BOOTS

FILEID**CONFIGMN

```
CCCCCCCC     000000    NN      NN  FFFFFFFFFF  IIIIII     GGGGGGGG  MM      MM  NN      NN
CCCCCCCC     000000    NN      NN  FFFFFFFFFF  IIIIII     GGGGGGGG  MM      MM  NN      NN
CC          00    00   NN      NN  FF            II      GG         MMMM  MMMM  NN      NN
CC          00    00   NN      NN  FF            II      GG         MMMM  MMMM  NN      NN
CC          00    00   NNNN    NN  FF            II      GG         MM  MM  MM  NNNN    NN
CC          00    00   NNNN    NN  FF            II      GG         MM      MM  NNNN    NN
CC          00    00   NN  NN  NN  FFFFFFF       II      GG         MM      MM  NN  NN  NN
CC          00    00   NN  NN  NN  FFFFFFF       II      GG         MM      MM  NN  NN  NN
CC          00    00   NN    NNNN  FF            II      GG  GGGGGG MM      MM  NN    NNNN
CC          00    00   NN    NNNN  FF            II      GG  GGGGGG MM      MM  NN    NNNN
CC          00    00   NN      NN  FF            II      GG      GG MM      MM  NN      NN
CC          00    00   NN      NN  FF            II      GG      GG MM      MM  NN      NN
CCCCCCCC     000000    NN      NN  FF          IIIIII      GGGGG    MM      MM  NN      NN    . . .
CCCCCCCC     000000    NN      NN  FF          IIIIII      GGGGG    MM      MM  NN      NN    . . .
```

```
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
```

CONFIGMN
V04-000

M 15
- MAIN ROUTINE FOR CONFIGURE PROCESS   15-SEP-1984 23:46:07 VAX/VMS Macro V04-00   Page 1
                                         4-SEP-1984 23:03:40   [BOOTS.SRC]CONFIGMN.MAR;1   (1)

```
0000      1              .TITLE  CONFIGMN - MAIN ROUTINE FOR CONFIGURE PROCESS
0000      2              .IDENT  'V04-000'
0000      3    ;
0000      4    ;********************************************************************
0000      5    ;*                                                                  *
0000      6    ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                         *
0000      7    ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.          *
0000      8    ;*  ALL RIGHTS RESERVED.                                            *
0000      9    ;*                                                                  *
0000     10    ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  *
0000     11    ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
0000     12    ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER  *
0000     13    ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  *
0000     14    ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY  *
0000     15    ;*  TRANSFERRED.                                                     *
0000     16    ;*                                                                  *
0000     17    ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE  *
0000     18    ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT  *
0000     19    ;*  CORPORATION.                                                     *
0000     20    ;*                                                                  *
0000     21    ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS  *
0000     22    ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.          *
0000     23    ;*                                                                  *
0000     24    ;*                                                                  *
0000     25    ;********************************************************************
0000     26
0000     27    ;++
0000     28    ;
0000     29    ; Facility:  System configuration
0000     30    ;
0000     31    ; Abstract: CONFIGURE is used to dynamically configure VAX MSCP-served and HSC-
0000     32    ;           served disks and tapes.  This is the main routine for this image.
0000     33    ;
0000     34    ; Environment: It is run as a process, in user, exec and kernel modes.
0000     35    ;
0000     36    ; Author:  Maryann Hinden, Creation date:  22-JUL-1983
0000     37    ;
0000     38    ; Modification History:
0000     39    ;
0000     40    ;--
0000     41
0000     42    ;
0000     43    ; Include files:
0000     44    ;
0000     45            $SYSGMSGDEF                                ; Define text for messages
0000     46
0000     47    ;
0000     48    ; Equated Symbols
0000     49    ;
0000     50
0000     51    ;
0000     52    ; Macros
0000     53    ;
0000     54
0000     55    ;
0000     56    ; Own Storage
0000     57    ;
```

```
        00000000      58          .PSECT  NONPAGED_DATA,NOEXE,WRT
                0000  59
00000000        0000  60 BOO$GL_CMDOPT:: .LONG   0                          ; Define this as all bits zero.
                0004  61
```

```
                            0004        63                    .SBTTL  CONFIGMN - main routine for CONFIGURE process
                            0004        64  ;++
                            0004        65  ;
                            0004        66  ;   PURPOSE
                            0004        67  ;       Main routine for CONFIGURE process
                            0004        68  ;
                            0004        69  ;   INPUT
                            0004        70  ;       None
                            0004        71  ;
                            0004        72  ;   OUTPUT
                            0004        73  ;       None
                            0004        74  ;
                            0004        75  ;   FUNCTIONAL DESCRIPTION
                            0004        76  ;       This routine locks down pages that might be referenced at elevated
                            0004        77  ;       IPL and calls BOO$CONFIGURE to start up the threads which actually
                            0004        78  ;       configure the devices.  Unless an error occurs, there is never any
                            0004        79  ;       return - BOO$CONFIGURE hibernates.
                            0004        80  ;--
                        00000000        81                    .PSECT  PAGED_CODE,EXE,WRT
                            0000        82
                  0000      0000        83                    .ENTRY  BOO$CONFIGMN, ^M<>
                            0002        84
                            0002        85  ;
                            0002        86  ;   Lock part of image working set in memory
                            0002        87  ;
                            0002        88                    $LKWSET_S        inadr  = BOO$GQ_LIMITS,-
                            0002        89                                     retadr = BOO$GQ_RETADR
           08 50     E9     0017        90                    BLBC    R0,LKWSETERR
                            001A        91
                            001A        92  ;
                            001A        93  ;   Start up process of configuring devices - asynchronous threads will drive
                            001A        94  ;   the rest
                            001A        95  ;
    00000000'EF    00  FB   001A        96                    CALLS   #0,BOO$CONFIGURE
                            0021        97
                     04     0021        98                    RET
                            0022        99
                            0022       100  ;
                            0022       101  ;   If we couldn't lock the working set for some obscure reason, let
                            0022       102  ;   the world know and exit
                            0022       103  ;
                            0022       104  LKWSETERR:
50     007C8132 8F    D0    0022       105                    MOVL    #SYSG$_CONFIGERR,R0
            FFD4'    30     0029       106                    BSBW    PUTERROR
                     04     002C       107                    RET
                            002D       108
```

