

[illegible]

```

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
LLLLLLLLLLLL IIIIII          SSSSSSSS
LLLLLLLLLLLL IIIIII          SSSSSSSS

```


(2)	47	DECLARATIONS
(3)	134	MAIN PROGRAM
(4)	257	CVTUPC - CONVERT LINE TO UPPER CASE

```
0000 1      .TITLE  LALOAD
0000 2      .IDENT  'V04-000'
0000 3
0000 4
0000 5 *****
0000 6 *
0000 7 *  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 8 *  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 9 *  ALL RIGHTS RESERVED.
0000 10 *
0000 11 *  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 12 *  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 13 *  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 14 *  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 15 *  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 16 *  TRANSFERRED.
0000 17 *
0000 18 *  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 19 *  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 20 *  CORPORATION.
0000 21 *
0000 22 *  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 23 *  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 24 *
0000 25 *
0000 26 *****
0000 27
0000 28
0000 29 ++
0000 30 : FACILITY:      LPA-11 UTILITY PROGRAMS
0000 31
0000 32 : ABSTRACT:
0000 33 :   THIS PROGRAM PROVIDES AN OPERATOR INTERFACE TO LOAD LPA-11
0000 34 :   MICROCODE.  THIS PROGRAM, READS OPERATOR COMMANDS AND SENDS
0000 35 :   LOAD REQUESTS TO THE LPA-11 LOADER PROCESS.
0000 36
0000 37 : ENVIRONMENT:  USER MODE
0000 38
0000 39 : AUTHOR:  STEVE BECKHARDT,      CREATION DATE:  9-OCT-78
0000 40
0000 41 : MODIFIED BY:
0000 42
0000 43 :   : VERSION
0000 44 : 01  -
0000 45 :--
```


DECLARATIONS

```
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
0000007A 0000 60
0000007A 0000 61 LOWER_Z = ^X7A        ; LOWERCASE Z
0000 62
0000 63      :
0000 64      : OWN STORAGE:
0000 65      :
0000 66
00000000 0000 67      .PSECT  LPADATA, LONG
0000 68
0000 69 INFAB:  $FAB      FAC = GET, -
0000 70                      FNM = <SYS$INPUT>, -
0000 71                      RAT = CR
0000 72
0000 73 OUTFAB: $FAB      FAC = PUT, -
0000 74                      FNM = <SYS$OUTPUT>, -
0000 75                      RAT = CR
0000 76
0000 77 INRAB:  $RAB      FAB = INFAB, -
0000 78                      PBF = PROMPT, -
0000 79                      PSZ = PROMPT$Z, -
0000 80                      ROP = <CCO, PMT>, -
0000 81                      UBF = BUFFER, -
0000 82                      USZ = 132
0000 83
0000 84 OUTRAB: $RAB      FAB = OUTFAB
0000 85
0000 86
00000130 0128 87 DEVDESC:          ; DEVICE DESCRIPTOR
00000130 0128 88      .BLKQ      1
00000130 0130 89
00000132 0130 90 DEVCHAN:          ; DEVICE CHANNEL
00000132 0130 91      .BLKW      1
00000132 0132 92
00000232 0132 93 BUFFER: .BLKB      256      ; BUFFER FOR INPUT AND OUTPUT
```

DECLARATIONS

```
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 'LAAO' ; 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 INVSWMMSG: ; 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 INVSWMMSGSZ = .-INVSWMMSG
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
```


