F 5

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

```
**FILE**ID**SYSBOOCMD                                    B 11




  SSSSSSSS  YY      YY    SSSSSSSS  BBBBBBBB      OOOOOO      OOOOOO      CCCCCCCC  MM      MM  DDDDDDDD
  SSSSSSSS  YY      YY    SSSSSSSS  BBBBBBBB      OOOOOO      OOOOOO      CCCCCCCC  MM      MM  DDDDDDDD
SS          YY      YY  SS          BB      BB  OO      OO  OO      OO  CC          MMMM  MMMM  DD      DD
SS          YY      YY  SS          BB      BB  OO      OO  OO      OO  CC          MMMM  MMMM  DD      DD
SS            YY  YY    SS          BB      BB  OO      OO  OO      OO  CC          MM  MM  MM  DD      DD
SS            YY  YY    SS          BB      BB  OO      OO  OO      OO  CC          MM  MM  MM  DD      DD
  SSSSSS        YY        SSSSSS    BBBBBBBB    OO      OO  OO      OO  CC          MM      MM  DD      DD
  SSSSSS        YY        SSSSSS    BBBBBBBB    OO      OO  OO      OO  CC          MM      MM  DD      DD
        SS      YY              SS  BB      BB  OO      OO  OO      OO  CC          MM      MM  DD      DD
        SS      YY              SS  BB      BB  OO      OO  OO      OO  CC          MM      MM  DD      DD
        SS      YY              SS  BB      BB  OO      OO  OO      OO  CC          MM      MM  DD      DD      ....
        SS      YY              SS  BB      BB  OO      OO  OO      OO  CC          MM      MM  DD      DD      ....
SSSSSSSS        YY      SSSSSSSS    BBBBBBBB      OOOOOO      OOOOOO      CCCCCCCC  MM      MM  DDDDDDDD        ....
SSSSSSSS        YY      SSSSSSSS    BBBBBBBB      OOOOOO      OOOOOO      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
```

S
V

```
    (1)      188        PARSE TABLES
    (2)     1103        BOO$FILESPEC - Parse file spec
    (2)     1148        BOO$USECUR - Use parameters from current image
    (2)     1165        BOO$SHOWV - Routine to show one parameter value
    (2)     1323        BOO$NOCHECK - Disable value checking
    (2)     1332        BOO$NOCHECK - Disable value checking
    (2)     1344        BOO$SEARCH - Lookup parameter name
    (2)     1401        BOO$SETVALUE - Store parameter value
    (2)     1504        BOO$SETASCII - Action routine to set ASCII parameter type
    (2)     1610        BOO$SHOVALUE - Action routine to show single value
    (2)     1635        BOO$SHOALL - Action routine to show all parameter values
    (2)     1730        BOO$MSGOUT - Output message
    (2)     1769        DUMMY COMMAND ROUTINES FOR COMMANDS NOT IN SYSBOOT
```

```
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    ;       V03-026 WHM0011         Bill Matthews           01-aug-1984
```

```
0000  ;                 Fix SET/STARTUP bug from V03-024.
0000  ;
0000  ;         V03-025 WHM0010         Bill Matthews           23-Jul-1984
0000  ;                 Change MSCP qualifier /SMALL to /MINIMUM and /FRACTION to
0000  ;                 /MAXIMUM.
0000  ;
0000  ;         V03-024 WHM0009         Bill Matthews           19-Jun-1984
0000  ;                 Fixed LOAD<CR> accvio. Fixed SET ascii-parameter 0 bug.
0000  ;                 Now allow an optional : and = in SET/STARTUP filespec.
0000  ;                 Now allow optional : in device name of the CONNECT command.
0000  ;
0000  ;         V03-023 WHM0008         Bill Matthews           20-Apr-1984
0000  ;                 Fixed SET of an ascii parameter to DEFAULT bug.
0000  ;                 Removed USE CURRENT that read SYSGEN parameters from SYS.EXE.
0000  ;
0000  ;         V03-022 WHM0007         Bill Matthews           11-Apr-1984
0000  ;                 Removed the QUORUM command.
