F 5

```
BBBBBBBBBBBB        OOOOOOOOO        OOOOOOOOO     TTTTTTTTTTTTTTT     SSSSSSSSSSSS
BBBBBBBBBBBB        OOOOOOOOO        OOOOOOOOO     TTTTTTTTTTTTTTT     SSSSSSSSSSSS
BBBBBBBBBBBB        OOOOOOOOO        OOOOOOOOO     TTTTTTTTTTTTTTT     SSSSSSSSSSSS
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
BBBBBBBBBBBB        OOOOOOOOO        OOOOOOOOO           TTT        SSSSSSSSSSSS
BBBBBBBBBBBB        OOOOOOOOO        OOOOOOOOO           TTT        SSSSSSSSSSSS
BBBBBBBBBBBB        OOOOOOOOO        OOOOOOOOO           TTT        SSSSSSSSSSSS
```

```
**FILE**ID**STANDCONF                                D 7
```

```
  SSSSSSSS  TTTTTTTTTT    AAAAAA    NN      NN  DDDDDDDD      CCCCCCCC    OOOOOO    NN      NN  FFFFFFFFFF
  SSSSSSSS  TTTTTTTTTT    AAAAAA    NN      NN  DDDDDDDD      CCCCCCCC    OOOOOO    NN      NN  FFFFFFFFFF
SS              TT      AA      AA  NN      NN  DD      DD  CC            OO      OO  NN      NN  FF
SS              TT      AA      AA  NN      NN  DD      DD  CC            OO      OO  NN      NN  FF
SS              TT      AA      AA  NNNN    NN  DD      DD  CC            OO      OO  NNNN    NN  FF
SS              TT      AA      AA  NNNN    NN  DD      DD  CC            OO      OO  NNNN    NN  FF
  SSSSSS        TT      AA      AA  NN  NN  NN  DD      DD  CC            OO      OO  NN  NN  NN  FFFFFFFF
  SSSSSS        TT      AA      AA  NN  NN  NN  DD      DD  CC            OO      OO  NN  NN  NN  FFFFFFFF
        SS      TT      AAAAAAAAAA  NN    NNNN  DD      DD  CC            OO      OO  NN    NNNN  FF
        SS      TT      AAAAAAAAAA  NN    NNNN  DD      DD  CC            OO      OO  NN    NNNN  FF
        SS      TT      AA      AA  NN      NN  DD      DD  CC            OO      OO  NN      NN  FF               ....
        SS      TT      AA      AA  NN      NN  DD      DD  CC            OO      OO  NN      NN  FF               ....
SSSSSSSS        TT      AA      AA  NN      NN  DDDDDDDD      CCCCCCCC    OOOOOO    NN      NN  FF               ....
SSSSSSSS        TT      AA      AA  NN      NN  DDDDDDDD      CCCCCCCC    OOOOOO    NN      NN  FF               ....

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  STANDCONF - MAIN PROGRAM FOR STANDALONE CONFIGURE
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:     STANDALONE CONFIGURE
0000    ;
0000    ; ABSTRACT:
0000    ;       THIS ROUTINE IS THE MAIN PROGRAM AND SUBROUTINES FOR STANDALONE
0000    ;       CONFIGURE FOR STANDALONE VMS (I.E. STANDALONE BACKUP AND STANDALONE
0000    ;       BAD).
0000    ;       THIS PROCESS IS USED TO CONFIGURE ALL DISK AND PORT DRIVERS FROM
0000    ;       SYSINIT.  ALL MSCP- AND HSC-SERVED DEVICES ARE CONFIGURED.
0000    ;
0000    ; ENVIRONMENT:  USER, EXEC, AND KERNEL MODE
0000    ;
0000    ; AUTHOR:  MARYANN HINDEN,      CREATION DATE:  18-SEP-1979
0000    ;           (ADAPTED FROM STASYSGEN)
0000    ;
0000    ; MODIFIED BY:
0000    ;
0000    ;       V03-006 CWH3006         CW Hobbs        8-Oct-1983
0000    ;               Properly handle the case of a null device type when displaying
0000    ;               configured devices.  Call them "unknown".