MAIN PROGRAM

```
006D 134 .SBTTL MAIN PROGRAM
006D 135 :++
006D 136 : FUNCTIONAL DESCRIPTION:
006D 137 :
006D 138 : THIS IS THE MAIN PROGRAM FOR THIS PROCESS. IT PROMPTS FOR
006D 139 : INPUT, READS THE INPUT, PARSES IT, AND THEN CALLS THE ROUTINE
006D 140 : WHICH SENDS THE LOAD REQUEST OVER A MAILBOX.
006D 141 :
006D 142 : CALLING SEQUENCE:
006D 143 :
006D 144 : ENTERED WHEN PROGRAM IS STARTED
006D 145 :
006D 146 : INPUT PARAMETERS:
006D 147 :
006D 148 : NONE
006D 149 :
006D 150 : OUTPUT PARAMETERS:
006D 151 :
006D 152 : NONE
006D 153 :
006D 154 :--
006D 155 :
01FC 006D 156 .ENTRY START, ^M<R2,R3,R4,R5,R6,R7,R8>
006F 157
006F 158 : OPEN INPUT AND OUTPUT FILES AND CONNECT RABS
006F 159 $OPEN FAB = INFAB
49 50 E9 007C 160 BLBC R0,13$
007F 161 $OPEN FAB = OUTFAB
39 50 E9 008C 162 BLBC R0,13$
008F 163 $CONNECT RAB = INRAB
29 50 E9 009C 164 BLBC R0,13$
009F 165 $CONNECT RAB = OUTRAB
19 50 E9 00AC 166 BLBC R0,13$
00AF 167
00AF 168 10$: : GET AN INPUT LINE
00AF 169 $GET RAB = INRAB
00000000'8F 50 D1 00BC 170 CMPL R0,#RMS$_EOF : END OF FILE?
06 50 E8 00C3 171 BEQL 14$ : YES
00E0 31 00C8 172 BLBS R0,15$ : SUCCESS
00DA 31 00CB 173 :
00CE 174 13$: BRW 100$ : ERROR
00CE 175 14$: BRW 98$ : EXIT NORMALLY
00CE 176
00CE 177 15$: : GET SIZE OF INPUT LINE
50 000000C2'EF B0 00CE 178 MOVW INRAB+RAB$_RSZ,R0
D8 13 00D5 179 BEQL 10$ : EMPTY LINE - REPROMPT
00D7 180
00D7 181 : SKIP LEADING BLANKS
00000132'EF 50 20 3B 00D7 182 SKPC #'A',R0,BUFFER
CE 13 00DF 183 BEQL 10$ : EMPTY LINE - REPROMPT
52 50 7D 00E1 184 MOVQ R0,R2 : MOVE SIZE AND ADDRESS TO R2,R3
54 50 7D 00E4 185 MOVQ R0,R4 : AND ALSO TO R4,R5
00C2 30 00E7 186 BSBW CVTUPC : CONVERT LINE TO UPPER CASE
00EA 187
00EA 188 20$: : NOW LOOP LOOKING FOR END OF NAME
FF7A CF 03 63 3A 00EA 189 LOCC (R3),#NTRMS,TRMS : LOOK FOR A TERMINATOR
05 12 00F0 190 BNEQ 30$ : FOUND ONE
```

