```
BBBBBBBBBBBB        OOOOOOOOO        OOOOOOOOO    TTTTTTTTTTTTTTT     SSSSSSSSSSS
BBBBBBBBBBBB        OOOOOOOOO        OOOOOOOOO    TTTTTTTTTTTTTTT     SSSSSSSSSSS
BBBBBBBBBBBB        OOOOOOOOO        OOOOOOOOO    TTTTTTTTTTTTTTT     SSSSSSSSSSS
BBB         BBB  OOO         OOO  OOO         OOO       TTT         SSS
BBB         BBB  OOO         OOO  OOO         OOO       TTT         SSS
BBB         BBB  OOO         OOO  OOO         OOO       TTT         SSS
BBB         BBB  OOO         OOO  OOO         OOO       TTT         SSS
BBB         BBB  OOO         OOO  OOO         OOO       TTT         SSS
BBB         BBB  OOO         OOO  OOO         OOO       TTT         SSS
BBBBBBBBBBBB     OOO         OOO  OOO         OOO       TTT            SSSSSSSSS
BBBBBBBBBBBB     OOO         OOO  OOO         OOO       TTT            SSSSSSSSS
BBBBBBBBBBBB     OOO         OOO  OOO         OOO       TTT            SSSSSSSSS
BBB         BBB  OOO         OOO  OOO         OOO       TTT                     SSS
BBB         BBB  OOO         OOO  OOO         OOO       TTT                     SSS
BBB         BBB  OOO         OOO  OOO         OOO       TTT                     SSS
BBB         BBB  OOO         OOO  OOO         OOO       TTT                     SSS
BBB         BBB  OOO         OOO  OOO         OOO       TTT                     SSS
BBB         BBB  OOO         OOO  OOO         OOO       TTT                     SSS
BBB         BBB  OOO         OOO  OOO         OOO       TTT                     SSS
BBBBBBBBBBBB        OOOOOOOOO        OOOOOOOOO         TTT         SSSSSSSSSSS
BBBBBBBBBBBB        OOOOOOOOO        OOOOOOOOO         TTT         SSSSSSSSSSS
BBBBBBBBBBBB        OOOOOOOOO        OOOOOOOOO         TTT         SSSSSSSSSSS
```

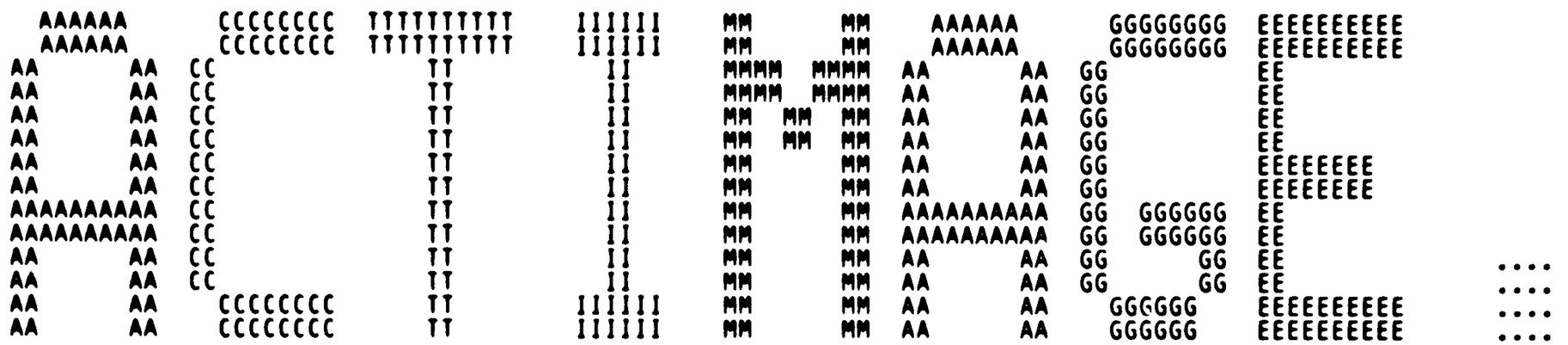
AAAAAA CCCCCCCC TTTTTTTTTT IIIIII MM MM AAAAAA GGGGGGGG EEEEEEEEEE
AAAAAA CCCCCCCC TTTTTTTTTT IIIIII MM MM AAAAAA GGGGGGGG EEEEEEEEEE
AA AA CC TT II MMMM MMMM AA AA GG EE
AA AA CC TT II MMMM MMMM AA AA GG EE
AA AA CC TT II MM MM MM AA AA GG EE
AA AA CC TT II MM MM MM AA AA GG EE
AA AA CC TT II MM MM AA AA GG EEEEEEEE
AA AA CC TT II MM MM AA AA GG EEEEEEEE
AAAAAAAAAA CC TT II MM MM AAAAAAAAAA GG GGGGGG EE
AAAAAAAAAA CC TT II MM MM AAAAAAAAAA GG GGGGGG EE
AA AA CC TT II MM MM AA AA GG GG EE
AA AA CC TT II MM MM AA AA GG GG EE
AA AA CCCCCCCC TT IIIIII MM MM AA AA GGGGGG EEEEEEEEEE
AA AA CCCCCCCC TT IIIIII MM MM AA AA GGGGGG 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     1          .TITLE  ACTIMAGE - ACTIVATE NEXT IMAGE
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:     STANDALONE SYSGEN
0000    31  ;
0000    32  ; ABSTRACT:
0000    33  ;       This module allows an image to activate another image in its
0000    34  ;       place.  The running image is run down and the specified
0000    35  ;       image is activated.
0000    36  ;
0000    37  ; ENVIRONMENT:  User and Exec modes, P0 and P1 space
0000    38  ;
0000    39  ; AUTHOR:  STEVE BECKHARDT,     CREATION DATE:  26-Sep-1979
0000    40  ;
0000    41  ; MODIFIED BY:
0000    42  ;
0000    43  ;       V03-001 MSH0054         Michael S. Harvey       30-May-1984
0000    44  ;               Replace obsolete image activator scratch space symbol.
0000    45  ;
0000    46  ;--
```

