```
B  7
```

```
MMM         MMM     CCCCCCCCCCCC  LLL              DDDDDDDDDDDD     RRRRRRRRRRRR
MMM         MMM     CCCCCCCCCCCC  LLL              DDDDDDDDDDDD     RRRRRRRRRRRR
MMM         MMM     CCCCCCCCCCCC  LLL              DDDDDDDDDDDD     RRRRRRRRRRRR
MMMMMM   MMMMMM  CCC              LLL              DDD         DDD  RRR         RRR
MMMMMM   MMMMMM  CCC              LLL              DDD         DDD  RRR         RRR
MMMMMM   MMMMMM  CCC              LLL              DDD         DDD  RRR         RRR
MMM   MMM   MMM  CCC              LLL              DDD         DDD  RRR         RRR
MMM   MMM   MMM  CCC              LLL              DDD         DDD  RRR         RRR
MMM   MMM   MMM  CCC              LLL              DDD         DDD  RRR         RRR
MMM         MMM  CCC              LLL              DDD         DDD  RRRRRRRRRRRR
MMM         MMM  CCC              LLL              DDD         DDD  RRRRRRRRRRRR
MMM         MMM  CCC              LLL              DDD         DDD  RRRRRRRRRRRR
MMM         MMM  CCC              LLL              DDD         DDD  RRR   RRR
MMM         MMM  CCC              LLL              DDD         DDD  RRR   RRR
MMM         MMM  CCC              LLL              DDD         DDD  RRR   RRR
MMM         MMM  CCC              LLL              DDD         DDD  RRR      RRR
MMM         MMM  CCC              LLL              DDD         DDD  RRR      RRR
MMM         MMM  CCC              LLL              DDD         DDD  RRR      RRR
MMM         MMM     CCCCCCCCCCCC  LLLLLLLLLLLLLLL  DDDDDDDDDDDD     RRR         RRR
MMM         MMM     CCCCCCCCCCCC  LLLLLLLLLLLLLLL  DDDDDDDDDDDD     RRR         RRR
MMM         MMM     CCCCCCCCCCCC  LLLLLLLLLLLLLLL  DDDDDDDDDDDD     RRR         RRR
```

```
_$2

Val
---
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
```

FILEID**LALOAD

G 10

```
LL            AAAAAA    LL            000000      AAAAAA    DDDDDDDD
LL            AAAAAA    LL            000000      AAAAAA    DDDDDDDD
LL          AA      AA  LL          00      00  AA      AA  DD      DD
LL          AA      AA  LL          00      00  AA      AA  DD      DD
LL          AA      AA  LL          00      00  AA      AA  DD      DD
LL          AA      AA  LL          00      00  AA      AA  DD      DD
LL          AA      AA  LL          00      00  AA      AA  DD      DD
LL          AA      AA  LL          00      00  AA      AA  DD      DD
LL          AAAAAAAAAA  LL          00      00  AAAAAAAAAA  DD      DD
LL          AAAAAAAAAA  LL          00      00  AAAAAAAAAA  DD      DD
LL          AA      AA  LL          00      00  AA      AA  DD      DD      ....
LL          AA      AA  LL          00      00  AA      AA  DD      DD      ....
LLLLLLLLLL  AA      AA  LLLLLLLLLL    000000    AA      AA  DDDDDDDD        ....
LLLLLLLLLL  AA      AA  LLLLLLLLLL    000000    AA      AA  DDDDDDDD        ....

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
```

