```
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    CCCCCCCC  MM      MM  DDDDDDDD
  SSSSSSSS  YY      YY    SSSSSSSS    GGGGGGGG  EEEEEEEEEE  NN      NN    CCCCCCCC  MM      MM  DDDDDDDD
SS          YY      YY  SS          GG          EE          NN      NN  CC          MMMM  MMMM  DD      DD
SS          YY      YY  SS          GG          EE          NN      NN  CC          MMMM  MMMM  DD      DD
SS            YY  YY    SS          GG          EE          NNNN    NN  CC          MM  MM  MM  DD      DD
SS            YY  YY    SS          GG          EE          NNNN    NN  CC          MM  MM  MM  DD      DD
  SSSSSS        YY        SSSSSS    GG          EEEEEEEE    NN  NN  NN  CC          MM      MM  DD      DD
  SSSSSS        YY        SSSSSS    GG          EEEEEEEE    NN  NN  NN  CC          MM      MM  DD      DD
        SS      YY              SS  GG  GGGGGG  EE          NN    NNNN  CC          MM      MM  DD      DD
        SS      YY              SS  GG  GGGGGG  EE          NN    NNNN  CC          MM      MM  DD      DD
        SS      YY              SS  GG      GG  EE          NN      NN  CC          MM      MM  DD      DD
        SS      YY              SS  GG      GG  EE          NN      NN  CC          MM      MM  DD      DD
SSSSSSSS        YY      SSSSSSSS      GGGGGG    EEEEEEEEEE  NN      NN    CCCCCCCC  MM      MM  DDDDDDDD
SSSSSSSS        YY      SSSSSSSS      GGGGGG    EEEEEEEEEE  NN      NN    CCCCCCCC  MM      MM  DDDDDDDD

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

Table of contents

```
00000001  0000          CMDSW=1                 ; SET SWITCH TO GENERATE SYSGEN
          0000          .IF     NDF,CMDSW
          0000          .TITLE  SYSBOOCMD - Command parsing for SYSBOOT
          0000          .IFF
          0000          .TITLE  SYSGENCMD - Command parsing for SYSGEN
          0000          .ENDC
          0000          .IDENT  'V04-000'
          0000          .DEFAULT DISPLACEMENT,LONG
          0000  ;
          0000  ;****************************************************************************
          0000  ;*                                                                          *
          0000  ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
          0000  ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
          0000  ;*  ALL RIGHTS RESERVED.                                                    *
          0000  ;*                                                                          *
          0000  ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
          0000  ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
          0000  ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
          0000  ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
          0000  ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
          0000  ;*  TRANSFERRED.                                                            *
          0000  ;*                                                                          *
          0000  ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
          0000  ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
          0000  ;*  CORPORATION.                                                            *
          0000  ;*                                                                          *
          0000  ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
          0000  ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
          0000  ;*                                                                          *
          0000  ;*                                                                          *
          0000  ;****************************************************************************
          0000
          0000  ;++
          0000  ;
          0000  ; Facility:  System generation and initialization
          0000  ;
          0000  ; Abstract: SYSBOOCMD is the interpreter for parameter modification
          0000  ;           commands both at bootstrap time and as part of the sysgen utility
          0000  ;           SYSGEN.
          0000  ;
          0000  ; Environment:
          0000  ;
          0000  ;       Both SYSGEN and SYSBOOT environments.
          0000  ;
          0000  ;       *********************************
          0000  ;
          0000  ;       WARNING: SYSBOOT code must be PIC
          0000  ;
          0000  ;       *********************************
          0000  ;
          0000  ; Author:  RICHARD I. HUSTVEDT, Creation date:  4-MAY-1978
          0000  ;
          0000  ; Modified by:
          0000  ;
          0000  ;       V03-027 WHM0012         Bill Matthews           02-aug-1984
          0000  ;               Fix bad movc instruction in BOO$SETSTART from V03-026.
          0000  ;
```

```
0000     57 ;       V03-026 WHM00.1         Bill Matthews           01-aug-1984
0000     58 ;               Fix SET/STARTUP bug from V03-024.
0000     59 ;
0000     60 ;       V03-025 WHM0010         Bill Matthews           23-Jul-1984
0000     61 ;               Change MSCP qualifier /SMALL to /MINIMUM and /FRACTION to
0000     62 ;               /MAXIMUM.
0000     63 ;
0000     64 ;       V03-024 WHM0009         Bill Matthews           19-Jun-1,84
0000     65 ;               Fixed LOAD<CR> accvio. Fixed SET ascii-parameter 0 bug.
0000     66 ;               Now allow an optional : and = in SET/STARTUP filespec.
0000     67 ;               Now allow optional : in device name of the CONNECT command.
0000     68 ;
0000     69 ;       V03-023 WHM0008         Bill Matthews           20-Apr-1984
0000     70 ;               Fixed SET of an ascii parameter to DEFAULT bug.
0000     71 ;               Removed USE CURRENT that read SYSGEN parameters from SYS.EXE.
0000     72 ;
0000     73 ;       V03-022 WHM0007         Bill Matthews           11-Apr-1984
0000     74 ;               Removed the QUORUM command.
0000     75 ;
0000     76 ;       V03-021 WHM0006         Bill Matthews           04-Apr-1984
0000     77 ;               Added support for sysgen ascii parameters longer than 4
0000     78 ;               characters.
0000     79 ;               Added support for a seperate default system parameter file.
0000     80 ;
0000     81 ;       V03-020 JLV0342         Jake VanNoy              3-APR-1984
0000     82 ;               Add TERMINAL/ECHO command.
0000     83 ;
0000     84 ;       V03-019 WHM0005         Bill Matthews           14-Mar-1984
0000     85 ;               Conditionally assembled TPARSE tables for SYSBOOT.
0000     86 ;               Change ascii input specifier from %A to '"'.
0000     87 ;               Output header for display of a single parameter value.
0000     88 ;
0000     89 ;       V03-018 WHM0004         Bill Matthews           13-Mar-1984
0000     90 ;               Move definition of BOO$GL_LOAD_ARGS from this module
0000     91 ;               to SYSGEN.MAR.
0000     92 ;
0000     93 ;       V03-017 WHM0003         Bill Matthews           23-Feb-1984
0000     94 ;               Add support for loading and starting the MSCP server.
0000     95 ;
0000     96 ;       V03-016 WHM0002         Bill Matthews           01-Feb-1984
0000     97 ;               Add support for SHOW/LGI.
0000     98 ;
0000     99 ;       V03-015 ACG0392         Andrew C. Goldstein,    19-Jan-1984  22:40
0000    100 ;               Tie off SYS$FILESCAN for TPARSE use
0000    101 ;
0000    102 ;       V03-014 WHM0001         Bill Matthews           14-Dec-1983
0000    103 ;               Add /REMOTE and /LOGICAL switches to the CONNECT CONSOLE command
0000    104 ;               Add /VECTOR_OFFSET and /CSR_OFFSET to the CONNECT command
0000    105 ;
0000    106 ;       V03-013 WMC0013         Wayne Cardoza           01-Dec-1983
0000    107 ;               Allow arbitrary ordering of install qualifiers
0000    108 ;
0000    109 ;       V03-012 JLV0311         Jake VanNoy             10-OCT-1983
0000    110 ;               Fix SHOW/ALL to really SHOW/ALL.
0000    111 ;
0000    112 ;       V03-011 SRB0103         Steve Beckhardt         .9-Sep-1983
0000    113 ;               Added temporary QUORUM command.
```

```
0000   114 ;
0000   115 ;       V03-010 BLS0239         Benn Schreiber          13-Sep-1983
0000   116 ;               Use TPA$_SYMBOL for %A so that '$' and '_' are allowed.
0000   117 ;
0000   118 ;       V03-009 ACG0345         Andrew C. Goldstein,    1-Aug-1983  16:53
0000   119 ;               Add dummy SYS$ASCTOID routine for TPARSE
0000   120 ;
0000   121 ;       V03-008 MSH0005         Maryann Hinden          13-Jul-1983
0000   122 ;               Don't need to echo input anymore.
0000   123 ;               Set ascii parameters correctly if smaller than a longword.
0000   124 ;
0000   125 ;       V03-007 MSH0004         Maryann Hinde           24-Jun-1983
0000   126 ;               Change $BOODEF to $BOOCMDDEF.
0000   127 ;
0000   128 ;       V03-006 MSH0003         Maryann Hinden          10-Jun-1983
0000   129 ;               Use $BOODEF.
0000   130 ;
0000   131 ;       V03-005 MSH0002         Maryann Hinden          14-Apr-1983
0000   132 ;               Teach SYSGEN to speak ASCII.
0000   133 ;
0000   134 ;       V03-004 MSH0001         Maryann Hinden          24-Mar-1983
0000   135 ;               Preserve values for system time and base registers
0000   136 ;               across USE DEFAULT and USE CURRENT commands.
0000   137 ;
0000   138 ;       V03-003 DWT0086         David W. Thiel          22-Mar-1983
0000   139 ;               Add PRM$M_CLUSTER to SHOW/ALL mask.  Add
0000   140 ;               SHOW/CLUSTER.
0000   141 ;
0000   142 ;       V03-002 WMC0001         Wayne Cardoza           12-Aug-1982
0000   143 ;               Add support for the /checkpoint qualifier on install /page
0000   144 ;
0000   145 ;       V03-001 JLV0196         Jake VanNoy             17-MAR-1982
0000   146 ;               Add new parsing for CREATE. Add PRM$M_SCS, PRM$M_TTY
0000   147 ;               and PRM$M_SYSGEN to SHOW/ALL mask. Change BOO$SEARCH
0000   148 ;               to return no such parameter if a search for a zero
0000   149 ;               length parameter is passed in.
0000   150 ;
0000   151 ;--
0000   152
0000   153 ;
0000   154 ; Include files:
0000   155 ;
0000   156         $BOOCMDDEF                      ; Flag bits in command options longword
0000   157         $CLUBDEF                        ; Cluster block offsets
0000   158         $IPLDEF                         ; IPL defs
0000   159         $PRVDEF                         ; Privilege definitions
0000   160         $PCBDEF                         ; PCB offsets
0000   161         $PRMDEF                         ; Parameter descriptor definitions
0000   162         $SSDEF                          ; System messages
0000   163         $SYSGMSGDEF                     ; Sysgen messages
0000   164         $TPADEF                         ; Define TPARSE symbols
0000   165
0000   166 ;
0000   167 ; MACROS:
0000   168 ;
0000   169 ;
0000   170 ; Macro to print message
```

```
                     0000   171 ;
                     0000   172 ;       MSG     message_text
                     0000   173 ;
                     0000   174          .MACRO  MSG,STR                 ;
                     0000   175          BSBW    BOO$FACMSG              ;
                     0000   176          .ASCIZ  \'STR'\                 ;
                     0000   177          .ENDM   MSG                     ;
                     0000   178
                     0000   179 ;
                     0000   180 ; Equated Symbols:
                     0000   181 ;
0000000D             0000   182          CR=13                           ; Character value for carriage return
0000000C             0000   183          FF=12                           ; Character value for form feed
0000000A             0000   184          LF=10                           ; Character value for line feed
00000100             0000   185          BUFFER_SIZE=256
                     0000   186
```

```
0000    188          .SBTTL  PARSE TABLES
0000    189 ;
0000    190 ;        DEFINE COMMAND SYNTAX
0000    191 ;
0000    192
0000    193          $INIT_STATE     STATE1,KEYTBL   ;
0000    194          $STATE                          ;
0000    195          $TRAN   !DISABLCMD,TPA$_EXIT    ; Disable option command
0000    196          $TRAN   !ENABLCMD,TPA$_EXIT     ; Enable option command
0000    197          $TRAN   'HELP',TPA$_EXIT,BOO$GIVEHELP   ; Help command
0000    198          $TRAN   !SETCMD,TPA$_EXIT       ; Set specific value
0000    199          $TRAN   !SHOCMD,TPA$_EXIT       ; Show values
0000    200          $TRAN   !USECMD,TPA$_EXIT       ; Set background values
0000    201          $TRAN   'EXIT',TPA$_EXIT,,BOOCMD$M_CONT,BOO$GL_CMDOPT   ; Same as continue
0000    202
0000    203          .IF     NDF,CMDSW               ;SYSBOOT specific commands
0000    204          $TRAN   'CONTINUE',TPA$_EXIT,,BOOCMD$M_CONT,BOO$GL_CMDOPT ; Continue command
0000    205
0000    206          .IFF                            ;SYSGEN specific commands
0000    207          $TRAN   !ADPCMD,TPA$_EXIT,BOO$CONADP    ; Set adapter TR number
0000    208          $TRAN   !CONECTCMD,TPA$_EXIT,BOO$CONNECT ; Connect command
0000    209          $TRAN   !CREATECMD,TPA$_EXIT,BOO$CREATE ; Create dump/page/swap file
0000    210          $TRAN   !INSTALCMD,TPA$_EXIT,BOO$INSTALL; Install swap/page file
0000    211          $TRAN   !LOADCMD,TPA$_EXIT,BOO$LOAD     ; Load driver
0000    212          $TRAN   !RELOADCMD,TPA$_EXIT,BOO$RELOAD ; Reload driver
0000    213          $TRAN   !MSCPCMD,TPA$_EXIT,BOO$LOAD     ; Load and start the MSCP server
0000    214          $TRAN   !SHARECMD,TPA$_EXIT,GEN$SHARE ; Share command
0000    215          $TRAN   !WRTCMD,TPA$_EXIT               ; Write parameter file
0000    216          $TRAN   !AUTOCONFIG,TPA$_EXIT           ; Auto-configure command
0000    217          $TRAN   !CONFIGCMD,TPA$_EXIT,BOO$CONFIGURE
0000    218          $TRAN   !TERMINALCMD,TPA$_EXIT          ; terminal command
0000    219          .ENDC
0000    220
0000    221          $TRAN   TPA$_EOS,TPA$_EXIT      ; END OF LINE
0000    222
0000    223 ;
0000    224 ; Disable command
0000    225 ;
0000    226          $STATE  DISABLCMD               ; Disable command
0000    227          $TRAN   'DISABLE'               ; Command verb
0000    228          $STATE                          ;
0000    229          $TRAN   'CHECKS',TPA$_EXIT,BOO$NOCHECK  ; Disable value checking
0000    230 ;
0000    231 ;        Recognize ENABLE command
0000    232 ;
0000    233          $STATE  ENABLCMD                ; ENABLE command
0000    234          $TRAN   'ENABLE'                ; Command verb
0000    235          $STATE                          ;
0000    236          $TRAN   'CHECKS',TPA$_EXIT,BOO$CHECK    ;
0000    237
0000    238 ;
0000    239 ;        Recognize SET Command
0000    240 ;
0000    241          $STATE  SETCMD                  ; SET command
0000    242          $TRAN   'SET'                   ; Command verb
0000    243          $STATE                          ;
0000    244          $TRAN   '/',SETSPEC             ;
```

```
                                                E 9
0000    245         $TRAN   '.',,BOO$DOT                ; Use last name
0000    246         $TRAN   TPA$_SYMBOL,,BOO$SEARCH     ; Lookup and verify symbol name
0000    247         $STATE                              ;
0000    248         $TRAN   !ASCII,TPA$_EXIT            ; Verify and set ASCII string
0000    249         $TRAN   !NUMBER,TPA$_EXIT,BOO$SETVALUE  ; Verify and set value
0000    250         $TRAN   'DEFAULT',TPA$_EXIT,BOO$SETDEF  ; Set to default value
0000    251         $STATE  SETSPEC                     ;
0000    252         $TRAN   !SETSTARTUP,TPA$_EXIT       ; Set startup file name
0000    253         $TRAN   'SETOUTPUT,TPA$_EXIT        ; Set output filespec
0000    254
0000    255         $STATE  ASCII
0000    256         $TRAN   '"',,,TPA$M_BLANKS,PARMBLK+TPA$L_OPTIONS; Make blanks significant
0000    257         $STATE  SYMBOL
0000    258         $TRAN   TPA$_SYMBOL,,BOO$SETASCII
0000    259         $TRAN   TPA$_BLANK,SYMBOL           ; ignore blanks
0000    260         $TRAN   '"',TPA$_EXIT,BOO$SETBLANK; null string => all blanks
0000    261         $STATE
0000    262         $TRAN   '"',TPA$_EXIT
0000    263
0000    264         $STATE  SETOUTPUT
0000    265         $TRAN   'OUTPUT'
0000    266         $STATE
0000    267         $TRAN   !SEPARATOR                  ; = or :
0000    268         $TRAN   TPA$_LAMBDA                 ; Or null
0000    269         $STATE
0000    270         $TRAN   !FILESPEC
0000    271         $STATE
0000    272         $TRAN   TPA$_EOS,TPA$_EXIT,BOO$SET_OUTPUT
0000    273
0000    274         $STATE  SETSTARTUP
0000    275         $TRAN   'STARTUP'
0000    276         $STATE
0000    277         $TRAN   !SEPARATOR                  ; = or :
0000    278         $TRAN   TPA$_LAMBDA                 ; Or null
0000    279         $STATE
0000    280         $TRAN   !FILESPEC
0000    281         $STATE
0000    282         $TRAN   TPA$_EOS,TPA$_EXIT,BOO$SETSTART
0000    283
0000    284 ;
0000    285 ;       Recognize SHOW Command
0000    286 ;
0000    287         $STATE  SHOCMD                      ; SHOW command
0000    288         $TRAN   'SHOW'                      ; Command verb
0000    289         $STATE  SHOSWITCH                   ;
0000    290         $TRAN   '/'                         ;
0000    291         $TRAN   '.',SHOWONE,BOO$DOT         ; SHOW .
0000    292         $TRAN   TPA$_SYMBOL,SHOWONE,BOO$SEARCH  ; Lookup and verify symbol name
0000    293         $STATE                              ;
0000    294         $TRAN   'HEX',SHOSWITCH,,BOOCMD$M_DISHEX,BOO$GL_CMDOPT
0000    295         $TRAN   'ACP',HEXQUAL2,,,,PRM$M_ACP; SHOW/ACP
0000    296 ;
0000    297 ; Note that PRM$M_ALL doesn't exist in $PRMDEF. It is used here simply as
0000    298 ; a flag to BOO$SHOALL.
0000    299 ;
0000001F 0000   300 PRM$V_ALL = 31
80000000 0000   301 PRM$M_ALL = 1@PRM$V_ALL
```