```
                        .SBTTL  DECLARATIONS
                ;
                ; INCLUDE FILES:
                ;

                ;
                ; MACROS:
                ;

                ;
                ; EQUATED SYMBOLS:
                ;

                ;
                ; OWN STORAGE:
                ;

                        .PSECT  BOO$ACTIMAGE,RD,NOWRT,EXE
```

```
                        0000     68          .SBTTL  BOO$ACTIMAGE
                        0000     69 ;++
                        0000     70 ; FUNCTIONAL DESCRIPTION:
                        0000     71 ;
                        0000     72 ;       This routine runs down the current image and activates the
                        0000     73 ;       specified image.  It does this by copying code into the
                        0000     74 ;       image header buffer in the control region and jumping to it.
                        0000     75 ;       This code sets up the image name in the image header buffer
                        0000     76 ;       and then jumps into PROCSTRT to activate the image.
                        0000     77 ;
                        0000     78 ; CALLING SEQUENCE:
                        0000     79 ;
                        0000     80 ;       CALLS   BOO$ACTIMAGE
                        0000     81 ;
                        0000     82 ; INPUT PARAMETERS:
                        0000     83 ;
                        0000     84 ;       4(AP)           Address of image name descriptor
                        0000     85 ;
                        0000     86 ; IMPLICIT INPUTS:
                        0000     87 ;
                        0000     88 ;       NONE
                        0000     89 ;
                        0000     90 ; OUTPUT PARAMETERS:
                        0000     91 ;
                        0000     92 ;       NONE
                        0000     93 ;
                        0000     94 ; IMPLICIT OUTPUTS:
                        0000     95 ;
                        0000     96 ;       NONE
                        0000     97 ;
                        0000     98 ; COMPLETION CODES:
                        0000     99 ;
                        0000    100 ;       NONE
                        0000    101 ;
                        0000    102 ; SIDE EFFECTS:
                        0000    103 ;
                        0000    104 ;       NONE
                        0000    105 ;
                        0000    106 ;--
                        0000    107
                        0000    108 BOO$ACTIMAGE::
                0000    0000    109         .WORD   0
                        0002    110
                        0002    111         $CMEXEC_S       B^10$,(AP)      ; Change to exec mode
                  04    000E    112         RET                             ; Should never return here
                        000F    113
                007C    000F    114 10$:    .WORD   ^M<R2,R3,R4,R5,R6>
                        0011    115
          50    01      D0    0011    116         MOVL    #1,R0                   ; Exec mode
5E  00000000'9F40      D0    0014    117         MOVL    @#CTL$AL_STACK[R0],SP   ; Reset exec stack
  56  00000000'9F      9E    001C    118         MOVAB   @#IAC$AL_IMGACTBUF,R6   ; Get address of image activator
                        0023    119                                         ; scratch area
  66  2E'AF   24'       28    0023    120         MOVC3   S^#CDSIZE,B^20$,(R6)    ; Move code
       55  04 AC        D0    0028    121         MOVL    4(AP),R5                ; Get address of image name descriptor
              66         17    002C    122         JMP     (R6)                    ; Jump to moved code
                        002E    123
                        002E    124 20$:    ;
```