0000  ;
0000  ;         V03-021 WHM0006         Bill Matthews           04-Apr-1984
0000  ;                 Added support for sysgen ascii parameters longer than 4
0000  ;                 characters.
0000  ;                 Added support for a seperate default system parameter file.
0000  ;
0000  ;         V03-020 JLV0342         Jake VanNoy              3-APR-1984
0000  ;                 Add TERMINAL/ECHO command.
0000  ;
0000  ;         V03-019 WHM0005         Bill Matthews           14-Mar-1984
0000  ;                 Conditionally assembled TPARSE tables for SYSBOOT.
0000  ;                 Change ascii input specifier from %A to '"'.
0000  ;                 Output header for display of a single parameter value.
0000  ;
0000  ;         V03-018 WHM0004         Bill Matthews           13-Mar-1984
0000  ;                 Move definition of BOO$GL_LOAD_ARGS from this module
0000  ;                 to SYSGEN.MAR.
0000  ;
0000  ;         V03-017 WHM0003         Bill Matthews           23-Feb-1984
0000  ;                 Add support for loading and starting the MSCP server.
0000  ;
0000  ;         V03-016 WHM0002         Bill Matthews           01-Feb-1984
0000  ;                 Add support for SHOW/LGI.
0000  ;
0000  ;         V03-015 ACG0392         Andrew C. Goldstein,    19-Jan-1984  22:40
0000  ;                 Tie off SYS$FILESCAN for TPARSE use
0000  ;
0000  ;         V03-014 WHM0001         Bill Matthews           14-Dec-1983
0000  ;                 Add /REMOTE and /LOGICAL switches to the CONNECT CONSOLE command
0000  ;                 Add /VECTOR_OFFSET and /CSR_OFFSET to the CONNECT command
0000  ;
0000  ;         V03-013 WMC0013         Wayne Cardoza           01-Dec-1983
0000  ;                 Allow arbitrary ordering of install qualifiers
0000  ;
0000  ;         V03-012 JLV0311         Jake VanNoy             10-OCT-1983
0000  ;                 Fix SHOW/ALL to really SHOW/ALL.
0000  ;
0000  ;         V03-011 SRB0103         Steve Beckhardt         19-Sep-1983
0000  ;                 Added temporary QUORUM command.
0000  ;
```

```
0000   ;       V03-010 BLS0239         Benn Schreiber          13-Sep-1983
0000   ;               Use TPA$_SYMBOL for %A so that '$' and '_' are allowed.
0000   ;
0000   ;       V03-009 ACG0345         Andrew C. Goldstein,    1-Aug-1983  16:53
0000   ;               Add dummy SYS$ASCTOID routine for TPARSE
0000   ;
0000   ;       V03-008 MSH0005         Maryann Hinden          13-Jul-1983
0000   ;               Don't need to echo input anymore.
0000   ;               Set ascii parameters correctly if smaller than a longword.
0000   ;
0000   ;       V03-007 MSH0004         Maryann Hinde           24-Jun-1983
0000   ;               Change $BOODEF to $BOOCMDDEF.
0000   ;
0000   ;       V03-006 MSH0003         Maryann Hinden          10-Jun-1983
0000   ;               Use $BOODEF.
0000   ;
0000   ;       V03-005 MSH0002         Maryann Hinden          14-Apr-1983
0000   ;               Teach SYSGEN to speak ASCII.
0000   ;
0000   ;       V03-004 MSH0001         Maryann Hinden          24-Mar-1983
0000   ;               Preserve values for system time and base registers
0000   ;               across USE DEFAULT and USE CURRENT commands.
0000   ;
0000   ;       V03-003 DWT0086         David W. Thiel          22-Mar-1983
0000   ;               Add PRM$M_CLUSTER to SHOW/ALL mask.  Add
0000   ;               SHOW/CLUSTER.
0000   ;
0000   ;       V03-002 WMC0001         Wayne Cardoza           12-Aug-1982
0000   ;               Add support for the /checkpoint qualifier on install /page
0000   ;
0000   ;       V03-001 JLV0196         Jake VanNoy             17-MAR-1982
0000   ;               Add new parsing for CREATE. Add PRM$M_SCS, PRM$M_TTY
0000   ;               and PRM$M_SYSGEN to SHOW/ALL mask. Change BOO$SEARCH
0000   ;               to return no such parameter if a search for a zero
0000   ;               length parameter is passed in.
