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
```

F 11

FILEID**DCLPARSE

```
DDDDDDDD      CCCCCCCC  LL          PPPPPPPP      AAAAAA    RRRRRRRR      SSSSSSSS  EEEEEEEEEE
DDDDDDDD      CCCCCCCC  LL          PPPPPPPP      AAAAAA    RRRRRRRR      SSSSSSSS  EEEEEEEEEE
DD      DD  CC          LL          PP      PP  AA      AA  RR      RR  SS          EE
DD      DD  CC          LL          PP      PP  AA      AA  RR      RR  SS          EE
DD      DD  CC          LL          PP      PP  AA      AA  RR      RR  SS          EE
DD      DD  CC          LL          PP      PP  AA      AA  RR      RR  SS          EE
DD      DD  CC          LL          PPPPPPPP    AA      AA  RRRRRRRR      SSSSSS    EEEEEEEE
DD      DD  CC          LL          PPPPPPPP    AA      AA  RRRRRRRR      SSSSSS    EEEEEEEE
DD      DD  CC          LL          PP          AAAAAAAAAA  RR  RR              SS  EE
DD      DD  CC          LL          PP          AAAAAAAAAA  RR  RR              SS  EE
DD      DD  CC          LL          PP          AA      AA  RR    RR            SS  EE              ....
DD      DD  CC          LL          PP          AA      AA  RR    RR            SS  EE              ....
DDDDDDDD      CCCCCCCC  LLLLLLLLLL  PP          AA      AA  RR      RR  SSSSSSSS    EEEEEEEEEE      ....
DDDDDDDD      CCCCCCCC  LLLLLLLLLL  PP          AA      AA  RR      RR  SSSSSSSS    EEEEEEEEEE      ....

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  DCLPARSE - PARSE A DCL COMMAND
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      ; AUTHOR: TIM HALVORSEN, NOV 1980
0000      ;
0000      ; MODIFIED BY:
0000      ;
0000      ;       V03-014 HWS0083         Harold Schultz  19-Jul-1984
0000      ;               Fix building of error message frame for LIB$SIGNAL.
0000      ;               In the event of a buffer overflow, signal CLI$_BUFOVF
0000      ;               instead of SS$_BUFFEROVF
0000      ;
0000      ;       V03-013 PCG0019         Peter George    08-Dec-1983
0000      ;               Fix bug in error message routine.
0000      ;
0000      ;       V03-012 PCG0018         Peter George    27-Jul-1983
0000      ;               Fill in WRK_L_SIGNALRTN.
0000      ;
0000      ;       V03-011 PCG0017         Peter George    15-Jun-1983
0000      ;               Return, do not signal, NOCOMD status.
0000      ;               Remove special code for negative statuses.
0000      ;
0000      ;       V03-010 PCG0016         Peter George    20-Apr-1983
0000      ;               Include command segment when signalling syntax errors.
0000      ;               Clear WRK_B_PARMSUM.
0000      ;
0000      ;       V03-009 PCG0015         Peter George    15-Feb-1983
0000      ;               Update to new structure level.
0000      ;
0000      ;       V03-008 PCG0014         Peter George    08-Jan-1983
0000      ;               Change .ASCID default prompt to .ASCIC.
0000      ;
0000      ;       V03-007 PCG0013         Peter George    27-Dec-1982
```

```
                               0000     ;               Fix accvio in referencing command descriptor.
                               0000     ;
                               0000     ;       V03-006 PCG0012         Peter George    14-Dec-1982
                               0000     ;               For compatibility, partially back off PCG0011.
                               0000     ;
                               0000     ;       V03-005 PCG0011         Peter George    02-Dec-1982
                               0000     ;               Command string is passed in by descriptor,
                               0000     ;               rather than address of descriptor.
                               0000     ;
                               0000     ;       V03-004 PCG0010         Peter George    16-Nov-1982
                               0000     ;               Get prompt string descriptor from INT instead
                               0000     ;               of just the address of the descriptor.
                               0000     ;               Use WRK_G_INPUF instead of WRK_G_DCLPRSBUF.
                               0000     ;
                               0000     ;       V03-003 PCG0009         Peter George    15-Nov-1982
                               0000     ;               Use STRTOOLNG instead of PMPTOOLNG.
                               0000     ;
                               0000     ;       V03-002 PCG0008         Peter George    15-Oct-1982
                               0000     ;               Get the PROBEs right this time.
                               0000     ;               Accept a prompt string as another argument.
                               0000     ;               Prompt for a command line if none is supplied.
                               0000     ;
                               0000     ;       V03-001 PCG0007         Peter George    15-Jul-1982
                               0000     ;               Add keyword parsing support.
                               0000     ;               Use INT data structure.
                               0000     ;               Add support for prompt and continuation routines.
                               0000     ;               Correct sense of branches after PROBEs.
                               0000     ;               Remove unneccessary macro library calls.
                               0000     ;
                               0000     ;--
                               0000
                               0000     ;
                               0000     ; MACRO LIBRARY CALLS
                               0000     ;
                               0000
                               0000             $$CLITABDEF                     ; DEFINE TABLE STRUCTURES
                               0000             $$INTDEF                        ; DEFINE INTERFACE FORMAT
                               0000             DCLDEF                          ; DEFINE CLINT OWN STORAGE AREA
                               0000             WRKDEF                          ; DEFINE COMMAND WORK AREA
                               0000             $SSDEF                          ; DEFINE SYSTEM MESSAGES
                               0000             $CLIMSGDEF                      ; DEFINE ERROR/STATUS VALUES
                               0000             $STSDEF                         ; DEFINE STATUS CODE FIELDS
                               0000             $PSLDEF                         ; DEFINE PROCESSOR STATUS FIELDS
                               0000
                    0000001A   0000     CTRLZ   = 26                            ; CONTROL/Z CHARACTER
                               0000
                    00000000   0000             .PSECT  DCL$ZCODE,BYTE,RD,NOWRT
                               0000
                               0000     DEFAULT_PROMPT:
 20 3E 44 4E 41 4D 4D 4F 43 00'  0000             .ASCIC  /COMMAND> /
                            09   0000
```

