, this is user mode
              0832'CF        D4  003E           CLRL    MISC+IMGDMP$L_SP        ; Tell service not to change anything
                                 0042           $ADJSTK_S ACMODE = #PSL$C_USER,-
                                 0042                   NEWADR = MISC+IMGDMP$L_SP ; Get the old value here
                                 0052   ;
```

```
                            0052   142 ; Make sure we can check the version number on analysis
                            0052   143 ;
        0846'CF    03   D0  0052   144 3$:      MOVL    #IMGDMP$C_VERSION,MISC+IMGDMP$L_VERSION
                            0057   145 ;
                            0057   146 ; Some misc data
                            0057   147 ;
                            0057   148          $GETJPI_S ITMLST = JPILST
   0842'CF    0000000C'9F   D0  006C   149      MOVL    @#CTL$AL_STACK+12,MISC+IMGDMP$L_USRSTK ; Top of user stack
                            0075   150 ;
                            0075   151 ; Get the image header, create dump file of same name as image
                            0075   152 ;
      52   00000004'9F  D0  0075   153          MOVL    @#MMG$IMGHDRBUF+4,R2      ; IFD
       03FE'CF   08 A2  3C  007C   154          MOVZWL  IFD$W_CHAN(R2),IMG_CHANNEL
                            0082   155          $QIOW_S CHAN = IMG_CHANNEL,-      ; Read first block of image header
                            0082   156                  FUNC = #IO$_READVBLK,-
                            0082   157                  P1   = IMGHDRBUF,-
                            0082   158                  P2   = #512,-
                            0082   159                  P3   = #1
            03 50       E8  00A9   160          BLBS    R0,4$
             01C9       31  00AC   161          BRW     EXIT
      52    0402'CF     9E  00AF   162 4$:      MOVAB   IMGHDRBUF,R2
         50    06 A2    3C  00B4   163          MOVZWL  IHD$W_IMGIDOFF(R2),R0     ; Get image name
            52   50     C0  00B8   164          ADDL    R0,R2
            50   62     9A  00BB   165          MOVZBL  IHI$T_IMGNAM(R2),R0       ; Length of image name
    0B74'CF   50   04   81  00BE   166          ADDB3   #4,R0,DMP_FAB+FAB$B_FNS ; Allow for the .DMP
  0702'CF   01 A2   50  28  00C4   167          MOVC3   R0,IHI$T_IMGNAM+1(R2),MAP+256 ; Move the image name
       63    504D442E 8F D0 00CB   168          MOVL    #^A/.DMP7,(R3)            ; Add default extension to end of MOVC3
    0B6C'CF    0702'CF  9E  00D2   169          MOVAB   MAP+256,DMP_FAB+FAB$L_FNA ; Address of file name
                            00D9   170 ;
                            00D9   171 ;        If the logical name SYS$PROCDMP exists it will be used to place the dump
                            00D9   172 ;
                            00D9   173          $TRNLOG_S LOGNAM = SYS$PROCDMP,-
                            00D9   174                    RSLBUF = PROCDMPNAM,-
                            00D9   175                    RSLLEN = PROCDMPNAM
            29 50       E9  00F2   176          BLBC    R0,5$                     ; Assume no translation
    00000000'8F   50    D1  00F5   177          CMPL    R0,#SS$_NOTRAN
               20       13  00FC   178          BEQL    5$
       51   0602'CF     9E  00FE   179          MOVAB   MAP,R1
       51   0A15'CF     C0  0103   180          ADDL    PROCDMPNAM,R1             ; Get to end of translation
     61   504D442E 8F   D0  0108   181          MOVL    #^A/.DMP/,(R1)            ; Add default extension
    0B70'CF   0602'CF   9E  010F   182          MOVAB   MAP,DMP_FAB+FAB$L_DNA     ; Put default in FAB
 0B75'CF   0A15'CF   04  81  0116   183          ADDB3   #4,PROCDMPNAM,DMP_FAB+FAB$B_DNS
                            011E   184 5$:      $CREATE FAB = DMP_FAB             ; Create dump file
                            0129   185          $CONNECT RAB = DMP_RAB
            03 50       E8  0134   186          BLBS    R0,20$
             013E       31  0137   187 10$:     BRW     EXIT
       0BC8'CF    01    D0  013A   188 20$:     MOVL    #1,DMP_RAB+RAB$L_BKT      ; First block
    0BB2'CF   0200 8F   B0  013F   189          MOVW    #512,DMP_RAB+RAB$W_RSZ    ; One page
    0BB8'CF   0402'CF   9E  0146   190          MOVAB   IMGHDRBUF,DMP_RAB+RAB$L_RBF
                            014D   191          $WRITE  RAB = DMP_RAB             ; Write image header to dump file
                            0158   192 ;
                            0158   193 ; Initialize to write P0 and user stack to dump file
                            0158   194 ;
       0BC8'CF    04    D0  0158   195          MOVL    #4,DMP_RAB+RAB$L_BKT      ; First data block
       0B35'CF    03    D0  015D   196          MOVL    #3,MAP_BLOCK              ; First map block
       083E'CF    03    D0  0162   197          MOVL    #3,MISC+IMGDMP$L_FIRST_MAP
  52   084A'CF    F7 8F 78  0167   198          ASHL    #-9,MISC+IMGDMP$C_FREE_P0,R2 ; Page count
```

