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
DDDDDDDDDDDD        CCCCCCCCCCCC  LLLLLLLLLLLLLLL
DDDDDDDDDDDD        CCCCCCCCCCCC  LLLLLLLLLLLLLLL
DDDDDDDDDDDD        CCCCCCCCCCCC  LLLLLLLLLLLLLLL
```

```
**FILE**ID**SET                                      D 8



  SSSSSSSS  EEEEEEEEEE  TTTTTTTTTT
  SSSSSSSS  EEEEEEEEEE  TTTTTTTTTT
SS          EE              TT
SS          EE              TT
SS          EE              TT
SS          EE              TT
  SSSSSS    EEEEEEEE        TT
  SSSSSS    EEEEEEEE        TT
        SS  EE              TT
        SS  EE              TT
        SS  EE              TT          ....
        SS  EE              TT          ....
SSSSSSSS    EEEEEEEEEE      TT          ....
SSSSSSSS    EEEEEEEEEE      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  SET - SET PARAMETER DCLS COMMAND 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   *
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     ; SET PARAMETER DCLS COMMAND EXECUTION
0000     ;
0000     ;       SET DIRECTORY
0000     ;       SET PROTECTION
0000     ;       SET USER IDENTIFICATION CODE
0000     ;       SET VERIFY MODE
0000     ;       SET ON
0000     ;       SET CONTROL
0000     ;       SET PROMPT
0000     ;
0000     ; D. N. CUTLER 17-APR-77
0000     ;
0000     ; MODIFIED BY:
0000     ;
0000     ;       V03-015 HWS0095         Harold Schultz  25-Jul-1984
0000     ;               Add support for execute-only command procedures.
0000     ;
0000     ;       V03-014 HWS0011         Harold Schultz  13-Feb-1984
0000     ;               Use PRC_V_CARRCNTL to indicate presence/absence of
0000     ;               CR/LF in prompt string.
0000     ;               Fix broken branch.
0000     ;
0000     ;       V03-013 PCG0012         Peter George    12-Oct-1983
0000     ;               Fix bug in SET NOON, ON severity, SET ON sequence.
0000     ;
0000     ;       V03-012 PCG0011         Peter George    18-Aug-1983
0000     ;               Change the way that default protection is changed.
0000     ;
0000     ;       V03-011 KBT0577         Keith B. Thompson       8-Aug-1983
0000     ;               Fix a bug in kbt0572
0000     ;
0000     ;       V03-010 KBT0572         Keith B. Thompson       1-Aug-1983
0000     ;               Use $TRNLNM in SET DEFAULT
0000     ;
0000     ;       V03-009 PCG0010         Peter George    07-Jul-1983
0000     ;               Update SET UIC.
0000     ;
0000     ;       V03-008 PCG0009         Peter George    31-May-1983
0000     ;               Reference $RMEDEF.
0000     ;
0000     ;       V03-007 PCG0008         Peter George    27-May-1983
0000     ;               Add image verification.
0000     ;
0000     ;       V03-006 PCG0007         Peter George    30-Apr-1983
0000     ;               Change reference to CRLF.
0000     ;
0000     ;       V03-005 PCG0006         Peter George    17-Feb-1983
0000     ;               Remove reference to $CLIDEFQUALSET.
0000     ;               Convert to new table structure.
0000     ;
0000     ;       V03-004 PCG0005         Peter George    19-Nov-1982
0000     ;               Let SET PROMPT with no argument restore the default
0000     ;               prompt.
0000     ;
0000     ;       V03-003 PCG0004         Peter George    28-Oct-1982
0000     ;               Add SET PROMPT.
```

```
                                      0000  ;
                                      0000  ;       V03-002 PCG0003         Peter George    22-Oct-1982
                                      0000  ;               Fix keyword parsing bug in SET PROTECTION.
                                      0000  ;
                                      0000  ;       V03-001 PCG0002         Peter George    01-Jul-1982
                                      0000  ;               Modify SET CONTROL and SET PROTECTION to interact with
                                      0000  ;               DCL keyword parsing.
                                      0000  ;
                                      0000  ;---
                                      0000
                                      0000  ;
                                      0000  ; MACRO LIBRARY CALLS
                                      0000  ;
                                      0000
                                      0000          $$CLITABDEF                             ;TABLE STRUCTURE DEFINITIONS
                                      0000          WRKDEF                                  ;DEFINE COMMAND WORK AREA
                                      0000          PRCDEF                                  ;DEFINE PROCESS WORK AREA
                                      0000          PTRDEF                                  ;DEFINE RESULT PARSE DESCRIPTOR FORMAT
                                      0000          IDFDEF                                  ;DEFINE INDIRECT FILE DATA STRUCTURE
                                      0000          $LNMDEF
                                      0000          $LOGDEF                                 ;LOGICAL NAME DEFINITIONS
                                      0000          $RMEDEF                                 ;DEFINE RME CONSTANTS
                                      0000          $PCBDEF                                 ;DEFINE PCB OFFSETS
                                      0000          $PRVDEF                                 ;PRIVILEGE BIT DEFINITIONS
                                      0000          $CLIMSGDEF                              ;DEFINE CLI RELATED ERRORS
                                      0000
                                      0000  ;
                                      0000  ; LOCAL DATA
                                      0000  ;
                                      0000
                                  00000000          .PSECT  DCL$ZCODE,BYTE,RD,NOWRT
                                      0000  ACCESS:                                         ;ACCESS PROTECTION CODES
                       52 57 45 44    0000          .ASCII  /DEWR/                          ;
                                      0004  CLASS:                                          ;PROTECTION CLASSES
                       53 4F 47 57    0004          .ASCII  /WGOS/                          ;
                                      0008
56 45 44 5F 45 4C 49 46 24 4D 4E 4C   0008  TABNAM: .ASCII  /LNM$FILE_DEV/                  ; Logical name table to search
                          0000000C    0014          TABNAMSZ=.-TABNAM                       ;  for device names
                                      0014
                                      0014  DCL$T_DSKNAM::                                  ; String for default device
         4B 53 49 44 24 53 59 53 00'  0014          .ASCIC  /SYS$DISK/                      ;
                                08    0014
                                      001D
                                      001D  CONTROL_CHARS:                                  ;SET CONTROL CHARACTERS
20 20 20 20 20 54 20 20 20 20 59 20   001D          .ASCII  / Y    T                          /
20 20 20 20 20 20 20 20 20 20 20 20   0029
                             20 20    0035
```

```
                     0037          .SBTTL  SET USER IDENTIFICATION CODE
                     0037     ;+
                     0037     ; DCL$SETUIC - SET USER IDENTIFICATION CODE
                     0037     ;
                     0037     ; THIS ROUTINE IS CALLED AS AN INTERNAL COMMAND TO EXECUTE THE SET USER
                     0037     ; IDENTIFICATION CODE DCLS COMMAND.
                     0037     ;
                     0037     ; INPUTS:
                     0037     ;
                     0037     ;       R8 = ADDRESS OF SCRATCH BUFFER DESCRIPTOR.
                     0037     ;       R9 = ADDRESS OF SCRATCH STACK.
                     0037     ;       R10 = BASE ADDRESS OF COMMAND WORK AREA.
                     0037     ;       R11 = BASE ADDRESS OF PROCESS WORK AREA.
                     0037     ;
                     0037     ; OUTPUTS:
                     0037     ;
                     0037     ;       THE CURRENT USER IDENTIFICATION CODE IS ESTABLISHED.
                     0037     ;-
                     0037
                     0037     DCL$SETUIC::                                    ;SET USER IDENTIFICATION CODE
            0C   C0  0037             ADDL    #PTR_C_LENGTH,-                 ;SKIP OPTION DESCRIPTOR
         BA AA       0039                     WRK_L_RSLNXT(R10)               ;
         FFC2'   30  003B             BSBW    DCL$GETDVAL                     ;GET THE VALUE OF THE TOKEN DESCRIPTOR
                     003E
                     003E     ;
                     003E     ; TRANSLATE THE OVERALL STRING.
                     003E     ;
      79    51   7D  003E             MOVQ    R1,-(R9)                        ;PUSH DESCRIPTOR INTO SCRATCH STACK
            52   DD  0041             PUSHL   R2                              ;ADDRESS OF STRING IN BUFFER
            3F   DD  0043             PUSHL   #63                             ;MAXIMUM STRING TO RETURN
      50    5E   D0  0045             MOVL    SP,R0                           ;GET ADDRESS OF OUTPUT DESCRIPTOR
                     0048             $TRNLOG_S (R9),(R0),(R0)                ;TRANSLATE THE NAME
      54    8E   7D  005B             MOVQ    (SP)+,R4                        ;RESET THE RESULTANT STRING DESCRIPTOR
                     005E
                     005E     ;
                     005E     ; SKIP PAST NODE AND DEVICE NAMES.  FIND START OF DIRECTORY SPECIFICATION.
                     005E     ;
          6544   94  005E             CLRB    (R5)[R4]                        ;MARK THE END OF STRING
  65  54    3A   3A  0061     10$:    LOCC    #^A/:/,R4,(R5)                  ;LOOK FOR DEVICE NAME DELIMITER
            13   13  0065             BEQL    20$                             ;BRANCH IF NO DEVICE HERE
  54  50    01   C3  0067             SUBL3   #1,R0,R4                        ;SKIP PAST DEVICE NAME
    55   01 A1   9E  006B             MOVAB   1(R1),R5                        ;
      61    81   91  006F             CMPB    (R1)+,(R1)                      ;IS THIS A NODE NAME?
            06   12  0072             BNEQ    20$                             ;BR IF ONLY DEVICE
            54   D7  0074             DECL    R4                              ;SKIP PAST SECOND COLON
            55   D6  0076             INCL    R5                              ;
            E7   11  0078             BRB     10$                             ;LOOK FOR MORE NODES OR DEVICE
                     007A
                     007A     ;
                     007A     ; CONVERT THE DIRECTORY STRING TO A UIC.
                     007A     ;
            29   10  007A     20$:    BSBB    DCL$CVTUIC                      ;GET THE UIC
         0F 50   E9  007C             BLBC    R0,90$                          ;BRANCH IF ERROR
      79    51   D0  007F             MOVL    R1,-(R9)                        ;SAVE LONGWORD UIC
                     0082             $CMKRNL_S B^SETUIC,(R9)                 ;SET USER IDENTIFICATION CODE
                 05  008E     90$:    RSB                                     ;RETURN WITH STATUS
                     008F
```