```
                              000A          .SBTTL  PARSE A DCL COMMAND
                              000A   ;+
                              000A   ; DCL$DCLPARSE - PARSE A DCL COMMAND STRING
                              000A   ;
                              000A   ; THIS ROUTINE PARSES A DCL COMMAND STRING GIVEN THE ADDRESS OF THE COMMAND
                              000A   ; TABLES WHICH DESCRIBE THE SYNTAX OF THE COMMAND SET.
                              000A   ;
                              000A   ; INPUTS:
                              000A   ;
                              000A   ;       4(AP) = ADDRESS OF REQUEST DESCRIPTOR
                              000A   ;
                              000A   ; OUTPUTS:
                              000A   ;
                              000A   ;       NONE
                              000A   ;-
                              000A
                        0FFC  000A          .ENTRY  DCL$DCLPARSE,-
                              000C                  ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
                              000C
      52    00000000'GF   DE  000C          MOVAL   G^CTL$GL_DCLPRSOWN,R2   ;GET ADDRESS OF WRK POINTER
         5B    04 AC      D0  0013          MOVL    4(AP),R11               ;GET ADDRESS OF REQUEST DESCRIPTOR
                              0017
                              0017   ;
                              0017   ; IF NO COMMAND STRING IS SPECIFIED, AND WE ARE NOT PROMPTING, THEN
                              0017   ; CAUSE ALL SUBSEQUENT INTERFACE REQUESTS TO REFER TO THE ALREADY PARSED
                              0017   ; ORIGINAL COMMAND LINE.
                              0017   ;
               0C AB      D5  0017          TSTL    INT_L_ENTADDR(R11)      ;COMMAND STRING SPECIFIED?
                  0E      12  001A          BNEQ    20$                     ;YES, THEN BRANCH
         50    18 AB      D0  001C          MOVL    INT_L_LIST(R11),R0      ;ROUTINE LIST SPECIFIED?
                  05      13  0020          BEQL    10$                     ;NO, THEN BRANCH
               08 A0      D5  0022          TSTL    INT_L_CONTINRTN(R0)     ;CONTINUATION ROUTINE SPECIFIED?
                  03      12  0025          BNEQ    20$                     ;YES, THEN BRANCH
                01A3      31  0027   10$:   BRW     RESTORE_SUPER_MODE      ;NO, RESTORE THE SUPERVISOR MODE COMMAND
                              002A
                              002A   ;
                              002A   ; VALIDATE THE TABLE STRUCTURE.  IF OLD, SUPPORTED VERSION, THEN UPGRADE THEM
                              002A   ; BEHIND THE USER'S BACK.
                              002A   ;
               14 AB      DD  002A   20$:   PUSHL   INT_L_FREEVM(R11)       ;PUSH ADDR OF LIB$FREE_VM ROUTINE
               10 AB      DD  002D          PUSHL   INT_L_GETVM(R11)        ;PUSH ADDR OF LIB$GET_VM ROUTINE
               04 AB      9F  0030          PUSHAB  INT_L_TABLES(R11)       ;PUSH PLACE TO RETURN CONVERTED ADDRESS
               04 AB      DD  0033          PUSHL   INT_L_TABLES(R11)       ;PUSH ADDRESS OF TABLES
  00000000'GF     04      FB  0036          CALLS   #4,G^CDU$UPGRADE_TABLE  ;VALIDATE AND POSSIBLY CONVERT TABLES
               21 50      E9  003D          BLBC    R0,35$                  ;SIGNAL ERROR AND EXIT
                              0040
                              0040   ;
                              0040   ; IF NO USER MODE WRK BLOCK HAS BEEN ALLOCATED BEFORE, THEN ALLOCATE
                              0040   ; AND INITIALIZE ONE NOW.
                              0040   ;
            5A    62      D0  0040          MOVL    (R2),R10                ;WRK BLOCK ALLOCATED?
                  0A      13  0043          BEQL    30$                     ;NO, THEN BRANCH
                  03      0D  0045          PROBEW  #PSL$C_USER,-           ;USER WRITABLE?
             0B7A 8F          0047                  #-WRK_K_LENGTH,-        ;
             F486 CA          004A                  WRK_K_LENGTH(R10)       ;
                  20      12  004D          BNEQ    50$                     ;YES, THEN BRANCH
                              004F
```