0000    ;
0000    ;       V03-005 CWH3005         CW Hobbs        24-Sep-1983
0000    ;               Add the display of configured devices so that a user has
0000    ;               a better idea of what is happening.
0000    ;
0000    ;       V03-004 CWH3004         CW Hobbs        13-Sep-1983
0000    ;               Borrow STACONFIG.MAR and produce a process suitable for
0000    ;               configuring HSC and MSCP-served devices in the environment
0000    ;               of standalone VMS.
```

```
0000       ;
0000       ;       V03-002 WMC0003         Wayne Cardoza   09-Aug-1983
0000       ;               Put D devices back.
0000       ;
0000       ;       V03-001 WMC0002         Wayne Cardoza   05-Aug-1983
0000       ;               Add lbr$output_help.
0000       ;               Disable autoconfigure of D devices until this module is made
0000       ;               to deal with noncontigous drivers.
0000       ;
0000       ;       V03-001 WMC0001         Wayne Cardoza   01-Aug-1983
0000       ;               Add BOO$EXEOPEN, BOO$FILCLOSE, BOO$UFOOPEN, EXE$LOAD_CODE
0000       ;
0000       ;--
```

```
                                         0000          .SBTTL  DECLARATIONS
                                         0000   ;
                                         0000   ; INCLUDE FILES:
                                         0000   ;
                                         0000          $ddbdef
                                         0000          $dcdef
                                         0000          $ipldef
                                         0000          $sbdef
                                         0000          $ucbdef
                                         0000
                                         0000   ;
                                         0000   ; MACROS:
                                         0000   ;
                                         0000          .macro  table_entry     entry_name
                                         0000                  table_counter = table_counter + 1
                                         0000                  assume  dt$_'entry_name' EQ table_counter
                                         0000          .psect  stacfn_tbldat   exe,rd,nowrt
                                         0000          td_'entry_name':        .ascic   /entry_name/
                                         0000          .psect  stacfn_tblptr   exe,rd,nowrt
                                         0000                                  .address td_'entry_name'
                                         0000          .endm
                                         0000
                                         0000   ;
                                         0000   ; EQUATED SYMBOLS:
                                         0000   ;
                                         0000
                                         0000   ;
                                         0000   ; OWN STORAGE:
                                         0000   ;
                                         0000
                                     00000000          .PSECT  BOO$SYSGEN,WRT
                                         0000
                               00000000  0000   BOO$GL_CMDOPT:: .LONG 0                 ; Options longword
                                         0004
                                     00000000   .psect  stacfn_data_rw  noexe,rd,wrt
                                         0000
                      00000000 00000000  0000   msgdsc:         .long   0,0
                               00000058  0008   msgbuf:         .blkb   80
                               00001058  0058   ucb_vector:     .blkl   1024            ; Remember 1000 ucb's
                                         1058   ucb_end:                                ; Marker for end of vector table
                                         1058
                                     00000000   .psect  stacfn_data_ro  exe,rd,nowrt
                                         0000
                      FFFFFFFF FD050F80  0000   BIN_TIME:       .LONG   -10*1000*1000*5,-1 ; Delta for 5 seconds
43 44 4E 41 54 53 00000010'010E0000'     0008   procnam:.ascid  "STANDCONF"
                            46 4E 4F     0016
67 69 66 6E 6F 43 00000021'010E0000'     0019   ctrstr: .ascid  "Configured  !17<!AC$!AC!UL:!>device type !AC"
41 21 3C 37 31 21 20 20 64 65 72 75      0027
64 3E 21 3A 4C 55 21 43 41 21 24 43      0033
21 20 65 70 79 74 20 65 63 69 76 65      003F
                               43 41     004B
6F 63 65 72 20 74 6F 6E 20 73 69 00'     004D   unknown:.ascic  "is not recognized"
                   64 65 7A 69 6E 67     0059
                                  11     004D
                                         005F
                               00000000  005F   table_counter = 0
                                     00000000   .psect  stacfn_tblptr   exe,rd,nowrt
```

```
                 0000    122 disk_table:
      0000001A'  0000    123         .long   disk_count
                 0004    124         table_entry     RK06
                 0008    125         table_entry     RK07
                 000C    126         table_entry     RP04
                 0010    127         table_entry     RP05
                 0014    128         table_entry     RP06
                 0018    129         table_entry     RM03
                 001C    130         table_entry     RP07
                 0020    131         table_entry     RP07HT
                 0024    132         table_entry     RL01
                 0028    133         table_entry     RL02
                 002C    134         table_entry     RX02