```
0000    302
0000    303          $TRAN   'ALL',,HEXQUAL2,,,,PRM$M_ALL; SHOW/ALL
0000    304          $TRAN   'RMS',,HEXQUAL2,,,,<PRM$M_RMS>; SHOW/RMS
0000    305          $TRAN   'SCS',,HEXQUAL2,,,,<PRM$M_SCS>; SHOW/SCS
0000    306          $TRAN   'SPECIAL',,HEXQUAL2,,,,<PRM$M_SPECIAL>; SHOW/SPECIAL
0000    307          $TRAN   'SYS',,HEXQUAL2,,,,<PRM$M_SYS>; SHOW/SYS
0000    308          $TRAN   'GEN',,HEXQUAL2,,,,<PRM$M_SYSGEN>; SHOW/GEN (Sysgen Parameters)
0000    309          $TRAN   'JOB',,HEXQUAL2,,,,<PRM$M_JBC>; SHOW/JOB (Job controller)
0000    310          $TRAN   'PQL',,HEXQUAL2,,,,<PRM$M_PQL>; SHOW/PQL (Process quota list)
0000    311          $TRAN   'TTY',,HEXQUAL2,,,,<PRM$M_TTY>; SHOW/TTY
0000    312          $TRAN   'LGI',,HEXQUAL2,,,,<PRM$M_LGI>; SHOW/LGI
0000    313          $TRAN   'CLUSTER',,HEXQUAL2,,,,<PRM$M_CLUSTER>    ; SHOW/CLUSTER show cluster
0000    314          $TRAN   'NAMES',TPA$_EXIT,BOO$SHONAMES   ; SHOW/NAMES show parameter names
0000    315          $TRAN   'MAJOR',,HEXQUAL2,,,,<PRM$M_MAJOR>        ; SHOW/MAJOR show major para
0000    316          $TRAN   'DYNAMIC',,HEXQUAL2,,,,<PRM$M_DYNAMIC>; SHOW/DYNAMIC show dyn. params
0000    317          $TRAN   'STARTUP',TPA$_EXIT,BOO$SHOSTART      ; SHO/STARTUP Show startup file
0000    318          .IF     DF,CMDSW                              ;SYSGEN specific qualifiers
0000    319          $TRAN   'ADAPTER',TPA$_EXIT,BOO$SHOW_ADAPTER ; SHOW/ADAPTER
0000    320          $TRAN   'CONFIGURATION',SHOWCON,BOO$RESET_IO ; SHOW/CONFIGURATION
0000    321          $TRAN   !SHOW_UNIBUS,TPA$_EXIT               ; /UNIBUS
0000    322          $TRAN   !DEV_OR_DRIV,TPA$_EXIT               ; /DEVICES and /DRIVER
0000    323          .ENDC
0000    324
0000    325          $STATE  SHOWONE                       ; SHOW value_name
0000    326          $TRAN   TPA$_LAMBDA,TPA$_EXIT,BOO$SHOVALUE      ;
0000    327
0000    328          $STATE  HEXQUAL2
0000    329          $TRAN   !HEXQUAL,TPA$_EXIT,BOO$SHOALL
0000    330
0000    331          $STATE  HEXQUAL
0000    332          $TRAN   '/'
0000    333          $TRAN   TPA$_LAMBDA,TPA$_EXIT
0000    334          $STATE
0000    335          $TRAN   'HEX',TPA$_EXIT,,BOOCMD$M_DISHEX,BOO$GL_CMDOPT
0000    336 ;
0000    337 ; Recognize USE command
0000    338 ;
0000    339          $STATE  USECMD                        ;
0000    340          $TRAN   'USE',,,BOOCMD$M_USEFILE,BOO$GL_CMDOPT ;
0000    341          $STATE                                ;
0000    342          $TRAN   !USECUR                       ;
0000    343          $TRAN   !USEACT                       ;
0000    344          $TRAN   !USEDEF                       ;
0000    345          $TRAN   !FILESPEC,,BOO$USEFILE        ;
0000    346          $STATE                                ;
0000    347          $TRAN   TPA$_LAMBDA,TPA$_EXIT         ;
0000    348
0000    349          $STATE  USECUR                        ; USE CURRENT
0000    350          $TRAN   'CURRENT'
0000    351          $STATE
0000    352          $TRAN   TPA$_EOS,TPA$_EXIT,BOO$USECUR
0000    353
0000    354          $STATE  USEACT                        ; USE ACTIVE
0000    355          $TRAN   'ACTIVE'
0000    356          $STATE
0000    357          $TRAN   TPA$_EOS,TPA$_EXIT,BOO$USEACT
0000    358
```

```
0000    359          $STATE  USEDEF                          ; USE DEFAULT
0000    360          $TRAN   'DEFAULT'
0000    361          $STATE
0000    362          $TRAN   TPA$_EOS,TPA$_EXIT,,BOOCMD$M_DEFAULT,BOO$GL_CMDOPT      ;
0000    363
0000    364  ;
0000    365  ; File Specification
0000    366  ;
0000    367          $STATE  FILESPEC                       ; GENERAL FILE SPEC CHECK
0000    368          $TRAN   TPA$_LAMBDA,TPA$_EXIT,BOO$FILESPEC
0000    369
0000    370  ;
0000    371  ; RECOGNIZE NUMBER
0000    372  ;
0000    373          $STATE NUMBER                          ;
0000    374          $TRAN   TPA$_DECIMAL,TPA$_EXIT         ; DECIMAL NUMBER
0000    375          $TRAN   '%'                            ; BASE PREFIX
0000    376          $STATE                                 ;
0000    377          $TRAN   'X',HEXNUM                     ; HEX BASE DESIGNATOR
0000    378          $TRAN   'O'                            ; OCTAL NUMBER
0000    379          $STATE                                 ;
0000    380          $TRAN   TPA$_OCTAL,TPA$_EXIT           ; INTRODUCED OCTAL NUMBER
0000    381          $STATE  HEXNUM                         ; INTRODUCED HEX NUMBER
0000    382          $TRAN   TPA$_HEX,TPA$_EXIT             ; HEX NUMBER
0000    383
0000    384  ;
0000    385  ; RECOGNIZE SWITCH/VALUE SEPARATOR
0000    386  ;
0000    387          $STATE  SEPARATOR                      ;
0000    388          $TRAN   '=',TPA$_EXIT                  ;
0000    389          $TRAN   ':',TPA$_EXIT                  ;
0000    390
0000    391  ;
0000    392  ; Get a numeric qualifier value
0000    393  ;
0000    394          $STATE  VALUE                          ; Get value for option
0000    395          $TRAN   !SEPARATOR                     ;
0000    396          $STATE                                 ;
0000    397          $TRAN   !NUMBER,TPA$_EXIT              ;
0000    398
0000    399          .IF     DF,CMDSW                       ;SYSGEN specific commands
0000    400  ;
0000    401  ; Adapter command
0000    402  ;
0000    403          $STATE  ADPCMD                         ; Command to set adapter TR number
0000    404          $TRAN   'ADAPTER',,BOO$RESET_ADAP      ; Command verb
0000    405          $STATE                                 ;
0000    406          $TRAN   !NUMBER,TPA$_EXIT              ; Numeric value
0000    407          $TRAN   !ADAP_STR,TPA$_EXIT            ; Generic Name
0000    408
0000    409  ;
0000    410  ; Autoconfigure command
0000    411  ;
0000    412          $STATE  AUTOCONFIG                     ; Auto configure command
0000    413          $TRAN   'AUTOCONFIGURE',,BOO$RESETLIST  ; Command verb
0000    414          $STATE                                 ;
0000    415          $TRAN   'ALL',CONFIGALL                ; Configure all
```

```
0000            $TRAN   !NUMBER                 ; Configure one TR number
0000            $TRAN   !ADAP_STR2              ; Generic Name
0000            $STATE                          ;
0000            $TRAN   !AUTOOPT,TPA$_EXIT,BOO$CONFIGONE        ;
0000
0000            $STATE  CONFIGALL               ;
0000            $TRAN   !AUTOOPT,TPA$_EXIT,BOO$CONFIGALL        ;
0000
0000            $STATE  AUTOOPT                 ; Select option
0000            $TRAN   '/'                     ; Switch introducer
0000            $TRAN   TPA$_LAMBDA,TPA$_EXIT   ; Else not specified
0000            $STATE                          ;
0000            $TRAN   'LOG',AUTOOPT,,BOOCMD$M_AUTOLOG,BOO$GL_CMDOPT ; LOG DEVICES
0000            $TRAN   'SELECT',,,BOOCMD$M_SELECT,BOO$GL_CMDOPT ; Option name
0000            $TRAN   'EXCLUDE',,,BOOCMD$M_EXCLUDE,BOO$GL_CMDOPT
0000                                            ; Reverse sense of select
0000            $STATE                          ;
0000            $TRAN   !SEPARATOR              ; : or =
0000            $STATE                          ;
0000            $TRAN   '('                     ; Allow parentheses
0000            $TRAN   TPA$_LAMBDA             ; But make it optional
0000
0000            $STATE  SELECTLIST              ; Device selectlist
0000            $TRAN   TPA$_SYMBOL,,BOO$MAKLIST        ; Select string
0000            $TRAN   ')',AUTOOPT             ; End ')'
0000            $TRAN   TPA$_LAMBDA,AUTOOPT     ; Else end of list
0000            $STATE                          ;
0000            $TRAN   <','>,SELECTLIST        ; Another option in list
0000            $TRAN   TPA$_LAMBDA,SELECTLIST  ; Else end
0000
0000    ;
0000    ; CONFIGURE command
0000    ;
0000
0000            $STATE  CONFIGCMD
0000            $TRAN   'CONFIGURE',,BOO$RESET_IO ; Reset IO and AUTORESET of devices names
0000            $STATE
0000            $TRAN   !CONFIG_LIST
0000            $TRAN   TPA$_LAMBDA,TPA$_EXIT
0000
0000            $STATE  CONFIG_LIST
0000            $TRAN   !CONFIG_OPT,CONFIG_LIST
0000            $TRAN   TPA$_LAMBDA,TPA$_EXIT
0000
0000            $STATE  CONFIG_OPT
0000            $TRAN   '/'
0000            $STATE
0000            $TRAN   !INPUT,TPA$_EXIT
0000            $TRAN   !OUTPUT,TPA$_EXIT
0000            $TRAN   !RESET,TPA$_EXIT
0000
0000            $STATE  INPUT
0000            $TRAN   'INPUT'
0000            $STATE
0000            $TRAN   !SEPARATOR
0000            $STATE
0000            $TRAN   !FILESPEC,TPA$_EXIT,BOO$INPUT_FILE
```

```
0000   473
0000   474          $STATE  OUTPUT
0000   475          $TRAN   'OUTPUT'
0000   476          $STATE
0000   477          $TRAN   !SEPARATOR
0000   478          $STATE
0000   479          $TRAN   !FILESPEC,TPA$_SYMBOL,BOO$OUTPUT_FILE
0000   480
0000   481          $STATE  RESET
0000   482          $TRAN   'RESET',TPA$_EXIT                   ; No action,this is the default
0000   483          $TRAN   'NORESET',TPA$_EXIT,BOO$NO_RESET ; Turn reset off this call
0000   484
0000   485
0000   486 ;
0000   487 ; Create command - Create contiguous file for paging, swapping or system dump
0000   488 ;
0000   489          $STATE  CREATECMD                   ;
0000   490          $TRAN   'CREATE'                    ; Command verb
0000   491          $STATE  CREOPT                      ; Create options
0000   492          $TRAN   '/',CREATE_QUAL             ;
0000   493          $TRAN   !FILESPEC,CREOPT,BOO$SETFILNAM; Set name of file
0000   494          $TRAN   TPA$_EOS,TPA$_EXIT          ;
0000   495
0000   496          $STATE  CREATE_QUAL                 ;
0000   497          $TRAN   'CONTIGUOUS',CREOPT,BOO$CRECONTIG    ; Contiguous
0000   498          $TRAN   'NOCONTIGUOUS',CREOPT,BOO$CRENCONTIG ; Set non-contiguous
0000   499          $TRAN   'SIZE'                      ; Get the allocation size
0000   500          $STATE                              ;
0000   501          $TRAN   !VALUE,CREOPT,BOO$FILESIZE ; Set file size
0000   502 ;
0000   503 ; Connect command - Connect specified device and load driver if required
0000   504 ;
0000   505          $STATE  CONECTCMD                   ;
0000   506          $TRAN   'CONNECT',,BOO$CONRESET     ; Command verb
0000   507          $STATE                              ;
0000   508          $TRAN   'CONSOLE',CONSOLCMD         ; Connect console command
0000   509          $TRAN   TPA$_SYMBOL,,BOO$DEVNAME    ; Device name
0000   510          $STATE                              ;
0000   511          $TRAN   ':'                         ;allow an optional ":"
0000   512          $TRAN   TPA$_LAMBDA                 ;
0000   513          $STATE  CONOPT                      ;
0000   514          $TRAN   !CONECTOPT,CONOPT           ; Connect option
0000   515          $TRAN   TPA$_LAMBDA,TPA$_EXIT       ;
0000   516 ;
0000   517 ; Connect console command
0000   518 ;
0000   519          $STATE  CONSOLCMD
0000   520          $TRAN   !CONSOLOPT,TPA$_EXIT,BOO$CONSOLE;
0000   521
0000   522          $STATE  CONSOLOPT
0000   523          $TRAN   '/'
0000   524          $TRAN   TPA$_LAMBDA,TPA$_EXIT
0000   525          $STATE
0000   526          $TRAN   'REMOTE',TPA$_EXIT,,BOOCMD$M_REMOTE,BOO$GL_CMDOPT; Connect remote co
0000   527          $TRAN   'LOGICAL',TPA$_EXIT,,BOOCMD$M_LOGICAL,BOO$GL_CMDOPT; Connect logical
0000   528          $TRAN   TPA$_LAMBDA,TPA$_EXIT
0000   529
```

```
0000    530 ;
0000    531 ;       Recognize INSTALL command
0000    532 ;
0000    533         $STATE  INSTALCMD                   ;
0000    534         $TRAN   ':NSTALL'                   ;
0000    535         $STATE                              ;
0000    536         $TRAN   !FILESPEC,,BOO$SETFILNAM;
0000    537         $STATE  INS1                        ;
0000    538         $TRAN   '/'                         ; Switch introducer
0000    539         $STATE
0000    540         $TRAN   'PAGEFILE',,BOO$   PGFL
0000    541         $TRAN   'SWAPFILE',INS_EXIT
0000    542         $TRAN   'CHECKPOINT',INS_PAGE
0000    543         $TRAN   'NOCHECKPOINT',INS_PAGE,BOO$NOCHKPNT
0000    544         $STATE
0000    545         $TRAN   '/'                         ; look for the checkpoint switch
0000    546         $TRAN   TPA$_EOS,TPA$_EXIT
0000    547         $STATE
0000    548         $TRAN   'CHECKPOINT',TPA$_EXIT
0000    549         $TRAN   'NOCHECKPOINT',TPA$_EXIT,BOO$NOCHKPNT
0000    550         $STATE  INS_PAGE
0000    551         $TRAN   '/'
0000    552         $STATE
0000    553         $TRAN   'PAGEFILE',,BOO$SETPGFL
0000    554         $STATE  INS_EXIT
0000    555         $TRAN   TPA$_EOS,TPA$_EXIT
0000    556 ;
0000    557 ;       Recognize LOAD command
0000    558 ;
0000    559         $STATE  LOADCMD                     ;
0000    560         $TRAN   'LOAD',,BOO$CONRESET; Command verb
0000    561         $STATE
0000    562         $TRAN   !FILESPEC,TPA$_EXIT,BOO$CONDRVNAM
0000    563         $STATE
0000    564         $TRAN   TPA$_LAMBDA,MSCP
0000    565 ;
0000    566 ;       Recognize MSCP command
0000    567 ;
0000    568         $STATE  MSCPCMD
0000    569         $TRAN   'MSCP_LOAD',,BOO$MSCP_RESET ; Loading and starting the MSCP server
0000    570         $STATE  MSCP
0000    571         $TRAN   !MSCPOPT,MSCP
0000    572         $TRAN   TPA$_LAMBDA,TPA$_EXIT
0000    573         $STATE  MSCPOPT
0000    574         $TRAN   '/'
0000    575         $STATE
0000    576         $TRAN   !MSCP_BUFFER,TPA$_EXIT,BOO$MSCP_ARG,,,2 ;BUFFERS IS PARAMETER 2
0000    577         $TRAN   !MSCP_PACKET,TPA$_EXIT,BOO$MSCP_ARG,,,3 ;PACKET IS PARAMETER 3
0000    578         $TRAN   !MSCP_HOSTS,TPA$_EXIT,BOO$MSCP_ARG,,,4  ;HOSTS IS PARAMETER 4
0000    579         $TRAN   !MSCP_TIME_OUT,TPA$_EXIT,BOO$MSCP_ARG,,,5;TIME_OUT IS PARAMETER 5
0000    580         $TRAN   !MSCP_PRIORITY,TPA$_EXIT,BOO$MSCP_ARG,,,6;PRIORITY IS PARAMETER 6
0000    581         $TRAN   !MSCP_SMALL,TPA$_EXIT,BOO$MSCP_ARG,,,7  ;SMALL IS PARAMETER 7
0000    582         $TRAN   !MSCP_FRACTION,TPA$_EXIT,BOO$MSCP_ARG,,,8;FRACTION IS PARAMETER 8
0000    583         $TRAN   !LOADARGCNT,TPA$_EXIT,BOO$MSCP_ARG,,,0  ;ARGUMENT COUNT
0000    584         $TRAN   !LOADP1,TPA$_EXIT,BOO$MSCP_ARG,,,1      ;LOAD PARAMETER 1
0000    585
0000    586         $STATE  MSCP_BUFFER
```