MAIN PROGRAM

```

      53 D6 00F2 191
F3 52 F5 00F4 192
      00F7 193
      00F7 194 30$:
54 52 C2 00F7 195
      00FA 196
      00FA 197
      58 D4 00FA 198
      00FC 199
63 52 2F 3A 00FC 200
      36 13 0100 201
      0102 202
      0102 203
      51 D6 0102 204
      50 D7 0104 205
      1E 13 0106 206
      56 50 7D 0108 207
61 50 20 3A 010B 208
      56 50 C2 010F 209
      0112 210
      0112 211 50$:
63 52 52 FEF9 CF48 7D 0112 212
      00 67 56 2D 0118 213
      50 D5 011E 214
      16 13 0120 215
      EC 58 03 F2 0122 216
      0126 217
      0126 218 55$:
0000010C'EF FF09 CF 9E 0126 219
00000106'EF 0E B0 012F 220
      5D 11 0136 221
      0138 222
      0138 223 60$:
51 00000128'EF 7E 0138 224
      61 54 7D 013F 225
      54 D5 0142 226
      05 12 0144 227
      51 FEBE CF 7E 0146 228
      014B 229
      014B 230 70$:
      014B 231
      014B 232
      14 50 E9 015C 233
      015F 234
      015F 235
      7E 58 01 C1 015F 236
      00000130'EF 3F 0163 237
00000000'EF 02 FB 0169 238
      32 50 E8 0170 239
      0173 240
      0173 241 80$:
      0173 242
      0173 243
      0173 244
0000010C'EF 00000132'EF 9E 018A 245
      0195 246
      0195 247 90$:
      INCL R3 ; BUMP CHAR. POINTER
      SOBGTR R2,20$ ; TRY NEXT CHAR.
      ; HAVE A TERMINATOR OR REACHED END OF LINE
      SUBL R2,R4 ; R4 CONTAINS LENGTH OF NAME
      ; LOOK FOR SWITCH
      CLRL R8 ; INIT. FOR SEARCH THRU SWITCH TABLE
      ; AND ALSO AS DEFAULT
      LOCC #'A'/'',R2,(R3)
      BEQL 60$ ; NO SWITCH - USE DEFAULT
      ; HAVE SLASH. R0,R1 CONTAIN REMAINING STRING DESC.
      INCL R1 ; MOVE PAST SLASH
      DECL R0 ; DEC. COUNT
      BEQL 55$ ; ERROR - INVALID SWITCH
      MOVQ R0,R6 ; COPY DESC. INTO R6,R7
      LOCC #'A'/'',R0,(R1) ; LOCATE END OF SWITCH
      SUBL R0,R6 ; CALCULATE LENGTH OF SWITCH
      ; COMPARE AGAINST NEXT ENTRY IN SWITCH TABLE
      MOVQ SWTB[R8],R2 ; GET LEN. AND ADDR. OF SWITCH IN R2,R3
      CMPC5 R6,(R7),#0,R2,(R3)
      TSTL R0 ; IF R0 = 0, THEN SUBSTRING MATCH
      BEQL 60$
      AOBLS5 #3,R8,50$
      ; ERROR - INVALID SWITCH
      MOVAB INVSMSG,OUTRAB+RAB$RBF ; ADDRESS OF MESSAGE
      MOVW #INVSMSGSZ,OUTRAB+RAB$RSZ ; SIZE OF MESSAGE
      BRB 90$ ; OUTPUT MESSAGE
      ; HAVE NAME DESCRIPTOR IN R4,R5; SWITCH VALUE IN R8
      MOVAQ DEVDESC,R1 ; GET ADDRESS OF DEVICE DESCRIPTOR
      MOVQ R4,(R1) ; COPY INFO. INTO DEVICE DESC.
      TSTL R4 ; IS LENGTH NON-ZERO?
      BNEQ 70$ ; YES
      MOVAQ DEFDEVDESC,R1 ; NO, USE DEFAULT DESCRIPTOR
      ; ASSIGN A CHANNEL TO DEVICE
      $ASSIGN_S DEVNAM = (R1),- ; DEVICE NAME
      ; CHANNEL
      CHAN = DEVCHAN ; CHANNEL
      BLBC R0,80$ ; ERROR
      ; NOW SEND A REQUEST TO LOADER
      ADDL3 #1,R8,-(SP) ; PUSH MICROCODE TYPE
      PUSHAW DEVCHAN ; PUSH CHANNEL ADDRESS
      CALLS #2,LPA$$$SNDLDRQ ; SEND LOAD REQUEST (& DEASSIGN CHANNEL)
      BLBS R0,95$ ; SUCCESS, REPROMPT
      ; USER ERROR IN R0 - GET MESSAGE
      $GETMSG_S MSGID = R0,- ; MESSAGE ID
      ; PLACE TO STORE MSG LEN
      MSGLEN = OUTRAB+RAB$RSZ,- ; PLACE TO STORE MSG LEN
      BUFADR = BUFFERDSC ; BUFFER DESCR.
      MOVAB BUFFER,OUTRAB+RAB$RBF ; STORE BUFFER ADDRESS IN RAB
      ; OUTPUT ERROR MESSAGE
```