LAL
V04

Table of contents

```
0000            .TITLE  LALOAD
0000            .IDENT  'V04-000'
0000
0000    ;
0000    ;****************************************************************************
0000    ;*                                                                          *
0000    ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
0000    ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
0000    ;*  ALL RIGHTS RESERVED.                                                    *
0000    ;*                                                                          *
0000    ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
0000    ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0000    ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
0000    ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
0000    ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
0000    ;*  TRANSFERRED.                                                            *
0000    ;*                                                                          *
0000    ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
0000    ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0000    ;*  CORPORATION.                                                            *
0000    ;*                                                                          *
0000    ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
0000    ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
0000    ;*                                                                          *
0000    ;*                                                                          *
0000    ;****************************************************************************
0000    ;
0000
0000    ;++
0000    ; FACILITY:     LPA-11 UTILITY PROGRAMS
0000    ;
0000    ; ABSTRACT:
0000    ;       THIS PROGRAM PROVIDES AN OPERATOR INTERFACE TO LOAD LPA-11
0000    ;       MICROCODE.  THIS PROGRAM, READS OPERATOR COMMANDS AND SENDS
0000    ;       LOAD REQUESTS TO THE LPA-11 LOADER PROCESS.
0000    ;
0000    ; ENVIRONMENT:  USER MODE
0000    ;
0000    ; AUTHOR:  STEVE BECKHARDT,     CREATION DATE:  9-OCT-78
0000    ;
0000    ; MODIFIED BY:
0000    ;
0000    ;       , : VERSION
0000    ; 01    -
0000    ;--
```

```
                     0000     47         .SBTTL  DECLARATIONS
                     0000     48 ;
                     0000     49 ; INCLUDE FILES:
                     0000     50 ;
                     0000     51
                     0000     52 ;
                     0000     53 ; MACROS:
                     0000     54 ;
                     0000     55
                     0000     56 ;
                     0000     57 ; EQUATED SYMBOLS:
                     0000     58 ;
          00000061   0000     59 LOWER_A = ^X61                  ; LOWERCASE A
                     0000     60
          0000007A   0000     61 LOWER_Z = ^X7A                  ; LOWERCASE Z
                     0000     62
                     0000     63 ;
                     0000     64 ; OWN STORAGE:
                     0000     65 ;
                     0000     66
                 00000000     67         .PSECT  LPADATA,LONG
                     0000     68
                     0000     69 INFAB:  $FAB    FAC = GET,-
                     0000     70                 FNM = <SYS$INPUT>,-
                     0000     71                 RAT = CR
                     0050     72
                     0050     73 OUTFAB: $FAB    FAC = PUT,-
                     0050     74                 FNM = <SYS$OUTPUT>,-
                     0050     75                 RAT = CR
                     00A0     76
                     00A0     77 INRAB:  $RAB    FAB = INFAB,-
                     00A0     78                 PBF = PROMPT,-
                     00A0     79                 PSZ = PROMPTSZ,-
                     00A0     80                 ROP = <CCO,PMT>,-
                     00A0     81                 UBF = BUFFER,-
                     00A0     82                 USZ = 132
                     00E4     83
                     00E4     84 OUTRAB: $RAB    FAB = OUTFAB
                     0128     85
                     0128     86
                     0128     87 DEVDESC:                        ; DEVICE DESCRIPTOR
          00000130   0128     88         .BLKQ   1
                     0130     89
                     0130     90 DEVCHAN:                        ; DEVICE CHANNEL
          00000132   0130     91         .BLKW   1
                     0132     92
          00000232   0132     93 BUFFER: .BLKB   256             ; BUFFER FOR INPUT AND OUTPUT
```

```
                          00000000     95              .PSECT  LPACODE,NOWRT,LONG
                              0000     96
                              0000     97 BUFFERDSC:                            ; BUFFER DESCRIPTOR
                    00000100  0000     98              .LONG   256
                    00000132' 0004     99              .LONG   BUFFER
                              0008    100
                              0008    101 DEFDEVDESC:                           ; DEFAULT DEVICE NAME DESCRIPTOR
                    00000004' 0008    102              .LONG   DEFDEVSZ
                    00000028' 000C    103              .LONG   DEFDEV
                              0010    104
                    0000000D' 0010    105 SWTBL:   .LONG   MRSWSIZ ; SWITCH TABLE
                    00000041' 0014    106              .LONG   MRSW
                    0000000E' 0018    107              .LONG   ADSWSIZ
                    0000004E' 001C    108              .LONG   ADSW
                    0000000E' 0020    109              .LONG   DASWSIZ
                    0000005C' 0024    110              .LONG   DASW
                              0028    111
                 30 41 41 4C  0028    112 DEFDEV: .ASCII  'LAA0'            ; DEFAULT DEVICE NAME
                    00000004  002C    113 DEFDEVSZ = .-DEFDEV
                              002C    114
         3E 44 41 4F 4C 41 4C  002C    115 PROMPT: .ASCII  'LALOAD>'         ; PROMPT STRING
                    00000007  0033    116 PROMPTSZ = .-PROMPT
                              0033    117
                              0033    118 INVSWMSG:                         ; INVALID SWITCH MESSAGE
74 69 77 73 20 64 69 6C 61 76 6E 49  0033    119         .ASCII  'Invalid switch'
                       68 63  003F
                    0000000E  0041    120 INVSWMSGSZ = .-INVSWMSG
                              0041    121
53 45 55 51 45 52 5F 49 54 4C 55 4D  0041    122 MRSW:    .ASCII  'MULTI_REQUEST' ; SWITCH NAMES
                          54  004D
                    0000000D  004E    123 MRSWSIZ = .-MRSW
                              004E    124
54 49 47 49 44 5F 47 4F 4C 41 4E 41  004E    125 ADSW:    .ASCII  'ANALOG_DIGITAL'
                       4C 41  005A
                    0000000E  005C    126 ADSWSIZ = .-ADSW
                              005C    127
4C 41 4E 41 5F 4C 41 54 49 47 49 44  005C    128 DASW:    .ASCII  'DIGITAL_ANALOG'
                       47 4F  0068
                    0000000E  006A    129 DASWSIZ = .-DASW
                              006A    130
                    2F 20 3A  006A    131 TRMS:    .ASCII  ': /'             ; TERMINATORS TABLE
                    00000003  006D    132 NTRMS = .-TRMS
```

```
                        006D          .SBTTL  MAIN PROGRAM
                        006D  ;++
                        006D  ; FUNCTIONAL DESCRIPTION:
                        006D  ;
                        006D  ;       THIS IS THE MAIN PROGRAM FOP THIS PROCESS.  IT PROMPTS FOR
                        006D  ;       INPUT, READS THE INPUT, PARSES IT, AND THEN CALLS THE ROUTINE
                        006D  ;       WHICH SENDS THE LOAD REQUEST OVER A MAILBOX.
                        006D  ;
                        006D  ; CALLING SEQUENCE:
                        006D  ;
                        006D  ;       ENTERED WHEN PROGRAM IS STARTED
                        006D  ;
                        006D  ; INPUT PARAMETERS:
                        006D  ;
                        006D  ;       NONE
                        006D  ;
                        006D  ; OUTPUT PARAMETERS:
                        006D  ;
                        006D  ;       NONE
                        006D  ;
                        006D  ;--
                        006D
                  01FC  006D          .ENTRY  START, ^M<R2,R3,R4,R5,R6,R7,R8>
                        006F
                        006F          ; OPEN INPUT AND OUTPUT FILES AND CONNECT RABS
                        006F          $OPEN           FAB = INFAB
            49 50   E9  007C          BLBC    R0,13$
                        007F          $OPEN           FAB = OUTFAB
            39 50   E9  008C          BLBC    R0,13$
                        008F          $CONNECT        RAB = INRAB
            29 50   E9  009C          BLBC    R0,13$
                        009F          $CONNECT        RAB = OUTRAB
            19 50   E9  00AC          BLBC    R0,13$
                        00AF
                        00AF  10$:    ; GET AN INPUT LINE
                        00AF          $GET    RAB = INRAB
   00000000'8F  50  D1  00BC          CMPL    R0,#RMS$_EOF                ; END OF FILE?
                06  13  00C3          BEQL    14$                         ; YES
             06 50  E8  00C5          BLBS    R0,15$                      ; SUCCESS
                        00C8
              00E0  31  00C8  13$:    BRW     100$                        ; ERROR
              00DA  31  00CB  14$:    BRW     98$                         ; EXIT NORMALLY
                        00CE
                        00CE  15$:    ; GET SIZE OF INPUT LINE
   50   000000C2'E  B0  00CE          MOVW    INRAB+RAB$W_RSZ,R0
                D8  13  00D5          BEQL    10$                         ; EMPTY LINE - REPROMPT
                        00D7
                        00D7          ; SKIP LEADING BLANKS
00000132'EF   50  20  3B  00D7        SKPC    #^A' ',R0,BUFFER
                  CE  13  00DF        BEQL    10$                         ; EMPTY LINE - REPROMPT
              52  50  7D  00E1        MOVQ    R0,R2                       ; MOVE SIZE AND ADDRESS TO R2,R3
              54  50  7D  00E4        MOVQ    R0,R4                       ; AND ALSO TO R4,R5
                00C2  30  00E7        BSBW    CVTUPC                      ; CONVERT LINE TO UPPER CASE
                          00EA
                          00EA  20$:  ; NOW LOOP LOOKING FOR END OF NAME
    FF7A CF   03  63  3A  00EA        LOCC    (R3),#NTRMS,TRMS            ; LOOK FOR A TERMINATOR
                  05  12  00F0        BNEQ    30$                         ; FOUND ONE
              53   D6 00F2           INCL    R3                      ; BUMP CHAR. POINTER
           F3 52   F5 00F4           SOBGTR  R2,20$                  ; TRY NEXT CHAR.
                      00F7
                      00F7      30$: ; HAVE A TERMINATOR OR REACHED END OF LINE
           54 52   C2 00F7           SUBL    R2,R4                   ; R4 CONTAINS LENGTH OF NAME
                      00FA
                      00FA           ; LOOK FOR SWITCH
              58   D4 00FA           CLRL    R8                      ; INIT. FOR SEARCH THRU SWITCH TABLE
                      00FC                                           ; AND ALSO AS DEFAULT
        63 52 2F   3A 00FC           LOCC    #^A'/',R2,(R3)
              36   13 0100           BEQL    60$                     ; NO SWITCH - USE DEFAULT
                      0102
                      0102           ; HAVE SLASH.  R0,R1 CONTAIN REMAINING STRING DESC.
              51   D6 0102           INCL    R1                      ; MOVE PAST SLASH
              50   D7 0104           DECL    R0                      ; DEC. COUNT
              1E   13 0106           BEQL    55$                     ; ERROR - INVALID SWITCH
           56 50   7D 0108           MOVQ    R0,R6                   ; COPY DESC. INTO R6,R7
        61 50 20   3A 010B           LOCC    #^A' ',R0,(R1)          ; LOCATE END OF SWITCH
           56 50   C2 010F           SUBL    R0,R6                   ; CALCULATE LENGTH OF SWITCH
                      0112
                      0112      50$: ; COMPARE AGAINST NEXT ENTRY IN SWITCH TABLE
     52  FEF9 CF48 7D 0112           MOVQ    SWTBL[R8],R2            ; GET LEN. AND ADDR. OF SWITCH IN R2,R3
 63  52  00  67 56 2D 0118           CMPC5   R6,(R7),#0,R2,(R3)
              50   D5 011E           TSTL    R0                      ; IF R0 = 0, THEN SUBSTRING MATCH
              16   13 0120           BEQL    60$
        EC 58 03   F2 0122           AOBLSS  #3,R8,50$
                      0126
                      0126      55$: ; ERROR - INVAL'D SWITCH
 0000010C'EF  FF09 CF 9E 0126        MOVAB   INVSWMsG,OUTRAB+RAB$L_RBF        ; ADDRESS OF MESSAGE
    00000106'EF    0E B0 012F        MOVW    #INVSWMSGSZ,OUTRAB+RAB$W_RSZ     ; SIZE OF MESSAGE
              5D   11 0136           BRB     90$                     ; OUTPUT MESSAGE
                      0138
                      0138      60$: ; HAVE NAME DESCRIPTOR IN R4,R5; SWITCH VALUE IN R8
     51  00000128'EF 7E 0138         MOVAQ   DEVDESC,R1              ; GET ADDRESS OF DEVICE DESCRIPTOR
           61 54   7D 013F           MOVQ    R4,(R1)                 ; COPY INFO. INTO DEVICE DESC.
              54   D5 0142           TSTL    R4                      ; IS LENGTH NON-ZERO?
              05   12 0144           BNEQ    70$                     ; YES
        51 FEBE CF 7E 0146           MOVAQ   DEFDEVDESC,R1           ; NO, USE DEFAULT DESCRIPTOR
                      014B
                      014B      70$: ; ASSIGN A CHANNEL TO DEVICE
                      014B           $ASSIGN_S       DEVNAM = (R1),- ; DEVICE NAME
                      014B                           CHAN = DEVCHAN  ; CHANNEL
           14 50   E9 015C           BLBC    R0,80$                  ; ERROR
                      015F
                      015F           ; NOW SEND A REQUEST TO LOADER
        7E 58 01   C1 015F           ADDL3   #1,R8,-(SP)             ; PUSH MICROCODE TYPE
      00000130'EF  3F 0163           PUSHAW  DEVCHAN                 ; PUSH CHANNEL ADDRESS
    00000000'EF 02 FB 0169           CALLS   #2,LPA$$SNDLDRQ         ; SEND LOAD REQUEST (& DEASSIGN CHANNEL)
           32 50   E8 0170           BLBS    R0,95$                  ; SUCCESS, REPROMPT
                      0173
                      0173      80$: ; USER ERROR IN R0 - GET MESSAGE
                      0173           $GETMSG_S       MSGID = R0,-                  ; MESSAGE ID
                      0173                           MSGLEN = OUTRAB+RAB$W_RSZ,-   ; PLACE TO STORE MSG LEN
                      0173                           BUFADR = BUFFERDSC            ; BUFFER DESCR.
 0000010C'EF  00000132'EF  9E 018A   MOVAB   BUFFER,OUTRAB+RAB$L_RBF ; STORE BUFFER ADDRESS IN RAB
                      0195
                      0195      90$: ; OUTPUT ERROR MESSAGE
```

```
                    0195    248              $PUT     RAB = OUTRAB
      06 50    E9   01A2    249              BLBC     R0,100$                      ; ERROR
       FF07    31   01A5    250 95$:         BRW      10$                          ; SUCCESS - REPROMPT
                    01A8    251
                    01A8    252 98$:         ; EXIT WITH SUCCESS
  50    00'    3C   01A8    253              MOVZWL   S^#SS$_NORMAL,R0
                    01AB    254
               04   01AB    255 100$:        RET
```

```
                       01AC          .SBTTL  CVTUPC - CONVERT LINE TO UPPER CASE
                       01AC     ;++