```
                                   008F    187 ;
                                   008F    188 ; SET USER IDENTIFICATION CODE
                                   008F    189 ;
                             0000  008F    190 SETUIC: .WORD    0                          ;ENTRY MASK
   50    00000000'9F    D0   0091    191           MOVL     @#SCH$GL_CURPCB,R0         ;GET CURRENT PROCESS PCB ADDRESS
      00BC C0     6C    D0   0098    192           MOVL     (AP),PCB$L_UIC(R0)         ;SET USER IDENTIFICATION CODE
                           009D    193           STATUS   NORMAL                     ;
                       04  00A4    194           RET                                 ;
```

```
                         00A5       .SBTTL  CONVERT STRING TO LONGWORD UIC
                         00A5  ;+
                         00A5  ; DCL$CVTUIC - CONVERT STRING TO LONGWORD UIC.
                         00A5  ;
                         00A5  ; INPUTS:
                         00A5  ;
                         00A5  ;            R4/R5 = DESCRIPTOR OF UIC STRING
                         00A5  ;
                         00A5  ; OUTPUTS:
                         00A5  ;
                         00A5  ;            R0 = STATUS
                         00A5  ;            R1 = LONGWORD UIC
                         00A5  ;            R2-R5 ARE TRASHED
                         00A5  ;-
                         00A5  DCL$CVTUIC::
               54    D7  00A5           DECL    R4                      ;SKIP LEADING BRACKET
               55    D6  00A7           INCL    R5                      ;
         7E    54    7D  00A9           MOVQ    R4,-(SP)                ;SAVE DIRECTORY DESCRIPTOR
               7E    D4  00AC           CLRL    -(SP)                   ;ALLOCATE LONGWORD FOR UIC
   FF A5    5B 8F    91  00AE           CMPB    #^A/[/,-1(R5)           ;START WITH A BRACKET?
               06    13  00B3           BEQL    10$                     ;IF EQL YES
      FF A5    3C    91  00B5           CMPB    #^A/</,-1(R5)           ;START WITH A BRACKET?
               2E    12  00B9           BNEQ    90$                     ;IF NEQ NO
         53    5E    D0  00BB  10$:     MOVL    SP,R3                   ;SAVE ADDRESS OF UIC LONGWORD
             0082    30  00BE           BSBW    CVTUIC                  ;CONVERT GROUP NUMBER
         85    2C    91  00C1           CMPB    #^A/,/,(R5)+            ;END WITH A COMMA?
               2E    12  00C4           BNEQ    50$                     ;IF NEQ NO
      02 A3    50    B0  00C6           MOVW    R0,2(R3)                ;SAVE GROUP NUMBER
             0076    30  00CA           BSBW    CVTUIC                  ;CONVERT MEMBER NUMBER
      65    5D 8F    91  00CD           CMPB    #^A/]/,(R5)             ;END WITH A BRACKET?
               05    13  00D1           BEQL    20$                     ;IF EQL YES
         65    3E    91  00D3           CMPB    #^A/>/,(R5)             ;END WITH A BRACKET?
               1C    12  00D6           BNEQ    50$                     ;IF NEQ NO
         63    50    B0  00D8  20$:     MOVW    R0,(R3)                 ;SAVE MEMBER NUMBER
               51  8ED0  00DB  30$:     POPL    R1                      ;GET UIC NUMBER
         5E    08    C0  00DE           ADDL    #8,SP                   ;POP UIC DESCRIPTOR
                         00E1           STATUS  NORMAL                  ;RETURN SUCCESS
                     05  00E8           RSB                             ;
                         00E9
                         00E9  ;
                         00E9  ; SIGNAL INVALID UIC SYNTAX
                         00E9  ;
                         00E9  90$:     STATUS  INVUIC                  ;SET INVALID UIC SYNTAX
         5E    0C    C0  00F0  95$:     ADDL    #12,SP                  ;RESTORE THE STACK
                     05  00F3           RSB                             ;
                         00F4
                         00F4  ;
                         00F4  ; TAKE UIC APART AND TRY TO CONVERT IT USING $ASCTOID.
                         00F4  ;
   54    04 A3       7D  00F4  50$:     MOVQ    4(R3),R4                ;GET UIC DESCRPITOR
   65    54    2C    3A  00F8           LOCC    #^A/,/,R4,(R5)          ;LOOK FOR A COMMA
               20    13  00FC           BEQL    60$                     ;BRANCH IF NONE
      04 A3    50    C2  00FE           SUBL    R0,4(R3)                ;GET LENGTH OF GROUP NAME
               50    D7  0102           DECL    R0                      ;CREATE DESCRIPTOR OF REST OF UIC
               51    D6  0104           INCL    R1                      ;
         54    50    7D  0106           MOVQ    R0,R4                   ;SAVE DESCRIPTOR OF REST OF UIC
                         0109           $ASCTOID_S  NAME=4(R3),-        ;GET THE GROUP ID
```

```
                       0109                     ID=(R3)                  ;
          D6 50    E9  0117            BLBC     R0,95$                   ;BRANCH IF ERROR
    04 A3    54    7D  011A            MOVQ     R4,4(R3)                 ;SAVE DESCRIPTOR OF REST OF UIC
                       011E
65   54   5D 8F    3A  011E  60$:      LOCC     #^A/]/,R4,(R5)           ;LOOK FOR A CLOSING BRACKET
             06    12  0123            BNEQ     65$                      ;BRANCH IF FOUND
   65   54   3E    3A  0125            LOCC     #^A/>/,R4,(R5)           ;LOOK FOR A CLOSING BRACKET
             BE    13  0129            BEQL     90$                      ;BRANCH IF NONE
    04 A3    50    C2  012B  65$:      SUBL     R0,4(R3)                 ;GET LENGTH OF MEMBER NAME
                       012F            $ASCTOID_S  NAME=4(R3),-          ;GET THE UIC
                       012F                     ID=(R3)                  ;
          B0 50    E9  013D            BLBC     R0,95$                   ;BRANCH IF ERROR
           FF98    31  0140            BRW      30$                      ;SET THE UIC
                       0143
                       0143      ;
                       0143      ; CONVERT ASCII OCTAL UIC COMPONENT TO NUMERIC WORD
                       0143      ;
             50    7C  0143  CVTUIC: CLRQ     R0                       ;CLEAR ACCUMULATION AND CHARACTER
   51   65   30    83  0145  10$:      SUBB3    #^A/0/,(R5),R1           ;GET NEXT CHARACTER
             0D    19  0149            BLSS     20$                      ;IF LSS NOT DIGIT
        51   08    D1  014B            CMPL     #8,R1                    ;OCTAL DIGIT?
             08    15  014E            BLEQ     20$                      ;IF LEQ NO
     50    6140    7E  0150            MOVAQ    (R1)[R0],R0              ;ACCUMULATE RESULT
             55    D6  0154            INCL     R5                       ;POINT TO NEXT CHARACTER
             ED    11  0156            BRB      10$                      ;
             05    05  0158  20$:      RSB                               ;
                       0159
                       0159          .SBTTL  SET DEFAULT DEVICE AND/OR DIRECTORY
                       0159  ;+
                       0159  ; DCL$SETDEFALT - SET DEFAULT DEVICE AND/OR DIRECTORY
                       0159  ;
                       0159  ; THIS ROUTINE IS CALLED AS AN INTERNAL COMMAND TO EXECUTE THE SET DEFAULT
                       0159  ; DCLS COMMAND.
                       0159  ;
                       0159  ; INPUTS:
                       0159  ;
                       0159  ;       R8 = ADDRESS OF SCRATCH BUFFER DESCRIPTOR.
                       0159  ;       R9 = ADDRESS OF SCRATCH STACK.
                       0159  ;       R10 = BASE ADDRESS OF COMMAND WORK AREA.
                       0159  ;       R11 = BASE ADDRESS OF PROCESS WORK AREA.
                       0159  ;
                       0159  ; OUTPUTS:
                       0159  ;
                       0159  ;       R4,R5 = STRING DESCRIPTOR FOR DIRECTORY PORTION
                       0159  ;       SYS$DISK = DEFAULT DISK
                       0159  ;       THE CURRENT DEFAULT DIRECTORY IS ESTABLISHED.
                       0159  ;-
                       0159
          0000000A     0159  MAX_TRANS_LVLS = 10                          ; maximum translation levels allowed
                       0159
                       0159  ;
                       0159  ; LNM service buffer offsets from R8
                       0159  ;
                       0159
          00000000     0159  Q_LOGNAM        = 0                          ; Logical name descriptor
          00000008     0159  Q_TABLE         = 8                          ; Table name descriptor
          00000010     0159  L_ATTR          = 16                         ; Attributes longword
          00000014     0159  L_MAX_INDEX     = 20                         ; Max Index
          00000018     0159  W_STRING_LEN    = 24                         ; String length
          0000001C     0159  T_STRING_BUF    = 28                         ; String buffer
          0000011C     0159  S_XLT_BUF       = 284                        ; Output buffer size
                       0159
                       0159
                       0159  DCL$SETDEFALT::                              ; SET DEFAULT
             0C    C0  0159          ADDL2   #PTR_C_LENGTH,-              ; skip option descriptor
          BA AA        015B                  WRK_L_RSLNXT(R10)            ;
          FEA0'    30  015D          BSBW    DCL$GETDVAL                  ; <R1,R2> = token
                       0160
                       0160  ;
                       0160  ; Translate the overall string to get 1 level of translation
                       0160  ;
                       0160
          5C    0A    D0  0160       MOVL    #MAX_TRANS_LVLS,AP           ; set max translation counter
59  0000011C    8F    C2  0163       SUBL2   #S_XLT_BUF,R9                ; allocate buffer
          58    59    D0  016A       MOVL    R9,R8                        ; save addr of buffer
                          016D
                          016D  ;
                          016D  ; Create item list for $TRNLNM
                          016D  ;
                          016D
          79    7C        016D       CLRQ    -(R9)                        ; clear last longword and length addr
     79   10    A8    DE  016F       MOVAL   L_ATTR(R8),-(R9)             ; set up attributes item
    00030004    8F    D0  0173       MOVL    #^<LNM$_ATTRIBUTES@16>+4,-   ;
          79              0179               -(R9)                        ;
```