```
                52   D6  016E          INCL    R2
           0BB8'CF   D4  0170          CLRL    DMP_RAB+RAB$L_RBF          ; VA
      55   0602'CF   9E  0174          MOVAB   MAP,R5                     ; Position in map block
                65   D4  0179          CLRL    (R5)                       ; Initialize the map
              00FB   30  017B          BSBW    WRITE_IT                   ; Process P0 space
                         017E    ;
 53  084E'CF  00000200 8F  C3  017E    SUBL3   #^X200,MISC+IMGDMP$L_FREE_P1,R3 ; Starting address of used stack
   52   0000000C'9F   53   C3  0188    SUBL3   R3,@#CTL$AL_STACK+12,R2
        52   52   F7 8F   78  0190     ASHL    #-9,R2,R2                  ; Page count
        0BB8'CF      53   D0  0195     MOVL    R3,DMP_RAB+RAB$L_RBF       ; VA
                   00DC   30  019A     BSBW    WRITE_IT                   ; Process P1 space
                         019D    ;
     0BC8'CF   0B35'CF   D0  019D      MOVL    MAP_BLOCK,DMP_RAB+RAB$L_BKT
     0BB2'CF   0200 8F   B0  01A4      MOVW    #512,DMP_RAB+RAB$W_RSZ     ; One page
     0BB8'CF   0602'CF   9E  01AB      MOVAB   MAP,DMP_RAB+RAB$L_RBF      ; Write the last map block
                         01B2          $WRITE RAB = DMP_RAB
                 01 50   E8  01BD      BLBS    R0,30$
                         05  01C0      RSB
                         01C1    ;
                          1C1    ; Save misc pieces of P1
                         .1C1    ; Errors on change mode calls are ignored - no priv
                         01C1    ;
                         01C1    30$:  $CMEXEC_S  ROUTIN = PRIV_STKS  ; Save EXEC and kernel stacks
    52   00000000'8F   D0  01CE        MOVL    #CTL$GL_VECTORS,R2         ; Base of vector page
              53   52   D0  01D5       MOVL    R2,R3                      ; Top (one page)
        08CE'CF   52   D0  01D8        MOVL    R2,MISC+IMGDMP$L_VECPAG_VA
   08CA'CF   0B39'CF   D0  01DD        MOVL    NEXT_FILE,MISC+IMGDMP$L_VECPAG_BLK ; The file block to be used
                 01B5   30  01E4       BSBW    WRITE_MISC
        08D2'CF   54   D0  01E7        MOVL    R4,MISC+IMGDMP$L_VECPAG_SIZ ; Actual page count
    52   FFFFFE00'8F   D0  01EC        MOVL    #PIO$A_TRACE-512,R2        ; Base of PIO (RMS) area
        53   09FF C2   9E  01F3        MOVAB   <<5*512>-1>(R2),R3         ; Top
        08DA'CF   52   D0  01F8        MOVL    R2,MISC+IMGDMP$L_PIO_VA
   08D6'CF   0B39'CF   D0  01FD        MOVL    NEXT_FILE,MISC+IMGDMP$L_PIO_BLK ; The file block to be used
                 0195   30  0204       BSBW    WRITE_MISC
        08DE'CF   54   D0  0207        MOVL    R4,MISC+IMGDMP$L_PIO_SIZ ; Actual page count
    52   00000000'8F   D0  020C        MOVL    #CTL$GL_IAFLINK,R2         ; Base of image activator context page
              53   52   D0  0213       MOVL    R2,R3                      ; Top (one page)
        08E6'CF   52   D0  0216        MOVL    R2,MISC+IMGDMP$L_IMGCTX_VA
   08E2'CF   0B39'CF   D0  021B        MOVL    NEXT_FILE,MISC+IMGDMP$L_IMGCTX_BLK ; The file block to be used
                 0177   30  0222       BSBW    WRITE_MISC
        08EA'CF   54   D0  0225        MOVL    R4,MISC+IMGDMP$L_IMGCTX_SIZ ; Actual page count
    52   FFFFF800'8F   D0  022A        MOVL    #CTL$A_COMMON-<4*512>,R2 ; Base user writeable context pages
    53   00000200'8F   D0  0231        MOVL    #CTL$AG_CMEDATA+512,R3  ; Top (one page after AME context)
        08F2'CF   52   D0  0238        MOVL    R2,MISC+IMGDMP$L_USRCTX_VA
   08EE'CF   0B39'CF   D0  023D        MOVL    NEXT_FILE,MISC+IMGDMP$L_USRCTX_BLK ; The file block to be used
                 0155   30  0244       BSBW    WRITE_MISC
        08F6'CF   54   D0  0247        MOVL    R4,MISC+IMGDMP$L_USRCTX_SIZ ; Actual page count
                         024C    ;
                         024C    ; Write out the misc data block
                         024C    ;
        0BC8'CF   02   D0  024C        MOVL    #2,DMP_RAB+RAB$L_BKT       ; Misc data block
   0BB2'CF   0200 8F   B0  0251        MOVW    #512,DMP_RAB+RAB$W_RSZ     ; One page
   0BB8'CF   0802'CF   9E  0258        MOVAB   MISC,DMP_RAB+RAB$L_RBF
                         025F          $WRITE  RAB = DMP_RAB              ; Write misc data block
              0B 50   E9  026A         BLBC    R0,EXIT
                         026D    ;
                         026D          $CLOSE FAB = DMP_FAB               ; Close the dump file
                         0278    256 ;
                     05  0278    257 EXIT:    RSB
                         0279    258
```

_S

Ps
--
$P

$G

$O

_L

_S

$C