```
                                                B 8
                     002E   125          ; This code doesn't get executed here.  Rather, it is moved
                     002E   126          ; into the image activator scratch area in the control region
                     002E   127          ; and executed there.
                     002E   128
  5C   00000000'9F   DE     002E   129          MOVAL   @#MMG$IMGHDRBUF,AP      ; Make AP point to image hdr buffer
          6C   65    D0     0035   130          MOVL    (R5),(AP)               ; Move size of image name
     04 AC   08 AC   DE     0038   131          MOVAL   8(AP),4(AP)             ; Set up pointer to image name
08 AC   04 B5   6C   28     003D   132          MOVC3   (AP),@4(R5),8(AP)       ; Move image name
                03   DD     0043   133          PUSHL   #3                      ; Access mode to run down (user)
  00000000'GF   01   FB     0045   134          CALLS   #1,G^SYS$RUNDWN         ; Run down user mode
       0000000C GF   17     004C   135          JMP     G^EXE$PROCIMGACT        ; Activate image
                     0052   136
                 00000024   0052   137 CDSIZE = .-20$                         ; Size of code to move
                     0052   138
                     0052   139
                     0052   140
                     0052   141          .END
```

```
BOO$ACTIMAGE                    00000000 RG    01
CDSIZE                        = 00000024
CTL$AL_STACK                    ********   X   01
EXE$PROCIMGACT                  ********   X   01
IAC$AL_IMGACTBUF                ********   X   01
MMG$IMGHDRBUF                   ********   X   01
SYS$CMEXEC                      ********  GX   01
SYS$RUNDWN                      ********   X   01
```

Psect synopsis

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
BOO$ACTIMAGE	00000052 (82.)	01 (1.)	NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC BYTE

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	17	00:00:00.10	00:00:01.33
Command processing	96	00:00:00.73	00:00:05.57
Pass 1	72	00:00:00.61	00:00:02.41
Symbol table sort	0	00:00:00.00	00:00:00.00
Pass 2	32	00:00:00.30	00:00:00.67
Symbol table output	2	00:00:00.01	00:00:00.01
Psect synopsis output	1	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	220	00:00:01.78	00:00:10.02

```
The working set limit was 1200 pages.
1796 bytes (4 pages) of virtual memory were used to buffer the intermediate code.
There were 10 pages of symbol table space allocated to hold 8 non-local and 2 local symbols.
141 source lines were read in Pass 1, producing 11 object records in Pass 2.
2 pages of virtual memory were used to define 2 macros.
```

Macro library statistics

Macro library name	Macros defined
_255DUA28:[BOOTS.OBJ]BOOTS.MLB;1	0
_255DUA28:[SYS.OBJ]LIB.MLB;1	0
_255DUA28:[SYSLIB]STARLET.MLB;2	2
TOTALS (all libraries)	2

```
13 GETS were required to define 2 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:ACTIMAGE/OBJ=OBJ$:ACTIMAGE MSRC$:ACTIMAGE/UPDATE=(ENH$:ACTIMAGE)+EXECML$/LIB+LIB$:BOOTS.MLB/LIB
```

BOOTS
MAR

AUTOCONFG
LIS

WRITEBOOT
MAP

BOOCVTATB
LIS

VMB
MAP

SYSGEN
MAP

ACTIMAGE
LIS

BOOT58
LIS

BOOTDEF
SDL