M 11

```
DDDDDDDDDDDD        CCCCCCCCCCCC  LLL
DDDDDDDDDDDD        CCCCCCCCCCCC  LLL
DDDDDDDDDDDD        CCCCCCCCCCCC  LLL
DDD         DDD  CCC              LLL
DDD         DDD  CCC              LLL
DDD         DDD  CCC              LLL
DDD         DDD  CCC              LLL
DDD         DDD  CCC              LLL
DDD         DDD  CCC              LLL
DDD         DDD  CCC              LLL
DDD         DDD  CCC              LLL
DDD         DDD  CCC              LLL
DDD         DDD  CCC              LLL
DDD         DDD  CCC              LLL
DDD         DDD  CCC              LLL
DDD         DDD  CCC              LLL
DDD         DDD  CCC              LLL
DDD         DDD  CCC              LLL
DDDDDDDDDDDD        CCCCCCCCCCCC  LLLLLLLLLLLLLLL
DDDDDDDDDDDD        CCCCCCCCCCCC  LLLLLLLLLLLLLLL
DDDDDDDDDDDD        CCCCCCCCCCCC  LLLLLLLLLLLLLLL
```

```
**FILE**ID**IMAGEXECT                                   N 12




IIIIII      MM      MM    AAAAAA      GGGGGGGG  EEEEEEEEEE  XX      XX  EEEEEEEEEE    CCCCCCCC  TTTTTTTTTT
IIIIII      MM      MM    AAAAAA      GGGGGGGG  EEEEEEEEEE  XX      XX  EEEEEEEEEE    CCCCCCCC  TTTTTTTTTT
  II        MMMM  MMMM  AA      AA  GG          EE          XX      XX  EE          CC              TT
  II        MMMM  MMMM  AA      AA  GG          EE          XX      XX  EE          CC              TT
  II        MM  MM  MM  AA      AA  GG          EE            XX  XX    EE          CC              TT
  II        MM  MM  MM  AA      AA  GG          EE            XX  XX    EE          CC              TT
  II        MM      MM  AA      AA  GG          EEEEEEEE        XX      EEEEEEEE    CC              TT
  II        MM      MM  AA      AA  GG          EEEEEEEE        XX      EEEEEEEE    CC              TT
  II        MM      MM  AAAAAAAAAA  GG  GGGGGG  EE            XX  XX    EE          CC              TT
  II        MM      MM  AAAAAAAAAA  GG  GGGGGG  EE            XX  XX    EE          CC              TT
  II        MM      MM  AA      AA  GG      GG  EE          XX      XX  EE          CC              TT    ....
  II        MM      MM  AA      AA  GG      GG  EE          XX      XX  EE          CC              TT    ....
IIIIII      MM      MM  AA      AA    GGGGGG    EEEEEEEEEE  XX      XX  EEEEEEEEEE    CCCCCCCC      TT    ....
IIIIII      MM      MM  AA      AA    GGGGGG    EEEEEEEEEE  XX      XX  EEEEEEEEEE    CCCCCCCC      TT    ....

LL          IIIIII        SSSSSSSS
LL          IIIIII        SSSSSSSS
LL            II        SS
LL            II        SS
LL            II        SS
LL            II        SS
LL            II          SSSSSS
LL            II          SSSSSS
LL            II                  SS
LL            II                  SS
LL            II                  SS
LL            II                  SS
LLLLLLLLLL  IIIIII      SSSSSSSS
LLLLLLLLLL  IIIIII      SSSSSSSS
```

Table of contents

```
0000            .TITLE  IMAGEXECT - IMAGE EXECUTION
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   *
0C00    ;*  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    ; IMAGE EXECUTION BY EXTERNAL IMAGE COMMAND OR RUN COMMAND
0000    ;
0000    ; D. N. CUTLER 4-APR-77
0000    ;
0000    ; MODIFIED BY:
0000    ;
0000    ;       V03-008 HWS0090         Harold Schultz          22-Jul-1984
0000    ;               Just delete the process if an image activation error
0000    ;               is encountered while attempting to logout via executing
0000    ;               the LOGINOUT image.
0000    ;
0000    ;       V03-007 HWS0075         Harold Schultz          28-Jun-1984
0000    ;               Move initialization of command interpreter command
0000    ;               pointers to the beginning of the command interpreter
0000    ;               exit handler in order to insure this initialization in
0000    ;               the event a CNTL-Y is pending
0000    ;
0000    ;       V03-006 HWS0037         Harold Schultz          21-Mar-1984
0000    ;               Use PRC_V_IRUNDWN flag to indicate whether or not an
0000    ;               image has been rundown by DCL
0000    ;
0000    ;       V03-005 MSH0001         Michael S. Harvey       12-Jan-1984
0000    ;               Defer reporting alternate success message until image
0000    ;               fixups are completed. This is necessary because message
0000    ;               output can clobber fixup context stored in P1.
0000    ;
0000    ;       V03-004 PCG0008         Peter George            27-May-1983
0000    ;               Change DCL$RESTORE_OUTPUT calling format.
0000    ;
0000    ;       V03-003 PCG0007         Peter George            13-Jan-1983
```

```
                                  0000     58 ;                        Set up exit handler block in INITIAL, not here.
                                  0000     59 ;                        Close redefined SYS$OUTPUT before logging out.
                                  0000     60 ;                        Add DCL$CLOSE_PPFS.
                                  0000     61 ;
                                  0000     62 ;               V03-002 PCG0006         Peter George            22-Oct-1982
                                  0000     63 ;                        Move DCL$EOJ to COMMAND.
                                  0000     64 ;
                                  0000     65 ;               V03-001 PCG0005         Peter George            09-Jun-1982
                                  0000     66 ;                        Allow MCR command/qualifier without a delimiting blank.
                                  0000     67 ;---
```

```
                                    0000     69 ;
                                    0000     70 ; MACRO LIBRARY CALLS
                                    0000     71 ;
                                    0000     72
                                    0000     73         $PPDDEF                                 ;PROCESS PERMANENT DATA AREA
                                    0000     74         PRCDEF                                  ;DEFINE PROCESS WORK AREA
                                    0000     75         WRKDEF                                  ;DEFINE COMMAND WORK AREA
                                    0000     76         PTRDEF                                  ;RESULT PARSE DESCRIPTORS
                                    0000     77         $CLIMSGDEF                              ;DEFINE ERROR/STATUS VALUES
                                    0000     78         $CHFDEF                                 ;DEFINE CONDITION ARGLIST OFFSETS
                                    0000     79         $FABDEF                                 ;DEFINE FAB OFFSETS
                                    0000     80         $IFDDEF                                 ;IMAGE FILE DESCRIPTOR DEFINITIONS
                                    0000     81         $IHDDEF                                 ;IMAGE HEADER DESCRIPTOR DEFINTITIONS
                                    0000     82         $NAMDEF                                 ;DEFINE NAME BLOCK OFFSETS
                                    0000     83         $PSLDEF                                 ;DEFINE PROCESSOR STATUS FIELDS
                                    0000     84         $RABDEF                                 ;DEFINE RAB OFFSETS
                                    0000     85         $STSDEF                                 ;DEFINE STATUS LONG WORD VALUES
                                    0000     86         $CLIDEF                                 ;DEFINE IMAGE ARGUMENT LIST FORMAT
                                    0000     87
                                    0000     88 ;
                                    0000     89 ; LOCAL DATA
                                    0000     90 ;
                                    0000     91
                                00000000     92         .PSECT  DCL$ZCODE,BYTE,RD,NOWRT
                                    0000     93 EXTDEFAULT:                                     ;EXTERNAL COMMAND IMAGE DEFAULT STRING
3A 4D 45 54 53 59 53 24 53 59 53 00' 0000    94         .ASCIC  'SYS$SYSTEM:.EXE'
                         45 58 45 2E 000C
                                  0F 0000
                                    0010     95 RUNDEFAULT:                                     ;RUN COMMAND IMAGE DEFAULT STRING
                      45 58 45 2E 00' 0010   96         .ASCIC  '.EXE'
                                  04 0010
                                    0015     97 LOGOUTIMG:                                      ;FILENAME TEXT STRING FOR LOGOUT IMAGE
       54 55 4F 4E 49 47 4F 4C 00'  0015     98         .ASCIC  'LOGINOUT'
                                  08 0015
                                    001E     99 MCRIMG:                                         ;IMAGE FOR MCR COMMAND
                         58 53 52 00' 001E  100         .ASCIC  'RSX'
                                  03 001E
                                    0022    101 SYSPRINT:                                       ;DEFAULT QUEUE LOGICAL NAME
       54 4E 49 52 50 24 53 59 53   0022    102         .ASCII  'SYS$PRINT'
                            00000009 002B   103 SYSPRTSIZ = . - SYSPRINT                        ;SIZE OF THAT STRING
                                    002B    104
                                    002B    105 ;
                                    002B    106 ; DEFINE IMAGE NAMES
                                    002B    107 ;
                                    002B    108         .MACRO  INTIMAGE NAME
                                    002B    109 IMG_K_'NAME = $INTIMAGE$
                                    002B    110 $INTIMAGE$ = $INTIMAGE$ + 1
                                    002B    111         .ENDM
                                    002B    112
                                    002B    113         INTIMAGES
```

```
                  002B                  .SBTTL  RUN IMAGE
                  002B          ;+
                  002B          ; DCL$RUN - RUN IMAGE
                  002B          ;
                  002B          ; THIS ROUTINE IS CALLED AS AN INTERNAL COMMAND TO EXECUTE THE RUN COMMAND.
                  002B          ;
                  002B          ; INPUTS:
                  002B          ;
                  002B          ;       R8 = ADDRESS OF SCRATCH BUFFER DESCRIPTOR.
                  002B          ;       R9 = ADDRESS OF SCRATCH STACK.