                 0030    135         table_entry     RX04
                 0034    136         table_entry     RM80
                 0038    137         table_entry     TU58
                 003C    138         table_entry     RM05
                 0040    139         table_entry     RX01
                 0044    140         table_entry     ML11
                 0048    141         table_entry     RB02
                 004C    142         table_entry     RB80
                 0050    143         table_entry     RA80
                 0054    144         table_entry     RA81
                 0058    145         table_entry     RA60
                 005C    146         table_entry     RC25
                 0060    147         table_entry     RCF25
                 0064    148         table_entry     RD51
                 0068    149         table_entry     RX50
      0000001A   006C    150 disk_count = table_counter
                 006C    151
      00000000   006C    152 table_counter = 0
           0000006C      153 .psect  stacfn_tblptr   exe,rd,nowrt
                 006C    154 tape_table:
      00000009'  006C    155         .long   tape_count
                 0070    156         table_entry     TE16
                 0074    157         table_entry     TU45
                 0078    158         table_entry     TU77
                 007C    159         table_entry     TS11
                 0080    160         table_entry     TU78
                 0084    161         table_entry     TA78
                 0088    162         table_entry     TU80
                 008C    163         table_entry     TU81
                 0090    164         table_entry     TA81
      00000009   0094    165 tape_count = table_counter
                 0094    166
             00000000    167 .psect  stacfn_code     exe,rd,nowrt
```

```
                                   0000           .SBTTL  STANDCONF - main program
                                   0000   ;++
                                   0000   ; FUNCTIONAL DESCRIPTION:
                                   0000   ;       This is the main program for standalone CONFIGURE.  It does the
                                   0000   ;       following:
                                   0000   ;
                                   0000   ;               1)  Locks the entire image into the working set.
                                   0000   ;               2)  Call BOO$CONFIGURE, which will configure all MSCP-
                                   0000   ;                   and HSC-served devices.
                                   0000   ;
                                   0000   ; CALLING SEQUENCE:
                                   0000   ;
                                   0000   ;       Called by STASYSGEN (as SYSINIT) (via the $CREPRC directive)
                                   0000   ;
                                   0000   ; INPUT PARAMETERS:
                                   0000   ;
                                   0000   ;       NONE
                                   0000   ;
                                   0000   ; OUTPUT PARAMETERS:
                                   0000   ;
                                   0000   ;       NONE
                                   0000   ;
                                   0000   ; ERROR INDICATIONS:
                                   0000   ;
                                   0000   ,       Various errors printed on the system console
                                   0000   ;
                                   0000   ;--
                                   0000
                             000C  0000           .ENTRY  STANDCONF,^M<R2,R3>
                                   0002
                                   0002   ;
                                   0002   ; Load the entire image into the working set.  If running from the console device, w
                                   0002   ; can not do any paging from the image, since the piece of media we loaded from migh
                                   0002   ; not be in the drive.
                                   0002   ;
                                   0002           $LKWSET_S INADR=BOO$GQ_LIMITS, RETADR=BOO$GQ_RETADR
                                   0017
                                   0017   ;
                                   0017   ; Change our process name to "STANDCONF".  SYSINIT (stasysgen) creates us with the
                                   0017   ; process name of "STANDLOAD".  When SYSINIT sees our name change, it knows that we
                                   0017   ; have been loaded and that it is OK to request a new piece of console media.
                                   0017   ;
                                   0017           $SETPRN_S       PRCNAM=PROCNAM
                                   0024
                                   0024   ;
                                   0024   ; Get current values for local copy of SYSPARAM
                                   0024   ;
      00000000'EF    00      FB    0024           CALLS   #0,BOO$USEACT
                                   002B
                                   002B   ;
                                   002B   ; Set a timer for 5 seconds
                                   002B   ;
                                   002B           $SETIMR_S       efn    = #3,-
                                   002B                           daytim = BIN_TIME,-
                                   002B                           astadr = AST_REC
                                   0042
                                   0042   ;
```

```
                                    0042   ; Start threads which will configure MSCP- and HSC-served disks.