                       01AC     ;FUNCTIONAL DESCRIPTION:
                       01AC     ;
                       01AC     ;       THIS ROUTINE CONVERTS THE INPUT LINE TO UPPER CASE
                       01AC     ;
                       01AC     ; CALLING SEQUENCE:
                       01AC     ;
                       01AC     ;       BSBW
                       01AC     ;
                       01AC     ; INPUT PARAMETERS:
                       01AC     ;
                       01AC     ;       R0      CONTAINS SIZE OF LINE
                       01AC     ;       R1      CONTAINS ADDRESS OF LINE
                       01AC     ;
                       01AC     ; OUTPUT PARAMETERS:
                       01AC     ;
                       01AC     ;       NONE
                       01AC     ;
                       01AC     ; SIDE EFFECTS:
                       01AC     ;
                       01AC     ;       R0,R1 ARE NOT PRESERVED
                       01AC     ;--
                       01AC
61 8F      61     91   01AC     CVTUPC: CMPB    (R1),#LOWER_A           ; IS IT BEFORE LOWERCASE A?
           09     1F   01B0             BLSSU   20$                     ; YES
7A 8F      61     91   01B2             CMPB    (R1),#LOWER_Z           ; IS IS AFTER LOWERCASE Z?
           03     1A   01B6             BGTRU   20$                     ; YES
   61      20     8A   01B8             BICB    #^X20,(R1)              ; CLEAR LOWERCASE BIT
           51     D6   01BB     20$:    INCL    R1                      ; MOVE TO NEXT CHARACTER
      EC   50     F5   01BD             SOBGTR  R0,CVTUPC               ; REPEAT
                  05   01C0             RSB
                       01C1
                       01C1
                       01C1
                       01C1             .END    START
```

```
$$.TAB                          = 000000E4 R      01
$$.TABEND                       = 00000125 R      01
$$.TMP                          = 00000000
$$.TMP1                         = 00000001
$$.TMP2                         = 000000CF
$$.TMPX                         = 00000009 R      03
$$.TMPX1                        = 0000000A
$$T1                            = 00000001
ADSW                              0000004E R      04
ADSWSIZ                         = 0000000E
BUFFER                            00000132 R      01
BUFFERDSC                         00000000 R      04
CVTUPC                            000001AC R      04
DASW                              0000005C R      04
DASWSIZ                         = 0000000E
DEFDEV                            00000028 R      04
DEFDEVDESC                        00000008 R      04
DEFDEVSZ                        = 00000004
DEVCHAN                           00000130 R      01
DEVDESC                           00000128 R      01
FAB$B_FNS                       = 00000034
FAB$C_BID                       = 00000003
FAB$C_BLN                       = 00000050
FAB$C_SEQ                       = 00000000
FAB$C_VAR                       = 00000002
FAB$L_ALQ                       = 00000010
FAB$L_FNA                       = 0000002C
FAB$L_FOP                       = 00000004
FAB$V_CHAN_MODE                 = 00000002
FAB$V_CR                        = 00000001
FAB$V_FILE_MODE                 = 00000004
FAB$V_GET                       = 00000001
FAB$V_LNM_MODE                  = 00000000
FAB$V_PUT                       = 00000000
FAB$W_GBC                       = 00000048
INFAB                             00000000 R      01
INRAB                             000000A0 R      01
INVSWMSG                          00000033 R      04
INVSWMSGSZ                      = 0000000E
LOWER_A                         = 00000061
LOWER_Z                         = 0000007A
LPA$$SNDLDRQ                      ********    X   04
MRSW                              00000041 R      04
MRSWSIZ                         = 0000000D
NTRMS                           = 00000003
OUTFAB                            00000050 R      01
OUTRAB                            000000E4 R      01
PROMPT                            0000002C R      04
PROMPTSZ                        = 00000007
RAB$B_PSZ                       = 00000034
RAB$B_RAC                       = 0000001E
RAB$C_BID                       = 00000001
RAB$C_BLN                       = 00000044
RAB$C_SEQ                       = 00000000
RAB$L_CTX                       = 00000018
RAB$L_PBF                       = 00000030
RAB$L_RBF                       = 00000028

RAB$L_ROP                       = 00000004
RAB$V_CCO                       = 0000001F
RAB$V_PMT                       = 0000001E
RAB$W_RSZ                       = 00000022
RMS$_EOF                          ********    X   04
SS$_NORMAL                        ********    X   04
START                             0000006D RG     04
SWTBL                             00000010 R      04
SYS$ASSIGN                        ********   GX   04
SYS$CONNECT                       ********   GX   04
SYS$GET                           ********   GX   04
SYS$GETMSG                        ********   GX   04
SYS$OPEN                          ********   GX   04
SYS$PUT                           ********   GX   04
TRMS                              0000006A R      04
```