```
                               004F     ;
                               004F     ; ALLOCATE USER MODE WRK BLOCK
                               004F     ;
        00000B7A 8F     DD     004F     30$:    PUSHL   #-WRK_K_LENGTH          ;LENGTH TO ALLOCATE
                  5E    DD     0055             PUSHL   SP                      ;PLACE TO RETURN ADDRESS
               04 AE    DF     0057             PUSHAL  4(SP)                   ;ADDRESS OF LONGWORD CONTAINING LENGTH
            10 BB   02  FB     005A             CALLS   #2,@INT_L_GETVM(R11)    ;ALLOCATE DYNAMIC MEMORY
               03 50    E8     005E             BLBS    R0,40$                  ;BRANCH IF SUCCESS
                010D    31     0061     35$:    BRW     EXIT                    ;EXIT IF ERROR
    00000B7A 8F     8E  C1     0064     40$:    ADDL3   (SP)+,#-WRK_K_LENGTH,-  ;COMPUTE ENDING ADDRESS WRK BLOCK
                    62         006B                     (R2)                    ;
            5A      62  D0     006C             MOVL    (R2),R10                ;GET ADDRESS OF WRK BLOCK
                               006F
                               006F     ;
                               006F     ; IF CLINT OWN STORAGE IS ALLOCATED, THEN FREE IT SO THAT SUBSEQUENT
                               006F     ; INTERFACE REQUESTS CAUSE THE OWN STORAGE TO BE REINITIALIZED.
                               006F     ;
    52      00000000'GF     DE     006F     50$:    MOVAL   G^CTL$GL_CLINTOWN,R2    ;GET ADDRESS OF CLINT OWN POINTER
                    62  D5     0076             TSTL    (R2)                    ;CLINT OWN STORAGE ALLOCATED?
                    03  13     0078             BEQL    60$                     ;NO, THEN BRANCH
                  018D  30     007A             BSBW    DEALLOC_OWN             ;DEALLOCATE OWN STORAGE
                               007D
                               007D     ;
                               007D     ; FILL IN WRK BLOCK FIELDS
                               007D     ;
             F492 CA    9E     007D     60$:    MOVAB   WRK_G_BUFFER(R10),-     ;SET ADDRESS OF EXPANSION BUFFER
             F486 CA           0081                     WRK_L_EXPANDPTR(R10)    ;
             F9B6 CA    9E     0084             MOVAB   WRK_G_RESULT(R10),-     ;SET ADDRESS OF RESULT PARSE TABLE
               BA AA           0088                     WRK_L_RSLNXT(R10)       ;
               F0 AA    B4     008A             CLRW    WRK_W_FLAGS(R10)        ;RESET COMMAND FLAGS
             2000 8F    A8     008D             BISW    #WRK_M_USRMODE,-        ;SET USER MODE PARSE FLAG
               F0 AA           0091                     WRK_W_FLAGS(R10)        ;
               C4 AA    94     0093             CLRB    WRK_B_VALLEV(R10)       ;RESET VALUE LEVEL
               CE AA    94     0096             CLRB    WRK_B_PARMCNT(R10)      ;RESET PARAMETER COUNT
               CF AA    94     0099             CLRB    WRK_B_PARMSUM(R10)      ;RESET PARAMETER TOTAL
             F9AA CA    D4     009C             CLRL    WRK_L_READRTN(R10)      ;ASSUME NO CONTINUATION ROUTINE
             F9A6 CA    D4     00A0             CLRL    WRK_L_PROMPTRTN(R10)    ;ASSUME NO PROMPT ROUTINE
             F99E CA    7C     00A4             CLRQ    WRK_W_PMPTLEN(R10)      ;ASSUME NO PROMPT STRING
             F9B2 CA    D4     00A8             CLRL    WRK_L_SPECRTN(R10)      ;NO SPECIAL CHARACTER PROCESSING
        00000177'EF     9E     00AC             MOVAB   ERROR,-                 ;SET ADDRESS OF ERROR HANDLER
             F9AE CA           00B2                     WRK_L_ERRORRTN(R10)     ;
        00000181'EF     9E     00B5             MOVAB   ERRORMSG,-              ;SET ADDRESS OF ERROR SIGNALER
               D6 AA           00BB                     WRK_L_SIGNALRTN(R10)    ;
          50   18 AB    D0     00BD             MOVL    INT_L_LIST(R11),R0      ;GET ADDRESS OF ROUTINE LIST
                  4E    13     00C1             BEQL    75$                     ;SKIP IF NONE
               08 A0    D0     00C3             MOVL    INT_L_CONTINRTN(R0),-   ;GET CONTINUATION ROUTINE
             F9AA CA           00C6                     WRK_L_READRTN(R10)      ;
               04 A0    D0     00C9             MOVL    INT_L_PROMPTRTN(R0),-   ;GET PROMPT ROUTINE
             F9A6 CA           00CC                     WRK_L_PROMPTRTN(R10)    ;
               0C A0    7D     00CF             MOVQ    INT_W_PMPTLEN(R0),-     ;GET PROMPT STRING
             F99E CA           00D2                     WRK_W_PMPTLEN(R10)      ;
                  0D    12     00D5             BNEQ    70$                     ;SKIP IF PRESENT
        51   FF25 CF    9E     00D7             MOVAB   DEFAULT_PROMPT,R1       ;GET ADDRESS OF ASCIC DEFAULT PROMPT
          50      81    9A     00DC             MOVZBL  (R1)+,R0                ;GET LENGTH OF PROMPT
      F99E CA     50    7D     00DF             MOVQ    R0,WRK_W_PMPTLEN(R10)   ;GET DEFAULT PROMPT STRING
                               00E4
                               00E4     ;
```