```
                                 0279  ;
                                 0279  ; This routine processes a range of address space, filling in the map,
                                 0279  ; writing the dump, and skipping unaccessable pages.
                                 0279  ;
                                 0279  ; Inputs
                                 0279  ;       R2 -> page count
                                 0279  ;       R5 -> current address map entry
                                 0279  ;       DMP_RAB+RAB$L_RBF -> starting VA
                                 0279  ;       DMP_RAB+RAB$L_BKT -> starting file block
                                 0279  ; Outputs
                                 0279  ;       R5 -> updated
                                 0279  ;       NEXT_FILE -> next available file block
                                 0279  ;
                                 0279  WRITE_IT:
     0BB2'CF    0200 8F    B0    0279  10$:    MOVW     #512,DMP_RAB+RAB$W_RSZ   ; One page
                                 0280          $WRITE   RAB = DMP_RAB            ; Try to write the page
                 5E 50     E9    028B          BLBC     R0,50$                   ; Error - try next page
       04 A5    0BB8'CF    D0    028E          MOVL     DMP_RAB+RAB$L_RBF,4(R5)  ; Starting VA
                    52     D7    0294          DECL     R2                       ; One less page
  0BB8'CF    00000200 8F   C0    0296          ADDL     #512,DMP_RAB+RAB$L_RBF   ; Next VA
                0BC8'CF    D6    029F          INCL     DMP_RAB+RAB$L_BKT        ; Next file block
                    65     D6    02A3          INCL     (R5)                     ; Another block
       56    0000007F 8F   D0    02A5          MOVL     #127,R6                  ; Start with 127 page write
                 53 52     D0    02AC  20$:    MOVL     R2,R3                    ; Remaining pages
                 56 53     D1    02AF          CMPL     R3,R6                    ; Check against maximum
                    03     1B    02B2          BLEQU    30$
                 53 56     D0    02B4          MOVL     R6,R3                    ; Write the maximum
           54    53 09     78    02B7  30$:    ASHL     #9,R3,R4                 ; Get byte count
            0BB2'CF 54     B0    02BB          MOVW     R4,DMP_RAB+RAB$W_RSZ
                                 02C0          $WRITE   RAB = DMP_RAB            ; Write next page in run
                 08 50     E8    02CB          BLBS     R0,40$                   ; Success - continue the run of pages
                    56     D7    02CE          DECL     R6                       ; Try one page less
                    DA     14    02D0          BGTR     20$                      ; Go try again
                    2C     10    02D2          BSBB     NEXT_MAP_SEGMENT         ; Failure - next segment
                    16     11    02D4          BRB      50$
            0BC8'CF 53     C0    02D6  40$:    ADDL     R3,DMP_RAB+RAB$L_BKT     ; Next file block
            0BB8'CF 54     C0    02DB          ADDL     R4,DMP_RAB+RAB$L_RBF     ; Next VA
                 65 53     C0    02E0          ADDL     R3,(R5)                  ; Keep count
                 52 53     C2    02E3          SUBL     R3,R2                    ; Remaining pages
                    C4     14    02E6          BGTR     20$                      ; Continue
                    16     10    02E8          BSBB     NEXT_MAP_SEGMENT         ; Done with the range - next segment
                    0C     11    02EA          BRB      60$
  0BB8'CF    00000200 8F   C0    02EC  50$:    ADDL     #512,DMP_RAB+RAB$L_RBF
                 81 52     F5    02F5          SOBGTR   R2,10$
     0B39'CF    0BC8'CF    D0    02F8  60$:    MOVL     DMP_RAB+RAB$L_BKT,NEXT_FILE ; Next available file block
                           05    02FF          RSB                               ; Done with the range
                                 0300  ;
                                 0300  ;
                                 0300  ; Set up the next map segment
                                 0300  ; A map block is written if necessary
                                 0300  ;
                                 0300  ; Inputs
                                 0300  ;       R5 -> current map entry
                                 0300  ;       MAP_BLOCK -> file block of current bap blcok
                                 0300  ; Outputs
                                 0300  ;       R5 -> updated
                                 0300  ;       MAP_BLOCK -> updated if map was written
                                 0300    317 ;
                                 0300    318 NEXT_MAP_SEGMENT:
                 55    08    C0  0300    319            ADDL     #8,R5                      ; Get to available space
                 07FA'CF     9F  0303    320            PUSHAB   MAP_END
                 8E    55    D1  0307    321            CMPL     R5,(SP)+                   ; Is this the end
                             12  030A    322            BNEQ     10$                        ; No - use it
                 65    01    CE  030C    323            MNEGL    #1,(R5)                    ; Mark map filled
                 0BB8'CF     DD  030F    324            PUSHL    DMP_RAB+RAB$L_RBF          ; Save VA
                 0BC8'CF     DD  0313    325            PUSHL    DMP_RAB+RAB$L_BKT          ; Save next available block
        0BC8'CF  0B35'CF     D0  0317    326            MOVL     MAP_BLOCK,DMP_RAB+RAB$L_BKT
        0BB2'CF  0200 8F     B0  031E    327            MOVW     #512,DMP_RAB+RAB$W_RSZ     ; One page
        0BB8'CF  0602'CF     9E  0325    328            MOVAB    MAP,DMP_RAB+RAB$L_RBF
                                 032C    329            $WRITE   RAB = DMP_RAB              ; Write out this map block
                    1A 50    E9  0337    330            BLBC     R0,20$
             0B35'CF   8E    D0  033A    331            MOVL     (SP)+,MAP_BLOCK            ; Next block is for map
    0BC8'CF  0B35'CF   01    C1  033F    332            ADDL3    #1,MAP_BLOCK,DMP_RAB+RAB$L_BKT ; Next available block
             0BB8'CF   8E    D0  0347    333            MOVL     (SP)+,DMP_RAB+RAB$L_RBF ; Restore VA
             55  0602'CF     9E  034C    334            MOVAB    MAP,R5
                       65    D4  0351    335 10$:       CLRL     (R5)                       ; No pages yet in map segment
                             05  0353    336            RSB
                                 0354    337 ;
                     FF21    31  0354    338 20$:       BRW      EXIT
```