C 16

CONFIGMN   - MAIN ROUTINE FOR CONFIGURE PROCESS   15-SEP-1984 23:46:07   VAX/VMS Macro V04-00   Page 4
V04-000   Dummy definitions for CONFIGURE   4-SEP-1984 23:03:40   [BOOTS.SRC]CONFIGMN.MAR;1   (1)

```
                        002D    110                 .SBTTL  Dummy definitions for CONFIGURE
                        002D    111  ;++
                        002D    112  ;
                        002D    113  ;  Dummy definitions to allow CONFIGURE to build.  These entry points are
                        002D    114  ;  expected to never be called in this image.
                        002D    115  ;
                        002D    116  ;--
                        002D    117
             0000       002D    118                 .ENTRY  BOO$INITSWPFIL,^M<>
                        002F    119
        50   D4         002F    120                 CLRL    R0
             04         0031    121                 RET
                        0032    122
             0000       0032    123                 .ENTRY  BOO$INITPAGFIL,^M<>
                        0034    124
        50   D4         0034    125                 CLRL    R0
             04         0036    126                 RET
                        0037    127
                        0037    128                 .END    BOO$CONFIGMN
```

```
BOO$CONFIGMN              00000000 RG     03
BOO$CONFIGURE            ********  X      03
BOO$GL_CMDOPT            00000000 RG      02
BOO$GQ_LIMITS           ********  X       03
BOO$GQ_RETADR           ********  X       03
BJO$INITPAGFIL          00000032 RG       03
BOO$INITSWPFIL          0000002D RG       03
LKWSETERR               00000022 R        03
PUTERROR                ********  X       03
SYS$LKWSET              ********  GX       03
SYSG$_CONFIGERR       = 007C8132
```

```
                       +-------------------+
                       ! 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	00000000 (0.)	01 (1.)	NOPIC	USR	CON	ABS	LCL	NOSHR	EXE	RD	WRT	NOVEC	BYTE
NONPAGED_DATA	00000004 (4.)	02 (2.)	NOPIC	USR	CON	REL	L``	NOSHR	NOEXE	RD	WRT	NOVEC	BYTE
PAGED_CODE	00000037 (55.)	03 (3.)	NOPIC	USR	CON	REL	LCL	NOSHR	EXE	RD	WRT	NOVEC	BYTE

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

Phase	Page faults	CPU Time	Elapsed Time
Initialization	37	00:00:00.08	00:00:00.41
Command processing	131	00:00:00.69	00:00:02.43
Pass 1	143	00:00:01.36	00:00:04.08
Symbol table sort	0	00:00:00.05	00:00:00.05
Pass 2	41	00:00:00.36	00:00:00.60
Symbol table output	3	00:00:00.02	00:00:00.02
Psect synopsis output	2	00:00:00.03	00:00:00.19
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	359	00:00:02.59	00:00:07.78

The working set limit was 1200 pages.
4635 bytes (10 pages) of virtual memory were used to buffer the intermediate code.
There were 10 pages of symbol table space allocated to hold 68 non-local and 0 local symbols.
128 source lines were read in Pass 1, producing 24 object records in Pass 2.
10 pages of virtual memory were used to define 9 macros.

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

Macro library name	Macros defined
_255DUA28:[BOOTS.OBJ]BOOTS.MLB;1	0
_255DUA28:[SYS.OBJ]LIB.MLB;1	1
_255DUA28:[SYSLIB]STARLET.MLB;2	5
TOTALS (all libraries)	6

136 GETS were required to define 6 macros.

There were no errors, warnings or information messages.

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

CONFIG
LIS

BTMEM8SS
LIS

BTMEM790
LIS

CONFIGURE
LIS

BOOTDEF
LIS

BOOTIO
LIS

CONFIGMN
LIS

BOOTDRIVR
LIS

BTMEM730   BTMEM250   BTMEM780
LIS   LIS   LIS

BOOTBLOCK
LIS