MAIN PROGRAM

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

CVTUPC - CONVERT LINE TO UPPER CASE

```
01AC 257 .SBTTL CVTUPC - CONVERT LINE TO UPPER CASE
01AC 258 :++
01AC 259 :FUNCTIONAL DESCRIPTION:
01AC 260 :
01AC 261 : THIS ROUTINE CONVERTS THE INPUT LINE TO UPPER CASE
01AC 262 :
01AC 263 : CALLING SEQUENCE:
01AC 264 :
01AC 265 : BSBW
01AC 266 :
01AC 267 : INPUT PARAMETERS:
01AC 268 :
01AC 269 : R0 CONTAINS SIZE OF LINE
01AC 270 : R1 CONTAINS ADDRESS OF LINE
01AC 271 :
01AC 272 : OUTPUT PARAMETERS:
01AC 273 :
01AC 274 : NONE
01AC 275 :
01AC 276 : SIDE EFFECTS:
01AC 277 :
01AC 278 : R0,R1 ARE NOT PRESERVED
01AC 279 :--
01AC 280
61 8F 61 91 01AC 281 CVTUPC: CMPB (R1),#LOWER_A ; IS IT BEFORE LOWERCASE A?
09 1F 01B0 282 BLSSU 20$ ; YES
7A 8F 61 91 01B2 283 CMPB (R1),#LOWER_Z ; IS IS AFTER LOWERCASE Z?
03 1A 01B6 284 BGTRU 20$ ; YES
61 20 8A 01B8 285 BICB #^X20,(R1) ; CLEAR LOWERCASE BIT
51 D6 01BB 286 20$: INCL R1 ; MOVE TO NEXT CHARACTER
EC 50 F5 01BD 287 SOBGTR R0,CVTUPC ; REPEAT
05 01C0 288 RSB
01C1 289
01C1 290
01C1 291
01C1 292 .END START
```


LALOAD
Symbol table

\$\$TAB	= 000000E4	R	01
\$\$TABEND	= 00000128	R	01
\$\$TMP	= 00000000		
\$\$TMP1	= 00000001		
\$\$TMP2	= 000000CF		
\$\$TMPX	= 00000009	R	03
\$\$TMPX1	= 0000000A		
\$\$T1	= 00000001		
ADSW	= 0000004E	R	04
ADSW\$IZ	= 0000000E		
BUFFER	= 00000132	R	01
BUFFERDSC	= 00000000	R	04
CVTUPC	= 000001AC	R	04
DASW	= 0000005C	R	04
DASW\$IZ	= 0000000E		
DEFDEV	= 00000028	R	04
DEFDEVDESC	= 00000008	R	04
DEFDEV\$Z	= 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
INVSWMMSG	= 00000033	R	04
INVSWMMSG\$Z	= 0000000E		
LOWER_A	= 00000061		
LOWER_Z	= 0000007A		
LPASS\$NDLDRQ	*****	X	04
MRSW	= 00000041	R	04
MRSW\$IZ	= 0000000D		
NTRMS	= 00000003		
OUTFAB	= 00000050	R	01
OUTRAB	= 000000E4	R	01
PROMPT	= 0000002C	R	04
PROMPT\$Z	= 00000007		
RAB\$B_PSZ	= 00000034		
RAB\$B_RAC	= 0000001E		
RAB\$C_BID	= 00000001		
RAB\$C_BLN	= 00000044		
RAB\$C_SEQ	= 00000000		
RAB\$C_CTX	= 00000018		
RAB\$C_PBF	= 00000030		
RAB\$C_RBF	= 00000028		

C 11

16-SEP-1984 01:56:34 VAX/VMS Macro V04-00
5-SEP-1984 01:53:31 [MCLDR.SRC]LALOAD.MAR;1

Page 8
(4)

RAB\$L_ROP	= 00000004		
RAB\$V_CCO	= 0000001F		
RAB\$V_PMT	= 0000001E		
RAB\$W_RSZ	= 00000022		
RMSS_EOF	*****	X	04
SS\$NORMAL	*****	X	04
START	= 0000006D	RG	04
SWTBL	= 00000010	R	04
SYSS\$ASSIGN	*****	GX	04
SYSS\$CONNECT	*****	GX	04
SYSS\$GET	*****	GX	04
SYSS\$GETMSG	*****	GX	04
SYSS\$OPEN	*****	GX	04
SYSS\$PUT	*****	GX	04
TRMS	= 0000006A	R	04

LAL
V04

+-----+
! 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
\$AB\$\$	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=LISS: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

LALOAD
LIS

XFLOADER
LIS

LADAMCODE
LIS

LALOAD
LIS

LAMRMCODE
LIS

LADAMCODE
LIS

LADAMCODE
LIS

MARBLI

MARBLI
MAP

MARBLI
LIS