```
      79   18 A8   3E  017A          MOVAW   W_STRING_LEN(R8),-(R9)  ; string size goes here
      79   1C A8   9E  017E          MOVAB   T_STRING_BUF(R8),-(R9)  ; string buffer
   000200FF   8F   D0  0182          MOVL    #<LNM$_STRING@16>+255,- ;
              79       0188                  -(R9)                   ;
              79   D4  0189          CLRL    -(R9)                   ; no output size
      79   14 A8   DE  018B          MOVAL   L_MAX_INDEX(R8),-(R9)   ; max index here
   00070004   8F   D0  018F          MOVL    #<LNM$_MAX_INDEX@16>+4,-;
              79       0195                  -(R9)                   ;
      08 A8   0C   9A  0196          MOVZBL  #TABNAMSZ,Q_TABLE(R8)   ; create descriptor of logical name
 0C A8   FE6A CF   9E  019A          MOVAB   TABNAM,Q_TABLE+4(R8)    ;  table to look in
         68   51   7D  01A0          MOVQ    R1,Q_LOGNAM(R8)         ; set up logical name
                       01A3          $TRNLNM_S -                     ; translate the logical name
                       01A3                  TABNAM=Q_TABLE(R8),-    ;
                       01A3                  LOGNAM=Q_LOGNAM(R8),-   ;
                       01A3                  ITMLST=(R9)             ;
    0000'8F   50   B1  01B5          CMPW    R0,#SS$_NORMAL          ; success?
              08   13  01BA          BEQL    10$                     ; yes
    0000'8F   50   B1  01BC          CMPW    R0,#SS$_NOLOGNAM        ; no translation?
              18   13  01C1          BEQL    15$                     ; yes
                   05  01C3          RSB                             ; error
                       01C4
                       01C4  ;
                       01C4  ; Check if there was a really a translation, was it a search list
                       01C4  ; and if it was a concealed device.
                       01C4  ;
                       01C4
         14 A8   D5  01C4  10$:    TSTL    L_MAX_INDEX(R8)              ; was there a real non-search list name
            17   14  01C7          BGTR    20$                          ; branch if >0, search list
            10   19  01C9          BLSS    15$                          ; branch if <0, null translation
            08   E0  01CB          BBS     #LNM$V_CONCEALED,-           ; ignore if translation concealed
      10 10 A8       01CD                  L_ATTR(R8),20$               ;
         18 A8   3C  01D0          MOVZWL  W_STRING_LEN(R8),-           ; set result string length
            68       01D3                  Q_LOGNAM(R8)                 ;
         18 A8   28  01D4          MOVC3   W_STRING_LEN(R8),-           ; copy translation into the buffer
         1C A8       01D7                  T_STRING_BUF(R8),-           ;  where the original token use to be
         04 B8       01D9                  @Q_LOGNAM+4(R8)              ;
      54    68   7D  01DB  15$:    MOVQ    Q_LOGNAM(R8),R4              ; setup string descriptor
            10   11  01DE          BRB     40$                          ; parse string
                     01E0
                     01E0  ;
                     01E0  ; We could not use the translation because of concealed name or search list
                     01E0  ; so use the original input string
                     01E0  ;
                     01E0
      54    68   7D  01E0  20$:    MOVQ    Q_LOGNAM(R8),R4              ; get source descriptor
                     01E3
                     01E3  ;
                     01E3  ; Make sure the last character is a ":" so it acts like a device name
                     01E3  ;
                     01E3
 3A    FF A544   91  01E3          CMPB    -1(R5)[R4],#^A':'            ; is last char a colon?
            06   13  01E8          BEQL    40$                          ; continue if so
   6544     3A   90  01EA          MOVB    #^A':',(R5)[R4]              ; append a colon if not
            54   D6  01EE          INCL    R4                           ; count it as well
                     01F0
                     01F0  ;
                     01F0  ; Locate the device portion of the string, include any node names found as well
```

```
                        01F0  ;
                        01F0
        6544     94     01F0  40$:    CLRB    (R5)[R4]                ; mark end of string
65   54   3A     3A     01F3          LOCC    #^A/:/,R4,(R5)          ; look for device name delimiter
          3D     13     01F7          BEQL    70$                     ; branch if no device here
     61   81     91     01F9          CMPB    (R1)+,(R1)              ; is this a node name?
          14     12     01FC          BNEQ    60$                     ; branch if only device
  53   01 A1     9E     01FE          MOVAB   1(R1),R3                ; set address of end of node string
     50   02     C2     0202          SUBL    #2,R0                   ;  and length of remainder
63   50   3A     3A     0205          LOCC    #^A/:/,R0,(R3)          ; see if device name is here
          04     13     0209          BEQL    50$                     ; branch if none, just use node
  53   01 A1     9E     020B          MOVAB   1(R1),R3                ; set end of device name
     51   53     D0     020F  50$:    MOVL    R3,R1                   ; set end of equivalence name for disk
     52   55     D0     0212  60$:    MOVL    R5,R2                   ; save start of device string
     55   51     D0     0215          MOVL    R1,R5                   ; set start of directory string
     51   52     C2     0218          SUBL    R2,R1                   ; find length of device name
     54   51     C2     021B          SUBL    R1,R4                   ; adjust directory string length
                        021E
                        021E  ;
                        021E  ; At this point: <R1,R2> = device (+node)
                        021E  ;                <R4,R5> = rest of string
                        021E  ;
                        021E  ; Check if the device portion = 'SYS$DISK', if so ignore it
                        021E  ;
                        021E
57   FDF2 CF     9E     021E          MOVAB   DCL$T_DSKNAM,R7         ; address of device name counted string
     56   87     9A     0223          MOVZBL  (R7)+,R6                ; get length and address of first byte
50   51   56     C3     0226          SUBL3   R6,R1,R0                ; find difference in name string sizes
          50     D7     022A          DECL    R0                      ; check if 1 byte difference(the colon!)
          0D     12     022C          BNEQ    80$                     ; br if no-can't be the special name
          06     BB     022E          PUSHR   #^M<R1,R2>              ; save registers to be used
67   62   56     29     0230          CMPC3   R6,(R2),(R7)            ; check for reserved system name
          06     BA     0234          POPR    #^M<R1,R2>              ; restore values
          03     12     0236  70$:    BNEQ    80$                     ; branch if no device name assignment
        007E     31     0238          BRW     130$                    ;  needed
                        023B
                        023B  ;
                        023B  ; If the device portion has a translation and it contains a
                        023B  ; directory specification, then repeat using the translation
                        023B  ; if a directory was specified in addition, then report an error
                        023B  ; that 2 directory specifications appeared in the same string
                        023B  ;
                        023B
          51     D7     023B  80$:    DECL    R1                      ; do not send colon into trnlnm
     68   51     7D     023D          MOVQ    R1,Q_LOGNAM(R8)         ; set up logical name
                        0240          $TRNLNM_S -                     ; translate the logical name
                        0240                  TABNAM=Q_TABLE(R8),-    ;
                        0240                  LOGNAM=Q_LOGNAM(R8),-   ;
                        0240                  ITMLST=(R9)             ;
0000'8F   50     B1     0252          CMPW    R0,#SS$_NORMAL          ; success?
          08     13     0257          BEQL    90$                     ; yes
0000'8F   50     B1     0259          CMPW    R0,#SS$_NOLOGNAM        ; no translation?
          3C     13     025E          BEQL    120$                    ; yes
          05            0260          RSB                             ; error
                        0261
       14 A8     D5     0261  90$:    TSTL    L_MAX_INDEX(R8)         ; branch if no translation or
          36     12     0264          BNEQ    120$                    ;  search list
```