                                    0042   ;
          00000000'EF    00    FB   0042          CALLS   #0,BOO$CONFIGURE
                                    0049
                                    0049   ;
                                    0049   ; BOO$CONFIGURE will go into a hibernate state, so we should never get here
                                    0049   ;
                               04   0049          RET
```

```
                              0004  004A  AST_REC:        .WORD ^M<R2>
                                    004C  ;
                                    004C  ; Go to kernel mode and print any devices configured from remote systems
                                    004C  ;
                                    004C          $cmkrnl_s routin=show_devices
                                    005B
                                    005B  ;
                                    005B  ; Reset the timer ast
                                    005B  ;
                                    005B          $SETIMR_S       efn    = #3,-          ; Reset timer
                                    005B                          daytim = BIN_TIME,-
                                    005B                          astadr = AST_REC
                                    006F
                                04  006F          RET                                     ; Dismiss AST
                                    0070
                                    0070  ;
                                    0070  ; Scan the io database and print names of devices served from remote systems
                                    0070  ;
                              0FFC  0070  .entry SHOW_DEVICES,^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
                                    0072
                                    0072  1$:     setipl  #ipl$_scs               ; Block SCS interrupts while scanning
     5B   00000000'GF           7E  0075          movaq   g^scs$gq_config,r11     ; Keep a pointer to the sb queue header
     5A   00000000'GF           9E  007C          movab   g^scs$ga_localsb,r10    ; Keep a pointer to the local sb
                   59   6B      D0  0083          movl    (r11),r9                ; Get a pointer to the first sb in the queue
                   5B   59      D1  0086  10$:    cmpl    r9,r11                  ; Are we back at the sb queue header
                        2E      13  0089          beql    60$                     ; Exit when done with sb's
                   5A   59      D1  008B          cmpl    r9,r10                  ; Is it our system?
                        24      13  008E          beql    50$                     ; Yes, we aren't interested
              58   44   A9      9E  0090          movab   sb$t_nodename(r9),r8    ; Now r8 points to the nodename
              57   54   A9      D0  0094          movl    sb$l_ddb(r9),r7         ; Get the pointer to the first ddb on the sb
                        1A      13  0098  20$:    beql    50$                     ; Done with ddb's, move to next sb
              56   14   A7      9E  009A          movab   ddb$t_name(r7),r6       ; Now r6 points to "DBA" or something simila
              55   04   A7      D0  009E          movl    ddb$l_ucb(r7),r5        ; Make r5 point to the first ucb on the ddb
                        0B      13  00A2  30$:    beql    40$                     ; Done with ucb's, move to next ddb
                        17      10  00A4          bsbb    display_ucb             ; Display one ucb
                   C9   50      E8  00A6          blbs    r0,1$                   ; If we did display, start over
              55   30   A5      D0  00A9          movl    ucb$l_link(r5),r5       ; Move to the next ucb
                        F3      11  00AD          brb     30$
                   57   67      D0  00AF  40$:    movl    ddb$l_link(r7),r7       ; Move to the next ddb
                        E4      11  00B2          brb     20$
                   59   69      D0  00B4  50$:    movl    sb$l_flink(r9),r9       ; Move to the next sb
                        CD      11  00B7          brb     10$
                                    00B9  60$:    setipl  #0                      ; Restore interrupts
                                04  00BC          ret
                                    00BD
                                    00BD  ;
                                    00BD  ; Display one ucb, first we check the ucb vector to see if this ucb address is
                                    00BD  ; in the vector.  If so, then we have already displayed this device.  If not,
                                    00BD  ; then put the ucb into the vector so that we won't display it again.
                                    )0BD  ;
                                    00BD  display_ucb:
     50   00000058'EF           DE  00BD          moval   ucb_vector,r0           ; Point r0 at the vector
     51   00001058'EF           DE  00C4          moval   ucb_end,r1
                   51   50      D1  00CB  1$:     cmpl    r0,r1                   ; At the end?