```
0000    587          $TRAN   'BUFFER',VALUE
0000    588          $TRAN   'P2',VALUE
0000    589
0000    590          $STATE  MSCP_PACKET
0000    591          $TRAN   'PACKET',VALUE
0000    592          $TRAN   'P3',VALUE
0000    593
0000    594          $STATE  MSCP_HOSTS
0000    595          $TRAN   'HOSTS',VALUE
0000    596          $TRAN   'P4',VALUE
0000    597
0000    598          $STATE  MSCP_TIME_OUT
0000    599          $TRAN   'TIME_OUT',VALUE
0000    600          $TRAN   'P5',VALUE
0000    601
0000    602          $STATE  MSCP_PRIORITY
0000    603          $TRAN   'PRIORITY',VALUE
0000    604          $TRAN   'P6',VALUE
0000    605
0000    606          $STATE  MSCP_SMALL
0000    607          $TRAN   'MINIMUM',VALUE
0000    608          $TRAN   'P7',VALUE
0000    609
0000    610          $STATE  MSCP_FRACTION
0000    611          $TRAN   'MAXIMUM',VALUE
0000    612          $TRAN   'P8',VALUE
0000    613
0000    614          $STATE  LOADARGCNT
0000    615          $TRAN   'ARGCOUNT',VALUE
0000    616
0000    617          $STATE  LOADP1
0000    618          $TRAN   'P1',VALUE
0000    619
0000    620 ;
0000    621 ;
0000    622 ;       Recognize RELOAD command
0000    623 ;
0000    624          $STATE  RELOADCMD                       ;
0000    625          $TRAN   'RELOAD',LOAD1,BOO$CONRESET; Command verb
0000    626
0000    627 ;
0000    628 ; Share command - Initialize and/or connect to a shared memory
0000    629 ;
0000    630          $STATE  SHARECMD
0000    631          $TRAN   'SHARE',,GEN$SHR_RESET  ; Command verb
0000    632          $STATE  SHARECMDOPT
0000    633          $TRAN   !SHAREOPT,SHARECMDOPT   ; Command options
0000    634          $TRAN   TPA$_LAMBDA
0000    635          $STATE
0000    636          $TRAN   'M'                     ; Multiport memory "MPMx"
0000    637          $STATE                          ;
0000    638          $TRAN   'P'                     ;
0000    639          $STATE                          ;
0000    640          $TRAN   'M'                     ;
0000    641          $STATE                          ;
0000    642          $TRAN   TPA$_DECIMAL,,GEN$SHR_UNIT ; Memory unit #
0000    643          $STATE                          ;
```

```
0000    644          $TRAN   TPA$_SYMBOL,,GEN$SHR_MEMNAME ; Memory name
0000    645          $STATE  SHROPT                       ;
0000    646          $TRAN   !SHAREOPT,SHROPT             ; Share options
0000    647          $TRAN   TPA$_EOS,TPA$_EXIT           ;
0000    648
0000    649 ;
0000    650 ; SYSGEN specific show qualifiers
0000    651 ;
0000    652          $STATE  DEV_OR_DRIV
0000    653          $TRAN   'DEVICES',,,,0               ; SHO/DEVICES[=devname]
0000    654          $TRAN   'DRIVER',,,,1                ; SHO/DRIVER [=devname]
0000    655          $STATE
0000    656          $TRAN   TPA$_EOS,TPA$_EXIT,BOO$SHODEV_ALL       ; SHOW ALL
0000    657          $TRAN   !SEPARATOR                   ;
0000    658          $TRAN   TPA$_LAMBDA
0000    659          $STATE                               ;
0000    660          $TRAN   TPA$_STRING,TPA$_EXIT,BOO$SHODEV        ; SHOW SPECIFIC DEVICE
0000    661
0000    662          $STATE  SHOWCON
0000    663          $TRAN   !SHOWCON_LOOP,TPA$_EXIT,BOO$SHOCONFIG
0000    664
0000    665          $STATE  SHOWCON_LOOP
0000    666          $TRAN   !SHOWCONOPT,SHOWCON_LOOP
0000    667          $TRAN   TPA$_EOS,TPA$_EXIT
0000    668          $TRAN   TPA$_LAMBDA,TPA$_FAIL
0000    669
0000    670          $STATE  SHOWCONOPT
0000    671          $TRAN   '/'
0000    672          $STATE
0000    673          $TRAN   'COMMAND_FILE',TPA$_EXIT,BOO$RESET_COMMAND ; Set command file spec
0000    674          $TRAN   !OUTPUT,TPA$_EXIT
0000    675          $TRAN   !ADAPTER,TPA$_EXIT,BOO$SET_TR
0000    676 ;
0000    677 ; SHOW /UNIBUS [/ADAPTER=n]
0000    678 ;
0000    679          $STATE  SHOW_UNIBUS
0000    680          $TRAN   'UNIBUS',,,,,0
0000    681          $STATE
0000    682          $TRAN   '/'
0000    683          $TRAN   TPA$_LAMBDA
0000    684          $STATE
0000    685          $TRAN   !ADAPTER,,BOO$SET_TR,,,1
0000    686          $TRAN   TPA$_LAMBDA
0000    687          $STATE
0000    688          $TRAN   TPA$_EOS,TPA$_EXIT,BOO$SHOW_UNIBUS
0000    689
0000    690          $STATE  ADAPTER                      ; Set adapter number
0000    691          $TRAN   'ADAPTER',,BOO$RESET_ADAP
0000    692          $STATE
0000    693          $TRAN   !SEPARATOR
0000    694          $STATE
0000    695          $TRAN   !NUMBER,TPA$_EXIT
0000    696          $TRAN   TPA$_LAMBDA,ADAP_STR
0000    697
0000    698          $STATE  ADAP_STR2
0000    699          $TRAN   TPA$_LAMBDA,ADAP_STR,BOO$RESET_ADAP
0000    700
```

```
0000    701          $STATE  ADAP_STR
0000    702          $TRAN   TPA$_ALPHA,ADAP_STR,BOO$ADAP_LETTER     ; One letter at a time
0000    703          $TRAN   TPA$_DECIMAL,TPA$_EXIT,BOO$ADAPTER_NAME ; Take number as end
0000    704
0000    705 ;
0000    706 ;        Recognize the TERMINAL command
0000    707 ;
0000    708          $STATE  TERMINALCMD
0000    709          $TRAN   'TERMINAL'
0000    710          $STATE
0000    711          $TRAN   '/'
0000    712          $STATE
0000    713          $TRAN   'ECHO',TPA$_EXIT,SYSG$LOAD_TT_STR ; /ECHO only qualifier
0000    714 ;
0000    715 ;        Recognize WRITE Command
0000    716 ;
0000    717          $STATE  WRTCMD                  ;
0000    718          $TRAN   'WRITE'                 ; Command verb
0000    719          $STATE                          ;
0000    720          $TRAN   !WRTCUR,TPA$_EXIT       ;
0000    721          $TRAN   !WRTACT,TPA$_EXIT       ;
0000    722          $TRAN   !FILESPEC,TPA$_EXIT,BOO$WRTFILE ;
0000    723
0000    724          $STATE  WRTCUR                  ; WRITE CURRENT
0000    725          $TRAN   'CURRENT'
0000    726          $STATE
0000    727          $TRAN   TPA$_EOS,TPA$_EXIT,BOO$WRTCUR
0000    728
0000    729          $STATE  WRTACT                  ; WRITE ACTIVE
0000    730          $TRAN   'ACTIVE'
0000    731          $STATE
0000    732          $TRAN   TPA$_EOS,TPA$_EXIT,BOO$WRTACT
0000    733 ;
0000    734 ; RECOGNIZE CONNECT OPTIONS
0000    735 ;
0000    736          $STATE  CONECTOPT               ;
0000    737          $TRAN   '/'                     ; Switch introducer
0000    738          $STATE                          ;
0000    739          $TRAN   !ADAPTER,TPA$_EXIT,BOO$CONADP           ; Adapter number
0000    740          $TRAN   'NOADAPTER',TPA$_EXIT,BOO$CONNLADP      ; Use null adapter
0000    741          $TRAN   !CONCREG,TPA$_EXIT,BOO$CONCREG          ; Control register (UBA)
0000    742          $TRAN   !CONCVECTOR,TPA$_EXIT,BOO$CONCVEC       ; Vector (UBA)
0000    743          $TRAN   !CONCNUMVEC,TPA$_EXIT,BOO$CONCNUM       ; Number of vectors
0000    744          $TRAN   !CONAUNIT,TPA$_EXIT,BOO$CONAUNIT        ; Adapter unit
0000    745          $TRAN   !CONUNITS,TPA$_EXIT,BOO$CONUNITS        ; Maximum units
0000    746          $TRAN   !CONSYSID_LO,TPA$_EXIT,BOO$CONSYSID_LOW  ; System ID (low)
0000    747          $TRAN   !CONSYSID_HI,TPA$_EXIT,BOO$CONSYSID_HIGH ; System ID (high)
0000    748          $TRAN   !CONVECOFF,TPA$_EXIT,BOO$CONVECOFFSET   ; Offset to vector(combo dev
0000    749          $TRAN   !CONCSROFF,TPA$_EXIT,BOO$CONCSROFFSET   ; Offset to CSR(combo device
0000    750          $TRAN   'DRIVERNAME'            ;
0000    751          $STATE                          ;
0000    752          $TRAN   !SEPARATOR              ;
0000    753          $STATE  LOAD1                   ;
0000    754          $TRAN   !FILESPEC,TPA$_EXIT,BOO$CONDRVNAM       ; Driver name
0000    755
0000    756          $STATE  CONCREG                 ; Control register address
0000    757          $TRAN   'CONTROLREGISTER',VALUE ;
```

```
0000          $TRAN   'CSR',VALUE                  ; Synonym
0000
0000          $STATE  CONCVECTOR                   ; Control vector address
0000          $TRAN   'VECTOR',VALUE               ;
0000
0000          $STATE  CONCNUMVEC                   ; Number of vectors
0000          $TRAN   'NUMVEC',VALUE               ;
0000
0000          $STATE  CONUNITS                     ; Maximum units
0000          $TRAN   'MAXUNITS',VALUE             ;
0000
0000          $STATE  CONSYSID_LO                  ; System id
0000          $TRAN   'SYSIDLOW',VALUE             ;
0000
0000          $STATE  CONSYSID_HI                  ; System id
0000          $TRAN   'SYSIDHIGH',VALUE            ;
0000
0000          $STATE  CONVECOFF                    ; Offset to vector from start of combo vecto
0000          $TRAN   'VECTOR_OFFSET',VALUE        ;
0000
0000          $STATE  CONCSROFF                    ; Offset to CSR from start of combo CSR
0000          $TRAN   'CSR_OFFSET',VALUE           ;
0000
0000          $STATE  CONAUNIT                     ; Adapter unit number
0000          $TRAN   'ADPUNIT',VALUE              ;
0000          $TRAN   ':',TPA$_EXIT                ;
0000    ;
0000    ; Recognize Share command options
0000    ;
0000          $STATE  SHAREOPT                     ;
0000          $TRAN   '/'                          ; Switch introducer
0000          $STATE                               ;
0000          $TRAN   !SHRGBLCNT,TPA$_EXIT,GEN$SHR_GBLCNT ; Global Section count
0000          $TRAN   !SHRMBXCNT,TPA$_EXIT,GEN$SHR_MBXCNT ; Mailbox count
0000          $TRAN   !SHRCLFCNT,TPA$_EXIT,GEN$SHR_CEFCNT ; Com Event Flags Clustr Cnt
0000          $TRAN   !SHRGBLMAX,TPA$_EXIT,GEN$SHR_GBLMAX ; Port max Global Sections
0000          $TRAN   !SHRMBXMAX,TPA$_EXIT,GEN$SHR_MBXMAX ; Port max mailboxes
0000          $TRAN   !SHRCEFMAX,TPA$_EXIT,GEN$SHR_CEFMAX ; Port max Com Event Flags
0000          $TRAN   !POOLCNT,TPA$_EXIT,GEN$SHR_POOLC ; Count of pool blocks
0000          $TRAN   !POOLSIZE,TPA$_EXIT,GEN$SHR_POOLS ; Size of pool blocks
0000          $TRAN   !PRQCNT,TPA$_EXIT,GEN$SHR_PRQCNT ; Count of PRQ blocks
0000
0000          $TRAN   !SHRSTART,TPA$_EXIT,GEN$SHR_START ; Start of useable mem.
0000          $TRAN   'INITIALIZE',TPA$_EXIT,GEN$SHR_INIT ; Initialize
0000
0000          $STATE  SHRGBLCNT                    ; Global section count
0000          $TRAN   'GBLSECTIONS',VALUE          ;
0000
0000          $STATE  SHRMBXCNT                    ; Mailbox count
0000          $TRAN   'MAILBOXES',VALUE            ;
0000
0000          $STATE  SHRCEFCNT                    ; Common event flag cluster count
0000          $TRAN   'CEFCLUSTERS',VALUE          ;
0000
0000          $STATE  SHRGBLMAX                    ; Port maximum Global Sections
0000          $TRAN   'MAXGBLSECTIONS',VALUE       ;
0000
```

```
0000    815         $STATE  SHRMBXMAX                  ; Port maximum Mailboxes
0000    816         $TRAN   'MAXMAILBOXES',VALUE      ;
0000    817
0000    818         $STATE  SHRCEFMAX                  ; Port maximum Common Ev Flag Clusters
0000    819         $TRAN   'MAXCEFCLUSTERS',VALUE     ;
0000    820
0000    821         $STATE  POOLCNT                    ; Total pool blocks count
0000    822         $TRAN   'POOLBCOUNT',VALUE         ;
0000    823         $TRAN   'POOLBCNT',VALUE           ;
0000    824
0000    825         $STATE  POOLSIZE                   ; Pool block size
0000    826         $TRAN   'POOLBSIZE',VALUE          ;
0000    827
0000    828         $STATE  PRQCNT                     ; Total PRQ blocks count
0000    829         $TRAN   'PRQCOUNT',VALUE           ;
0000    830         $TRAN   'PRQCNT',VALUE  ;
0000    831
0000    832         $STATE  SHRSTART                   ; Starting relative PFN
0000    833         $TRAN   'START',VALUE              ;
0000    834
0000    835         .ENDC                              ; End SYSGEN specific command
0000    836         $END_STATE
0000    837 ;
```

```
                                0000    839 ; Own Storage:
                                0000    840 ;
                                0000    841
                         00000000       842 .Psect NONPAGED_DATA,   noexe,rd,wrt,quad
                                0000    843
                                0000    844 BOO$GL_CMDOPT::                             ; Command options
                         00010000  0000 845         .LONG   BOOCMD$M_TERMINAL           ; Default is all off, except for terminal
                                0004    846
                                0004    847 SAVE_TODCBASE:                              ; Save area for system time and base
                         0000000C  0004 848         .BLKQ   1                           ; registers
                                000C    849 SAVE_TODR:
                         00000010  000C 850         .BLKL   1
                                0010    851
                               00000000 852         .PSECT  SYSBOOCMD,LONG
                                0000    853
                                0000    854 PARMBLK:                                    ; TPARSE parameter block
                         00000024  0000 855         .BLKB   TPA$K_LENGTH0               ;
                                0024    856 BOO$GL_DOT::                                ; Last parameter address
                         00000000  0024 857         .LONG   0                           ;
                                0028    858 BOO$GQ_FILDESC::                            ; File name descriptor
                       00000000 00000000 0028 859   .LONG   0,0                         ;
                                0030    860 BOO$GT_FILNAME::                            ; File name buffer
                         00000070  0030 861         .BLKB   64                          ;
                                0070    862 BOO$GT_COMBUF::                             ; Command Line Buffer
                         00000138  0070 863         .BLKB   200                         ;
                         000000C8  0138 864 BOO$C_COMBUFSZ==.-BOO$GT_COMBUF             ; Size of command buffer
                                0138    865 BOO$GT_COMSTR::                             ; Command string
                         00000538  0138 866         .BLKB   1024                        ;
                         00000400  0538 867 BOO$C_COMSTRLEN==.-BOO$GT_COMSTR            ; Length of command string buffer
                                0538    868 BOO$GT_SYSNAME::                            ; System name string
3A 4D 45 54 53 59 53 24 53 59 53 00'    869         .ASCIC  \SYS$SYSTEM:SYS.EXE\        ; Name of sytem image
         45 58 45 2E 53 59 53   0544
                        12      0538
                                054B    870 BOO$GT_SYSPARNAME::
3A 4D 45 54 53 59 53 24 53 59 53 00'    871         .ASCIC  \SYS$SYSTEM:VAXVMSSYS.PAR\; Name of the system .PAR file
41 50 2E 53 59 53 53 4D 56 58 41 56     0557
                                52      0563
                                18      054B
                                0564    872
                           44 00'       0564    873 BOO$T_DYNAMIC:          .ascic  /D/
                                01      0564
                                00'     0566    874 BOO$T_NODYNAMIC:        .ascic  //
                                00      0566
                                0567    875
20 20 20 20 20 20 20 20 20 20 20 00'    0567    876 CUR_BLANKS:     .ASCIC  /                   /
                     20 20 20 20 20 20  0573
                                11      0567
                        20 20 20 20 00' 0579    877 BLANKS:         .ASCIC  /    /
                                04      0579
                                057E    878
20 43 41 35 31 21 00000586'010E0000'    057E    879 CTRLSTR:        .ASCID  a!15AC          !4(10SL) !11AC !AC@
31 28 34 21 20 20 20 20 20 20 20 20     058C
21 20 43 41 31 31 21 20 29 4C 53 30     0598
                           43 41        05A4
                                05A6    880
20 43 41 35 31 21 000005AE'010E0000'    05A6    881 HEXSTR:         .ASCID  a!15AC          !4(10XL) !11AC !AC@
31 28 34 21 20 20 20 20 20 20 20 20     05B4
```

```
21 20 43 41 31 31 21 20 29 4C 58 30  05C0
                              43 41  05CC
                                     05CE
21 43 41 35 31 21 000005D6'010E0000' 05CE      ASCSTR:        .ASCID  @!15AC!AC"!AF"   !AC"!AF"   !AC".AF"   !AC"!AF" !11AC !AC@
41 21 20 20 20 22 46 41 21 22 43 41  05DC
43 41 21 20 20 20 22 46 41 21 22 43  05E8
22 43 41 21 20 20 20 22 46 41 21 22  05F4
21 20 43 41 31 31 21 20 22 46 41 21  0600
                              43 41  060C
                                     060E
31 28 23 21 20 20 00000616'010E0000' 060E      NCTRLSTR:      .ASCID  @ !#(17AC) @
               20 29 43 41 37        061C
                                     0621
72 61 74 53 20 20 00000629'010E0000' 0621      SCTRLSTR:      .ASCID  @ Startup command file = !AC@
20 64 6E 61 6D 6D 6F 63 20 70 75 74  062F
   43 41 21 20 3D 20 65 6C 69 66     063B
                                     0645
61 72 61 50 2F 21 0000064D'010E0000' 0645      CTR_PARINUSE:  .ASCID  @!/Parameters in use: !AC@
73 75 20 6E 69 20 73 72 65 74 65 6D  0653
            43 41 21 20 3A 65        065F
                                     0665
                                     0665      SDVHDR:
61 4E 20 72 65 74 65 6D 61 72 61 50  0665              .ASCII  \Parameter Name          Current   Default   Minimum   Maximum\
20 20 20 20 20 20 20 20 20 20 65 6D  0671
20 20 74 6E 65 72 72 75 43 20 20 20  067D
4D 20 20 20 74 6C 75 61 66 65 44 20  0689
78 61 4D 20 20 20 6D 75 6D 69 6E 69  0695
                  6D 75 6D 69        06A1
6D 61 6E 79 44 20 20 74 69 6E 55 20  06A5              .ASCII  \ Unit  Dynamic\
                              63 69  06B1
                              0A 0D  06B3              .ASCII  <CR><LF>                 ;
2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D  06B5              .ASCII  \--------------          -------   -------   -------   -------\
20 20 20 20 20 20 20 20 20 20 2D 2D  06C1
20 20 20 2D 2D 2D 2D 2D 2D 20 20 20  06CD
2D 20 20 20 2D 2D 2D 2D 2D 2D 20 20  06D9
2D 2D 2D 20 20 20 2D 2D 2D 2D 2D 2D  06E5
                  2D 2D 2D 2D        06F1
2D 2D 2D 2D 2D 20 20 20 2D 2D 2D 20  06F5              .ASCII  \ ----  -------\
                              2D 2D  0701
                          0000009E   0703      SDVHDRLEN=.-SDVHDR
                                     0703
                                     0703      .IF     NDF,CMDSW                ; SYSBOOCMD definitions for RIO$OUTPUT_LINE
                                     0703
                                     0703      RIO$AB_OUTBUF::          .long   BUFFER_SIZE
                                     0703                               .long   RIO$AB_BUFFER
                                     0703      RIO$AB_BUFFER::          .blkb   BUFFER_SIZE
                                     0703      RIO$GW_OUTLEN::          .word   0
                                     0703
                                     0703      BOO$GB_FILELEN:          .byte   0
                                     0703      BOO$GL_FILEADDR:         .long   0
                                     0703      BOO$GT_CURRENT:
                                     0703      BOO$GT_DEFAULT:
                                     0703      BOO$GL_PARINUSE:         .long   0
                                     0703
                                     0703      .ENDC
```

```
                     0703   914          .SBTTL
                     0703   915 ;++
                     0703   916 ;
                     0703   917 ; Functional Description:
                     0703   918 ;
                     0703   919 ;
                     0703   920 ; Calling Sequence:
                     0703   921 ;       NONE
                     0703   922 ;
                     0703   923 ; Input Parameters:
                     0703   924 ;       NONE
                     0703   925 ;
                     0703   926 ; Implicit Inputs:
                     0703   927 ;       NONE
                     0703   928 ;
                     0703   929 ; Output Parameters:
                     0703   930 ;       NONE
                     0703   931 ;
                     0703   932 ; Implicit Outputs:
                     0703   933 ;       NONE
                     0703   934 ;
                     0703   935 ; Side Effects:
                     0703   936 ;       NONE
                     0703   937 ;
                     0703   938 ;--
                     0703   939          .LIST   MEB                     ; Show macro expansions
                     0703   940
                     0703   941
                0FFC 0703   942 BOO$GETPARAM::  .WORD   ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>     ; Save all registers
                     0705   943
                     0705   944          .IF     NDF,CMDSW       ; SYSBOOCMD only
                     0705   945 ;
                     0705   946 ; Make descriptors PIC (only needed in SYSBOOCMD)
                     0705   947 ;
                     0705   948
                     0705   949          MOVAB   CTRLSTR+8,CTRLSTR+4     ; Set address in descriptor
                     0705   950          MOVAB   HEXSTR+8,HEXSTR+4       ; Set address in descriptor
                     0705   951          MOVAB   ASCSTR+8,ASCSTR+4       ; Set address in descriptor
                     0705   952          MOVAB   NCTRLSTR+8,NCTRLSTR+4   ; Set address in descriptor
                     0705   953          MOVAB   SCTRLSTR+8,SCTRLSTR+4   ; Set address in descriptor
                     0705   954          MOVAB   RIO$AB_BUFFER,-
                     0705   955                  RIO$AB_OUTBUF+4         ; Set address in descriptor
                     0705   956
                     0705   957          .ENDC
                     0705   958
                     0705   959 READCMD:                                 ;
    57   F8F7 CF     0705   960          MOVAL   PARMBLK,R7              ; Get address of parameter block
           08 A7     070A   961          CLRL    TPA$L_STRINGCNT(R7)     ; Initialize string length
 0C A7   FA27 CF     070D   962          MOVAB   BOO$GT_COMSTR,TPA$L_STRINGPTR(R7)       ; And address
                     0713   963 READLINE:                                ;
    52   F959 CF     0713   964          MOVAB   BOO$GT_COMBUF,R2        ; Set address of buffer
                52   0718   965          PUSHL   R2                      ; Set buffer address into argument list
      7E    C8 8F     071A   966          MOVZBL  #BOO$C_COMBUFSZ,-(SP)   ;  and maximum size for read
  00000000'EF   9F     071E   967          PUSHAB  BOO$GT_PROMPT           ; Address of prompt string
00000000'EF    03     0724   968          CALLS   #3,L^BOO$READPROMPT     ; Prompt for and accept command
         01 50 E8     072B   969          BLBS    R0,5$                   ; Exit if end of file.
                04   072E   970          RET                             ;
```

```
                                072F    971
                                072F    972 ; Upcase input
                                072F    973
                50    52    D0  072F    974 5$:     MOVL    R2,R0                   ; Set address of string
                51    80    9A  0732    975         MOVZBL  (R0)+,R1                ; Get address and count
             61 8F    60    91  0735    976 7$:     CMPB    (R0),#^A/a/             ; Lower case possible ?
                      09    1F  0739    977         BLSSU   8$                      ; No, Branch
             7A 8F    60    91  073B    978         CMPB    (R0),#^A/z/             ; Lower case possible ?
                      03    1A  073F    979         BGTRU   8$                      ; No, Branch
                60    20    8A  0741    980         BICB2   #^X20,(R0)              ; Clear bit, make character upper case
                      50    D6  0744    981 8$:     INCL    R0                      ; Increment pointer
                   EC 51    F5  0746    982         SOBGTR  R1,7$                   ; Loop
                                0749    983
          53   08 A7   0C A7    C1  0749    984         ADDL3   TPA$L_STRINGPTR(R7),TPA$L_STRINGCNT(R7),R3 ; Get current pointer
                50    82    9A  074F    985         MOVZBL  (R2)+,R0                ; Get length of input line
                      BF    13  0752    986         BEQL    READLINE                ; Ignore null input
                51    52    D0  0754    987         MOVL    R2,R1                   ; Move to LOCC address register
                52    81    9A  0757    988 30$:    MOVZBL  (R1)+,R2                ; Get a character
                52    2D    91  075A    989         CMPB    #^A/-/,R2               ; Is this a possible continuation?
                      1D    13  075D    990         BEQL    50$                     ; Branch if yes
                52    21    91  075F    991         CMPB    #^A/!/,R2               ; Is this the start of a comment?
                      0B    13  0762    992         BEQL    40$                     ; Branch if yes
                   08 A7    D6  0764    993         INCL    TPA$L_STRINGCNT(R7)     ; Bump characters in command string
                83    52    90  0767    994         MOVB    R2,(R3)+                ; Copy character to command string
                   EA 50    F5  076A    995 35$:    SOBGTR  R0,30$                  ; Continue for all characters in put
                      4B    11  076D    996         BRB     PARSE                   ; Done, parse command
                      50    D7  076F    997 40$:    DECL    R0                      ; One less character
          01 A1    50    21    3A  0771    998         LOCC    #^A/!/,R0,1(R1)         ; Scan remaining string for !
                      42    13  0776    999         BEQL    PARSE                   ; None end of line first
                      51    D6  0778   1000         INCL    R1                      ; Advance to next character
                      EE    11  077A   1001         BRB     35$                     ; Continue with line scan
                54    53    D0  077C   1002 50$:    MOVL    R3,R4                   ; Save string insertion pointer
             55    08 A7    D0  077F   1003         MOVL    TPA$L_STRINGCNT(R7),R5  ; and current length
                83    52    90  0783   1004         MOVB    R2,(R3)+                ; Copy to buffer anyway
                   08 A7    D6  0786   1005         INCL    TPA$L_STRINGCNT(R7)     ; Advance counter
                      15    11  0789   1006         BRB     65$                     ; And check for end of string
                52    81    9A  078B   1007 60$:    MOVZBL  (R1)+,R2                ; Get another character
                83    52    90  078E   1008         MOVB    R2,(R3)+                ; Copy to buffer
                   08 A7    D6  0791   1009         INCL    TPA$L_STRINGCNT(R7)     ; Bump string count
                52    20    91  0794   1010         CMPB    #^A/ /,R2               ; Blank?
                      07    13  0797   1011         BEQL    65$                     ; Yes, still might be a continuation
                52    21    91  0799   1012         CMPB    #^A/!/,R2               ; Is this a comment?
                      0F    13  079C   1013         BEQL    80$                     ; Branch if yes
                      CA    11  079E   1014         BRB     35$                     ; Not a continuation
                   E8 50    F5  07A0   1015 65$:    SOBGTR  R0,60$                  ; Continue to end of line
                53    54    D0  07A3   1016 70$:    MOVL    R4,R3                   ; Drop everything after continuation
             08 A7    55    D0  07A6   1017         MOVL    R5,TPA$L_STRINGCNT(R7)  ; By restoring count
                FF66  31    07AA   1018         BRW     READLINE                ; Read another line
                      50    D7  07AD   1019 80$:    DECL    R0                      ; One less character
          01 A1    50    21    3A  07AF   1020         LOCC    #^A/!/,R0,1(R1)         ; Scan for end of comment
                      ED    13  07B4   1021         BEQL    70$                     ; None
                      51    D6  07B6   1022         INCL    R1                      ; Skip trailing !
                      E6    11  07B8   1023         BRB     65$                     ; and continue scan for end of line
                67    08    D0  07BA   1024 PARSE:  MOVL    #TPA$K_COUNTO,TPA$L_COUNT(R7)   ; Init count field
             04 A7    02    C8  07BD   1025         BISL    #TPA$M_ABBREV,TPA$L_OPTIONS(R7) ; Permit abbreviations
                      CA  07C1   1026         BICL2   #^C<BOOCMD$M_NOCHECK!-
                          07C2   1027                 BOOCMD$M_SETOUTPUT!-
```

```
                                07C2  1028                    BOOCMD$M_TERMINAL>,-
00000000'EF   FFFE7FFE 8F       07C2  1029                    BOO$GL_CMDOPT           ; Clear all options but specified
                      18 A7  94 07CC  1030            CLRB    TPA$B_CHAR(R7)          ; Last character parsed
                00000000'EF  9F 07CF  1031            PUSHAB  KEYTBL                  ; Pass address of key table
                00000000'EF  9F 07D5  1032            PUSHAB  STATE1                  ;  and state table
                          57 DD 07DB  1033            PUSHL   R7                      ; Set address of parameter block
             00000000'GF  03 FB 07DD  1034            CALLS   #3,G^LIB$TPARSE         ; Parse input
                       58 50 E8 07E4  1035            BLBS    R0,20$                  ; Branch if no syntax error
                    52 50  1F E0 07E7  1036            BBS     #31,R0,15$              ; Branch if error already given
                                07EB  1037
                                07EB  1038 .IF      NDF,CMDSW                       ; SYSBOOCMD
                                07EB  1039
                                07EB  1040          MSG     <-E-Syntax error>       ; SYSBOOT error message
                                07EB  1041
                                07EB  1042 .IFF                                     ; SYSGENCMD
                                07EB  1043
             50 00000000'8F  D1 07EB  1044          CMPL    #LIB$_SYNTAXERR,R0      ; Tparse Syntax error ?
                          10 13 07F2  1045          BEQLU   10$                     ; Branch if yes
                          50 D5 07F4  1046          TSTL    R0                      ; Zero ?
                          0C 13 07F6  1047          BEQL    10$                     ; Branch if yes
                          50 DD 07F8  1048          PUSHL   R0                      ; Push REAL error code
             00000000'GF  01 FB 07FA  1049          CALLS   #1,G^LIB$SIGNAL         ; Signal Error
                       009B 31 0801  1050          BRW     30$                     ; Continue
                                0804  1051
                                0804  1052 ; Heuristically determine where syntax error occured
                                0804  1053
                  54 18 A7  9A  0804  1054 10$:   MOVZBL  TPA$B_CHAR(R7),R4       ; Was there a character parsed ?
                          06 12 0808  1055          BNEQ    12$                     ; Branch if yes
                  7E 08 A7  7D  080A  1056          MOVQ    TPA$L_STRINGCNT(R7),-(SP) ; Push entire read-in string
                          1E 11 080E  1057          BRB     14$                     ; Branch
                                0810  1058
          52  08 A7  10 A7  C3  0810  1059 12$:    SUBL3   TPA$L_TOKENCNT(R7),TPA$L_STRINGCNT(R7),R2 ; Length
          53  0C A7  10 A7  C1  0816  1060          ADDL3   TPA$L_TOKENCNT(R7),TPA$L_STRINGPTR(R7),R3 ; Address
                    54 2F 91  081C  1061          CMPB    #^A'/',R4               ; Was it a qualifier error ?
                          05 13 081F  1062          BEQL    13$                     ; No
                    7E 52 7D  0821  1063          MOVQ    R2,-(SP)
                          08 11 0824  1064          BRB     14$
                                0826  1065
             63 52  18 A7  3A  0826  1066 13$:    LOCC    TPA$B_CHAR(R7),R2,(R3)  ; Find it then
                    7E 50 7D  082B  1067          MOVQ    R0,-(SP)                ; Push length and address
                          02 DD 082E  1068 14$:    PUSHL   #2                      ; Number of FAO params
                00 7C 809A 8F DD  0830  1069          PUSHL   #SYSG$_SYNTAX           ; Error message
          00000000'GF  04 FB 0836  1070          CALLS   #4,G^LIB$SIGNAL         ; Signal the error
                                083D  1071
                                083D  1072 .ENDC
                                083D  1073
                          60 11 083D  1074 15$:    BRB     30$                     ; and get another command
             56 00000000'EF  D0 083F  1075 20$:    MOVL    BOO$GL_CMDOPT,R6        ; Get command option flags
                   58 56  08 E0 0846  1076          BBS     #BOOCMD$V_CONT,R6,EXIT  ; Exit if continue flag
                   51 56  09 E1 084A  1077          BBC     #BOOCMD$V_DEFAULT,R6,30$ ; Read another command if Help
                                084E  1078 ;
                                084E  1079 ; The Default values for system parameters are selected and must be copied to
                                084E  1080 ; the current system parameter area.
                                084E  1081 ;
00000004'EF   00000000'EF  7D   084E  1082          MOVQ    EXE$GQ_TODCBASE,SAVE_TODCBASE ; Save time base register
0000000C'EF   00000000'EF  D0   0859  1083          MOVL    EXE$GL_TODR,SAVE_TODR   ; Save time register
                                0864  1084
```

```
                    0000'8F    28  0864  1085          MOVC3   #EXE$C_SYSPARSZ,-
                00000000'EF        0868  1086                  BOO$A_SYSPARAM,-
                00000000'EF        086D  1087                  EXE$A_SYSPARAM          ; Copy defaults
                                   0872  1088
00000000'EF     00000004'EF    7D  0872  1089          MOVQ    SAVE_TODCBASE,EXE$GQ_TODCBASE ; Restore
00000000'EF     0000000C'EF    D0  087D  1090          MOVL    SAVE_TODR,EXE$GL_TODR
                                   0888  1091
                00000000'8F    E2  0888  1092          BBSS    #EXE$V_WRITESYSPARAMS,- ; Use default => write current needed
              00 00000000'GF        088E  1093                  G^EXE$GL_DYNAMIC_FLAGS,1$;
                                   0894  1094 1$:
                00000000'EF    DE  0894  1095          MOVAL   BOO$GT_DEFAULT,-
                00000000'EF        089A  1096                  BOO$GL_PARINUSE         ; Set default in use
                        FE63   31  089F  1097 30$:     BRW     READCMD                 ; Read more commands
                                   08A2  1098
                    50    01   D0  08A2  1099 EXIT:    MOVL    #1,R0                   ; Return success
                                   04  08A5  1100          RET                     ;
                                       08A6  1101
```

```
                                 08A6        .SBTTL  BOO$FILESPEC - Parse file spec
                                 08A6   ;+
                                 08A6   ;
                                 08A6   ; CALLING SEQUENCE:
                                 08A6   ;
                                 08A6   ;       called as a TPARSE action routine
                                 08A6   ;
                                 08A6   ; INPUT:
                                 08A6   ;
                                 08A6   ;       The tparse parameter block (AP)
                                 08A6   ;
                                 08A6   ; OUTPUT:
                                 08A6   ;
                                 08A6   ;       A possible file spec is found.
                                 08A6   ;
                                 08A6   ; SIDE EFFECTS:
                                 08A6   ;
                                 08A6   ;       The tparse parameter block is updated.
                                 08A6   ;
                                 08A6   ;-
                                 08A6
                         00FC    08A6   .Entry  BOO$FILESPEC, ^M<R2,R3,R4,R5,R6,R7>
                                 08A8
       52    0C AC    D0    08A8                MOVL    TPA$L_STRINGPTR(AP),R2   ; Get address of current parse
       14 AC    52    D0    08AC                MOVL    R2,TPA$L_TOKENPTR(AP)    ; Set token pointer
 00000000'EF    52    D0    08B0                MOVL    R2,BOO$GL_FILEADDR       ; Set file spec pointer
       53    08 AC    D0    08B7                MOVL    TPA$L_STRINGCNT(AP),R3   ; Remainder of parse string length
             25       13    08BB                BEQL    100$                     ; Error if zero
                                 08BD
    62    53    20    3A    08BD                LOCC    #^A/ /,R3,(R2)           ; is there a blank?
             04       12    08C1                BNEQ    50$                      ; Branch if yes
    62    53    2F    3A    08C3                LOCC    #^A@/@,R3,(R2)           ; is there a slash?
                                 08C7
     08 AC    50    D0    08C7   50$:        MOVL    R0,TPA$L_STRINGCNT(AP)   ; Remaining length
     0C AC    51    D0    08CB                MOVL    R1,TPA$L_STRINGPTR(AP)   ; Address of blank or slash
        51    52    C2    08CF                SUBL2   R2,R1                    ; Calculate length
     10 AC    51    D0    08D2                MOVL    R1,TPA$L_TOKENCNT(AP)    ; Set length of file spec
 00000000'EF    51    90    08D6                MOVB    R1,BOO$GB_FILELEN        ; Set length of file spec
        50    01    D0    08DD   60$:        MOVL    #SS$_NORMAL,R0           ; Set success
              07    11    08E0                BRB     110$                     ; Exit
                                 08E2
 50   00000000'8F    D0    08E2   100$:       MOVL    #LIB$_SYNTAXERR,R0
                    04    08E9   110$:       RET
                                 08EA
```

```
                                J 10
                     08EA  1148          .SBTTL  BOO$USECUR - Use parameters from current image
                     08EA  1149 BOO$USECUR::                             ; Set to current system values
                 03FC  08EA  1150          .WORD   ^M<R2,R3,R4,R5,R6,R7,R8,R9>;
50   0000054B'GF   9E  08EC  1151          MOVAB   G^BOO$GT_SYSPARNAME,R0  ; Get address of system .PAR file name
    5C    F709 CF   DE  08F3  1152          MOVAL   PARMBLK,AP              ; Get address of the TPARSE parameter block
      10 AC    80   9A  08F8  1153          MOVZBL  (R0)+,TPA$L_TOKENCNT(AP); Set up for call to BOO$USEFILE
      14 AC    50   D0  08FC  1154          MOVL    R0,TPA$L_TOKENPTR(AP)   ;
00000000'GF    6C   FA  0900  1155          CALLG   (AP),G^BOO$USEFILE      ; Call routine to process the .PAR file
            17 50   E9  0907  1156          BLBC    R0,10$                  ; Branch to failure code
     00000000'8F    E5  090A  1157          BBCC    #EXE$V_WRITESYSPARAMS,- ; Use current => no write current needed
  00 00000000'GF        0910  1158                  G^EXE$GL_DYNAMIC_FLAGS,5$;
     00000000'EF    DE  0916  1159 5$:      MOVAL   BOO$GT_CURRENT,-
     00000000'EF        091C  1160                  BOO$GL_PARINUSE         ; Set parameters in use
         50   01    D0  0921  1161 10$:     MOVL    #1,R0                   ; Return success
                    04  0924  1162          RET                             ;
                        0925  1163
```

```
                                0925          .SBTTL  BOO$SHOWV - Routine to show one parameter value
                                0925  ;
                                0925  ; Input Parameters:
                                0925  ;       R4 - Pointer to PRM block to be displayed.
                                0925  ;
                                0925  ; Output Parameters:
                                0925  ;       Content of parameter block is displayed by calling RIO$OUTPUT_LINE
                                0925  ;
                                0925  BOO$SHOWV:                                            ;
              50     64    D0   0925          MOVL    (R4),R0                               ; Get address of value
       50   00000000'EF40  9E   0928          MOVAB   BOO$A_SYSPARAM[R0],R0                 ; Add current base address
         50   00000000'8F  C2   0930          SUBL    #BOO$A_SYSPARAM,R0                    ;  and subtract link-time value
            53    FC6B CF  7E   0937          MOVAQ   HEXSTR,R3                             ; Assume hex display
                       0B  E0   093C          BBS     #BOOCMD$V_DISHEX,-
            0A 00000000'EF      093E                  BOO$GL_CMDOPT,1$                      ; If set, then display hex
            53    FC36 CF  7E   0944          MOVAQ   CTRLSTR,R3                            ; Assume not ascii data
            03 10 A4   10  E0   0949          BBS     #PRM$V_ASCII,PRM$L_FLAGS(R4),2$ ; Branch if ascii
                     0089  31   094E  1$:     BRW     15$                                   ;
                                0951
                                0951  ;
                                0951  ; ASCII data
                                0951  ;
                       55  DD   0951  2$:     PUSHL   R5                                    ; Save a register
              5E     10    C2   0953          SUBL2   #16,SP                                ; Allocate a buffer on the stack
           52     14 A4    9A   0956          MOVZBL  PRM$B_SIZE(R4),R2                     ; Get size (in bits)
       52     52    FD 8F  78   095A          ASHL    #-3,R2,R2                             ; Convert size from bit to byte count
              51     52    D0   095F          MOVL    R2,R1                                 ; Make a copy of the size
              04     52    D1   0962          CMPL    R2,#4                                 ; Size > 4?
                     03    15   0965          BLEQ    3$                                    ; If geq yes
              51     04    D0   0967          MOVL    #4,R1                                 ; Max of 4 for default, max and min
     FC09 CF   05    51    83   096A  3$:     SUBB3   R1,#5,BLANKS                          ; Calculate number of blank spaces needed
                     16    BB   0970          PUSHR   #^M<R1,R2,R4>                         ; Save some registers
 0C AE    10   20   60   52  2C   0972        MOVC5   R2,(R0),#^A/ /,#16,<3*4>(SP) ; Move the parameter value into the buf
                     16    BA   0979          POPR    #^M<R1,R2,R4>                         ; Restore the registers
              53     5E    D0   097B          MOVL    SP,R3                                 ; Save a pointer to the buffer
     FBE3 CF   11    52    83   097E          SUBB3   R2,#17,CUR_BLANKS                     ; Calculate number of pad blanks
              FBDE CF      DF   0984          PUSHAL  BOO$T_NODYNAMIC                       ; Assume not dynamic
        05 10 A4     00    E1   0988          BBC     #PRM$V_DYNAMIC,PRM$L_FLAGS(R4),10$ ; Branch if not
           6E     FBD3 CF  DE   098D          MOVAL   BOO$T_DYNAMIC,(SP)                    ; Change to dynamic string
                  26 A4    9F   0992  10$:    PUSHAB  PRM$T_UNIT(R4)                        ; Stack address of unit name string
                  0C A4    9F   0995          PUSHAB  PRM$L_MAX(R4)                         ; Stack maximum value
                     51    DD   0998          PUSHL   R1                                    ;  and size
                FBDB CF    9F   099A          PUSHAB  BLANKS                                ; Blanks for padding
                  08 A4    9F   099E          PUSHAB  PRM$L_MIN(R4)                         ; Minimum value
                     51    DD   09A1          PUSHL   R1                                    ;  and size
                FBD2 CF    9F   09A3          PUSHAB  BLANKS                                ; Blanks for padding
                  04 A4    9F   09A7          PUSHAB  PRM$L_DEFAULT(R4)                     ; Default value
                     51    DD   09AA          PUSHL   R1                                    ;  and size
                FBC9 CF    9F   09AC          PUSHAB  BLANKS                                ; Blanks for padding
                     53    DD   09B0          PUSHL   R3                                    ; Address of current value
                     52    DD   09B2          PUSHL   R2                                    ;  and size
                FBAF CF    9F   09B4          PUSHAB  CUR_BLANKS                            ; Blanks for padding
                  16 A4    9F   09B8          PUSHAB  PRM$T_NAME(R4)                        ; Stack address of parameter name
            00000000'EF    7F   09BB          PUSHAQ  RIO$AB_OUTBUF                         ; Stack address of buffer descriptor
            00000000'EF    DF   09C1          PUSHAL  RIO$GW_OUTLEN                         ; Set address of loc to receive size
                FC03 CF    7F   09C7          PUSHAQ  ASCSTR                                ; Control string for ascii
                                09CB
```

```
00000000'EF    12    FB   09CB  1222          CALLS   #18,SYS$FAO                     ; Format value for output
          5E   10    C0   09D2  1223          ADDL2   #16,SP                          ; Remove the buffer from the stack
               55  8ED0   09D5  1224          POPL    R5                              ; Restore a register
               41    11   09D8  1225          BRB     55$                             ; and join common code
                          09DA  1226
                          09DA  1227 ;
                          09DA  1228 ; Decimal or hex display - R3 contains address of control string
                          09DA  1229 ;
               46    10   09DA  1230 15$:     BSBB    GETDATA                         ; Get data item according to size
           FB86  CF    DF   09DC  1231          PUSHAL  BOO$T_NODYNAMIC                 ; Assume not dynamic
     05 10 A4    00    E1   09E0  1232          BBC     #PRM$V_DYNAMIC,PRM$L_FLAGS(R4),20$ ; Branch if not
      6E   FB7B  CF    DE   09E5  1233          MOVAL   BOO$T_DYNAMIC,(SP)              ; Change to dynamic string
                          09EA  1234
            26 A4    9F   09EA  1235 20$:     PUSHAB  PRM$T_UNIT(R4)                  ; Stack address of unit name string
            0C A4    DD   09ED  1236          PUSHL   PRM$L_MAX(R4)                   ; Stack maximum value
            08 A4    DD   09F0  1237          PUSHL   PRM$L_MIN(R4)                   ;  and minimum value
            04 A4    DD   09F3  1238          PUSHL   PRM$L_DEFAULT(R4)               ; Default value
     10 A4  1000  8F    B3   09F6  1239          BITW    #PRM$M_NEG,PRM$L_FLAGS(R4)      ; check for negated value
               03    13   09FC  1240          BEQL    30$                             ; Branch if not
            6E   6E    CE   09FE  1241          MNEGL   (SP),(SP)                       ; Make absolute value
               51    DD   0A01  1242 30$:     PUSHL   R1                              ; Current value
            16 A4    9F   0A03  1243          PUSHAB  PRM$T_NAME(R4)                  ; Stack address of parameter name
     00000000'EF     7F   0A06  1244          PUSHAQ  RIO$AB_OUTBUF                   ; Stack address of buffer descriptor
     00000000'EF     DF   0A0C  1245          PUSHAL  RIO$GW_OUTLEN                   ; Set address of loc to receive size
               53    DD   0A12  1246          PUSHL   R3                              ; Push address of control string
                          0A14  1247
00000000'EF    0A    FB   0A14  1248          CALLS   #10,SYS$FAO                     ; Format value for output
          03 50    E9   0A1B  1249 55$:     BLBC    R0,60$
                          0A1E  1250
           F5DF'   30   0A1E  1251          BSBW    RIO$OUTPUT_LINE                 ; Output the line
               05   0A21  1252 60$:     RSB                                     ; and return
                          0A22  1253
                          0A22  1254 GETDATA:                                    ;
          51    15 A4    9A   0A22  1255          MOVZBL  PRM$B_POS(R4),R1                ; GET SIZE OF DATUM
51   60    14 A4    51    EF   0A26  1256          EXTZV   R1,PRM$B_SIZE(R4),(R0),R1       ; GET DATUM
     10 A4  1000  8F    B3   0A2C  1257          BITW    #PRM$M_NEG,PRM$L_FLAGS(R4)      ; CHECK FOR NEGATED VALUE
               0D    13   0A32  1258          BEQL    10$                             ; NO
          51    15 A4    9A   0A34  1259          MOVZBL  PRM$B_POS(R4),R1                ; GET POSITION AGAIN
51   60    14 A4    51    EE   0A38  1260          EXTV    R1,PRM$B_SIZE(R4),(R0),R1       ; CONVERT TO SIGNED NUMBER
               51    51    CE   0A3E  1261          MNEGL   R1,R1                           ; ABSOLUTE VALUE
               05   0A41  1262 10$:     RSB                                     ;
                          0A42  1263
                          0A42  1264 ;
                          0A42  1265 ;       Show names of parameters
                          0A42  1266 ;
                          0A42  1267 BOO$SHONAMES:                               ;
                0FC   0A42  1268          .WORD   ^M<R2,R3,R4,R5,R6,R7>           ;
     56   00000000'EF    9E   0A44  1269          MOVAB   BOO$A_PRMBLK,R6                 ; Set base of parameter blocks
                          0A48  1270
          57    05    D0   0A48  1271 10$:     MOVL    #5,R7                           ; Init argument count
                66    D5   0A4E  1272          TSTL    (R6)                            ; At end of list?
                57    13   0A50  1273          BEQL    90$                             ; Yes, finished
          51    16 A6    DE   0A52  1274          MOVAL   PRM$T_NAME(R6),R1               ; Set parameter name address
          56    32 A6    9E   0A56  1275          MOVAB   PRM$C_LENGTH(R6),R6             ; Next parameter block
                66    D5   0A5A  1276          TSTL    (R6)                            ; At end of list?
                26    13   0A5C  1277          BEQL    70$                             ; yes
          52    16 A6    DE   0A5E  1278          MOVAL   PRM$T_NAME(R6),R2               ; Set second address
```

```
                     57   D6  0A62  1279          INCL    R7                              ; Advance argument count
               56    32 A6   9E  0A64  1280          MOVAB   PRM$C_LENGTH(R6),R6             ; Next argument
                     66   D5  0A68  1281          TSTL    (R6)                            ; At end of list ?
                     18   13  0A6A  1282          BEQL    70$                             ; yes
               53    16 A6   DE  0A6C  1283          MOVAL   PRM$T_NAME(R6),R3               ; Set third address
                     57   D6  0A70  1284          INCL    R7                              ; Another argument
               56    32 A6   9E  0A72  1285          MOVAB   PRM$C_LENGTH(R6),R6             ; Next parameter block
                     66   D5  0A76  1286          TSTL    (R6)                            ; At end of list?
                     0A   13  0A78  1287          BEQL    70$                             ; Yes
                     57   D6  0A7A  1288          INCL    R7                              ; another argument
               54    16 A6   9E  0A7C  1289          MOVAB   PRM$T_NAME(R6),R4               ; Set fourth address
               56    32 A6   DE  0A80  1290          MOVAL   PRM$C_LENGTH(R6),R6             ; Next parameter block
                     1E   BB  0A84  1291 70$:        PUSHR   #^M<R1,R2,R3,R4>                ; Stack args
            7E    57    04   C3  0A86  1292          SUBL3   #4,R7,-(SP)                     ; Set number of strings on line
             00000000'EF    7F  0A8A  1293          PUSHAQ  RIO$AB_OUTBUF                   ; Stack address of buffer descriptor
             00000000'EF    DF  0A90  1294          PUSHAL  RIO$GW_OUTLEN                   ; Set address of loc to receive size
                 FB74 CF    7F  0A96  1295          PUSHAQ  CTRLSTR                         ; Stack address of control string descr
00000000'EF     57    FB  0A9A  1296          CALLS   R7,SYS$FAO                      ; Format value for output
               08 50   E9  0AA1  1297          BLBC    R0,100$
                           0AA4  1298
                 F559'     30  0AA4  1299          BSBW    RIO$OUTPUT_LINE                 ; Output line
                    A2     11  0AA7  1300          BRB     10$                             ; Loop
                           0AA9  1301
               50   01     D0  0AA9  1302 90$:     MOVL    #1,R0                           ; Success status
                           04  0AAC  1303 100$:    RET                                     ; Return
                           0AAD  1304
                           0AAD  1305 ;
                           0AAD  1306 ; Show name of Startup command file
                           0AAD  1307 ;
                           0AAD  1308 BOO$SHOSTART:                                    ;
                    00FC   0AAD  1309          .WORD   ^M<R2,R3,R4,R5,R6,R7>           ;
                           0AAF  1310
50    00000000'EF    9E  0AAF  1311          MOVAB   L^EXE$GT_STARTUP,R0             ; Set address of string
                           0AB6  1312          $FAO_S  -
                           0AB6  1313                  CTRSTR  = SCTRLSTR,-            ; Stack address of control string descr
                           0AB6  1314                  OUTLEN  = RIO$GW_OUTLEN,-       ; Set address of loc to receive size
                           0AB6  1315                  OUTBUF  = RIO$AB_OUTBUF,-       ; Stack address of buffer descriptor
                           0AB6  1316                  P1      = R0                    ; Set address of startup string
                     50    DD  0AB6                  PUSHL   R0
             00000000'EF    7F  0AB8                  PUSHAQ  RIO$AB_OUTBUF
             00000000'EF    3F  0ABE                  PUSHAW  RIO$GW_OUTLEN
                 FB59 CF    7F  0AC4                  PUSHAQ  SCTRLSTR
00000000'GF     04    FB  0AC8                  CALLS   #$$T2,G^SYS$FAO
               03 50   E9  0ACF  1317          BLBC    R0,10$
                           0AD2  1318
                 F52B'     30  0AD2  1319          BSBW    RIO$OUTPUT_LINE
                           04  0AD5  1320 10$:     RET                                     ;
                           0AD6  1321
```

```
                        0AD6  1323          .SBTTL  BOO$NOCHECK - Disable value checking
                        0AD6  1324 ;
                        0AD6  1325 ; Disable Value Checking and Limiting
                        0AD6  1326 ;
                  0000  0AD6  1327 BOO$NOCHECK:     .WORD   0                       ;
00 00000000'EF    00    E3  0AD8  1328              BBCS    #BOOCMD$V_NOCHECK,BOO$GL_CMDOPT,10$ ; Set value check inhibit
                  50    01  D0  0AE0  1329 10$:     MOVL    #1,R0                   ; Return success
                        04  0AE3  1330              RET                             ;
```

```
                                0AE4  1332          .SBTTL  BOO$NOCHECK - Disable value checking
                                0AE4  1333
                                0AE4  1334 ;
                                0AE4  1335 ; ENABLE VALUE CHECKING AND LIMITING
                                0AE4  1336 ;
                                0AE4  1337 BOO$CHECK::                                  ;
                          0000  0AE4  1338          .WORD   0                           ; Null entry mask
00 00000000'EF    00      E5  0AE6  1339          BBCC    #BOOCMD$V_NOCHECK,BOO$GL_CMDOPT,10$ ; Clear check flag
                  50  01    D0  0AEE  1340 10$:     MOVL    #1,R0                       ; Return with success
                              04  0AF1  1341          RET                                 ;
                                0AF2  1342
```

```
                                0AF2  1344          .SBTTL  BOO$SEARCH - Lookup parameter name
                                0AF2  1345 ;
                                0AF2  1346 ; Input Parameters:
                                0AF2  1347 ;       TPA$L_TOKENCNT(AP) - Count of characters in token
                                0AF2  1348 ;       TPA$L_TOKENPTR(AP) - Address of token
                                0AF2  1349 ;
                                0AF2  1350 ; Output Parameters:
                                0AF2  1351 ;       TPA$L_PARAM(AP) - Address of PRM block for specified parameter
                                0AF2  1352 ;                         name if found.
                                0AF2  1353 ;       R0      - 0 => Name not found
                                0AF2  1354 ;                 1 => Name found
                                0AF2  1355 ;
                         003C   0AF2  1356 BOO$SEARCH::    .WORD   ^M<R2,R3,R4,R5> ;
                                0AF4  1357
00000000'EF    0A        E1    0AF4  1358          BBC     #BOOCMD$V_USEFILE, -
                       06          0AFB  1359                  BOO$GL_CMDOPT,5$        ; Skip count check if not USE <file>
                                0AFC  1360
      10 AC    03        91    0AFC  1361          CMPB    #3,TPA$L_TOKENCNT(AP)   ; Check for count of characters
                       32        18  0B00  1362          BGEQ    50$                     ; Exit if if not > 3
                                0B02  1363
54    00000000'EF    9E        0B02  1364 5$:      MOVAB   BOO$A_PRMBLK,R4         ; Set base of parameter blocks
                       64        D5  0B09  1365 10$:     TSTL    (R4)                    ; Check for end of list
                       02        12  0B0B  1366          BNEQ    30$                     ; Not yet
                       25        11  0B0D  1367          BRB     50$                     ; Symbol not found error
      55    16 A4    9E        0B0F  1368 30$:     MOVAB   PRM$T_NAME(R4),R5       ; Get pointer to name string
      85    10 AC    91        0B13  1369          CMPB    TPA$L_TOKENCNT(AP),(R5)+; Check for too many characters
                       08        14  0B17  1370          BGTR    35$                     ; Yes, cant be a match
            10 AC    29        0B19  1371          CMPC3   TPA$L_TOKENCNT(AP),-    ;
      65    14 BC              0B1C  1372                  @TPA$L_TOKENPTR(AP),(R5); Is this a match?
                       06        13  0B1F  1373          BEQL    40$                     ; Yes, return PRM pointer
      54    32 A4    9E        0B21  1374 35$:     MOVAB   PRM$C_LENGTH(R4),R4     ; Advance to nex parameter descriptor
                       E2        11  0B25  1375          BRB     10$                     ; and try another
         20 AC       54        D0  0B27  1376 40$:     MOVL    R4,TPA$L_PARAM(AP)      ; Return address of parameter block
      F4F4 CF        54        D0  0B2B  1377          MOVL    R4,BOO$GL_DOT           ; And save as dot
            50       01        D0  0B30  1378          MOVL    #1,R0                   ; Indicate success
                       04        0B33  1379          RET                             ;  and return
OD 00000000'EF    0A        E0    0B34  1380 50$:     BBS     #BOOCMD$V_USEFILE,BOO$GL_CMDOPT,60$; No message on 'USE filename'
                                0B3C  1381
                                0B3C  1382 .IF      NDF,CMDSW                       ; SYSBOOCMD
                                0B3C  1383          MSG     <-E-No such parameter>
                                0B3C  1384 .IFF                                     ; SYSGENCMD
         007C80F2 8F        DD  0B3C  1385          PUSHL   #SYSG$_NOPARAM          ; Set message
   00000000'GF    01        FB  0B42  1386          CALLS   #1,G^LIB$SIGNAL         ; Signal
                                0B49  1387 .ENDC
                                0B49  1388
            50       02        CE  0B49  1389 60$:     MNEGL   #2,R0                   ; Give unique error code
                       04        0B4C  1390          RET
                                0B4D  1391
                                0B4D  1392 ;
                                0B4D  1393 ; BOO$DOT - Use last parameter name if any
                                0B4D  1394 ;
                           0000 0B4D  1395 BOO$DOT:.WORD   0                       ; Null entry mask
   20 AC    F4D1 CF        D0  0B4F  1396          MOVL    BOO$GL_DOT,TPA$L_PARAM(AP)      ; Get dot address
                       02        12  0B55  1397          BNEQ    10$                     ; Have pointer
                       50        D4  0B57  1398          CLRL    R0                      ; Give error status
                       04        0B59  1399 10$:     RET                             ;
```

```
                              0B5A          .SBTTL  BOO$SETVALUE - Store parameter value
                              0B5A  ;
                              0B5A  ; Input Parameters:
                              0B5A  ;       TPA$L_PARAM - Address of parameter descriptor
                              0B5A  ;       TPA$L_NUMBER- Value to be checked and stored
                              0B5A  ;
                              0B5A  ; Output Parameters:
                              0B5A  ;       If value is within bounds set by parameter descriptor, the
                              0B5A  ;       value is moved to the address specified by the parameter descriptor
                              0B5A  ;       R0 - Completion status 0 => value out of allowable range
                              0B5A  ;                              1 => legal value successfully stored
                              0B5A  ;
                              0B5A  BOO$SETVALUE::                                    ;
                        0010  0B5A          .WORD   ^M<R4>                            ; Entry mask
                              0B5C
         00000000'8F     E2   0B5C          BBSS    #EXE$V_WRITESYSPARAMS,-           ; Set a value => write current needed
      00 00000000'GF          0B62                  G^EXE$GL_DYNAMIC_FLAGS,1$;
                              0B68  1$:
          54    20 AC    D0   0B68          MOVL    TPA$L_PARAM(AP),R4                ; Get pointer to parameter descriptor
                   10    E1   0B6C          BBC     #PRM$V_ASCII,-                    ; Ascii parameter?
             03 10 A4         0B6E                  PRM$L_FLAGS(R4),10$               ; If BC no continue
                 0080    31   0B71          BRW     65$                               ; Branch to error
                   01    DD   0B74  10$:    PUSHL   #1                                ; Assume good value
   25 00000000'EF   00   E0   0B76          BBS     #BOOCMD$V_NOCHECK,BOO$GL_CMDOPT,30$ ; Should values be checked
          50    08 A4    D0   0B7E          MOVL    PRM$L_MIN(R4),R0                  ; Get minimum allowable value
                   0D    19   0B82          BLSS    20$                               ; No minimum
          50    1C AC    D1   0B84          CMPL    TPA$L_NUMBER(AP),R0               ; Check input value
                   07    1E   0B88          BGEQU   20$                               ; Branch if above minimum
          1C AC    50    D0   0B8A          MOVL    R0,TPA$L_NUMBER(AP)               ; Use minimum value
             6E    02    CE   0B8E          MNEGL   #2,(SP)                           ; Note bad value
          50    0C A4    D0   0B91  20$:    MOVL    PRM$L_MAX(R4),R0                  ; Get maximum allowable value
                   0C    19   0B95          BLSS    30$                               ; Branch if no maximum
          1C AC    50    D1   0B97          CMPL    R0,TPA$L_NUMBER(AP)               ; Check for maximum
                   06    1E   0B9B          BGEQU   30$                               ; Continue if value legal
          1C AC    50    D0   0B9D          MOVL    R0,TPA$L_NUMBER(AP)               ; Limit to max value
                   6E    D4   0BA1          CLRL    (SP)                              ; Indicate error
          50    20 AC    D0   0BA3  30$:    MOVL    TPA$L_PARAM(AP),R0                ; Get address at which to store
             50    60    D0   0BA7          MOVL    PRM$L_ADDR(R0),R0                 ;
   50   00000000'EF40    9E   0BAA          MOVAB   BOO$A_SYSPARAM[R0],R0             ; Add present base of parameters
     50   00000000'8F    C2   0BB2          SUBL    #BOO$A_SYSPARAM,R0                ; And subtract link-time base
      10 A4    1000 8F   B3   0BB9          BITW    #PRM$M_NEG,PRM$L_FLAGS(R4)        ; Check for negative
                   05    13   0BBF          BEQL    35$                               ; No
        1C AC    1C AC   CE   0BC1          MNEGL   TPA$L_NUMBER(AP),TPA$L_NUMBER(AP)         ; Complement
          51    15 A4    9A   0BC6  35$:    MOVZBL  PRM$B_POS(R4),R1                  ; Get position
 60    14 A4  51    1C AC    F0   0BCA          INSV    TPA$L_NUMBER(AP),R1,PRM$B_SIZE(R4),(R0); Set value in field
             50    8E    D0   0BD1  40$:    MOVL    (SP)+,R0                          ; Get completion status
                   11    19   0BD4          BLSS    60$                               ; Low value limit
                0D 50    E8   0BD6          BLBS    R0,50$                            ; Success, return
                              0BD9
                              0BD9  .IF     NDF,CMDSW                 ; SYSBOOCMD
                              0BD9
                              0BD9          MSG     <-W-Value set to maximum>
                              0BD9  50$:    RET                                       ; and return
                              0BD9  60$:    MSG     <-W-Value set to minimum>
                              0BD9          RET                                       ;
                              0BD9  65$:    MSG     <-E-Parameter is not numeric type>
                              0BD9          RET
```

```
                                  0BD9
                                  0BD9  .IFF                                          ; SYSGENCMD
                                  0BD9
                    16 A4    9F   0BD9           PUSHAB  PRM$T_NAME(R4)               ; Address of parameter name
                       01    DD   0BDC           PUSHL   #1                           ; Number of FAO param's
              007C9028 8F    DD   0BDE           PUSHL   #SYSG$_SETMAX                ; Error status
                       19    11   0BE4           BRB     70$
                             04   0BE6  50$:     RET
                                  0BE7
                    16 A4    9F   0BE7  60$:     PUSHAB  PRM$T_NAME(R4)               ; Address of parameter name
                       01    DD   0BEA           PUSHL   #1                           ; Number of FAO param's
              007C9030 8F    DD   0BEC           PUSHL   #SYSG$_SETMIN                ; Error status
                       00    11   0BF2           BRB     65$                          ;
                    16 A4    9F   0BF4  65$:     PUSHAB  PRM$T_NAME(R4)               ; Address of parameter name
                       01    DD   0BF7           PUSHL   #1                           ; Number of FAO param's
              007C8112 8F    DD   0BF9           PUSHL   #SYSG$_NOTASCII              ; Error status
          00000000'GF  03    FB   0BFF  70$:     CALLS   #3,G^LIB$SIGNAL              ; Signal
                 50    01    D0   0C06           MOVL    #SS$_NORMAL,R0               ; Set success
                             04   0C09           RET                                  ; and return
                                  0C0A
                                  0C0A  .ENDC
                                  0C0A
                                  0C0A  ;
                                  0C0A  ; Set to default value
                                  0C0A  ;
                                  0C0A  BOO$SETDEF:                                   ;
                           00FC   0C0A           .WORD   ^M<R2,R3,R4,R5,R6,R7>        ;
              54    20 AC    D0   0C0C           MOVL    TPA$L_PARAM(AP),R4           ; Get address of parameter block
                       10    E0   0C10           BBS     #PRM$V_ASCII,-               ; Ascii parameter?
                 08 10 A4         0C12                   PRM$L_FLAGS(R4),10$          ; If BS yes
           1C AC    04 A4    D0   0C15           MOVL    PRM$L_DEFAULT(R4),TPA$L_NUMBER(AP); Set default as value
                     FF3F    31   0C1A           BRW     BOO$SETVALUE+2               ; Call routine to set the value
           14 AC    04 A4    9E   0C1D  10$:     MOVAB   PRM$L_DEFAULT(R4),TPA$L_TOKENPTR(AP); Set ptr to default string
              54    14 A4    9A   0C22           MOVZBL  PRM$B_SIZE(R4),R4            ; Get size in bits
       10 AC    54  FD 8F    78   0C26           ASHL    #-3,R4,TPA$L_TOKENCNT(AP); Set size in bytes
                      09'    11   0C2C           BRB     BOO$SETASCII+2               ; Call routine to set the default string
                                  0C2E
                                  0C2E  ;
                                  0C2E  ; Set acsii parameter to all blanks
                                  0C2E  ;
                                  0C2E  BOO$SETBLANK:
                           00FC   0C2E           .WORD   ^M<R2,R3,R4,R5,R6,R7>
                    10 AC    D4   0C30           CLRL    TPA$L_TOKENCNT(AP)           ; Set string count zero (null string)
                      02'    11   0C33           BRB     BOO$SETASCII+2               ; join common code
                                  0C35
```

```
                                0C35  1504          .SBTTL  BOO$SETASCII - Action routine to set ASCII parameter type
                                0C35  1505 ;
                                0C35  1506 ; Input Parameters:
                                0C35  1507 ;       TPA$L_PARAM(AP) - Address of parameter descriptor
                                0C35  1508 ;       TPA$L_TOKENCNT(AP) - Length of parsed string
                                0C35  1509 ;       TPA$L_TOKENPTR(AP) - Address of parsed string
                                0C35  1510 ;
                                0C35  1511 ; Output Parameters:
                                0C35  1512 ;       The parameter is checked to ensure it is ASCII type, then length
                                0C35  1513 ;       of the parsed string is compared to size of parameter.  If no
                                0C35  1514 ;       error, then parameter is set to new string.
                                0C35  1515 ;
                                0C35  1516 BOO$SETASCII::
                         00FC   0C35  1517          .WORD   ^M<R2,R3,R4,R5,R6,R7>
                                0C37  1518
    00000000'8F    E2           0C37  1519          BBSS    #EXE$V_WRITESYSPARAMS,- ; Set an value => write current needed
 00 00000000'GF                 0C3D  1520                  G^EXE$GL_DYNAMIC_FLAGS,1$;
                                0C43  1521 1$:
  00 04 AC    00    E5           0C43  1522          BBCC    #TPA$V_BLANKS,TPA$L_OPTIONS(AP),2$; Make blanks no longer significan
     56    20 AC    D0           0C48  1523 2$:      MOVL    TPA$L_PARAM(AP),R6      ; Get address of parameter block
     10 A6    10    E0           0C4C  1524          BBS     #PRM$V_ASCII,PRM$L_FLAGS(R6),-
                03                0C50  1525                  5$                      ; If set, then ASCII type
              00AE    31          0C51  1526          BRW     90$
     57    14 A6    9A           0C54  1527 5$:      MOVZBL  PRM$B_SIZE(R6),R7       ; Get size (in bits)
57   57    FD 8F    78           0C58  1528          ASHL    #-3,R7,R7               ; Convert size from bit to byte count
     57    10 AC    D1           0C5D  1529          CMPL    TPA$L_TOKENCNT(AP),R7   ; Compare with parsed string size
                03    1B          0C61  1530          BLEQU   10$                     ; If LEQU, then fits
              008D    31          0C63  1531          BRW     80$                     ; Else string too big
                01    DD          0C66  1532 10$:     PUSHL   #1                      ; Assume success
           5E   10    C2          0C68  1533          SUBL    #16,SP                  ; Make room for octaword buffer on stack
        53    14 AC    D0          0C6B  1534          MOVL    TPA$L_TOKENPTR(AP),R3   ; Get address of token
        54    10 AC    D0          0C6F  1535          MOVL    TPA$L_TOKENCNT(AP),R4   ; Get count of token
6E   57    20    63    54          2C  0C73  1536          MOVC5   R4,(R3),#^A/ /,R7,(SP)  ; New value on stack temporarily
   2D 00000000'EF    00          E0  0C79  1537          BBS     #BOOCMD$V_NOCHECK,BOO$GL_CMDOPT,30$ ; If checks disabled, branch
     08 A6    6E    57          2D  0C81  1538          CMPC5   R7,(SP),PRM$L_MIN(R6),-; Compare min value with parsed value
        08 A6    04              0C86  1539                  #4,PRM$L_MIN(R6)        ;
                0E    1E          0C89  1540          BGEQU   20$                     ; Branch if input is greater
        08 A6    04    2C          0C8B  1541          MOVC5   #4,PRM$L_MIN(R6),-      ; Set min value
     6E    57    08 A6           0C8F  1542                  PRM$L_MIN(R6),R7,(SP)   ;
        10 AE    02    CE          0C93  1543          MNEGL   #2,16(SP)               ; Ind error
                15    11          0C97  1544          BRB     30$
        0C A6    04    2D          0C99  1545 20$:     CMPC5   #4,PRM$L_MAX(R6),-      ; Compare max value with parsed value
     6E    57    0C A6           0C9D  1546                  PRM$L_MAX(R6),R7,(SP)   ;
                0B    1E          0CA1  1547          BGEQU   30$                     ; Branch if input is greater
        0C A6    04    2C          0CA3  1548          MOVC5   #4,PRM$L_MAX(R6),-      ; Set max value
     6E    57    0C A6           0CA7  1549                  PRM$L_MAX(R6),R7,(SP)   ;
           10 AE    D4          0CAB  1550          CLRL    16(SP)                  ; Ind error
        50    66    D0          0CAE  1551 30$:     MOVL    PRM$L_ADDR(R6),R0       ; Get address parameter
  50    00000000'EF40          9E  0CB1  1552          MOVAB   BOO$A_SYSPARAM[R0],R0   ; Add present base of parameters
    50    00000000'8F    C2          0CB9  1553          SUBL    #BOO$A_SYSPARAM,R0      ; And subtract link-time base
        60    6E    57    28      0CC0  1554          MOVC3   R7,(SP),(R0)            ; Set value in system
              5E    10    C0      0CC4  1555          ADDL    #16,SP                  ; Remove value from stack
              50    8E    D0      0CC7  1556          MOVL    (SP)+,R0                ; Get status
                    11    19      0CCA  1557          BLSS    60$                     ; If neg, value set to min
                 0D 50    E8      0CCC  1558          BLBS    R0,50$                  ; If LBS, success
                                0CCF  1559
                                0CCF  1560 .IF      NDF,CMDSW               ; SYSBOOCMD
```

```
                                G 11
                     0CCF  1561
                     0CCF  1562          MSG     <-W-Value set to maximum>
                     0CCF  1563 50$:     RET                             ; and return
                     0CCF  1564 60$:     MSG     <-W-Value set to minimum>
                     0CCF  1565          RET                             ;
                     0CCF  1566
                     0CCF  1567 .IFF                                     ; SYSGENCMD
                     0CCF  1568
           16 A4  9F 0CCF  1569          PUSHAB  PRM$T_NAME(R4)          ; Address of parameter name
              01  DD 0CD2  1570          PUSHL   #1                      ; Number of FAO param's
     007C9028 8F  DD 0CD4  1571          PUSHL   #SYSG$_SETMAX           ; Error status
              0C  11 0CDA  1572          BRB     70$
                  04 0CDC  1573 50$:     RET
                     0CDD  1574
           16 A4  9F 0CDD  1575 60$:     PUSHAB  PRM$T_NAME(R4)          ; Address of parameter name
              01  DD 0CE0  1576          PUSHL   #1                      ; Number of FAO param's
     007C9030 8F  DD 0CE2  1577          PUSHL   #SYSG$_SETMIN           ; Error status
00000000'GF   03  FB 0CE8  1578 70$:     CALLS   #3,G^LIB$SIGNAL         ; Signal
           50 01  D0 0CEF  1579 75$:     MOVL    #SS$_NORMAL,R0          ; Set success
                  04 0CF2  1580          RET                             ; and return
                     0CF3  1581
                     0CF3  1582 .ENDC
                     0CF3  1583
                     0CF3  1584          .IF NDF,CMDSW                   ; SYSBOOCMD
                     0CF3  1585
                     0CF3  1586 80$:
                     0CF3  1587          MSG     <-E-Specified string is too long>
                     0CF3  1588          RET
                     0CF3  1589
                     0CF3  1590 90$:
                     0CF3  1591          MSG     <-E-Parameter is not ASCII type>
                     0CF3  1592          RET
                     0CF3  1593
                     0CF3  1594          .IFF                            ; SYSGENCMD
                     0CF3  1595
                     0CF3  1596 80$:
     007C811A 8F  DD 0CF3  1597          PUSHL   #SYSG$_STRTOOLNG        ; Error status
00000000'GF   01  FB 0CF9  1598          CALLS   #1,G^LIB$SIGNAL         ; Output it
              ED  11 0D00  1599          BRB     75$
                     0D02  1600
                     0D02  1601 90$:
           16 A4  9F 0D02  1602          PUSHAB  PRM$T_NAME(R4)          ; Address of parameter name
              01  DD 0D05  1603          PUSHL   #1                      ; FAO arg count
     007C8112 8F  DD 0D07  1604          PUSHL   #SYSG$_NOTASCII         ; Error status
00000000'GF   03  FB 0D0D  1605          CALLS   #3,G^LIB$SIGNAL         ; Output it
              D9  11 0D14  1606          BRB     75$
                     0D16  1607
                     0D16  1608          .ENDC
```

```
                                0D16         .SBTTL  BOO$SHOVALUE - Action routine to show single value
                                0D16    ;
                                0D16    ; Input Parameters:
                                0D16    ;        TPA$L_PARAMETER(AP) - Address of parameter block
                                0D16    ;
                                0D16    BOO$SHOVALUE:                                 ;
                          003C  0D16         .WORD   ^M<R2,R3,R4,R5>                  ;
                                0D18    ;
                                0D18    ; Output header
                                0D18    ;
                                0D18    .IF      DF,CMDSW                             ; SYSGEN Only
                    10    E1    0D18         BBC     #BOOCMD$V_TERMINAL,-             ; Output header to terminals only
          18 00000000'EF        0D1A                 BOO$GL_CMDOPT,10$                ;
                                0D20         .ENDC
                                0D20
              009E 8F     28    0D20         MOVC3   #SDVHDRLEN,-
  00000000'EF F93E CF           0D24                 SDVHDR,RIO$AB_BUFFER            ; Move in string
  00000000'EF 009E 8F     B0    0D2C         MOVW    #SDVHDRLEN,RIO$GW_OUTLEN        ; Set length
                F2C8'     30    0D35         BSBW    RIO$OUTPUT_LINE                  ; Output line
                                0D38
          54    20 AC     D0    0D38    10$: MOVL    TPA$L_PARAM(AP),R4              ; Get address of parameter block
                 FBE6     30    0D3C         BSBW    BOO$SHOWV                        ; Show value
                          04    0D3F         RET                                      ; Return with BOO$SHOWV status
                                0D40
```

```
                                                I 11
                     0D40  1635          .SBTTL  BOO$SHOALL - Action routine to show all parameter values
                     0D40  1636 ;
                     0D40  1637 ; Input Parameters:
                     0D40  1638 ;        (AP)            Pointer to the TPARSE table
                     0D40  1639 ;        TPA$L_PARAM(AP) The mask of acceptable types
                     0D40  1640 ; Output Parameters:
                     0D40  1641 ;        All parameters except special parameters are displayed.
                     0D40  1642 ;
                     0D40  1643 BOO$SHOALL:                                  ;
                0FFC 0D40  1644          .WORD   ^M<R2,R3,R4,R5,R6,R7>      ;
                     0D42  1645
                     0D42  1646 .IF      DF,CMDSW                            ; SYSGEN ONLY
                     0D42  1647
            10   E1 0D42  1648          BBC     #BOOCMD$V_TERMINAL,-
 09 00000000'EF      0D44  1649                  BOO$GL_CMDOPT,5$            ; If terminal,
          7E   7C   0D4A  1650          CLRQ    -(SP)                       ;  clear the whole screen
00000000'GF   02   FB 0D4C  1651          CALLS   #2,G^SCR$ERASE_PAGE
                     0D53  1652 5$:
                     0D53  1653 ;
                     0D53  1654 ; Format "Parameters in use message"
                     0D53  1655 ;
                     0D53  1656          $FAO_S  CTRSTR = CTR_PARINUSE,-
                     0D53  1657                  OUTLEN = RIO$GW_OUTLEN,-
                     0D53  1658                  OUTBUF = RIO$AB_OUTBUF,-
                     0D53  1659                  P1 = BOO$GL_PARINUSE
    00000000'EF   DD 0D53                       PUSHL   BOO$GL_PARINUSE
    00000000'EF   7F 0D59                       PUSHAQ  RIO$AB_OUTBUF
    00000000'EF   3F 0D5F                       PUSHAW  RIO$GW_OUTLEN
        F8DC CF   7F 0D65                       PUSHAQ  CTR_PARINUSE
00000000'GF   04   FB 0D69                      CALLS   #$$T2,G^SYS$FAO
          F28D'   30 0D70  1660          BSBW    RIO$OUTPUT_LINE
                     0D73  1661
                     0D73  1662 .ENDC
                     0D73  1663
          009E 8F    28 0D73  1664          MOVC3   #SDVHDRLEN,-
00000000'EF   F8EB CF    0D77  1665                  SDVHDR,RIO$AB_BUFFER    ; Move in string
00000000'EF   009E 8F    B0 0D7F  1666          MOVW    #SDVHDRLEN,RIO$GW_OUTLEN ; Set length
              F275'      30 0D88  1667          BSBW    RIO$OUTPUT_LINE         ; Output line
                     0D8B  1668
                     0D8B  1669 .IF      DF,CMDSW                            ; SYSGEN ONLY
            10   E1 0D8B  1670          BBC     #BOOCMD$V_TERMINAL,-
  0B 00000000'EF      0D8D  1671                  BOO$GL_CMDOPT,7$            ; If terminal,
            18   DD 0D93  1672          PUSHL   #24                         ;  use only 24 lines
            05   DD 0D95  1673          PUSHL   #5                          ;  and scroll only the bottom portion
00000000'GF   02   FB 0D97  1674          CALLS   #2,G^SCR$SET_SCROLL         ;  and setup a scrolling region
                     0D9E  1675 7$:
                     0D9E  1676 .ENDC
                     0D9E  1677
54  00000000'EF   9E 0D9E  1678          MOVAB   BOO$A_PRMBLK,R4             ; Set starting parameter block address
     55   20 AC   D0 0DA5  1679          MOVL    TPA$L_PARAM(AP),R5          ; Set mask of acceptable types
                     0DA9  1680 ;
                     0DA9  1681 ; Loop through all parameters
                     0DA9  1682 ;
            64   D5 0DA9  1683 10$:     TSTL    (R4)                        ; Check for end of list
            1C   13 0DAB  1684          BEQL    50$                         ;  yes, done
      06 55   1F   E0 0DAD  1685          BBS     #PRM$V_ALL,R5,20$           ; Branch if SHOW/ALL
      10 A4   55   D3 0DB1  1686          BITL    R5,PRM$L_FLAGS(R4)          ; Is this one to output?
```

```
                 0C  13  0DB5  1687          BEQL   40$                     ; No, try another
                 07  E1  0DB7  1688 20$:     BBC    #PRM$V_SPECIAL,-        ; Yes, is it a special parameter?
           04 10 A4      0DB9  1689                 PRM$L_FLAGS(R4),30$     ; Branch if not
           03 55 07  E1  0DBC  1690          BBC    #PRM$V_SPECIAL,R5,40$   ; It's special - branch if unasked for
                         0DC0  1691 30$:
               FB62  30  0DC0  1692          BSBW   BOO$SHOWV               ; Display values
           54  32 A4 9E  0DC3  1693 40$:     MOVAB  PRM$C_LENGTH(R4),R4     ; Next parameter block
                 E0  11  0DC7  1694          BRB    10$                     ;
                         0DC9  1695 50$:
                         0DC9  1696
                         0DC9  1697 .IF      DF,CMDSW
                 10  E1  0DC9  1698          BBC    #BOOCMD$V_TERMINAL,-
       0B 00000000'EF    0DCB  1699                 BOO$GL_CMDOPT,60$       ; If terminal,
                 18  DD  0DD1  1700          PUSHL  #24                     ;  Use only 24 lines
                 01  DD  0DD3  1701          PUSHL  #1                      ;  and scroll only the bottom portion
     00000000'GF 02  FB  0DD5  1702          CALLS  #2,G^SCR$SET_SCROLL     ;  and setup a scrolling region
                         0DDC  1703 60$:
                         0DDC  1704 .ENDC
                     04  0DDC  1705          RET                            ;
```

```
                                ODDD  1707 ;
                                ODDD  1708 ; Set name of startup command file
                                ODDD  1709 ;
                                ODDD  1710 BOO$SETSTART:                                                ;
                          00FC  ODDD  1711          .WORD   ^M<R2,R3,R4,R5,R6,R7>                       ;
                                ODDF  1712
          00000000'8F     E2    ODDF  1713          BBSS    #EXE$V_WRITESYSPARAMS,- ; Set startup name => write current needed
       00 00000000'GF           ODE5  1714                  G^EXE$GL_DYNAMIC_FLAGS,1$;
                                ODEB  1715 1$:
    56  00000000'EF     9E    ODEB  1716          MOVAB   L^EXE$GT_STARTUP,R6     ; Point to slot for startup file name
                  50    D4    ODF2  1717          CLRL    R0                      ; Assume error
    00000000'EF   1F    91    ODF4  1718          CMPB    #31,BOO$GB_FILELEN      ; Check for fit
                  01    18    ODFB  1719          BGEQ    10$                     ; Continue if legal size
                        04    ODFD  1720          RET                             ;
    50  00000000'EF     9A    ODFE  1721 10$:     MOVZBL  BOO$GB_FILELEN,R0       ; Get count
              86  50    90    0E05  1722          MOVB    R0,(R6)+                ; Set count for string
66  00000000'FF  50    28    0E08  1723          MOVC3   R0,@BOO$GL_FILEADDR,(R6); Set file name
                  50    01    3C    0E10  1724          MOVZWL  #1,R0                   ; Return success indication
                        04    0E13  1725          RET                             ;
                              0E14  1726
```

```
                                                L 11
0E14  1728          .IF     NDF,CMDSW                   ; SYSBOOCMD ONLY
0E14  1729
0E14  1730          .SBTTL  BOO$MSGOUT - Output message
0E14  1731 ;
0E14  1732 ; Calling Sequence:
0E14  1733 ;        BSBW    BOO$MSGOUT
0E14  1734 ;        .ASCIZ  message_string
0E14  1735 ;
0E14  1736 BOO$MSGOUT::                                 ;
0E14  1737          CLRQ    -(SP)                       ; Null read buffer
0E14  1738          PUSHL   8(SP)                       ; Address of string
0E14  1739          CALLS   #3,L^BOO$READPROMPT         ; Output string
0E14  1740          LOCC    #0,#64000,@(SP)+            ; Find end of string
0E14  1741          MOVL    #1,R0                       ; Set success code
0E14  1742          JMP     1(R1)                       ; Return to caller
0E14  1743
0E14  1744 ;+
0E14  1745 ; This routine is in RMSCONIO for SYSGEN, is used here to map SYSBOOT
0E14  1746 ; calls to this routine into calls to BOO$READPROMPT.
0E14  1747 ;
0E14  1748 ; Inputs:
0E14  1749 ;        RIO$GW_OUTLEN - length of string to output
0E14  1750 ;        RIO$AB_BUFFER - buffer to output
0E14  1751 ;-
0E14  1752
0E14  1753 RIO$OUTPUT_LINE::
0E14  1754
0E14  1755          MOVQ    R1,-(SP)                    ; Save R1,R2
0E14  1756          MOVZWL  RIO$GW_OUTLEN,R1            ; Set length
0E14  1757          MOVAB   RIO$AB_BUFFER,R2            ; Set address
0E14  1758          MOVAB   (R2)[R1],R1                 ; Set address of end of string
0E14  1759          MOVL    #^X00000A0D,(R1)            ; Set CR, LF, zero byte at end
0E14  1760
0E14  1761          CLRQ    -(SP)                       ; Null read buffer
0E14  1762          PUSHL   R2                          ; Address of string
0E14  1763          CALLS   #3,L^BOO$READPROMPT         ; Output string
0E14  1764
0E14  1765          MOVQ    (SP)+,R1                    ; Restore R1,R2
0E14  1766          RSB                                 ; Return
0E14  1767
0E14  1768          .PAGE
0E14  1769          .SBTTL  DUMMY COMMAND ROUTINES FOR COMMANDS NOT IN SYSBOOT
0E14  1770 BOO$SET_OUTPUT::
0E14  1771 BOO$USEACT::
0E14  1772 SYS$ASCTOID::
0E14  1773 SYS$FILESCAN::
0E14  1774
0E14  1775          .WORD   0                           ; Null entry mask
0E14  1776          MSG     <-E-Syntax error>           ; SYSBOOT error message
0E14  1777          MOVL    #1,R0                       ;
0E14  1778          RET                                 ;
0E14  1779
0E14  1780          .ENDC   ; End of SYSBOOT conditional code
0E14  1781
0E14  1782          .END                                ;
```