```
                        00E4   ; IF COMMAND IS MISSING, THEN PROMPT FOR IT NOW
                        00E4   ;
  50  000388FA 8F   D0  00E4   70$:    MOVL    #CLI$_STRTOOLNG,R0      ;ASSUME THAT PROMPT IS TOO LONG
           F99E CA  B1  00EB           CMPW    WRK_W_PMPTLEN(R10),-    ;IS IT?
                20      00EF                   #ENT_K_MAX_PROMPT       ;
                63  1A  00F0           BGTRU   200$                    ;YES, THEN EXIT
             0C AB  D5  00F2           TSTL    INT_L_ENTADDR(R11)      ;IS COMMAND PRESENT?
                1A  12  00F5           BNEQ    75$                     ;YES, THEN SKIP
      50   F9AA CA  D0  00F7           MOVL    WRK_L_READRTN(R10),R0   ;GET PROMPT ROUTINE
      51   F99E CA  9E  00FC           MOVAB   WRK_W_PMPTLEN(R10),R1   ;GET PROMPT STRING
              011B  30  0101           BSBW    DCL$USER_INPUT          ;GET THE COMMAND LINE
                        0104
                        0104   ;
                        0104   ; IF EOF WAS FOUND, THEN RETURN THAT STATUS.
                        0104   ;
  50  00000000'8F   D1  0104           CMPL    #RMS$_EOF,R0            ;END OF INPUT?
                24  12  010B           BNEQ    80$                     ;NO, START PROCESSING THE COMMAND
             FEF0'  30  010D           BSBW    DCL$GENEOL              ;GENERATE EOL DESCRIPTOR
                    04  0110           RET                             ;EXIT
                        0111
                        0111   ;
                        0111   ; COPY USER SUPPLIED INPUT STRING INTO SCRATCH AREA
                        0111   ;
        50   0C AB  D0  0111   75$:    MOVL    INT_L_ENTADDR(R11),R0   ;GET ADDRESS OF COMMAND DESCRIPTOR
          0000'C0   B1  0115           CMPW    DSC$W_LENGTH(R0),-      ;CHECK IF TOO BIG FOR INPUT BUFFER
          0100 8F       0119                   #WRK_C_INPBUFSIZ        ;
                27  1A  011C           BGTRU   180$                    ;BRANCH IF STRING IS TOO BIG
           F895 CA  9E  011E           MOVAB   WRK_G_INPBUF-1(R10),-   ;SET INPUT STRING POINTER
           F48E CA      0122                   WRK_L_CHARPTR(R10)      ;
          0000'C0   28  0125           MOVC    DSC$W_LENGTH(R0),-      ;COPY STRING INTO INPUT BUFFER
          0000'D0       0129                   @DSC$A_POINTER(R0),-    ;
           F896 CA      012C                   WRK_G_INPBUF(R10)       ;
                63  94  012F           CLRB    (R3)                    ;PUT NULL STOPPER AS END-OF-LINE
                        0131
                        0131   ;
                        0131   ; GET FIRST TOKEN FROM COMMAND LINE
                        0131   ;
             FECC'  30  0131   80$:    BSBW    DCL$MARK                ;MARK CURRENT PARSE POSITION
             FEC9'  30  0134           BSBW    DCL$GETOKEN             ;GET COMMAND VERB
                15  13  0137           BEQL    190$                    ;IF NONE, RETURN ERROR
        58   04 AB  D0  0139           MOVL    INT_L_TABLES(R11),R8    ;GET ADDRESS OF COMMAND TABLES
             FEC0'  30  013D           BSBW    DCL$SEARCH_VERB         ;SEARCH VERB TABLE FOR VERB
             12 50  E9  0140           BLBC    R0,200$                 ;BRANCH IF ERROR
                15  11  0143           BRB     PARSE_VERB_QUALS
                        0145
                        0145   180$:   STATUS  BUFOVF                  ;INPUT STRING IS TOO BIG FOR BUFFER
                07  11  014C           BRB     200$
                        014E
                        014E   190$:   STATUS  NOCOMD                  ;SET NO COMMAND STATUS
             FEA8'  30  0155   200$:   BSBW    DCL$GENEOL              ;GENERATE EOL DESCRIPTOR
                17  11  0158           BRB     EXIT                    ;EXIT WITH ERROR
                        015A
                        015A   ;
                        015A   ; PROCESS COMMAND QUALIFIERS AND PARAMETERS
                        015A   ;
                        015A   PARSE_VERB_QUALS:
             FEA3'  30  015A           BSBW    DCL$PARSE_COMMAND       ;PARSE THE REST OF THE COMMAND
      000380B0 8F   50   D1  015D          CMPL    R0,#CLI$_NOCOMD           ;IF NOT CTRLZ-ED
                    01   12  0164          BNEQ    10$                       ;THEN CHECK FOR ERRORS
                         04  0166          RET                               ;ELSE RETURN NO COMMAND STATUS
                 07 50   E9  0167  10$:    BLBC    R0,EXIT                   ;SIGNAL ANY ERRORS
                             016A
                             016A  NORMAL_EXIT:
                             016A          STATUS  NORMAL                    ;SET NORMAL STATUS
                             0171
                             0171  ;
                             0171  ; SIGNAL ALL ERRORS.
                             0171  ;
                 02 50   E8  0171  EXIT:   BLBS    R0,90$
                    0B   10  0174          BSBB    ERRORMSG
                         04  0176  90$:    RET
                             0177
                             0177  ;
                             0177  ; HANDLE ERRORS DETECTED BY THE CHARACTER INPUT ROUTINES
                             0177  ;
                    50   DD  0177  ERROR:  PUSHL   R0                        ;SAVE ERROR/STATUS VALUE
      00000000'GF   01   FB  0179          CALLS   #1,G^LIB$SIGNAL           ;SIGNAL THE ERROR
                         04  0180          RET                               ;RETURN TO CALLER
                         0181        .SBTTL  SIGNAL ERROR MESSAGE
                         0181  ;+
                         0181  ; ERRORMSG - SIGNAL ERROR MESSAGE
                         0181  ;
                         0181  ; THIS ROUTINE IS CALLED TO SIGNAL AN ERROR MESSAGE AND DISPLAY THE SEGMENT
                         0181  ; OF THE COMMAND LINE THAT IS IN ERROR.
                         0181  ;
                         0181  ; INPUTS:
                         0181  ;
                         0181  ;       R0 = ERROR NUMBER.
                         0181  ;       WRK_L_MARKPTR = ADDRESS OF START OF TOKEN IN EXPANSION BUFFER.
                         0181  ;       WRK_L_EXPANDPTR = ADDRESS OF NEXT BYTE IN EXPANSION BUFFER.
                         0181  ;       R10 = BASE ADDRESS OF COMMAND WORK AREA.
                         0181  ;       R11 = BASE ADDRESS OF PROCESS WORK AREA.
                         0181  ;
                         0181  ; OUTPUTS:
                         0181  ;
                         0181  ;       THE APPROPRIATE ERROR MESSAGE IS DISPLAYED ALONG WITH THE SEGMENT OF
                         0181  ;       THE COMMAND LINE IN ERROR.
                         0181  ;
                         0181  ;       R0 IS PRESERVED ACROSS CALL.
                         0181  ;-
                         0181
                         0181  ERRORMSG:                                       ;OUTPUT ERROR MESSAGE
              3F    BB   0181          PUSHR   #^M<R0,R1,R2,R3,R4,R5>  ;SAVE REGISTERS
                         0183
                         0183  ;
                         0183  ; Check if message should be suppressed.
                         0183  ;
000380B0 8F   50    D1   0183          CMPL    R0,#CLI$_NOCOMD         ;IS IT NOCOMD STATUS?
              3B    13   018A          BEQL    60$                     ;YES, DO NOT SIGNAL
   37 50      1C    E0   018C          BBS     #STS$V_INHIB_MSG,R0,60$ ;BR IF NO MESSAGE DESIRED
                         0190
                         0190  ;
                         0190  ; Check if offending text should be output as part of this error message.
                         0190  ;
         55   02    D0   0190          MOVL    #2,R5                   ;ASSUME NO COMMAND SET WILL BE OUTPUT
              54    D4   0193          CLRL    R4                      ;SET STACK USAGE
              01    E0   0195          BBS     #WRK_V_COMMAND,-        ;DO NOT OUTPUT IF COMMAND IN EXECUTION
   22 F0 AA              0197                  WRK_W_FLAGS(R10),40$    ;
       FE63'  30         019A          BSBW    DCL$MARKEDTOKEN         ;GET DESCRIPTOR OF CURRENT PARSE STRING
              62    95   019D          TSTB    (R2)                    ;DOES TOKEN START WITH EOL CHAR?
              1B    13   019F          BEQL    40$                     ;IF SO, ASSUME AT EOL AND SKIP TEXT
              51    D5   01A1          TSTL    R1                      ;WILL ANY TOKEN BE SHOWN?
              17    13   01A3          BEQL    40$                     ;IF NOT, SKIP TEXT
                         01A5
                         01A5  ;
                         01A5  ; Build the command line part of the message argument vector.
                         01A5  ;
         7E   51    7D   01A5  30$:    MOVQ    R1,-(SP)                ;PUSH SEGMENT DESCRIPTOR ON STACK
              5E    DD   01A8          PUSHL   SP                      ;PUSH ADDRESS OF SEGMENT DESCRIPTOR
         7E   11    B0   01AA          MOVW    #^X0011,-(SP)           ;ONLY OUTPUT THE TEXT PART
         7E   01    B0   01AD          MOVW    #1,-(SP)                ;ONE FAO ARGUMENT
    00038248 8F     DD   01B0          PUSHL   #CLI$_CMDSEG            ;PUSH MESSAGE CODE
         55   05    D0   01B6          MOVL    #5,R5                   ;SET ARGUMENT COUNT
         54   08    D0   01B9          MOVL    #2*4,R4                 ;SET STACK USAGE
                         01BC
                         01BC   359 ;
                         01BC   360 ; Build the status part of the message argument vector.
                         01BC   361 ;
                 00  DD  01BC   362 40$:    PUSHL   #0                      ;SET FAO COUNT
                 50  DD  01BE   363         PUSHL   R0                      ;SET STATUS CODE
     00000000'GF 55  FB  01C0   364         CALLS   R5,G^LIB$SIGNAL         ;SIGNAL THE ERROR
              5E 54  C0  01C7   365 60$:    ADDL    R4,SP                   ;POP EVERYTHING UP TO BUFFER AND DESC.
                 3F  BA  01CA   366 70$:    POPR    #^M<R0,R1,R2,R3,R4,R5>  ;RESTORE REGISTERS
                 05  01CC   367         RSB                             ;
```

```
                              01CD             .SBTTL  RESTORE SUPERVISOR MODE DATA STRUCTURES
                              01CD        ;+
                              01CD        ; RESTORE_SUPER_MODE - RESTORE SUPERVISOR MODE DATA STRUCTURES
                              01CD        ;
                              01CD        ; CAUSE ALL SUBSEQUENT INTERFACE REQUESTS TO REFER TO THE ALREADY
                              01CD        ; PARSED ORIGINAL COMMAND LINE.  THIS IS DONE BY DEALLOCATING THE
                              01CD        ; USER MODE WRK BLOCK, DEALLOCATING THE CLINT OWN STORAGE, AND ZEROING
                              01CD        ; THE POINTERS TO BOTH OF THESE AREAS, CAUSING THE INTERFACE ROUTINES
                              01CD        ; TO BE REINITIALIZED WITH THE THE SUPERVISOR MODE WRK BLOCK.
                              01CD        ;
                              01CD        ;       R2 =    ADDRESS OF CURRENT WRK DATA STRUCTURE
                              01CD        ;
                              01CD        ;-
                              01CD
                              01CD        RESTORE_SUPER_MODE:
              5A    62   D0   01CD              MOVL    (R2),R10                ;WRK BLOCK ALLOCATED?
                    03   13   01D0              BEQL    5$                      ;NO, THEN BRANCH
                  0016   30   01D2              BSBW    DEALLOC_WRK             ;YES, DEALLOCATE THE WRK BLOCK
                              01D5        ;
                              01D5        ;
                              01D5        ; IF CLINT OWN STORAGE IS ALLOCATED, THEN FREE IT SO THAT SUBSEQUENT
                              01D5        ; INTERFACE REQUESTS CAUSE THE OWN STORAGE TO BE REINITIALIZED.
                              01D5        ;
       52    00000000'GF   DE   01D5      5$:       MOVAL   G^CTL$GL_CLINTOWN,R2    ;GET ADDRESS OF CLINT OWN POINTER
                    62   D5   01DC              TSTL    (R2)                    ;CLINT OWN STORAGE ALLOCATED?
                    03   13   01DE              BEQL    10$                     ;NO, THEN BRANCH
                  0027   30   01E0              BSBW    DEALLOC_OWN             ;DEALLOCATE OWN STORAGE
                              01E3        10$:      STATUS  NORMAL                  ;RETURN SUCCESSFUL
                         04   01EA              RET
```

```
                              01EB              .SBTTL  DEALLOCATE USER MODE WRK DATA STRUCTURE
                              01EB      ;+
                              01EB      ; DEALLOC_WRK - DEALLOCATE USER MODE WRK DATA STRUCTURE
                              01EB      ;
                              01EB      ; DEALLOCATE USER MODE WRK BLOCK.  IF SUPERVISOR MODE, THEN JUST ZERO
                              01EB      ; THE POINTER.
                              01EB      ;
                              01EB      ;       R2 =                    ADDRESS OF CTL$GL_DCLPRSOWN
                              01EB      ;       R10 =                   ADDRESS OF WRK BLOCK
                              01EB      ;       INT_L_FREEVM(R11) =     ADDRESS OF LIB$FREE_VM ROUTINE
                              01EB      ;
                              01EB      ;-
                              01EB
                              01EB      DEALLOC_WRK:
                03      0D    01EB              PROBEW  #PSL$C_USER,-           ;USER WRITABLE?
           0B7A 8F            01ED                      #-WRK_K_LENGTH,-        ;
           F486 CA            01F0                      WRK_K_LENGTH(R10)       ;
                12      13    01F3              BEQL    10$                     ;NO, THEN BRANCH
                              01F5                                              ;YES, THEN DEALLOCATE
       00000B7A 8F      DD    01F5              PUSHL   #-WRK_K_LENGTH          ;LENGTH OF LOCAL WRK BLOCK
                52      DD    01FB              PUSHL   R2                      ;ADDRESS OF LONGWORD CONTAINING ADDRESS
             04 AE      DF    01FD              PUSHAL  4(SP)                   ;ADDRESS OF WORD CONTAINING LENGTH
          14 BB 02      FB    0200              CALLS   #2,@INT_L_FREEVM(R11)   ;DEALLOCATE WRK BLOCK
             5E 04      C0    0204              ADDL    #4,SP                   ;RESTORE THE STACK
                62      D4    0207      10$:    CLRL    (R2)                    ;INDICATE BLOCK NO LONGER EXISTS
                        05    0209              RSB
```

```
                         020A        .SBTTL  DEALLOCATE CLINT OWN STORAGE
                         020A  ;+
                         020A  ; DEALLOC_OWN - DEALLOCATE CLINT OWN STORAGE
                         020A  ;
                         020A  ; DEALLOCATE CLINT OWN STORAGE.
                         020A  ;
                         020A  ;     R2 =                     ADDRESS OF CLINT OWN STORAGE
                         020A  ;     INT_L_FREEVM(R11) =      ADDRESS OF LIB$FREE_VM ROUTINE
                         020A  ;
                         020A  ;-
                         020A
                         020A  DEALLOC_OWN:
   00000090 8F      DD   020A        PUSHL    #DCL_C_SIZE             ;LENGTH OF CLINT OWN STORAGE
               52   DD   0210        PUSHL    R2                      ;ADDRESS OF LONGWORD CONTAINING ADDRESS
            04 AE   DF   0212        PUSHAL   4(SP)                   ;ADDRESS OF WORD CONTAINING LENGTH
      14 BB    02   FB   0215        CALLS    #2,@INT_L_FREEVM(R11)   ;DEALLOCATE CLINT OWN STORAGE
         5E    04   C0   0219        ADDL     #4,SP                   ;RESTORE THE STACK
               62   D4   021C        CLRL     (R2)                    ;INDICATE STORAGE NO LONGER EXISTS
                    05   021E        RSB
```

```
                      021F          .SBTTL  GET INPUT FROM THE USER
                      021F   ;+
                      021F   ; DCL$USER_INPUT - GET INPUT FROM THE USER
                      021F   ;
                      021F   ; THIS ROUTINE CALLS A USER-SUPPLIED INPUT ROUTINE WITH A PROMPT TO GET
                      021F   ; ADDITIONAL COMMAND INFORMATION FROM THE USER.
                      021F   ;
                      021F   ; THE FORMAT OF THE CALL IS:
                      021F   ;
                      021F   ;       P1 =    ADDRESS OF RETURN DESCRIPTOR
                      021F   ;       P2 =    ADDRESS OF PROMPT DESCRIPTOR
                      021F   ;       P3 =    ADDRESS OF WORD TO RECEIVE THE RETURN LENGTH
                      021F   ;
                      021F   ; INPUTS:
                      021F   ;
                      021F   ;       R0 =    ADDRESS OF USER INPUT ROUTINE
                      021F   ;       R1 =    ADDRESS OF PROMPT STRING DESCRIPTOR
                      021F   ;       R10 =   ADDRESS OF COMMAND DATA STRUCTURE
                      021F   ;
                      021F   ; OUTPUTS:
                      021F   ;
                      021F   ;       R0 =    STATUS OF THE READ
                      021F   ;-
                      021F
                      021F   DCL$USER_INPUT::
      7E   54   7D    021F          MOVQ    R4,-(SP)                ;SAVE R4 AND R5
      7E   52   7D    0222          MOVQ    R2,-(SP)                ;SAVE R2 AND R3
      5E   21   C2    0225          SUBL    #ENT_K_MAX_PROMPT+1,SP  ;ALLOCATE PROMPT BUFFER
      55   50   D0    0228          MOVL    R0,R5                   ;SAVE ROUTINE ADDR
                      022B
                      022B   ;
                      022B   ; IF A PROMPT WAS SUPPLIED, THEN PUSH ITS DESCRIPTOR ON THE STACK.
                      022B   ;
           51   D5    022B          TSTL    R1                      ;WAS A PROMPT SUPPLIED?
           05   13    022D          BEQL    10$                     ;NO, THEN CREATE ONE
      7E   61   7D    022F          MOVQ    (R1),-(SP)              ;PUSH DESCRIPTOR OF PROMPT STRING
           1C   11    0232          BRB     20$                     ;ISSUE THE PROMPT
                      0234
                      0234   ;
                      0234   ; IF NO PROMPT WAS SUPPLIED, THEN BUILD A CONTINUATION PROMPT ON THE STACK.
                      0234   ;
   6E    5F 8F   90   0234   10$:   MOVB    #^A/_/,(SP)             ;INSERT AN UNDERSCORE
           5E   DD    0238          PUSHL   SP                      ;PUSH PROMPT BUFFER ADDRESS
      7E   01   D0    023A          MOVL    #1,-(SP)                ;INIT PROMPT LENGTH
  6E  F99E CA   A0    023D          ADDW    WRK_W_PMPTLEN(R10),(SP) ;SET PROMPT BUFFER LENGTH
           55   DD    0242          PUSHL   R5                      ;SAVE ROUTINE ADDRESS
     F99E CA    28    0244          MOVC3   WRK_W_PMPTLEN(R10),-    ;COPY IT INTO THE PROMPT BUFFER
     F9A2 DA          0248                  @WRK_L_PMPTADDR(R10),-  ;
       0D AE          024B                  13(SP)                  ;
       55 8ED0        024D          POPL    R5                      ;RESTORE ROUTINE ADDRESS
                      0250
                      0250   ;
                      0250   ; SET UP PROMPT PARAMETERS AND THEN ISSUE THE PROMPT.
                      0250   ;
     F894 CA    9E    0250   20$:   MOVAB   WRK_G_INPBUF-2(R10),-   ;GET ADDRESS OF INPUT STRING
       52             0254                  R2                      ;
       FF A2    9E    0255          MOVAB   -1(R2),-                ;RESET THE CHARACTER POINTER
```

```
          F48E CA      0258              WRK_L_CHARPTR(R10)           ;
             52    DD  025B      PUSHL   R2                           ;PUSH ADDRESS OF INPUT BUFFER
    7E   0100 8F   3C  025D      MOVZWL  #WRK_C_INPBUFSIZ,-(SP)       ;PUSH SIZE OF INPUT BUFFER
                       0262
             6E    DF  0262      PUSHAL  (SP)                         ;PUSH ADDRESS OF RETURN LENGTH
          0C AE    DF  0264      PUSHAL  12(SP)                       ;PUSH ADDRESS OF PROMPT STRING DESC
          08 AE    DF  0267      PUSHAL  8(SP)                        ;PUSH ADDRESS OF RETURN DESC
       65    03    FB  026A      CALLS   #3,(R5)                      ;GET THE INPUT
             53  8ED0  026D      POPL    R3                           ;GET INPUT SIZE
             52  8ED0  0270      POPL    R2                           ;GET INPUT ADDRESS
          16 50    E9  0273      BLBC    R0,90$                       ;SIGNAL ANY ERROR
                       0276
                       0276  ;
                       0276  ; INSERT AN EOL MARKER
                       0276  ;
           6243    94  0276      CLRB    (R2)[R3]                     ;INSERT EOL CHARACTER
                       0279
                       0279  ;
                       0279  ; IF THE RECORD IS A FULL LINE COMMENT, REPROMPT NOW
                       0279  ;
       62    21    91  0279      CMPB    #^A/!/,(R2)                  ;IS FIRST CHAR AN EXCLAMATION MARK?
             D2    13  027C      BEQL    20$                          ;REPROMPT IF SO
                       027E
                       027E  ;
                       027E  ; IF THE PREVIOUS RECORD ENDED WITH TRAILING SPACES OR TABS,
                       027E  ; INSERT A SPACE AT THE FRONT OF THE CURRENT INPUT RECORD SO
                       027E  ; THAT PARAMETERS ARE DELIMITED PROPERLY.
                       027E  ;
             09    E5  027E      BBCC    #WRK_V_TRAILSPC,-            ;IF CLR, NO TRAILING SPACE SEEN
        06 F0 AA       0280              WRK_Q_FLAGS(R10),80$         ;
       50    20    90  0283      MOVB    #^A/ /,R0                    ;SET SPACE CHARACTER
           FD77'   30  0286      BSBW    DCL$BACKUPCHAR               ;APPEND TO FRONT OF INPUT BUFFER
                       0289
                       0289  ;
                       0289  ; EXIT FROM INPUT ROUTINE
                       0289  ;
       50    01    D0  0289  80$:    MOVL    #1,R0                        ;RETURN SUCCESS
       5E    29    C0  028C  90$:    ADDL    #ENT_K_MAX_PROMPT+1+8,SP;RESTORE THE STACK
       52    8E    7D  028F      MOVQ    (SP)+,R2                     ;RESTORE REGISTERS
       54    8E    7D  0292      MOVQ    (SP)+,R4                     ;
          0B 50    E8  0295      BLBS    R0,100$                      ;SKIP IF SUCCESS
    0000'8F    50  B1  0298      CMPW    R0,#RMS$_EOF&^XFFFF          ;EOF STATUS?
             05    13  029D      BEQL    110$                         ;YES, RETURN GENERIC EOF
          F9AE DA  16  029F      JSB     @WRK_L_ERRORRTN(R10)         ;CALL ERROR HANDLER
                   05  02A3  100$:   RSB
                       02A4
50   00000000'8F   D0  02A4  110$:   MOVL    #RMS$_EOF,R0                 ;RETURN GENERIC EOF
                   05  02AB          RSB
                       02AC
                       02AC          .END
```

```
                                                    H 12
DCLPARSE                    - PARSE A DCL COMMAND                  15-SEP-1984 23:42:55  VAX/VMS Macro V04-00        Page  14
Symbol table                                                        4-SEP-1984 23:40:07  [DCL.SRC]DCLPARSE.MAR;1           (7)