                        0E      13  00CE          beql    3$                      ; Ignore it
                   55   60      D1  00D0          cmpl    (r0),r5                 ; Already displayed this one?
                        09      13  00D3          beql    3$                      ; Ignore it
```

```
                  60  D5  00D5            tstl    (r0)                        ; Have we found an empty slot?
                  08  13  00D7            beql    5$                          ; Yes, then we must print this one
              50  04  C0  00D9            addl2   #4,r0                       ; Move to the next ucb entry
                  F7  11  00DC            brb     1$
                  50  D4  00DE  3$:       clrl    r0                          ; Let them know we did not print
                      05  00E0            rsb
                          00E1  ;
                          00E1  ; Haven't displayed it yet.
                          00E1  ;
              60  55  D0  00E1  5$:       movl    r5,(r0)                     ; Record this ucb as printed
          54   40 A5  90  00E4            movb    ucb$b_devclass(r5),r4       ; Grab the device class
              54  01  91  00E8            cmpb    #dc$_disk,r4                ; Is it a disk?
                  09  12  00EB            bneq    10$                         ; Not a disk, skip to check next
      50  00000000'EF  DE  00ED            moval   disk_table,r0               ; Make r0 point at the disk table
                  0C  11  00F4            brb     20$
              54  02  91  00F6  10$:      cmpb    #dc$_tape,r4                ; Is it a tape?
                  E3  12  00F9            bneq    3$                          ; We only do tapes and disks
      50  0000006C'EF  DE  00FB            moval   tape_table,r0               ; Make r0 point at the tape table
          54   41 A5  9A  0102  20$:      movzbl  ucb$b_devtype(r5),r4        ; Now get the device class
                  0B  13  0106            beql    30$                         ; Null type?  Call it unknown.
              60  54  D1  0108            cmpl    r4,(r0)                     ; Check against number of entries in table
                  06  14  010B            bgtr    30$                         ; Too big?  Call it unknown.
           53   6044  D0  010D            movl    (r0)[r4],r3                 ; Get the pointer to the name
                  07  11  0111            brb     40$
      53  0000004D'EF  9E  0113  30$:      movab   unknown,r3                  ; Print the unknown device
          54   54 A5  3C  011A  40$:      movzwl  ucb$w_unit(r5),r4           ; Grab the unit number
  00000000'EF   50 8F  9A  011E            movzbl  #80,msgdsc                  ; Set the full length of the buffer
00000004'EF  00000008'EF  9E  0126            movab   msgbuf,msgdsc+4             ; Move address to the descriptor
                          0131            $fao_s  ctrstr=ctrstr,outlen=msgdsc,outbuf=msgdsc,p1=r8,p2=r6,p3=r4,p4=r3
                          0152            setipl  #0                          ; Enable interrupts while printing
                          0155            $brdcst_s msgbuf=msgdsc             ; Tell them about it
              50  01  D0  0168            movl    #1,r0                       ; Let them know we printed
                      05  016B            rsb
```

```
                 016C          .SBTTL  Dummy entry points
                 016C
                 016C  ;
                 016C  ; These entry points are need to resolve references to routines
                 016C  ; not linked with the standalone version of CONFIG.
                 016C  ;
                 016C
                 016C  EXE$LOAD_CODE::
                 016C  LBR$OUTPUT_HELP::
                 016C  LIB$PUT_OUTPUT::
                 016C  LIB$GET_INPUT::
           0000  016C          .WORD   0                       ; ENTRY MASK
      50     D4  016E          CLRL    R0
             04  0170          RET
                 0171
                 0171  BOO$READPROMPT::
                 0171  BOO$READFILE::
                 0171  BOO$FILCLOSE::
                 0171  BOO$UFOOPEN::
                 0171  BOO$EXEOPEN::
                 0171  RIO$OUTPUT_LINE::
                 0171  PUTERROR::
      50     D4  0171          CLRL    R0
             05  0173          RSB
                 0174
                 0174  RIO$GW_OUTLEN::
                 0174  RIO$AB_BUFFER::
       00000175  0174          .BLKB   1
                 0175
                 0175          .END    STANDCONF
```

```
$$T1                      = 00000000                    DT$_TU81                  = 00000008
$$T2                      = 00000007                    EXE$LOAD_CODE               0000016C RG     07
AST_REC                     0000004A R       07         IPL$_SCS                  = 00000008
BIN_TIME                    00000000 R       04         LBR$OUTPUT_HELP             0000016C RG     07
BOO$CONFIGURE               *********     X  07         LIB$GET_INPUT               0000016C RG     07
BOO$EXEOPEN                 00000171 RG      07         LIB$PUT_OUTPUT              0000016C RG     07
BOO$FILCLOSE                00000171 RG      07         MSGBUF                      00000008 R      03
BOO$GL_CMDOP                00000000 RG      02         MSGDSC                      00000000 R      03
BOO$GQ_LIMIT                *********     X  07         PR$_IPL                     *********    X  07
BOO$GQ_RETADR               *********     X  07         PROCNAM                     00000008 R      04
BOO$READFILE                00000171 RG      07         PUTERROR                    00000171 RG     07
BOO$READPROMPT              00000171 RG      07         RIO$AB_BUFFER               00000174 RG     07
BOO$UFOOPEN                 00000171 RG      07         RIO$GW_OUTLEN               00000174 RG     07
BOO$USEACT                  *********     X  07         RIO$OUTPUT_LINE             00000171 RG     07
CTRSTR                      00000019 R       04         SB$L_DDB                  = 00000054
DC$_DISK                  = 00000001                    SB$L_FLINK                = 00000000
DC$_TAPE                  = 00000002                    SB$T_NODENAME             = 00000044
DDB$L_LINK                = 00000000                    SCS$GA_LOCALSB              *********    X  07
DDB$L_UCB                 = 00000004                    SCS$GQ_CONFIG               *********    X  07
DDB$T_NAME                = 00000014                    SHOW_DEVICES                00000070 RG     07
DISK_COUNT                = 0000001A                    STANDCONF                   00000000 RG     07
DISK_TABLE                  00000000 R       05         SYS$BRDCST                  *********   GX  07
DISPLAY_UCB                 000000BD R       07         SYS$CMKRNL                  *********   GX  07
DT$_ML11                  = 00000011                    SYS$FAO                     *********    X  07
DT$_RA60                  = 00000016                    SYS$LKWSET                  *********   GX  07
DT$_RA80                  = 00000014                    SYS$SETIMR                  *********   GX  07
DT$_RA81                  = 00000015                    SYS$SETPRN                  *********   GX  07
DT$_RB02                  = 00000012                    TABLE_COUNTER             = 00000009
DT$_RB80                  = 00000013                    TAPE_COUNT                = 00000009
DT$_RC25                  = 00000017                    TAPE_TABLE                  0000006C R      05
DT$_RCF25                 = 00000018                    TD_ML11                     00000052 R      06
DT$_RD51                  = 00000019                    TD_RA60                     0000006B R      06
DT$_RK06                  = 00000001                    TD_RA80                     00000061 R      06
DT$_RK07                  = 00000002                    TD_RA81                     00000066 R      06
DT$_RL01                  = 00000009                    TD_RB02                     00000057 R      06
DT$_RL02                  = 0000000A                    TD_RB80                     0000005C R      06
DT$_RM03                  = 00000006                    TD_RC25                     00000070 R      06
DT$_RM05                  = 0000000F                    TD_RCF25                    00000075 R      06
DT$_RM80                  = 0000000D                    TD_RD51                     0000007B R      06
DT$_RP04                  = 00000003                    TD_RK06                     00000000 R      06
DT$_RP05                  = 00000004                    TD_RK07                     00000005 R      06