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

PSECT name                     Allocation          PSECT No.  Attributes
----------                     ----------          ---------  ----------
.  ABS  .                      00000000  (     0.)  00 (  0.)  NOPIC   USR   CON   ABS   LCL NOSHR NOEXE NORD  NOWRT NOVEC BYTE
LPADATA                        00000232  (   562.)  01 (  1.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD    WRT NOVEC LONG
$ABS$                          00000000  (     0.)  02 (  2.)  NOPIC   USR   CON   ABS   LCL NOSHR   EXE   RD    WRT NOVEC BYTE
$RMSNAM                        00000013  (    19.)  03 (  3.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD    WRT NOVEC BYTE
LPACODE                        000001C1  (   449.)  04 (  4.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD  NOWRT NOVEC LONG

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

Phase                    Page faults    CPU Time       Elapsed Time
-----                    -----------    --------       ------------
Initialization                  29    00:00:00.09    00:00:00.37
Command processing             100    00:00:00.47    00:00:01.05
Pass 1                         197    00:00:05.02    00:00:10.38
Symbol table sort                0    00:00:00.35    00:00:00.58
Pass 2                          66    00:00:01.11    00:00:02.45
Symbol table output             10    00:00:00.07    00:00:00.08
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           406    00:00:07.15    00:00:14.94

The working set limit was 1050 pages.
24588 bytes (49 pages) of virtual memory were used to buffer the intermediate code.
There were 20 pages of symbol table space allocated to hold 344 non-local and 16 local symbols.
292 source lines were read in Pass 1, producing 21 object records in Pass 2.
27 pages of virtual memory were used to define 21 macros.

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

Macro library name                        Macros defined
------------------                        --------------
_$255$DUA28:[SYSLIB]STARLET.MLB;2                18

593 GETS were required to define 18 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:LALOAD/OBJ=OBJ$:LALOAD MSRC$:LALOAD/UPDATE=(ENH$:LALOAD)
```

0233 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

MELDR

MDL32

LALOAD
MAP

MDL32
MAP

XFLOADER
MAP

LALOADER
LIS

XFLOADER
LIS

LADAMCODE
LIS

LALOAD
LIS

MARBLI

MARBLI
MAP

LALOADER
MAP

LAMRMCODE
LIS

LAADMCODE
LIS

MARBLI
LIS

LADMQT
LIS