```
$$$AST                          =  00000000
$$$CNT                          =  00000007
$$$FLG                          =  FFFFFFFF
$$$KEY                          =  00000074
$$$KFG                          =  FFFFFFFF
$$$MOD                          =  00000001
$$$TMP                          =  000003C2  R     04
$$KEYTAB                        =  00000000  R     03
$$T2                            =  00000004
ADAPTER                            000005B9  R     02
ADAP_STR                           000005D5  R     02
ADAP_STR2                          000005CD  R     02
ADPCMD                             00000294  R     02
ASCII                              000000EC  R     02
ASCSTR                             000005CE  R     06
AUTOCONFIG                         000002A6  R     02
AUTOOPT                            000002CC  R     02
BLANKS                             00000579  R     06
BOO$ADAPTER_NAME                   ********    X   02
BOO$ADAP_LETTER                    ********    X   02
BOO$A_PRMBLK                       ********    X   06
BOO$A_SYSPARAM                     ********    X   06
BOO$CHECK                          00000AE4  R     06
BOO$CONADP                         ********    X   02
BOO$CONAUNIT                       ********    X   02
BOO$CONCNUM                        ********    X   02
BOO$CONCREG                        ********    X   02
BOO$CONCSROFFSET                   ********    X   02
BOO$CONCVEC                        ********    X   02
BOO$CONDRVNAM                      ********    X   02
BOO$CONFIGALL                      ********    X   02
BOO$CONFIGONE                      ********    X   02
BOO$CONFIGURE                      ********    X   02
BOO$CONNECT                        ********    X   02
BOO$CONNLADP                       ********    X   02
BOO$CONRESET                       ********    X   02
BOO$CONSOLE                        ********    X   02
BOO$CONSYSID_HIGH                  ********    X   02
BOO$CONSYSID_LOW                   ********    X   02
BOO$CONUNITS                       ********    X   02
BOO$CONVECOFFSET                   ********    X   02
BOO$CREATE                         ********    X   02
BOO$CRECONTIG                      ********    X   02
BOO$CRENCONTIG                     ********    X   02
BOO$C_COMBUFSZ                  =  000000C8  G
BOO$C_COMSTRLEN                 =  00000400  G
BOO$DEVNAME                        ********    X   02
BOO$DOT                            00000B4D  R     06
BOO$FILESIZE                       ********    X   02
BOO$FILESPEC                       000008A6  RG    06
BOO$GB_FILELEN                     ********    X   06
BOO$GETPARAM                       00000703  RG    06
BOO$GIVEHELP                       ********    X   02
BOO$GL_CMDOPT                      00000000  RG    05
BOO$GL_DOT                         00000024  RG    06
BOO$GL_FILEADDR                    ********    X   06
BOO$GL_PARINUSE                    ********    X   06
BOO$GQ_FILDESC                     00000028  RG    06
BOO$GT_COMBUF                      00000070  RG    06
BOO$GT_COMSTR                      00000138  RG    06
BOO$GT_CURRENT                     ********    X   06
BOO$GT_DEFAULT                     ********    X   06
BOO$GT_FILNAME                     00000030  RG    06
BOO$GT_PROMPT                      ********    X   06
BOO$GT_SYSNAME                     00000538  RG    06
BOO$GT_SYSPARNAME                  0000054B  RG    06
BOO$INPUT_FILE                     ********    X   02
BOO$INSTALL                        ********    X   02
BOO$LOAD                           ********    X   02
BOO$MAKLIST                        ********    X   02
BOO$MSCP_ARG                       ********    X   02
BOO$MSCP_RESET                     ********    X   02
BOO$NOCHECK                        00000AD6  R     06
BOO$NOCHKPNT                       ********    X   02
BOO$NO_RESET                       ********    X   02
BOO$OUTPUT_FILE                    ********    X   02
BOO$READPROMPT                     ********    X   06
BOO$RELOAD                         ********    X   02
BOO$RESETLIST                      ********    X   02
BOO$RESET_ADAP                     ********    X   02
BOO$RESET_COMMAND                  ********    X   02
BOO$RESET_IO                       ********    X   02
BOO$SEARCH                         00000AF2  RG    06
BOO$SETASCII                       00000C35  RG    06
BOO$SETBLANK                       00000C2E  R     06
BOO$SETDEF                         00000C0A  R     06
BOO$SETFILNAM                      ********      X 02
BOO$SETPGFL                        ********      X 02
BOO$SETSTART                       00000DDD  R     06
BOO$SETVALUE                       00000B5A  RG    06
BOO$SET_OUTPUT                     ********      X 02
BOO$SET_TR                         ********      X 02
BOO$SHOALL                         00000D40  R     06
BOO$SHOCONFIG                      ********      X 02
BOO$SHODEV                         ********      X 02
BOO$SHODEV_ALL                     ********      X 02
BOO$SHONAMES                       00000A42  R     06
BOO$SHOSTART                       00000AAD  R     06
BOO$SHOVALUE                       00000D16  R     06
BOO$SHOWV                          00000925  R     06
BOO$SHOW_ADAPTER                   ********      X 02
BOO$SHOW_UNIBUS                    ********      X 02
BOO$T_DYNAMIC                      00000564  R     06
BOO$T_NODYNAMIC                    00000566  R     06
BOO$USEACT                         ********      X 02
BOO$USECUR                         000008EA  RG    06
BOO$USEFILE                        ********      X 02
BOO$WRTACT                         ********      X 02
BOO$WRTCUR                         ********      X 02
BOO$WRTFILE                        ********      X 02
BOOCMD$M_AUTOLOG                =  00001000
BOOCMD$M_CONT                   =  00000100
BOOCMD$M_DEFAULT                =  00000200
BOOCMD$M_DISHEX                 =  00000800
```

```
BOOCMD$M_EXCLUDE        = 00000080
BOOCMD$M_LOGICAL        = 00100000
BOOCMD$M_NOCHECK        = 00000001
BOOCMD$M_REMOTE         = 00080000
BOOCMD$M_SELECT         = 00000040
BOOCMD$M_SETOUTPUT      = 00008000
BOOCMD$M_TERMINAL       = 00010000
BOOCMD$M_USEFILE        = 00000400
BOOCMD$V_CONT           = 00000008
BOOCMD$V_DEFAULT        = 00000009
BOOCMD$V_DISHEX         = 0000000B
BOOCMD$V_NOCHECK        = 00000000
BOOCMD$V_TERMINAL       = 00000010
BOOCMD$V_USEFILE        = 0000000A
BUFFER_SIZE             = 00000100
CMDSW                   = 00000001
CONAUNIT                  000006BF R      02
CONCNUMVEC                000006A7 R      02
CONCREG                   0000069B R      02
CONCSROFF                 000006BB R      02
CONCVECTOR                000006A3 R      02
CONECTCMD                 00000398 R      02
CONECTOPT                 0000061D R      02
CONFIGALL                 000002C2 R      02
CONFIGCMD                 00000310 R      02
CONFIG_LIST               0000031E R      02
CONFIG_OPT                00000328 R      02
CONOPT                    000003AC R      02
CONSOLCMD                 000003B6 R      02
CONSOLOPT                 000003C0 R      02
CONSYSID_HI               000006B3 R      02
CONSYSID_LO               000006AF R      02
CONUNITS                  000006AB R      02
CONVECOFF                 000006B7 R      02
CR                      = 0000000D
CREATECMD                 00000368 R      02
CREATE_QUAL               0000037C R      02
CREOPT                    0000036A R      02
CTRLSTR                   0000057E R      06
CTR_PARINUSE              00000645 R      06
CUR_BLANKS                00000567 R      06
DEV_OR_DRIV               00000541 R      02
DISABLCMD                 000000A2 R      02
ENABLCMD                  000000AC R      02
EXE$A_SYSPARAM            ********    X   06
EXE$C_SYSPARSZ            ********    X   06
EXE$GL_DYNAMIC_FLAGS      ********    X   06
EXE$GL_TODR               ********    X   06
EXE$GQ_TODCBASE           ********    X   06
EXE$GT_STARTUP            ********    X   06
EXE$V_WRITESYSPARAMS      ********    X   06
EXIT                      000008A2 R      06
FF                      = 0000000C
FILESPEC                  00000266 R      02
GEN$SHARE                 ********    X   02
GEN$SHR_CEFCNT            ********    X   02
GEN$SHR_CEFMAX            ********    X   02
GEN$SHR_GBLCNT            ********    X   02
GEN$SHR_GBLMAX            ********    X   02
GEN$SHR_INIT              ********    X   02
GEN$SHR_MBXCNT            ********    X   02
GEN$SHR_MBXMAX            ********    X   02
GEN$SHR_MEMNAME           ********    X   02
GEN$SHR_POOLC             ********    X   02
GEN$SHR_POOLS             ********    X   02
GEN$SHR_PRQCNT            ********    X   02
GEN$SHR_RESET             ********    X   02
GEN$SHR_START             ********    X   02
GEN$SHR_UNIT              ********    X   02
GETDATA                   00000A22 R      06
HEXNUM                    0000027E R      02
HEXQUAL                   00000210 R      02
HEXQUAL2                  00000206 R      02
HEXSTR                    000005A6 R      06
INPUT                     0000033C R      02
INS1                      000003EC R      02
INSTALCMD                 000003E2 R      02
INS_EXIT                  0000041E R      02
INS_PAGE                  00000416 R      02
KEYTBL                    00000000 RG     03
LF                      = 0000000A
LIB$SIGNAL                ********    X   06
LIB$TPARSE                ********    X   06
LIB$_SYNTAXERR            ********    X   06
LOADT                     00000691 R      02
LOADARGCNT                00000507 R      02
LOADCMD                   00000422 R      02
LOADP1                    0000050B R      02
MSCP                      0000043C R      02
MSCPCMD                   00000436 R      02
MSCPOPT                   00000446 R      02
MSCP_BUFFER               000004CF R      02
MSCP_FRACTION             000004FF R      02
MSCP_HOSTS                000004DF R      02
MSCP_PACKET               000004D7 R      02
MSCP_PRIORITY             000004EF R      02
MSCP_SMALL                000004F7 R      02
MSCP_TIME_OUT             000004E7 R      02
NCTRCSTR                  0000060E R      06
NUMBER                    0000026E R      02
OUTPUT                    0000034C R      02
PARMBLK                   00000000 R      06
PARSE                     000007BA R      06
POOLCNT                   0000074D R      02
POOLSIZE                  00000755 R      02
PRM$B_POS               = 00000015
PRM$B_SIZE              = 00000014
PRM$C_LENGTH            = 00000032
PRM$L_ADDR              = 00000000
PRM$L_DEFAULT           = 00000004
PRM$L_FLAGS             = 00000010
PRM$L_MAX               = 0000000C
PRM$L_MIN               = 00000008
PRM$M_ACP               = 00000008
```

```
PRM$M_ALL                       = 80000000
PRM$M_CLUSTER                   = 00008000
PRM$M_DYNAMIC                   = 00000001
PRM$M_JBC                       = 00000010
PRM$M_LGI                       = 00020000
PRM$M_MAJOR                     = 00000400
PRM$M_NEG                       = 00001000
PRM$M_PQL                       = 00000800
PRM$M_RMS                       = 00000020
PRM$M_SCS                       = 00004000
PRM$M_SPECIAL                   = 00000080
PRM$M_SYS                       = 00000040
PRM$M_SYSGEN                    = 00000004
PRM$M_TTY                       = 00002000
PRM$T_NAME                      = 00000016
PRM$T_UNIT                      = 00000026
PRM$V_ALL                       = 0000001F
PRM$V_ASCII                     = 00000010
PRM$V_DYNAMIC                   = 00000000
PRM$V_SPECIAL                   = 00000007
PRQCNT                            00000759 R      02
READCMD                           00000705 R      06
READLINE                          00000713 R      06
RELOADCMD                         0000050F R      02
RESET                             0000035C R      02
RIO$AB_BUFFER                     ********   X    06
RIO$AB_OUTBUF                     ********   X    06
RIO$GW_OUTLEN                     ********   X    06
RIO$OUTPUT_LINE                   ********   X    06
SAVE_TODCBASE                     00000004 R      05
SAVE_TODR                         0000000C R      05
SCR$ERASE_PAGE                    ********   X    06
SCR$SET_SCROLL                    ********   X    06
SCTRLSTR                          00000621 R      06
SDVHDR                            00000665 R      06
SDVHDRLEN                       = 0000009E
SELECTLIST                        000002FA R      02
SEPARATOR                         00000282 R      02
SETCMD                            000000B6 R      02
SETOUTPUT                         0000010C R      02
SETSPEC                           000000E0 R      02
SETSTARTUP                        00000120 R      02
SHARECMD                          00000517 R      02
SHARECMDOPT                       0000051D R      02
SHAREOPT                          000006C7 R      02
SHOCMD                            00000134 R      02
SHOSWITCH                         00000136 R      02
SHOWCON                           00000565 R      02
SHOWCONOPT                        0000057D R      02
SHOWCON_LOOP                      0000056F R      02
SHOWONE                           000001FE R      02
SHOW_UNIBUS                       00000597 R      02
SHRCEFCNT                         0000073D R      02
SHRCEFMAX                         00000749 R      02
SHRGBLCNT                         00000735 R      02
SHRGBLMAX                         00000741 R      02
SHRMBXCNT                         00000739 R      02
SHRMBXMAX                         00000745 R      02
SHROPT                            00000537 R      02
SHRSTART                          00000761 R      02
SS$_NORMAL                      = 00000001
STATE1                            00000000 RG     02
SYMBOL                            000000F6 R      02
SYS$FAO                           ********   X    06
SYSG$LOAD_TT_STR                  ********   X    02
SYSG$_NOPARAM                   = 007C80F2
SYSG$_NOTASCII                  = 007C8112
SYSG$_SETMAX                    = 007C9028
SYSG$_SETMIN                    = 007C9030
SYSG$_STRTOOLNG                 = 007C8  A
SYSG$_SYNTAX                    = 007C809A
TERMINALCMD                       000005E5 R      02
TPA$B_CHAR                      = 00000018
TPA$K_COUNTO                    = 00000008
TPA$K_LENGTHO                   = 00000024
TPA$L_COUNT                     = 00000000
TPA$L_NUMBER                    = 0000001C
TPA$L_OPTIONS                   = 00000004
TPA$L_PARAM                     = 00000020
TPA$L_STRINGCNT                 = 00000008
TPA$L_STRINGPTR                 = 0000000C
TPA$L_TOKENCNT                  = 00000010
TPA$L_TOKENPTR                  = 00000014
TPA$M_ABBREV                    = 00000002
TPA$M_BLANKS                    = 00000001
TPA$V_BLANKS                    = 00000000
TPA$_ALPHA                      = 000001EE
TPA$_ANY                        = 000001ED
TPA$_BLANK                      = 000001F2
TPA$_DECIMAL                    = 000001F3
TPA$_DIGIT                      = 000001EF
TPA$_EOS                        = 000001F7
TPA$_EXIT                       = FFFFFFFF
TPA$_FAIL                       = FFFFFFFE
TPA$_FILESPEC                   = 000001EA
TPA$_HEX                        = 000001F5
TPA$_IDENT                      = 000001EC
TPA$_KEYWORD                    = 00000100
TPA$_LAMBDA                     = 000001F6
TPA$_MAXKEY                     = 000000DC
TPA$_OCTAL                      = 000001F4
TPA$_STRING                     = 000001F0
TPA$_SUBXPR                     = 000001F8
TPA$_SYMBOL                     = 000001F1
TPA$_UIC                        = 000001EB
USEACT                            0000024E R      02
USECMD                            00000222 R      02
USECUR                            00000244 R      02
USEDEF                            00000258 R      02
VALUE                             0000028A R      02
WRTACT                            00000613 R      02
WRTCMD                            000005F1 R      02
WRTCUR                            00000609 R      02
```

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
_LIB$STATE$	00000765 (1893.)	02 (2.)	PIC USR CON REL LCL SHR EXE RD NOWRT NOVEC BYTE
_LIB$KEY0$	000000EA (234.)	03 (3.)	PIC USR CON REL LCL SHR EXE RD NOWRT NOVEC WORD
_LIB$KEY1$	000003C9 (969.)	04 (4.)	PIC USR CON REL LCL SHR EXE RD NOWRT NOVEC WORD
NONPAGED_DATA	00000010 (16.)	05 (5.)	NOPIC USR CON REL LCL NOSHR NOEXE RD WRT NOVEC QUAD
SYSBOOCMD	00000E14 (3604.)	06 (6.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC LONG

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.10	00:00:01.19
Command processing	111	00:00:00.81	00:00:05.59
Pass 1	1193	00:01:28.43	00:02:43.16
Symbol table sort	0	00:00:01.99	00:00:03.50
Pass 2	514	00:00:19.10	00:00:30.89
Symbol table output	44	00:00:00.30	00:00:00.77
Psect synopsis output	6	00:00:00.03	00:00:00.11
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	1899	00:01:50.76	00:03:25.21

The working set limit was 2000 pages.
475517 bytes (929 pages) of virtual memory were used to buffer the intermediate code.
There were 70 pages of symbol table space allocated to hold 1240 non-local and 84 local symbols.
1783 source lines were read in Pass 1, producing 70 object records in Pass 2.
40 pages of virtual memory were used to define 33 macros.

Macro library statistics

Macro library name	Macros defined
_255DUA28:[BOOTS.OBJ]BOOTS.MLB;1	1
_255DUA28:[SYS.OBJ]LIB.MLB;1	5
_255DUA28:[SYSLIB]STARLET.MLB;2	13
TOTALS (all libraries)	19

1365 GETS were required to define 19 macros.

There were no errors, warnings or information messages.

```
MACRO/LIS=LIS$:SYSGENCMD/OBJ=OBJ$:SYSGENCMD MSRC$:CMDSW/UPDATE=(ENH$:CMDSW)+MSRC$:SYSBOOCMD/UPDATE=(ENH$:SYSBOOCMD)+EXECML$/LIB+LIB$
```

SYSGENMN
LIS

VMB
LIS

SYSGEN
LIS

SYSGTTSTR
LIS

SYSGENCMD
LIS

SYSGTERM
LIS

TS8BOOTIO
LIS