```
                              0357      ;
                              0357      ; Routines to save misc portions of P1 address space
                              0357      ;
                              0357      ;
                              0357      ; Save the EXEC and kernel stacks
                              0357      ;
                              0357      PRIV_STKS:
                        0000  0357              .WORD   0
    52   00000000'9F    D0    0359              MOVL    @#CTL$AL_STACKLIM,R2     ; Kernel stack low limit
    53   00000000'9F    D0    0360              MOVL    @#CTL$AL_STACK,R3        ; High limit
         0BB6'CF   52   D0    0367              MOVL    R2,MISC+IMGDMP$L_KSTK_VA
    0BB2'CF   0B39'CF   D0    036C              MOVL    NEXT_FILE,MISC+IMGDMP$L_KSTK_BLK ; Next file block
                   27   10    0373              BSBB    WRITE_MISC               ; Write out misc address space
         0BBA'CF   54   D0    0375              MOVL    R4,MISC+IMGDMP$L_KSTK_SIZ
    52   00000004'9F    D0    037A              MOVL    @#CTL$AL_STACKLIM+4,R2   ; Exec stack low limit
    53   00000004'9F    D0    0381              MOVL    @#CTL$AL_STACK+4,R3      ; High limit
         0BC2'CF   52   D0    0388              MOVL    R2,MISC+IMGDMP$L_ESTK_VA
    0BBE'CF   0B39'CF   D0    038D              MOVL    NEXT_FILE,MISC+IMGDMP$L_ESTK_BLK ; Next file block
                   06   10    0394              BSBB    WRITE_MISC               ; Write out misc address space
         0BC6'CF   54   D0    0396              MOVL    R4,MISC+IMGDMP$L_ESTK_SIZ
                        04    039B              RET
                              039C      ;
                              039C      ;
                              039C      ; Write out misc address space
                              039C      ;  Input
                              039C      ;         R2 = low address
                              039C      ;         R3 = high address
                              039C      ;         NEXT_FILE = next available file block
                              039C      ;  Output
                              039C      ;         R4 = page count
                              039C      ;         NEXT_FILE = updated
                              039C      ;
                              039C      WRITE_MISC:
    0BC8'CF   0B39'CF   D0    039C              MOVL    NEXT_FILE,DMP_RAB+RAB$L_BKT ; First block
         0BB8'CF   52   D0    03A3              MOVL    R2,DMP_RAB+RAB$L_RBF     ; Starting VA
              53   52   C2    03A8              SUBL    R2,R3                    ; Byte count
      53   53   F7 8F   78    03AB              ASHL    #-9,R3,R3                ; Pages - 1
                   53   D6    03B0              INCL    R3                       ; Pages
                   54   D4    03B2              CLRL    R4                       ; No pages
 55   53   00000200 8F  C5    03B4      10$:    MULL3   #512,R3,R5               ; Byte count
   0000FE00 8F     55   D1    03BC              CMPL    R5,#<127*512>            ; Is it greater than maximum
                   07   1B    03C3              BLEQU   20$                      ; No
      55   0000FE00 8F  D0    03C5              MOVL    #<127*512>,R5
         0BB2'CF   55   B0    03CC      20$:    MOVW    R5,DMP_RAB+RAB$W_RSZ
                              03D1              $WRITE  RAB = DMP_RAB
                17 50   E9    03DC              BLBC    R0,30$                   ; Done
         0BB8'CF   55   C0    03DF              ADDL    R5,DMP_RAB+RAB$L_RBF     ; New address
      55   55   F7 8F   78    03E4              ASHL    #-9,R5,R5                ; Page count
              54   55   C0    03E9              ADDL    R5,R4                    ; Count pages
         0BC8'CF   55   C0    03EC              ADDL    R5,DMP_RAB+RAB$L_BKT     ; New file block number
              53   55   C2    03F1              SUBL    R5,R3                    ; Pages left
                   BE   1A    03F4              BGTRU   10$                      ; More to do
    0B39'CF   0BC8'CF   D0    03F6      30$:    MOVL    DMP_RAB+RAB$L_BKT,NEXT_FILE ; Next available file block
                        05    03FD              RSB
```

```
                                03FE   395 ; Data
                                03FE   396 ;
                                03FE   397 ;
                                03FE   398 IMG_CHANNEL:                                     ; Channel for the image header
                     00000000   03FE   399          .LONG   0
                                0402   400 ;
                                0402   401 IMGHDRBUF:                                       ; Buffer for image header (first block)
                     00000602   0402   402          .BLKB   512
                                0602   403 ;
                     00000802   0602   404 MAP:     .BLKB   512                             ; Address map
                     000007FA   0802   405 MAP_END = .-8                                    ; Last available map slot
                                0802   406 ;
                     00000A02   0802   407 MISC:    .BLKB   512                             ; Misc data block
                                0A02   408 ;
                                0A02   409 SYS$PROCDMP:
52 50 24 53 59 53 00000A0A'010E0000'   0A02   410          .ASCID  /SYS$PROCDMP/
                  50 4D 44 43 4F   0A10
                                0A15   411 ;
                                0A15   412 PROCDMPNAM:                                      ; Translation of SYS$PROCDMP
                     000000FF   0A15   413          .LONG   255
                     00000602'  0A19   414          .ADDRESS MAP
                                0A1D   415 ;
                    0404 0004   0A1D   416 JPILST: .WORD   4,JPI$_FREPOVA
                     0000084A'  0A21   417          .ADDRESS MISC+IMGDMP$L_FREE_PO
                     00000000   0A25   418          .LONG   0
                    0405 0004   0A29   419          .WORD   4,JPI$_FREP1VA
                     0000084E'  0A2D   420          .ADDRESS MISC+IMGDMP$L_FREE_P1
                     00000000   0A31   421          .LONG   0
                    0301 0004   0A35   422          .WORD   4,JPI$_ASTEN
                     00000856'  0A39   423          .ADDRESS MISC+IMGDMP$L_ASTEN
                     00000000   0A3D   424          .LONG   0
                    0300 0004   0A41   425          .WORD   4,JPI$_ASTACT
                     00000852'  0A45   426          .ADDRESS MISC+IMGDMP$L_ASTACT
                     00000000   0A49   427          .LONG   0
                    0400 0008   0A4D   428          .WORD   8,JPI$_CURPRIV
                     0000085A'  0A51   429          .ADDRESS MISC+IMGDMP$Q_CURPRIV
                     00000000   0A55   430          .LONG   0
                    0317 0004   0A59   431          .WORD   4,JPI$_EFCS
                     00000862'  0A5D   432          .ADDRESS MISC+IMGDMP$L_EFCS
                     00000000   0A61   433          .LONG   0
                    0318 0004   0A65   434          .WORD   4,JPI$_EFCU
                     00000866'  0A69   435          .ADDRESS MISC+IMGDMP$L_EFCU
                     00000000   0A6D   436          .LONG   0
                    030F 0004   0A71   437          .WORD   4,JPI$_BIOCNT
                     0000086A'  0A75   438          .ADDRESS MISC+IMGDMP$L_BIOCNT
                     00000000   0A79   439          .LONG   0
                    0310 0004   0A7D   440          .WORD   4,JPI$_BIOLM
                     0000086E'  0A81   441          .ADDRESS MISC+IMGDMP$L_BIOLM
                     00000000   0A85   442          .LONG   0
                    040C 0004   0A89   443          .WORD   4,JPI$_BUFIO
                     00000872'  0A8D   444          .ADDRESS MISC+IMGDMP$L_BUFIO
                     00000000   0A91   445          .LONG   0
                    0311 0004   0A95   446          .WORD   4,JPI$_BYTCNT
                     00000876'  0A99   447          .ADDRESS MISC+IMGDMP$L_BYTCNT
                     00000000   0A9D   448          .LONG   0
                    031A 0004   0AA1   449          .WORD   4,JPI$_BYTLM
                     0000087A'  0AA5   450          .ADDRESS MISC+IMGDMP$L_BYTLM
```

```
 00000000  0AA9   451           .LONG   0
0312 0004  0AAD   452           .WORD   4,JPI$_DIOCNT
 0000087E' 0AB1   453           .ADDRESS MISC+IMGDMP$L_DIOCNT
 00000000  0AB5   454           .LONG   0
0313 0004  0AB9   455           .WORD   4,JPI$_DIOLM
 00000882' 0ABD   456           .ADDRESS MISC+IMGDMP$L_DIOLM
 00000000  0AC1   457           .LONG   0
040B 0004  0AC5   458           .WORD   4,JPI$_DIRIO
 00000886' 0AC9   459           .ADDRESS MISC+IMGDMP$L_DIRIO
 00000000  0ACD   460           .LONG   0
0314 0004  0AD1   461           .WORD   4,JPI$_FILCNT
 0000088A' 0AD5   462           .ADDRESS MISC+IMGDMP$L_FILCNT
 00000000  0AD9   463           .LONG   0
040F 0004  0ADD   464           .WORD   4,JPI$_FILLM
 0000088E' 0AE1   465           .ADDRESS MISC+IMGDMP$L_FILLM
 00000000  0AE5   466           .LONG   0
0315 0004  0AE9   467           .WORD   4,JPI$_TQCNT
 0000089A' 0AED   468           .ADDRESS MISC+IMGDMP$L_TQCNT
 00000000  0AF1   469           .LONG   0
0410 0004  0AF5   470           .WORD   4,JPI$_TQLM
 0000089E' 0AF9   471           .ADDRESS MISC+IMGDMP$L_TQLM
 00000000  0AFD   472           .LONG   0
030E 0004  0B01   473           .WORD   4,JPI$_ASTCNT
 000008A2' 0B05   474           .ADDRESS MISC+IMGDMP$L_ASTCNT
 00000000  0B09   475           .LONG   0
0409 0004  0B0D   476           .WORD   4,JPI$_ASTLM
 000008A6' 0B11   477           .ADDRESS MISC+IMGDMP$L_ASTLM
 00000000  0B15   478           .LONG   0
031F 0004  0B19   479           .WORD   4,JPI$_ENQCNT
 000008AA' 0B1D   480           .ADDRESS MISC+IMGDMP$L_ENQCNT
 00000000  0B21   481           .LONG   0
0320 0004  0B25   482           .WORD   4,JPI$_ENQLM
 000008AE' 0B29   483           .ADDRESS MISC+IMGDMP$L_ENQLM
 00000000  0B2D   484           .LONG   0
 00000000  0B31   485           .LONG   0
           0B35   486 ;
           0B35   487 MAP_BLOCK:                                   ; File block for current address map
 00000000  0B35   488           .LONG   0
           0B39   489 ;
           0B39   490 NEXT_FILE:                                   ; Next available file block
 00000000  0B39   491           .LONG   0
           0B3D   492 ;
           0B3D   493           .ALIGN  LONG
           0B40   494 DMP_FAB: $FAB   FAC = <BIO,PUT>,-
           0B40   495                   DNM = <.DMP>
           0B90   496 DMP_RAB: $RAB   FAB = DMP_FAB,-
           0B90   497                   RSZ = 512,-
           0B90   498                   ROP = BIO
           0BD4   499 ;
           0BD4   500           .END
```

```
$$.TAB                          = 00000B90 R       01      IMGDMP$L_ESTK_SIZ               = 000000C4
$$.TABEND                       = 00000BD4 R       01      IMGDMP$L_ESTK_VA                = 000000C0
$$.TMP                          = 00000800                 IMGDMP$L_FILCNT                 = 00000088
$$.TMP1                         = 00000001                 IMGDMP$L_FILLM                  = 0000008C
$$.TMP2                         = 000000CF                 IMGDMP$L_FIRST_MAP              = 0000003C
$$.TMPX                         = 00000000 R       03      IMGDMP$L_FP                     = 00000038
$$.TMPX1                        = 00000004                 IMGDMP$L_FREE_P0                = 00000048
$$T1                            = 00000001                 IMGDMP$L_FREE_P1                = 0000004C
BEGIN_CODE                        0000000C R       01      IMGDMP$L_IMGCTX_BLK             = 000000E0
BEGIN_DUMP                        00000000 R       01      IMGDMP$L_IMGCTX_SIZ             = 000000E8
CTL$AG_CMEDATA                    ********     X   01      IMGDMP$L_IMGCTX_VA              = 000000E4
CTL$AL_STACK                      ********     X   01      IMGDMP$L_KSTK_BLK               = 000000B0
CTL$AL_STACKLIM                   ********     X   01      IMGDMP$L_KSTK_SIZ               = 000000B8
CTL$A_COMMON                      ********     X   01      IMGDMP$L_KSTK_VA                = 000000B4
CTL$GL_IAFLINK                    ********     X   01      IMGDMP$L_PIO_BLK                = 000000D4
CTL$GL_VECTORS                    ********     X   01      IMGDMP$L_PIO_SIZ                = 000000DC
DMP_FAB                           00000B40 R       01      IMGDMP$L_PIO_VA                 = 000000D8
DMP_RAB                           00000B90 R       01      IMGDMP$L_PRCCNT                 = 00000090
EXIT                              00000278 R       01      IMGDMP$L_PRCLM                  = 00000094
FAB$B_DNS                       = 00000035                 IMGDMP$L_R0                     = 00000000
FAB$C_BID                       = 00000003                 IMGDMP$L_R1                     = 00000004
FAB$C_BLN                       = 00000050                 IMGDMP$L_R10                    = 00000028
FAB$C_SEQ                       = 00000000                 IMGDMP$L_R11                    = 0000002C
FAB$C_VAR                       = 00000002                 IMGDMP$L_R2                     = 00000008
FAB$L_ALQ                       = 00000010                 IMGDMP$L_R3                     = 0000000C
FAB$L_DNA                       = 00000030                 IMGDMP$L_R4                     = 00000010
FAB$L_FOP                       = 00000004                 IMGDMP$L_R5                     = 00000014
FAB$V_BIO                       = 00000005                 IMGDMP$L_R6                     = 00000018
FAB$V_CHAN_MODE                 = 00000002                 IMGDMP$L_R7                     = 0000001C
FAB$V_FILE_MODE                 = 00000004                 IMGDMP$L_R8                     = 00000020
FAB$V_LNM_MODE                  = 00000000                 IMGDMP$L_R9                     = 00000024
FAB$V_PUT                       = 00000000                 IMGDMP$L_SP                     = 00000030
FAB$W_GBC                       = 00000048                 IMGDMP$L_TQCNT                  = 00000098
IFD$W_CHAN                      = 00000008                 IMGDMP$L_TQLM                   = 0000009C
IHD$W_IMGIDOFF                  = 00000006                 IMGDMP$L_USRCTX_BLK             = 000000EC
IHI$T_IMGNAM                    = 00000000                 IMGDMP$L_USRCTX_SIZ             = 000000F4
IMGDMP                          = 00000000                 IMGDMP$L_USRCTX_VA              = 000000F0
IMGDMP$C_LENGTH                 = 000000F8                 IMGDMP$L_USRSTK                 = 00000040
IMGDMP$C_VERSION                = 00000003                 IMGDMP$L_VECPAG_BLK             = 000000C8
IMGDMP$L_AP                     = 00000034                 IMGDMP$L_VECPAG_SIZ             = 000000D0
IMGDMP$L_ASTACT                 = 00000050                 IMGDMP$L_VECPAG_VA              = 000000CC
IMGDMP$L_ASTCNT                 = 000000A0                 IMGDMP$L_VERSION                = 00000044
IMGDMP$L_ASTEN                  = 00000054                 IMGDMP$Q_CURPRIV                = 00000058
IMGDMP$L_ASTLM                  = 000000A4                 IMGDMP$S_CURPRIV                = 00000008
IMGDMP$L_BIOCNT                 = 00000068                 IMGDMP$S_IMGDMP                 = 000000F8
IMGDMP$L_BIOLM                  = 0000006C                 IMGHDRBUF                         00000402 R       01
IMGDMP$L_BUFIO                  = 00000070                 IMG_CHANNEL                       000003FE R       01
IMGDMP$L_BYTCNT                 = 00000074                 IO$_READVBLK                      ********     X   01
IMGDMP$L_BYTLM                  = 00000078                 JPI$_ASTACT                     = 00000300
IMGDMP$L_DIOCNT                 = 0000007C                 JPI$_ASTCNT                     = 0000030E
IMGDMP$L_DIOLM                  = 00000080                 JPI$_ASTEN                      = 00000301
IMGDMP$L_DIRIO                  = 00000084                 JPI$_ASTLM                      = 00000409
IMGDMP$L_EFCS                   = 00000060                 JPI$_BIOCNT                     = 0000030F
IMGDMP$L_EFCU                   = 00000064                 JPI$_BIOLM                      = 00000310
IMGDMP$L_ENQCNT                 = 000000A8                 JPI$_BUFIO                      = 0000040C
IMGDMP$L_ENQLM                  = 000000AC                 JPI$_BYTCNT                     = 00000311
IMGDMP$L_ESTK_BLK               = 000000BC                 JPI$_BYTLM                      = 0000031A
```

```
JPI$_CURPRIV                    = 00000400
JPI$_DIOCNT                     = 00000312
JPI$_DIOLM                      = 00000313
JPI$_DIRIO                      = 0000040B
JPI$_EFCS                       = 00000317
JPI$_EFCU                       = 00000318
JPI$_ENQCNT                     = 0000031F
JPI$_ENQLM                      = 00000320
JPI$_FILCNT                     = 00000314
JPI$_FILLM                      = 0000040F
JPI$_FREP0VA                    = 00000404
JPI$_FREP1VA                    = 00000405
JPI$_TQCNT                      = 00000315
JPI$_TQLM                       = 00000410
JPILST                            00000A1D R       01
MAP                               00000602 R       01
MAP_BLOCK                         00000B35 R       01
MAP_END                         = 000007FA R       01
MISC                              00000802 R       01
MMG$IMGHDRBUF                     ********   X     01
NEXT_FILE                         00000B39 R       01
NEXT_MAP_SEGMENT                  00000300 R       01
PIO$A_TRACE                       ********   X     01
PRIV_STKS                         00000357 R       01
PROCDMPNAM                        00000A15 R       01
PSL$C_USER                      = 00000003
PSL$S_CURMOD                    = 00000002
PSL$V_CURMOD                    = 00000018
RAB$B_RAC                       = 0000001E
RAB$C_BID                       = 00000001
RAB$C_BLN                       = 00000044
RAB$C_SEQ                       = 00000000
RAB$L_CTX                       = 00000018
RAB$L_ROP                       = 00000004
RAB$V_BIO                       = 0000000B
SF$L_SAVE_FP                    = 0000000C
SF$L_SAVE_REGS                  = 00000014
SS$_NOTRAN                        ********   X     01
SYS$ADJSTK                        ********  GX     01
SYS$CLOSE                         ********  GX     01
SYS$CMEXEC                        ********  GX     01
SYS$CONNECT                       ********  GX     01
SYS$CREATE                        ********  GX     01
SYS$GETJPI                        ********  GX     01
SYS$PROCDMP                       00000A02 R       01
SYS$QIOW                          ********  GX     01
SYS$TRNLOG                        ********  GX     01
SYS$WRITE                         ********  GX     01
WRITE_IT                          00000279 R       01
WRITE_MISC                        0000039C 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
$AAIMGDMP                      00000BD4  (  3028.)  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                        00000004  (     4.)  03 (  3.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD     WRT NOVEC BYTE

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

Phase                      Page faults    CPU Time        Elapsed Time
-----                      -----------    --------        ------------
Initialization                    29      00:00:00.10     00:00:00.75
Command processing               155      00:00:00.77     00:00:03.10
Pass 1                           301      00:00:09.19     00:00:17.54
Symbol table sort                  0      00:00:00.87     00:00:00.96
Pass 2                           102      00:00:01.98     00:00:04.18
Symbol table output               20      00:00:00.16     00:00:00.82
Psect synopsis output              2      00:00:00.03     00:00:00.03
Cross-reference output             0      00:00:00.00     00:00:00.00
Assembler run totals             612      00:00:13.10     00:00:27.38

The working set limit was 1500 pages.
46825 bytes (92 pages) of virtual memory were used to buffer the intermediate code.
There were 40 pages of symbol table space allocated to hold 676 non-local and 17 local symbols.
500 source lines were read in Pass 1, producing 20 object records in Pass 2.
40 pages of virtual memory were used to define 32 macros.

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

Macro library name                                Macros defined
------------------                                --------------
_$255$DUA28:[IMGDMP.OBJ]IMGDMPLIB.MLB;1                  1
_$255$DUA28:[SYS.OBJ]LIB.MLB;1                           3
_$255$DUA28:[SYSLIB]STARLET.MLB;2                       25
TOTALS (all libraries)                                  29

997 GETS were required to define 29 macros.

There were no errors, warnings or information messages.

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

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

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY