F 5

```
BBBBBBBBBBBB       000000000       000000000     TTTTTTTTTTTTTTT     SSSSSSSSSSSS
BBBBBBBBBBBB       000000000       000000000     TTTTTTTTTTTTTTT     SSSSSSSSSSSS
BBBBBBBBBBBB       000000000       000000000     TTTTTTTTTTTTTTT     SSSSSSSSSSSS
BBB         BBB 000         000 000         000        TTT        SSS
BBB         BBB 000         000 000         000        TTT        SSS
BBB         BBB 000         000 000         000        TTT        SSS
BBB         BBB 000         000 000         000        TTT        SSS
BBB         BBB 000         000 000         000        TTT        SSS
BBB         BBB 000         000 000         000        TTT        SSS
BBBBBBBBBBBB    000         000 000         000        TTT           SSSSSSSSS
BBBBBBBBBBBB    000         000 000         000        TTT           SSSSSSSSS
BBBBBBBBBBBB    000         000 000         000        TTT           SSSSSSSSS
BBB         BBB 000         000 000         000        TTT                    SSS
BBB         BBB 000         000 000         000        TTT                    SSS
BBB         BBB 000         000 000         000        TTT                    SSS
BBB         BBB 000         000 000         000        TTT                    SSS
BBB         BBB 000         000 000         000        TTT                    SSS
BBB         BBB 000         000 000         000        TTT                    SSS
BBBBBBBBBBBB       000000000       000000000           TTT        SSSSSSSSSSSS
BBBBBBBBBBBB       000000000       000000000           TTT        SSSSSSSSSSSS
BBBBBBBBBBBB       000000000       000000000           TTT        SSSSSSSSSSSS
```

```
  CCCCCCCC    OOOOOO    NN      NN  FFFFFFFFFF    IIIIII      GGGGGGGG  UU      UU  TTTTTTTTTT  LL
  CCCCCCCC    OOOOOO    NN      NN  FFFFFFFFFF    IIIIII      GGGGGGGG  UU      UU  TTTTTTTTTT  LL
CC          OO      OO  NN      NN  FF              II      GG          UU      UU      TT      LL
CC          OO      OO  NN      NN  FF              II      GG          UU      UU      TT      LL
CC          OO      OO  NNNN    NN  FF              II      GG          UU      UU      TT      LL
CC          OO      OO  NNNN    NN  FF              II      GG          UU      UU      TT      LL
CC          OO      OO  NN  NN  NN  FFFFFFFF        II      GG          UU      UU      TT      LL
CC          OO      OO  NN  NN  NN  FFFFFFFF        II      GG          UU      UU      TT      LL
CC          OO      OO  NN    NNNN  FF              II      GG  GGGGGG  UU      UU      TT      LL
CC          OO      OO  NN    NNNN  FF              II      GG  GGGGGG  UU      UU      TT      LL
CC          OO      OO  NN      NN  FF              II      GG      GG  UU      UU      TT      LL              ....
CC          OO      OO  NN      NN  FF              II      GG      GG  UU      UU      TT      LL              ....
  CCCCCCCC    OOOOOO    NN      NN  FF            IIIIII      GGGGGG    UUUUUUUUUU      TT      LLLLLLLLLL      ....
  CCCCCCCC    OOOOOO    NN      NN  FF            IIIIII      GGGGGG    UUUUUUUUUU      TT      LLLLLLLLLL      ....

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

```
00000001  0000        CONFIGSW=1               ; SET SWITCH TO GENERATE CODE USED BY
          0000                                 ; CONFIGURE PROCESS
          0000        .IF     NDF,CONFIGSW
          0000        .TITLE  SYSGEN - SYSGEN UTILITY AND PARAMETER FILE EDITOR
          0000        .IFF
          0000        .TITLE  CONFIGUTL - SYSGEN UTILITIES FOR CONFIGURE PROCESS
          0000        .ENDC
          0000        .IDENT  'V04-002'
          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:  System generation and initialization
          0000   ;
          0000   ; Abstract: SYSGEN is the main routine to provide all SYSBOOT parameter
          0000   ;       alteration commands in an online environment.
          0000   ;
          0000   ; Environment:
          0000   ;
          0000   ; Author:  RICHARD I. HUSTVEDT, Creation date:  4-MAY-1978
          0000   ;
          0000   ; MODIFIED BY:
          0000   ;
          0000   ;       V04-002 WHM0011         Bill Matthews           14-Sep-1984
          0000   ;               Changed the defaults for the MSCP command.
          0000   ;
          0000   ;       V04-001 WHM0010         Bill Matthews           04-Sep-1984
          0000   ;               Changed IO PRIORITY default for the MSCP command and
          0000   ;               disallow loading of the MSCP server multiple times.
          0000   ;
          0000   ;       V03-023 WHM0009         Bill Matthews           23-Jul-1984
          0000   ;               Changed defaults for the MSCP command.
          0000   ;
          0000   ;       V03-022 WHM0008         Bill Matthews           20-Apr-1984
          0000   ;               Removed WRITE CURRENT code that wrote the SYSGEN parameters
```

```
0000    56 ;            to SYS.EXE.
0000    57 ;
0000    58 ;    V03-021 WHM0007         Bill Matthews           04-Apr-1984
0000    59 ;            Added support to write current to write to a seperate
0000    60 ;            default system parameter file.
0000    61 ;            Added support to use file to accept long ascii sysgen parameters
0000    62 ;
0000    63 ;    V03-020 WHM0006         Bill Matthews           14-Mar-1984
0000    64 ;            Modify SGN$GET_DEVICE to take out the I/O database MUTEX and
0000    65 ;            raise IPL before calling IOC$SEARCHALL.
0000    66 ;
0000    67 ;    V03-019 WHM0005         Bill Matthews           13-Mar-1984
0000    68 ;            Move definition of BOO$GL_LOAD_ARGS from SYSBOOCMD to
0000    69 ;            this module.
0000    70 ;
0000    71 ;    V03-018 ACG0399         Andrew C. Goldstein,    10-Mar-1984  0:36
0000    72 ;            Change check for SS$_NODEVAVL to SS$_NOSUCHDEV due to
0000    73 ;            rewrite of IOC$SEARCHDEV.
0000    74 ;
0000    75 ;    V03-016 WHM0004         Bill Matthews   23-Feb-1984
0000    76 ;            Added support for loading and starting the MSCP server.
0000    77 ;
0000    78 ;    V03-015 WHM0003         Bill Matthews   04-Feb-1984
0000    79 ;            Added support for ACF$B_COMBO_VECTOR_OFFSET to clean up support
0000    80 ;            of combo style devices.
0000    81 ;
0000    82 ;    V03-014 TMK0001         Todd M. Katz    31-Jan-1984
0000    83 ;            Change a BSBW to a JSB.
0000    84 ;
0000    85 ;    V03-013 WHM0002         Bill Matthews           13-Dec-1983
0000    86 ;            Fixed several calls to SGN$GET_DEVICE to pass the unit number
0000    87 ;            to be connected not the maximum units.
0000    88 ;            Added support for the new CONNECT command qualifiers
0000    89 ;            /CSR_OFFSET and /VECTOR_OFFSET.
0000    90 ;
0000    91 ;    V03-012 JLV0312         Jake VanNoy             26-Oct-1983
0000    92 ;            Fix bug for microVAX that allows nexus 0 in CONNECT.
0000    93 ;
0000    94 ;    V03-011 WHM0001         Bill Matthews           09-Dec-1983
0000    95 ;            Changed some bsbw's to jsb's
0000    96 ;
0000    97 ;    V03-010 WMC0003         Wayne Cardoza           09-Aug-1983
0000    98 ;            Fix loadable code error handling.
0000    99 ;            USEACTIVE should be in configutl.
0000   100 ;
0000   101 ;    V03-009 WMC0002         Wayne Cardoza           29-Jul-1983
0000   102 ;            More features for code loading.
0000   103 ;
0000   104 ;    V03-008 WMC0001         Wayne Cardoza           27-Jul-1983
0000   105 ;            Support general code loading.
0000   106 ;
0000   107 ;    V03-007 MSH0006         Maryann Hinden          24-Jun-1983
0000   108 ;            Use $BOOCMDDEF instead of $BOODEF.
0000   109 ;
0000   110 ;    V03-006 MSH0005         Maryann Hinden          04-May-1983
0000   111 ;            Changes to support CONFIGURE process.
0000   112 ;
```

```
                 0000  ;       V03-005 MSH0004         Maryann Hinden          13-May-1983
                 0000  ;               Change some BSBW PUTERROR instructions to JSB instead.
                 0000  ;
                 0000  ;       V03-004 MSH0003         Maryann Hinden          31-Jan-1983
                 0000  ;               Add support for cluster device names.
                 0000  ;
                 0000  ;       V03-003 TCM0001         Trudy C. Matthews       8-Nov-1982
                 0000  ;               Use new ADP$L_AVECTOR field in calculation of ACF$W_AVECTOR,
                 0000  ;               instead of calculating it from the adapter's TR number.
                 0000  ;
                 0000  ;       V03-002 MSH0002         Maryann Hinden          22-Oct-1982
                 0000  ;               Fix broken BSBW.
                 0000  ;
                 0000  ;       V03-001 MSH0001         Maryann Hinden          30-Sep-1982
                 0000  ;               Check for DDB$L_UCB   0.
                 0000  ;--
                 0000
                 0000  ;
                 0000  ; Include files:
                 0000  ;
                 0000          $ACFDEF                         ; Define autoconfiguration block
                 0000          $ADPDEF                         ; Define adapter control block
                 0000          $BOOCMDDEF                      ; Define SYSGEN command options
                 0000          $CLIDEF                         ; Define CLI codes and values
                 0000          $CRBDEF                         ; Define CRB offsets
                 0000          $DDBDEF                         ; Define DDB offsets
                 0000          $DYNDEF                         ; Block types
                 0000          $HLPDEF                         ; Define HELP symbols
                 0000          $IDBDEF                         ; Define IDB offsets
                 0000          $IHDDEF                         ; Image header offsets
                 0000          $IPLDEF                         ; Define IPLs
                 0000          $JPIDEF                         ; $GETJPI definitions
                 0000          $LBRDEF                         ; Librarian symbols
                 0000          $OPCDEF                         ; Operator message definitions
                 0000          $PRDEF                          ; Define processor registers
                 0000          $PRMDEF                         ; Parameter descriptor definitions
                 0000          $SBDEF                          ; SCS system block definitions
                 0000          $SHRDEF                         ; Error codes
                 0000          $SLVDEF                         ; Loadable code header
                 0000          $SSDEF                          ; Define system status values
                 0000          $SYSGMSGDEF                     ; Sysgen messages
                 0 00          $TPADEF                         ; TPARSE definitions
                 0000          $UCBDEF                         ; Define UCB offsets
                 0000          $VECDEF                         ; Define VEC offsets
                 0000
                 0000  ;
                 0000  ; Equated Symbols:
                 0000  ;
       0000000D  0000          CR=13                           ; Character value for carriage return
       0000000C  0000          FF=12                           ; Character value for form feed
       0000000A  0000          LF=10                           ; Character value for line feed
       00001000  0000          UBA_IOBASE=8*512                ; Offset from UBA configuration register
                 0000                                          ; to base of I/O page
                 0000  ;
                 0000  ; Own Storage
                 0000  ;
             00000000          .PSECT  $$$$000,NOEXE,NOWRT     ; PSECT to mark lower address
```

```
                0000  BOO$LOLIM::                                      ; Marker definition
            00000000             .PSECT   ____ZZZ,WRT,PAGE             ; PSECT to mark upper address limit
                0000  BOO$HILIM::                                      ;
            00000000             .PSECT  NONPAGED_DATA   rd,wrt,noexe,quad
                0000
                0000  BOO$AB_PATCH::                                   ; Non-paged Patch area
   00000200     0000             .BLKB   512                           ; One page
                0200  BOO$AB_PRMBUF::                                  ; Parameter buffer
   00002200     0200             .BLKB   512*16                        ; A generous buffer
                2200  BOO$AB_LOADBUF:                                  ; Buffer for code loader
   00002400     2200             .BLKB   512
                2400  ACF$GL_DDB::
   00000000     2400             .LONG   0
                2404  ACF$GL_UCB::
   00000000     2404             .long   0
                2408  ACF$GL_IDB::
   00000000     2408             .long   0
                240C  ACF$GL_CRB::
   00000000     240C             .long   0
                2410  ACF$GL_LASTDDB::
   00000000     2410             .long   0
                2414  ACF$GL_DPT::
   00000000     2414             .long   0
                2418  ACF$GL_SB::
   00000000     2418             .LONG   0
                241C  BOO$GL_COMBO_VECTOR_OFFSET::                     ; Offset to vector from start of combo
   00000000     241C             .LONG   0                             ; device's vectors
                2420  BOO$GL_COMBO_CSR_OFFSET::                        ; Offset to CSR from start of combo
   00000000     2420             .LONG   0                             ; device's CSR
                2424  BOO$GL_CONADP::                                  ; Adapter TR number
   FFFFFFFE     2424             .LONG   -2                            ; Null value
                2428  BOO$GL_CONCREG::                                 ; Control register
   FFFFFFFF     2428             .LONG   -1                            ; Null value
                242C  BOO$GL_CONCUNIT::                                ; Controller unit
   FFFFFFFF     242C             .LONG   -1                            ; Null value
                2430  BOO$GL_CONNUMU::                                 ; Number of Units to configure
   00000001     2430             .LONG   1                             ; Default value is 1 unit
                2434  BOO$GL_CONVECT::                                 ; Vector offset
   FFFFFFFF     2434             .LONG   -1                            ; Null value
                2438  BOO$GL_CONNUMV::                                 ; Number of vectors
   FFFFFFFF     2438             .LONG   -1                            ; Null value
                243C  BOO$GL_CONAUNIT::                                ; Adapter unit
   FFFFFFFF     243C             .LONG   -1                            ; Null value
                2440  BOO$GL_CONDEV::                                  ; Device name string address
   FFFFFFFF     2440             .LONG   -1                            ; Null value
                2444  BOO$GL_CONDRV::                                  ; Driver name string address
   FFFFFFFF     2444             .LONG   -1                            ; Null value
                2448  BOO$GL_CONUNITS::                                ; Maximum units
   00000000     2448             .LONG   0                             ;
                244C  BOO$GQ_CONSYSID::                                ; System ID
   00000000     244C             .LONG   0                             ; quadword
   00000000     2450             .LONG   0                             ;
                2454  BOO$GL_CONCRB::                                  ; CRB address
   00000000     2454             .LONG   0                             ;
                2458  BOO$GL_CONFLAGS::                                ; Flags
   00000000     2458             .LONG   0                             ;
                245C  BOO$GL_NEXTSTR::                                 ; Next string location
```

```
                       00000000  245C          .LONG   0                          ;
                                 2460  BOO$GL_SELECT::                            ; Address of select list
                       00000000  2460          .LONG   0                          ;
                                 2464  BOO$AL_CLIBLK::                            ; CLI call back block
                                 2464          $CLIREQDESC     -                  ; Get command call back block
                                 2464                  RQTYPE=CLI$K_GETCMD        ;
                       0000246C  2480  BOO$GQ_CMDESC==BOO$AL_CLIBLK+CLI$W_RQSIZE       ; Command descriptor address
                                 2480  BOO$GT_PROMPT::                            ; Prompt string
00 20 20 3E 4E 45 47 53 59 53 0A 0D  2480  .ASCIZ  <CR><LF>%SYSGEN>  %            ;
                                 248C  BOO$AL_ACF::                               ; Auto-configuration block
                       000024B4  248C          .BLKB   ACF$C_LENGTH               ; Allocate space for it
                                 24B4  BOO$GQ_LIMITS::                            ; High and low address limits for lockdown
                      00000000'  24B4          .LONG   BOO$LOLIM                  ; Lower address bound
                      FFFFFFFF'  24B8          .LONG   BOO$HILIM-1                ;
                                 24BC  BOO$GQ_RETADR::                            ; Return address receiver
              00000000 00000000  24BC          .LONG   0,0                        ;
                                 24C4  BOO$GL_RETSAVE::                           ; Saved co-routine return address
                       00000000  24C4          .LONG   0                          ;
                                 24C8  FACNAMED::                                 ; Facility name descriptor
            000024D0'00000006'   24C8          .LONG   FACNAMSZ,FACNAME           ;
              4E 45 47 53 59 53  24D0  FACNAME:.ASCII  /SYSGEN/                   ;
                       00000006  24D6  FACNAMSZ=.-FACNAME                         ; Length of facility name
                                 24D6  CONSNAME:                                  ; Console block storage
                    41 53 43 00' 24D6          .ASCIC  /CSA/                      ; device name
                             03  24D6
                                 24DA  BOO$GT_OPNAME::                            ; Console terminal device name
                    41 50 4F 00' 24DA          .ASCIC  /OPA/                      ;
                             03  24DA
                                 24DE  BOO$GT_CVNAME::                            ; Name of RL02 driver
      52 45 56 49 52 44 56 43 00' 24DE         .ASCIC  /CVDRIVER/
                             08  24DE
                                 24E7  BOO$GT_DXNAME::                            ; Name of floppy driver
      52 45 56 49 52 44 58 44 00' 24E7         .ASCIC  /DXDRIVER/
                             08  24E7
                                 24F0  BOO$GT_DDNAME::                            ; Name of TU58 driver
      52 45 56 49 52 44 44 44 00' 24F0         .ASCIC  /DDDRIVER/
                             08  24F0
                                 24F9
                                 24F9  BOO$GL_FILEADDR::                          ; File spec address
                       00000000  24F9          .LONG   0                          ;
                                 24FD  BOO$GB_FILELEN::                           ; File spec length
                             00  24FD          .BYTE   0                          ;
                                 24FE
                       00000000  24FE  BOO$GL_PARINUSE::       .LONG   0
         74 6E 65 72 72 75 43 00' 2502 BOO$GT_CURRENT::        .ASCIC  /Current/
                             07  2502
            65 76 69 74 63 41 00' 250A BOO$GT_ACTIVE::         .ASCIC  /Active/
                             06  250A
         74 6C 75 61 66 65 44 00' 2511 BOO$GT_DEFAULT::        .ASCIC  /Default/
                             07  2511
                       00002559  2519  BOO$GT_FILE::           .BLKB   64
                                 2559
                                 2559  HELP_FILE:                                 ; Help library file name
45 48 24 53 59 53 00002561'010E0000' 2559      .ASCID  /SYS$HELP:SYSGEN.HLB/
4C 48 2E 4E 45 47 53 59 53 3A 50 4C  2567
                             42  2573
                       00000001  2574  HELP_FLAG:      .long           hlp$m_prompt
```

```
             00002580'010E0000'  2578      274 HELP_DESC:        .ascid  //                ; Filled in as pointer
                                 2580      275
                                 2580      276 VALID_PAR_FILE:                             ; Valid parameter file flag
                      00000000   2580      277          .LONG   0
                                 2584      278 SAVE_DOT:                                   ; Save dot through USE filespec
                      00000000   2584      279          .LONG   0
                                 2588      280 FULL_NAME_PTR::                             ; Full device name
                      00000000   2588      281          .LONG   0
                                 258C      282
                                 258C      283 ; MSCP initialization routine default argument list
                                 258C      284
                                 258C      285 MSCP_ARG_LIST:
                      00000008   258C      286          .LONG   8               ; Number of arguments
                      00000001   2590      287          .LONG   1               ; Function code(load and start server)
                      00008000   2594      288          .LONG   32768           ; Default buffer size
                      00000004   2598      289          .LONG   4               ; Default number of receive credits for each host
                      0000000F   259C      290          .LONG   15              ; Default number of hosts supported
                      00000014   25A0      291          .LONG   20              ; Default time out
                      00000004   25A4      292          .LONG   4               ; Default priority
                      00001000   25A8      293          .LONG   4096            ; Default for minimum qualifier
                      00004000   25AC      294          .LONG   16384           ; Default for maximum qualifier
                      000025BC   25B0      295          .BLKL   3               ; Space for new args
                      00000030   25BC      296 MSCP_ARG_LIST_SIZE = .-MSCP_ARG_LIST
                                 25BC      297
                                 25BC      298 BOO$GL_LOAD_ARGS::                          ; Argument list block loadable code init
                      000025EC   25BC      299          .BLKB   MSCP_ARG_LIST_SIZE         ; routine
                                 25EC      300
                                 25EC      301
          50 43 53 4D 00'        25EC      302 MSCP_NAME:        .ASCIC  /MSCP/  ;MSCP server name
                      04         25EC
                                 25F1      303
                                 25F1      304 ; AUTO ALL /LOG storage
                                 25F1      305
55 21 43 41 21 20 000025F9'010E0000'  25F1 306 CTRSTR_AUTOLOG:          .ascid  / !AC!UW/
                      57         25FF
      57 55 21 2C 00002608'010E0000'  2600 307 CTRSTR_AUTOLOG_UNIT:     .ascid  /,!UW/
                      00000000   260C      308 Outlen_unit:             .long   0
                      00000000   2610      309 Outlen:                  .long   0
                      00002628   2614      310 Boo$gt_save_devname:     .blkb   20
             00002630'010E0000'  2628      311 outbuf:                  .ascid  //
                      00002694   2630      312 outbuf_str:              .blkb   100
                                 2694      313
                                 2694      314 ; Send operator message data
                                 2694      315
                                 2694      316 OPERGETJPI:                                       ; $GETJPI item list
                      0004       2694      317          .WORD           4                           ; Buffer length
                      0319       2696      318          .WORD           JPI$_PID                    ; Process ID code
                      000026B0'  2698      319          .ADDRESS        OPERMSGPID                  ; Buffer address
                      00000000   269C      320          .LONG           0                           ; Don't return length
                      00000000   26A0      321          .LONG           0                           ; List terminator
                                 26A4      322
                                 26A4      323 OPERMSGVEC:                                       ; $PUTMSG message vector
                      0003       26A4      324          .WORD   3                                   ; Argument count
                      000F       26A6      325          .WORD   ^B1111                              ; Default message flags
                                 26A8      326 OPERMSGID:
                      00000000   26A8      327          .LONG   0                                   ; Message ID
                                 26AC      328 OPERMSGFAO:
```

```
                0001  26AC          .WORD   1                                       ; FAO argument count
                0000  26AE          .WORD   0                                       ; No new message flags
                      26B0  OPERMSGPID:                                             ; PID of this process
            00000000  26B0          .LONG   0
                      26B4  OPERMSGNAM:                                             ; File specification
           000026B8'  26B4          .ADDRESS        OPERNAMDESC
                      26B8
                      26B8  OPERNAMDESC:
   00000000 00000000  26B8          .LONG   0,0
                      26C0
                      26C0  OPERMSG:                                        ; Message descriptor
            00000000  26C0          .LONG           0
           000026C8'  26C4          .ADDRESS        OPERMSGBUF
                      26C8
                      26C8  OPERMSGBUF:                                     ; Message buffer
            00000103  26C8          .LONG   OPC$_RQ_RQST!<OPC$M_NM_CENTRL@8>        ; Message type and target
            00000000  26CC          .LONG   0                                       ; No reply message
                      26D0  OPERMSGTXT:                                             ; Message text
            000027D0  26D0          .BLKB   256
                      27D0
                      27D0          .IF     NDF,CONFIGSW                    ; SYSGEN-specific code
                      27D0          .PAGE
                      27D0          .SBTTL  BOO$USEFILE - Use parameter file
                      27D0  ;++
                      27D0  ; Functional description:
                      27D0  ;       BOO$USEFILE reads the specified file in response to the USE
                      27D0  ;       command and merges all of the values specified in that file into
                      27D0  ;       the working copy of the parameter values.  This is accomplished
                      27D0  ;       by looking up each value specified and merging the associated
                      27D0  ;       value.
                      27D0  ;
                      27D0  ; Calling sequence:
                      27D0  ;       CALLG   arglist,BOO$USEFILE
                      27D0  ;
                      27D0  ; Input Parameters:
                      27D0  ;       TPA$L_TOKENCNT(AP) - Length of file name string
                      27D0  ;       TPA$L_TOKENPTR(AP) - Address fo file name string
                      27D0  ; Output Parameters:
                      27D0  ;       R0 - Completion status code
                      27D0  ;
                      27D0  ;--
                      27D0
                      27D0  .PSECT  PAGED_CODE      rd,nowrt,exe,long
                      27D0
                      27D0  .Entry  BOO$USEFILE, ^M<R2,R3,R4,R5,R6,R7,R8,R9>        ; Entry mask
                      27D0
                      27D0
                      27D0          BBSS    #EXE$V_WRITESYSPARAMS,- ; Use a file => write current needed
                      27D0                  G^EXE$GL_DYNAMIC_FLAGS,1$;
                      27D0  1$:
                      27D0          MOVL    BOO$GL_DOT,L^SAVE_DOT   ; Save dot
                      27D0          MOVAB   TPA$L_TOKENCNT(AP),R7   ; Set address of file name descriptor
                      27D0          BSBW    BOO$FILOPEN             ; Open specified file
                      27D0          BLBS    R0,20$                  ; Continue if success
                      27D0  10$:    MOVZWL  #1,R0                   ; Force success
                      27D0          RET                             ;
                      27D0  20$:    MOVAB   BOO$AB_PRMBUF,R6        ; Set address of parameter buffer
```

```
27D0              MOVL    #16,R9                    ; Set size of buffer
27D0              BSBW    BOO$READFILE              ; Read file content into parameter buffer
27D0              BLBC    R0,10$                    ; Exit if error
27D0              MOVAB   BOO$AB_PRMBUF,R8          ; Init pointer to parameter buffer
27D0              MOVC3   #32,(R8),EXE$GT_STARTUP ; Set startup command file name
27D0              ADDL    #32,R8                    ;  and advance buffer pointer
27D0              CLRL    VALID_PAR_FILE            ; Initialize valid parameter file flag
27D0      30$:    TSTL    (R8)                      ; Check for end of list
27D0              BEQL    DONE                      ; Branch if yes
27D0              MOVZBL  (R8),TPA$L_TOKENCNT(AP) ; Set token count for search
27D0              MOVAB   1(R8),TPA$L_TOKENPTR(AP); And address of string
27D0              ADDL    #16,R8                    ; Advance to value
27D0              MOVL    (R8)+,TPA$L_NUMBER(AP)  ; Set number
27D0              CALLG   (AP),L^BOO$SEARCH         ; Search for parameter
27D0              BLBC    R0,30$                    ; Next parameter if not found
27D0              MOVL    #1,VALID_PAR_FILE         ; Indicate valid parameter file
27D0              MOVL    TPA$L_PARAM(AP),R4        ; Get a pointer to the parameter descripttor
27D0              BBC     #PRM$V_ASCII,PRM$L_FLAGS(R4),40$; Branch if not an ascii parameter
27D0              MOVAL   -(R8),TPA$L_TOKENPTR(AP); Get a pointer to the parameter value
27D0              MOVZBL  PRM$B_SIZE(R4),R0         ; Get parameter size in bits
27D0              ASHL    #-3,R0,R0                 ; Set parameter size
27D0              MOVZBL  R0,TPA$L_TOKENCNT(AP)     ;
27D0              ADDL2   #3,R0                     ; Round size up to the next longword
27D0              BICL2   #3,R0                     ;
27D0              ADDL2   R0,R8                     ; Advance past value
27D0              CALLG   (AP),W^BOO$SETASCII       ; Set the value of the parameter
27D0              BRW     30$                       ; Continue with the next parameter
27D0      40$:    CALLG   (AP),L^BOO$SETVALUE       ; Set value of parameter
27D0              BRW     30$                       ; Continue with next parameter
27D0      DONE:   BSBW    BOO$FILCLOSE              ; Close the file
27D0              BLBS    VALID_PAR_FILE,10$        ; If LBS, valid parameter file
27D0              MOVL    #SYSG$_NOTPARAM,R0        ; Set error
27D0              BRB     20$                       ; Branch
27D0      10$:
27D0      ;
27D0      ; Set file name in BOO$GL_PARINUSE
27D0      ;
27D0              MOVAL   BOO$GT_FILE,R8            ; Set address of String
27D0              MOVL    R8,BOO$GL_PARINUSE        ; Set address
27D0              MOVZBL  BOO$GB_FILELEN,(R8)       ; Set count
27D0              MOVC3   (R8),@BOO$GL_FILEADDR,-
27D0                      1(R8)                     ; Move string
27D0
27D0              MOVZWL  #SS$_NORMAL,R0            ; Return success
27D0      20$:    MOVL    L^SAVE_DOT,BOO$GL_DOT     ; Restore dot
27D0              RET                               ;
27D0              .ENDC                             ; End of SYSGEN-specific code
```

```
                                    27D0   434         .SBTTL  BOO$USEACT - Use active parameters
                                    27D0   435 ;++
                                    27D0   436 ; Functional description:
                                    27D0   437 ;       This routine copies the parameter values from the running
                                    27D0   438 ;       system to the working copy of the parameter values.
                                    27D0   439 ; Calling sequence:
                                    27D0   440 ;
                                    27D0   441 ;       CALLS   #0,BOO$USEACT
                                    27D0   442 ;
                                    27D0   443 ; Input parameters:
                                    27D0   444 ;       None
                                    27D0   445 ; Output Parameters:
                                    27D0   446 ;       R0 - Completion status code
                                    27D0   447 ;--
                                    27D0   448
                              003C  27D0   449 .Entry  BOO$USEACT,^M<R2,R3,R4,R5>
                                    27D2   450
                  0000'8F     28    27D2   451         MOVC3   #EXE$C_SYSPARSZ,-       ; Move parameters
   00000000'EF    00000000'EF       27D6   452                 MMG$A_SYSPARAM,EXE$A_SYSPARAM
                  FD26 CF     DE    27E0   453         MOVAL   BOO$GT_ACTIVE,-
                  FD17 CF           27E4   454                 BOO$GL_PARINUSE         ; Set parameter in use
                  50   01     D0    27E7   455         MOVL    #1,R0                   ; Return success
                              04    27EA   456         RET
                                    27EB   457         .IF     NDF,CONFIGSW                    ; SYSGEN-specific code
```

```
27EB             .SBTTL  BOO$WRTACT - Write parameters to system
27EB     ;++
27EB     ; Functional Description:
27EB     ;       This routine writes the parameters in the working parameter
27EB     ;       buffer to the system's parameter area.  Only dynamic
27EB     ;       parameters are copied.
27EB     ;
27EB     ; Calling Sequence:
27EB     ;       CALLS   #0,BOO$WRTACT
27EB     ;
27EB     ; Input Parameters:
27EB     ;       None
27EB     ;
27EB     ; Output Parameters:
27EB     ;       R0 - Completion status code
27EB     ;--
27EB
27EB     .PSECT  NONPAGED_CODE   rd,nowrt,exe,long
27EB
27EB     .Entry  BOO$WRTACT, ^M<>
27EB
27EB             $CMKRNL_S       B^10$,(AP)      ; Do it in kernel mode
27EB             BLBC    R0,1$                   ; If LBC, error
27EB             JSB     BOO$SENDOPER            ; Notify operator of WRITE ACTIVE
27EB                     .LONG   SYSG$_WRITEACT
27EB             BLBS    R0,5$                   ; If LBS, success
27EB     1$:     JSB     PUTERROR                ; Report error
27EB             MOVL    #1,R0                   ; Force success
27EB     5$:     RET
27EB
27EB     10$:    .WORD   ^M<R2,R3,R4,R5>
27EB             MOVAB   L^BOO$A_PRMBLK,R5       ; Get base of parameter blocks
27EB             DSBINT  #IPL$_SCHED             ; Raise IPL to prevent being unscheduled
27EB                                             ; (Assumes pages are locked in W.S.)
27EB
27EB             ASSUME  PRM$L_ADDR  EQ  0
27EB
27EB     20$:    MOVL    PRM$L_ADDR(R5),R3       ; Get address of parameter
27EB             BEQL    40$                     ; Reached the end
27EB             BBC     #PRM$V_DYNAMIC,-        ; Branch if this is not a
27EB                     PRM$L_FLAGS(R5),30$     ; dynamic parameter
27EB             MOVZBL  PRM$B_POS(R5),R1        ; Get position of parameter
27EB             EXTZV   R1,PRM$B_SIZE(R5),(R3),R2  ; Extract parameter value
27EB             MOVAB   L^EXE$A_SYSPARAM,R0     ; Get address of working buffer
27EB             SUBL    R0,R3                   ; Get parameter offset
27EB             INSV    R2,R1,PRM$B_SIZE(R5),-  ; Store in system
27EB                     L^MMG$A_SYSPARAM(R3)
27EB
27EB     30$:    ADDL    #PRM$C_LENGTH,R5        ; Point to next paramter block
27EB             BRB     20$                     ; Repeat
27EB
27EB     ; Copy dynamic flags from default flags to R0
27EB
27EB     40$:    BICL3   #^C<PRM$M_DYNFLAGS>,-
27EB                     MMG$A_SYSPARAM+<EXE$GL_DEFFLAGS-EXE$A_SYSPARAM>,R0
27EB             BICL    #PRM$M_DYNFLAGS,-       ; Clear dynamic flags in real flags
27EB                     EXE$GL_FLAGS
```

```
        27EB   516          BISL    R0,EXE$GL_FLAGS         ; Set dynamic flags in real flags
        27EB   517
        27EB   518          ENBINT                          ; Lower IPL
        27EB   519          MOVL    #1,R0                   ; Set success
        27EB   520          RET
```

```
27EB           .SBTTL  BOO$WRTCUR - Write Current Parameters
27EB    ;++
27EB    ; Functional Description:
27EB    ;       This routine writes the parameters from the working parameter
27EB    ;       buffer to the system parameter file on disk.  They will take effect the
27EB    ;       next time the system is booted.
27EB    ;
27EB    ; Calling Sequence:
27EB    ;       CALLS   #0,BOO$WRTCUR
27EB    ;
27EB    ; Input parameters:
27EB    ;       None
27EB    ;
27EB    ; Output Parameters:
27EB    ;       R0 - Completion status code
27EB    ;--
27EB
27EB    .PSECT  PAGED_CODE      rd,nowrt,exe,long
27EB
27EB    .Entry  BOO$WRTCUR, ^M<R2,R3,R4,R5,R6,R7,R8,R9>
27EB
27EB            BBCC    #EXE$V_WRITESYSPARAMS,- ; Don't do WRITE CURRENT again in startup
27EB                    G^EXE$GL_DYNAMIC_FLAGS,10$;
27EB    10$:    MOVAB   BOO$GT_SYSPARNAME,R0    ; Get address of system .PAR file name
27EB            MOVZBL  (R0)+,TPA$L_TOKENCNT(AP); Set up for call to BOO$WRTSYSPARFILE
27EB            MOVL    R0,TPA$L_TOKENPTR(AP)   ;
27EB            CALLG   (AP),G^BOO$WRTSYSPARFILE; Call the routine to write out the file
27EB            BLBC    R0,20$                  ; Branch if error
27EB            BSBW    BOO$SENDOPER            ; Notify operator of WRITE CURRENT
27EB                    .LONG   SYSG$_WRITECUR
27EB            BLBS    R0,30$                  ; If LBS, success
27EB    20$:    BSBW    PUTERROR                ; Report error
27EB    30$:    MOVL    #1,R0                   ; Return success
27EB            RET
27EB
```

```
27EB             .SBTTL  BOO$SENDOPER - Output facility error message to operator
27EB     ;
27EB     ; Functional Description:
27EB     ;       BOO$SENDOPER outputs an error message to the operator.
27EB     ;
27EB     ; Calling Sequence:
27EB     ;       BSBW    BOO$SENDOPER
27EB     ;               .LONG   <msg-id>
27EB     ;
27EB     BOO$SENDOPER::
27EB             MOVL    @(SP),OPERMSGID                 ; Put message ID in vector
27EB             ADDL2   #4,(SP)                         ; Advance return address
27EB             $GETJPI_S       ITMLST=OPERGETJPI       ; Get process ID
27EB             BLBC    R0,10$                          ; If LBC, error
27EB             MOVL    #3,OPERMSGVEC                   ; Assume WRITE ACTIVE
27EB             MOVL    #1,OPERMSGFAO
27EB             CLRL    OPERMSGNAM
27EB             CMPL    #SYSG$_WRITECUR,OPERMSGID       ; WRITE CURRENT ?
27EB             BNEQ    5$                              ; If NEQ, no
27EB             INCL    OPERMSGVEC                      ; Set up WRITE CURRENT
27EB             INCL    OPERMSGFAO
27EB             MOVAB   OPERNAMDESC,OPERMSGNAM
27EB             MOVZBL  RIO_INPNAM+NAM$B_RSL,OPERNAMDESC; Build descriptor
27EB             MOVL    RIO_INPNAM+NAM$L_RSA,OPERNAMDESC+4
27EB     5$:     $PUTMSG_S -                             ; Get and format message
27EB                             MSGVEC=OPERMSGVEC, -
27EB                             ACTRTN=666$
27EB             BLBC    R0,10$                          ; If LBC, error
27EB             $SNDOPR_S       MSGBUF=OPERMSG
27EB             BLBS    R0,20$                          ; If LBS, success
27EB     10$:    BSBW    PUTERROR                        ; Report error
27EB             MOVL    #1,R0                           ; Force success
27EB     20$:
27EB             RSB
27EB     666$:
27EB             .WORD   ^M<R2,R3,R4,R5>
27EB             MOVQ    @4(AP),R0               ; Get string descriptor
27EB             ADDL3   #OPC$L_MS_TEXT,R0,OPERMSG; Store total operator message size
27EB             MOVC3   R0,(R1),OPERMSGTXT      ; Copy text to operator message buffer
27EB             CLRL    R0                      ; Prevent message output to SYS$OUTPUT
27EB             RET
27EB
27EB             .ENDC                           ; End of SYSGEN-specific code
```

```
                              27EB         .SBTTL  BOO$CONFIGALL - Auto-configure all adapters
                              27EB    ;++
                              27EB    ; Functional Description:
                              27EB    ;        BOO$CONFIGALL is called to implement the "AUTOCONFIGURE ALL"
                              27EB    ;        command. All standard devices supported by VAX/VMS will be
                              27EB    ;        located and connected for use with any necessary drivers being
                              27EB    ;        loaded.
                              27EB    ;
                              27EB    ; Calling Sequence:
                              27EB    ;        CALLG   ARGLIST,BOO$CONFIGALL
                              27EB    ;
                              27EB    ; Output parameters:
                              27EB    ;        R0 - Completion status code
                              27EB    ;--
                              27EB
                          00000000    .PSECT  NONPAGED_CODE   rd,nowrt,exe,long
                              0000
                     0FFC     0000    .Entry  BOO$CONFIGALL, ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>      ; Entry mask
                              0002
 08 00000000'EF   00000000'8F   E1  0002         BBC     #EXE$V_NOAUTOCNF,EXE$GL_DEFFLAGS,5$; do we allow auto configure
          50      007C8002 8F   D0  000E         MOVL    #SYSG$_NOAUTOCNF,R0     ;Give them a no autoconfigure error
                                04  0015         RET                             ; and return
                                    0016
                     FFE7'      30  0016  5$:    BSBW    BOO$LOCK_GEN            ; Lock SYSGEN database
                     07 50      E8  0019         BLBS    R0,7$                   ; If no error, continue
              00000000'EF       16  001C         JSB     PUTERROR
                                04  0022         RET
                                    0023
                     FFDA'      30  0023  7$:    BSBW    IOC$AUTORESET           ; Reset controller characters for device
                                    0026                                         ; names
                        5B      D4  0026         CLRL    R11                     ; Indicate no ADP address yet
                        5B      DD  0028  10$:   PUSHL   R11                     ; Set as argument
        000000B2'EF     01      FB  002A         CALLS   #1,NEXTADP              ; Get next ADP address
                     29 50      E9  0031         BLBC    R0,CONFIG_EXIT          ; Branch if error (NOPRIV)
                  5B    51      D0  0034         MOVL    R1,R11                  ; Check return status
                        10      18  0037         BGEQ    20$                     ; Branch if done
                        5B      DD  0039         PUSHL   R11                     ; Set as ADP argument
           0103'CF      01      FB  003B         CALLS   #1,W^CONFIGADP          ; Configure the entire adapter
                     E5 50      E8  0040         BLBS    R0,10$                  ; Continue if no error
              00000000'EF       16  0043         JSB     PUTERROR                ; Report error
                  50    01      D0  0049  20$:   MOVL    #1,R0                   ; Set success
                                    004C
      09 00000000'EF    0C      E1  004C         BBC     #BOOCMD$V_AUTOLOG,L^BOO$GL_CMDOPT,CONFIG_EXIT ; Branch if not /LOG
              00002614'EF       D4  0054         CLRL    BOO$GT_SAVE_DEVNAME     ; Clear name
                     01A9       30  005A         BSBW    AUTOLOG                 ; Output last line if there is one
                                    005D
                                    005D  CONFIG_EXIT:
                        50      DD  005D         PUSHL   R0                      ; Save status
                     FF9E'      30  005F         BSBW    BOO$UNLOCK_GEN          ; Unlock SYSGEN database
                     06 50      E8  0062         BLBS    R0,35$                  ; If no error, continue
              00000000'EF       16  0065         JSB     PUTERROR                ; Give error message
                     50 8ED0        006B  35$:   POPL    R0                      ; Restore status
                                04  006E         RET                             ;
                                    006F
                     0FFC     006F    .Entry  BOO$CONFIGONE, ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>      ; Entry mask
                              0071
                     FF8C'      30  0071         BSBW    BOO$LOCK_GEN            ; Lock SYSGEN database
              07 50   E8 0074              BLBS    R0,5$                      ; If no error, continue
        00000000'EF   16 0077              JSB     PUTERROR                   ; Give error message
                 04      007D              RET                                ;
                         007E
             FF7F'    30 007E      5$:     BSBW    IOC$AUTORESET              ; Reset controller characters for device
                         0081                                                 ; device names
              1C AC   DD 0081              PUSHL   TPA$L_NUMBER(AP)           ; Set TR number of adapter
          DA'AF  01   FB 0084              CALLS   #1,B^LOCADP                ; Locate adapter control block
              D2 50   E9 0088              BLBC    R0,CONFIG_EXIT             ; Branch if error (NOPRIV)
                 51   DD 008B              PUSHL   R1                         ; Set as argument to CONFIGADP
                 0D   13 008D              BEQL    10$                        ; Done if no adapter
          03'AF  01   FB 008F              CALLS   #1,B^CONFIGADP             ; Configure adapter
              06 50   E8 0093              BLBS    R0,10$                     ; Continue if no error
        00000000'EF   16 0096              JSB     PUTERROR                   ; Give error status
              50 01   D0 009C      10$:    MOVL    #1,R0                      ; Set success for parse
  B6 00000000'EF 0C   E1 009F              BBC     #BOOCMD$V_AUTOLOG,L^BOO$GL_CMDOPT,CONFIG_EXIT ; Branch if not /LOG
        00002614'EF   D4 00A7              CLRL    BOO$GT_SAVE_DEVNAME        ; Clear name
               0156   30 00AD              BSBW    AUTOLOG                    ; Output last line if there is one
                 AB   11 00B0      20$:    BRB     CONFIG_EXIT                ;
                         00B2
                         00B2      NEXTADP:                                   ; Return next ADP address in R0
                  0000   00B2              .WORD   0                          ; Null entry mask
                         00B4              $CMEXEC_S       B^10$,(AP)         ; Call real routine in exec mode
                    04   00C0              RET                                ;
                         00C1
                  0000   00C1      10$:    .WORD   0                          ; Null entry mask
           51  04 AC  D0 00C3              MOVL    4(AP),R1                   ; Get current address
                  06  13 00C7              BEQL    20$                        ; 0 => start of list
           51  04 A1  D0 00C9              MOVL    ADP$L_LINK(R1),R1          ; Flink onward
                  07  11 00CD              BRB     30$
      51   00000000'EF D0 00CF      20$:   MOVL    IOC$GL_ADPLIST,R1          ; Return head of list
              50   01 D0 00D6      30$:    MOVL    #1,R0
                   04    00D9              RET                                ;
                         00DA
                         00DA      LOCADP:                                    ; Return address of ADP for TR number
                  0000   00DA              .WORD   0                          ;
                         00DC              $CMEXEC_S       B^5$,(AP)          ; Call routine in exec mode
                    04   00E8              RET                                ;
                         00E9
                  0000   00E9      5$:     .WORD   0                          ; Null entry mask
      51   FFFFFFFC'EF 9E 00EB             MOVAB   IOC$GL_ADPLIST-ADP$L_LINK,R1 ; Set starting address
           51  04 A1  D0 00F2      10$:    MOVL    ADP$L_LINK(R1),R1          ; Flink onward
                  07  13 00F6              BEQL    20$                        ; Done if at end
        0C A1  04 AC  B1 00F8              CMPW    4(AP),ADP$W_TR(R1)         ; Is this the specified TR?
                  F3  12 00FD              BNEQ    10$                        ; No, try another
              50   01 D0 00FF      20$:    MOVL    #1,R0
                   04    0102              RET                                ;
                         0103
                  00FC   0103      .Entry  CONFIGADP, ^M<R2,R3,R4,R5,R6,R7>; Entry mask
                         0105
        000024C4'EF   D4 0105              CLRL    BOO$GL_RETSAVE             ; Zap return address for initial call
  10 00000000'EF 06   E1 010B              BBC     #BOOCMD$V_SELECT,L^BOO$GL_CMDOPT,10$ ; Mutually exclusive - test
  08 00000000'EF 07   E1 0113              BBC     #BOOCMD$V_EXCLUDE,L^BOO$GL_CMDOPT,10$ ; to make sure one bit clear
  50     007C808A 8F  D0 011B              MOVL    #SYSG$_CONFQUAL,R0         ; Conflicting qualifiers
                  04     0122              RET
                         0123
      0171'CF    6C   FA 0123      10$:    CALLG   (AP),W^50$                 ; Call configure one device
```

```
                09 50   E8  0128              BLBS    R0,20$                  ; Branch if not done with this adapter
             50    24   B1  012B              CMPW    #SS$_NOPRIV,R0          ; Was there a privilege error
                   03   13  012E              BEQL    15$                     ; Yes, branch
             50    01   D0  0130              MOVL    #1,R0                   ; Set success
                        04  0133        15$:  RET                             ; and return
                            0134
   55    0000248C'EF    9E  0134        20$:  MOVAB   BOO$AL_ACF,R5           ; Set address of arguments describing device
                            013B
                1C A5   B4  013B              CLRW    ACF$W_MAXUNITS(R5)      ; Always use driver specified max units
   56    00002460'EF    D0  013E              MOVL    L^BOO$GL_SELECT,R6      ; Get pointer to select list
                   06   13  0145              BEQL    35$                     ; Branch if null
                 0087   30  0147              BSBW    SELECT                  ; Check select/exclude string
                D6 50   E9  014A              BLBC    R0,10$                  ; Branch if device is not to be configured
                            014D
          11 0B A5 03   E0  014D        35$:  BBS     #ACF$V_NOLOAD_DB,ACF$B_AFLAG(R5),38$ ; Branch if not loading databas
    09 00000000'EF 0C   E1  0152              BBC     #BOOCMD$V_AUTOLOG,L^BOO$GL_CMDOPT,38$ ; Branch if not logging
                 00A9   30  015A              BSBW    AUTOLOG                 ; Branch to output log
                03 50   E8  015D              BLBS    R0,38$                  ; Branch if no error
                FE9D'   30  0160              BSBW    PUTERROR                ; Give error message
                            0163
         0000'CF   65   FA  0163        38$:  CALLG   (R5),W^IOGEN$LOADER     ; Load database and driver if necessary
                B8 50   E8  0168              BLBS    R0,10$                  ; Branch if no error
                FE92'   30  016B              BSBW    PUTERROR                ; Give error message
                 FFB2   31  016E              BRW     10$                     ; continue loop
                            0171
                      0000  0171        50$:  .WORD   0                       ;
                            0173              $CMKRNL_S       B^55$,(AP)      ; Call auto configure in kernel mode
                        04  017F              RET                             ;
                            0180
                      0FFC  0180        55$:  .WORD   ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>     ;
   50    000024C4'EF    D0  0182              MOVL    BOO$GL_RETSAVE,R0       ; Get saved return address
                   07   12  0189              BNEQ    60$                     ; Branch if one present
   50    00000000'EF    9E  018B              MOVAB   IOC$AUTOCONFIG,R0       ; Else use main entry point
                   50   DD  0192        60$:  PUSHL   R0                      ; Stack call back address
             58 04 AC   D0  0194              MOVL    4(AP),R8                ; Get address of ADP
                56 68   D0  0198              MOVL    ADP$L_CSR(R8),R6        ; Get Configuration register address
   57    0000248C'EF    9E  019B              MOVAB   BOO$AL_ACF,R7           ; Address of configuration control block
                            01A2              SETIPL  #31                     ; Disable interrupts
                9E 16       01A5              JSB     @(SP)+                  ; Call Auto configurtation code
                            01A7              SETIPL  #0                      ; Enable interrupts
      000024C4'EF  8E   D0  01AA              MOVL    (SP)+,BOO$GL_RETSAVE    ; Save return
                06 50   E8  01B1              BLBS    R0,70$                  ; Continue if another device
         000024C4'EF    D4  01B4              CLRL    BOO$GL_RETSAVE          ; Else clear return
                            01BA
          0D 0B A7 03   E1  01BA        70$:  BBC     #ACF$V_NOLOAD_DB,ACF$B_AFLAG(R7),80$ ; Branch if loading database
             7E 12 A7   3C  01BF              MOVZWL  ACF$W_CUNIT(R7),-(SP)   ; Get unit number
                14 A7   DD  01C3              PUSHL   ACF$L_DEVNAME(R7)       ; Get device name
                 010B   30  01C6              BSBW    SGN$GET_DEVICE_LOCK_IODB; Get device database
             5E    08   C0  01C9              ADDL2   #8,SP                   ; Clear stack
                        04  01CC        80$:  RET                             ; And return
             50    01   3C  01CD        90$:  MOVZWL  #1,R0                   ; Set success status
                        04  01D0              RET                             ; and return
                            01D1        ;
                            01D1        ; SELECT - decide whether current device name is one of those either
                            01D1        ;          specified in /SELECT or /EXCLUDE
                            01D1        ;
                            01D1        ; Returns: R0 = 1   ==> configure device
```

```
                              01D1   773 ;          R0 = 0   ==> don't configure device
                              01D1   774 ;
                              01D1   775
                              01D1   776 SELECT:
           57    14 A5    D0  01D1   777 10$:     MOVL    ACF$L_DEVNAME(R5),R7     ; Get pointer to device name
                 54    86  9A  01D5   778          MOVZBL  (R6)+,R4                 ; Get length of select entry
                       10  13  01D8   779          BEQL    20$                      ; End of list, no match
                 54    87  91  01DA   780          CMPB    (R7)+,R4                 ; Compare with device entry
                       06  19  01DD   781          BLSS    15$                      ; Branch if select longer than device
           67    66    54  29  01DF   782          CMPC3   R4,(R6),(R7)             ; Do we have a match?
                       13  13  01E3   783          BEQL    40$                      ; Yes, check SELECT or EXCLUDE
                 56    54  C0  01E5   784 15$:     ADDL    R4,R6                    ; Advance to next entry in select list
                       E7  11  01E8   785          BRB     10$                      ; And try again
                               01EA   786
                       50  D4  01EA   787 20$:     CLRL    R0                       ; Assume don't configure
       03 00000000'EF  07  E1  01EC   788          BBC     #BOOCMD$V_EXCLUDE,BOO$GL_CMDOPT,30$ ; Branch if SELECT
                 50    01  D0  01F4   789          MOVL    #1,R0                    ; EXCLUDE - configure device
                       05      01F7   790 30$:     RSB
                               01F8   791
                       50  D4  01F8   792 40$:     CLRL    R0                       ; Assume don't configure
       03 00000000'EF  07  E0  01FA   793          BBS     #BOOCMD$V_EXCLUDE,BOO$GL_CMDOPT,50$ ; Branch if EXCLUDE
                 50    01  D0  0202   794          MOVL    #1,R0                    ; SELECT - configure device
                       05      0205   795 50$:     RSB
```

```
                                 0206    797 .SBTTL  AUTOLOG - AUTO ALL /LOG formating
                                 0206    798
                                 0206    799 AUTOLOG::
      55  0000248C'EF      9E    0206    800          MOVAB   BOO$AL_ACF,R5            ; Address of configuration control block
            56   14 A5     D0    020D    801          MOVL    ACF$L_DEVNAME(R5),R6     ; Get address of current device
               57   86     9A    0211    802          MOVZBL  (R6)+,R7                 ; Get count and addr.
  00002614'EF   66   57    29    0214    803          CMPC3   R7,(R6),BOO$GT_SAVE_DEVNAME ; Compare to previous string
                    39     12    021C    804          BNEQ    50$                      ; Branch if new device
                                 021E    805
                                 021E    806          $FAO_S  CTRSTR=CTRSTR_AUTOLOG_UNIT ,- ; Format Unit Number
                                 021E    807                  OUTBUF=OUTBUF,-
                                 021E    808                  OUTLEN=OUTLEN_UNIT ,-
                                 021E    809                  P1=ACF$W_CUNIT(R5)
                 03 50     E8    023A    810          BLBS    R0,40$                   ; Branch if OK
                  0081     31    023D    811          BRW     100$                     ; Branch if error
                                 0240    812
      2610'CF   260C'CF    C0    0240    813 40$:     ADDL2   W^OUTLEN_UNIT,W^OUTLEN   ; Add to total length
      262C'CF   260C'CF    C0    0247    814          ADDL2   W^OUTLEN_UNIT,W^OUTBUF+4; Add to descriptor
      2628'CF   260C'CF    A2    024E    815          SUBW2   W^OUTLEN_UNIT,W^OUTBUF   ; Subtract from length
                    6A     11    0255    816          BRB     100$                     ; Return with success
                                 0257    817
                2610'CF    D5    0257    818 50$:     TSTL    W^OUTLEN                 ; Is this a first call to this routine?
                    21     13    025B    819          BEQL    70$                      ; Branch if yes
                                 025D    820
      262C'CF   2630'CF    DE    025D    821          MOVAL   W^OUTBUF_STR,W^OUTBUF+4 ; reset descriptor
      0000'CF   2610'CF    B0    0264    822          MOVW    W^OUTLEN,W^RIO$GW_OUTLEN ; Length of string
                0000'CF    28    026B    823          MOVC3   W^RIO$GW_OUTLEN,-
                2630'CF          026F    824                  W^OUTBUF_STR,-
                0000'CF          0272    825                  W^RIO$AB_BUFFER          ; Move text into global buffer
                                 0275    826
            00000000'EF    16    0275    827          JSB     RIO$OUTPUT_LINE
                 43 50     E9    027B    828          BLBC    R0,100$                  ; Branch on error
                                 027E    829
      2628'CF   0064 8F    B0    027E    830 70$:     MOVW    #100,W^OUTBUF            ; Set full buffer length
  00002614'EF   66   57    28    0285    831          MOVC3   R7,(R6),BOO$GT_SAVE_DEVNAME ; Save new devname
      55  0000248C'EF      9E    028D    832          MOVAB   BOO$AL_ACF,R5            ; Reset R5
                                 0294    833          $FAO_S  CTRSTR=CTRSTR_AUTOLOG,- ; Format device name
                                 0294    834                  OUTBUF=OUTBUF,-
                                 0294    835                  OUTLEN=OUTLEN,-
                                 0294    836                  P1=ACF$L_DEVNAME(R5),-
                                 0294    837                  P2=ACF$W_CUNIT(R5)
      262C'CF   2610'CF    C0    02B3    838          ADDL2   W^OUTLEN,W^OUTBUF+4      ; Add to descriptor
      2628'CF   2610'CF    A2    02BA    839          SUBW2   W^OUTLEN,W^OUTBUF        ; Subtract from length
                                 02C1    840                                           ; Return with FAO status
                           05    02C1    841 100$:    RSB
                                 02C2    842
                           02C2    .SBTTL  SGN$GET_DEVICE - Locate device database
                           02C2
                           02C2    ;
                           02C2    ; Inputs:
                           02C2    ;       4(SP) - Address of Device name in ascic format
                           02C2    ;       8(SP) - Unit number
                           02C2    ;
                           02C2    ; Outputs:
                           02C2    ;       (Any of these are 0 if the data block doesn't exist)
                           02C2    ;       ACF$GL_DDB - Address of DDB
                           02C2    ;       ACF$GL_UCB - Address of UCB
                           02C2    ;       ACF$GL_IDB - Address of IDB
                           02C2    ;       ACF$GL_CRB - Address of CRB
                           02C2    ;       ACF$GL_SB  - Address of SB
                           02C2    ;       ACF$GL_LASTDDB - If ACF$GL_DDB is non-zero, then equal to that,
                           02C2    ;                    otherwise, last DDB in DEVLIST
                           02C2    ;       R0 = 0 - error
                           02C2    ;          = 1 - success
                           02C2    ;
                           02C2    ; Must be called at IPL=0 and KERNEL mode
                           02C2    ;
                           02C2            .ENABL  LSB
                           02C2
                           02C2    SGN$GET_DEVICE::                            ; Entry with IODB MUTEX & raised IPL
                           02C2
         007C 8F      BB   02C2            PUSHR   #^M<R2,R3,R4,R5,R6>         ; ADDS 20 to offset to input
                           02C6
54    00000000'GF     D0   02C6            MOVL    G^CTL$GL_PCB,R4             ; PICK UP PCB POINTER
               32     10   02CD            BSBB    10$                         ; Call real routine
                           02CF
         007C 8F      BA   02CF            POPR    #^M<R2,R3,R4,R5,R6>         ; restore regs
                      05   02D3            RSB                                 ; Return
                           02D4
                           02D4    SGN$GET_DEVICE_LOCK_IODB:                   ; Entry without IODB MUTEX and IPL 0
                           02D4
         007C 8F      BB   02D4            PUSHR   #^M<R2,R3,R4,R5,R6>         ; ADDS 20 to offset to input
                           02D8
54    00000000'GF     D0   02D8            MOVL    G^CTL$GL_PCB,R4             ;PICK UP PCB POINTER
      00000000'GF     16   02DF            JSB     G^SCH$IOLOCKR               ;GET THE IODB MUTEX FOR READ & RAISE IPL
               1A     10   02E5            BSBB    10$                         ;
               50     DD   02E7            PUSHL   R0                          ;SAVE RETURN STATUS
54    00000000'GF     D0   02E9            MOVL    G^CTL$GL_PCB,R4             ;PICK UP PCB POINTER
      00000000'GF     16   02F0            JSB     G^SCH$IOUNLOCK              ;RELEASE THE IODB MUTEX
                           02F6            SETIPL  #0                          ;LOWER IPL
                           02F9
            50 8ED0        02F9            POPL    R0                          ;RESTORE RETURN STATUS FROM LOCAL ROUTINE
         007C 8F      BA   02FC            POPR    #^M<R2,R3,R4,R5,R6>
                      05   0300            RSB
                           0301
         2400'CF      D4   0301    10$:    CLRL    W^ACF$GL_DDB                ;INIT TO ZERO
         2404'CF      D4   0305            CLRL    W^ACF$GL_UCB                ;INIT TO ZERO
         2408'CF      D4   0309            CLRL    W^ACF$GL_IDB                ;INIT TO ZERO
         240C'CF      D4   030D            CLRL    W^ACF$GL_CRB                ;INIT TO ZERO
         2418'CF      D4   0311            CLRL    W^ACF$GL_SB                 ;INIT TO ZERO
                           0315
                           0315            SAVIPL  -(SP)                       ;SAVE THE CURRENT IPL
56       20 AE        D0   0318            MOVL    32(SP),R6                   ;GET ADDR OF DEVICE NAME
```

```
             55  86   9A 031C         MOVZBL  (R6)+,R5                        ;GET SIZE OF DEVICE NAME
             7E  55   7D 031F         MOVQ    R5,-(SP)                        ;FORM DESCRIPTOR
             51  5E   D0 0322         MOVL    SP,R1                           ;ADDRESS OF DESCRIPTOR
       00000000'GF    16 0325         JSB     G^IOC$SEARCHALL                 ;SEARCH FOR DEVICE
                 8E   7C 032B         CLRQ    (SP)+                           ;GET RID OF TRASH
                         032D         SETIPL  (SP)+                           ;RESTORE OLD IPL
        2418'CF  53   D0 0330         MOVL    R3,W^ACF$GL_SB                  ;STUFF THE SYSTEM BLOCK
                 04   12 0335         BNEQ    20$                             ;NO ERROR, CONTINUE
                 50   D4 0337         CLRL    R0                              ;INDICATE ERROR
                 5E   11 0339         BRB     70$                             ;EXIT
                         033B
             0B  50   E8 033B  20$:   BLBS    R0,25$                          ;SUCCES - FOUND DEVICE
        0908 8F  50   B1 033E         CMPW    R0,#SS$_NOSUCHDEV               ;CHECK IF ERROR WAS "UNIT NOT FOUND"
                 36   12 0343         BNEQ    60$                             ;IF NOT, PUNT
                 51   D5 0345         TSTL    R1                              ;SEE IF WE GOT BACK A UCB ADDRESS
                 32   12 0347         BNEQ    60$                             ;IF NON-ZERO, IS LISTHEAD - NO DDB FOUND
    00002400'EF  52   D0 0349  25$:   MOVL    R2,L^ACF$GL_DDB                 ;ADDRESS OF DDB
          54  04 A2   D0 0350         MOVL    DDB$L_UCB(R2),R4                ;GET ADDRESS OF FIRST UCB
                 25   13 0354         BEQL    60$                             ;IF NO UCB, EXIT WITH OTHER FIELDS ZERO
          51  24 A4   D0 0356         MOVL    UCB$L_CRB(R4),R1                ;GET ADDR OF CRB
    0000240C'EF  51   D0 035A         MOVL    R1,L^ACF$GL_CRB                 ;SAVE
      2408'CF  2C A1  D0 0361         MOVL    CRB$L_INTD+VEC$L_IDB(R1),W^ACF$GL_IDB ;GET ADDR OF IDB
                         0367
     54 A4    20 AE   B1 0367  30$:   CMPW    32(SP),UCB$W_UNIT(R4)           ;IS UCB ALREADY LOADED?
                 08   13 036C         BEQL    50$                             ;BRANCH IF IT IS
          54  30 A4   D0 036E  40$:   MOVL    UCB$L_LINK(R4),R4               ;GET ADDR OF NEXT UCB
                 F3   12 0372         BNEQ    30$                             ;BR IF THERE IS ONE
                 05   11 0374         BRB     60$                             ;EXIT WITH UCB = 0
                         0376
        2404'CF  54   D0 0376  50$:   MOVL    R4,W^ACF$GL_UCB
    00002410'EF  52   D0 037B  60$:   MOVL    R2,ACF$GL_LASTDDB               ;LAST DDB IN LIST AS SEARCHED
    50   00000000'GF  DE 0382         MOVAL   G^SCS$GA_LOCALSB,R0             ;GET ADDRESS OF LOCAL SYSTEM BLOCK
             50  53   D1 0389         CMPL    R3,R0                           ;IS THIS SB LOCAL?
                 08   13 038C         BEQL    65$                             ;YES, LEAVE NOW
              18 A3   7D 038E         MOVQ    SB$B_SYSTEMID(R3),-
       0000244C'EF       0391                 L^BOO$GQ_CONSYSID               ;NO, SET IN THE SYSTEM ID
                         0396
             50  01   D0 0396  65$:   MOVL    #1,R0                           ;SUCCESS
                      05 0399  70$:   RSB
                         039A
                         039A         .DSABL  LSB
                         039A
```

```
                                  039A        .SBTTL  Reset routines BOO$RESETLIST and BOO$CONRESET and BOO$MSCP_RESET
                                  039A   ;
                                  039A   ; BOO$CONRESET - Reset values for connect command
                                  039A   ;
                                  039A
                              00000000   .PSECT  PAGED_CODE      rd,nowrt,exe,long
                                  0000
                            0000  0000   .Entry  BOO$CONRESET, ^M<>                      ; Null entry mask
                                  0002
0000245C'EF   00000200'EF   9E    0002          MOVAB   L^BOO$AB_PRMBUF,BOO$GL_NEXTSTR  ; Reset for string allocation
        00002428'EF   01    CE    000D          MNEGL   #1,BOO$GL_CONCREG         ; Null control register
        0000243C'EF   01    CE    0014          MNEGL   #1,BOO$GL_CONAUNIT        ; Null adapter unit
        00002434'EF   01    CE    001B          MNEGL   #1,BOO$GL_CONVECT         ; Null vector
        00002438'EF   01    D0    0022          MOVL    #1,BOO$GL_CONNUMV         ; Default number of vectors
        00002424'EF   02    CE    0029          MNEGL   #2,BOO$GL_CONADP          ; Invalidate adapter TR value
              00002440'EF   D4    0030          CLRL    BOO$GL_CONDEV             ; Clear device name pointer
              00002444'EF   D4    0036          CLRL    BOO$GL_CONDRV             ; and driver name pointer
              00002448'EF   D4    003C          CLRL    BOO$GL_CONUNITS           ; and maximum units
              0000244C'EF   7C    0042          CLRQ    BOO$GQ_CONSYSID           ; and system id
              00002458'EF   D4    0048          CLRL    BOO$GL_CONFLAGS           ; and flags
        00002430'EF   01    D0    004E          MOVL    #1,L^BOO$GL_CONNUMU       ; Set number of units to 1
              0000241C'EF   D4    0055          CLRL    BOO$GL_COMBO_VECTOR_OFFSET; Set vector offset from combo vectors to
              00002420'EF   D4    005B          CLRL    BOO$GL_COMBO_CSR_OFFSET ; Set CSR offset from combo CSR to 0
                            04    0061          RET                               ; Return
                                  0062   ;
                                  0062   ;      BOO$RESETLIST - Reset select list values
                                  0062   ;
                            0000  0062   .Entry  BOO$RESETLIST, ^M<>             ; Null entry mask
                                  0064
              00002460'EF   D4    0064          CLRL    BOO$GL_SELECT             ; Zap select list pointer
0000245C'EF   00000200'EF   9E    006A          MOVAB   BOO$AB_PRMBUF,BOO$GL_NEXTSTR ; Set next string address
              00002614'EF   D4    0075          CLRL    BOO$GT_SAVE_DEVNAME       ; Clear autolog string
              00002610'EF   D4    007B          CLRL    OUTLEN                    ; Clear autolog output size
0000262C'EF   00002630'EF   DE    0081          MOVAL   OUTBUF_STR,OUTBUF+4       ; Set address in descriptor of block
              00002497'EF   94    008C          CLRB    BOO$AL_ACF+ACF$B_AFLAG    ; Clear ACF flags
        00002430'EF   01    D0    0092          MOVL    #1,L^BOO$GL_CONNUMU       ; Set number of units to 1
                            04    0099          RET                               ; and return
                                  009A
                                  009A   ;
                                  009A   ; BOO$MSCP_RESET - Reset the MSCP server initialization argument list
                                  009A   ;
                            003C  009A   .Entry  BOO$MSCP_RESET, ^M<R2,R3,R4,R5> ; Entry mask
                                  009C
          FF5F CF   00      FB    009C          CALLS   #0,BOO$CONRESET           ; Reset the connect command globals
          50   0084 8F      3C    00A1          MOVZWL  #SS$_DEVOFFLINE,R0        ; Assume error
           00000000'GF      D5    00A6          TSTL    G^SCS$GL_CDL              ; SCS loaded?
                    2C      13    00AC          BEQL    10$                       ; If eql no, error
          50   02C4 8F      3C    00AE          MOVZWL  #SS$_DEVACTIVE,R0         ; Assume error
     50   03   00   02      F0    00B3          INSV    #2,#0,#3,R0               ; Set E class error status
           00000000'GF      D5    00B8          TSTL    G^SCS$GL_MSCP             ; If neq already loaded
                    1A      12    00BE          BNEQ    10$                       ; Exit with error
00002444'GF   000025EC'EF   DE    00C0          MOVAL   MSCP_NAME,G^BOO$GL_CONDRV; Set pointer to MSCP server name
                    30      28    00CB          MOVC3   #MSCP_ARG_LIST_SIZE,-     ; Set up default argument list for
000025BC'GF   0000258C'EF         00CD                  MSCP_ARG_LIST,G^BOO$GL_LOAD_ARGS; MSCP server init routine
               50   01      D0    00D7          MOVL    #1,R0                     ; Set success
                            04    00DA   10$:   RET                               ; and return
                                  00DB
                              00DB  ;
                              00DB  ; BOO$MSCP_ARG - Load MSCP arguments
                              00DB  ;
                        0000  00DB  .Entry  BOO$MSCP_ARG, ^M<>                ; Entry mask
                              00DD
          50    20 AC   D0    00DD          MOVL    TPA$L_PARAM(AP),R0         ; Get longword offset
                1C AC   D0    00E1          MOVL    TPA$L_NUMBER(AP),-         ; Load argument value
     000025BC'GF40            00E4                  G^BOO$GL_LOAD_ARGS[R0]     ;
          50    01      D0    00EA          MOVL    #1,R0                      ; Set success
                        04    00ED          RET                                ; and return
                              00EE
                              00EE  ;
                              00EE  ;
                              00EE  ; BOO$MAKLIST - Make a select list entry
                              00EE  ;
                        007C  00EE  .Entry  BOO$MAKLIST, ^M<R2,R3,R4,R5,R6> ; Entry mask
                              00F0
  56   0000245C'EF      D0    00F0          MOVL    L^BOO$GL_NEXTSTR,R6        ; Get pointer to next available string space
       00002460'EF      D5    00F7          TSTL    L^BOO$GL_SELECT            ; Is selection pointer already set
                07      12    00FD          BNEQ    10$                        ; Yes, continue to add entry
  00002460'EF   56      D0    00FF          MOVL    R6,L^BOO$GL_SELECT         ; Else set pointer to first select entry
          50    10 AC   D0    0106  10$:    MOVL    TPA$L_TOKENCNT(AP),R0      ; Get string length
             86   50    90    010A          MOVB    R0,(R6)+                   ; Set count for string
   66    14 BC    50    28    010D          MOVC3   R0,@TPA$L_TOKENPTR(AP),(R6) ; Copy string body
                  63    94    0112          CLRB    (R3)                       ; Mark end of list
  0000245C'EF     53    D0    0114          MOVL    R3,L^BOO$GL_NEXTSTR        ; Save next string address
             50   01    D0    011B          MOVL    #1,R0                      ; Set success status
                        04    011E          RET                                ;
```

```
                                                                 G 3
CONFIGUTL                     - SYSGEN UTILITIES FOR CONFIGURE PROCESS 15-SEP-1984 23:46:56  VAX/VMS Macro V04-00       Page  23
V04-002                       BOO$CONADP - Set connect adapter number  14-SEP-1984 16:09:11  [BOOTS.SRC]SYSGEN.MAR;3          (4)

                              011F  1030         .SBTTL  BOO$CONADP - Set connect adapter number
                              011F  1031
                        0000  011F  1032 .Entry  BOO$CONADP, ^M<>                  ;
     00002424'EF    1C AC   D0  0121  1033         MOVL    TPA$L_NUMBER(AP),L^BOO$GL_CONADP ; Set adapter number
                          04  0129  1034         RET                               ; and return
                              012A  1035
                        0000  012A  1036 .Entry  BOO$CONNLADP ^M<>                 ; Connect with null adapter
       00002424'EF    01   CE  012C  1037         MNEGL   #1,L^BOO$GL_CONADP        ; Clear adapter number
                          04  0133  1038         RET                               ; and return
                              0134  1039
                        0000  0134  1040 .Entry  BOO$CONVECOFFSET, ^M<>            ; Offset from start of combo vectors
                     1C AC   D0  0136  1041         MOVL    TPA$L_NUMBER(AP),-        ; Set offset value
                 0000241C'EF      0139  1042                 L^BOO$GL_COMBO_VECTOR_OFFSET
                          04  013E  1043         RET                               ; and return
                              013F  1044
                        0000  013F  1045 .Entry  BOO$CONCSROFFSET, ^M<>            ; Offset from start of combo CSRs
                     1C AC   D0  0141  1046         MOVL    TPA$L_NUMBER(AP),-        ; Set offset value
                 00002420'EF      0144  1047                 L^BOO$GL_COMBO_CSR_OFFSET
                          04  0149  1048         RET                               ; and return
                              014A  1049
                        0000  014A  1050 .Entry  BOO$CONCREG, ^M<>                 ; Control register address
00002428'EF   1C AC    0D    00   EF  014C  1051         EXTZV   #0,#13,TPA$L_NUMBER(AP),L^BOO$GL_CONCREG; Set control register
                          04  0156  1052         RET                               ; and return
                              0157  1053
                        0000  0157  1054 .Entry  BOO$CONCVEC, ^M<>                 ; Set controller vector
     1C AC    FFFFFE03 8F   CB  0159  1055         BICL3   #^XFFFFFE03,TPA$L_NUMBER(AP),L^BOO$GL_CONVECT ; Set vector offset
                 00002434'EF      0161
                          04  0166  1056         RET                               ; and return
                              0167  1057
                        0000  0167  1058 .Entry  BOO$CONCNUM, ^M<>                 ; Number of vectors
     00002438'EF    1C AC   D0  0169  1059         MOVL    TPA$L_NUMBER(AP),L^BOO$GL_CONNUMV ; Set number of vectors
                          04  0171  1060         RET                               ; and return
                              0172  1061
                        0000  0172  1062 .Entry  BOO$CONAUNIT, ^M<>                ; Adapter unit number
     0000243C'EF    1C AC   D0  0174  1063         MOVL    TPA$L_NUMBER(AP),L^BOO$GL_CONAUNIT; Set adapter unit number
                          04  017C  1064         RET                               ; and return
                              017D  1065
                        007C  017D  1066 .Entry  BOO$CONDRVNAM, ^M<R2,R3,R4,R5,R6> ; Entry mask (R2-R6)
                              017F  1067
         56   0000245C'EF   D0  017F  1068         MOVL    L^BOO$GL_NEXTSTR,R6       ; Address of next string storage
       00002444'EF    56   D0  0186  1069         MOVL    R6,BOO$GL_CONDRV          ; Save pointer to driver name
               86    10 AC   90  018D  1070         MOVB    TPA$L_TOKENCNT(AP),(R6)+      ; Set count for string
  0000245C'EF   56    10 AC   C1  0191  1071         ADDL3   TPA$L_TOKENCNT(AP),R6,BOO$GL_NEXTSTR ; Mark string allocated
        66   14 BC    10 AC   28  019A  1072         MOVC3   TPA$L_TOKENCNT(AP),@TPA$L_TOKENPTR(AP),(R6)     ; Copy string
                  50    01   D0  01A0  1073         MOVL    #1,R0                     ; and return success
                          04  01A3  1074         RET                               ;
                              01A4  1075
                        00FC  01A4  1076 .Entry  BOO$DEVNAME, ^M<R2,R3,R4,R5,R6,R7> ; Device name/unit
                              01A6  1077
         56   0000245C'EF   D0  01A6  1078         MOVL    BOO$GL_NEXTSTR,R6         ; Get pointer to next available string
             54    14 AC   D0  01AD  1079         MOVL    TPA$L_TOKENPTR(AP),R4     ; Get pointer to string
             53    10 AC   D0  01B1  1080         MOVL    TPA$L_TOKENCNT(AP),R3     ; And number of characters
             00002588'EF   D4  01B5  1081         CLRL    FULL_NAME_PTR             ; Initialize full device name
                  57    86   9E  01B8  1082         MOVAB   (R6)+,R7                  ; Save pointer
         64    53    24   3A  01BE  1083         LOCC    #^A/$/,R3,(R4)            ; Find any possible '$'
                        22   13  01C2  1084         BEQL    8$                        ; None, just continue
       00002588'EF    57   D0  01C4  1085         MOVL    R7,FULL_NAME_PTR          ; Store pointer
```

```
      55    53    50    C3   01CB   1086           SUBL3    R0,R3,R5                 ; Number of characters in node
      67    55    01    81   01CF   1087           ADDB3    #1,R5,(R7)               ; Set in size (incl "$")
                  03    BB   01D3   1088           PUSHR    #^M<R0,R1>               ; Save registers
      66    64    53    28   01D5   1089           MOVC3    R3,(R4),(R6)             ; Copy full string
            56    53    D0   01D9   1090           MOVL     R3,R6                    ; Save ending address
                  03    BA   01DC   1091           POPR     #^M<R0,R1>               ; Restore registers
      53    50    01    C3   01DE   1092           SUBL3    #1,R0,R3                 ; Number of characters left
      54    51    01    C1   01E2   1093           ADDL3    #1,R1,R4                 ; Pointer to string
            55    86    9E   01E6   1094 8$:       MOVAB    (R6)+,R5                 ; Save pointer to count byte
                  65    94   01E9   1095           CLRB     (R5)                     ; Initialize count to zero
                  52    D4   01EB   1096           CLRL     R2                       ; Initialize unit accumulator
            50    84    9A   01ED   1097 10$:      MOVZBL   (R4)+,R0                 ; Get a character from device name
            30    50    91   01F0   1098           CMPB     R0,#^A/0/                ; And check for a digit
                  05    1F   01F3   1099           BLSSU    20$                      ; Branch if not
            39    50    91   01F5   1100           CMPB     R0,#^A/9/                ; Final check for digit
                  0F    1B   01F8   1101           BLEQU    40$                      ; Yes it is
            86    50    90   01FA   1102 20$:      MOVB     R0,(R6)+                 ; Part of device name
                  65    96   01FD   1103           INCB     (R5)                     ; Increase count
                  67    96   01FF   1104           INCB     (R7)                     ; Including nodename
               E9 53    F5   0201   1105           SOBGTR   R3,10$                   ; Continue
                  16    11   0204   1106           BRB      50$                      ;
            50    84    9A   0206   1107 30$:      MOVZBL   (R4)+,R0                 ; Get another digit
            50    30    C2   0209   1108 40$:      SUBL     #^A/0/,R0                ; Get value
            52    0A    C4   020C   1109           MULL     #10,R2                   ; Scale accumulator before adding digit
                  2F    19   020F   1110           BLSS     60$                      ; Error
            50    09    D1   0211   1111           CMPL     #9,R0                    ; Check for numeric
                  2A    19   0214   1112           BLSS     60$                      ; Error if not
            52    50    C0   0216   1113           ADDL     R0,R2                    ; And add new digit
               EA 53    F5   0219   1114           SOBGTR   R3,30$                   ; Continue for entire unit number
  0000245C'EF     56    D0   021C   1115 50$:      MOVL     R6,BOO$GL_NEXTSTR        ; Save updated string pointer
  0000242C'EF     52    D0   0223   1116           MOVL     R2,BOO$GL_CONCUNIT       ; Set unit number
  0000243C'EF     52    D0   022A   1117           MOVL     R2,BOO$GL_CONAUNIT       ; Assume same for adapter unit
  00002440'EF     55    D0   0231   1118           MOVL     R5,BOO$GL_CONDEV         ; Save device name pointer
                  65    95   0238   1119           TSTB     (R5)                     ; Must not be null device name
                  04    13   023A   1120           BEQL     60$                      ; Error if so
            50    01    D0   023C   1121           MOVL     #1,R0                    ; Return success
                        04   023F   1122           RET                               ; and return
                  50    D4   0240   1123 60$:      CLRL     R0                       ; Return error status
                        04   0242   1124           RET                               ;
                             0243   1125
                      0000   0243   1126 .Entry    BOO$CONUNITS, ^M<>                ; Maximun units to be connected
00002448'EF     1C AC   D0   0245   1127           MOVL     TPA$L_NUMBER(AP),L^BOO$GL_CONUNITS ; Set maximum units
                        04   024D   1128           RET                               ; and return
                             024E   1129
                      0000   024E   1130 .Entry    BOO$CONSYSID_LOW, ^M<>            ; System ID
0000244C'EF     1C AC   D0   0250   1131           MOVL     TPA$L_NUMBER(AP), -
                             0258   1132                    L^BOO$GQ_CONSYSID        ; Set System ID (low longword)
                        04   0258   1133           RET                               ; and return
                             0259   1134
                      0000   0259   1135 .Entry    BOO$CONSYSID_HIGH, ^M<>           ; System ID
00002450'EF     1C AC   D0   025B   1136           MOVL     TPA$L_NUMBER(AP), -
                             0263   1137                    L^BOO$GQ_CONSYSID+4      ; Set System ID (high longword)
                        04   0263   1138           RET                               ; and return
                             0264   1139
                      0000   0264   1140 .Entry    BOO$CONSOLE, ^M<>                 ; Connect console block stor. device
                             0266   1141
  00002424'EF     01    CE   0266   1142           MNEGL    #1,L^BOO$GL_CONADP       ; No adapter
```

```
          0000243C'EF    01   D0  026D          MOVL    #1,L^BOO$GL_CONAUNIT      ; Set adapter unit = 1 (not used)
          0000242C'EF    01   D0  0274          MOVL    #1,L^BOO$GL_CONCUNIT      ; Set unit = 1
 00002440'EF    000024D6'EF   9E  027B          MOVAB   L^CONSNAME,L^BOO$GL_CONDEV  ; Set device name pointer
          00002438'EF    02   D0  0286          MOVL    #2,L^BOO$GL_CONNUMV       ; Set 2 vectors
                00002428'EF   D4  028D          CLRL    L^BOO$GL_CONCREG          ; No control register
          00002430'EF    01   D0  0293          MOVL    #1,L^BOO$GL_CONNUMU       ; Set number of units to 1
          00002448'EF    02   D0  029A          MOVL    #2,L^BOO$GL_CONUNITS      ; Set max units to 2 (OPA0 is 1st unit)
                00000000'EF   16  02A1          JSB     IOGEN$CONSOLE             ; Do cpu dependent stuff
                      50 01   D0  02A7          MOVL    #1,R0
                              04  02AA          RET
```

```
                                02AB          .SBTTL  BOO$CONNECT - Connnect specified device and load driver
                                02AB    ;
                                02AB    ; BOO$CONNECT - Allows a single device to be introduced, appropriate data
                                02AB    ;               structures allocated and initialized, the driver loaded if
                                02AB    ;               required and the controller and device initialized.
                                02AB    ;
                          0FFC  02AB    .Entry  BOO$CONNECT, ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>       ;
                                02AD
                   FD50'   30   02AD            BSBW    BOO$LOCK_GEN             ; Lock SYSGEN database
                  7F 50    E9   02B0            BLBC    R0,70$                   ; If error, exit
                                02B3
                                02B3    ;
                                02B3    ; Value of BOO$GL_CONADP
                                02B3    ;
                                02B3    ;       0 or greater    => /ADADPTER=n specified
                                02B3    ;       -1              => /NOADAPTER specified
                                02B3    ;       -2              => not specified
                                02B3    ;
                                02B3
           00002424'EF     D5   02B3    5$:     TSTL    L^BOO$GL_CONADP          ; Has an adapter been specified?
                    44     18   02B9            BGEQ    20$                      ; If so, branch
00002424'EF  FFFFFFFF 8F   D1   02BB            CMPL    #-1,L^BOO$GL_CONADP      ; Null adapter?
                    28     13   02C6            BEQL    10$                      ; Branch if yes
00002424'EF  FFFFFFFE 8F   D1   02C8            CMPL    #-2,L^BOO$GL_CONADP      ; None specified in CONNECT?
                    09     13   02D3            BEQL    7$                       ; Figure it out from the database
         50  007C80D2 8F   D0   02D5            MOVL    #SYSG$_NOADAPTER,R0      ; Set no adapter specified error
                    45     11   02DC            BRB     60$                      ; exit
                                02DE
                                02DE    7$:     $CMKRNL_S W^CONN_ADAP            ; Get adapter number from I/O database
                 35 50     E9   02EB            BLBC    R0,60$                   ; Exit with error
                    C3     11   02EE            BRB     5$                       ; Dispatch now on adapter type
                                02F0
                                02F0    10$:    $CMKRNL_S       W^CONNLADP       ; Change mode to see data base
                    0D     11   02FD            BRB     30$                      ; Continue
                                02FF
                                02FF    20$:    $CMKRNL_S       W^CONNECT        ; Change mode to see data base
                 0E 50     E9   030C    30$:    BLBC    R0,40$                   ; Error occured
00000000'EF  0000248C'EF   FA   030F            CALLG   L^BOO$AL_ACF,IOGEN$LOADER ; Load database and driver
                 03 50     E8   031A            BLBS    R0,50$                   ; Branch if success
                 FCE0'     30   031D    40$:    BSBW    PUTERROR                 ; Give error message
              50    01     D0   0320    50$:    MOVL    #1,R0                    ; Set success for parser
                    50     DD   0323    60$:    PUSHL   R0                       ; Save error status
                 FCD8'     30   0325            BSBW    BOO$UNLOCK_GEN           ; Unlock SYSGEN database
                 03 50     E8   0328            BLBS    R0,65$                   ; If no error, continue
                 FCD2'     30   032B            BSBW    PUTERROR                 ; Give error message
                  50 8ED0       032E    65$:    POPL    R0                       ; Restore status
                    04          0331            RET                              ;
                 FCCB'     30   0332    70$:    BSBW    PUTERROR                 ; Give error message
                    04          0335            RET                              ;
                                0336    ;
                                0336    ; Local routine to get adapter number from I/O database
                                0336    ; Must be called by a CMKRNL since SGN$GET_DEVICE must be called
                                0336    ; in Kernel mode.
                                0336    ;
                          0000  0336    .Entry  CONN_ADAP, ^M<>
                                0338
         7E  0000242C'EF   3C   0338            MOVZWL  L^BOO$GL_CONCUNIT,-(SP) ; Unit number
```

```
          00002440'EF   DD  033F              PUSHL   L^BOO$GL_CONDEV          ; Device name
          000002D4'EF   16  0345              JSB     SGN$GET_DEVICE_LOCK_IODB; Get device data base addresses
               5E   08  C0  034B              ADDL2   #8,SP                    ; Pop off input parameters
                            034E
     50   00002408'EF   D0  034E              MOVL    L^ACF$GL_IDB,R0          ; Address of IDB
                    09  12  0355              BNEQ    5$                       ; Error if zero
     50   007C80D2 8F   D0  0357              MOVL    #SYSG$_NOADAPTER,R0      ; Set no adapter specified error
                    18  11  035E              BRB     20$                      ; Branch to exit
                            0360
     00002424'EF    01  CE  0360      5$:     MNEGL   #1,L^BOO$GL_CONADP       ; Assume null adapter
            50   14 A0  D0  0367              MOVL    IDB$L_ADP(R0),R0         ; Address of ADP block
                    08  13  036B              BEQL    10$                      ; Null adapter if zero
  00002424'EF    0C A0  3C  036D              MOVZWL  ADP$W_TR(R0),L^BOO$GL_CONADP ;Set adapter number
                            0375
                50  01  D0  0375      10$:    MOVL    #1,R0                    ; Set success
                        04  0378      20$:    RET                              ; Return
                            0379
```

```
                                 0379  .ENABL  LSB
                                 0379  ; Connect with null adapter
                                 0379
                           0FFC  0379  .Entry  CONNLADP, ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
                                 037B
          50   00002588'EF  D0   037B          MOVL    L^FULL_NAME_PTR,R0       ; Try full device name
                        07  12   0382          BNEQ    5$                       ; Good, continue
          50   00002440'EF  D0   0384          MOVL    L^BOO$GL_CONDEV,R0       ; Use normal name
          5A   0000248C'EF  9E   0388  5$:     MOVAB   L^BOO$AL_ACF,R10         ; Address ACF
                        6A  D4   0392          CLRL    ACF$L_ADAPTER(R10)       ; Set no adapter
                     04 AA  D4   0394          CLRL    ACF$L_CONFIGREG(R10)     ; Set address of config reg
                     08 AA  B4   0397          CLRW    ACF$W_AVECTOR(R10)       ; Set SCB offset for adapter
                        01  E1   039A          BBC     #ACF$V_CRBBLT,-          ; Br. if CRB built flag is clear
        4C 00002458'EF           039C                  BOO$GL_CONFLAGS,17$
        6A     00002454'EF  D0   03A2          MOVL    BOO$GL_CONCRB,ACF$L_ADAPTER(R10)  ; Store CRB address
                        43  11   03A9          BRB     17$                      ; Join common code
                                 03AB
                           0FFC  03AB  .Entry  CONNECT, ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>    ;
                                 03AD
          5B   FFFFFFFC'GF  9E   03AD          MOVAB   G^IOC$GL_ADPLIST-ADP$L_LINK,R11 ; Get address of adapter list
               5B    04 AB  D0   03B4  10$:    MOVL    ADP$L_LINK(R11),R11      ; Flink onward through adapter list
                        08  12   03B8          BNEQ    15$                      ; Continue if another adapter
          50   007C80BA 8F  D0   03BA          MOVL    #SYSG$_INVADAP,R0        ; Set invalid adapter error
                            04   03C1          RET                              ; Return
                                 03C2
     00002424'EF    0C AB   B1   03C2  15$:    CMPW    ADP$W_TR(R11),L^BOO$GL_CONADP   ; Is this the specified TR?
                        E8  12   03CA          BNEQ    10$                      ; No, try another
          5A   0000248C'EF  9E   03CC          MOVAB   L^BOO$AL_ACF,R10         ; Get address of ACF
                  6A    5B  D0   03D3          MOVL    R11,ACF$L_ADAPTER(R10)   ; Set address of ADP
               04 AA    6B  D0   03D6          MOVL    ADP$L_CSR(R11),ACF$L_CONFIGREG(R10)      ; Set address of config reg
  50    1C AB  00000000'GF  C3   03DA          SUBL3   G^EXE$GL_SCB, -          ; Calculate offset into SCB of
                                 03E3                  ADP$L_AVECTOR(R11),R0    ; adapter's interrupt vectors.
               08 AA    50  B0   03E3          MOVW    R0,ACF$W_AVECTOR(R10)    ; Store offset in ACF.
          50   00002440'EF  D0   03E7          MOVL    L^BOO$GL_CONDEV,R0       ; Device name
                                 03EE
                                 03EE  17$:
     14 AA     00002440'EF  D0   03EE          MOVL    BOO$GL_CONDEV,ACF$L_DEVNAME(R10); Set pointer to device name
                                 03F6
                                 03F6  ; Now try to get driver name from DDB if it exists and load BOO$GQ_CONSYSID
                                 03F6  ; if HSC device.
                                 03F6
          7E   0000242C'EF  3C   03F6          MOVZWL  L^BOO$GL_CONCUNIT,-(SP) ; Unit number
                        50  DD   03FD          PUSHL   R0                       ; Device name
               000002D4'EF  16   03FF          JSB     SGN$GET_DEVICE_LOCK_IODB; Get device data base addresses
                  5E    08  C0   0405          ADDL2   #8,SP                    ; Pop off input parameters
                  08    50  E8   0408          BLBS    R0,20$                   ; All okay
          50   007C9010 8F  D0   040B          MOVL    #SYSG$_NODEV,R0          ; Set error code - "Device not known"
                            04   0412          RET                              ; Leave
                                 0413
                        05  E2   0413  20$:    BBSS    #ACF$V_GETDONE,-
        00 00002458'EF           0415                  L^BOO$GL_CONFLAGS,21$    ; Notify LOADER that GET was done
                                 041B
     18 AA     00002444'EF  D0   041B  21$:    MOVL    BOO$GL_CONDRV,ACF$L_DRVNAME(R10); And driver name
                        31  14   0423          BGTR    30$                      ; Branch if driver specified
          51   00002400'EF  D0   0425          MOVL    ACF$GL_DDB,R1            ; DDB address
                        07  13   042C          BEQL    25$                      ; Branch if none
        18 AA       24 A1   DE   042E          MOVAL   DDB$T_DRVNAME(R1),ACF$L_DRVNAME(R10) ; Address from DDB
```

```
                         21    11  0433           BRB     30$                       ; Branch around name hackery
                                   0435
         56   0000245C'EF      D0  0435 25$:      MOVL    L^BOO$GL_NEXTSTR,R6       ; Get address of next free space
                 18 AA   56    D0  043C           MOVL    R6,ACF$L_DRVNAME(R10)     ; Set as driver name address
                    86   08    90  0440           MOVB    #8,(R6)+                  ; Set count for string
   66   52455649 52442020 8F   7D  0443           MOVQ    #^A/  DRIVER/,(R6)        ; Set driver suffix
                 51   14 AA    D0  044E           MOVL    ACF$L_DEVNAME(R10),R1     ; Pointer to device name
                 66   01 A1    B0  0452           MOVW    1(R1),(R6)                ; Form default driver name
                                   0456
      0A AA   0000243C'EF      90  0456 30$:      MOVB    BOO$GL_CONAUNIT,ACF$B_AUNIT(R10); Set adapter unit
      21 AA   00002430'EF      90  045E           MOVB    L^BOO$GL_CONNUMU,ACF$B_NUMUNIT(R10)
                                   0466                                             ; Store number of units to configure
      0B AA   00002458'EF      90  0466           MOVB    BOO$GL_CONFLAGS,ACF$B_AFLAG(R10)  ; Store flags
 0000241C'EF  00002434'EF      A1  046E           ADDW3   BOO$GL_CONVECT,BOO$GL_COMBO_VECTOR_OFFSET,-;
                      10 AA        0479                   ACF$W_CVECTOR(R10)        ; Set vector address
  50   0000241C'EF    08   02  EF  047B           EXTZV   #2,#8,BOO$GL_COMBO_VECTOR_OFFSET,R0; Save vector offset in longwords
          [illegible] AA   50  90  0484           MOVB    R0,ACF$B_COMBO_VECTOR_OFFSET(R10);
                                   0488 ;
                                   0488 ; Set up ACF$L_CONTRLREG - can either be UNIBUS CSR or address of CI
                                   0488 ;                                 System id.
                                   0488 ;
              0000244C'EF      D5  0488           TSTL    BOO$GQ_CONSYSID           ; See if SYSIDLOW was specified
                       0A      13  048E           BEQL    40$                       ; Branch if not
      0C AA   0000244C'EF      9E  0490           MOVAB   BOO$GQ_CONSYSID, -
                                   0498                   ACF$L_CONTRLREG(R10)      ; Set system id address
                         22    11  0498           BRB     50$                       ; Branch
                                   049A ;
                                   049A ; Calculate system virtual address of UNIBUS CSR
                                   049A ;
 00002428'EF  00001000 8F      C1  049A 40$:      ADDL3   #UBA_IOBASE, -
                    0C AA          04A5                   BOO$GL_CONCREG, -
                                   04A7                   ACF$L_CONTRLREG(R10)      ; control register address
         0C AA   04 AA         C0  04A7           ADDL    ACF$L_CONFIGREG(R10), -
                                   04AC                   ACF$L_CONTRLREG(R10)      ; Add adapter va base
              00002420'EF      C0  04AC           ADDL    BOO$GL_COMBO_CSR_OFFSET,-; Add offset to get true CSR address
                    0C AA          04B2                   ACF$L_CONTRLREG(R10)      ;
              00002420'EF      8E  04B4           MNEGB   BOO$GL_COMBO_CSR_OFFSET,-; Calculate offset back to CSR start
                    20 AA          04BA                   ACF$B_COMBO_CSR_OFFSET(R10); Save offset
                                   04BC
      12 AA   0000242C'EF      B0  04BC 50$:      MOVW    BOO$GL_CONCUNIT, -
                                   04C4                   ACF$W_CUNIT(R10)          ; Set controller unit number
      1C AA   00002448'EF      B0  04C4           MOVW    BOO$GL_CONUNITS, -
                                   04CC                   ACF$W_MAXUNITS(R10)       ; Set maximum units
      1E AA   00002438'EF      90  04CC           MOVB    BOO$GL_CONNUMV, -
                                   04D4                   ACF$B_CNUMVEC(R10)        ; Set count of vectors
                    50   01    D0  04D4 55$:      MOVL    #1,R0                     ; Set success
                               04  04D7           RET                               ;
                                   04D8
                                   04D8                   .DSABL LSB
                                   04D8
                      04D8  1337         .SBTTL  BOO$LOAD - Load a driver or misc code if not already loaded
                      04D8  1338 ;
                      04D8  1339 ; BOO$LOAD -    Loads the driver or misc code if not already loaded.
                      04D8  1340 ;
                OFFC  04D8  1341 .Entry  BOO$LOAD, ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
                      04DA  1342
          52    D4    04DA  1343         CLRL    R2                          ; Clear reload flag
          05    11    04DC  1344         BRB     LOADRV                      ; And merge with common code
```

```
                          04DE          .SBTTL  BOO$RELOAD - Reload a specified driver
                          04DE  ;
                          04DE  ; BOO$RELOAD -  Reloads the specified driver replacing any existing copy
                          04DE  ;               unless there are busy units requiring the driver that would
                          04DE  ;               be replaced.
                          04DE  ;
                    0FFC  04DE  .Entry  BOO$RELOAD, ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;
                          04E0
          52    01    90  04E0          MOVB    #ACF$M_RELOAD,R2        ; Set flag to force reload
                          04E3  LOADRV:
                          04E3  ;
                          04E3  ; The first block of the file will be read to determine if it is a driver or
                          04E3  ; misc code by looking at the type field.
                          04E3  ;
       00002444'EF    DD  04E3          PUSHL   BOO$GL_CONDRV                   ; File name string
                6E    D6  04E9          INCL    (SP)                            ; Get past the count
    7E 00002444'FF    9A  04EB          MOVZBL  @BOO$GL_CONDRV,-(SP)            ; Length of file name
          57    5E    D0  04F2          MOVL    SP,R7                           ; Address of desriptor for file name
             FB08'    30  04F5          BSBW    BOO$EXEOPEN                     ; Open the file (default SYS$SYSTEM:.EXE)
             5E 50    E9  04F8          BLBC    R0,40$                          ; Error
          5E    08    C0  04FB          ADDL    #8,SP                           ; Clean up stack
    56 00002200'EF    9E  04FE          MOVAB   BOO$AB_LOADBUF,R6               ; Buffer for file read
          58    02    D0  0505          MOVL    #2,R8                           ; First block after image header
          59    01    D0  0508          MOVL    #1,R9                           ; One page
             FAF2'    30  050B          BSBW    BOO$READFILE
             48 50    E9  050E          BLBC    R0,40$                          ; Error
             FAEC'    30  0511          BSBW    BOO$FILCLOSE                    ; Close the currently open file
             42 50    E9  0514          BLBC    R0,40$                          ; Error
       0000220A'EF    91  0517          CMPB    BOO$AB_LOADBUF+SLV$B_TYPE,-
             62 8F        051D                  #DYN$C_LOADCODE                 ; Check for misc code
                3C    13  051F          BEQL    LOADCODE
             FADC'    30  0521          BSBW    BOO$LOCK_GEN                    ; Lock SYSGEN database
             32 50    E9  0524          BLBC    R0,40$                          ; If lbc, didn't get lock
    5A 0000248C'EF    9E  0527          MOVAB   L^BOO$AL_ACF,R10                ; Get base address for ACF block
 18 AA 00002444'EF    D0  052E          MOVL    BOO$GL_CONDRV,ACF$L_DRVNAME(R10)        ;
          0B AA 52    90  0536          MOVB    R2,ACF$B_AFLAG(R10)             ; Set flags for load or reload
             14 AA    D4  053A          CLRL    ACF$L_DEVNAME(R10)              ; No device name
   00000000'EF  6A    FA  053D          CALLG   (R10),L^IOGEN$LOADER            ; Load requested driver
             03 50    E8  0544          BLBS    R0,20$                          ; Continue if no error
             FAB6'    30  0547          BSBW    PUTERROR                        ; Give error message
                50    DD  054A  20$:    PUSHL   R0                              ; Save status
             FAB1'    30  054C          BSBW    BOO$UNLOCK_GEN                  ; Unlock SYSGEN database
             03 50    E8  054F          BLBS    R0,30$                          ; If no error, continue
             FAAB'    30  0552          BSBW    PUTERROR                        ; Give error message
             50 8ED0      0555  30$:    POPL    R0                              ; Restore status
                      04  0558          RET                                     ; Exit
             FAA4'    30  0559  40$:    BSBW    PUTERROR                        ; Give error message
                      04  055C          RET                                     ; Exit
                          055D  ;
                          055D  LOADCODE:
       00002444'EF    DD  055D          PUSHL   BOO$GL_CONDRV                   ; File name string
                6E    D6  0563          INCL    (SP)                            ; Get past the count
    7E 00002444'FF    9A  0565          MOVZBL  @BOO$GL_CONDRV,-(SP)            ; Length of file name
          57    5E    D0  056C          MOVL    SP,R7                           ; Address of desriptor for file name
             FA8E'    30  056F          BSBW    BOO$UFOOPEN                     ; Open the file for user access (default SYS
             22 50    E9  0572          BLBC    R0,10$                          ; Error
          5E    08    C0  0575          ADDL    #8,SP                           ; Clean up stack
```

```
        00002200'EF   9F  0578             PUSHAB  BOO$AB_LOADBUF              ; Use code buffer for return address array
                 51   DD  057E             PUSHL   R1                          ; Channel
                 02   DD  0580             PUSHL   #2                          ; Arg count
        50       5E   D0  0582             MOVL    SP,R0
                          0585             $CMKRNL_S ROUTIN = EXE$LOAD_CODE,-
                          0585                       ARGLST = (R0)
              04 50   E8  0594             BLBS    R0,20$
              FA66'   30  0597     10$:    BSBW    PUTERROR
                      04  059A             RET
                          059B
                          059B     20$:    $CMKRNL_S ROUTIN = LINK_CODE
              EA 50   E9  05AA             BLBC    R0,10$
                      04  05AD             RET
                          05AE     ;
                          05AE     LINK_CODE:
                  001C    05AE             .WORD   ^M<R2,R3,R4>
   52   00002200'GF   D0  05B0             MOVL    G^BOO$AB_LOADBUF,R2         ; Address of loaded code
             54   52  D0  05B7             MOVL    R2,R4                       ; Save address of loaded code
          53   10 A4  D0  05BA             MOVL    SLV$A_SYSVECS(R4),R3        ; Get address of vectors in SYS.EXE
        00000000'GF   16  05BE             JSB     G^EXE$LINK_VEC              ; Connect vectors to loaded routines.
              10 50   E9  05C4             BLBC    R0,10$                      ; Leave on error
   5C   000025BC'GF   DE  05C7             MOVAL   G^BOO$GL_LOAD_ARGS,AP       ; Argument list for initialization routine
          50   04 A4  D0  05CE             MOVL    SLV$L_INITRTN(R4),R0        ; Possible initialization routine
                 03   13  05D2             BEQL    10$                         ; None, leave
               6044   16  05D4             JSB     (R0)[R4]                    ; Call it
                      04  05D7     10$:    RET
                          05D8
```

```
                            05D8  1431          .SBTTL  BOO$GIVEHELP - Print Help information
                            05D8  1432 ;
                            05D8  1433 ;        Print Help Information
                            05D8  1434 ;
                      003C  05D8  1435 .Entry   BOO$GIVEHELP, ^M<R2,R3,R4,R5>       ;
                            05DA  1436
       00000000'GF     9F   05DA  1437          PUSHAB  G^LIB$GET_INPUT             ; Input routine
       00002574'EF     9F   05E0  1438          PUSHAB  HELP_FLAG                   ; Flags
       00002559'EF     9F   05E6  1439          PUSHAB  L^HELP_FILE                 ; Library
             08 AC     B0   05EC  1440          MOVW    TPA$L_STRINGCNT(AP),-
       00002578'EF          05EF  1441                  HELP_DESC                   ; Set length
             0C AC     D0   05F4  1442          MOVL    TPA$L_STRINGPTR(AP),-
       0000257C'EF          05F7  1443                  HELP_DESC+4                 ; Set address
       00002578'EF     9F   05FC  1444          PUSHAB  HELP_DESC                   ; Input string
                7E     D4   0602  1445          CLRL    -(SP)                       ; Width
       00000000'GF     9F   0604  1446          PUSHAB  G^LIB$PUT_OUTPUT            ; Output routine
00000000'GF     06     FB   060A  1447          CALLS   #6,G^LBR$OUTPUT_HELP        ; Call help routine
                            0611  1448
                       04   0611  1449          RET                                 ; Return with status
                            0612  1450
                            0612  1451          .END
```

```
                                                                   E  4
CONFIGUTL                          - SYSGEN UTILITIES FOR CONFIGURE PROCESS 15-SEP-1984 23:46:56  VAX/VMS Macro V04-00        Page  34
Symbol table                                                                14-SEP-1984 16:09:11  [BOOTS.SRC]SYSGEN.MAR;3             (4)

$$T2                              = 00000005                    BOO$DEVNAME                          000001A4 RG    06
$CLI.                             = 00002464 R      04          BOO$EXEOPEN                          ********   X   06
$CLI..                            = 00002480 R      04          BOO$FILCLOSE                         ********   X   06
ACF$B_AFLAG                       = 0000000B                    BOO$GB_FILELEN                       000024FD RG    04
ACF$B_AUNIT                       = 0000000A                    BOO$GIVEHELP                         000005D8 RG    06
ACF$B_CNUMVEC                     = 0000001E                    BOO$GL_CMDOPT                        ********   X   05
ACF$B_COMBO_CSR_OFFSET            = 00000020                    BOO$GL_COMBO_CSR_OFFSET              00002420 RG    04
ACF$B_COMBO_VECTOR_OFFSET         = 0000001F                    BOO$GL_COMBO_VECTOR_OFFSET           0000241C RG    04
ACF$B_NUMUNIT                     = 00000021                    BOO$GL_CONADP                        00002424 RG    04
ACF$C_LENGTH                      = 00000028                    BOO$GL_CONAUNIT                      0000243C RG    04
ACF$GL_CRB                          0000240C RG     04          BOO$GL_CONCRB                        00002454 RG    04
ACF$GL_DDB                          00002400 RG     04          BOO$GL_CONCREG                       00002428 RG    04
ACF$GL_DPT                          00002414 RG     04          BOO$GL_CONCUNIT                      0000242C RG    04
ACF$GL_IDB                          00002408 RG     04          BOO$GL_CONDEV                        00002440 RG    04
ACF$GL_LASTDDB                      00002410 RG     04          BOO$GL_CONDRV                        00002444 RG    04
ACF$GL_SB                           00002418 RG     04          BOO$GL_CONFLAGS                      00002458 RG    04
ACF$GL_UCB                          00002404 RG     04          BOO$GL_CONNUMU                       00002430 RG    04
ACF$L_ADAPTER                     = 00000000                    BOO$GL_CONNUMV                       00002438 RG    04
ACF$L_CONFIGREG                   = 00000004                    BOO$GL_CONUNITS                      00002448 RG    04
ACF$L_CONTRLREG                   = 0000000C                    BOO$GL_CONVECT                       00002434 RG    04
ACF$L_DEVNAME                     = 00000014                    BOO$GL_FILEADDR                      000024F9 RG    04
ACF$L_DRVNAME                     = 00000018                    BOO$GL_LOAD_ARGS                     000025BC RG    04
ACF$M_RELOAD                      = 00000001                    BOO$GL_NEXTSTR                       0000245C RG    04
ACF$V_CRBBLT                      = 00000001                    BOO$GL_PARINUSE                      000024FE RG    04
ACF$V_GETDONE                     = 00000005                    BOO$GL_RETSAVE                       000024C4 RG    04
ACF$V_NOLOAD_DB                   = 00000003                    BOO$GL_SELECT                        00002460 RG    04
ACF$W_AVECTOR                     = 00000008                    BOO$GQ_CMDESC                      = 0000246C RG    04
ACF$W_CUNIT                       = 00000012                    BOO$GQ_CONSYSID                      0000244C RG    04
ACF$W_CVECTOR                     = 00000010                    BOO$GQ_LIMITS                        000024B4 RG    04
ACF$W_MAXUNITS                    = 0000001C                    BOO$GQ_RETADR                        000024BC RG    04
ADP$L_AVECTOR                     = 0000001C                    BOO$GT_ACTIVE                        0000250A RG    04
ADP$L_CSR                         = 00000000                    BOO$GT_CURRENT                       00002502 RG    04
ADP$L_LINK                        = 00000004                    BOO$GT_CVNAME                        000024DE RG    04
ADP$W_TR                          = 0000000C                    BOO$GT_DDNAME                        000024F0 RG    04
AUTOLOG                             00000206 RG     05          BOO$GT_DEFAULT                       00002511 RG    04
BOO$AB_LOADBUF                      00002200 R      04          BOO$GT_DXNAME                        000024E7 RG    04
BOO$AB_PATCH                        00000000 RG     04          BOO$GT_FILE                          00002519 RG    04
BOO$AB_PRMBUF                       00000200 RG     04          BOO$GT_OPNAME                        000024DA RG    04
BOO$AL_ACF                          0000248C RG     04          BOO$GT_PROMPT                        00002480 RG    04
BOO$AL_CLIBLK                       00002464 RG     04          BOO$GT_SAVE_DEVNAME                  00002614 R     04
BOO$CONADP                          0000011F RG     06          BOO$HILIM                            00000000 RG    03
BOO$CONAUNIT                        00000172 RG     06          BOO$LOAD                             000004D8 RG    06
BOO$CONCNUM                         00000167 RG     06          BOO$LOCK_GEN                         ********   X   05
BOO$CONCREG                         0000014A RG     06          BOO$LOLIM                            00000000 RG    02
BOO$CONCSROFFSET                    0000013F RG     06          BOO$MAKLIST                          000000EE RG    06
BOO$CONCVEC                         00000157 RG     06          BOO$MSCP_ARG                         000000DB RG    06
BOO$CONDRVNAM                       0000017D RG     06          BOO$MSCP_RESET                       0000009A RG    06
BOO$CONFIGALL                       00000000 RG     05          BOO$READFILE                         ********   X   06
BOO$CONFIGONE                       0000006F RG     05          BOO$RELOAD                           000004DE RG    06
BOO$CONNECT                         000002AB RG     06          BOO$RESETLIST                        00000062 RG    06
BOO$CONNLADP                        0000012A RG     06          BOO$UFOOPEN                          ********   X   06
BOO$CONRESET                        00000000 RG     06          BOO$UNLOCK_GEN                       ********   X   05
BOO$CONSOLE                         00000264 RG     06          BOO$USEACT                           000027D0 RG    04
BOO$CONSYSID_HIGH                   00000259 RG     06          BOOCMD$V_AUTOLOG                   = 0000000C
BOO$CONSYSID_LOW                    0000024E RG     06          BOOCMD$V_EXCLUDE                   = 00000007
BOO$CONUNITS                        00000243 RG     06          BOOCMD$V_SFLECT                    = 00000006
BOO$CONVECOFFSET                    00000134 RG     06          CLI$B_RQTYPE                       = 00000000
```

```
CLI$C_REQDESC                 = 0000001C                   OPERGETJPI                     00002694 R      04
CLI$K_GETCMD                  = 00000001                   OPERMSG                        000026C0 R      04
CLI$W_RQSIZE                  = 00000008                   OPERMSGBUF                     000026C8 R      04
CONFIGADP                       00000103 RG      05        OPERMSGFAO                     000026AC R      04
CONFIGSW                      = 00000001                   OPERMSGID                      000026A8 R      04
CONFIG_EXIT                     0000005D R       05        OPERMSGNAM                     000026B4 R      04
CONNECT                         000003AB RG      06        OPERMSGPID                     000026B0 R      04
CONNLADP                        00000379 RG      06        OPERMSGTXT                     000026D0 R      04
CONN_ADAP                       00000336 RG      06        OPERMSGVEC                     000026A4 R      04
CONSNAME                        000024D6 R       04        OPERNAMDESC                    000026B8 R      04
CR                            = 0000000D                   OUTBUF                         00002628 R      04
CRB$L_INTD                    = 00000024                   OUTBUF_STR                     00002630 R      04
CTL$GL_PCB                      ********   X     05        OUTLEN                         00002610 R      04
CTRSTR_AUTOLOG                  000025F1 R       04        OUTLEN_UNIT                    0000260C R      04
CTRSTR_AUTOLOG_UNIT             00002600 R       04        PR$_IPL                      = 00000012
DDB$L_UCB                     = 00000004                   PUTERROR                       ********   X    05
DDB$T_DRVNAME                 = 00000024                   RIO$AB_BUFFER                  ********   X    05
DYN$C_LOADCODE                = 00000062                   RIO$GW_OUTLEN                  ********   X    05
EXE$A_SYSPARAM                  ********   X     04        RIO$OUTPUT_LINE                ********   X    05
EXE$C_SYSPARSZ                  ********   X     04        SAVE_DOT                       00002584 R      04
EXE$GL_DEFFLAGS                 ********   X     05        SB$B_SYSTEMID                = 00000018
EXE$GL_SCB                      ********   X     06        SCH$IOLOCKR                    ********   X    05
EXE$LINK_VEC                    ********   X     06        SCH$IOUNLOCK                   ********   X    05
EXE$LOAD_CODE                   ********   X     06        SCS$GA_LOCALSB                 ********   X    05
EXE$V_NOAUTOCNF                 ********   X     05        SCS$GL_CDL                     ********   X    06
FACNAME                         000024D0 R       04        SCS$GL_MSCP                    ********   X    06
FACNAMED                        000024C8 RG      04        SELECT                         000001D1 R      05
FACNAMSZ                      = 00000006                   SGN$GET_DEVICE                 000002C2 RG     05
FF                            = 0000000C                   SGN$GET_DEVICE_LOCK_IODB       000002D4 R      05
FULL_NAME_PTR                   00002588 RG      04        SLV$A_SYSVECS                = 00000010
HELP_DESC                       00002578 R       04        SLV$B_TYPE                   = 0000000A
HELP_FILE                       00002559 R       04        SLV$L_INITRTN                = 00000004
HELP_FLAG                       00002574 R       04        SS$_DEVACTIVE                = 000002C4
HLP$M_PROMPT                  = 00000001                   SS$_DEVOFFLINE               = 00000084
IDB$L_ADP                     = 00000014                   SS$_NOPRIV                   = 00000024
IOC$AUTOCONFIG                  ********   X     05        SS$_NOSUCHDEV                = 00000908
IOC$AUTORESET                   ********   X     05        SYS$CMEXEC                     ********  GX    05
IOC$GL_ADPLIST                  ********   X     05        SYS$CMKRNL                     ********  GX    05
IOC$SEARCHALL                   ********   X     05        SYS$FAO                        ********   X    05
IOGEN$CONSOLE                   ********   X     06        SYSG$_CONFQUAL               = 007C808A
IOGEN$LOADER                    ********   X     05        SYSG$_INVADAP                = 007C80BA
JPI$_PID                      = 00000319                   SYSG$_NOADAPTER              = 007C80D2
LBR$OUTPUT_HELP                 ********   X     06        SYSG$_NOAUTOCNF              = 007C8002
LF                            = 0000000A                   SYSG$_NODEV                  = 007C9010
LIB$GET_INPUT                   ********   X     06        TPA$L_NUMBER                 = 0000001C
LIB$PUT_OUTPUT                  ********   X     06        TPA$L_PARAM                  = 00000020
LINK_CODE                       000005AE R       06        TPA$L_STRINGCNT              = 00000008
LOADCODE                        0000055D R       06        TPA$L_STRINGPTR              = 0000000C
LOADRV                          000004E3 R       06        TPA$L_TOKENCNT               = 00000010
LOCADP                          000000DA R       05        TPA$L_TOKENPTR               = 00000014
MMG$A_SYSPARAM                  ********   X     04        UBA_IOBASE                   = 00001000
MSCP_ARG_LIST                   0000258C R       04        UCB$L_CRB                    = 00000024
MSCP_ARG_LIST_SIZE            = 00000030                   UCB$L_LINK                   = 00000030
MSCP_NAME                       000025EC R       04        UCB$W_UNIT                   = 00000054
NEXTADP                         000000B2 R       05        VALID_PAR_FILE                 00002580 R      04
OPC$M_NM_CENTRL               = 00000001                   VEC$L_IDB                    = 00000008
OPC$_RQ_RQST                  = 00000003
```

```
                                                +-----------------+
                                                ! 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
$$$$000                         00000000  (    0.)  02 (  2.)  NOPIC    USR    CON    REL    LCL NOSHR NOEXE   RD   NOWRT NOVEC BYTE
     ZZZ                        00000000  (    0.)  03 (  3.)  NOPIC    USR    CON    REL    LCL NOSHR   EXE   RD     WRT NOVEC PAGE
NONPAGED_DATA                   000027EB  (10219.)  04 (  4.)  NOPIC    USR    CON    REL    LCL NOSHR NOEXE   RD     WRT NOVEC QUAD
NONPAGED_CODE                   0000039A  (  922.)  05 (  5.)  NOPIC    USR    CON    REL    LCL NOSHR   EXE   RD   NOWRT NOVEC LONG
PAGED_CODE                      00000612  ( 1554.)  06 (  6.)  NOPIC    USR    CON    REL    LCL NOSHR   EXE   RD   NOWRT NOVEC LONG

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

Phase                    Page faults    CPU Time       Elapsed Time
-----                    -----------    --------       ------------
Initialization                   29     00:00:00.07    00:00:00.51
Command processing              110     00:00:00.77    00:00:02.09
Pass 1                          571     00:00:23.37    00.00:47.02
Symbol table sort                 0     00:00:03.60    00:00:07.09
Pass 2                          276     00:00:05.53    00:00:09.68
Symbol table output              27     00:00:00.20    00:00:00.26
Psect synopsis output             2     00:00:00.04    00:00:00.07
Cross-reference output            0     00:00:00.00    00:00:00.00
Assembler run totals           1017     00:00:33.58    00:01:06.72

The working set limit was 1950 pages.
132448 bytes (259 pages) of virtual memory were used to buffer the intermediate code.
There were 130 pages of symbol table space allocated to hold 2263 non-local and 83 local symbols.
1453 sour e lines were read in Pass 1, producing 119 object records in Pass 2.
43 pages of virtual memory were used to define 40 macros.

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

Macro library name                         Macros defined
------------------                         --------------
_$255$DUA28:[BOOTS.OBJ]BOOTS.MLB;1                1
_$255$DUA28:[SYS.OBJ]LIB.MLB;1                   16
_$255$DUA28:[SYSLIB]STARLET.MLB;2                20
TOTALS (all libraries)                           37

2358 GETS were required to define 37 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:CONFIGUTL/OBJ=OBJ$:CONFIGUTL MSRC$:CONFIGSW/UPDATE=(ENH$:CONFIGSW)+MSRC$:SYSGEN/UPDATE=(ENH$:SYSGEN)+EXECML$/LIB+LIB$
```

0038 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

CVBTDRIVR
LIS

DDBTDRIVR
LIS

DMBTDRIVR
LIS

DQBTDRIVR
LIS

CONIO
LIS

LOADERSUB
LIS

CONFIGUTL
LIS

DLBTDRIVR
LIS

DXBTDRIVR
LIS

INITPGFIL
LIS

LOCKDATA
LIS

LOADDRIV
LIS

LOADER
LIS