                  002B          ;       R10 = BASE ADDRESS OF COMMAND WORK AREA.
                  002B          ;       R11 = BASE ADDRESS OF PROCESS WORK AREA.
                  002B          ;
                  002B          ; OUTPUTS:
                  002B          ;
                  002B          ;       THE SPECIFED IMAGE IS INITIATED.
                  002B          ;-
                  002B
                  002B          DCL$RUN::                                       ;RUN IMAGE
0600 8F     AA    002B                  BICW    #<PRC_M_DBGQUAL!PRC_M_DBGTRUE>,- ;CLEAR DEBUG QUALIFIER SEEN,
  68 AB           002F                  PRC_W_FLAGS(R11)                ;AND DEBUG QUALIFIER TRUE FLAGS.
     7E     7C    0031                  CLRQ    -(SP)                           ;MAKE A SCRATCH BUFFER
   FFCA'    30    0033          10$:    BSBW    DCL$GETDVAL                     ;GET NEXT DESCRIPTOR VALUES
04   55     91    0036                  CMPB    R5,#PTR_K_ENDLINE               ;IS THIS THE END OF THE PARAMETERS?
     1A     13    0039                  BEQL    30$                             ;BR IF YES
03   55     91    003B                  CMPB    R5,#PTR_K_PARAMETR              ;IS THIS THE FILE TO RUN?
     10     13    003E                  BEQL    20$                             ;BR IF YES
0600 8F     A8    0040                  BISW    #<PRC_M_DBGQUAL!PRC_M_DBGTRUE>,- ;SET DEBUG QAULIFIER SEEN
  68 AB           0044                  PRC_W_FLAGS(R11)                ;AND ASSUME QUALIFIER TRUE
  EA 53     E9    0046                  BLBC    R3,10$                          ;BR IF NOT NEQATED
                  0049                  CLRBIT  PRC_V_DBGTRUE,PRC_W_FLAGS(R11) ;SET STATE TO FALSE
     E3     11    004E                  BRB     10$                             ;
6E   51     7D    0050          20$:    MOVQ    R1,(SP)                         ;SAVE FILE SPECIFICATION DESCRIPTOR
     DE     11    0053                  BRB     10$                             ;LOOK FOR MORE
     06     BA    0055          30$:    POPR    #^M<R1,R2>                      ;GET FILE SPEC OF FILE TO RUN
   00A0     31    0057                  BRW     SETRUNDEF                       ;SET RUN DEFAULT AND GO RUN IT
```

```
                      005A         .SBTTL  LOGOUT PROCESS
                      005A  ;+
                      005A  ; DCL$LOGOUT - LOGOUT THE PROCESS
                      005A  ;
                      005A  ; THIS ROUTINE IS CALLED AS AN INTERNAL COMMAND TO EXECUTE THE LOGOUT
                      005A  ; COMMAND OR TO EFFECT A LOGOUT WHEN END OF FILE OCCURS FOR A DETACED JOB.
                      005A  ; ALSO, MAY BE ENTERED IF A HANGUP OCCURS ON A PROCESS WITH A DIAL UP INPUT.
                      005A  ;
                      005A  ; INPUTS:
                      005A  ;
                      005A  ;          R8 = ADDRESS OF SCRATCH BUFFER DESCRIPTOR.
                      005A  ;          R9 = ADDRESS OF SCRATCH STACK.
                      005A  ;          R10 = BASE ADDRESS OF COMMAND WORK AREA.
                      005A  ;          R11 = BASE ADDRESS OF PROCESS WORK AREA.
                      005A  ;
                      005A  ; OUTPUTS:
                      005A  ;
                      005A  ;          THE LOGOUT IMAGE IS INITIATED.
                      005A  ;-
                      005A
                      005A  DCL$LOGOUT::
                      005A  ;
                      005A  ; IF SILENT LOGOUT SPECIFIED, THEN SKIP LOGOUT AND BRANCH TO THERE.
                      005A  ;
 03 68 AB   08   E1   005A         BBC     #PRC_V_AUTOLOGO,PRC_W_FLAGS(R11),10$ ; BRANCH IF FLAG NOT SET
          FF9E'  31   005F         BRW     SILENT_LOGOUT
                      0062
                      0062  ;
                      0062  ; RUN DOWN ACTIVE PROCEDURES AND IMAGES.  CLOSE OPEN PPF FILES.
                      0062  ;
             0F  10   0062  10$:   BSBB    DCL$CLOSE_PPFS
                      0064
                      0064  ;
                      0064  ; ACTIVATE LOGOUT IMAGE
                      0064  ;
     00B0 CB     D0   0064         MOVL    PRC_L_LSTSTATUS(R11),-   ;PASS FINAL STATUS TO LOGOUT
  00000018'GF         0068                 G^CTL$AG_CLIDATA+PPD$L_LSTSTATUS
   52   A5 AF    9E   006D         MOVAB   LOGOUTIMG,R2             ;START OF THE COUNTED STRING
          42     11   0071         BRB     SETIMGNAME               ;GO SET LENGTH, DEFAULT, & EXECUTE IT
```

```
                      0073          .SBTTL  PREPARE THE PROCESS FOR LOGOUT
                      0073      ;+
                      0073      ; DCL$CLOSE_PPFS - PREPARE THE PROCESS FOR LOGOUT
                      0073      ;
                      0073      ; THIS ROUTINE IS CALLED TO PREPARE THE PROCESS FOR LOGOUT.  ACTIVE COMMAND
                      0073      ; PROCEDURES AND IMAGES ARE RUN DOWN.  OPEN PPF FILES ARE CLOSED.
                      0073      ;
                      0073      ; INPUTS:
                      0073      ;
                      0073      ;       R11 = BASE ADDRESS OF PROCESS WORK AREA.
                      0073      ;
                      0073      ; OUTPUTS:
                      0073      ;
                      0073      ;       NONE
                      0073      ;-
                      0073
                      0073      DCL$CLOSE_PPFS::
                      0073      ;
                      0073      ; RUN DOWN ANY PROCEDURES OR IMAGES STILL ACTIVE
                      0073      ;
         5C AB   D5   0073      10$:    TSTL    PRC_L_INDEPTH(R11)          ;INDIRECT LEVEL ZERO?
            05   13   0076              BEQL    20$                         ;IF EQL YES
         FF85'   30   0078              BSBW    DCL$UNSTACK                 ;UNSTACK INDIRECT LEVEL
            F6   11   007B              BRB     10$                         ;
                      007D
                      007D      ;
                      007D      ; CLOSE ALL PROCESS PERMANENT FILES STILL OPEN
                      007D      ;
    59   1C AB   D0   007D      20$:    MOVL    PRC_L_INDFAB(R11),R9        ;GET ADDRESS OF INDIRECT FAB
    52   70 AB   9E   0081              MOVAB   PRC_L_PPFLIST(R11),R2       ;GET ADDRESS OF FILE DESCRIPTOR LISTHEAD
       52   62   D0   0085      30$:    MOVL    (R2),R2                     ;GET ADDRESS OF NEXT FILE DESCRIPTOR
            10   13   0088              BEQL    40$                         ;IF EQL END OF LIST
         1C A2   B0   008A              MOVW    RAB$L_CTX+4(R2),-           ;INSERT INTERNAL FILE INDEX
         02 A9        008D                      FAB$W_IFI(R9)               ;
                      008F              $CLOSE  FAB=(R9)                    ;CLOSE FILE
            EB   11   0098              BRB     30$                         ;
                      009A
                      009A      ;
                      009A      ; CLOSE POSSIBLY REDEFINED SYS$OUTPUT FILE
                      009A      ;
  52   0114 CB   9E   009A      40$:    MOVAB   PRC_W_OUTIFI(R11),R2        ;GET ADDRESS OF PROCESS PERMANENT OUTPUT FIL
  58   00BC CB   D0   009F              MOVL    PRC_L_IDFLNK(R11),R8        ;GET ADDRESS OF CURRENT IDF FRAME
         FF59'   30   00A4              BSBW    DCL$RESTORE_OUTPUT          ;CLOSE TEMPORARY OUTPUT FILE
                 05   00A7              RSB
```

```
                          00A8          .SBTTL  MCR COMMAND
                          00A8  ;+
                          00A8  ; DCL$MCR - EXECUTE THE MCR COMMAND
                          00A8  ;
                          00A8  ; THIS ROUTINE IS CALLED AS AN INTERNAL COMMAND TO EXECUTE THE MCR COMMAND
                          00A8  ;
                          00A8  ; INPUTS:
                          00A8  ;
                          00A8  ;       R8 = ADDRESS OF SCRATCH BUFFER DESCRIPTOR.
                          00A8  ;       R9 = ADDRESS OF SCRATCH STACK.
                          00A8  ;       R10 = BASE ADDRESS OF COMMAND WORK AREA.
                          00A8  ;       R11 = BASE ADDRESS OF PROCESS WORK AREA.
                          00A8  ;
                          00A8  ; OUTPUTS:
                          00A8  ;
                          00A8  ;       THE FIRST TOKEN ON THE COMMAND LINE IS USED TO ACTIVATE AN IMAGE.
                          00A8  ;       IF THE COMMAND LINE IS NULL, THE MCR IMAGE IS ACTIVATED.