```
                 08   E0  0266   452          BBS     #LNM$V_CONCEALED,-        ;  or concealed
              31 10 A8    0268   453                  L_ATTR(R8),120$
        18 A8  5B 8F  3A  026B   454          LOCC    #^A/[/,W_STRING_LEN(R8),-; is there a directory in there?
                 1C A8    0270   455                  T_STRING_BUF(R8)          ;
                    08 12 0272   456          BNEQ    95$                       ; ignore unless device/dir translation
         18 A8   3C   3A  0274   457          LOCC    #^A/</,W_STRING_LEN(R8),-; is there a directory in there?
                 1C A8    0278   458                  T_STRING_BUF(R8)          ;
                    20 13 027A   459          BEQL    120$                      ; ignore unless device/dir translation
                    54 D5 027C   460 95$:     TSTL    R4                        ; any directory specified explicitly?
                    11 12 027E   461          BNEQ    100$                      ; if so, then error in specification
                          0280   462
              18 A8   3C  0280   463          MOVZWL  W_STRING_LEN(R8),-        ; set result string length
                 68       0283   464                  Q_LOGNAM(R8)              ;
              18 A8   28  0284   465          MOVC3   W_STRING_LEN(R8),-        ; copy translation into the buffer
              1C A8       0287   466                  T_STRING_BUF(R8),-        ;  where the original token use to be
              04 B8       0289   467                  @Q_LOGNAM+4(R8)           ;
           54    68   7D  028B   468          MOVQ    Q_LOGNAM(R8),R4           ; setup string descriptor
              08 5C   F5  028E   469          SOBGTR  AP,110$                   ; limit translation levels
                          0291   470 100$:    STATUS  DIRECT                    ; error in directory specification
                      05  0298   471          RSB
              FF54    31  0299   472 110$:    BRW     40$                       ; continue translat.on device portion
                          029C   473
           51    68   7D  029C   474 120$:    MOVQ    Q_LOGNAM(R8),R1           ; restore device portion descriptor
                 51   D6  029F   475          INCL    R1                        ; restore colon to end of string
                          02A1   476
                          02A1   477 ;
                          02A1   478 ; Create/update the logical name sys$disk which holds the current
                          02A1   479 ; default disk device.
                          02A1   480 ;
                          02A1   481
          00C6 8F     BB  02A1   482          PUSHR   #^M<R1,R2,R6,R7>          ; descriptors for logical and equivalence na
                 00   DD  02A5   483          PUSHL   #0                        ; access mode is defaulted
              04 AE   7F  02A7   484          PUSHAQ  4(SP)                     ; address of equivalence name desc
              10 AE   7F  02AA   485          PUSHAQ  16(SP)                    ; descriptor of name to relate with
                 02   DD  02AD   486          PUSHL   #LOG$C_PROCESS            ; table number
      00000000'9F 08  FB  02AF   487          CALLS   #8,@#SYS$CRELOG           ; clear descriptor on return
              1C 50   E9  02B6   488          BLBC    R0,150$                   ; branch if error creating name
                          02B9   489
                          02B9   490 ;
                          02B9   491 ; Change the default directory specification (if any);
                          02B9   492 ;
                          02B9   493
                 54   D5  02B9   494 130$:    TSTL    R4                        ; any directory field
                 11   13  02BB   495          BEQL    140$                      ; branch if no
                 30   BB  02BD   496          PUSHR   #^M<R4,R5>                ; descriptor for directory name
                 7E   7C  02BF   497          CLRQ    -(SP)                     ; zeros as arguments 2 & 3
              08 AE   9F  02C1   498          PUSHAB  8(SP)                     ; address of directory string
      00000000'GF 05  FB  02C4   499          CALLS   #5,G^SYS$SETDDIR          ; set the default directory
              07 50   E9  02CB   500          BLBC    R0,150$                   ; branch if error from rms
                          02CE   501 140$:    STATUS  NORMAL                    ; assume all is aok
                      05  02D5   502 150$:    RSB
                          02D6   503
```

```
                              02D6          .SBTTL  SET PROTECTION
                              02D6  ;+
                              02D6  ; DCL$SETPROT - SET PROTECTION
                              02D6  ;
                              02D6  ; THIS ROUTINE IS CALLED AS AN INTERNAL COMMAND TO EXECUTE THE SET PROTECTION
                              02D6  ; DCLS COMMAND.
                              02D6  ;
                              02D6  ; INPUTS:
                              02D6  ;
                              02D6  ;         R8 = ADDRESS OF SCRATCH BUFFER DESCRIPTOR.
                              02D6  ;         R9 = ADDRESS OF SCRATCH STACK.
                              02D6  ;         R10 = BASE ADDRESS OF COMMAND WORK AREA.
                              02D6  ;         R11 = BASE ADDRESS OF PROCESS WORK AREA.
                              02D6  ;
                              02D6  ; OUTPUTS:
                              02D6  ;
                              02D6  ;         THE CURRENT DEFAULT PROTECTION IS ESTABLISHED.
                              02D6  ;-
                              02D6
                              02D6  DCL$SETPROT::                                                ;SET PROTECTION
                     7E   D4  02D6          CLRL    -(SP)                                   ;WHERE TO RETURN PROTECTION
               7E    5E   D0  02D8          MOVL    SP,-(SP)                                ;NOTE WHERE PROTECTION IS TO BE PUT
                     7E   D4  02DB          CLRL    -(SP)                                   ;DON'T WANT TO SET PROTECTION
     00000000'9F     02   FB  02DD          CALLS   #2,@#SYS$SETDFPROT                      ;GET DEFAULT PROTECTION
               59    8E   D0  02E4          MOVL    (SP)+,R9                                ;COPY PROTECTION TO USEFUL REG
                     18   C0  02E7          ADDL    #2*PTR_C_LENGTH,-                       ;SKIP PAST OPTION DESCRIPTOR
                  BA AA       02E9                  WRK_L_RS[NXT(R10)                       ; AND /DEFAULT QUALIFIER
                  FD12'   30  02EB  10$:    BSBW    DCL$GETDVAL                             ;GET NEXT DESCRIPTOR VALUES
               55    03   91  02EE          CMPB    #PTR_K_PARAMETR,R5                      ;PARAMETER VALUE?
                     34   12  02F1          BNEQ    40$                                     ;IF NEQ NO
 FD0B CF       04    62   3A  02F3          LOCC    (R2),#4,CLASS                           ;LOCATE PROTECTION CLASS
                     43   13  02F9          BEQL    60$                                     ;IF EQL INVALID CLASS
                     50   D7  02FB          DECL    R0                                      ;CALCULATE STARTING BIT NUMBER
          58   50    04   C5  02FD          MULL3   #4,R0,R8
 59       04   58    0F   F0  0301          INSV    #^XF,R8,#4,R9                           ;START WITH NO ACCESS
               54    02   91  0306          CMPB    #PTR_K_COLON,R4                         ;PROTECTION VALUE SPECIFIED?
                     E0   12  0309          BNEQ    10$                                     ;IF NEQ NO
                  FCF2'   30  030B          BSBW    DCL$GETDVAL                             ;GET PROTECTION VALUE DESCRIPTOR
               57    51   D0  030E          MOVL    R1,R7                                   ;SAVE LENGTH OF VALUE STRING
 FCE9 CF       04    82   3A  0311  20$:    LOCC    (R2)+,#4,ACCESS                         ;LOCATE PROTECTION CODE
                     1D   13  0317          BEQL    50$                                     ;IF EQL INVALID PROTECTION CODE
                     50   D7  0319          DECL    R0                                      ;CALCULATE RELATIVE BIT NUMBER IN FIELD
               50    58   C0  031B          ADDL    R8,R0                                   ;CALCULATE ACTUAL BIT NUMBER
            00 59    50   E5  031E          BBCC    R0,R9,30$                               ;ALLOW SPECIFIED ACCESS
                  EC 57   F5  0322  30$:    SOBGTR  R7,20$                                  ;ANY MORE TO SCAN?
                     C4   11  0325          BRB     10$
                     59   DD  0327  40$:    PUSHL   R9                                      ;SET NEW DEFAULT PROTECTION ARGUMENT
                     7E   D4  0329          CLRL    -(SP)                                   ;ZERO ADDRESS OF RETURN DESCRIPTOR
                  04 AE   DF  032B          PUSHAL  4(SP)                                   ;ADDRESS OF NEW PROTECTION
     00000000'9F     03   FB  032E          CALLS   #3,@#SYS$SETDFPROT                      ;SET DEFAULT PROTECTION
                          05  0335          RSB                                             ;
                              0336  50$:    STATUS  IVPROT                                  ;SET INVALID PROCTECTION CODE
                          05  033D          RSB                                             ;
                              033E  60$:    STATUS  IVKEYW                                  ;SET INVALID KEYWORD
                          05  0345          RSB                                             ;
```

```
                              0346   561            .SBTTL  SET VERIFY MODE
                              0346   562    ;+
                              0346   563    ; DCL$SETVERIFY - SET VERIFY MODE
                              0346   564    ;
                              0346   565    ; THIS ROUTINE IS CALLED AS AN INTERNAL COMMAND TO EXECUTE THE SET VERIFY
                              0346   566    ; MODE DCLS COMMAND.
                              0346   567    ;
                              0346   568    ; INPUTS:
                              0346   569    ;
                              0346   570    ;       R8 = ADDRESS OF SCRATCH BUFFER DESCRIPTOR.
                              0346   571    ;       R9 = ADDRESS OF SCRATCH STACK.
                              0346   572    ;       R10 = BASE ADDRESS OF COMMAND WORK AREA.
                              0346   573    ;       R11 = BASE ADDRESS OF PROCESS WORK AREA.
                              0346   574    ;
                              0346   575    ; OUTPUTS:
                              0346   576    ;
                              0346   577    ;       THE VERIFY MODE IS ESTABLISHED.
                              0346   578    ;-
                              0346   579
                              0346   580    DCL$SETVERIFY::                                         ;SET VERIFY MODE
                              0346   581
                              0346   582    ;
                              0346   583    ; PARSE THE COMMAND.
                              0346   584    ;
           FCB7'      30      0346   585            BSBW    DCL$GETDVAL                     ;GET OPTION DESCRIPTOR
         04 53        E9      0349   586            BLBC    R3,10$                          ;IF LBC VERIFICATION SPECIFIED
            56        D4      034C   587            CLRL    R6                              ;DISABLE ALL VERIFICATION
            3F        11      034E   588            BRB     40$                             ;IGNORE ANY KEYWORDS
      56    03        D0      0350   589    10$:    MOVL    #3,R6                           ;ASSUME ALL VERIFICATION IS SPECIFIE
          FCAA'       30      0353   590            BSBW    DCL$GETDVAL                     ;GET KEYWORD DESCRIPTOR
      55    04        D1      0356   591            CMPL    #PTR_K_ENDLINE,R5               ;EOL?
            34        13      0359   592            BEQL    40$                             ;YES, THEN SET SPECIFIED MODES
      56    0F        D0      035B   593            MOVL    #15,R6                          ;ASSUME NO KEYWORDS ARE SPECIFIED
   62    50 8F        91      035E   594    20$:    CMPB    #^A/P/,(R2)                     ;IS FIRST CHAR ''P''?
            07        13      0362   595            BEQL    25$                             ;YES, THEN PROCESS ''PROCEDURE''
02 A2    50 8F        91      0364   596            CMPB    #^A/P/,2(R2)                    ;IS THIRD CHAR ''P''?
            0E        12      0369   597            BNEQ    30$                             ;NO, THEN PROCESS ''IMAGE''
      56    08        CA      036B   598    25$:    BICL    #8,R6                           ;INDICATE ''PROCEDURE'' SEEN
      56    02        C8      036E   599            BISL    #2,R6                           ;ENABLE PROCEDURE VERIFICATION
         11 53        E9      0371   600            BLBC    R3,35$                          ;IF LBC PROCEDURE VERIFY SPECIFIED
      56    02        CA      0374   601            BICL    #2,R6                           ;DISABLE PROCEDURE VERIFICATION
            0C        11      0377   602            BRB     35$                             ;GET NEXT
      56    04        CA      0379   603    30$:    BICL    #4,R6                           ;INDICATE ''IMAGE'' SEEN
      56    01        C8      037C   604            BISL    #1,R6                           ;ENABLE IMAGE VERIFICATION
         03 53        E9      037F   605            BLBC    R3,35$                          ;IF LBC IMAGE VERIFY SPECIFIED
      56    01        CA      0382   606            BICL    #1,R6                           ;DISABLE IMAGE VERIFICATION
          FC78'       30      0385   607    35$:    BSBW    DCL$GETDVAL                     ;GET KEYWORD DESCRIPTOR
      55    04        D1      0388   608            CMPL    #PTR_K_ENDLINE,R5               ;EOL?
            02        13      038B   609            BEQL    40$                             ;YES, THEN SET SPECIFIED MODES
            CF        11      038D   610            BRB     20$                             ;GET NEXT
                              038F   611
                              038F   612    ;
                              038F   613    ; UPDATE PROCEDURE VERIFICATION STATE.
                              038F   614    ;
      10 56   03      E0      038F   615    40$:    BBS     #3,R6,50$                       ;BRANCH IF ''PROC'' NOT SPECIFIED
68 AB  0080 8F        A8      0393   616            BISW    #PRC_M_VERIFY,PRC_W_FLAGS(R11)  ;ASSUME VERIFICATION IS SPECIFIED
      06 56   01      E0      0399   617            BBS     #1,R6,50$                       ;BRANCH IF SO
```

```
        68 AB   0080 8F    AA  039D     618          BICW    #PRC_M_VERIFY,PRC_W_FLAGS(R11)  ;DISABLE VERIFICATION
              08 56   02    E1  03A3     619 50$:     BBC     #2,R6,60$                      ;BRANCH IF "IMAGE" SPECIFIED
                                03A7     620          STATUS  NORMAL                         ;SET STATUS
                            05  03AE     621          RSB                                    ;RETURN
```

```
                         03AF      .SBTTL  SET IMAGE VERIFY MODE
                         03AF ;+
                         03AF ; DCL$SETVERIFY_IMAGE - SET IMAGE VERIFY MODE
                         03AF ;
                         03AF ; THIS ROUTINE IS CALLED TO SET IMAGE VERIFY MODE.
                         03AF ;
                         03AF ; INPUTS:
                         03AF ;
                         03AF ;         R6 = IMAGE VERIFY FLAGS, LBC MEANS CLEAR, LBS MEANS SET
                         03AF ;         R11 = BASE ADDRESS OF PROCESS WORK AREA.
                         03AF ;
                         03AF ; OUTPUTS:
                         03AF ;
                         03AF ;         THE IMAGE VERIFY MODE IS ESTABLISHED.
                         03AF ;-
                         03AF
                         03AF 60$:
                         03AF DCL$SETVERIFY_IMAGE::                                        ;SET IMAGE VERIFY MODE
                         03AF ;
                         03AF ; GET INPUT STREAM INFORMATION.
                         03AF ;
         51   1C AB  D0  03AF         MOVL    PRC_L_INDFAB(R11),R1                 ;GET ADDRESS OF GENERIC FAB
         52 00BC CB  D0  03B3         MOVL    PRC_L_IDFLNK(R11),R2                 ;GET ADDR OF CURRENT IND FRAME
    0000'C1   04 A2  B0  03B8         MOVW    IDF_W_INPIFI(R2),FAB$W_IFI(R1)       ;GET INPUT IFI
                         03BE
                         03BE ;
                         03BE ; UPDATE IMAGE VERIFICATION STATE BOTH IN PRC AND FOR CURRENT INPUT STREAM.
                         03BE ;
           0F 56     E9  03BE         BLBC    R6,70$                               ;BRANCH IF /NOIMAGE
              07     E0  03C1         BBS     #PRC_V_VERIMAGE,-                    ;IF IMAGE VERIFY ALREADY SET,
         1B 00AF CB      03C3                 PRC_B_FLAGS2(R11),90$                ;  THEN DONE
 00AF CB   0080 8F   A8  03C7         BISW    #PRC_M_VERIMAGE,PRC_B_FLAGS2(R11);ENABLE IMAGE VERIFICATION
              0D     11  03CE         BRB     80$                                  ;EXECUTE $MODIFY
                         03D0
              07     E1  03D0 70$:    BBC     #PRC_V_VERIMAGE,-                    ;IF IMAGE VERIFY ALREADY CLEAR,
         0C 00AF CB      03D2                 PRC_B_FLAGS2(R11),90$                ;  THEN DONE
 00AF CB   0080 8F   AA  03D6         BICW    #PRC_M_VERIMAGE,PRC_B_FLAGS2(R11);DISABLE IMAGE VERIFICATION
                         03DD
              0B     10  03DD 80$:    BSBB    DCL$VERIFY_IMAGE                     ;ENABLE OR DISABLE VERIFICATION
           07 50     E9  03DF         BLBC    R0,95$                               ;RETURN ERROR STATUS
                         03E2
                         03E2 90$:    STATUS  NORMAL                               ;RETURN SUCCESS
                     05  03E9 95$:    RSB                                          ;
```

```
                                 03EA    667         .SBTTL  MODIFY INPUT STREAM CHARACTERISTICS
                                 03EA    668 ;++
                                 03EA    669 ; DCL$VERIFY_IMAGE - MODIFY THE INPUT STREAM CHARACTERISTICS.
                                 03EA    670 ;
                                 03EA    671 ; INPUTS:
                                 03EA    672 ;
                                 03EA    673 ;          R1 = INPUT FAB
                                 03EA    674 ;          R11 = ADDRESS OF PRC DATA STRUCTURE
                                 03EA    675 ;
                                 03EA    676 ; OUTPUTS:
                                 03EA    677 ;
                                 03EA    678 ;          R0 = STATUS
                                 03EA    679 ;--
                                 03EA    680
                                 03EA    681 DCL$VERIFY_IMAGE::
              012D CB      95    03EA    682            TSTB    PRC_B_EXONLYL(R11)                ;ARE WE IN EXE-ONLY MODE?
                    4B     12    03EE    683            BNEQ    90$                               ;YES, DON'T DO ANYTHING.
                                 03F0    684
          08 68 AB    06   E0    03F0    685            BBS     #PRC_V_MODE,PRC_W_FLAGS(R11),10$;BRANCH IF NOT INTERACTIVE
               50     01   D0    03F5    686            MOVL    #1,R0                             ;ASSUME SUCCESS
                 5C AB     D5    03F8    687            TSTL    PRC_L_INDEPTH(R11)                ;BRANCH IF LEVEL 0
                    3E     13    03FB    688            BEQL    90$                               ;
           0000'C1    02   B0    03FD    689 10$:       MOVW    #RME$C_PPFECHO,FAB$L_CTX(R1)      ;SET TYPE CODE
                0002'C1    B4    0402    690            CLRW    FAB$L_CTX+2(R1)                   ;ZERO ISI VALUE
                    07     E1    0406    691            BBC     #PRC_V_VERIMAGE,-                 ;IF IMAGE VERIFY SET,
             0B 00AF CB          0408    692                    PRC_B_FLAGS2(R11),20$             ;  THEN SET THE ISI
             50    18 AB   D0    040C    693            MOVL    PRC_L_INDOUTRAB(R11),R0           ;GET ADDR OF OUTPUT RAB
                0000'C0    B0    0410    694            MOVW    RAB$W_ISI(R0),-                   ;SET OUTPUT ISI
                0002'C1          0414    695                    FAB$L_CTX+2(R1)                   ;
                     51    DD    0417    696 20$:       PUSHL   R1                                ;SAVE R1
0000'C1     00000000'8F    C8    0419    697            BISL    #FAB$M_ESC,FAB$L_FOP(R1)          ;SET ESC BIT IN FOP
                                 0422    698            $MODIFY FAB=(R1)                          ;MODIFY THE INPUT STREAM
                  51 8ED0        042B    699            POPL    R1                                ;RESTORE R1
0000'C1     00000000'8F    CA    042E    700            BICL    #FAB$M_ESC,FAB$L_FOP(R1)          ;CLEAR ESC BIT IN FOP
                0000'C1    D4    0437    701            CLRL    FAB$L_CTX(R1)                     ;
                           05    043B    702 90$:       RSB                                       ;RETURN STATUS
```

```
                      043C            .SBTTL  SET ON MODE
                      043C      ;+
                      043C      ; DCL$SETON - SET ON MODE
                      043C      ;
                      043C      ; THIS ROUTINE IS CALLED AS AN INTERNAL COMMAND TO EXECUTE THE SET ON
                      043C      ; MODE DCLS COMMAND.
                      043C      ;
                      043C      ; INPUTS:
                      043C      ;
                      043C      ;         R8 = ADDRESS OF SCRATCH BUFFER DESCRIPTOR.
                      043C      ;         R9 = ADDRESS OF SCRATCH STACK.
                      043C      ;         R10 = BASE ADDRESS OF COMMAND WORK AREA.
                      043C      ;         R11 = BASE ADDRESS OF PROCESS WORK AREA.
                      043C      ;
                      043C      ; OUTPUTS:
                      043C      ;
                      043C      ;         THE ON MODE IS ESTABLISHED.
                      043C      ;-
                      043C
                      043C      DCL$SETON::                                      ;SET ON MODE
          FBC1'   30  043C            BSBW    DCL$GETDVAL                      ;GET THE DESCRIPTOR FOR "ON"
                      043F            STATUS  NORMAL                           ;SET NORMAL COMPLETION STATUS
 51    6A AB      9E  0446            MOVAB   PRC_W_ONLEVEL(R11),R1            ;GET ADDRESS OF ON LEVEL CODE
    61    08      91  044A            CMPB    #8,(R1)                          ;CHECK "ON" LEVEL FOR RESERVED LEVEL
       07 53      E8  044D            BLBS    R3,20$                           ;BR IF OPTION WAS NEGATED (NOON)
          04      14  0450            BGTR    10$                              ;BR IF "ON" ALREADY ACTIVE
 61    01 A1      90  0452            MOVB    1(R1),(R1)                       ;RESET TO SAVED VALUE
                  05  0456      10$:  RSB                                      ;
          07      13  0457      20$:  BEQL    30$                              ;BR IF "ON" ALREADY AT RESEVED LEVEL
 01 A1    61      90  0459            MOVB    (R1),1(R1)                       ;SAVE PREVIOUS "ON" LEVEL
    61    08      90  045D            MOVB    #8,(R1)                          ;SET TO RESERVED LEVEL
                  05  0460      30$:  RSB                                      ;END OF NOON HANDLING
```

```
                                0461              .SBTTL  SET CONTROL ENABLE/DISABLE
                                0461      ;+
                                0461      ; DCL$SETCTLY - SET CONTROL MODE
                                0461      ;
                                0461      ; THIS ROUTINE IS CALLED AS AN INTERNAL COMMAND TO EXECUTE THE SET CONTROL=KEY
                                0461      ; MODE DCLS COMMAND.
                                0461      ;
                                0461      ; INPUTS:
                                0461      ;
                                0461      ;       R8 = ADDRESS OF SCRATCH BUFFER DESCRIPTOR.
                                0461      ;       R9 = ADDRESS OF SCRATCH STACK.
                                0461      ;       R10 = BASE ADDRESS OF COMMAND WORK AREA.
                                0461      ;       R11 = BASE ADDRESS OF PROCESS WORK AREA.
                                0461      ;
                                0461      ; OUTPUTS:
                                0461      ;
                                0461      ;       CONTROL Y AND OUT-OF-BAND AST'S ARE ENABLED OR DISABLED FOR THIS
                                0461      ;       PROCESS.
                                0461      ;-
                                0461      DCL$SETCTLY::                                           ;SET CONTROL MODE
                      7E    D4  0461              CLRL    -(SP)                                   ;ALLOCATE CHAR MASK ON STACK
                   FB9A'    30  0463              BSBW    DCL$GETDVAL                             ;GET OPTION DESCRIPTOR
                                0466              ASSUME  PTR_V_NEGATE EQ 20
                56    53    D0  0466              MOVL    R3,R6                                   ;SAVE [NO] STATUS FOR FUTURE USE
                                0469
                   FB94'    30  0469              BSBW    DCL$GETDVAL                             ;GET FIRST LETTER
                04    55    91  046C              CMPB    R5,#PTR_K_ENDLINE                       ;END OF LINE?
                      04    12  046F              BNEQ    30$                                     ;IF YES, THEN ASSUME Y
                      3A    10  0471              BSBB    CTRLY                                   ;OTHERWISE, SET CONTROL_Y BY DEFAULT
                      2D    11  0473              BRB     80$                                     ;ALL DONE
                                0475
     FBA2 CF    1A    62    3A  0475      30$:    LOCC    (R2),#26,CONTROL_CHARS                  ;GET INDEX OF LETTER
             00 6E    50    E2  047B              BBSS    R0,(SP),40$                             ;SET CHAR BIT IN MASK
                   FB7E'    30  047F      40$:    BSBW    DCL$GETDVAL                             ;GET NEXT PARAMETER
                04    55    91  0482              CMPB    R5,#PTR_K_ENDLINE                       ;END OF LINE?
                      EE    12  0485              BNEQ    30$                                     ;LOOP IF NOT
                                0487
   51    00B4 CB      6E    C9  0487      50$:    BISL3   (SP),PRC_L_OUTOFBAND(R11),R1      ;GET CHARACTER MASK
                   0C 56    E9  048D              BLBC    R6,70$                                  ;IF LBC, THEN ENABLE SPECIFIED
             02 6E    19    E1  0490              BBC     #PRC_V_CTRLY,(SP),60$                   ;IF NOT CTRL/Y, THEN SKIP
                      17    10  0494              BSBB    CTRLY                                   ;DO SPECIAL CTRL/Y PROCESSING
   51    00B4 CB      6E    CB  0496      60$:    BICL3   (SP),PRC_L_OUTOFBAND(R11),R1          ;SET MASK FOR DISABLE
             00000000'EF    16  049C      70$:    JSB     DCL$RESETOOB                            ;ENABLE/DISABLE APPROPRIATE AST ROUTINES
                                04A2
                50    8E    D0  04A2      80$:    MOVL    (SP)+,R0                                ;RESTORE STACK
                                04A5              STATUS  NORMAL                                  ;SET NORMAL COMPLETION STATUS
                            05  04AC              RSB                                             ;
                                04AD
00B4 CB   02000000 8F       C8  04AD      CTRLY:  BISL    #PRC_M_CTRLY,PRC_L_OUTOFBAND(R11)   ;ASSUME ENABLE SPECIFIED
                   0C 56    E9  04B6              BLBC    R6,10$                                  ;IF LBC, THEN ENABLE SPECIFED
00B4 CB   02000000 8F       CA  04B9              BICL    #PRC_M_CTRLY,PRC_L_OUTOFBAND(R11)   ;CLEAR CTRL/Y BIT IN MASK
                   FB3B'    30  04C2              BSBW    W^DCL$ONCTLYRST                         ;RESET CONTROL Y COMMAND TEXT
                            05  04C5      10$:    RSB
                                04C6
                                04C6            .SBTTL  SET PROMPT
                                04C6      ;+
                                04C6      ; DCL$SETPROMPT - SET PROMPT
                                04C6      ;
                                04C6      ; THIS ROUTINE IS CALLED AS AN INTERNAL COMMAND TO EXECUTE THE SET PROMPT
                                04C6      ; DCLS COMMAND.
                                04C6      ;
                                04C6      ; INPUTS:
                                04C6      ;
                                04C6      ;         R8 = ADDRESS OF SCRATCH BUFFER DESCRIPTOR.
                                04C6      ;         R9 = ADDRESS OF SCRATCH STACK.
                                04C6      ;         R10 = BASE ADDRESS OF COMMAND WORK AREA.
                                04C6      ;         R11 = BASE ADDRESS OF PROCESS WORK AREA.
                                04C6      ;
                                04C6      ; OUTPUTS:
                                04C6      ;
                                04C6      ;         THE DCL PROMPT STRING IS CHANGED.
                                04C6      ;-
                                04C6      DCL$SETPROMPT::                                  ;SET PROMPT
 00F1 CB   00000000'EF     B0   04C6              MOVW    DCL$CRLF,PRC_W_PMPTCTRL(R11);ASSUME /CONTROL
                                04CF              SETBIT  PRC_V_CARRCNTL,PRC_W_FLAGS(R11)  ;SET CR/LF FLAG
                 FB2A'     30   04D3              BSBW    DCL$GETDVAL              ;GET FIRST TOKEN
           00     55       91   04D6              CMPB    R5,#PTR_K_COMDQUAL       ;/[NO]CONTROL QUALIFIER?
                  0E       12   04D9              BNEQ    20$                      ;NO, THEN BRANCH
                                04DB              ASSUME  PTR_V_NEGATE EQ 20       ;
               08 53       E9   04DB              BLBC    R3,10$                   ;BRANCH IF NOT NEGATED
             00F1 CB       B4   04DE              CLRW    PRC_W_PMPTCTRL(R11)      ;SET NOCONTROL
                                04E2              CLRBIT  PRC_V_CARRCNTL,PRC_W_FLAGS(R11)  ;INDICATE NO CR/LF
                 FB17'     30   04E6      10$:    BSBW    DCL$GETDVAL              ;GET "PROMPT" TOKEN
                 FB14'     30   04E9      20$:    BSBW    DCL$GETDVAL              ;GET PROMPT STRING
           04     55       91   04EC              CMPB    R5,#PTR_K_ENDLINE        ;IF PRESENT
                  11       12   04EF              BNEQ    30$                      ;THEN RESET THE PROMPT
           00000000'EF     D0   04F1              MOVL    DCL$T_PROMPT,-           ;ELSE RESTORE THE DEFAULT
             00F3 CB            04F7                      PRC_B_CONTINUE(R11)      ;
                00'8F      90   04FA              MOVB    #DCL$C_PROMPTLEN,-       ;
             00F0 CB            04FD                      PRC_B_PROMPTLEN(R11)     ;
                  18       11   0500              BRB     80$                      ;DONE
  50   000388FA 8F         D0   0502      30$:    MOVL    #CLI$_STRTOOLNG,R0       ;ASSUME STRING IS TOO LONG
           20     51       D1   0509              CMPL    R1,#ENT_K_MAX_PROMPT     ;IS IT TOO LONG?
                  13       1A   050C              BGTRU   90$                      ;YES, THEN ERROR
                                050E              ASSUME  ENT_K_MAX_PROMPT LT 256  ;
 00F0 CB     51    03      81   050E              ADDB3   #3,R1,-                  ;SAVE LENGTH OF PROMPT
                                0514                      PRC_B_PROMPTLEN(R11)     ;
 00F4 CB     62    51      28   0514              MOVC3   R1,(R2),PRC_G_PROMPT(R11)  ;SAVE PROMPT STRING
                                051A      80$:    STATUS  NORMAL                   ;RETURN NORMAL STATUS
                           05   0521      90$:    RSB
                                0522
                                0522              .END
```

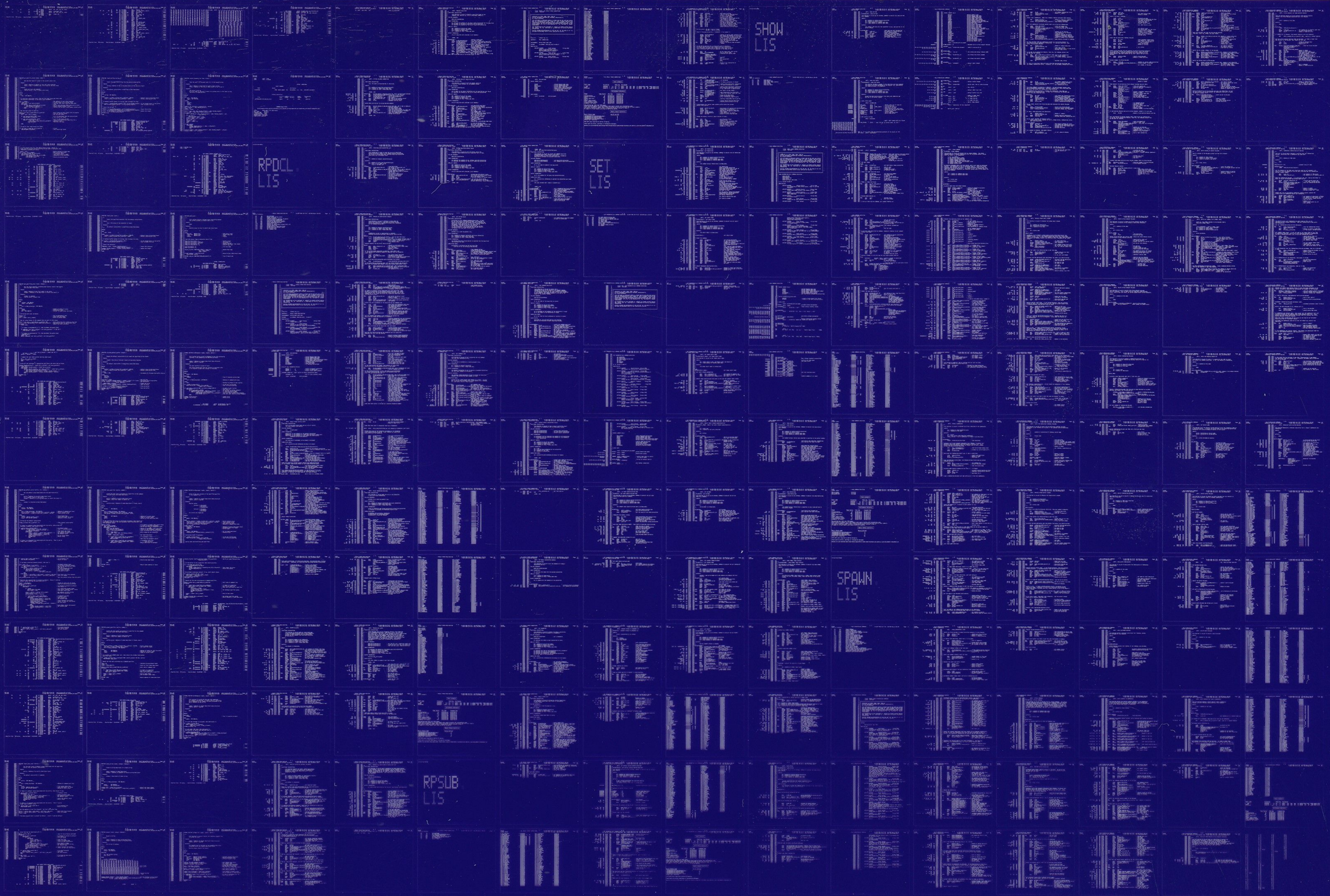
```
$$.TMP1                 = 00000001              LNM$V_CONCEALED         = 00000008
$$.TMP2                 = 00000061              LNM$_ATTRIBUTES         = 00000003
ACCESS                    00000000 R      02    LNM$_MAX_INDEX          = 00000007
CLASS                     00000004 R      02    LNM$_STRING             = 00000002
CLI$_DIRECT             = 00038030              LOG$C_PROCESS           = 00000002
CLI$_INVUIC             = 000381A8              L_ATTR                  = 00000010
CLI$_IVKEYW             = 00038060              L_MAX_INDEX             = 00000014
CLI$_IVPROT             = 00038070              MAX_TRANS_LVLS          = 0000000A
CLI$_NORMAL             = 00030001              PCB$L_UIC               = 000000BC
CLI$_STRTOOLNG          = 000388FA              PRC_B_CONTINUE            000000F3
CONTROL_CHARS             0000001D R      02    PRC_B_DEFRADIX            000000AE
CTRLY                     000004AD R      02    PRC_B_EXMDEPMOD           000000AD
CVTUIC                    00000143 R      02    PRC_B_EXMDEPWID           000000AC
DCL$CRLF                  ********  X     02    PRC_B_EXONLYL             0000012D
DCL$CVTUIC                000000A5 RG     02    PRC_B_FLAGS2              000000AF
DCL$C_PROMPTLEN           ********  X     02    PRC_B_IMGFLAG             00000078
DCL$GETDVAL               ********  X     02    PRC_B_OUTFLAGS            0000012C
DCL$ONCTLYRST             ********  X     02    PRC_B_PROMPTLEN           000000F0
DCL$RESETOOB              ********  X     02    PRC_C_LENGTH              00000534
DCL$SETCTLY               00000461 RG     02    PRC_G_COMMANDS            00000133
DCL$SETDEFALT             00000159 RG     02    PRC_G_PROMPT              000000F4
DCL$SETON                 0000043C RG     02    PRC_K_LENGTH              00000534
DCL$SETPROMPT             000004C6 RG     02    PRC_L_CURRKEY             00000048
DCL$SETPROT               000002D6 RG     02    PRC_L_EXMDEPADR           000000A8
DCL$SETUIC                00000037 RG     02    PRC_L_EXTARG              00000094
DCL$SETVERIFY             00000346 RG     02    PRC_L_EXTBLK              0000008C
DCL$SETVERIFY_IMAGE       000003AF RG     02    PRC_L_EXTCOD              0000009C
DCL$T_DSKNAM              00000014 RG     02    PRC_L_EXTHND              00000090
DCL$T_PROMPT              ********  X     02    PRC_L_EXTPRM              00000098
DCL$VERIFY_IMAGE          000003EA RG     02    PRC_L_IDFLNK              000000BC
ENT_K_MAX_PROMPT        = 00000020              PRC_L_IMGACTSTS           00000080
FAB$L_CTX                 ********  X     02    PRC_L_INDCLOCK            0000007C
FAB$L_FOP                 ********  X     02    PRC_L_INDEPTH             0000005C
FAB$M_ESC                 ********  X     02    PRC_L_INDFAB              0000001C
FAB$W_IFI                 ********  X     02    PRC_L_INDINPRAB           00000014
IDF_B_OUTFLAGS            00000038              PRC_L_INDOUTRAB           00000018
IDF_C_LENGTH              00000074              PRC_L_INPRAB              00000008
IDF_K_LENGTH              00000074              PRC_L_LASTKEY             0000004C
IDF_L_FILENAME            00000068              PRC_L_LSTSTATUS           000000B0
IDF_L_INPRABCTX           0000000C              PRC_L_ONCTLY              000000B8
IDF_L_LNK                 00000000              PRC_L_ONERROR             0000006C
IDF_L_ONCTLY              00000060              PRC_L_OUTOFBAND           000000B4
IDF_L_ONERROR             00000008              PRC_L_OUTRAB              0000000C
IDF_L_OUTRABCTX           00000024              PRC_L_OUTRABCTX           00000118
IDF_L_SEARCHCTX           00000064              PRC_L_PPFLIST             00000070
IDF_Q_LABEL               00000018              PRC_L_RECALLPTR           0000012F
IDF_Q_LOCAL               00000010              PRC_L_RESTART             00000058
IDF_T_INPDVI              0000003C              PRC_L_SAVAP               00000000
IDF_T_OUTDVI              00000028              PRC_L_SAVFP               00000004
IDF_W_FLAG                0000005E              PRC_L_SEVERITY            00000050
IDF_W_INPDID              00000052              PRC_L_SPWN                000000C0
IDF_W_INPFID              0000004C              PRC_L_STACKLM             000000A4
IDF_W_INPIFI              00000004              PRC_L_STACKPT             000000A0
IDF_W_INPRFA              00000058              PRC_L_STATUS              00000054
IDF_W_ONLEVEL             00000006              PRC_L_STS                 00000084
IDF_W_OUTIFI              00000020              PRC_L_STV                 00000088
IDF_W_OUTISI              00000022              PRC_L_SYMBOL              00000060
```