CDU$UPGRADE_TABLE             ********   X   02         WRK_B_MINPARM               FFFFFFD1
CLI$_BUFOVF                 = 00038018                  WRK_B_PARMCNT               FFFFFFCE
CLI$_CMDSEG                 = 00038248                  WRK_B_PARMSUM               FFFFFFCF
CLI$_NOCOMD                 = 000380B0                  WRK_B_RECALLCNT             FFFFFFC5
CLI$_NORMAL                 = 00030001                  WRK_B_VALLEV                FFFFFFC4
CLI$_STRTOOLNG              = 000388FA                  WRK_B_VERBTYP               FFFFFFC2
CTL$GL_CLINTOWN               ********   X   02         WRK_C_INPBUFSIZ           = 00000100
CTL$GL_DCLPRSOWN              ********   X   02         WRK_C_LENGTH                FFFFF486
CTRLZ                       = 0000001A                  WRK_G_BUFFER                FFFFF492
DCL$BACKUPCHAR                ********   X   02         WRK_G_INPBUF                FFFFF896
DCL$DCLPARSE                  0000000A RG    02         WRK_G_RESULT                FFFFF9B6
DCL$GENEOL                    ********   X   02         WRK_K_LENGTH                FFFFF486
DCL$GETOKEN                   ********   X   02         WRK_L_CHARPTR               FFFFF48E
DCL$MARK                      ********   X   02         WRK_L_DISALLOW              FFFFFFE6
DCL$MARKEDTOKEN               ********   X   02         WRK_L_ERRORRTN              FFFFF9AE
DCL$PARSE_COMMAND             ********   X   02         WRK_L_EXPANDPTR             FFFFF486
DCL$SEARCH_VERB               ********   X   02         WRK_L_IMAGE                 FFFFFFE2
DCL$USER_INPUT                0000021F RG    02         WRK_L_MARKPTR               FFFFF48A
DCL_B_FLAGS                   0000008C                  WRK_L_PAROUT                FFFFFFD2
DCL_B_PARAM                   0000008F                  WRK_L_PMPTADDR              FFFFF9A2
DCL_C_SIZE                    00000090                  WRK_L_PROMPTRTN             FFFFF9A6
DCL_K_SIZE                    00000090                  WRK_L_PROPTR                FFFFFFC6
DCL_L_DEFADDR                 00000088                  WRK_L_QUABLK                FFFFFFCA
DCL_L_ENTITY                  00000040                  WRK_L_READRTN               FFFFF9AA
DCL_L_FREEVM                  00000080                  WRK_L_RECALLPTR             FFFFFFEA
DCL_L_GETVM                   0000007C                  WRK_L_RSLEND                FFFFFFB6
DCL_L_PRMLIM                  00000000                  WRK_L_RSLNXT                FFFFFFBA
DCL_L_QUAL                    00000078                  WRK_L_SAVAP                 FFFFFFF8
DCL_L_TOKEN                   0000005C                  WRK_L_SAVFP                 FFFFFFFC
DCL_W_BUFLEN                  0000008D                  WRK_L_SAVSP                 FFFFFFF4
DCL_W_DEFLEN                  00000084                  WRK_L_SIGNALRTN             FFFFFFD6
DEALLOC_OWN                   0000020A R     02         WRK_L_SPECRTN               FFFFF9B2
DEALLOC_WRK                   000001EB R     02         WRK_L_TAB_VEC               FFFFFFDE
DEFAULT_PROMPT                00000000 R     02         WRK_L_VERB                  FFFFFFBE
DSC$A_POINTER                 ********   X   02         WRK_M_USRMODE             = 00002000
DSC$W_LENGTH                  ********   X   02         WRK_V_COMMAND             = 00000001
ENT_K_MAX_PROMPT            = 00000020                  WRK_V_TRAILSPC            = 00000009
ERROR                         00000177 R     02         WRK_W_FLAGS                 FFFFFFF0
ERRORMSG                      00000181 R     02         WRK_W_FLAGS2                FFFFFFF2
EXIT                          00000171 R     02         WRK_W_IMGCHAN               FFFFFFEE
INT_L_CONTINRTN             = 00000008                  WRK_W_PMPTLEN               FFFFF99E
INT_L_ENTADDR               = 0000000C
INT_L_FREEVM                = 00000014
INT_L_GETVM                 = 00000010
INT_L_LIST                  = 00000018
INT_L_PROMPTRTN             = 00000004
INT_L_TABLES                = 00000004
INT_W_PMPTLEN               = 0000000C
LIB$SIGNAL                    ********   X   02
NORMAL_EXIT                   0000016A R     02
PARSE_VERB_QUALS              0000015A R     02
PSL$C_USER                  = 00000003
RESTORE_SUPER_MODE            000001CD R     02
RMS$_EOF                      ********   X   02
STS$V_INHIB_MSG             = 0000001C
WRK_B_CMDOPT                  FFFFFFC3
WRK_B_MAXPARM                 FFFFFFD0
```

```
                                              +----------------+
                                              ! 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                       000002AC  (   684.)  02 (  2.)  NOPIC     USR     CON    REL     LCL NOSHR   EXE   RD   NOWRT NOVEC BYTE

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

Phase                      Page faults    CPU Time       Elapsed Time
-----                      -----------    --------       ------------
Initialization                      15    00:00:00.05    00:00:02.30
Command processing                  96    00:00:00.67    00:00:09.82
Pass 1                             285    00:00:10.06    00:00:28.80
Symbol table sort                    0    00:00:01.43    00:00:02.25
Pass 2                             118    00:00:02.04    00:00:06.73
Symbol table output                 11    00:00:00.11    00:00:00.57
Psect synopsis output                2    00:00:00.03    00:00:00.03
Cross-reference output               0    00:00:00.00    00:00:00.00
Assembler run totals               527    00:00:14.40    00:00:50.51

The working set limit was 1200 pages.
53043 bytes (104 pages) of virtual memory were used to buffer the intermediate code.
There were 50 pages of symbol table space allocated to hold 950 non-local and 28 local symbols.
552 source lines were read in Pass 1, producing 18 object records in Pass 2.
24 pages of virtual memory were used to define 18 macros.

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

Macro library name                        Macros defined
------------------                        --------------
_$255$DUA28:[SYSLIB]SYSBLDMLB.MLB;1                    0
_$255$DUA28:[DCL.OBJ]DCL.MLB;1                         7
_$255$DUA28:[SYS.OBJ]LIB.MLB;1                         0
_$255$DUA28:[SYSLIB]STARLET.MLB;2                      6
TOTALS (all libraries)                                13

1047 GETS were required to define 13 macros.

There were no errors, warnings or information messages.

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

0069 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

INTDEF
SDL

COMMAND
LIS

DESCRVAL
LIS

CONVERT
LIS

CLIMAC
MAR

CLIMSG
LIS

CONNECT
LIS

DCLPARSE
LIS

CHARMANIP
LIS

EXAMDEP
LIS

EXIT
LIS

INTIMAGES
MAR

DCXSTART
LIS

CLIGBL
LIS

DISALLOW
LIS

CLINT
LIS

CANCEL
LIS