0000   ;
0000   ;--
0000
0000   ;
0000   ; Include files:
0000   ;
0000           $BOOCMDDEF                      ; Flag bits in command options longword
0000           $CLUBDEF                        ; Cluster block offsets
0000           $IPLDEF                         ; IPL defs
0000           $PRVDEF                         ; Privilege definitions
0000           $PCBDEF                         ; PCB offsets
0000           $PRMDEF                         ; Parameter descriptor definitions
0000           $SSDEF                          ; System messages
0000           $SYSGMSGDEF                     ; Sysgen messages
0000           $TPADEF                         ; Define TPARSE symbols
0000
0000   ;
0000   ; MACROS:
0000   ;
0000   ;
0000   ; Macro to print message
0000   ;
```

```
                    0000  ;       MSG     message_text
                    0000  ;
                    0000          .MACRO  MSG,STR                 ;
                    0000          BSBW    BOO$FACMSG              ;
                    0000          .ASCIZ  \'STR'\                 ;
                    0000          .ENDM   MSG                     ;
                    0000
                    0000  ;
                    0000  ; Equated Symbols:
                    0000  ;
          0000000D  0000          CR=13                           ; Character value for carriage return
          0000000C  0000          FF=12                           ; Character value for form feed
          0000000A  0000          LF=10                           ; Character value for line feed
          00000100  0000          BUFFER_SIZE=256
                    0000
```

```
0000            .SBTTL  PARSE TABLES
0000      ;
0000      ;     DEFINE COMMAND SYNTAX
0000      ;
0000
0000            $INIT_STATE     STATE1,KEYTBL   ;
0000            $STATE                          ;
0000            $TRAN   !DISABLCMD,TPA$_EXIT     ; Disable option command
0000            $TRAN   !ENABLCMD,TPA$_EXIT      ; Enable option command
0000            $TRAN   'HELP',TPA$_EXIT,BOO$GIVEHELP   ; Help command
0000            $TRAN   !SETCMD,TPA$_EXIT        ; Set specific value
0000            $TRAN   !SHOCMD,TPA$_EXIT        ; Show values
0000            $TRAN   !USECMD,TPA$_EXIT        ; Set background values
0000            $TRAN   'EXIT',TPA$_EXIT,,BOOCMD$M_CONT,BOO$GL_CMDOPT    ; Same as continue
0000
0000            .IF     NDF,CMDSW               ;SYSBOOT specific commands
0000            $TRAN   'CONTINUE',TPA$_EXIT,,BOOCMD$M_CONT,BOO$GL_CMDOPT ; Continue command
0000
0000            .IFF                            ;SYSGEN specific commands
0000            $TRAN   !ADPCMD,TPA$_EXIT,BOO$CONADP     ; Set adapter TR number
0000            $TRAN   !CONECTCMD,TPA$_EXIT,BOO$CONNECT ; Connect command
0000            $TRAN   !CREATECMD,TPA$_EXIT,BOO$CREATE ; Create dump/page/swap file
0000            $TRAN   !INSTALCMD,TPA$_EXIT,BOO$INSTALL; Install swap/page file
0000            $TRAN   !LOADCMD,TPA$_EXIT,BOO$LOAD      ; Load driver
0000            $TRAN   !RELOADCMD,TPA$_EXIT,BOO$RELOAD ; Reload driver
0000            $TRAN   !MSCPCMD,TPA$_EXIT,BOO$LOAD      ; Load and start the MSCP server
0000            $TRAN   !SHARECMD,TPA$_EXIT,GEN$SHARE ; Share command
0000            $TRAN   !WRTCMD,TPA$_EXIT                ; Write parameter file