DT$_RP06                  = 00000005                    TD_RL01                     0000002A R      06
DT$_RP07                  = 00000007                    TD_RL02                     0000002F R      06
DT$_RP07HT                = 00000008                    TD_RM03                     00000019 R      06
DT$_RX01                  = 00000010                    TD_RM05                     00000048 R      06
DT$_RX02                  = 0000000B                    TD_RM80                     0000003E R      06
DT$_RX04                  = 0000000C                    TD_RP04                     0000000A R      06
DT$_RX50                  = 0000001A                    TD_RP05                     0000000F R      06
DT$_TA78                  = 00000006                    TD_RP06                     00000014 R      06
DT$_TA81                  = 00000009                    TD_RP07                     0000001E R      06
DT$_TE16                  = 00000001                    TD_RP07HT                   00000023 R      06
DT$_TS11                  = 00000004                    TD_RX01                     0000004D R      06
DT$_TU45                  = 00000002                    TD_RX02                     00000034 R      06
DT$_TU58                  = 0000000E                    TD_RX04                     00000039 R      06
DT$_TU77                  = 00000003                    TD_RX50                     00000080 R      06
DT$_TU78                  = 00000005                    TD_TA78                     0000009E R      06
DT$_TU80                  = 00000007                    TD_TA81                     000000AD R      06
```

```
TD_TE16                          00000085 R      06
TD_TS11                          00000094 R      06
TD_TU45                          0000008A R      06
TD_TU58                          00000043 R      06
TD_TU77                          0000008F R      06
TD_TU78                          00000099 R      06
TD_TU80                          000000A3 R      06
TD_TU81                          000C00A8 R      06
UCB$B_DEVCLASS                 = 00000040
UCB$B_DEVTYPE                  = 00000041
UCB$L_LINK                     = 00000030
UCB$W_UNIT                     = 00000054
UCB_END                          00001058 R      03
UCB_VECTOR                       00000058 R      03
UNKNOWN                          0000004D 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
$ABS$                           00000000  (     0.)  01 (  1.)  NOPIC    USR    CON    ABS    LCL NOSHR   EXE   RD     WRT NOVEC BYTE
BOO$SYSGEN                      00000004  (     4.)  02 (  2.)  NOPIC    USR    CON    REL    LCL NOSHR   EXE   RD     WRT NOVEC BYTE
STACFN_DATA_RW                  00001058  (  4184.)  03 (  3.)  NOPIC    USR    CON    REL    LCL NOSHR NOEXE   RD     WRT NOVEC BYTE
STACFN_DATA_RO                  0000005F  (    95.)  04 (  4.)  NOPIC    USR    CON    REL    LCL NOSHR   EXE   RD   NOWRT NOVEC BYTE
STACFN_TBLPTR                   00000094  (   148.)  05 (  5.)  NOPIC    USR    CON    REL    LCL NOSHR   EXE   RD   NOWRT NOVEC BYTE
STACFN_TBLDAT                   000000B2  (   178.)  06 (  6.)  NOPIC    USR    CON    REL    LCL NOSHR   EXE   RD   NOWRT NOVEC BYTE
STACFN_CODE                     00000175  (   373.)  07 (  7.)  NOPIC    USR    CON    REL    LCL NOSHR   EXE   RD   NOWRT NOVEC BYTE

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

Phase                    Page faults    CPU Time       Elapsed Time
-----                    -----------    --------       ------------
Initialization                   29    00:00:00.09    00:00:00.45
Command processing              108    00:00:00.72    00:00:02.63
Pass 1                          275    00:00:08.01    00:00:18.11
Symbol table sort                 0    00:00:01.02    00:00:01.87
Pass 2                           82    00:00:01.69    00:00:03.63
Symbol table output              16    00:00:00.12    00:00:00.22
Psect synopsis output             3    00:00:00.04    00:00:00.04
Cross-reference output            0    00:00:00.00    00:00:00.00
Assembler run totals            515    00:00:11.70    00:00:26.97

The working set limit was 1350 pages.
42571 bytes (84 pages) of virtual memory were used to buffer the intermediate code.
There were 40 pages of symbol table space allocated to hold 684 non-local and 14 local symbols.
355 source lines were read in Pass 1, producing 32 object records in Pass 2.
26 pages of virtual memory were used to define 24 macros.
```

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

Macro library name                              Macros defined
------------------                              --------------
_$255$DUA28:[BOOTS.OBJ]BOOTS.MLB;1                      0
_$255$DUA28:[SYS.OBJ]LIB.MLB;1                          5
_$255$DUA28:[SYSLIB]STARLET.MLB;2                      15
TOTALS (all libraries)                                 20

786 GETS were required to define 20 macros.

There were no errors, warnings or information messages.

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

0040 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY