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

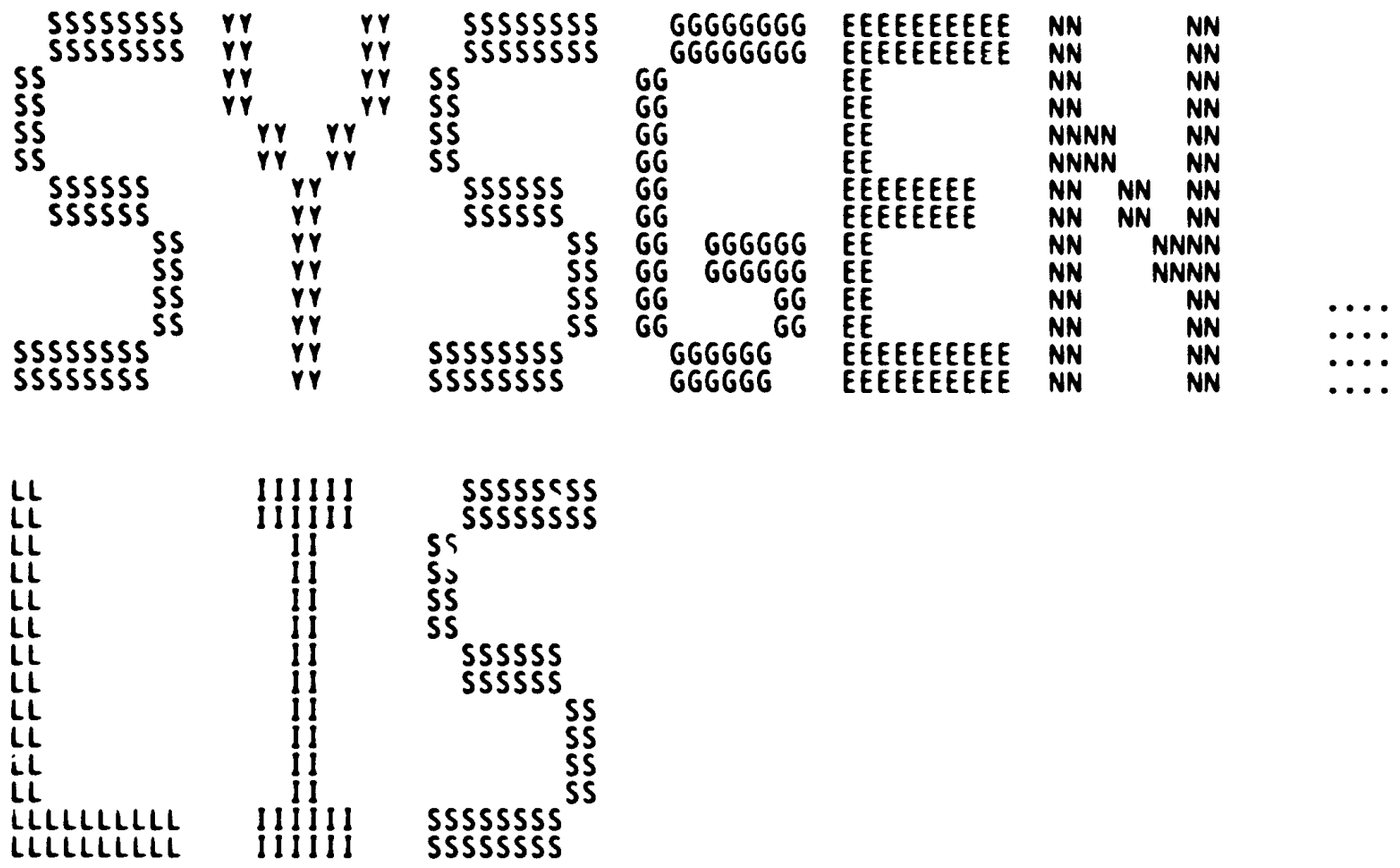
SSSSSSSS YY YY SSSSSSSS GGGGGGGG EEEEEEEEEE NN NN
SSSSSSSS YY YY SSSSSSSS GGGGGGGG EEEEEEEEEE NN NN
SS YY YY SS GG EE NN NN
SS YY YY SS GG EE NN NN
SS YY YY SS GG EE NNNN NN
SS YY YY SS GG EE NNNN NN
SSSSSS YY SSSSSS GG EEEEEEEE NN NN NN
SSSSSS YY SSSSSS GG EEEEEEEE NN NN NN
SS YY SS GG GGGGGG EE NN NNNN
SS YY SS GG GGGGGG EE NN NNNN
SS YY SS GG GG EE NN NN ....
SS YY SS GG GG EE NN NN ....
SSSSSSSS YY SSSSSSSS GGGGGG EEEEEEEEEE NN NN ....
SSSSSSSS YY SSSSSSSS GGGGGG EEEEEEEEEE NN NN ....
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
LLLLLLLLLL IIIIII SSSSSSSS
LLLLLLLLLL IIIIII SSSSSSSS

Table of contents

```
0000     1          .IF     NDF,CONFIGSW
0000     2          .TITLE  SYSGEN - SYSGEN UTILITY AND PARAMETER FILE EDITOR
0000     3          .IFF
0000     4          .TITLE  CONFIGUTL - SYSGEN UTILITIES FOR CONFIGURE PROCESS
0000     5          .ENDC
0000     6          .IDENT  'V04-002'
0000     7  ;
0000     8  ;*****************************************************************************
0000     9  ;*                                                                           *
0000    10  ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
0000    11  ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                    *
0000    12  ;*  ALL RIGHTS RESERVED.                                                     *
0000    13  ;*                                                                           *
0000    14  ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED     *
0000    15  ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE    *
0000    16  ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER     *
0000    17  ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY     *
0000    18  ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY     *
0000    19  ;*  TRANSFERRED.                                                             *
0000    20  ;*                                                                           *
0000    21  ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE     *
0000    22  ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT    *
0000    23  ;*  CORPORATION.                                                             *
0000    24  ;*                                                                           *
0000    25  ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS     *
0000    26  ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                   *
0000    27  ;*                                                                           *
0000    28  ;*                                                                           *
0000    29  ;*****************************************************************************
0000    30
0000    31  ;++
0000    32  ;
0000    33  ; Facility:  System generation and initialization
0000    34  ;
0000    35  ; Abstract: SYSGEN is the main routine to provide all SYSBOOT parameter
0000    36  ;       alteration commands in an online environment.
0000    37  ;
0000    38  ; Environment:
0000    39  ;
0000    40  ; Author:  RICHARD I. HUSTVEDT, Creation date:  4-MAY-1978
0000    41  ;
0000    42  ; MODIFIED BY:
0000    43  ;
0000    44  ;       V04-002 WHM0011         Bill Matthews           14-Sep-1984
0000    45  ;               Changed the defaults for the MSCP command.
0000    46  ;
0000    47  ;       V04-C01 WHM0010         Bill Matthews           04-Sep-1984
0000    48  ;               Changed IO PRIORITY default for the MSCP command and
0000    49  ;               disallow loading of the MSCP server multiple times.
0000    50  ;
0000    51  ;       V03-023 WHM0009         Bill Matthews           23-Jul-1984
0000    52  ;               Changed defaults for the MSCP command.
0000    53  ;
0000    54  ;       V03-022 WHM0008         Bill Matthews           20-Apr-1984
0000    55  ;               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    113 ;       V03-005 MSH0004         Maryann Hinden          13-May-1983
0000    114 ;               Change some BSBW PUTERROR instructions to JSB instead.
```

```
0000    115 ;
0000    116 ;       V03-004 MSH0003         Maryann Hinden          31-Jan-1983
0000    117 ;               Add support for cluster device names.
0000    118 ;
0000    119 ;       V03-003 TCM0001         Trudy C. Matthews       8-Nov-1982
0000    120 ;               Use new ADP$L_AVECTOR field in calculation of ACF$W_AVECTOR,
0000    121 ;               instead of calculating it from the adapter's TR number.
0000    122 ;
0000    123 ;       V03-002 MSH0002         Maryann Hinden          22-Oct-1982
0000    124 ;               Fix broken BSBW.
0000    125 ;
0000    126 ;       V03-001 MSH0001         Maryann Hinden          30-Sep-1982
0000    127 ;               Check for DDB$L_UCB = 0.
0000    128 ;--
0000    129
0000    130 ;
0000    131 ; Include files:
0000    132 ;
0000    133         $ACFDEF                         ; Define autoconfiguration block
0000    134         $ADPDEF                         ; Define adapter control block
0000    135         $BOOCMDDEF                      ; Define SYSGEN command options
0000    136         $CLIDEF                         ; Define CLI codes and values
0000    137         $CRBDEF                         ; Define CRB offsets
0000    138         $DDBDEF                         ; Define DDB offsets
0000    139         $DYNDEF                         ; Block types
0000    140         $HLPDEF                         ; Define HELP symbols
0000    141         $IDBDEF                         ; Define IDB offsets
0000    142         $IHDDEF                         ; Image header offsets
0000    143         $IPLDEF                         ; Define IPLs
0000    144         $JPIDEF                         ; $GETJPI definitions
0000    145         $LBRDEF                         ; Librarian symbols
0000    146         $OPCDEF                         ; Operator message definitions
0000    147         $PRDEF                          ; Define processor registers
0000    148         $PRMDEF                         ; Parameter descriptor definitions
0000    149         $SBDEF                          ; SCS system block definitions
0000    150         $SHRDEF                         ; Error codes
0000    151         $SLVDEF                         ; Loadable code header
0000    152         $SSDEF                          ; Define system status values
0000    153         $SYSGMSGDEF                     ; Sysgen messages
0000    154         $TPADEF                         ; TPARSE definitions
0000    155         $UCBDEF                         ; Define UCB offsets
0000    156         $VECDEF                         ; Define VEC offsets
0000    157
0000    158 ;
0000    159 ; Equated Symbols:
0000    160 ;
0000000D  0000  161         CR=13                           ; Character value for carriage return
0000000C  0000  162         FF=12                           ; Character value for form feed
0000000A  0000  163         LF=10                           ; Character value for line feed
00001000  0000  164         UBA_IOBASE=8*512                ; Offset from UBA configuration register
0000    165                                                 ; to base of I/O page
0000    166 ;
0000    167 ; Own Storage
0000    168 ;
00000000  169         .PSECT  $$$$000,NOEXE,NOWRT     ; PSECT to mark lower address
0000    170 BOO$LOLIM::                                 ; Marker definition
00000000  171         .PSECT  ____ZZZ,WRT,PAGE        ; PSECT to mark upper address limit
```

```
                     0000   172 BOO$HILIM::                                 ;
                 00000000   173         .PSECT  NONPAGED_DATA   rd,wrt,noexe,quad
                     0000   174
                     0000   175 BOO$AB_PATCH::                              ; Non-paged Patch area
                 00000200   176         .BLKB   512                         ; One page
                     0200   177 BOO$AB_PRMBUF::                            ; Parameter buffer
                 00002200   178         .BLKB   512*16                      ; A generous buffer
                     2200   179 BOO$AB_LOADBUF:                            ; Buffer for code loader
                 00002400   180         .BLKB   512
                     2400   181 ACF$GL_DDB::
                 00000000   182         .LONG   0
                     2404   183 ACF$GL_UCB::
                 00000000   184         .long   0
                     2408   185 ACF$GL_IDB::
                 00000000   186         .long   0
                     240C   187 ACF$GL_CRB::
                 00000000   188         .long   0
                     2410   189 ACF$GL_LASTDDB::
                 00000000   190         .long   0
                     2414   191 ACF$GL_DPT::
                 00000000   192         .long   0
                     2418   193 ACF$GL_SB::
                 00000000   194         .LONG   0
                     241C   195 BOO$GL_COMBO_VECTOR_OFFSET::                ; Offset to vector from start of combo
                 00000000   196         .LONG   0                           ; device's vectors
                     2420   197 BOO$GL_COMBO_CSR_OFFSET::                   ; Offset to CSR from start of combo
                 00000000   198         .LONG   0                           ; device's CSR
                     2424   199 BOO$GL_CONADP::                             ; Adapter TR number
                 FFFFFFFE   200         .LONG   -2                          ; Null value
                     2428   201 BOO$GL_CONCREG::                            ; Control register
                 FFFFFFFF   202         .LONG   -1                          ; Null value
                     242C   203 BOO$GL_CONCUNIT::                           ; Controller unit
                 FFFFFFFF   204         .LONG   -1                          ; Null value
                     2430   205 BOO$GL_CONNUMU::                           ; Number of Units to configure
                 00000001   206         .LONG   1                           ; Default value is 1 unit
                     2434   207 BOO$GL_CONVECT::                           ; Vector offset
                 FFFFFFFF   208         .LONG   -1                          ; Null value
                     2438   209 BOO$GL_CONNUMV::                           ; Number of vectors
                 FFFFFFFF   210         .LONG   -1                          ; Null value
                     243C   211 BOO$GL_CONAUNIT::                           ; Adapter unit
                 FFFFFFFF   212         .LONG   -1                          ; Null value
                     2440   213 BOO$GL_CONDEV::                             ; Device name string address
                 FFFFFFFF   214         .LONG   -1                          ; Null value
                     2444   215 BOO$GL_CONDRV::                             ; Driver name string address
                 FFFFFFFF   216         .LONG   -1                          ; Null value
                     2448   217 BOO$GL_CONUNITS::                           ; Maximum units
                 00000000   218         .LONG   0                           ;
                     244C   219 BOO$GQ_CONSYSID::                          ; System ID
                 00000000   220         .LONG   0                           ; quadword
                 00000000   221         .LONG   0                           ;
                     2454   222 BOO$GL_CONCRB::                            ; CRB address
                 00000000   223         .LONG   0                           ;
                     2458   224 BOO$GL_CONFLAGS::                          ; Flags
                 00000000   225         .LONG   0                           ;
                     245C   226 BOO$GL_NEXTSTR::                           ; Next string location
                 00000000   227         .LONG   0                           ;
                     2460   228 BOO$GL_SELECT::                            ; Address of select list
```

```
                        00000000  2460   229          .LONG   0                       ;
                                  2464   230 BOO$AL_CLIBLK::                          ; CLI call back block
                                  2464   231          $CLIREQDESC     -               ; Get command call back block
                                  2464   232                  RQTYPE=CLI$K_GETCMD    ;
                        0000246C  2480   233 BOO$GQ_CMDESC==BOO$AL_CLIBLK+CLI$W_RQSIZE      ; Command descriptor address
                                  2480   234 BOO$GT_PROMPT::                          ; Prompt string
00 20 20 3E 4E 45 47 53 59 53 0A 0D  2480   235          .ASCIZ  <CR><LF>%SYSGEN>  %    ;
                                  248C   236 BOO$AL_ACF::                            ; Auto-configuration block
                        000024B4  248C   237          .BLKB   ACF$C_LENGTH           ; Allocate space for it
                                  24B4   238 BOO$GQ_LIMITS::                         ; High and low address limits for lockdown
                        00000000' 24B4   239          .LONG   BOO$LOLIM              ; Lower address bound
                        FFFFFFFF' 24B8   240          .LONG   BOO$HILIM-1            ;
                                  24BC   241 BOO$GQ_RETADR::                         ; Return address receiver
                00000000 00000000 24BC   242          .LONG   0,0                    ;
                                  24C4   243 BOO$GL_RETSAVE::                        ; Saved co-routine return address
                        00000000  24C4   244          .LONG   0                      ;
                                  24C8   245 FACNAMED::                              ; Facility name descriptor
                000024D0'00000006' 24C8   246          .LONG   FACNAMSZ,FACNAME       ;
                4E 45 47 53 59 53 24D0   247 FACNAME:.ASCII  /SYSGEN/                ;
                        00000006  24D6   248 FACNAMSZ=.-FACNAME                      ; Length of facility name
                                  24D6   249 CONSNAME:                               ; Console block storage
                     41 53 43 00' 24D6   250          .ASCIC  /CSA/                  ; device name
                              03  24D6
                                  24DA   251 BOO$GT_OPNAME::                         ; Console terminal device name
             4        41 50 4F 00' 24DA   252          .ASCIC  /OPA/                  ;
                              03  24DA
                                  24DE   253 BOO$GT_CVNAME::                         ; Name of RL02 driver
          52 45 56 49 52 44 56 43 00' 24DE   254          .ASCIC  /CVDRIVER/
                              08  24DE
                                  24E7   255 BOO$GT_DXNAME::                         ; Name of floppy driver
          52 45 56 49 52 44 58 44 00' 24E7   256          .ASCIC  /DXDRIVER/
                              08  24E7
                                  24F0   257 BOO$GT_DDNAME::                         ; Name of TU58 driver
          52 45 56 49 52 44 44 44 00' 24F0   258          .ASCIC  /DDDRIVER/
                              08  24F0
                                  24F9   259
                                  24F9   260 BOO$GL_FILEADDR::                       ; File spec address
                        00000000  24F9   261          .LONG   0                      ;
                                  24FD   262 BOO$GB_FILELEN::                        ; File spec length
                              00  24FD   263          .BYTE   0                      ;
                                  24FE   264
                        00000000  24FE   265 BOO$GL_PARINUSE::       .LONG   0
             74 6E 65 72 72 75 43 00' 2502   266 BOO$GT_CURRENT::       .ASCIC  /Current/
                              07  2502
                65 76 69 74 63 41 00' 250A   267 BOO$GT_ACTIVE::        .ASCIC  /Active/
                              06  250A
             74 6C 75 61 66 65 44 00' 2511   268 BOO$GT_DEFAULT::       .ASCIC  /Default/
                              07  2511
                        00002559  2519   269 BOO$G_FILE::             .BLKB   64
                                  2559   270
                                  2559   271 HELP_FILE:                             ; Help library file name
45 48 24 53 59 53 00002561'010E0000' 2559   272          .ASCID  /SYS$HELP:SYSGEN.HLB/
4C 48 2E 4E 45 47 53 59 53 3A 50 4C  2567
                              42  2573
                        00000001  2574   273 HELP_FLAG:      .long           hlp$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
                00002680'   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   329         .WORD   1                               ; FAO argument count
                    0000    26AE   330         .WORD   0                               ; No new message flags
```

```
                     26B0   331 OPERMSGPID:                                         ; PID of this process
00000000             26B0   332         .LONG   0
                     26B4   333 OPERMSGNAM:                                        ; File specification
000026B8'            26B4   334         .ADDRESS        OPERNAMDESC
                     26B8   335
                     26B8   336 OPERNAMDESC:
00000000 00000000    26B8   337         .LONG   0,0
                     26C0   338
                     26C0   339 OPERMSG:                                   ; Message descriptor
00000000             26C0   340         .LONG           0
000026C8'            26C4   341         .ADDRESS        OPERMSGBUF
                     26C8   342
                     26C8   343 OPERMSGBUF:                                ; Message buffer
00000103             26C8   344         .LONG   OPC$_RQ_RQST!<OPC$M_NM_CENTRL@8>        ; Message type and target
00000000             26CC   345         .LONG   0                                       ; No reply message
                     26D0   346 OPERMSGTXT:                                        ; Message text
000027D0             26D0   347         .BLKB   256
                     27D0   348
                     27D0   349         .IF     NDF,CONFIGSW                 ; SYSGEN-specific code
```

```
                     27D0    351          .SBTTL  BOO$USEFILE - Use parameter file
                     27D0    352 ;++
                     27D0    353 ; Functional description:
                     27D0    354 ;       BOO$USEFILE reads the specified file in response to the USE
                     27D0    355 ;       command and merges all of the values specified in that file into
                     27D0    356 ;       the working copy of the parameter values.  This is accomplished
                     27D0    357 ;       by looking up each value specified and merging the associated
                     27D0    358 ;       value.
                     27D0    359 ;
                     27D0    360 ; Calling sequence:
                     27D0    361 ;       CALLG   arglist,BOO$USEFILE
                     27D0    362 ;
                     27D0    363 ; Input Parameters:
                     27D0    364 ;       TPA$L_TOKENCNT(AP) - Length of file name string
                     27D0    365 ;       TPA$L_TOKENPTR(AP) - Address fo file name string
                     27D0    366 ; Output Parameters:
                     27D0    367 ;       R0 - Completion status code
                     27D0    368 ;
                     27D0    369 ;--
                     27D0    370
                00000000     371 .PSECT  PAGED_CODE      rd,nowrt,exe,long
                     0000    372
               03FC  0000    373 .Entry  BOO$USEFILE, ^M<R2,R3,R4,R5,R6,R7,R8,R9>        ; Entry mask
                     0002    374
                     0002    375
   00000000'8F   E2  0002    376          BBSS    #EXE$V_WRITESYSPARAMS,- ; Use a file => write current needed
00 00000000'GF       0008    377                  G^EXE$GL_DYNAMIC_FLAGS,1$;
                     000E    378 1$:
00002584'EF   00000000'EF  D0  000E    379          MOVL    BOO$GL_DOT,L^SAVE_DOT   ; Save dot
         57   10 AC    9E  0019    380          MOVAB   TPA$L_TOKENCNT(AP),R7   ; Set address of file name descriptor
              FFE0'     30  001D    381          BSBW    BOO$FILOPEN             ; Open specified file
           04 50        E8  0020    382          BLBS    R0,20$                  ; Continue if success
        50    01        3C  0023    383 10$:     MOVZWL  #1,R0                   ; Force success
                        04  0026    384          RET                             ;
      56   00000200'EF   9E  0027    385 20$:     MOVAB   BOO$AB_PRMBUF,R6        ; Set address of parameter buffer
           59   10       D0  002E    386          MOVL    #16,R9                  ; Set size of buffer
              FFCC'     30  0031    387          BSBW    BOO$READFILE            ; Read file content into parameter buffer
           EC 50        E9  0034    388          BLBC    R0,10$                  ; Exit if error
      58   00000200'EF   9E  0037    389          MOVAB   BOO$AB_PRMBUF,R8        ; Init pointer to parameter buffer
00000000'EF   68   20     28  003E    390          MOVC3   #32,(R8),EXE$GT_STARTUP ; Set startup command file name
           58   20       C0  0046    391          ADDL    #32,R8                  ;  and advance buffer pointer
      00002580'EF   D4    0049    392          CLRL    VALID_PAR_FILE          ; Initialize valid parameter file flag
                   68     D5  004F    393 30$:     TSTL    (R8)                    ; Check for end of list
                   56     13  0051    394          BEQL    DONE                    ; Branch if yes
        10 AC      68     9A  0053    395          MOVZBL  (R8),TPA$L_TOKENCNT(AP) ; Set token count for search
     14 AC    01   A8     9E  0057    396          MOVAB   1(R8),TPA$L_TOKENPTR(AP); And address of string
           58   10       C0  005C    397          ADDL    #16,R8                  ; Advance to value
        1C AC      88     D0  005F    398          MOVL    (R8)+,TPA$L_NUMBER(AP)  ; Set number
      00000000'EF   6C     FA  0063    399          CALLG   (AP),L^BOO$SEARCH       ; Search for parameter
           E2 50        E9  006A    400          BLBC    R0,30$                  ; Next parameter if not found
      00002580'EF   01     D0  006D    401          MOVL    #1,VALID_PAR_FILE       ; Indicate valid parameter file
           54   20 AC     D0  0074    402          MOVL    TPA$L_PARAM(AP),R4      ; Get a pointer to the parameter descripttor
     22 10 A4   10        E1  0078    403          BBC     #PRM$V_ASCII,PRM$L_FLAGS(R4),40$; Branch if not an ascii parameter
           14 AC   78     DE  007D    404          MOVAL   -(R8),TPA$L_TOKENPTR(AP); Get a pointer to the parameter value
           50   14 A4     9A  0081    405          MOVZBL  PRM$B_SIZE(R4),R0       ; Get parameter size in bits
   50      50   FD 8F     78  0085    406          ASHL    #-3,R0,R0               ; Set parameter size
           10 AC   50     9A  008A    407          MOVZBL  R0,TPA$L_TOKENCNT(AP)   ;
```

```
                50   03   C0  008E   408          ADDL2   #3,R0                   ; Round size up to the next longword
                50   03   CA  0091   409          BICL2   #3,R0                   ;
                58   50   C0  0094   410          ADDL2   R0,R8                   ; Advance past value
          0000'CF    6C   FA  0097   411          CALLG   (AP),W^BOO$SETASCII     ; Set the value of the parameter
                FFB0   31   009C   412          BRW     30$                     ; Continue with the next parameter
    00000000'EF    6C   FA  009F   413 40$:     CALLG   (AP),L^BOO$SETVALUE     ; Set value of parameter
                FFA6   31   00A6   414          BRW     30$                     ; Continue with next parameter
                FF54'  30   00A9   415 DONE:    BSBW    BOO$FILCLOSE            ; Close the file
      09 00002580'EF   E8   00AC   416          BLBS    VALID_PAR_FILE,10$      ; If LBS, valid parameter file
    50   007C80EA 8F   D0   00B3   417          MOVL    #SYSG$_NOTPARAM,R0      ; Set error
                21   11   00BA   418          BRB     20$                     ; Branch
                          00BC   419 10$:
                          00BC   420 ;
                          00BC   421 ; Set file name in BOO$GL_PARINUSE
                          00BC   422 ;
    58   00002519'EF   DE   00BC   423          MOVAL   BOO$GT_FILE,R8          ; Set address of String
    000024FE'EF   58   D0   00C3   424          MOVL    R8,BOO$GL_PARINUSE      ; Set address
    68   000024FD'EF   9A   00CA   425          MOVZBL  BOO$GB_FILELEN,(R8)     ; Set count
    000024F9'FF   68   28   00D1   426          MOVC3   (R8),@BOO$GL_FILEADDR,-
                01 A8          00D8   427                  1(R8)               ; Move string
                          00DA   428
                50   01   3C   00DA   429          MOVZWL  #SS$_NORMAL,R0          ; Return success
00000000'EF   00002584'EF   D0   00DD   430 20$:     MOVL    L^SAVE_DOT,BOO$GL_DOT   ; Restore dot
                04   00E8   431          RET                             ;
                          00E9   432          .ENDC                           ; End of SYSGEN-specific code
```

```
                                00E9   434          .SBTTL  BOO$USEACT - Use active parameters
                                00E9   435 ;++
                                00E9   436 ; Functional description:
                                00E9   437 ;       This routine copies the parameter values from the running
                                00E9   438 ;       system to the working copy of the parameter values.
                                00E9   439 ; Calling sequence:
                                00E9   440 ;
                                00E9   441 ;       CALLS   #0,BOO$USEACT
                                00E9   442 ;
                                00E9   443 ; Input parameters:
                                00E9   444 ;       None
                                00E9   445 ; Output Parameters:
                                00E9   446 ;       R0 - Completion status code
                                00E9   447 ;--
                                00E9   448
                         003C   00E9   449 .Entry  BOO$USEACT,^M<R2,R3,R4,R5>
                                00EB   450
                    0000'8F     28 00EB   451          MOVC3   #EXE$C_SYSPARSZ,-       ; Move parameters
00000000'EF     00000000'EF            00EF   452                  MMG$A_SYSPARAM,EXE$A_SYSPARAM
                0000250A'EF     DE 00F9   453          MOVAL   BOO$GL_ACTIVE,-
                000024FE'EF            00FF   454                  BOO$GL_PARINUSE         ; Set parameter in use
                    50    01    D0 0104   455          MOVL    #1,R0                   ; Return success
                                04 0107   456          RET
                                0108   457          .IF     NDF,CONFIGSW                    ; SYSGEN-specific code
```

```
                     0108   459          .SBTTL  BOO$WRTACT - Write parameters to system
                     0108   460 ;++
                     0108   461 ; Functional Description:
                     0108   462 ;       This routine writes the parameters in the working parameter
                     0108   463 ;       buffer to the system's parameter area.  Only dynamic
                     0108   464 ;       parameters are copied.
                     0108   465 ;
                     0108   466 ; Calling Sequence:
                     0108   467 ;       CALLS   #0,BOO$WRTACT
                     0108   468 ;
                     0108   469 ; Input Parameters:
                     0108   470 ;       None
                     0108   471 ;
                     0108   472 ; Output Parameters:
                     0108   473 ;       R0 - Completion status code
                     0108   474 ;--
                     0108   475
                00000000    476 .PSECT  NONPAGED_CODE   rd,nowrt,exe,long
                     0000   477
                0000 0000   478 .Entry  BOO$WRTACT, ^M<>
                     0002   479
                     0002   480          $CMKRNL_S       B^10$,(AP)      ; Do it in kernel mode
          0D 50      E9 000E   481          BLBC    R0,1$                   ; If LBC, error
00000140'EF          16 0011   482          JSB     BOO$SENDOPER            ; Notify operator of WRITE ACTIVE
        007CA013     0017   483                  .LONG   SYSG$_WRITEACT
          09 50      E8 001B   484          BLBS    R0,5$                   ; If LBS, success
00000000'EF          16 001E   485 1$:       JSB     PUTERROR                ; Report error
      50   01      D0 0024   486          MOVL    #1,R0                   ; Force success
                   04 0027   487 5$:       RET
                     0028   488
                 003C 0028   489 10$:      .WORD   ^M<R2,R3,R4,R5>
      55   00000000'EF    9E 002A   490          MOVAB   L^BOO$A_PRMBLK,R5       ; Get base of parameter blocks
                     0031   491          DSBINT  #IPL$_SCHED             ; Raise IPL to prevent being unscheduled
                     0037   492                                          ; (Assumes pages are locked in W.S.)
                     0037   493
                     0037   494          ASSUME  PRM$L_ADDR  EQ  0
                     0037   495
           53    65    D0 0037   496 20$:      MOVL    PRM$L_ADDR(R5),R3       ; Get address of parameter
                 28    13 003A   497          BEQL    40$                     ; Reached the end
                 00    E1 003C   498          BBC     #PRM$V_DYNAMIC,-        ; Branch if this is not a
                    1E 10 A5   003E   499                  PRM$L_FLAGS(R5),30$     ; dynamic parameter
            51    15 A5    9A 0041   500          MOVZBL  PRM$B_POS(R5),R1        ; Get position of parameter
52   63    14 A5    51    EF 0045   501          EXTZV   R1,PRM$B_SIZE(R5),(R3),R2  ; Extract parameter value
      50   00000000'EF    9E 004B   502          MOVAB   L^EXE$A_SYSPARAM,R0     ; Get address of working buffer
           53    50    C2 0052   503          SUBL    R0,R3                   ; Get parameter offset
       14 A5    51    52    F0 0055   504          INSV    R2,R1,PRM$B_SIZE(R5),-  ; Store in system
           00000000'E3         005A   505                  L^MMG$A_SYSPARAM(R3)
                     005F   506
           55    32    C0 005F   507 30$:      ADDL    #PRM$C_LENGTH,R5        ; Point to next paramter block
                 D3    11 0062   508          BRB     20$                     ; Repeat
                     0064   509
                     0064   510 ; Copy dynamic flags from default flags to R0
                     0064   511
           FFFFFFFF'8F    CB 0064   512 40$:      BICL3   #^C<PRM$M_DYNFLAGS>,-
      50   00000000'EF         006A   513                  MMG$A_SYSPARAM+<EXE$GL_DEFFLAGS-EXE$A_SYSPARAM>,R0
           00000000'8F    CA 0070   514          BICL    #PRM$M_DYNFLAGS,-       ; Clear dynamic flags in real flags
           00000000'EF         0076   515                  EXE$GL_FLAGS
```

```
                00000000'EF    50    C8  007B    516          BISL    R0,EXE$GL_FLAGS         ; Set dynamic flags in real flags
                                           0082    517
                                           0082    518          ENBINT                          ; Lower IPL
                          50    01    D0  0085    519          MOVL    #1,R0                   ; Set success
                                      04  0088    520          RET
```

```
                        0089    522          .SBTTL  BOO$WRTCUR - Write Current Parameters
                        0089    523 ;++
                        0089    524 ; Functional Description:
                        0089    525 ;       This routine writes the parameters from the working parameter
                        0089    526 ;       buffer to the system parameter file on disk.  They will take effect the
                        0089    527 ;       next time the system is booted.
                        0089    528 ;
                        0089    529 ; Calling Sequence:
                        0089    530 ;       CALLS   #0,BOO$WRTCUR
                        0089    531 ;
                        0089    532 ; Input parameters:
                        0089    533 ;       None
                        0089    534 ;
                        0089    535 ; Output Parameters:
                        0089    536 ;       R0 - Completion status code
                        0089    537 ;--
                        0089    538
                00000108    539 .PSECT  PAGED_CODE      rd,nowrt,exe,long
                        0108    540
                    03FC    0108    541 .Entry  BOO$WRTCUR, ^M<R2,R3,R4,R5,R6,R7,R8,R9>
                        010A    542
    00000000'8F     E5      010A    543          BBCC    #EXE$V_WRITESYSPARAMS,- ; Don't do WRITE CURRENT again in startup
 00 00000000'GF             0110    544                  G^EXE$GL_DYNAMIC_FLAGS,10$;
50  00000000'EF     9E      0116    545 10$:     MOVAB   BOO$GT_SYSPARNAME,R0    ; Get address of system .PAR file name
      10 AC    80     9A      011D    546          MOVZBL  (R0)+,TPA$L_TOKENCNT(AP); Set up for call to BOO$WRTSYSPARFILE
      14 AC    50     D0      0121    547          MOVL    R0,TPA$L_TOKENPTR(AP)   ;
00000000'GF    6C     FA      0125    548          CALLG   (AP),G^BOO$WRTSYSPARFILE; Call the routine to write out the file
           0A 50     E9      012C    549          BLBC    R0,20$                  ; Branch if error
            000E     30      012F    550          BSBW    BOO$SENDOPER            ; Notify operator of WRITE CURRENT
             007CA01B        0132    551                  .LONG   SYSG$_WRITECUR
          03 50     E8      0136    552          BLBS    R0,30$                  ; If LBS, success
           FEC4'    30      0139    553 20$:     BSBW    PUTERROR                ; Report error
        50    01     D0      013C    554 30$:     MOVL    #1,R0                   ; Return success
                     04      013F    555          RET
                        0140    556
```

```
                                0140   558          .SBTTL  BOO$SENDOPER - Output facility error message to operator
                                0140   559 ;
                                0140   560 ; Functional Description:
                                0140   561 ;       BOO$SENDOPER outputs an error message to the operator.
                                0140   562 ;
                                0140   563 ; Calling Sequence:
                                0140   564 ;       BSBW    BOO$SENDOPER
                                0140   565 ;               .LONG   <msg-id>
                                0140   566 ;
                                0140   567 BOO$SENDOPER::
000026A8'EF    00 BE    D0      0140   568          MOVL    @(SP),OPERMSGID                 ; Put message ID in vector
               6E   04    C0    0148   569          ADDL2   #4,(SP)                         ; Advance return address
                                014B   570          $GETJPI_S       ITMLST=OPERGETJPI       ; Get process ID
                 7B 50    E9    0162   571          BLBC    R0,10$                          ; If LBC, error
000026A4'EF      03       D0    0165   572          MOVL    #3,OPERMSGVEC                   ; Assume WRITE ACTIVE
000026AC'EF      01       D0    016C   573          MOVL    #1,OPERMSGFAO
          000026B4'EF       D4   0173   574          CLRL    OPERMSGNAM
000026A8'EF    007CA01B 8F      D1    0179   575          CMPL    #SYSG$_WRITECUR,OPERMSGID       ; WRITE CURRENT ?
                        2D      12    0184   576          BNEQ    5$                              ; If NEQ, no
          000026A4'EF       D6   0186   577          INCL    OPERMSGVEC                      ; Set up WRITE CURRENT
          000026AC'EF       D6   018C   578          INCL    OPERMSGFAO
000026B4'EF    000026B8'EF      9E    0192   579          MOVAB   OPERNAMDESC,OPERMSGNAM
000026B8'EF    00000000'EF      9A    0197   580          MOVZBL  RIO_INPNAM+NAM$B_RSL,OPERNAMDESC; Build descriptor
000026BC'EF    00000000'EF      D0    01A8   581          MOVL    RIO_INPNAM+NAM$L_RSA,OPERNAMDESC+4
                                01B3   582 5$:       $PUTMSG_S -                             ; Get and format message
                                01B3   583                       MSGVEC=OPERMSGVEC, -
                                01B3   584                       ACTRTN=666$
                 13 50    E9    01CA   585          BLBC    R0,10$                          ; If LBC, error
                                01CD   586          $SNDOPR_S       MSGBUF=OPERMSG
                 06 50    E8    01DD   587          BLBS    R0,20$                          ; If LBS, success
                   FE1D'  30    01E0   588 10$:      BSBW    PUTERROR                        ; Report error
               50    01    D0    01E3   589          MOVL    #1,R0                           ; Force success
                                01E6   590 20$:
                       05       01E6   591          RSB
                                01E7   592 666$:
                   003C         01E7   593          .WORD   ^M<R2,R3,R4,R5>
            50     04 BC    7D    01E9   594          MOVQ    @4(AP),R0                       ; Get string descriptor
000026C0'EF    50    08    C1    01ED   595          ADDL3   #OPC$L_MS_TEXT,R0,OPERMSG; Store total operator message size
000026D0'EF    61    50    28    01F5   596          MOVC3   R0,(R1),OPERMSGTXT              ; Copy text to operator message buffer
                     50    D4    01FD   597          CLRL    R0                              ; Prevent message output to SYS$OUTPUT
                     04          01FF   598          RET
                                0200   599
                                0200   600          .ENDC                                   ; End of SYSGEN-specific code
```

```
                                0200   602          .SBTTL  BOO$CONFIGALL - Auto-configure all adapters
                                0200   603 ;++
                                0200   604 ; Functional Description:
                                0200   605 ;       BOO$CONFIGALL is called to implement the "AUTOCONFIGURE ALL"
                                0200   606 ;       command. All standard devices supported by VAX/VMS will be
                                0200   607 ;       located and connected for use with any necessary drivers being
                                0200   608 ;       loaded.
                                0200   609 ;
                                0200   610 ; Calling Sequence:
                                0200   611 ;       CALLG   ARGLIST,BOO$CONFIGALL
                                0200   612 ;
                                0200   613 ; Output parameters:
                                0200   614 ;       R0 - Completion status code
                                0200   615 ;--
                                0200   616
                         00000089   617 .PSECT  NONPAGED_CODE   rd,nowrt,exe,long
                                0089   618
                           0FFC 0089   619 .Entry  BOO$CONFIGALL, ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>      ; Entry mask
                                008B   620
 08 00000000'EF   00000000'8F   E1   008B   621          BBC     #EXE$V_NOAUTOCNF,EXE$GL_DEFFLAGS,5$; do we allow auto configure
              50   007C8002 8F   D0   0097   622          MOVL    #SYSG$_NOAUTOCNF,R0     ;Give them a no autoconfigure error
                                04   009E   623          RET                             ; and return
                                009F   624
                      FF5E'     30   009F   625 5$:      BSBW    BOO$LOCK_GEN            ; Lock SYSGEN database
                    07 50       E8   00A2   626          BLBS    R0,7$                   ; If no error, continue
                00000000'EF     16   00A5   627          JSB     PUTERROR
                                04   00AB   628          RET
                                00AC   629
                      FF51'     30   00AC   630 7$:      BSBW    IOC$AUTORESET           ; Reset controller characters for device
                                00AF   631                                             ; names
                         5B     D4   00AF   632          CLRL    R11                     ; Indicate no ADP address yet
                         5B     DD   00B1   633 10$:     PUSHL   R11                     ; Set as argument
           0000013B'EF    01     FB   00B3   634          CALLS   #1,NEXTADP              ; Get next ADP address
                    29 50       E9   00BA   635          BLBC    R0,CONFIG_EXIT          ; Branch if error (NOPRIV)
                   5B   51      D0   00BD   636          MOVL    R1,R11                  ; Check return status
                        10      18   00C0   637          BGEQ    20$                     ; Branch if done
                         5B     DD   00C2   638          PUSHL   R11                     ; Set as ADP argument
                018C'CF    01     FB   00C4   639          CALLS   #1,W^CONFIGADP          ; Configure the entire adapter
                    E5 50       E8   00C9   640          BLBS    R0,10$                  ; Continue if no error
                00000000'EF     16   00CC   641          JSB     PUTERROR                ; Report error
                   50   01      D0   00D2   642 20$:     MOVL    #1,R0                   ; Set success
                                00D5   643
 09 00000000'EF    0C     E1   00D5   644          BBC     #BOOCMD$V_AUTOLOG,L^BOO$GL_CMDOPT,CONFIG_EXIT ; Branch if not /LOG
                00002614'EF     D4   00DD   645          CLRL    BOO$GT_SAVE_DEVNAME     ; Clear name
                      01A9      30   00E3   646          BSBW    AUTOLOG                 ; Output last line if there is one
                                00E6   647
                                00E6   648 CONFIG_EXIT:
                         50     DD   00E6   649          PUSHL   R0                      ; Save status
                      FF15'     30   00E8   650          BSBW    BOO$UNLOCK_GEN          ; Unlock SYSGEN database
                    06 50       E8   00EB   651          BLBS    R0,35$                  ; If no error, continue
                00000000'EF     16   00EE   652          JSB     PUTERROR                ; Give error message
                   50 8ED0      00F4   653 35$:     POPL    R0                      ; Restore status
                        04      00F7   654          RET                             ;
                                00F8   655
                           0FFC 00F8   656 .Entry  BOO$CONFIGONE, ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>      ; Entry mask
                                00FA   657
                      FF03'     30   00FA   658          BSBW    BOO$LOCK_GEN            ; Lock SYSGEN database
```

```
                07 50   E8  00FD   659          BLBS    R0,5$                   ; If no error, continue
       00000000'EF      16  0100   660          JSB     PUTERROR                ; Give error message
                        04  0106   661          RET                             ;
                            0107   662
                FEF6'   30  0107   663 5$:      BSBW    IOC$AUTORESET           ; Reset controller characters for device
                            010A   664                                          ; device names
                1C AC   DD  010A   665          PUSHL   TPA$L_NUMBER(AP)        ; Set TR number of adapter
             63'AF  01  FB  010D   666          CALLS   #1,B^LOCADP             ; Locate adapter control block
                D2 50   E9  0111   667          BLBC    R0,CONFIG_EXIT          ; Branch if error (NOPRIV)
                   51   DD  0114   668          PUSHL   R1                      ; Set as argument to CONFIGADP
                   0D   13  0116   669          BEQL    10$                     ; Done if no adapter
             8C'AF  01  FB  0118   670          CALLS   #1,B^CONFIGADP          ; Configure adapter
                06 50   E8  011C   671          BLBS    R0,10$                  ; Continue if no error
       00000000'EF      16  011F   672          JSB     PUTERROR                ; Give error status
                50  01   D0  0125   673 10$:     MOVL    #1,R0                   ; Set success for parse
B6 00000000'EF      0C   E1  0128   674          BBC     #BOOCMD$V_AUTOLOG,L^BOO$GL_CMDOPT,CONFIG_EXIT ; Branch if not /LOG
       00002614'EF      D4  0130   675          CLRL    BOO$GT_SAVE_DEVNAME     ; Clear name
                0156    30  0136   676          BSBW    AUTOLOG                 ; Output last line if there is one
                   AB   11  0139   677 20$:     BRB     CONFIG_EXIT             ;
                            013B   678
                            013B   679 NEXTADP:                                 ; Return next ADP address in R0
                    0000    013B   680          .WORD   0                       ; Null entry mask
                            013D   681          $CMEXEC_S       B^10$,(AP)      ; Call real routine in exec mode
                        04  0149   682          RET                             ;
                            014A   683
                    0000    014A   684 10$:     .WORD   0                       ; Null entry mask
             51  04 AC   D0  014C   685          MOVL    4(AP),R1                ; Get current address
                   06   13  0150   686          BEQL    20$                     ; 0 => start of list
             51  04 A1   D0  0152   687          MOVL    ADP$L_LINK(R1),R1       ; Flink onward
                   07   11  0156   688          BRB     30$
          51 00000000'EF   D0  0158   689 20$:     MOVL    IOC$GL_ADPLIST,R1       ; Return head of list
                50  01   D0  015F   690 30$:     MOVL    #1,R0
                        04  0162   691          RET                             ;
                            0163   692
                            0163   693 LOCADP:                                  ; Return address of ADP for TR number
                    0000    0163   694          .WORD   0                       ;
                            0165   695          $CMEXEC_S       B^5$,(AP)       ; Call routine in exec mode
                        04  0171   696          RET                             ;
                            0172   697
                    0000    0172   698 5$:      .WORD   0                       ; Null entry mask
          51  FFFFFFFC'EF      9E  0174   699          MOVAB   IOC$GL_ADPLIST-ADP$L_LINK,R1 ; Set starting address
             51  04 A1   D0  017B   700 10$:     MOVL    ADP$L_LINK(R1),R1       ; Flink onward
                   07   13  017F   701          BEQL    20$                     ; Done if at end
          0C A1  04 AC   B1  0181   702          CMPW    4(AP),ADP$W_TR(R1)      ; Is this the specified TR?
                   F3   12  0186   703          BNEQ    10$                     ; No, try another
                50  01   D0  0188   704 20$:     MOVL    #1,R0
                        04  018B   705          RET                             ;
                            018C   706
                    00FC    018C   707 .Entry   CONFIGADP, ^M<R2,R3,R4,R5,R6,R7>; Entry mask
                            018E   708
             000024C4'EF      D4  018E   709          CLRL    BOO$GL_RETSAVE          ; Zap return address for initial call
10 00000000'EF      06   E1  0194   710          BBC     #BOOCMD$V_SELECT,L^BOO$GL_CMDOPT,10$ ; Mutually exclusive - test
08 00000000'EF      07   E1  019C   711          BBC     #BOOCMD$V_EXCLUDE,L^BOO$GL_CMDOPT,10$ ; to make sure one bit clear
   50   007C808A 8F      D0  01A4   712          MOVL    #SYSG$_CONFQUAL,R0      ; Conflicting qualifiers
                        04  01AB   713          RET
                            01AC   714
             01FA'CF    6C   FA  01AC   715 10$:     CALLG   (AP),W^50$              ; Call configure one device
```

```
                     09 50    E8  01B1    716          BLBS    R0,20$                  ; Branch if not done with this adapter
                  50   24    B1  01B4    717          CMPW    #SS$_NOPRIV,R0          ; Was there a privilege error
                       03    13  01B7    718          BEQL    15$                     ; Yes, branch
                  50   01    D0  01B9    719          MOVL    #1,R0                   ; Set success
                             04  01BC    720 15$:     RET                             ; and return
                                 01BD    721
              55   0000248C'EF  9E  01BD    722 20$:     MOVAB   BOO$AL_ACF,R5           ; Set address of arguments describing device
                                 01C4    723
                        1C A5    B4  01C4    724          CLRW    ACF$W_MAXUNITS(R5)      ; Always use driver specified max units
              56   00002460'EF  D0  01C7    725          MOVL    L^BOO$GL_SELECT,R6      ; Get pointer to select list
                           06    13  01CE    726          BEQL    35$                     ; Branch if null
                        0087     30  01D0    727          BSBW    SELECT                  ; Check select/exclude string
                     D6 50       E9  01D3    728          BLBC    R0,10$                  ; Branch if device is not to be configured
                                 01D6    729
                  11 0B A5       03  E0  01D6    730 35$:     BBS     #ACF$V_NOLOAD_DB,ACF$B_AFLAG(R5),38$ ; Branch if not loading databas
           09 00000000'EF        0C  E1  01DB    731          BBC     #BOOCMD$V_AUTOLOG,L^BOO$GL_CMDOPT,38$ ; Branch if not logging
                        00A9     30  01E3    732          BSBW    AUTOLOG                 ; Branch to output log
                     03 50       E8  01E6    733          BLBS    R0,38$                  ; Branch if no error
                      FE14'      30  01E9    734          BSBW    PUTERROR                ; Give error message
                                 01EC    735
                0000'CF    65    FA  01EC    736 38$:     CALLG   (R5),W^IOGEN$LOADER     ; Load database and driver if necessary
                     B8 50       E8  01F1    737          BLBS    R0,10$                  ; Branch if no error
                      FE09'      30  01F4    738          BSBW    PUTERROR                ; Give error message
                      FFB2       31  01F7    739          BRW     10$                     ; continue loop
                                 01FA    740
                           0000  01FA    741 50$:     .WORD   0                       ;
                                 01FC    742          $CMKRNL_S       B^55$,(AP)      ; Call auto configure in kernel mode
                             04  0208    743          RET                             ;
                                 0209    744
                           0FFC  0209    745 55$:     .WORD   ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>     ;
           50   000024C4'EF      D0  020B    746          MOVL    BOO$GL_RETSAVE,R0       ; Get saved return address
                           07    12  0212    747          BNEQ    60$                     ; Branch if one present
           50   00000000'EF      9E  0214    748          MOVAB   IOC$AUTOCONFIG,R0       ; Else use main entry point
                           50    DD  021B    749 60$:     PUSHL   R0                      ; Stack call back address
                  58   04 AC     D0  021D    750          MOVL    4(AP),R8                ; Get address of ADP
                     56   68     D0  0221    751          MOVL    ADP$L_CSR(R8),R6        ; Get Configuration register address
              57   0000248C'EF   9E  0224    752          MOVAB   BOO$AL_ACF,R7           ; Address of configuration control block
                                 022B    753          SETIPL  #31                     ; Disable interrupts
                        9E       16  022E    754          JSB     @(SP)+                  ; Call Auto configurtation code
                                 0230    755          SETIPL  #0                      ; Enable interrupts
           000024C4'EF           8E  D0  0233    756          MOVL    (SP)+,BOO$GL_RETSAVE    ; Save return
                     06 50       E8  023A    757          BLBS    R0,70$                  ; Continue if another device
                000024C4'EF      D4  023D    758          CLRL    BOO$GL_RETSAVE          ; Else clear return
                                 0243    759
                OD 0B A7         03  E1  0243    760 70$:     BBC     #ACF$V_NOLOAD_DB,ACF$B_AFLAG(R7),80$ ; Branch if loading database
                   7E   12 A7    3C  0248    761          MOVZWL  ACF$W_CUNIT(R7),-(SP)   ; Get unit number
                        14 A7    DD  024C    762          PUSHL   ACF$L_DEVNAME(R7)       ; Get device name
                         010B    30  024F    763          BSBW    SGN$GET_DEVICE_LOCK_IODB; Get device database
                      5E   08    C0  0252    764          ADDL2   #8,SP                   ; Clear stack
                                 04  0255    765 80$:     RET                             ; And return
                      50   01    3C  0256    766 90$:     MOVZWL  #1,R0                   ; Set success status
                                 04  0259    767          RET                             ; and return
                                     025A    768 ;
                                     025A    769 ; SELECT - decide whether current device name is one of those either
                                     025A    770 ;          specified in /SELECT or /EXCLUDE
                                     025A    771 ;
                                     025A    772 ; Returns: R0 = 1    ==> configure device
```

```
                          025A   773 ;          R0 = 0   ==> don't configure device
                          025A   774 ;
                          025A   775
                          025A   776 SELECT:
          57    14 A5     D0   025A   777 10$:     MOVL    ACF$L_DEVNAME(R5),R7    ; Get pointer to device name
             54    86     9A   025E   778         MOVZBL  (R6)+,R4                ; Get length of select entry
                   10     13   0261   779         BEQL    20$                     ; End of list, no match
             54    87     91   0263   780         CMPB    (R7)+,R4                ; Compare with device entry
                   06     19   0266   781         BLSS    15$                     ; Branch if select longer than device
          67   66    54     29   0268   782         CMPC3   R4,(R6),(R7)            ; Do we have a match?
                   13     13   026C   783         BEQL    40$                     ; Yes, check SELECT or EXCLUDE
             56    54     C0   026E   784 15$:     ADDL    R4,R6                   ; Advance to next entry in select list
                   E7     11   0271   785         BRB     10$                     ; And try again
                          0273   786
                   50     D4   0273   787 20$:     CLRL    R0                      ; Assume don't configure
03 00000000'EF     07     E1   0275   788         BBC     #BOOCMD$V_EXCLUDE,BOO$GL_CMDOPT,30$ ; Branch if SELECT
                   50     01   D0   027D   789         MOVL    #1,R0                   ; EXCLUDE - configure device
                          05   0280   790 30$:     RSB
                          0281   791
                   50     D4   0281   792 40$:     CLRL    R0                      ; Assume don't configure
03 00000000'EF     07     E0   0283   793         BBS     #BOOCMD$V_EXCLUDE,BOO$GL_CMDOPT,50$ ; Branch if EXCLUDE
                   50     01   D0   028B   794         MOVL    #1,R0                   ; SELECT - configure device
                          05   028E   795 50$:     RSB
```

```
                                028F    797 .SBTTL  AUTOLOG - AUTO ALL /LOG formating
                                028F    798
                                028F    799 AUTOLOG::
        55   0000248C'EF    9E  028F    800         MOVAB   BOO$AL_ACF,R5              ; Address of configuration control block
                  56   14 A5  D0  0296    801         MOVL    ACF$L_DEVNAME(R5),R6     ; Get address of current device
                       57   86  9A  029A    802         MOVZBL  (R6)+,R7                 ; Get count and addr.
00002614'EF     66   57     29  029D    803         CMPC3   R7,(R6),BOO$GT_SAVE_DEVNAME ; Compare to previous string
                       39     12  02A5    804         BNEQ    50$                      ; Branch if new device
                                02A7    805
                                02A7    806         $FAO_S  CTRSTR=CTRSTR_AUTOLOG_UNIT ,- ; Format Unit Number
                                02A7    807                 OUTBUF=OUTBUF,-
                                02A7    808                 OUTLEN=OUTLEN_UNIT ,-
                                02A7    809                 P1=ACF$W_CUNIT(R5)
                         03 50  E8  02C3    810         BLBS    R0,40$                   ; Branch if OK
                          0081  31  02C6    811         BRW     100$                     ; Branch if error
                                02C9    812
    2610'CF     260C'CF     C0  02C9    813 40$:    ADDL2   W^OUTLEN_UNIT,W^OUTLEN   ; Add to total length
    262C'CF     260C'CF     C0  02D0    814         ADDL2   W^OUTLEN_UNIT,W^OUTBUF+4 ; Add to descriptor
    2628'CF     260C'CF     A2  02D7    815         SUBW2   W^OUTLEN_UNIT,W^OUTBUF   ; Subtract from length
                      6A    11  02DE    816         BRB     100$                     ; Return with success
                                02E0    817
                 2610'CF    D5  02E0    818 50$:    TSTL    W^OUTLEN                 ; Is this a first rall to this routine?
                      21    13  02E4    819         BEQL    70$                      ; Branch if yes
                                02E6    820
    262C'CF     2630'CF     DE  02E6    821         MOVAL   W^OUTBUF_STR,W^OUTBUF+4  ; reset descriptor
    0000'CF     2610'CF     B0  02ED    822         MOVW    W^OUTLEN,W^RIO$GW_OUTLEN ; Length of string
                0000'CF     28  02F4    823         MOVC3   W^RIO$GW_OUTLEN,-
                2630'CF         02F8    824                 W^OUTBUF_STR,-
                0000'CF         02FB    825                 W^RIO$AB_BUFFER          ; Move text into global buffer
                                02FE    826
          00000000'EF       16  02FE    827         JSB     RIO$OUTPUT_LINE
                 43 50      E9  0304    828         BLBC    R0,100$                  ; Branch on error
                                0307    829
    2628'CF     0064 8F     B0  0307    830 70$:    MOVW    #100,W^OUTBUF            ; Set full buffer length
00002614'EF     66     57   28  030E    831         MOVC3   R7,(R6),BOO$GT_SAVE_DEVNAME ; Save new devname
      55   0000248C'EF       9E  0316    832         MOVAB   BOO$AL_ACF,R5            ; Reset R5
                                031D    833         $FAO_S  CTRSTR=CTRSTR_AUTOLOG,- ; Format device name
                                031D    834                 OUTBUF=OUTBUF,-
                                031D    835                 OUTLEN=OUTLEN,-
                                031D    836                 P1=ACF$L_DEVNAME(R5),-
                                031D    837                 P2=ACF$W_CUNIT(R5)
    262C'CF     2610'CF     C0  033C    838         ADDL2   W^OUTLEN,W^OUTBUF+4      ; Add to descriptor
    2628'CF     2610'CF     A2  0343    839         SUBW2   W^OUTLEN,W^OUTBUF        ; Subtract from length
                                034A    840                                          ; Return with FAO status
                          05    034A    841 100$:   RSB
                                034B    842
```

```
                                034B   844 .SBTTL  SGN$GET_DEVICE - Locate device database
                                034B   845
                                034B   846 ;
                                034B   847 ; Inputs:
                                034B   848 ;       4(SP) - Address of Device name in ascic format
                                034B   849 ;       8(SP) - Unit number
                                034B   850 ;
                                034B   851 ; Outputs:
                                034B   852 ;       (Any of these are 0 if the data block doesn't exist)
                                034B   853 ;       ACF$GL_DDB - Address of DDB
                                034B   854 ;       ACF$GL_UCB - Address of UCB
                                034B   855 ;       ACF$GL_IDB - Address of IDB
                                034B   856 ;       ACF$GL_CRB - Address of CRB
                                034B   857 ;       ACF$GL_SB  - Address of SB
                                034B   858 ;       ACF$GL_LASTDDB - If ACF$GL_DDB is non-zero, then equal to that,
                                034B   859 ;                        otherwise, last DDB in DEVLIST
                                034B   860 ;       R0 = 0 - error
                                034B   861 ;          = 1 - success
                                034B   862 ;
                                034B   863 ; Must be called at IPL=0 and KERNEL mode
                                034B   864 ;
                                034B   865         .ENABL  LSB
                                034B   866
                                034B   867 SGN$GET_DEVICE::                         ; Entry with IODB MUTEX & raised IPL
                                034B   868
                    007C 8F     BB  034B   869         PUSHR   #^M<R2,R3,R4,R5,R6>  ; ADDS 20 to offset to input
                                034F   870
          54  00000000'GF     DO  034F   871         MOVL    G^CTL$GL_PCB,R4      ; PICK UP PCB POINTER
                         32     10  0356   872         BSBB    10$                  ; Call real routine
                                0358   873
                    007C 8F     BA  0358   874         POPR    #^M<R2,R3,R4,R5,R6>  ; restore regs
                                05  035C   875         RSB                          ; Return
                                035D   876
                                035D   877 SGN$GET_DEVICE_LOCK_IODB:                ; Entry without IODB MUTEX and IPL 0
                                035D   878
                    007C 8F     BB  035D   879         PUSHR   #^M<R2,R3,R4,R5,R6>  ; ADDS 20 to offset to input
                                0361   880
          54  00000000'GF     DO  0361   881         MOVL    G^CTL$GL_PCB,R4      ;PICK UP PCB POINTER
              00000000'GF     16  0368   882         JSB     G^SCH$IOLOCKR        ;GET THE IODB MUTEX FOR READ & RAISE IPL
                         1A     10  036E   883         BSBB    10$                  ;
                         50     DD  0370   884         PUSHL   R0                   ;SAVE RETURN STATUS
          54  00000000'GF     DO  0372   885         MOVL    G^CTL$GL_PCB,R4      ;PICK UP PCB POINTER
              00000000'GF     16  0379   886         JSB     G^SCH$IOUNLOCK       ;RELEASE THE IODB MUTEX
                                037F   887         SETIPL  #0                   ;LOWER IPL
                                0382   888
                         50 8ED0    0382   889         POPL    R0                   ;RESTORE RETURN STATUS FROM LOCAL ROUTINE
                    007C 8F     BA  0385   890         POPR    #^M<R2,R3,R4,R5,R6>
                                05  0389   891         RSB
                                038A   892
                    2400'CF     D4  038A   893 10$:     CLRL    W^ACF$GL_DDB         ;INIT TO ZERO
                    2404'CF     D4  038E   894         CLRL    W^ACF$GL_UCB         ;INIT TO ZERO
                    2408'CF     D4  0392   895         CLRL    W^ACF$GL_IDB         ;INIT TO ZERO
                    240C'CF     D4  0396   896         CLRL    W^ACF$GL_CRB         ;INIT TO ZERO
                    2418'CF     D4  039A   897         CLRL    W^ACF$GL_SB          ;INIT TO ZERO
                                039E   898
                                039E   899         SAVIPL  -(SP)                ;SAVE THE CURRENT IPL
                     56   20 AE     D0  03A1   900         MOVL    32(SP),R6            ;GET ADDR OF DEVICE NAME
```

```
                55   86   9A  03A5   901          MOVZBL  (R6)+,R5                 ;GET SIZE OF DEVICE NAME
                7E   55   7D  03A8   902          MOVQ    R5,-(SP)                ;FORM DESCRIPTOR
                51   5E   D0  03AB   903          MOVL    SP,R1                   ;ADDRESS OF DESCRIPTOR
         00000000'GF   16   03AE   904          JSB     G^IOC$SEARCHALL         ;SEARCH FOR DEVICE
                8E   7C   03B4   905          CLRQ    (SP)+                   ;GET RID OF TRASH
                     03B6   906          SETIPL  (SP)+                   ;RESTORE OLD IPL
        2418'CF   53   D0   03B9   907          MOVL    R3,W^ACF$GL_SB          ;STUFF THE SYSTEM BLOCK
                04   12   03BE   908          BNEQ    20$                     ;NO ERROR, CONTINUE
                50   D4   03C0   909          CLRL    R0                      ;INDICATE ERROR
                5E   11   03C2   910          BRB     70$                     ;EXIT
                     03C4   911
             0B 50   E8   03C4   912 20$:     BLBS    R0,25$                  ;SUCCES - FOUND DEVICE
        0908 8F   50   B1   03C7   913          CMPW    R0,#SS$_NOSUCHDEV       ;CHECK IF ERROR WAS "UNIT NOT FOUND"
                36   12   03CC   914          BNEQ    60$                     ;IF NOT, PUNT
                51   D5   03CE   915          TSTL    R1                      ;SEE IF WE GOT BACK A UCB ADDRESS
                32   12   03D0   916          BNEQ    60$                     ;IF NON-ZERO, IS LISTHEAD - NO DDB FOUND
    00002400'EF   52   D0   03D2   917 25$:     MOVL    R2,L^ACF$GL_DDB         ;ADDRESS OF DDB
          54 04   A2   D0   03D9   918          MOVL    DDB$L_UCB(R2),R4        ;GET ADDRESS OF FIRST UCB
                25   13   03DD   919          BEQL    60$                     ;IF NO UCB, EXIT WITH OTHER FIELDS ZERO
          51 24   A4   D0   03DF   920          MOVL    UCB$L_CRB(R4),R1        ;GET ADDR OF CRB
    0000240C'EF   51   D0   03E3   921          MOVL    R1,L^ACF$GL_CRB         ;SAVE
     2408'CF   2C A1   D0   03EA   922          MOVL    CRB$L_INTD+VEC$L_IDB(R1),W^ACF$GL_IDB ;GET ADDR OF IDB
                     03F0   923
       54 A4   20 AE   B1   03F0   924 30$:     CMPW    32(SP),UCB$W_UNIT(R4)   ;IS UCB ALREADY LOADED?
                08   13   03F5   925          BEQL    50$                     ;BRANCH IF IT IS
          54    30 A4   D0   03F7   926 40$:     MOVL    UCB$L_LINK(R4),R4       ;GET ADDR OF NEXT UCB
                F3   12   03FB   927          BNEQ    30$                     ;BR IF THERE IS ONE
                05   11   03FD   928          BRB     60$                     ;EXIT WITH UCB = 0
                     03FF   929
         2404'CF   54   D0   03FF   930 50$:     MOVL    R4,W^ACF$GL_UCB
    00002410'EF   52   D0   0404   931 60$:     MOVL    R2,ACF$GL_LASTDDB       ;LAST DDB IN LIST AS SEARCHED
50  00000000'GF   DE   040B   932          MOVAL   G^SCS$GA_LOCALSB,R0     ;GET ADDRESS OF LOCAL SYSTEM BLOCK
             50   53   D1   0412   933          CMPL    R3,R0                   ;IS THIS SB LOCAL?
                08   13   0415   934          BEQL    65$                     ;YES, LEAVE NOW
              18 A3   7D   0417   935          MOVQ    SB$B_SYSTEMID(R3),-
        0000244C'EF        041A   936                  L^BOO$GQ_CONSYSID       ;NO, SET IN THE SYSTEM ID
                     041F   937
             50   01   D0   041F   938 65$:     MOVL    #1,R0                   ;SUCCESS
                05   0422   939 70$:     RSB
                     0423   940
                     0423   941          .DSABL  LSB
                     0423   942
```

```
                                0423   944          .SBTTL  Reset routines BOO$RESETLIST and BOO$CONRESET and BOO$MSCP_RESET
                                0423   945 ;
                                0423   946 ; BOO$CONRESET - Reset values for connect command
                                0423   947 ;
                                0423   948
                            00000200   949 .PSECT  PAGED_CODE      rd,nowrt,exe,long
                                0200   950
                              0000  0200   951 .Entry  BOO$CONRESET, ^M<>                        ; Null entry mask
                                0202   952
0000245C'EF   00000200'EF   9E  0202   953          MOVAB   L^BOO$AB_PRMBUF,BOO$GL_NEXTSTR  ; Reset for string allocation
        00002428'EF    01   CE  020D   954          MNEGL   #1,BOO$GL_CONCREG               ; Null control register
        0000243C'EF    01   CE  0214   955          MNEGL   #1,BOO$GL_CONAUNIT              ; Null adapter unit
        00002434'EF    01   CE  021B   956          MNEGL   #1,BOO$GL_CONVECT               ; Null vector
        00002438'EF    01   D0  0222   957          MOVL    #1,BOO$GL_CONNUMV               ; Default number of vectors
        00002424'EF    02   CE  0229   958          MNEGL   #2,BOO$GL_CONADP                ; Invalidate adapter TR value
             00002440'EF    D4  0230   959          CLRL    BOO$GL_CONDEV                   ; Clear device name pointer
             00002444'EF    D4  0236   960          CLRL    BOO$GL_CONDRV                   ; and driver name pointer
             00002448'EF    D4  023C   961          CLRL    BOO$GL_CONUNITS                 ; and maximum units
             0000244C'EF    7C  0242   962          CLRQ    BOO$GQ_CONSYSID                 ; and system id
             00002458'EF    D4  0248   963          CLRL    BOO$GL_CONFLAGS                 ; and flags
        00002430'EF    01   D0  024E   964          MOVL    #1,L^BOO$GL_CONNUMU             ; Set number of units to 1
             0000241C'EF    D4  0255   965          CLRL    BOO$GL_COMBO_VECTOR_OFFSET; Set vector offset from combo vectors to
             00002420'EF    D4  025B   966          CLRL    BOO$GL_COMBO_CSR_OFFSET ; Set CSR offset from combo CSR to 0
                    04      0261   967          RET                                     ; Return
                            0262   968 ;
                            0262   969 ;       BOO$RESETLIST - Reset select list values
                            0262   970 ;
                              0000  0262   971 .Entry  BOO$RESETLIST, ^M<>                    ; Null entry mask
                                0264   972
        00002460'EF    D4  0264   973          CLRL    BOO$GL_SELECT                   ; Zap select list pointer
0000245C'EF   00000200'EF   9E  026A   974          MOVAB   BOO$AB_PRMBUF,BOO$GL_NEXTSTR ; Set next string address
        00002614'EF    D4  0275   975          CLRL    BOO$GT_SAVE_DEVNAME             ; Clear autolog string
        00002610'EF    D4  027B   976          CLRL    OUTLEN                          ; Clear autolog output size
0000262C'EF   00002630'EF   DE  0281   977          MOVAL   OUTBUF_STR,OUTBUF+4             ; Set address in descriptor of block
        00002497'EF    94  028C   978          CLRB    BOO$AL_ACF+ACF$B_AFLAG  ; Clear ACF flags
        00002430'EF    01   D0  0292   979          MOVL    #1,L^BOO$GL_CONNUMU             ; Set number of units to 1
                    04      0299   980          RET                                     ; and return
                            029A   981
                            029A   982 ;
                            029A   983 ; BOO$MSCP_RESET - Reset the MSCP server initialization argument list
                            029A   984 ;
                              003C  029A   985 .Entry  BOO$MSCP_RESET, ^M<R2,R3,R4,R5> ; Entry mask
                                029C   986
             FF5F   CF    00  FB  029C   987          CALLS   #0,@BOO$CONRESET                ; Reset the connect command globals
             50    0084   8F  3C  02A1   988          MOVZWL  #SS$_DEVOFFLINE,R0              ; Assume error
              00000000'GF    D5  02A6   989          TSTL    G^SCS$GL_CDL                    ; SCS loaded?
                     2C    13  02AC   990          BEQL    10$                             ; If eql no, error
             50    02C4   8F  3C  02AE   991          MOVZWL  #SS$_DEVACTIVE,R0               ; Assume error
        50   03    00    02  F0  02B3   992          INSV    #2,#0,#3,R0                     ; Set E class error status
              00000000'GF    D5  02B8   993          TSTL    G^SCS$GL_MSCP                   ; If neq already loaded
                     1A    12  02BE   994          BNEQ    10$                             ; Exit with error
00002444'GF   000025EC'EF   DE  02C0   995          MOVAL   MSCP_NAME,G^BOO$GL_CONDRV; Set pointer to MSCP server name
                     30    28  02CB   996          MOVC3   #MSCP_ARG_LIST_SIZE,-           ; Set up default argument list for
000025BC'GF   0000258C'EF       02CD   997                  MSCP_ARG_LIST,G^BOO$GL_LOAD_ARGS; MSCP server init routine
                  50    01    D0  02D7   998          MOVL    #1,R0                           ; Set success
                    04      02DA   999 10$:     RET                                     ; and return
                            02DB  1000
```

```
                        02DB  1001 ;
                        02DB  1002 ; BOO$MSCP_ARG - Load MSCP arguments
                        02DB  1003 ;
                  0000  02DB  1004 .Entry  BOO$MSCP_ARG, ^M<>                 ; Entry mask
                        02DD  1005
      50    20 AC     D0  02DD  1006          MOVL    TPA$L_PARAM(AP),R0      ; Get longword offset
            1C AC     D0  02E1  1007          MOVL    TPA$L_NUMBER(AP),-      ; Load argument value
 000025BC'GF40            02E4  1008                  G^BOO$GL_LOAD_ARGS[R0]  ;
            50    01     D0  02EA  1009          MOVL    #1,R0                   ; Set success
                  04     D0  02ED  1010          RET                             ; and return
                        02EE  1011
                        02EE  1012 ;
                        02EE  1013 ;
                        02EE  1014 ; BOO$MAKLIST - Make a select list entry
                        02EE  1015 ;
                  007C  02EE  1016 .Entry  BOO$MAKLIST, ^M<R2,R3,R4,R5,R6> ; Entry mask
                        02F0  1017
56  0000245C'EF     D0  02F0  1018          MOVL    L^BOO$GL_NEXTSTR,R6     ; Get pointer to next available string space
      00002460'EF   D5  02F7  1019          TSTL    L^BOO$GL_SELECT         ; Is selection pointer already set
                  07  12  02FD  1020          BNEQ    10$                     ; Yes, continue to add entry
00002460'EF       56  D0  02FF  1021          MOVL    R6,L^BOO$GL_SELECT      ; Else set pointer to first select entry
      50    10 AC     D0  0306  1022 10$:     MOVL    TPA$L_TOKENCNT(AP),R0   ; Get string length
            86    50  90  030A  1023          MOVB    R0,(R6)+                ; Set count for string
 66   14 BC    50     28  030D  1024          MOVC3   R0,@TPA$L_TOKENPTR(AP),(R6) ; Copy string body
                  63  94  0312  1025          CLRB    (R3)                    ; Mark end of list
0000245C'EF       53  D0  0314  1026          MOVL    R3,L^BOO$GL_NEXTSTR     ; Save next string address
            50    01  D0  031B  1027          MOVL    #1,R0                   ; Set success status
                  04  031E  1028          RET                             ;
```

```
                                 031F  1030          .SBTTL  BOO$CONADP - Set connect adapter number
                                 031F  1031
                           0000  031F  1032 .Entry   BOO$CONADP, ^M<>                        ;
    00002424'EF    1C AC     D0  0321  1033          MOVL    TPA$L_NUMBER(AP),L^BOO$GL_CONADP ; Set adapter number
                             04  0329  1034          RET                                     ; and return
                                 032A  1035
                           0000  032A  1036 .Entry   BOO$CONNLADP ^M<>                       ; Connect with null adapter
       00002424'EF      01   CE  032C  1037          MNEGL   #1,L^BOO$GL_CONADP              ; Clear adapter number
                             04  0333  1038          RET                                     ; and return
                                 0334  1039
                           0000  0334  1040 .Entry   BOO$CONVECOFFSET, ^M<>                  ; Offset from start of combo vectors
                     1C AC     D0  0336  1041          MOVL    TPA$L_NUMBER(AP),-              ; Set offset value
                00002410'EF          0339  1042                  L^BOO$GL_COMBO_VECTOR_OFFSET
                             04  033E  1043          RET                                     ; and return
                                 033F  1044
                           0000  033F  1045 .Entry   BOO$CONCSROFFSET, ^M<>                  ; Offset from start of combo CSRs
                     1C AC     D0  0341  1046          MOVL    TPA$L_NUMBER(AP),-              ; Set offset value
                00002420'EF          0344  1047                  L^BOO$GL_COMBO_CSR_OFFSET
                             04  0349  1048          RET                                     ; and return
                                 034A  1049
                           0000  034A  1050 .Entry   BOO$CONCREG, ^M<>                       ; Control register address
    00002428'EF    1C AC   0D   00   EF  034C  1051          EXTZV   #0,#13,TPA$L_NUMBER(AP),L^BOO$GL_CONCREG; Set control register
                             04  0356  1052          RET                                     ; and return
                                 0357  1053
                           0000  0357  1054 .Entry   BOO$CONCVEC, ^M<>                       ; Set controller vector
          1C AC    FFFFFE03 8F   CB  0359  1055          BICL3   #^XFFFFFE03,TPA$L_NUMBER(AP),L^BOO$GL_CONVECT ; Set vector offset
                00002434'EF          0361
                             04  0366  1056          RET                                     ; and return
                                 0367  1057
                           0000  0367  1058 .Entry   BOO$CONCNUM, ^M<>                       ; Number of vectors
       00002438'EF    1C AC     D0  0369  1059          MOVL    TPA$L_NUMBER(AP),L^BOO$GL_CONNUMV ; Set number of vectors
                             04  0371  1060          RET                                     ; and return
                                 0372  1061
                           0000  0372  1062 .Entry   BOO$CONAUNIT, ^M<>                      ; Adapter unit number
       0000243C'EF    1C AC     D0  0374  1063          MOVL    TPA$L_NUMBER(AP),L^BOO$GL_CONAUNIT; Set adapter unit number
                             04  037C  1064          RET                                     ; and return
                                 037D  1065
                           007C  037D  1066 .Entry   BOO$CONDRVNAM, ^M<R2,R3,R4,R5,R6> ; Entry mask (R2-R6)
                                 037F  1067
          56    0000245C'EF     D0  037F  1068          MOVL    L^BOO$GL_NEXTSTR,R6             ; Address of next string storage
          00002444'EF    56     D0  0386  1069          MOVL    R6,BOO$GL_CONDRV                ; Save pointer to driver name
                  86    10 AC   90  038D  1070          MOVB    TPA$L_TOKENCNT(AP),(R6)+        ; Set count for string
    0000245C'EF   56    10 AC   C1  0391  1071          ADDL3   TPA$L_TOKENCNT(AP),R6,BOO$GL_NEXTSTR ; Mark string allocated
          66  14 BC    10 AC   28  039A  1072          MOVC3   TPA$L_TOKENCNT(AP),@TPA$L_TOKENPTR(AP),(R6)     ; Copy string
                50    01   D0  03A0  1073          MOVL    #1,R0                           ; and return success
                             04  03A3  1074          RET                                     ;
                                 03A4  1075
                           00FC  03A4  1076 .Entry   BOO$DEVNAME, ^M<R2,R3,R4,R5,R6,R7> ; Device name/unit
                                 03A6  1077
          56    0000245C'EF     D0  03A6  1078          MOVL    BOO$GL_NEXTSTR,R6               ; Get pointer to next available string
                  54    14 AC   D0  03AD  1079          MOVL    TPA$L_TOKENPTR(AP),R4           ; Get pointer to string
                  53    10 AC   D0  03B1  1080          MOVL    TPA$L_TOKENCNT(AP),R3           ; And number of characters
                00002588'EF     D4  03B5  1081          CLRL    FULL_NAME_PTR                   ; Initialize full device name
                     57    86   9E  03B8  1082          MOVAB   (R6)+,R7                        ; Save pointer
                64   53    24   3A  03BE  1083          LOCC    #^A/$/,R3,(R4)                  ; Find any possible "$"
                           22   13  03C2  1084          BEQL    8$                              ; None, just continue
       00002588'EF    57     D0  03C4  1085          MOVL    R7,FULL_NAME_PTR                ; Store pointer
```

```
                55   53   50   C3   03CB  1086          SUBL3   R0,R3,R5                ; Number of characters in node
                67   55   01   81   03CF  1087          ADDB3   #1,R5,(R7)              ; Set in size (incl "$")
                          03   BB   03D3  1088          PUSHR   #^M<R0,R1>              ; Save registers
                66   64   53   28   03D5  1089          MOVC3   R3,(R4),(R6)            ; Copy full string
                     56   53   D0   03D9  1090          MOVL    R3,R6                   ; Save ending address
                          03   BA   03DC  1091          POPR    #^M<R0,R1>              ; Restore registers
                53   50   01   C3   03DE  1092          SUBL3   #1,R0,R3                ; Number of characters left
                54   51   01   C1   03E2  1093          ADDL3   #1,R1,R4                ; Pointer to string
                     55   86   9E   03E6  1094 8$:       MOVAB   (R6)+,R5                ; Save pointer to count byte
                          65   94   03E9  1095          CLRB    (R5)                    ; Initialize count to zero
                          52   D4   03EB  1096          CLRL    R2                      ; Initialize unit accumulator
                          50   84   9A   03ED  1097 10$:      MOVZBL  (R4)+,R0                ; Get a character from device name
                          30   50   91   03F0  1098          CMPB    R0,#^A/0/               ; And check for a digit
                               05   1F   03F3  1099          BLSSU   20$                     ; Branch if not
                          39   50   91   03F5  1100          CMPB    R0,#^A/9/               ; Final check for digit
                               0F   1B   03F8  1101          BLEQU   40$                     ; Yes it is
                          86   50   90   03FA  1102 20$:      MOVB    R0,(R6)+                ; Part of device name
                               65   96   03FD  1103          INCB    (R5)                    ; Increase count
                               67   96   03FF  1104          INCB    (R7)                    ; Including nodename
                     E9   53   F5   0401  1105          SOBGTR  R3,10$                  ; Continue
                               16   11   0404  1106          BRB     50$                     ;
                          50   84   9A   0406  1107 30$:      MOVZBL  (R4)+,R0                ; Get another digit
                          50   30   C2   0409  1108 40$:      SUBL    #^A/0/,R0               ; Get value
                          52   0A   C4   040C  1109          MULL    #10,R2                  ; Scale accumulator before adding digit
                               2F   19   040F  1110          BLSS    60$                     ; Error
                          50   09   D1   0411  1111          CMPL    #9,R0                   ; Check for numeric
                               2A   19   0414  1112          BLSS    60$                     ; Error if not
                          52   50   C0   0416  1113          ADDL    R0,R2                   ; And add new digit
                     EA   53   F5   0419  1114          SOBGTR  R3,30$                  ; Continue for entire unit number
0000245C'EF              56   D0   041C  1115 50$:      MOVL    R6,BOO$GL_NEXTSTR       ; Save updated string pointer
0000242C'EF              52   D0   0423  1116          MOVL    R2,BOO$GL_CONCUNIT      ; Set unit number
0000243C'EF              52   D0   042A  1117          MOVL    R2,BOO$GL_CONAUNIT      ; Assume same for adapter unit
00002440'EF              55   D0   0431  1118          MOVL    R5,BOO$GL_CONDEV        ; Save device name pointer
                         65   95   0438  1119          TSTB    (R5)                    ; Must not be null device name
                         04   13   043A  1120          BEQL    60$                     ; Error if so
                    50   01   D0   043C  1121          MOVL    #1,R0                   ; Return success
                         04   043F  1122          RET                             ; and return
                         50   D4   0440  1123 60$:      CLRL    R0                      ; Return error status
                         04   0442  1124          RET                             ;
                              0443  1125
                    0000  0443  1126 .Entry   BOO$CONUNITS, ^M<>              ; Maximun units to be connected
00002448'EF    1C AC   D0   0445  1127          MOVL    TPA$L_NUMBER(AP),L^BOO$GL_CONUNITS ; Set maximum units
                         04   044D  1128          RET                             ; and return
                              044E  1129
                    0000  044E  1130 .Entry   BOO$CONSYSID_LOW, ^M<>          ; System ID
0000244C'EF    1C AC   D0   0450  1131          MOVL    TPA$L_NUMBER(AP), -
                              0458  1132                  L^BOO$GQ_CONSYSID       ; Set System ID (low longword)
                         04   0458  1133          RET                             ; and return
                              0459  1134
                    0000  0459  1135 .Entry   BOO$CONSYSID_HIGH, ^M<>         ; System ID
00002450'EF    1C AC   D0   045B  1136          MOVL    TPA$L_NUMBER(AP), -
                              0463  1137                  L^BOO$GQ_CONSYSID+4     ; Set System ID (high longword)
                         04   0463  1138          RET                             ; and return
                              0464  1139
                    0000  0464  1140 .Entry   BOO$CONSOLE, ^M<>               ; Connect console block stor. device
                              0466  1141
00002424'EF         01   CE   0466  1142          MNEGL   #1,L^BOO$GL_CONADP      ; No adapter
```

```
                0000243C'EF    01   D0  046D  1143          MOVL    #1,L^BOO$GL_CONAUNIT    ; Set adapter unit = 1 (not used)
                0000242C'EF    01   D0  0474  1144          MOVL    #1,L^BOO$GL_CONCUNIT    ; Set unit = 1
    00002440'EF    000024D6'EF   9E  047B  1145          MOVAB   L^CONSNAME,L^BOO$GL_CONDEV ; Set device name pointer
                00002438'EF    02   D0  0486  1146          MOVL    #2,L^BOO$GL_CONNUMV     ; Set 2 vectors
                     00002428'EF   D4  048D  1147          CLRL    L^BOO$GL_CONCREG        ; No control register
                00002430'EF    01   D0  0493  1148          MOVL    #1,L^BOO$GL_CONNUMU     ; Set number of units to 1
                00002448'EF    02   D0  049A  1149          MOVL    #2,L^BOO$GL_CONUNITS    ; Set max units to 2 (OPA0 is 1st unit)
                     00000000'EF   16  04A1  1150          JSB     IOGEN$CONSOLE           ; Do cpu dependent stuff
                           50   01   D0  04A7  1151          MOVL    #1,R0
                                     04  04AA  1152          RET
```

```
                                04AB  1154          .SBTTL  BOO$CONNECT - Connnect specified device and load driver
                                04AB  1155 ;
                                04AB  1156 ; BOO$CONNECT - Allows a single device to be introduced, appropriate data
                                04AB  1157 ;               structures allocated and initialized, the driver loaded if
                                04AB  1158 ;               required and the controller and device initialized.
                                04AB  1159 ;
                           0FFC 04AB  1160 .Entry  BOO$CONNECT, ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>        ;
                                04AD  1161
                     FB50'  30  04AD  1162          BSBW    BOO$LOCK_GEN            ; Lock SYSGEN database
                   7F 50    E9  04B0  1163          BLBC    R0,70$                  ; If error, exit
                                04B3  1164
                                04B3  1165 ;
                                04B3  1166 ; Value of BOO$GL_CONADP
                                04B3  1167 ;
                                04B3  1168 ;       0 or greater    => /ADADPTER=n specified
                                04B3  1169 ;       -1              => /NOADAPTER specified
                                04B3  1170 ;       -2              => not specified
                                04B3  1171 ;
                                04B3  1172
          00002424'EF    D5  04B3  1173 5$:     TSTL    L^BOO$GL_CONADP         ; Has an adapter been specified?
                      44   18  04B9  1174          BGEQ    20$                     ; If so, branch
00002424'EF    FFFFFFFF 8F   D1  04BB  1175          CMPL    #-1,L^BOO$GL_CONADP     ; Null adapter?
                      28   13  04C6  1176          BEQL    10$                     ; Branch if yes
00002424'EF    FFFFFFFE 8F   D1  04C8  1177          CMPL    #-2,L^BOO$GL_CONADP     ; None specified in CONNECT?
                      09   13  04D3  1178          BEQL    7$                      ; Figure it out from the database
          50   007C80D2 8F   D0  04D5  1179          MOVL    #SYSG$_NOADAPTER,R0     ; Set no adapter specified error
                      45   11  04DC  1180          BRB     60$                     ; exit
                                04DE  1181
                                04DE  1182 7$:      $CMKRNL_S W^CONN_ADAP           ; Get adapter number from I/O database
                   35 50    E9  04EB  1183          BLBC    R0,60$                  ; Exit with error
                      C3    11  04EE  1184          BRB     5$                      ; Dispatch now on adapter type
                                04F0  1185
                                04F0  1186 10$:     $CMKRNL_S       W^CONNLADP      ; Change mode to see data base
                      0D    11  04FD  1187          BRB     30$                     ; Continue
                                04FF  1188
                                04FF  1189 20$:     $CMKRNL_S       W^CONNECT       ; Change mode to see data base
                   0E 50    E9  050C  1190 30$:     BLBC    R0,40$                  ; Error occured
00000000'EF    0000248C'EF   FA  050F  1191          CALLG   L^BOO$AL_ACF,IOGEN$LOADER ; Load database and driver
                   03 50    E8  051A  1192          BLBS    R0,50$                  ; Branch if success
                    FAE0'   30  051D  1193 40$:     BSBW    PUTERROR                ; Give error message
                50    01    D0  0520  1194 50$:     MOVL    #1,R0                   ; Set success for parser
                      50    DD  0523  1195 60$:     PUSHL   R0                      ; Save error status
                    FAD8'   30  0525  1196          BSBW    BOO$UNLOCK_GEN          ; Unlock SYSGEN database
                   03 50    E8  0528  1197          BLBS    R0,65$                  ; If no error, continue
                    FAD2'   30  052B  1198          BSBW    PUTERROR                ; Give error message
                      50 8ED0  052E  1199 65$:     POPL    R0                      ; Restore status
                         04    0531  1200          RET                             ;
                    FACB'   30  0532  1201 70$:     BSBW    PUTERROR                ; Give error message
                         04    0535  1202          RET                             ;
                                0536  1203 ;
                                0536  1204 ; Local routine to get adapter number from I/O database
                                0536  1205 ; Must be called by a CMKRNL since SGN$GET_DEVICE must be called
                                0536  1206 ; in Kernel mode.
                                0536  1207 ;
                          0000  0536  1208 .Entry  CONN_ADAP, ^M<>
                                0538  1209
          7E   0000242C'EF   3C  0538  1210          MOVZWL  L^BOO$GL_CONCUNIT,-(SP) ; Unit number
```

```
                00002440'EF   DD  053F  1211          PUSHL   L^BOO$GL_CONDEV          ; Device name
                0000035D'EF   16  0545  1212          JSB     SGN$GET_DEVICE_LOCK_IODB; Get device data base addresses
                    5E   08   C0  054B  1213          ADDL2   #8,SP                    ; Pop off input parameters
                                0054E  1214
     50         00002408'EF   D0  054E  1215          MOVL    L^ACF$GL_IDB,R0          ; Address of IDB
                              09  12  0555  1216          BNEQ    5$                       ; Error if zero
     50         007C80D2      8F  D0  0557  1217          MOVL    #SYSG$_NOADAPTER,R0      ; Set no adapter specified error
                              18  11  055E  1218          BRB     20$                      ; Branch to exit
                                0560  1219
                00002424'EF   01  CE  0560  1220 5$:      MNEGL   #1,L^BOO$GL_CONADP       ; Assume null adapter
                    50   14   A0  D0  0567  1221          MOVL    IDB$L_ADP(R0),R0         ; Address of ADP block
                              08  13  056B  1222          BEQL    10$                      ; Null adapter if zero
     00002424'EF   0C   A0  3C  056D  1223          MOVZWL  ADP$W_TR(R0),L^BOO$GL_CONADP ;Set adapter number
                                0575  1224
                    50   01   D0  0575  1225 10$:      MOVL    #1,R0                    ; Set success
                              04  0578  1226 20$:      RET                              ; Return
                                0579  1227
```

```
                                0579  1229 .ENABL  LSB
                                0579  1230 ; Connect with null adapter
                                0579  1231
                          0FFC  0579  1232 .Entry  CONNLADP, ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
                                057B  1233
      50  00002588'EF     07    D0  057B  1234          MOVL    L^FULL_NAME_PTR,R0      ; Try full device name
                            07    12  0582  1235          BNEQ    5$                      ; Good, continue
      50  00002440'EF           D0  0584  1236          MOVL    L^BOO$GL_CONDEV,R0      ; Use normal name
      5A  0000248C'EF           9E  058B  1237 5$:      MOVAB   L^BOO$AL_ACF,R10        ; Address ACF
                            6A    D4  0592  1238          CLRL    ACF$L_ADAPTER(R10)      ; Set no adapter
                         04 AA    D4  0594  1239          CLRL    ACF$L_CONFIGREG(R10)    ; Set address of config reg
                         08 AA    B4  0597  1240          CLRW    ACF$W_AVECTOR(R10)      ; Set SCB offset for adapter
                            01    E1  059A  1241          BBC     #ACF$V_CRBBLT,-         ; Br. if CRB built flag is clear
                  4C 00002458'EF      059C  1242                  BOO$GL_CONFLAGS,17$
      6A  00002454'EF           D0  05A2  1243          MOVL    BOO$GL_CONCRB,ACF$L_ADAPTER(R10)  ; Store CRB address
                            43    11  05A9  1244          BRB     17$                     ; Join common code
                                05AB  1245
                          0FFC  05AB  1246 .Entry  CONNECT, ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>    ;
                                05AD  1247
      5B  FFFFFFFC'GF           9E  05AD  1248          MOVAB   G^IOC$GL_ADPLIST-ADP$L_LINK,R11 ; Get address of adapter list
                   5B   04 AB   D0  05B4  1249 10$:     MOVL    ADP$L_LINK(R11),R11     ; Flink onward through adapter list
                            08    12  05B8  1250          BNEQ    15$                     ; Continue if another adapter
      50  007C80BA 8F           D0  05BA  1251          MOVL    #SYSG$_INVADAP,R0       ; Set invalid adapter error
                            04    05C1  1252          RET                             ; Return
                                05C2  1253
00002424'EF   0C AB           B1  05C2  1254 15$:     CMPW    ADP$W_TR(R11),L^BOO$GL_CONADP   ; Is this the specified TR?
                            E8    12  05CA  1255          BNEQ    10$                     ; No, try another
      5A  0000248C'EF           9E  05CC  1256          MOVAB   L^BOO$AL_ACF,R10        ; Get address of ACF
                   6A       5B   D0  05D3  1257          MOVL    R11,ACF$L_ADAPTER(R10)  ; Set address of ADP
                04 AA       6B   D0  05D6  1258          MOVL    ADP$L_CSR(R11),ACF$L_CONFIGREG(R10)     ; Set address of config reg
50  1C AB  00000000'GF      C3  05DA  1259          SUBL3   G^EXE$GL_SCB, -         ; Calculate offset into SCB of
                                05E3  1260                  ADP$L_AVECTOR(R11),R0   ; adapter's interrupt vectors.
                08 AA       50   B0  05E3  1261          MOVW    R0,ACF$W_AVECTOR(R10)   ; Store offset in ACF.
      50  00002440'EF           D0  05E7  1262          MOVL    L^BOO$GL_CONDEV,R0      ; Device name
                                05EE  1263
                                05EE  1264 17$:
      14 AA  00002440'EF        D0  05EE  1265          MOVL    BOO$GL_CONDEV,ACF$L_DEVNAME(R10); Set pointer to device name
                                05F6  1266
                                05F6  1267 ; Now try to get driver name from DDB if it exists and load BOO$GQ_CONSYSID
                                05F6  1268 ; if HSC device.
                                05F6  1269
      7E  0000242C'EF           3C  05F6  1270          MOVZWL  L^BOO$GL_CONCUNIT,-(SP) ; Unit number
                            50    DD  05FD  1271          PUSHL   R0                      ; Device name
                0000035D'EF       16  05FF  1272          JSB     SGN$GET_DEVICE_LOCK_IODB; Get device data base addresses
                   5E       08    C0  0605  1273          ADDL2   #8,SP                   ; Pop off input parameters
                      08 50       E8  0608  1274          BLBS    R0,20$                  ; All okay
      50  007C9010 8F           D0  060B  1275          MOVL    #SYSG$_NODEV,R0         ; Set error code - "Device not known"
                            04    12  0612  1276          RET                             ; Leave
                                0613  1277
                            05    E2  0613  1278 20$:     BBSS    #ACF$V_GETDONE,-
                  00 00002458'EF      0615  1279                  L^BOO$GL_CONFLAGS,21$   ; Notify LOADER that GET was done
                                061B  1280
      18 AA  00002444'EF        D0  061B  1281 21$:     MOVL    BOO$GL_CONDRV,ACF$L_DRVNAME(R10); And driver name
                            31    14  0623  1282          BGTR    30$                     ; Branch if driver specified
      51  00002400'EF           D0  0625  1283          MOVL    ACF$GL_DDB,R1           ; DDB address
                            07    13  062C  1284          BEQL    25$                     ; Branch if none
                   18 AA    24 A1   DE  062E  1285          MOVAL   DDB$T_DRVNAME(R1),ACF$L_DRVNAME(R10) ; Address from DDB
```

```
                                 21   11  0633  1286          BRB     30$                         ; Branch around name hackery
                                              0635  1287
                     56  0000245C'EF   D0  0635  1288 25$:     MOVL    L^BOO$GL_NEXTSTR,R6         ; Get address of next free space
                      18 AA    56   D0  063C  1289           MOVL    R6,ACF$L_DRVNAME(R10)       ; Set as driver name address
                         86    08   90  0640  1290           MOVB    #8,(R6)+                    ; Set count for string
                66 52455649 52442020 8F  7D  0643  1291           MOVQ    #^A/  DRIVER/,(R6)          ; Set driver suffix
                      51   14 AA   D0  064E  1292           MOVL    ACF$L_DEVNAME(R10),R1       ; Pointer to device name
                      66   01 A1   B0  0652  1293           MOVW    1(R1),(R6)                  ; Form default driver name
                                              0656  1294
                   0A AA  0000243C'EF   90  0656  1295 30$:     MOVB    BOO$GL_CONAUNIT,ACF$B_AUNIT(R10); Set adapter unit
                   21 AA  00002430'EF   90  065E  1296           MOVB    L^BOO$GL_CONNUMU,ACF$B_NUMUNIT(R10)
                                              0666  1297                                       ; Store number of units to configure
                   0B AA  00002458'EF   90  0666  1298           MOVB    BOO$GL_CONFLAGS,ACF$B_AFLAG(R10)  ; Store flags
            0000241C'EF   00002434'EF   A1  066E  1299           ADDW3   BOO$GL_CONVECT,BOO$GL_COMBO_VECTOR_OFFSET,-;
                         10 AA          0679  1300                   ACF$W_CVECTOR(R10)          ; Set vector address
            50  0000241C'EF    08   02  EF  067B  1301           EXTZV   #2,#8,BOO$GL_COMBO_VECTOR_OFFSET,R0; Save vector offset in longwords
                      1F AA   50   90  0684  1302           MOVB    R0,ACF$B_COMBO_VECTOR_OFFSET(R10);
                                              0688  1303 ;
                                              0688  1304 ; Set up ACF$L_CONTRLREG - can either be UNIBUS CSR or address of CI
                                              0688  1305 ;                           System id.
                                              0688  1306 ;
                      0000244C'EF   D5  0688  1307           TSTL    BOO$GQ_CONSYSID             ; See if SYSIDLOW was specified
                                0A  13  068E  1308           BEQL    40$                         ; Branch if not
                   0C AA  0000244C'EF   9E  0690  1309           MOVAB   BOO$GQ_CONSYSID, -
                                              0698  1310                   ACF$L_CONTRLREG(R10)        ; Set system id address
                                 22   11  0698  1311           BRB     50$                         ; Branch
                                              069A  1312 ;
                                              069A  1313 ; Calculate system virtual address of UNIBUS CSR
                                              069A  1314 ;
00002428'EF   00001000 8F   C1  069A  1315 40$:     ADDL3   #UBA_IOBASE, -
                         0C AA          06A5  1316                   BOO$GL_CONCREG, -
                                              06A7  1317                   ACF$L_CONTRLREG(R10)        ; control register address
                   0C AA   04 AA   C0  06A7  1318           ADDL    ACF$L_CONFIGREG(R10), -
                                              06AC  1319                   ACF$L_CONTRLREG(R10)        ; Add adapter va base
                      00002420'EF   C0  06AC  1320           ADDL    BOO$GL_COMBO_CSR_OFFSET,-; Add offset to get true CSR address
                         0C AA          06B2  1321                   ACF$L_CONTRLREG(R10)        ;
                      00002420'EF   8E  06B4  1322           MNEGB   BOO$GL_COMBO_CSR_OFFSET,-; Calculate offset back to CSR start
                         20 AA          06BA  1323                   ACF$B_COMBO_CSR_OFFSET(R10); Save offset
                                              06BC  1324
                12 AA  0000242C'EF   B0  06BC  1325 50$:     MOVW    BOO$GL_CONCUNIT, -
                                              06C4  1326                   ACF$W_CUNIT(R10)            ; Set controller unit number
                1C AA  00002448'EF   B0  06C4  1327           MOVW    BOO$GL_CONUNITS, -
                                              06CC  1328                   ACF$W_MAXUNITS(R10)         ; Set maximum units
                1E AA  00002438'EF   90  06CC  1329           MOVB    BOO$GL_CONNUMV, -
                                              06D4  1330                   ACF$B_CNUMVEC(R10)          ; Set count of vectors
                            50   01   D0  06D4  1331 55$:     MOVL    #1,R0                       ; Set success
                                 04   04  06D7  1332           RET                                 ;
                                              06D8  1333
                                              06D8  1334                   .DSABL LSB
                                              06D8  1335
```

```
                     06D8  1337          .SBTTL  BOO$LOAD - Load a driver or misc code if not already loaded
                     06D8  1338 ;
                     06D8  1339 ; BOO$LOAD -    Loads the driver or misc code if not already loaded.
                     06D8  1340 ;
                0FFC 06D8  1341 .Entry  BOO$LOAD, ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
                     06DA  1342
            52   D4  06DA  1343          CLRL    R2                      ; Clear reload flag
            05   11  06DC  1344          BRB     LOADRV                  ; And merge with common code
```

```
                                06DE  1346          .SBTTL  BOO$RELOAD - Reload a specified driver
                                06DE  1347 ;
                                06DE  1348 ; BOO$RELOAD -  Reloads the specified driver replacing any existing copy
                                06DE  1349 ;               unless there are busy units requiring the driver that would
                                06DE  1350 ;               be replaced.
                                06DE  1351 ;
                          0FFC  06DE  1352 .Entry  BOO$RELOAD, ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;
                                06E0  1353
                    52    01  90  06E0  1354          MOVB    #ACF$M_RELOAD,R2        ; Set flag to force reload
                                06E3  1355 LOADRV:
                                06E3  1356 ;
                                06E3  1357 ; The first block of the file will be read to determine if it is a driver or
                                06E3  1358 ; misc code by looking at the type field.
                                06E3  1359 ;
          00002444'EF           DD  06E3  1360          PUSHL   BOO$GL_CONDRV           ; File name string
                     6E         D6  06E9  1361          INCL    (SP)                    ; Get past the count
    7E    00002444'FF           9A  06EB  1362          MOVZBL  @BOO$GL_CONDRV,-(SP)    ; Length of file name
                57    5E        D0  06F2  1363          MOVL    SP,R7                   ; Address of desriptor for file name
                       F908'    30  06F5  1364          BSBW    BOO$EXEOPEN             ; Open the file (default SYS$SYSTEM:.EXE)
                     5E 50      E9  06F8  1365          BLBC    R0,40$                  ; Error
                     5E    08   C0  06FB  1366          ADDL    #8,SP                   ; Clean up stack
    56    00002200'EF           9E  06FE  1367          MOVAB   BOO$AB_LOADBUF,R6       ; Buffer for file read
                58    02        D0  0705  1368          MOVL    #2,R8                   ; First block after image header
                59    01        D0  0708  1369          MOVL    #1,R9                   ; One page
                       F8F2'    30  070B  1370          BSBW    BOO$READFILE
                     48 50      E9  070E  1371          BLBC    R0,40$                  ; Error
                       F8EC'    30  0711  1372          BSBW    BOO$FILCLOSE            ; Close the currently open file
                     42 50      E9  0714  1373          BLBC    R0,40$                  ; Error
          0000220A'EF           91  0717  1374          CMPB    BOO$AB_LOADBUF+SLV$B_TYPE,-
                     62 8F          071D  1375                  #DYN$C_LOADCODE         ; Check for misc code
                        3C     13  071F  1376          BEQL    LOADCODE
                       F8DC'    30  0721  1377          BSBW    BOO$LOCK_GEN            ; Lock SYSGEN database
                     32 50      E9  0724  1378          BLBC    R0,40$                  ; If lbc, didn't get lock
      5A    0000248C'EF           9E  0727  1379          MOVAB   L^BOO$AL_ACF,R10        ; Get base address for ACF block
    18 AA   00002444'EF           D0  072E  1380          MOVL    BOO$GL_CONDRV,ACF$L_DRVNAME(R10)        ;
             0B AA    52          90  0736  1381          MOVB    R2,ACF$B_AFLAG(R10)     ; Set flags for load or reload
                   14 AA          D4  073A  1382          CLRL    ACF$L_DEVNAME(R10)      ; No device name
          00000000'EF    6A     FA  073D  1383          CALLG   (R10),L^IOGEN$LOADER    ; Load requested driver
                     03 50      E8  0744  1384          BLBS    R0,20$                  ; Continue if no error
                       F8B6'    30  0747  1385          BSBW    PUTERROR                ; Give error message
                        50      DD  074A  1386 20$:     PUSHL   R0                      ; Save status
                       F8B1'    30  074C  1387          BSBW    BOO$UNLOCK_GEN          ; Unlock SYSGEN database
                     03 50      E8  074F  1388          BLBS    R0,30$                  ; If no error, continue
                       F8AB'    30  0752  1389          BSBW    PUTERROR                ; Give error message
                        50 8ED0     0755  1390 30$:     POPL    R0                      ; Restore status
                           04       0758  1391          RET                             ; Exit
                       F8A4'    30  0759  1392 40$:     BSBW    PUTERROR                ; Give error message
                           04       075C  1393          RET                             ; Exit
                                075D  1394 ;
                                075D  1395 LOADCODE:
          00002444'EF           DD  075D  1396          PUSHL   BOO$GL_CONDRV           ; File name string
                     6E         D6  0763  1397          INCL    (SP)                    ; Get past the count
    7E    00002444'FF           9A  0765  1398          MOVZBL  @BOO$GL_CONDRV,-(SP)    ; Length of file name
                57    5E        D0  076C  1399          MOVL    SP,R7                   ; Address of desriptor for file name
                       F88E'    30  076F  1400          BSBW    BOO$UFOOPEN             ; Open the file for user access (default SYS
                     22 50      E9  0772  1401          BLBC    R0,10$                  ; Error
                     5E    08   C0  0775  1402          ADDL    #8,SP                   ; Clean up stack
```

```
                                                E 8
00002200'EF   9F   0778  1403          PUSHAB  BOO$AB_LOADBUF           ; Use code buffer for return address array
        51    DD   077E  1404          PUSHL   R1                      ; Channel
        02    DD   0780  1405          PUSHL   #2                      ; Arg count
     50 5E    D0   0782  1406          MOVL    SP,R0
                   0785  1407          $CMKRNL_S ROUTIN = EXE$LOAD_CODE,-
                   0785  1408                    ARGLST = (R0)
     04 50    E8   0794  1409          BLBS    R0,20$
      F866'   30   0797  1410 10$:     BSBW    PUTERROR
              04   079A  1411          RET
                   079B  1412
                   079B  1413 20$      $CMKRNL_S ROUTIN = LINK_CODE
     EA 50    E9   07AA  1414          BLBC    R0,10$
              04   07AD  1415          RET
                   07AE  1416 ;
                   07AE  1417 LINK_CODE:
              001C 07AE  1418          .WORD   ^M<R2,R3,R4>
52 00002200'GF   D0   07B0  1419          MOVL    G^BOO$AB_LOADBUF,R2     ; Address of loaded code
   54    52   D0   07B7  1420          MOVL    R2,R4                   ; Save address of loaded code
   53 10 A4   D0   07BA  1421          MOVL    SLV$A_SYSVECS(R4),R3    ; Get address of vectors in SYS.EXE
00000000'GF   16   07BE  1422          JSB     G^EXE$LINK_VEC          ; Connect vectors to loaded routines.
      10 50   E9   07C4  1423          BLBC    R0,10$                  ; Leave on error
5C 000025BC'GF   DE   07C7  1424          MOVAL   G^BOO$GL_LOAD_ARGS,AP   ; Argument list for initialization routine
   50 04 A4   D0   07CE  1425          MOVL    SLV$L_INITRTN(R4),R0    ; Possible initialization routine
         03   13   07D2  1426          BEQL    10$                     ; None, leave
       6044   16   07D4  1427          JSB     (R0)[R4]                ; Call it
              04   07D7  1428 10$:     RET
                   07D8  1429
```

```
                        07D8  1431          .SBTTL  BOO$GIVEHELP - Print Help information
                        07D8  1432 ;
                        07D8  1433 ;       Print Help Information
                        07D8  1434 ;
                 003C   07D8  1435 .Entry  BOO$GIVEHELP, ^M<R2,R3,R4,R5>   ;
                        07DA  1436
     00000000'GF   9F   07D'  1437          PUSHAB  G^LIB$GET_INPUT          ; Input routine
     00002574'EF   9F   07E0  1438          PUSHAB  HELP_FLAG                ; Flags
     00002559'EF   9F   07E6  1439          PUSHAB  L^HELP_FILE              ; Library
           08 AC   B0   07EC  1440          MOVW    TPA$L_STRINGCNT(AP),-
     00002578'EF        07EF  1441                  HELP_DESC                ; Set length
           0C AC   D0   07F4  1442          MOVL    TPA$L_STRINGPTR(AP),-
     0000257C'EF        07F7  1443                  HELP_DESC+4              ; Set address
     00002578'EF   9F   07FC  1444          PUSHAB  HELP_DESC                ; Input string
              7E   D4   0802  1445          CLRL    -(SP)                    ; Width
     00000000'GF   9F   0804  1446          PUSHAB  G^LIB$PUT_OUTPUT         ; Output routine
00000000'GF    06   FB   0808  1447          CALLS   #6,G^LBR$OUTPUT_HELP     ; Call help routine
                        0811  1448
                 04   0811  1449          RET                              ; Return with status
                        0812  1450
                        0812  1451          .END
```

```
$$T1                            = 00000001
$$T2                            = 00000005
$CLI.                           = 00002464 R      04
$CLI..                          = 00002480 R      04
ACF$B_AFLAG                     = 0000000B
ACF$B_AUNIT                     = 0000000A
ACF$B_CNUMVEC                   = 0000001E
ACF$B_COMBO_CSR_OFFSET          = 00000020
ACF$B_COMBO_VECTOR_OFFSET       = 0000001F
ACF$B_NUMUNIT                   = 00000021
ACF$C_LENGTH                    = 00000028
ACF$GL_CRB                        0000240C RG     04
ACF$GL_DDB                        00002400 RG     04
ACF$GL_DPT                        00002414 RG     04
ACF$GL_IDB                        00002408 RG     04
ACF$GL_LASTDDB                    00002410 RG     04
ACF$GL_SB                         00002418 RG     04
ACF$GL_UCB                        00002404 RG     04
ACF$L_ADAPTER                   = 00000000
ACF$L_CONFIGREG                 = 00000004
ACF$L_CONTRLREG                 = 0000000C
ACF$L_DEVNAME                   = 00000014
ACF$L_DRVNAME                   = 00000018
ACF$M_RELOAD                    = 00000001
ACF$V_CRBBLT                    = 00000001
ACF$V_GETDONE                   = 00000005
ACF$V_NOLOAD_DB                 = 00000003
ACF$W_AVECTOR                   = 00000008
ACF$W_CUNIT                     = 00000012
ACF$W_CVECTOR                   = 00000010
ACF$W_MAXUNITS                  = 0000001C
ADP$L_AVECTOR                   = 0000001C
ADP$L_CSR                       = 00000000
ADP$L_LINK                      = 00000004
ADP$W_TR                        = 0000000C
AUTOLOG                           0000028F RG     06
BOO$AB_LOADBUF                    00002200 R      04
BOO$AB_PATCH                      00000000 RG     04
BOO$AB_PRMBUF                     00000200 RG     04
BOO$AL_ACF                        0000248C RG     04
BOO$AL_CLIBLK                     00002464 RG     04
BOO$A_PRMBLK                      ******** X      06
BOO$CONADP                        0000031F RG     05
BOO$CONAUNIT                      00000372 RG     05
BOO$CONCNUM                       00000367 RG     05
BOO$CONCREG                       0000034A RG     05
BOO$CONCSROFFSET                  0000033F RG     05
BOO$CONCVEC                       00000357 RG     05
BOO$CONDRVNAM                     0000037D RG     05
BOO$CONFIGALL                     00000089 RG     06
BOO$CONFIGONE                     000000F8 RG     06
BOO$CONNECT                       000004AB RG     05
BOO$CONNLADP                      0000032A RG     05
BOO$CONRESET                      00000200 RG     05
BOO$CONSOLE                       00000464 RG     05
BOO$CONSYSID_HIGH                 00000459 RG     05
BOO$CONSYSID_LOW                  0000044E RG     05
BOO$CONUNITS                      00000443 RG     05
BOO$CONVECOFFSET                  00000334 RG     05
BOO$DEVNAME                       000003A4 RG     05
BOO$EXEOPEN                       ********   X    05
BOO$FILCLOSE                      ********   X    05
BOO$FILOPEN                       ********   X    05
BOO$GB_FILELEN                    000024FD RG     04
BOO$GIVEHELP                      000007D8 RG     05
BOO$GL_CMDOPT                     ********     X  06
BOO$GL_COMBO_CSR_OFFSET           00002420 RG     04
BOO$GL_COMBO_VECTOR_OFFSET        0000241C RG     04
BOO$GL_CONADP                     00002424 RG     04
BOO$GL_CONAUNIT                   0000243C RG     04
BOO$GL_CONCRB                     00002454 RG     04
BOO$GL_CONCREG                    00002428 RG     04
BOO$GL_CONCUNIT                   0000242C RG     04
BOO$GL_CONDEV                     00002440 RG     04
BOO$GL_CONDRV                     00002444 RG     04
BOO$GL_CONFLAGS                   00002458 RG     04
BOO$GL_CONNUMU                    00002430 RG     04
BOO$GL_CONNUMV                    00002438 RG     04
BOO$GL_CONUNITS                   00002448 RG     04
BOO$GL_CONVECT                    00002434 RG     04
BOO$GL_DOT                        ********   X    05
BOO$GL_FILEADDR                   000024F9 RG     04
BOO$GL_LOAD_ARGS                  000025BC RG     04
BOO$GL_NEXTSTR                    0000245C RG     04
BOO$GL_PARINUSE                   000024FE RG     04
BOO$GL_RETSAVE                    000024C4 RG     04
BOO$GL_SELECT                     00002460 RG     04
BOO$GQ_CMDESC                   = 0000246C RG     04
BOO$GQ_CONSYSID                   0000244C RG     04
BOO$GQ_LIMITS                     000024B4 RG     04
BOO$GQ_RETADR                     000024BC RG     04
BOO$GT_ACTIVE                     0000250A RG     04
BOO$GT_CURRENT                    00002502 RG     04
BOO$GT_CVNAME                     000024DE RG     04
BOO$GT_DDNAME                     000024F0 RG     04
BOO$GT_DEFAULT                    00002511 RG     04
BOO$GT_DXNAME                     000024E7 RG     04
BOO$GT_FILE                       00002519 RG     04
BOO$GT_OPNAME                     000024DA RG     04
BOO$GT_PROMPT                     00002480 RG     04
BOO$GT_SAVE_DEVNAME               00002614 R      04
BOO$GT_SYSPARNAME                 ********   X    05
BOO$HICIM                         00000000 RG     03
BOO$LOAD                          000006D8 RG     05
BOO$LOCK_GEN                      ********     X  06
BOO$LOLIM                         00000000 RG     02
BOO$MAKLIST                       000002EE RG     05
BOO$MSCP_ARG                      000002DB RG     05
BOO$MSCP_RESET                    0000029A RG     05
BOO$READFILE                      ********   X    05
BOO$RELOAD                        000006DE RG     05
BOO$RESETLIST                     00000262 RG     05
BOO$SEARCH                        ********     X  05
BOO$SENDOPER                      00000140 RG     05
```

```
BOO$SETASCII                    ********    X   05
BOO$SETVALUE                    ********    X   05
BOO$UFOOPEN                     ********    X   05
BOO$UNLOCK_GEN                  ********    X   06
BOO$USEACT                      000000E9 RG     05
BOO$USEFILE                     00000000 RG     05
BOO$WRTACT                      00000000 RG     06
BOO$WRTCUR                      00000108 RG     05
BOO$WRTSYSPARFILE               ********    X   05
BOOCMD$V_AUTOLOG              = 0000000C
BOOCMD$V_EXCLUDE              = 00000007
BOOCMD$V_SELECT               = 00000006
CLI$B_RQTYPE                  = 00000000
CLI$C_REQDESC                 = 0000001C
CLI$K_GETCMD                  = 00000001
CLI$W_RQSIZE                  = 00000008
CONFIGADP                       0000018C RG     06
CONFIG_EXIT                     000000E6 R      06
CONNECT                         000005AB RG     05
CONNLADP                        00000579 RG     05
CONN_ADAP                       00000536 RG     05
CONSNAME                        000024D6 R      04
CR                            = 0000000D
CRB$L_INTD                    = 00000024
CTL$GL_PCB                      ********    X   06
CTRSTR_AUTOLOG                  000025F1 R      04
CTRSTR_AUTOLOG_UNIT             00002600 R      04
DDB$L_DCB                     = 00000004
DDB$T_DRVNAME                 = 00000024
DONE                            000000A9 R      05
DYN$C_LOADCODE                = 00000062
EXE$A_SYSPARAM                  ********    X   05
EXE$C_SYSPARSZ                  ********    X   05
EXE$GL_DEFFLAGS                 ********    X   06
EXE$GL_DYNAMIC_FLAGS            ********    X   05
EXE$GL_FLAGS                    ********    X   06
EXE$GL_SCB                      ********    X   05
EXE$GT_STARTUP                  ********    X   05
EXE$LINK_VEC                    ********    X   05
EXE$LOAD_CODE                   ********    X   05
EXE$V_NOAUTOCNF                 ********    X   06
EXE$V_WRITESYSPARAMS            ********    X   05
FACNAME                         000024D0 R      04
FACNAMED                        000024C8 RG     04
FACNAMSZ                      = 00000006
FF                            = 0000000C
FULL_NAME_PTR                   00002588 RG     04
HELP_DESC                       00002578 R      04
HELP_FILE                       00002559 R      04
HELP_FLAG                       00002574 R      04
HLP$A_PROMPT                  = 00000001
IDB$L_ADP                     = 00000014
IOC$AUTOCONFIG                  ********    X   06
IOC$AUTORESET                   ********    X   06
IOC$GL_ADPLIST                  ********    X   06
IOC$SEARCHALL                   ********    X   06
IOGEN$CONSOLE                   ********    X   05
IOGEN$LOADER                    ********    X   06
IPL$_SCHED                    = 00000003
JPI$_PID                      = 00000319
LBR$OUTPUT_HELP                 ********    X   05
LF                            = 0000000A
LIB$GET_INPUT                   ********    X   05
LIB$PUT_OUTPUT                  ********    X   05
LINK_CODE                       000007AE R      05
LOADCODE                        0000075D R      05
LOADRV                          000006E3 R      05
LOCADP                          00000163 R      06
MMG$A_SYSPARAM                  ********    X   05
MSCP_ARG_LIST                   0000258C R      04
MSCP_ARG_LIST_SIZE            = 00000030
MSCP_NAME                       000025EC R      04
NAM$B_RSL                       ********    X   05
NAM$L_RSA                       ********    X   05
NEXTADP                         0000013B R      06
OPC$L_MS_TEXT                 = 00000008
OPC$M_NM_CENTRL               = 00000001
OPC$_RQ_RQST                  = 00000003
OPERGETJPI                      00002694 R      04
OPERMSG                         000026C0 R      04
OPERMSGBUF                      000026C8 R      04
OPERMSGFAO                      000026AC R      04
OPERMSGID                       000026A8 R      04
OPERMSGNAM                      000026B4 R      04
OPERMSGPID                      000026B0 R      04
OPERMSGTXT                      000026D0 R      04
OPERMSGVEC                      000026A4 R      04
OPERNAMDESC                     000026B8 R      04
OUTBUF                          00002628 R      04
OUTBUF_STR                      00002630 R      04
OUTLEN                          00002610 R      04
OUTLEN_UNIT                     0000260C R      04
PR$_IPL                       = 00000012
PRM$B_POS                     = 00000015
PRM$B_SIZE                    = 00000014
PRM$C_LENGTH                  = 00000032
PRM$L_ADDR                    = 00000000
PRM$L_FLAGS                   = 00000010
PRM$M_DYNFLAGS                  ********    X   06
PRM$V_ASCII                   = 00000010
PRM$V_DYNAMIC                 = 00000000
PUTERROR                        ********    X   06
RIO$AB_BUFFER                   ********    X   06
RIO$GW_OUTLEN                   ********    X   06
RIO$OUTPUT_LINE                 ********    X   06
RIO_INPNAM                      ********    X   05
SAVE_DOT                        00002584 R      04
SB$B_SYSTEMID                 = 00000018
SCH$IOLOCKR                     ********    X   06
SCH$IOUNLOCK                    ********    X   06
SCS$GA_LOCALSB                  ********    X   06
SCS$GL_CDL                      ********    X   05
SCS$GL_MSCP                     ********    X   05
SELECT                          0000025A R      06
```

```
SGN$GET_DEVICE                  0000034B RG    06
SGN$GET_DEVICE_LOCK_IODB        0000035D R     06
SLV$A_SYSVECS                 = 00000010
SLV$B_TYPE                    = 0000000A
SLV$L_INITRTN                 = 00000004
SS$_DEVACTIVE                 = 000002C4
SS$_DEVOFFLINE                = 00000084
SS$_NOPRIV                    = 00000024
SS$_NORMAL                    = 00000001
SS$_NOSUCHDEV                 = 00000908
SYS$CMEXEC                      ********  GX   06
SYS$CMKRNL                      ********  GX   06
SYS$FAO                         ********   X   06
SYS$GETJPI                      ********  GX   05
SYS$PUTMSG                      ********  GX   05
SYS$SNDOPR                      ********  GX   05
SYSG$_CONFQUAL                = 007C808A
SYSG$_INVADAP                 = 007C80BA
SYSG$_NOADAPTER               = 007C80D2
SYSG$_NOAUTOCNF               = 007C8002
SYSG$_NODEV                   = 007C9010
SYSG$_NOTPARAM                = 007C80EA
SYSG$_WRITEACT                = 007CA013
SYSG$_WRITECUR                = 007CA01B
TPA$L_NUMBER                  = 0000001C
TPA$L_PARAM                   = 00000020
TPA$L_STRINGCNT               = 00000008
TPA$L_STRINGPTR               = 0000000C
TPA$L_TOKENCNT                = 00000010
TPA$L_TOKENPTR                = 00000014
UBA_IOBASE                    = 00001000
UCB$L_CRB                     = 00000024
UCB$L_LINK                    = 00000030
UCB$W_UNIT                    = 00000054
VALID_PAR_FILE                  00002580 R     04
VEC$L_IDB                     = 00000008
```

+-----------------+
! 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	000027D0	(10192.)	04	(4.)	NOPIC	USR	CON	REL	LCL	NOSHR	NOEXE	RD	WRT	NOVEC QUAD
PAGED_CODE	00000812	(2066.)	05	(5.)	NOPIC	USR	CON	REL	LCL	NOSHR	EXE	RD	NOWRT	NOVEC LONG
NONPAGED_CODE	00000423	(1059.)	06	(6.)	NOPIC	USR	CON	REL	LCL	NOSHR	EXE	RD	NOWRT	NOVEC LONG

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	37	00:00:00.09	00:00:00.49
Command processing	138	00:00:00.74	00:00:03.19
Pass 1	573	00:00:24.19	00:00:47.20
Symbol table sort	0	00:00:03.72	00:00:06.05
Pass 2	296	00:00:05.60	00:00:13.47
Symbol table output	31	00:00:00.26	00:00:00.27
Psect synops's output	4	00:00:00.04	00:00:00.05
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	1081	00:00:34.65	00:01:10.73

The working set limit was 2000 pages.
137940 bytes (270 pages) of virtual memory were used to buffer the intermediate code.
There were 130 pages of symbol table space allocated to hold 2287 non-local and 103 local symbols.
1451 source lines were read in Pass 1, producing 130 object records in Pass 2.
50 pages of virtual memory were used to define 47 macros.

Macro library statistics

Macro library name	Macros defined
_255DUA28:[BOOTS.OBJ]BOOTS.MLB;1	1
_255DUA28:[SYS.OBJ]LIB.MLB;1	18
_255DUA28:[SYSLIB]STARLET.MLB;2	25
TOTALS (all libraries)	44

2422 GETS were required to define 44 macros.

There were no errors, warnings or information messages.

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

SYSGENMN
LIS

VMB
LIS

SYSGEN
LIS

SYSGTTSTR
LIS

SYSGENCMD
LIS

SYSGTERM
LIS

TS8BOOTIO
LIS