0000            $TRAN   !AUTOCONFIG,TPA$_EXIT            ; Auto-configure command
0000            $TRAN   !CONFIGCMD,TPA$_EXIT,BOO$CONFIGURE
0000            $TRAN   !TERMINALCMD,TPA$_EXIT           ; terminal command
0000            .ENDC
0000
0000            $TRAN   TPA$_EOS,TPA$_EXIT      ; END OF LINE
0000
0000      ;
0000      ; Disable command
0000      ;
0000            $STATE  DISABLCMD               ; Disable command
0000            $TRAN   'DISABLE'               ; Command verb
0000            $STATE                          ;
0000            $TRAN   'CHECKS',TPA$_EXIT,BOO$NOCHECK  ; Disable value checking
0000      ;
0000      ;     Recognize ENABLE command
0000      ;
0000            $STATE  ENABLCMD                ; ENABLE command
0000            $TRAN   'ENABLE'                ; Command verb
0000            $STATE                          ;
0000            $TRAN   'CHECKS',TPA$_EXIT,BOO$CHECK    ;
0000
0000      ;
0000      ;     Recognize SET Command
0000      ;
0000            $STATE  SETCMD                  ; SET command
0000            $TRAN   'SET'                   ; Command verb
0000            $STATE                          ;
0000            $TRAN   '/',SETSPEC             ;
```

```
                  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          $ST'`-
                  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
0000              $TRAN   'ALL',HEXQUAL2,,,,PRM$M_ALL; SHOW/ALL
0000              $TRAN   'RMS',HEXQUAL2,,,,<PRM$M_RMS>; SHOW/RMS
0000              $TRAN   'SCS',HEXQUAL2,,,,<PRM$M_SCS>; SHOW/SCS
0000              $TRAN   'SPECIAL',HEXQUAL2,,,,<PRM$M_SPECIAL>; SHOW/SPECIAL
0000              $TRAN   'SYS',HEXQUAL2,,,,<PRM$M_SYS>; SHOW/SYS
0000              $TRAN   'GEN',HEXQUAL2,,,,<PRM$M_SYSGEN>; SHOW/GEN (Sysgen Parameters)
0000              $TRAN   'JOB',HEXQUAL2,,,,<PRM$M_JBC>; SHOW/JOB (Job controller)
0000              $TRAN   'PQL',HEXQUAL2,,,,<PRM$M_PQL>; SHOW/PQL (Process quota list)
0000              $TRAN   'TTY',HEXQUAL2,,,,<PRM$M_TTY>; SHOW/TTY
0000              $TRAN   'LGI',HEXQUAL2,,,,<PRM$M_LGI>; SHOW/LGI
0000              $TRAN   'CLUSTER',HEXQUAL2,,,,<PRM$M_CLUSTER>   ; SHOW/CL STER show cluster
0000              $TRAN   'NAMES',TPA$_EXIT,BOO$SHONAMES  ; SHOW/NAMES show parameter names
0000              $TRAN   'MAJOR',HEXQUAL2,,,,<PRM$M_MAJOR>       ; SHOW/MAJOR show major para
0000              $TRAN   'DYNAMIC',HEXQUAL2,,,,<PRM$M_DYNAMIC>; SHOW/DYNAMI. show dyn. params
0000              $TRAN   'STARTUP',TPA$_EXIT,BOO$SHOSTART      ; SHO/STARTUP Show startup file
0000              .IF     DF,CMDSW                      ;SYSGEN specific qualifiers
0000              $TRAN   'ADAPTER',TPA$_EXIT,BOO$SHOW_ADAPTER ; SHOW/ADAPTER
0000              $TRAN   'CONFIGURATION',SHOWCON,BOO$RESET_IO ; SHOW/CONFIGURATION
0000              $TRAN   !SHOW_UNIBUS,TPA$_EXIT           ; /UNIBUS
0000              $TRAN   !DEV_OR_DRIV,TPA$_EXIT           ; /DEVICES and /DRIVER
0000              .ENDC
0000
0000              $STATE  SHOWONE                   ; SHOW value_name
0000              $TRAN   TPA$_LAMBDA,TPA$_EXIT,BOO$SHOVALUE      ;
0000
0000              $STATE  HEXQUAL2
0000              $TRAN   !HEXQUAL,TPA$_EXIT,BOO$SHOALL
0000
0000              $STATE  HEXQUAL
0000              $TRAN   '/'
0000              $TRAN   TPA$_LAMBDA,TPA$_EXIT
0000              $STATE
0000              $TRAN   'HEX',TPA$_EXIT,,BOOCMD$M_DISHEX,BOO$GL_CMDOPT
0000      ;
0000      ; Recognize USE command
0000      ;
0000              $STATE  USECMD                ;
0000              $TRAN   'USE',,,BOOCMD$M_USEFILE,BOO$GL_CMDOPT ;
0000              $STATE                        ;
0000              $TRAN   !USECUR               ;
0000              $TRAN   !USEACT               ;
0000              $TRAN   !USEDEF               ;
0000              $TRAN   !FILESPEC,,BOO$USEFILE  ;
0000              $STATE                        ;
0000              $TRAN   TPA$_LAMBDA,TPA$_EXIT  ;
0000
0000              $STATE  USECUR                ; USE CURRENT
0000              $TRAN   'CURRENT'
0000              $STATE
0000              $TRAN   TPA$_EOS,TPA$_EXIT,BOO$USECUR
0000
0000              $STATE  USEACT                ; USE ACTIVE
0000              $TRAN   'ACTIVE'
0000              $STATE
0000              $TRAN   TPA$_EOS,TPA$_EXIT,BOO$USEACT
0000
```

```
0000            $STATE  USEDEF                  ; USE DEFAULT
0000            $TRAN   'DEFAULT'
0000            $STATE
0000            $TRAN   TPA$_EOS,TPA$_EXIT,,BOOCMD$M_DEFAULT,BOO$GL_CMDOPT      ;
0000
0000    ;
0000    ; File Specification
0000    ;
0000            $STATE  FILESPEC                ; GENERAL FILE SPEC CHECK
0000            $TRAN   TPA$_LAMBDA,TPA$_EXIT,BOO$FILESPEC
0000
0000    ;
0000    ; RECOGNIZE NUMBER
0000    ;
0000            $STATE NUMBER                   ;
0000            $TRAN   TPA$_DECIMAL,TPA$_EXIT  ; DECIMAL NUMBER
0000            $TRAN   '%'                     ; BASE PREFIX
0000            $STATE                          ;
0000            $TRAN   'X',HEXNUM              ; HEX BASE DESIGNATOR
0000            $TRAN   'O'                     ; OCTAL NUMBER
0000            $STATE                          ;
0000            $TRAN   TPA$_OCTAL,TPA$_EXIT    ; INTRODUCED OCTAL NUMBER
0000            $STATE  HEXNUM                  ; INTRODUCED HEX NUMBER
0000            $TRAN   TPA$_HEX,TPA$_EXIT      ; HEX NUMBER
0000
0000    ;
0000    ; RECOGNIZE SWITCH/VALUE SEPARATOR
0000    ;
0000            $STATE  SEPARATOR               ;
0000            $TRAN   '=',TPA$_EXIT           ;
0000            $TRAN   ':',TPA$_EXIT           ;
0000
0000    ;
0000    ; Get a numeric qualifier value
0000    ;
0000            $STATE  VALUE                   ; Get value for option
0000            $TRAN   !SEPARATOR              ;
0000            $STATE                          ;
0000            $TRAN   !NUMBER,TPA$_EXIT       ;
0000
0000            .IF     DF,CMDSW                ;SYSGEN specific commands
0000    ;
0000    ; Adapter command
0000    ;
0000            $STATE  ADPCMD                  ; Command to set adapter TR number
0000            $TRAN   'ADAPTER',,BOO$RESET_ADAP ; Command verb
0000            $STATE                          ;
0000            $TRAN   !NUMBER,TPA$_EXIT       ; Numeric value
0000            $TRAN   !ADAP_STR,TPA$_EXIT     ; Generic Name
0000
0000    ;
0000    ; Autoconfigure command
0000    ;
0000            $STATE  AUTOCONFIG              ; Auto configure command
0000            $TRAN   'AUTOCONFIGURE',,BOO$RESETLIST  ; Command verb
0000            $STATE                          ;
0000            $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
0000            $STATE  OUTPUT
0000            $TRAN   'OUTPUT'
0000            $STATE
0000            $TRAN   !SEPARATOR
0000            $STATE
0000            $TRAN   !FILESPEC,TPA$_EXIT,BOO$OUTPUT_FILE
0000
0000            $STATE  RESET
0000            $TRAN   'RESET',TPA$_EXIT                ; No action,this is the default
0000            $TRAN   'NORESET',TPA$_EXIT,BOO$NO_RESET ; Turn reset off this call
0000
0000
0000    ;
0000    ; Create command - Create contiguous file for paging, swapping or system dump
0000    ;
0000            $STATE  CREATECMD                   ;
0000            $TRAN   'CREATE'                    ; Command verb
0000            $STATE  CREOPT                      ; Create options
0000            $TRAN   '/',CREATE_QUAL             ;
0000            $TRAN   !FILESPEC,CREOPT,BOO$SETFILNAM; Set name of file
0000            $TRAN   TPA$_EOS,TPA$_EXIT          ;
0000
0000            $STATE  CREATE_QUAL                 ;
0000            $TRAN   'CONTIGUOUS',CREOPT,BOO$CRECONTIG    ; Contiguous
0000            $TRAN   'NOCONTIGUOUS',CREOPT,BOO$CRENCONTIG ; Set non-contiguous
0000            $TRAN   'SIZE'                      ; Get the allocation size
0000            $STATE                              ;
0000            $TRAN   !VALUE,CREOPT,BOO$FILESIZE ; Set file size
0000    ;
0000    ; Connect command - Connect specified device and load driver if required
0000    ;
0000            $STATE  CONECTCMD                   ;
0000            $TRAN   'CONNECT',,BOO$CONRESET     ; Command verb
0000            $STATE                              ;
0000            $TRAN   'CONSOLE',CONSOLCMD         ; Connect console command
0000            $TRAN   TPA$_SYMBOL,,BOO$DEVNAME;   Device name
0000            $STATE                              ;
0000            $TRAN   ':'                         ;allow an optional ":"
0000            $TRAN   TPA$_LAMBDA                 ;
0000            $STATE  CONOPT                      ;
0000            $TRAN   !CONECTOPT,CONOPT           ; Connect option
0000            $TRAN   TPA$_LAMBDA,TPA$_EXIT       ;
0000    ;
0000    ; Connect console command
0000    ;
0000            $STATE  CONSOLCMD
0000            $TRAN   !CONSOLOPT,TPA$_EXIT,BOO$CONSOLE;
0000
0000            $STATE  CONSOLOPT
0000            $TRAN   '/'
0000            $TRAN   TPA$_LAMBDA,TPA$_EXIT
0000            $STATE
0000            $TRAN   'REMOTE',TPA$_EXIT,,BOOCMD$M_REMOTE,BOO$GL_CMDOPT; Connect remote co
0000            $TRAN   'LOGICAL',TPA$_EXIT,,BOOCMD$M_LOGICAL,BOO$GL_CMDOPT; Connect logical
0000            $TRAN   TPA$_LAMBDA,TPA$_EXIT
0000
```

```
0000         ;
0000         ;       Recognize INSTALL command
0000         ;
0000                 $STATE  INSTALCMD                       ;
0000                 $TRAN   'INSTALL'                       ;
0000                 $STATE                                  ;
0000                 $TRAN   !FILESPEC,,BOO$SETFILNAM;
0000                 $STATE  INS1                            ;
0000                 $TRAN   '/'                             ; Switch introducer
0000                 $STATE
0000                 $TRAN   'PAGEFILE',,BOO$SETPGFL
0000                 $TRAN   'SWAPFILE',INS_EXIT
0000                 $TRAN   'CHECKPOINT',INS_PAGE
0000                 $TRAN   'NOCHECKPOINT',INS_PAGE,BOO$NOCHKPNT
0000                 $STATE
0000                 $TRAN   '/'                             ; look for the checkpoint switch
0000                 $TRAN   TPA$_EOS,TPA$_EXIT
0000                 $STATE
0000                 $TRAN   'CHECKPOINT',TPA$_EXIT
0000                 $TRAN   'NOCHECKPOINT',TPA$_EXIT,BOO$NOCHKPNT
0000                 $STATE  INS_PAGE
0000                 $TRAN   '/'
0000                 $STATE
0000                 $TRAN   'PAGEFILE',,BOO$SETPGFL
0000                 $STATE  INS_EXIT
0000                 $TRAN   TPA$_EOS,TPA$_EXIT
0000         ;
0000         ;       Recognize LOAD command
0000         ;
0000                 $STATE  LOADCMD                         ;
0000                 $TRAN   'LOAD',,BOO$CONRESET; Command verb
0000                 $STATE
0000                 $TRAN   !FILESPEC,TPA$_EXIT,BOO$CONDRVNAM
0000                 $STATE
0000                 $TRAN   TPA$_LAMBDA,MSCP
0000         ;
0000         ;       Recognize MSCP command
0000         ;
0000                 $STATE  MSCPCMD
0000                 $TRAN   'MSCP_LOAD',,BOO$MSCP_RESET ; Loading and starting the MSCP server
0000                 $STATE  MSCP
0000                 $TRAN   !MSCPOPT,MSCP
0000                 $TRAN   TPA$_LAMBDA,TPA$_EXIT
0000                 $STATE  MSCPOPT
0000                 $TRAN   '/'
0000                 $STATE
0000                 $TRAN   !MSCP_BUFFER,TPA$_EXIT,BOO$MSCP_ARG,,,2 ;BUFFERS IS PARAMETER 2
0000                 $TRAN   !MSCP_PACKET,TPA$_EXIT,BOO$MSCP_ARG,,,3 ;PACKET IS PARAMETER 3
0000                 $TRAN   !MSCP_HOSTS,TPA$_EXIT,BOO$MSCP_ARG,,,4  ;HOSTS IS PARAMETER 4
0000                 $TRAN   !MSCP_TIME_OUT,TPA$_EXIT,BOO$MSCP_ARG,,,5;TIME_OUT IS PARAMETER 5
0000                 $TRAN   !MSCP_PRIORITY,TPA$_EXIT,BOO$MSCP_ARG,,,6;PRIORITY IS PARAMETER 6
0000                 $TRAN   !MSCP_SMALL,TPA$_EXIT,BOO$MSCP_ARG,,,7  ;SMALL IS PARAMETER 7
0000                 $TRAN   !MSCP_FRACTION,TPA$_EXIT,BOO$MSCP_ARG,,,8;FRACTION IS PARAMETER 8
0000                 $TRAN   !LOADARGCNT,TPA$_EXIT,BOO$MSCP_ARG,,,0  ;ARGUMENT COUNT
0000                 $TRAN   !LOADP1,TPA$_EXIT,BOO$MSCP_ARG,,,1      ;LOAD PARAMETER 1
0000
0000                 $STATE  MSCP_BUFFER
```

```
0000                 $TRAN   'BUFFER',VALUE
0000                 $TRAN   'P2',VALUE
0000
0000                 $STATE  MSCP_PACKET
0000                 $TRAN   'PACKET',VALUE
0000                 $TRAN   'P3',VALUE
0000
0000                 $STATE  MSCP_HOSTS
0000                 $TRAN   'HOSTS',VALUE
0000                 $TRAN   'P4',VALUE
0000
0000                 $STATE  MSCP_TIME_OUT
0000                 $TRAN   'TIME_OUT',VALUE
0000                 $TRAN   'P5',VALUE
0000
0000                 $STATE  MSCP_PRIORITY
0000                 $TRAN   'PRIORITY',VALUE
0000                 $TRAN   'P6',VALUE
0000
0000                 $STATE  MSCP_SMALL
0000                 $TRAN   'MINIMUM',VALUE
0000                 $TRAN   'P7',VALUE
0000
0000                 $STATE  MSCP_FRACTION
0000                 $TRAN   'MAXIMUM',VALUE
0000                 $TRAN   'P8',VALUE
0000
0000                 $STATE  LOADARGCNT
0000                 $TRAN   'ARGCOUNT',VALUE
0000
0000                 $STATE  LOADP1