                          00A8  ;-
                          00A8
                          00A8          .ENABL  LSB
                          00A8  DCL$MCR::                                ;EXECUTE THE MCR COMMAND
           FF55'   30     00A8          BSBW    DCL$GETDVAL              ;GET DECSRIPTOR VALUE
        54    51   D0     00AB          MOVL    R1,R4                    ;COPY LENGTH OF STRING
              0A   12     00AE          BNEQ    10$                      ;BR IF IMAGE NAME SUPPLIED
   52   FF6A  CF   9E     00B0          MOVAB   MCRIMG,R2                ;SET ADDRESS OF IMAGE NAME
                          00B5  SETIMGNAME:                              ;SET IMAGE NAME, DEFAULT, & EXECUTE IT
        51    82   9A     00B5          MOVZBL  (R2)+,R1                 ;SET IMAGE NAME LENGTH
              2F   11     00B8          BRB     SETIMGDEF                ;GO SET DEFAULT AND EXECUTE IT
        53    51   D0     00BA  10$:    MOVL    R1,R3                    ;SAVE LENGTH OF STRING
   62   53    20   3A     00BD          LOCC    #^A/ /,R3,(R2)           ;FIND SPACE DELIMITER
        55    50   7D     00C1          MOVQ    R0,R5                    ;SAVE POSITION INDICATORS
   62   53    2F   3A     00C4          LOCC    #^A*/*,R3,(R2)           ;FIND SLASH DELIMITER
        56    51   D1     00C8          CMPL    R1,R6                    ;WHICH CAME FIRST?
              03   19     00CB          BLSS    15$                      ;USE POSITON OF SLASH
        50    55   7D     00CD          MOVQ    R5,R0                    ;USE POSITION OF SPACE
        54    50   C2     00D0  15$:    SUBL    R0,R4                    ;FIND LENGTH OF TOKEN
62   54    5B 8F   3A     00D3          LOCC    #^A/[/,R4,(R2)           ;LOOK FOR A DIRECTORY
              06   12     00D8          BNEQ    20$                      ;BR IF DIRECTORY IN THE SPEC
   62   54    3C   3A     00DA          LOCC    #^A/</,R4,(R2)           ;TRY OTHER SYNTAX
              06   13     00DE          BEQL    30$                      ;BR IF NO DIRECTORY FOUND
00 00AF CB    02   E2     00E0  20$:    BBSS    #PRC_V_RUNDEF,PRC_B_FLAGS2(R11),30$ ;USE RUN DEFAULT
        51    54   D0     00E6  30$:    MOVL    R4,RT                    ;GET BYTE COUNT OF IMAGE TO RUN
                          00E9          .DSABL  LSB
```

```
                    00E9          .SBTTL  EXTERNAL COMMAND EXECUTION
                    00E9     ;+
                    00E9     ; DCL$EXTIMAGE - EXTERNAL COMMAND EXECUTION
                    00E9     ;
                    00E9     ; THIS ROUTINE IS CALLED AS AN INTERNAL COMMAND TO INITIATE EXECUTION OF AN
                    00E9     ; EXTERNAL IMAGE.
                    00E9     ;
                    00E9     ; INPUTS:
                    00E9     ;
                    00E9     ;       R1 = LENGTH OF IMAGE FILE SPECIFICATION.
                    00E9     ;       R2 = ADDRESS OF IMAGE FILE SPECIFICATION.
                    00E9     ;       R8 = ADDRESS OF SCRATCH BUFFER DESCRIPTOR.
                    00E9     ;       R9 = ADDRESS OF SCRATCH STACK.
                    00E9     ;       R10 = BASE ADDRESS OF COMMAND WORK AREA.
                    00E9     ;       R11 = BASE ADDRESS OF PROCESS WORK AREA.
                    00E9     ;
                    00E9     ; OUTPUTS:
                    00E9     ;
                    00E9     ;       THE SPECIFED IMAGE IS INITIATED.
                    00E9     ;-
                    00E9
                    00E9     DCL$EXTIMAGE::                               ;EXTERNAL COMMAND EXECUTION
                    00E9     SETIMGDEF:                                   ;SET CORRECT DEFAULT FOR IMAGE
     0600 8F    AA  00E9            BICW    #<PRC_M_DBGQUAL!PRC_M_DBGTRUE>,- ;CLEAR DEBUG QUALIFIER SEEN,
       68 AB        00ED                    PRC_W_FLAGS(R11)             ;AND DEBUG QUALIFIER TRUE FLAGS.
  53  FF0D CF   9E  00EF            MOVAB   EXTDEFAULT,R3                ;SET IMAGE DEFAULT FOR EXTERNAL IMAGE
05 00AF CB      02  E5  00F4        BBCC    #PRC_V_RUNDEF,PRC_B_FLAGS2(R11),FORCEXIT ;CORRECT DEFAULT?
                    00FA     SETRUNDEF:                                   ;SET RUN DEFAULT FOR IMAGE
  53  FF12 CF   9E  00FA            MOVAB   RUNDEFAULT,R3                ;SET IMAGE DEFAULT FOR RUN'S
                    00FF     ;
                    00FF     ; FORCE IMAGE EXIT IF ONE IS ACTIVE
                    00FF     ;
                    00FF     ; INPUTS:
                    00FF     ;
                    00FF     ;       R1 = LENGTH OF IMAGE FILE SPECIFICATION
                    00FF     ;       R2 = ADDRESS OF IMAGE FILE SPECIFICATION
                    00FF     ;       R3 = ADDRESS OF COUNTED ASCII STRING FOR DEFAULT FILE NAME
                    00FF     ;
                    00FF
                    00FF     FORCEXIT:                                    ;FORCE IMAGE EXIT
      023F      30  00FF            BSBW    DCL$FORCEXIT                 ;
```

```
                              0102   323            .SBTTL  INITIATE IMAGE
                              0102   324  ;
                              0102   325  ; INITIATE AN IMAGE BY ENTERING USER MODE AND THEN SETTING UP THE INITIAL CALL
                              0102   326  ; FRAMES AND ESTABLISHING A CATCH-ALL CONDITION HANDLER.
                              0102   327  ;
                              0102   328  ; INPUTS:
                              0102   329  ;
                              0102   330  ;       R1 = LENGTH OF IMAGE FILE SPECIFICATION
                              0102   331  ;       R2 = ADDRESS OF IMAGE FILE SPECIFICATION
                              0102   332  ;       R3 = ADDRESS OF COUNTED ASCII STRING FOR DEFAULT FILE NAME
                              0102   333  ;
                              0102   334
                              0102   335        .ENABL  LSB
                              0102   336 INITIATE:                                   ;INITIATE IMAGE
          14 68 AB   03   E2  0102   337        BBSS    #PRC_V_EXIT,PRC_W_FLAGS(R11),10$ ;IF SET, EXIT HANDLER ESTABLISHED
                     51   DD  0107   338        PUSHL   R1                          ;REMEMBER LENGTH OF IMAGE FILE SPEC
                              0109   339        $DCLEXH_S PRC_L_EXTBLK(R11)         ;DECLARE EXIT HANDLER
                51   8E   D0  0114   340        MOVL    (SP)+,R1                    ;RESTORE LENGTH OF IMAGE FILE SPEC
                  01 50   E8  0117   341        BLBS    R0,10$                      ;BRANCH IF SUCCESSFUL
                          05  011A   342        RSB                                 ;RETURN IF ERROR
                              011B   343
                              011B   344  ;
                              011B   345  ; SET UP IMAGE ADDRESS SPACE
                              011B   346  ;
                              011B   347
       55   00000000'GF   DE  011B   348 10$:   MOVAL   G^MMG$IMGHDRBUF,R5          ;ADDRESS TO RETURN IMAGE HEADER INFO
                7E   51   7D  0122   349        MOVQ    R1,-(SP)                    ;PUT IMAGE NAME DSC ON STACK
                51   5E   D0  0125   350        MOVL    SP,R1                       ;GET ADR OF FILNAM STRING DESCRIPTOR
                52   83   9A  0128   351        MOVZBL  (R3)+,R2                    ;GET SIZE OF DEFAULT NAME STRING
                7E   52   7D  012B   352        MOVQ    R2,-(SP)                    ;PUT DEFAULT NAME DSC ON STACK
                50   5E   D0  012E   353        MOVL    SP,R0                       ;GET ADR OF DFLTNAM STRING DESCRIPTOR
                              0131   354        $IMGACT_S -                         ;ACTIVATE THE IMAGE
                              0131   355                NAME=(R1),-                 ;ADDRESS OF FILE NAME DESCRIPTOR
                              0131   356                DFLNAM=(R0),-               ;ADDRESS OF DEFAULT NAME DESCRIPTOR
                              0131   357                HDRBUF=(R5)                 ;ADDRESS OF IMAGE HEADER BUFFER
                  06 50   E9  0146   358        BLBC    R0,11$                      ;BRANCH IF ACTIVATION ERROR
                5E   10   C0  0149   359        ADDL    #16,SP                      ;CLEAN OFF IMGACT DESCRIPTORS
                   00DB   31  014C   360        BRW     30$                         ;AND GO ACTIVATE THE IMAGE
                     6E   7C  014F   361 11$:   CLRQ    (SP)                        ;SET UP NULL RESULT NAME DESCRIPTOR
                53   50   D0  0151   362        MOVL    R0,R3                       ;SAVE STATUS
                              0154   363        $RUNDWN_S #PSL$C_USER               ;RUN DOWN IMAGE
                              015D   364  ;
                              015D   365  ; CHECK IF THE IMAGE THAT JUST FAILED WAS LOGINOUT.
                              015D   366  ;
        08 AE   FEB4 CF   91  015D   367        CMPB    LOGOUTIMG,8(SP)             ;CHECK LENGTH FIRST
                     31   12  0163   368        BNEQ    111$                        ;BR IF NOT SAME LENGTH
  FEA9 CF   0C BE   08 AE   29  0165   369        CMPC3   8(SP),@12(SP),LOGOUTIMG+1          ;WAS IMAGE LOGINOUT?
                     27   12  016D   370        BNEQ    111$                        ;NO, PROCESS NORMAL ERROR PATH
                              016F   371  ;
                              016F   372  ; FAILED IMAGE ACTIVATION WAS LOGINOUT. WILL USE $EXIT_S TO TERMINATE
                              016F   373  ; THE PROCESS INSTEAD
                              016F   374  ;
             50   1C AB   D0  016F   375        MOVL    PRC_L_INDFAB(R11),R0        ;GET ADDR OF INDIRECT FAB
                0114 CB   B0  0173   376        MOVW    PRC_W_OUTIFI(R11),-         ;GET INTERNAL FILE INDEX OF SYS$OUTPUT
                  02 A0       0177   377                FAB$W_IFI(R0)               ;
                              0179   378        $CLOSE  FAB=(R0)                    ;CLOSE INDIRECT OUTPUT FILE
                              0182   379        $CANEXH_S                           ;CANCEL SUPERVISOR MODE EXIT HANDLERS
```

```
                           018B          $EXIT_S PRC_L_LSTSTATUS(R11)    ;EXIT PROCESS WITH FINAL STATUS
                           0196
                           0196
                   54  D4  0196  111$:   CLRL    R4                      ;INIT NULL STV FROM FAB
         55    08  A5  D0  0198          MOVL    8(R5),R5                ;FAB ADDRESS OF ERROR IF ANY
                   29  13  019C          BEQL    14$                     ;BRANCH IF NONE PRESENT
         52    2C  A5  D0  019E          MOVL    FAB$L_FNA(R5),R2        ;ORIGINAL FILE NAME ADDRESS
                   23  13  01A2          BEQL    14$                     ;BRANCH IF FAB IS EMPTY
         51    34  A5  9A  01A4          MOVZBL  FAB$B_FNS(R5),R1        ;SIZE OF ORIGINAL FILE NAME STRING
         08  AE    51  7D  01A8          MOVQ    R1,8(SP)                ;USE THIS FOR ORIGINAL FILE NAME
         54    0C  A5  D0  01AC          MOVL    FAB$L_STV(R5),R4        ;GET FAB'S STV FIELD
         50    28  A5  D0  01B0          MOVL    FAB$L_NAM(R5),R0        ;GET ADDRESS OF NAME BLOCK
         52    04  A0  D0  01B4          MOVL    NAM$L_RSA(R0),R2        ;ADDRESS OF NAME STRING
         51    03  A0  9A  01B8          MOVZBL  NAM$B_RSL(R0),R1        ;SIZE OF RESULT NAME STRING IF ANY
                   06  12  01BC          BNEQ    12$                     ;BRANCH IF IT EXISTS
         51    0B  A0  9A  01BE          MOVZBL  NAM$B_ESL(R0),R1        ;EXPANDED NAME STRING SIZE IF ANY
                   03  13  01C2          BEQL    14$                     ;BRANCH IF NO NAME TO OUTPUT
               6E  51  7D  01C4  12$:    MOVQ    R1,(SP)                 ;RESULT/EXPANDED NAME STRING
               55  5E  D0  01C7  14$:    MOVL    SP,R5                   ;R5 IS ADR OF NAME DESCRIPTORS
                           01CA  ;
                           01CA  ; 0(SP) = DESCRIPTOR FOR RESULT/EXPANDED FILE NAME
                           01CA  ; 8(SP) = DESCRIPTOR FOR ORIGINAL FILE NAME
                           01CA  ;
    52    000388AA 8F  D0  01CA          MOVL    #CLI$_IMGNAME,R2        ;ERROR CODE FOR 2ND MSG
    53    00000000'8F  D1  01D1          CMPL    #RMS$_FNF,R3            ;IF ORIGINAL ERR IS FILE NOT FOUND
                   09  12  01D8          BNEQ    16$
                   53  D4  01DA          CLRL    R3                      ;THEN 3RD MSG IS NULL
    52    000388B2 8F  D0  01DC          MOVL    #CLI$_IMAGEFNF,R2       ;AND 2ND IS ''IMAGE FILE NOT FOUND''
                           01E3  ;
                           01E3  ; NOW BUILD THE PUTMSG ARGUMENT
                           01E3  ;
 01   53    0C     10  ED  01E3  16$:    CMPZV   #STS$V_FAC_NO,#STS$S_FAC_NO,R3,#1 ;IF THIS IS AN RMS ERROR CODE
                   02  12  01E8          BNEQ    17$
                   54  DD  01EA          PUSHL   R4                      ;THEN USE SAVED STV FROM FAB
                   53  DD  01EC  17$:    PUSHL   R3                      ;ERROR CODE FOR 3RD MSG
                   65  D5  01EE          TSTL    (R5)                    ;IF NO RESULT NAME STRING
                   09  12  01F0          BNEQ    18$
    52    000388AA 8F  D1  01F2          CMPL    #CLI$_IMGNAME,R2                ;THEN SKIP THE IMAGE NAME MESSAGE
                   06  13  01F9          BEQL    182$
                   65  7F  01FB  18$:    PUSHAQ  (R5)                    ;ADR OF DESCRIPTOR FOR RSLT NAME
                   01  DD  01FD          PUSHL   #1                      ;ONE FAO ARGUMENT
                   52  DD  01FF          PUSHL   R2                      ;ERR CODE FOR 2ND MSG
               08  A5  7F  0201  182$:   PUSHAQ  8(R5)                   ;ADR OF DESCRIPTOR FOR ORIG NAME
                   01  DD  0204          PUSHL   #1                      ;ONE FAO ARGUMENT
          00030000'8F  DD  0206          PUSHL   #<SHR$_ACTIMAGE ! -     ;ERR CODE FOR 1ST MSG
                           020C                  <CLI$_IMGNAME & STS$M_FAC_NO>>
       7E    55    5E  C3  020C          SUBL3   SP,R5,-(SP)             ;FORM SIZE OF ARG LIST IN BYTES
             6E    04  C6  0210          DIVL    #4,(SP)                 ;ARG LIST SIZE IN LONG WORDS
                           0213  ;
                           0213  ; NOW OUTPUT THE ERROR MESSAGES
                           0213  ;
             50    5E  D0  0213          MOVL    SP,R0                   ;ADDRESS OF PUTMSG PARAMETERS
                FDE7'  30  0216          BSBW    DCL$PUTMSG              ;CALL THE PUTMSG FACILITY
       5E    55    10  C1  0219          ADDL3   #16,R5,SP               ;POP ALL INFO OFF STACK
             50    53  D0  021D          MOVL    R3,R0                   ;GET ERROR CODE TO RETURN
                   03  12  0220          BNEQ    19$                     ;BRANCH IF THIS IS THE ONE
             50    52  D0  0222          MOVL    R2,R0                   ;ONLY 2 MESSAGES, USE THE 2ND
          00 50    1C     E2  0225    437 19$:    BBSS    #STS$V_INHIB_MSG,R0,20$ ;INHIBIT ERROR MESSAGE OUTPUT
                          05  0229    438 20$:    RSB                             ;RETURN WITH STATUS CODE IN R0
                              022A    439
                              022A    440 ;
                              022A    441 ; RAISE ACCESS LEVEL TO USER
                              022A    442 ;
                              022A    443
                              022A    444 30$:
             51    65     D0  022A    445         MOVL    (R5),R1                 ;GET ADDRESS OF IMAGE HEADER
             50    00'    D1  022D    446         CMPL    S^#SS$_NORMAL,R0        ;IF OTHER THAN NORMAL COMPLETION
                   13     13  0230    447         BEQL    35$
       50    00000000'8F  D1  0232    448         CMPL    #SS$_SYSVERDIF,R0       ;"SYSTEM VERSION MISMATCH"?
                   0A     12  0239    449         BNEQ    35$                     ;BRANCH IF NOT
          02 A1    28     B1  023B    450         CMPW    #IHD$L_SYSVER,IHD$W_ACTIVOFF(R1) ;IF RELEASE 1 IMAGE
                              023F    451                                         ;WITH NO SYSVER STORED IN HEADER
                   04     19  023F    452         BLSS    35$                     ;THEN AVOID ISSUING SPURIOUS WARNING
          50     00'8F    9A  0241    453         MOVZBL  #SS$_NORMAL,R0          ;BY CONVERTING TO NORMAL STATUS
                              0245    454 35$:
                   01     E1  0245    455         BBC     #IHD$V_LNKNOTFR,-       ;BRANCH IF TRANSFER ADDRESS PRESENT
             19 20 A1         0247    456                 IHD$L_LNKFLAGS(R1),37$
             50    00'    D1  024A    457         CMPL    S^#SS$_NORMAL,R0        ;DID IMAGE ACTIVATION SUCCEED NORMALLY?
                   03     13  024D    458         BEQL    36$                     ;IF EQL YES
                 FDAE'    30  024F    459         BSBW    DCL$ERRORMSG            ;PRINT ALTERNATE SUCCESS MESSAGE
                              0252    460 36$:    $RUNDWN_S #PSL$C_USER           ;RUN DOWN IMAGE IF NO XFER ADDRESS
       50    0003889A 8F  D0  025B    461         MOVL    #CLI$_NOTFR,R0          ;AND RETURN STATUS OF "NO TRANSFER ADDRESS"
                          05  0262    462         RSB
        0080 CB    50     D0  0263    463 37$:    MOVL    R0,PRC_L_IMGACTSTS(R11) ;STORE IMGACT STATUS FOR LATER
          50    04 A5     D0  0268    464         MOVL    4(R5),R0                ;IMAGE FILE DESCRIPTOR BLOCK ADDRESS
                              026C    465
                              026C    466         ASSUME  IFD$V_EXEONLY+1 EQ IFD$V_PRIV
                              026C    467         ASSUME  PRC_V_EXEONLY+1 EQ PRC_V_PRIV
  50    10 A0    02    00     EF  026C    468         EXTZV   #IFD$V_EXEONLY,#2,IFD$Q_FLAGS(R0),R0 ;GET "EXECUTE ONLY"
                              0272    469                                         ;AND "PRIVILEGED IMAGE BITS"
 00AF CB    02    03    50    F0  0272    470         INSV    R0,#PRC_V_EXEONLY,#2,PRC_B_FLAGS2(R11) ;SAVE IN PROCESS FLAGS
                              0279    471         SETBIT  PRC_V_IRUNDWN,PRC_B_IMGFLAG(R11) ;INDICATE IMAGE ACTIVATED
          F4 AA    5E     D0  027D    472         MOVL    SP,WRK_L_SAVSP(R10)     ;SAVE CURRENT STACK POINTER
        7E    0F   16     78  0281    473         ASHL    #PSL$V_PRVMOD,#PSL$C_USER@2+PSL$C_USER,-(SP) ;SET USER PSL
                97'AF     9F  0285    474         PUSHAB  B^50$                   ;SET USER PC
                              0288    475 DCL$LOW_LIMIT::                         ;LOW LIMIT OF CONTROL Y/C ADDRESS RANGE
       09 68 AB    01     E1  0288    476         BBC     #PRC_V_CNTRLY,PRC_W_FLAGS(R11),40$ ;IF CLR, NO AST PENDING
             7E    50     7D  028D    477         MOVQ    R0,-(SP)                ;PUSH R0 AND R1
             7E    05     7D  0290    478         MOVQ    #5,-(SP)                ;PUSH #5 AND ZERO ASTPRM
                 FD6A'    31  0293    479         BRW     DCL$SCNTRLY             ;SIMULATE A CONTROL Y
                          02  0296    480 40$:    REI                             ;ENTER USER MODE
                              0297    481 DCL$HIGH_LIMIT::                        ;HIGH LIMIT OF CONTROL Y/C ADDRESS RANGE
                              0297    482
                              0297    483 ;
                              0297    484 ; BUILD TOP LEVEL CALL FRAME
                              0297    485 ;
                              0297    486
                   5C     7C  0297    487 50$:    CLRQ    AP                      ;CLEAR INITIAL ARGUMENT AND FRAME POINTERS
          9D'AF    00     FB  0299    488         CALLS   #0,B^60$                ;CONSTRUCT TOP LEVEL CALL FRAME
                              029D    489
                              029D    490 ;
                              029D    491 ; ESTABLISH CATCH-ALL CONDITION HANDLER AND CALL IMAGE
                              029D    492 ;
                              029D    493
```

```
                     0000  029D  60$:   .WORD   0                               ;ENTRY MASK
6D    00000000'GF      9E  029F         MOVAB   G^EXE$CATCH_ALL,(FP)            ;ESTABLISH CATCH-ALL HANDLER
                           02A6         $SETEXV_S #2,G^EXE$CATCH_ALL            ;ESTABLISH LAST CHANCE HANDLER
                           02B9         $IMGFIX_S                               ;PERFORM ADDRESS RELOCATION
      0080 CB    00'   D1  02C0         CMPL    S^#SS$_NORMAL,PRC_L_IMGACTSTS(R11) ;NORMAL IMAGE ACTIVATION?
                 2A    13  02C5         BEQL    65$                             ;IF EQL YES
                 50    DD  02C7         PUSHL   R0                              ;SAVE IMGFIX STATUS
                           02C9
                 00    DD  02C9         PUSHL   #0                              ;CREATE PUTMSG VECTOR (FAO COUNT)
            0080 CB    DD  02CB         PUSHL   PRC_L_IMGACTSTS(R11)            ;SET IMGACT STATUS CODE
                 02    DD  02CF         PUSHL   #2                              ;# ARGS ON PUTMSG VECTOR
            50   5E    D0  02D1         MOVL    SP,R0                           ;ADDRESS OF THE BUFFER DESCRIPTOR
                           02D4
        204C4344 8F    DD  02D4         PUSHL   #^A'DCL '                       ; FACILITY NAME
                 5E    DD  02DA         PUSHL   SP                              ; MAKE DESCRIPTOR OF NAME
                 03    DD  02DC         PUSHL   #3
                 5E    DD  02DE         PUSHL   SP                              ; SET ADDRESS OF FACNAM
                 00    DD  02E0         PUSHL   #0                              ; NO ACTION ROUTINE
                 50    DD  02E2         PUSHL   R0                              ; R0 = ADDRESS OF MESSAGE VECTOR
00000000'GF      06    FB  02E4         CALLS   #6,G^SYS$PUTMSG                 ; WRITE THE MESSAGE TO SYS$ERROR,OUTPUT
         5E      0C    C0  02EB         ADDL    #12,SP                          ; RESTORE THE STACK
                           02EE
            50   8E    D0  02EE         MOVL    (SP)+,R0                        ;RESTORE IMGFIX STATUS
              47 50    E9  02F1  65$:   BLBC    R0,110$                         ;IF LBC, WASH UP IMAGE
54    00000000'GF      7D  02F4         MOVQ    G^MMG$IMGHDRBUF,R4              ;GET IMAGE HEADER DESCRIPTOR
                           02FB
                           02FB         ASSUME  CLI$V_DEBUG   EQ 0
                           02FB         ASSUME  CLI$V_DBGTRU  EQ 1
                           02FB         ASSUME  PRC_V_DBGTRUE EQ PRC_V_DBGQUAL+1
                           02FB
            02   09    EF  02FB         EXTZV   #PRC_V_DBGQUAL,#2,-             ;BUILD PROTOTYPE CLI OPTIONS VALUE BY
       7E     68 AB        02FE                 PRC_W_FLAGS(R11),-(SP)          ;GETTING THE DEBUG QUALIFIER AND STATE
   03 68 AB      06    E1  0301         BBC     #PRC_V_MODE,PRC_W_FLAGS(R11),70$ ;BR IF NOT A BATCH JOB
                           0306         SETBIT  CLI$V_BATCH,(SP)                ;OR IN THE BATCH BIT IF THIS IS BATCH
   03 68 AB      07    E1  0309  70$:   BBC     #PRC_V_VERIFY,PRC_W_FLAGS(R11),80$ ;BR IF VERIFY IS CLEAR
                           030E         SETBIT  CLI$V_VERIFY,(SP)               ;PROPOGATE VERIFY IF TRUE
              5C AB    D5  0311  80$:   TSTL    PRC_L_INDEPTH(R11)              ;INDIRECT LEVEL ZERO?
                 03    13  0314         BEQL    90$                             ;IF EQL YES
                           0316         SETBIT  CLI$V_INDIRECT,(SP)             ;PASS INDIRECT NON ZERO FLAG
         6E      18    93  0319  90$:   BITB    #<CLI$M_INDIRECT!CLI$M_BATCH>,(SP) ;COMMANDS COMING FROM FILE?
                 07    13  031C         BEQL    100$                            ;BR IF TERMINAL JOB
      03 6E      02    E1  031E         BBC     #CLI$V_VERIFY,(SP),100$         ;BR IF VERIFY NOT REQUESTED
                           0322         SETBIT  CLI$V_VFYINP,(SP)               ;INDICATE INPUT VERIFY IS NEEDED
              20 A4    DD  0325  100$:  PUSHL   IHD$L_LNKFLAGS(R4)              ;PASS LINKTIME OPTION FLAGS
         7E      54    7D  0328         MOVQ    R4,-(SP)                        ;NEXT TWO PARAMETERS IN USER FRAME
           0000'CF     9F  032B         PUSHAB  W^DCL$UTLSERV                   ;SET ADDRESS OF UTILITY ROUTINE DISPATCHER
      50      02 A4    3C  032F         MOVZWL  IHD$W_ACTIVOFF(R4),R0           ;OFFSET TO ACTIVATION DATA (TRANSFER VECTORS
         50      54    C0  0333         ADDL    R4,R0                           ;ADDRESS OF TRANSFER VECTOR ARRAY
                 60    DF  0336         PUSHAL  (R0)                            ;ADDRESS OF TRANSFER VECTOR ARRAY
         90      06    FB  0338         CALLS   #6,@(R0)+                       ;CALL IMAGE ENTRY POINT
00000000'GF      17        033B  110$:  JMP     G^EXE$EXIT_IMAGE                ;
                           0341         .DSABL  LSB
                     0341              .SBTTL  DCL$FORCEXIT - FORCE IMAGE EXIT
                     0341      ;+
                     0341      ; FORCE IMAGE EXIT IF ONE IS ACTIVE
                     0341      ;
                     0341      ; INPUTS:
                     0341      ;
                     0341      ;       R2 = POSSIBLE ADDRESS OF ITEM ON STACK
                     0341      ;
                     0341      ; OUTPUTS:
                     0341      ;
                     0341      ;       R2 = RELOCATED ADDRESS IF STACK IS SHUFFLED
                     0341      ;       R0 ALTERED, R1, R3 PRESERVED
                     0341      ;-
                     0341
                     0341      DCL$FORCEXIT::                                  ;FORCE IMAGE EXIT
 50     5E     D0    0341              MOVL    SP,R0                           ;SAVE SP ADR IN ORDER TO DETECT A SHUFFLE
        0F     BB    0344              PUSHR   #^M<R0,R1,R2,R3>                ;SAVE NAME STRING PARAMETERS
     FCB7'     30    0346              BSBW    DCL$RUNDOWN                     ;RUN DOWN PREVIOUS IMAGE AND INDIRECT LEVELS
        0F     BA    0349              POPR    #^M<R0,R1,R2,R3>                ;RESTORE NAME STRING PARAMETERS
 5E     50     D1    034B              CMPL    R0,SP                           ;WAS THE STACK SHUFFLED?
        0B     13    034E              BEQL    10$                             ;BRANCH IF NOT
 50     52     D1    0350              CMPL    R2,R0                           ;WAS NAME STRING ON THE STACK?
        06     1F    0353              BLSSU   10$                             ;BRANCH IF NOT
 52     50     C2    0355              SUBL    R0,R2                           ;ADJUST THE ADDRESS
 52     5E     C0    0358              ADDL    SP,R2                           ;TO REFLECT THE SHUFFLE
               05    035B      10$:    RSB
```

```
                      035C              .SBTTL  ALLOCATE BUFFER AND BUILD DESCRIPTOR
                      035C       ;+
                      035C       ; DCL$ALLOCBUF - ALLOCATE BUFFER AND BUILD DESCRIPTOR
                      035C       ;
                      035C       ; THIS ROUTINE IS CALLED TO ALLOCATE A MESSAGE BUFFER ON THE STACK AND BUILD
                      035C       ; A BUFFER DESCRIPTOR FOR THE BUFFER.
                      035C       ;
                      035C       ; INPUTS:
                      035C       ;
                      035C       ;       NONE.
                      035C       ;
                      035C       ; OUTPUTS:
                      035C       ;
                      035C       ;       A MESSAGE BUFFER IS ALLOCATED ON THE STACK.
                      035C       ;
                      035C       ;       R2 = ADDRESS OF BUFFER DESCRIPTOR.
                      035C       ;
                      035C       ;       REGISTERS R3, R4, AND R5 ARE PRESERVED ACROSS CALL.
                      035C       ;-
                      035C
                      035C      DCL$ALLOCBUF::                                          ;ALLOCATE BUFFER AND BUILD DESCRIPTOR
             01   BA  035C              POPR    #^M<R0>                                 ;SAVE RETURN ADDRESS
  5E   FF7C  CE   9E  035E              MOVAB   -WRK_C_MSGBUFSIZ(SP),SP                 ;ALLOCATE SPACE TO STORE FORMAT STRING
             6E   9F  0363              PUSHAB  (SP)                                    ;BUILD OUTPUT BUFFER DESCRIPTOR
     7E   84 8F   9A  0365              MOVZBL  #WRK_C_MSGBUFSIZ,-(SP)                  ;
        52   5E   D0  0369              MOVL    SP,R2                                   ;COPY ADDRESS OF OUTPUT BUFFER DESCRIPTOR
             60   17  036C              JMP     (R0)                                    ;
```

```
                          036E          .SBTTL  COMMAND INTERPRETER EXIT HANDLER
                          036E  ;+
                          036E  ; DCL$EXITHAND - COMMAND INTERPRETER EXIT HANDLER
                          036E  ;
                          036E  ; THIS ROUTINE IS ENTERED WHEN A PREVIOUSLY INITIATED IMAGE EXITS. ITS FUNC-
                          036E  ; TION IS TO CLEAN UP THE STACK, SHUTDOWN THE IMAGE, AND RETURN CONTROL TO THE
                          036E  ; ADDRESS SPECIFIED BY THE TOP LONGWORD OF THE STACK.
                          036E  ;
                          036E  ; INPUTS:
                          036E  ;
                          036E  ;       @4(AP) = REASON FOR EXIT.
                          036E  ;
                          036E  ; OUTPUTS:
                          036E  ;
                          036E  ;       THE SAVED FRAME POINTER IS RESTORED, THE STACK IS CLEANED UP, IMAGE
                          036E  ;       SHUT DOWN IS PERFORMED,  AND CONTROL IS RETURNED TO THE ADDRESS SPECIFIED
                          036E  ;       BY THE TOP LONGWORD OF THE STACK.
                          036E  ;
                          036E  ;       R0 = REASON FOR EXIT.
                          036E  ;-
                          036E
                    0000  036E          .ENTRY  DCL$EXITHAND,^M<>
                          0370
               FC8D'  30  0370          BSBW    CLI$GET_PRC                     ;GET ADDRESS OF CLI WORK AREA
      68 AB      08   AA  0373          BICW    #PRC_M_EXIT,PRC_W_FLAGS(R11)    ;CLEAR EXIT HANDLER ESTABLISHED
      5D   04 AB      D0  0377          MOVL    PRC_L_SAVFP(R11),FP             ;RESTORE SAVED FRAME POINTER
         5A      5D   D0  037B          MOVL    FP,R10                          ;AND RESTORE WRK ADDRESS
                          037E  ;
                          037E  ; ZERO COMMAND INTERPRETER COMMAND POINTERS
                          037E  ;
        00000000'GF   D4  037E          CLRL    G^CTL$GL_CLINTOWN               ;ZERO CLINT OWN STORAGE POINTER
        00000000'GF   D4  0384          CLRL    G^CTL$GL_DCLPRSOWN              ;ZERO DCL PARSE OWN STORAGE
                          038A  ;
                          038A  ; ISSUE ERROR MESSAGE (IF ANY) RETURNED BY IMAGE IN R0
                          038A  ;
       50    04 BC    D0  038A  10$:    MOVL    @4(AP),R0                       ;RETRIEVE FINAL EXIT STATUS
             0C 50    E8  038E          BLBS    R0,20$                          ;BRANCH IF SUCCESSFUL
    00000000'8F  50   D1  0391          CMPL    R0,#SS$_CLIFRCEXT               ;NEVER ISSUE CLI FORCED EXIT MESSAGE
                 03   13  0398          BEQL    20$                             ;IF IMPLIED IMAGE RUNDOWN
               FC63'  30  039A          BSBW    DCL$ERRORMSG                    ;ISSUE ERROR MESSAGE USING PER-IMAGE
                          039D                                                  ;MESSAGES IF PRESENT
                          039D  ;
                          039D  ; RUNDOWN ALL RMS FILES AND FLUSH ANY DATA RECORDS
                          039D  ;
               FC60'  30  039D  20$:    BSBW    DCL$SHUTDOWN                    ;SHUT DOWN IMAGE
                 52   D5  03A0          TSTL    R2                              ;ANY DATA RECORDS SKIPPED?
                 0A   13  03A2          BEQL    30$                             ;IF EQL NO
                          03A4          STATUS  SKPDAT                          ;SET SKIPPED DATA STATUS
               FC52'  30  03AB          BSBW    DCL$ERRORMSG                    ;OUTPUT ERROR MESSAGE
                          03AE  ;
                          03AE  ; RUNDOWN THE IMAGE
                          03AE  ;
                          03AE  30$:    $RUNDWN_S  #PSL$C_USER                  ;RUN DOWN THE IMAGE
    00AF CB      18   8A  03B7          BICB    #<PRC_M_EXEONLY ! PRC_M_PRIV>,PRC_B_FLAGS2(R11)
                          03BC                                                  ;RESET "EXECUTE ONLY" AND
                          03BC                                                  ;"PRIVILEGED" IMAGE BITS
                          03BC          CLRBIT  PRC_V_IRUNDWN,PRC_B_IMGFLAG(R11) ;INDICATE IMAGE RUNDOWN.
```

```
                              03C0      ;
                              03C0      ; RESTORE STACK POINTER TO SP SAVED ON THE ACTIVATE CALL
                              03C0      ;
        5E      F4 AA    D0   03C0              MOVL    WRK_L_SAVSP(R10),SP     ;RESTORE SAVED STACK POINTER
                              03C4      ;
                              03C4      ; SET THE FINAL RETURN STATUS IN $STATUS AND TAKE ANY ON CONDITION
                              03C4      ;
        50      04 BC    D0   03C4              MOVL    @4(AP),R0               ;GET FINAL IMAGE STATUS
                09 50    E8   03C8              BLBS    R0,40$                  ;BRANCH IF SUCCESSFUL
 00000000'8F       50    D1   03CB              CMPL    R0,#SS$_CLIFRCEXT       ;NEVER SET IMPLIED IMAGE RUNDOWN STATUS
                   03    13   03D2              BEQL    50$                     ;BRANCH IF FORCED EXIT
                FC29'    30   03D4      40$:    BSBW    DCL$SET_STATUS          ;SET $STATUS AND TAKE ON CONDITION
                              03D7                                              ;NOTE: WILL NOT RETURN IF ON CONDITION
                              03D7      ;
                              03D7      ; SET STATUS TO SUCCESS AND SET NOSTAT BIT TO INDICATE STATUS ALREADY SET
                              03D7      ;
                              03D7      50$:    SETBIT  #WRK_V_NOSTAT,WRK_W_FLAGS(R10) ;INDICATE STATUS ALREADY SAVED
        50         01    D0   03DC              MOVL    #1,R0                   ;AND SET SUCCESSFUL
                         05   03DF              RSB                             ;RETURN TO CALLER (COMMAND OR RUNDOWN)
                              03E0
                              03E0              .END
```

```
$$.TMP1                 = 00000001              IMG_K_ASSIGN            = 00000081
$$.TMP2                 = 00000060              IMG_K_ATTACH            = 000000A9
$$T1                    = 00000000              IMG_K_CANCEL            = 000000AB
$INTIMAGES$             = 000000B6              IMG_K_CLOSE             = 00000082
CLI$GET_PRC               ********    X    02   IMG_K_CONNECT           = 000000B4
CLI$M_BATCH             = 00000008              IMG_K_CONTINUE          = 00000083
CLI$M_INDIRECT          = 00000010              IMG_K_CRETABLE          = 000000B3
CLI$V_BATCH             = 00000003              IMG_K_DEALLOCAT         = 00000084
CLI$V_DBGTRU            = 00000001              IMG_K_DEASSIGN          = 00000085
CLI$V_DEBUG             = 00000000              IMG_K_DEBUG             = 00000086
CLI$V_INDIRECT          = 00000004              IMG_K_DECK              = 00000087
CLI$V_VERIFY            = 00000002              IMG_K_DEFINE            = 00000088
CLI$V_VFYINP            = 00000005              IMG_K_DEFKEY            = 000000AE
CLI$_IMAGEFNF           = 000388B2              IMG_K_DELKEY            = 000000B0
CLI$_IMGNAME            = 000388AA              IMG_K_DELSYM            = 0000008A
CLI$_NOTFR              = 0003889A              IMG_K_DEPOSIT           = 00000089
CLI$_SKPDAT             = 00038120              IMG_K_DISCONNECT        = 000000B5
CTL$AG_CLIDATA            ********    X    02   IMG_K_EOD               = 0000008B
CTL$GL_CLINTOWN           ********    X    02   IMG_K_EOJ               = 000000AA
CTL$GL_DCLPRSOWN          ********    X    02   IMG_K_EXAMINE           = 0000008C
DCL$ALLOCBUF              0000035C RG      02   IMG_K_EXIT              = 0000008D
DCL$CLOSE_PPFS            00000073 RG      02   IMG_K_EXTIMAGE          = 0000008E
DCL$ERRORMSG              ********    X    02   IMG_K_GOTO              = 0000008F
DCL$EXITHAND              0000036E RG      02   IMG_K_IF                = 00000090
DCL$EXTIMAGE              000000E9 RG      02   IMG_K_INQUIRE           = 00000091
DCL$FORCEXIT              00000341 RG      02   IMG_K_LOGOUT            = 00000092
DCL$GETDVAL               ********    X    02   IMG_K_MCR               = 00000093
DCL$HIGH_LIMIT            00000297 RG      02   IMG_K_ON                = 00000094
DCL$LOGOUT                0000005A RG      02   IMG_K_OPEN              = 00000095
DCL$LOW_LIMIT             00000288 RG      02   IMG_K_READ              = 00000096
DCL$MCR                   000000A8 RG      02   IMG_K_RECALL            = 000000AD
DCL$PUTMSG                ********    X    02   IMG_K_RUN               = 00000097
DCL$RESTORE_OUTPUT        ********    X    02   IMG_K_SETCTLY           = 00000098
DCL$RUN                   0000002B RG      02   IMG_K_SETDEFALT         = 00000099
DCL$RUNDOWN               ********    X    02   IMG_K_SETFLUSH          = 000000B2
DCL$SCNTRLY               ********    X    02   IMG_K_SETKEY            = 000000B1
DCL$SET_STATUS            ********    X    02   IMG_K_SETON             = 0000009A
DCL$SHUTDOWN              ********    X    02   IMG_K_SETPROMPT         = 000000AC
DCL$UNSTACK               ********    X    02   IMG_K_SETPROT           = 0000009B
DCL$UTLSERV               ********    X    02   IMG_K_SETUIC            = 0000009C
EXE$CATCH_ALL             ********    X    02   IMG_K_SETVERIFY         = 0000009D
EXE$EXIT_IMAGE            ********    X    02   IMG_K_SHOWDEF           = 0000009E
EXTDEFAULT                00000000 R       02   IMG_K_SHOWKEY           = 000000AF
FAB$B_FNS               = 00000034              IMG_K_SHOWPROT          = 0000009F
FAB$L_FNA               = 0000002C              IMG_K_SHOWQUOTA         = 000000A0
FAB$L_NAM               = 00000028              IMG_K_SHOWSTAT          = 000000A1
FAB$L_STV               = 0000000C              IMG_K_SHOWSYMBL         = 000000A2
FAB$W_IFI               = 00000002              IMG_K_SHOWTIME          = 000000A3
FORCEXIT                  000000FF R       02   IMG_K_SHOWTRAN          = 000000A4
IFD$V_EXEONLY           = 00000000              IMG_K_SPAWN             = 000000A8
IFD$V_PRIV              = 00000001              IMG_K_STOP              = 000000A5
IFD$W_FLAGS             = 00000010              IMG_K_WAIT              = 000000A6
IHD$L_LNKFLAGS          = 00000020              IMG_K_WRITE             = 000000A7
IHD$L_SYSVER            = 00000028              INITIATE                  00000102 R       02
IHD$V_LNKNOTFR          = 00000001              LOGOUTIMG                 00000015 R       02
IHD$W_ACTIVOFF          = 00000002              MCRIMG                    0000001E R       02
IMG_K_ALLOCATE          = 00000080              MMG$IMGHDRBUF             ********    X    02
```

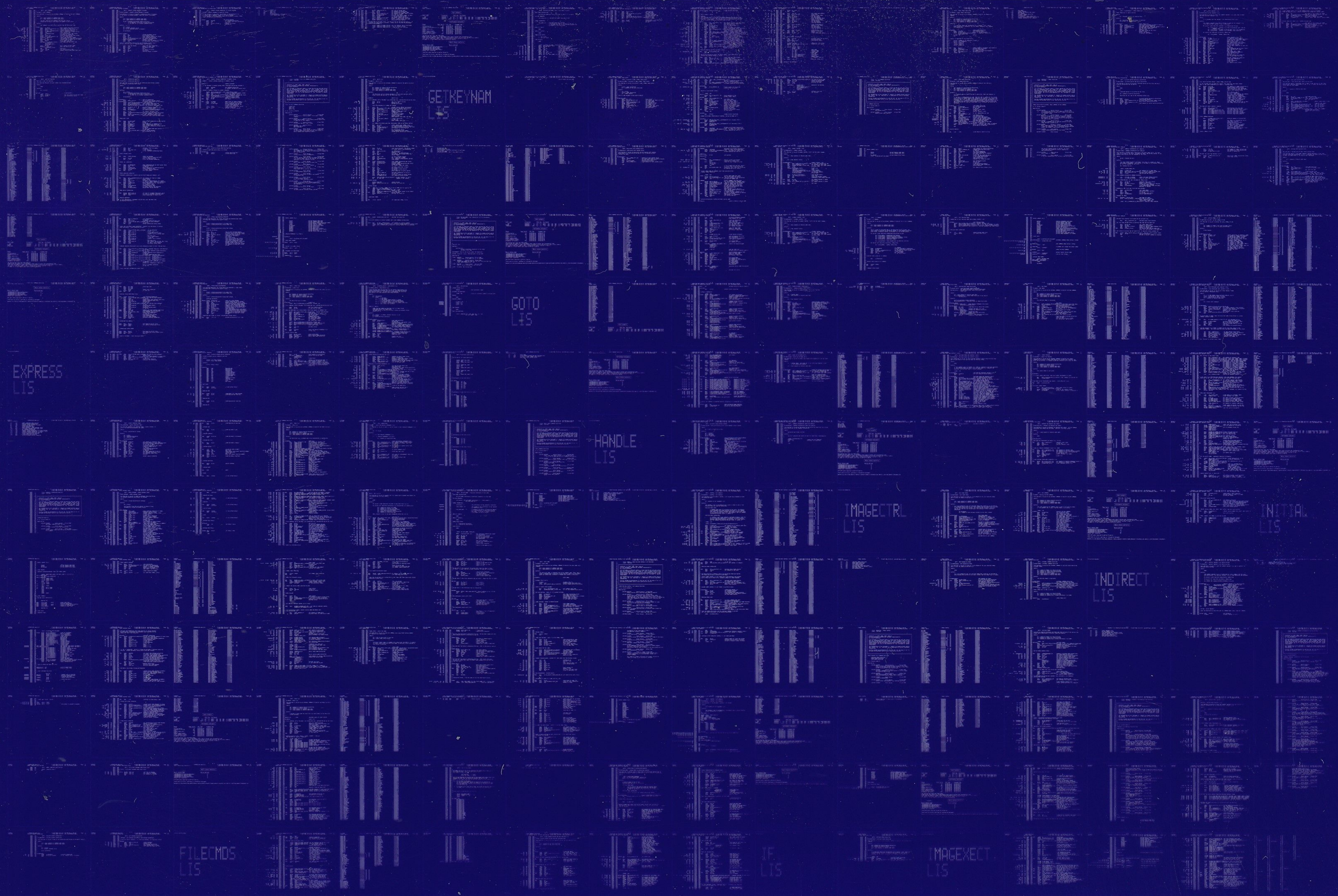
```
NAM$B_ESL                        = 0000000B              PRC_L_ONCTLY                       000000B8
NAM$B_RSL                        = 00000003              PRC_L_ONERROR                      0000006C
NAM$L_RSA                        = 00000004              PRC_L_OUTOFBAND                    000000B4
PPD$B_NPROCS                       0000001C              PRC_L_OUTRAB                       0000000C
PPD$C_LENGTH                       00000168              PRC_L_OUTRABCTX                    00000118
PPD$K_LENGTH                       00000168              PRC_L_PPFLIST                      00000070
PPD$L_INPDEV                       00000044              PRC_L_RECALLPTR                    0000012F
PPD$L_LGI                          00000014              PRC_L_RESTART                      00000058
PPD$L_LSTSTATUS                    00000018              PRC_L_SAVAP                        00000000
PPD$L_OUTDEV                       00000064              PRC_L_SAVFP                        00000004
PPD$L_PRC                          00000008              PRC_L_SEVERITY                     00000050
PPD$Q_CLIREG                       00000004              PRC_L_SPWN                         000000C0
PPD$Q_CLISYMTBL                    0000000C              PRC_L_STACKLM                      000000A4
PPD$T_FILENAME                     00000068              PRC_L_STACKPT                      000000A0
PPD$T_INPDVI                       00000028              PRC_L_STATUS                       00000054
PPD$T_OUTDVI                       00000048              PRC_L_STS                          00000084
PPD$W_FLAGS                        00000002              PRC_L_STV                          00000088
PPD$W_INPCHAN                      0000001E              PRC_L_SYMBOL                       00000060
PPD$W_INPDID                       0000003E              PRC_L_TMBX                         00000074
PPD$W_INPFID                       00000038              PRC_L_TRMLIST                      00000010
PPD$W_INPIFI                       00000020              PRC_M_DBGQUAL                    = 00000200
PPD$W_INPISI                       00000022              PRC_M_DBGTRUE                    = 00000400
PPD$W_OUTDID                       0000005E              PRC_M_EXEONLY                    = 00000008
PPD$W_OUTFID                       00000058              PRC_M_EXIT                       = 00000008
PPD$W_OUTIFI                       00000024              PRC_M_PRIV                       = 00000010
PPD$W_OUTISI                       00000026              PRC_Q_ALLOCREG                     00000020
PPD$W_SIZE                         00000000              PRC_Q_COMMAND                      000000E0
PRC_B_CONTINUE                     000000F3              PRC_Q_FLUSHTIME                    000000D0
PRC_B_DEFRADIX                     000000AE              PRC_Q_GLOBAL                       00000028
PRC_B_EXMDEPMOD                    000000AD              PRC_Q_IMAGENAME                    000000D8
PRC_B_EXMDEPWID                    000000AC              PRC_Q_KEYPAD                       00000040
PRC_B_EXONLYL                      0000012D              PRC_Q_LABEL                        00000030
PRC_B_FLAGS2                       000000AF              PRC_Q_LOCAL                        00000038
PRC_B_IMGFLAG                      00000078              PRC_Q_SAVEPRIV                     000000E8
PRC_B_OUTFLAGS                     0000012C              PRC_T_OUTDVI                       0000011C
PRC_B_PROMPTLEN                    000000F0              PRC_V_AUTOLOGO                   = 00000008
PRC_C_LENGTH                       00000534              PRC_V_CNTRLY                     = 00000001
PRC_G_COMMANDS                     00000133              PRC_V_DBGQUAL                    = 00000009
PRC_G_PROMPT                       000000F4              PRC_V_DBGTRUE                    = 0000000A
PRC_K_LENGTH                       00000534              PRC_V_EXEONLY                    = 00000003
PRC_L_CURRKEY                      00000048              PRC_V_EXIT                       = 00000003
PRC_L_EXMDEPADR                    000000A8              PRC_V_IRUNDWN                    = 00000000
PRC_L_EXTARG                       00000094              PRC_V_MODE                       = 00000006
PRC_L_EXTBLK                       0000008C              PRC_V_PRIV                       = 00000004
PRC_L_EXTCOD                       0000009C              PRC_V_RUNDEF                     = 00000002
PRC_L_EXTHND                       00000090              PRC_V_VERIFY                     = 00000007
PRC_L_EXTPRM                       00000098              PRC_W_ASTIOSB                      000000C6
PRC_L_IDFLNK                       000000BC              PRC_W_ASTRETN                      000000C8
PRC_L_IMGACTSTS                    00000080              PRC_W_ASTSTATUS                    000000C4
PRC_L_INDCLOCK                     0000007C              PRC_W_ATTMBX                       0000007A
PRC_L_INDEPTH                      0000005C              PRC_W_FLAGS                        00000068
PRC_L_INDFAB                       0000001C              PRC_W_INPCHAN                      00000064
PRC_L_INDINPRAB                    00000014              PRC_W_ONLEVEL                      0000006A
PRC_L_INDOUTRAB                    00000018              PRC_W_OUTIFI                       00000114
PRC_L_INPRAB                       00000008              PRC_W_OUTISI                       00000116
PRC_L_LASTKEY                      0000004C              PRC_W_OUTMBXCHN                    000000CA
PRC_L_LSTSTATUS                    000000B0              PRC_W_OUTMBXREF                    000000CE
```

```
PRC_W_OUTMBXSIZ                   000000CC
PRC_W_PMPTCTRL                    000000F1
PRC_W_WAITIOSB                    00000066
PSL$C_USER                      = 00000003
PSL$V_PRVMOD                    = 00000016
PTR_B_LEVEL                       00000004
PTR_B_NUMBER                      00000005
PTR_B_PARMCNT                     00000006
PTR_B_VALUE                       00000000
PTR_C_LENGTH                      0000000C
PTR_K_ENDLINE                   = 00000004
PTR_K_LENGTH                      0000000C
PTR_K_PARAMETR                  = 00000003
PTR_L_DESCR                       00000000
PTR_L_ENTITY                      00000008
RAB$L_CTX                       = 00000018
RMS$_FNF                          ********     X     02
RUNDEFAULT                        00000010 R         02
SETIMGDEF                         000000E9 R         02
SETIMGNAME                        000000B5 R         02
SETRUNDEF                         000000FA R         02
SHR$_ACTIMAGE                     ********     X     02
SILENT_LOGOUT                     ********     X     02
SS$_CLIFRCEXT                     ********     X     02
SS$_NORMAL                        ********     X     02
SS$_SYSVERDIF                     ********     X     02
STS$M_FAC_NO                    = 0FFF0000
STS$S_FAC_NO                    = 0000000C
STS$V_FAC_NO                    = 00000010
STS$V_INHIB_MSG                 = 0000001C
SYS$CANEXH                        ********    GX     02
SYS$CLOSE                         ********    GX     02
SYS$DCLEXH                        ********    GX     02
SYS$EXIT                          ********    GX     02
SYS$IMGACT                        ********    GX     02
SYS$IMGFIX                        ********    GX     02
SYS$PUTMSG                        ********     X     02
SYS$RUNDWN                        ********    GX     02
SYS$SETEXV                        ********    GX     02
SYSPRINT                          00000022 R         02
SYSPRTSIZ                       = 00000009
WRK_B_CMDOPT                      FFFFFFC3
WRK_B_MAXPARM                     FFFFFFD0
WRK_B_MINPARM                     FFFFFFD1
WRK_B_PARMCNT                     FFFFFFCE
WRK_B_PARMSUM                     FFFFFFCF
WRK_B_RECALLCNT                   FFFFFFC5
WRK_B_VALLEV                      FFFFFFC4
WRK_B_VERBTYP                     FFFFFFC2
WRK_C_LENGTH                      FFFFF486
WRK_C_MSGBUFSIZ                 = 00000084
WRK_G_BUFFER                      FFFFF492
WRK_G_INPBUF                      FFFFF896
WRK_G_RESULT                      FFFFF9B6
WRK_K_LENGTH                      FFFFF486
WRK_L_CHARPTR                     FFFFF48E
WRK_L_DISALLOW                    FFFFFFE6
WRK_L_ERRORRTN                    FFFFF9AE
WRK_L_EXPANDPTR                   FFFFF486
WRK_L_IMAGE                       FFFFFFE2
WRK_L_MARKPTR                     FFFFF48A
WRK_L_PAROUT                      FFFFFFD2
WRK_L_PMPTADDR                    FFFFF9A2
WRK_L_PROMPTRTN                   FFFFF9A6
WRK_L_PROPTR                      FFFFFFC6
WRK_L_QUABLK                      FFFFFFCA
WRK_L_READRTN                     FFFFF9AA
WRK_L_RECALLPTR                   FFFFFFEA
WRK_L_RSLEND                      FFFFFFB6
WRK_L_RSLNXT                      FFFFFFBA
WRK_L_SAVAP                       FFFFFFF8
WRK_L_SAVFP                       FFFFFFFC
WRK_L_SAVSP                       FFFFFFF4
WRK_L_SIGNALRTN                   FFFFFFD6
WRK_L_SPECRTN                     FFFFF9B2
WRK_L_TAB_VEC                     FFFFFFDE
WRK_L_VERB                        FFFFFFBE
WRK_V_NOSTAT                    = 00000008
WRK_W_FLAGS                       FFFFFFF0
WRK_W_FLAGS2                      FFFFFFF2
WRK_W_IMGCHAN                     FFFFFFEE
WRK_W_PMPTLEN                     FFFFF99E
_$$_                            = 00000000
```

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

PSECT name                 Allocation          PSECT No.  Attributes
----------                 ----------          ---------  ----------
.  ABS  .                  00000000  (     0.)  00 (  0.)  NOPIC   USR   CON   ABS   LCL NOSHR NOEXE NORD  NOWRT NOVEC BYTE
$ABS$                      FFFFFFFC  (     0.)  01 (  1.)  NOPIC   USR   CON   ABS   LCL NOSHR   EXE   RD    WRT NOVEC BYTE
DCL$ZCODE                  000003E0  (   992.)  02 (  2.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD  NOWRT NOVEC BYTE

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

Phase                    Page faults    CPU Time       Elapsed Time
-----                    -----------    --------       ------------
Initialization                  15    00:00:00.07    00:00:01.03
Command processing             102    00:00:00.64    00:00:05.73
Pass 1                         384    00:00:15.47    00:01:03.55
Symbol table sort                4    00:00:02.12    00:00:06.27
Pass 2                         120    00:00:03.03    00:00:12.20
Symbol table output             35    00:00:00.25    00:00:01.40
Psect synopsis output            2    00:00:00.02    00:00:00.02
Cross-reference output           0    00:00:00.00    00:00:00.00
Assembler run totals           662    00:00:21.60    00:01:30.20

The working set limit was 1500 pages.
82068 bytes (161 pages) of virtual memory were used to buffer the intermediate code.
There were 80 pages of symbol table space allocated to hold 1390 non-local and 42 local symbols.
679 source lines were read in Pass 1, producing 19 object records in Pass 2.
58 pages of virtual memory were used to define 42 macros.

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

Macro library name                           Macros defined
------------------                           --------------
_$255$DUA28:[SYSLIB]SYSBLDMLB.MLB;1                  0
_$255$DUA28:[DCL.OBJ]DCL.MLB;1                      10
_$255$DUA28:[SYS.OBJ]LIB.MLB;1                       2
_$255$DUA28:[SYSLIB]STARLET.MLB;2                   23
TOTALS (all libraries)                              35

1665 GETS were required to define 35 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:IMAGEXECT/OBJ=OBJ$:IMAGEXECT MSRC$:IMAGEXECT/UPDATE=(ENH$:IMAGEXECT)+EXECML$/LIB+LIB$:DCL/LIB+SYS$LIBRARY:SYSBLDMLB/L
```

0070 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



GETKEYNAM
LIS

GOTO
LIS

EXPRESS
LIS

HANDLE
LIS

IMAGECTRL
LIS

INITIAL
LIS

INDIRECT
LIS

FILECMDS
LIS

IF
LIS

IMAGEXECT
LIS