```
PRC_L_TMBX             00000074                 SYS$MODIFY             ********   GX   02
PRC_L_TRMLIST          00000010                 SYS$SETDDIR            ********    X   02
PRC_M_CTRLY          = 02000000                 SYS$SETDFPROT          ********    X   02
PRC_M_VERIFY         = 00000080                 SYS$TRNLNM             ********   GX   02
PRC_M_VERIMAGE       = 00000080                 SYS$TRNLOG             ********   GX   02
PRC_Q_ALLOCREG         00000020                 S_XLT_BUF            = 0000011C
PRC_Q_COMMAND          000000E0                 TABNAM                 00000008 R      02
PRC_Q_FLUSHTIME        000000D0                 TABNAMSZ             = 0000000C
PRC_Q_GLOBAL           00000028                 T_STRING_BUF         = 0000001C
PRC_Q_IMAGENAME        000000D8                 WRK_B_CMDOPT           FFFFFFC3
PRC_Q_KEYPAD           00000040                 WRK_B_MAXPARM          FFFFFFD0
PRC_Q_LABEL            00000030                 WRK_B_MINPARM          FFFFFFD1
PRC_Q_LOCAL            00000038                 WRK_B_PARMCNT          FFFFFFCE
PRC_Q_SAVEPRIV         000000E8                 WRK_B_PARMSUM          FFFFFFCF
PRC_T_OUTDVI           0000011C                 WRK_B_RECALLCNT        FFFFFFC5
PRC_V_CARRCNTL       = 00000000                 WRK_B_VALLEV           FFFFFFC4
PRC_V_CTRLY          = 00000019                 WRK_B_VERBTYP          FFFFFFC2
PRC_V_MODE           = 00000006                 WRK_C_LENGTH           FFFFF486
PRC_V_VERIMAGE       = 00000007                 WRK_G_BUFFER           FFFFF492
PRC_W_ASTIOSB          000000C6                 WRK_G_INPBUF           FFFFF896
PRC_W_ASTRETN          000000C8                 WRK_G_RESULT           FFFFF9B6
PRC_W_ASTSTATUS        000000C4                 WRK_K_LENGTH           FFFFF486
PRC_W_ATTMBX           0000007A                 WRK_L_CHARPTR          FFFFF48E
PRC_W_FLAGS            00000068                 WRK_L_DISALLOW         FFFFFFE6
PRC_W_INPCHAN          00000064                 WRK_L_ERRORRTN         FFFFF9AE
PRC_W_ONLEVEL          0000006A                 WRK_L_EXPANDPTR        FFFFF486
PRC_W_OUTIFI           00000114                 WRK_L_IMAGE            FFFFFFE2
PRC_W_OUTISI           00000116                 WRK_L_MARKPTR          FFFFF48A
PRC_W_OUTMBXCHN        000000CA                 WRK_L_PAROUT           FFFFFFD2
PRC_W_OUTMBXREF        000000CE                 WRK_L_PMPTADDR         FFFFF9A2
PRC_W_OUTMBXSIZ        000000CC                 WRK_L_PROMPTRTN        FFFFF9A6
PRC_W_PMPTCTRL         000000F1                 WRK_L_PROPTR           FFFFFFC6
PRC_W_WAITIOSB         00000066                 WRK_L_QUABLK           FFFFFFCA
PTR_B_LEVEL            00000004                 WRK_L_READRTN          FFFFF9AA
PTR_B_NUMBER           00000005                 WRK_L_RECALLPTR        FFFFFFEA
PTR_B_PARMCNT          00000006                 WRK_L_RSLEND           FFFFFFB6
PTR_B_VALUE            00000000                 WRK_L_RSLNXT           FFFFFFBA
PTR_C_LENGTH           0000000C                 WRK_L_SAVAP            FFFFFFF8
PTR_K_COLON          = 00000002                 WRK_L_SAVFP            FFFFFFFC
PTR_K_COMDQUAL       = 00000000                 WRK_L_SAVSP            FFFFFFF4
PTR_K_ENDLINE        = 00000004                 WRK_L_SIGNALRTN        FFFFFFD6
PTR_K_LENGTH           0000000C                 WRK_L_SPECRTN          FFFFF9B2
PTR_K_PARAMETR       = 00000003                 WRK_L_TAB_VEC          FFFFFFDE
PTR_L_DESCR            00000000                 WRK_L_VERB             FFFFFFBE
PTR_L_ENTITY           00000008                 WRK_W_FLAGS            FFFFFFF0
PTR_V_NEGATE         = 00000014                 WRK_W_FLAGS2           FFFFFFF2
Q_LOGNAM             = 00000000                 WRK_W_IMGCHAN          FFFFFFEE
Q_TABLE              = 00000008                 WRK_W_PMPTLEN          FFFFF99E
RAB$W_ISI              ********    X   02       W_STRING_LEN         = 00000018
RME$C_PPFECHO        = 00000002                 _$$_                 = 000000EF
SCH$GL_CURPCB          ********    X   02
SETUIC                 0000008F R      02
SS$_NOLOGNAM           ********    X   02
SS$_NORMAL             ********    X   02
SYS$ASCTOID            ********   GX   02
SYS$CMKRNL             ********   GX   02
SYS$CRELOG             ********    X   02
```

```
                                             +----------------+
                                             ! 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                       00000522  (   1314.)  02 (  2.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD  NOWRT NOVEC BYTE

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

Phase                    Page faults    CPU Time       Elapsed Time
-----                    -----------    --------       ------------
Initialization                     9    00:00:00.05    00:00:01.73
Command processing                81    00:00:00.68    00:00:06.50
Pass 1                           308    00:00:12.24    00:00:38.76
Symbol table sort                  0    00:00:01.49    00:00:02.52
Pass 2                           146    00:00:02.71    00:00:07.51
Symbol table output               25    00:00:00.21    00:00:00.80
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             571    00:00:17.41    00:00:57.85

The working set limit was 1500 pages.
63039 bytes (124 pages) of virtual memory were used to buffer the intermediate code.
There were 60 pages of symbol table space allocated to hold 944 non-local and 65 local symbols.
839 source lines were read in Pass 1, producing 18 object records in Pass 2.
50 pages of virtual memory were used to define 33 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                       14
TOTALS (all libraries)                                  26

1173 GETS were required to define 26 macros.

There were no errors, warnings or information messages.

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

0073 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



SHOW
LIS

RPDCL
LIS

SET
LIS

SPAWN
LIS

RPSUB
LIS