0000                 $TRAN   'P1',VALUE
0000
0000         ;
0000         ;
0000         ;       Recognize RELOAD command
0000         ;
0000                 $STATE  RELOADCMD                  ;
0000                 $TRAN   'RELOAD',LOAD1,BOO$CONRESET; Command verb
0000
0000         ;
0000         ; Share command - Initialize and/or connect to a shared memory
0000         ;
0000                 $STATE  SHARECMD
0000                 $TRAN   'SHARE',,GEN$SHR_RESET  ; Command verb
0000                 $STATE  SHARECMDOPT
0000                 $TRAN   !SHAREOPT,SHARECMDOPT   ; Command options
0000                 $TRAN   TPA$_LAMBDA
0000                 $STATE
0000                 $TRAN   'M'                     ; Multiport memory "MPMx"
0000                 $STATE                          ;
0000                 $TRAN   'P'                     ;
0000                 $STATE                          ;
0000                 $TRAN   'M'                     ;
0000                 $STATE                          ;
0000                 $TRAN   TPA$_DECIMAL,,GEN$SHR_UNIT ; Memory unit #
0000                 $STATE                          ;
```

```
0000          $TRAN   TPA$_SYMBOL,,GEN$SHR_MEMNAME ; Memory name
0000          $STATE  SHROPT                 ;
0000          $TRAN   !SHAREOPT,SHROPT       ; Share options
0000          $TRAN   TPA$_EOS,TPA$_EXIT     ;
0000
0000  ;
0000  ; SYSGEN specific show qualifiers
0000  ;
0000          $STATE  DEV_OR_DRIV
0000          $TRAN   'DEVICES',,,,,0        ; SHO/DEVICES[=devname]
0000          $TRAN   'DRIVER',,,,,1         ; SHO/DRIVER [=devname]
0000          $STATE
0000          $TRAN   TPA$_EOS,TPA$_EXIT,BOO$SHODEV_ALL       ; SHOW ALL
0000          $TRAN   !SEPARATOR             ;
0000          $TRAN   TPA$_LAMBDA
0000          $STATE                         ;
0000          $TRAN   TPA$_STRING,TPA$_EXIT,BOO$SHODEV        ; SHOW SPECIFIC DEVICE
0000
0000          $STATE  SHOWCON
0000          $TRAN   !SHOWCON_LOOP,TPA$_EXIT,BOO$SHOCONFIG
0000
0000          $STATE  SHOWCON_LOOP
0000          $TRAN   !SHOWCONOPT,SHOWCON_LOOP
0000          $TRAN   TPA$_EOS,TPA$_EXIT
0000          $TRAN   TPA$_LAMBDA,TPA$_FAIL
0000
0000          $STATE  SHOWCONOPT
0000          $TRAN   '/'
0000          $STATE
0000          $TRAN   'COMMAND_FILE',TPA$_EXIT,BOO$RESET_COMMAND ; Set command file spec
0000          $TRAN   !OUTPUT,TPA$_EXIT
0000          $TRAN   !ADAPTER,TPA$_EXIT,BOO$SET_TR
0000  ;
0000  ; SHOW /UNIBUS [/ADAPTER=n]
0000  ;
0000          $STATE  SHOW_UNIBUS
0000          $TRAN   'UNIBUS',,,,,0
0000          $STATE
0000          $TRAN   '/'
0000          $TRAN   TPA$_LAMBDA
0000          $STATE
0000          $TRAN   !ADAPTER,,BOO$SET_TR,,,1
0000          $TRAN   TPA$_LAMBDA
0000          $STATE
0000          $TRAN   TPA$_EOS,TPA$_EXIT,BOO$SHOW_UNIBUS
0000
0000          $STATE  ADAPTER                ; Set adapter number
0000          $TRAN   'ADAPTER',,BOO$RESET_ADAP
0000          $STATE
0000          $TRAN   !SEPARATOR
0000          $STATE
0000          $TRAN   !NUMBER,TPA$_EXIT
0000          $TRAN   TPA$_LAMBDA,ADAP_STR
0000
0000          $STATE  ADAP_STR2
0000          $TRAN   TPA$_LAMBDA,ADAP_STR,BOO$RESET_ADAP
0000
```

```
0000          $STATE  ADAP_STR
0000          $TRAN   TPA$_ALPHA,ADAP_STR,BOO$ADAP_LETTER     ; One letter at a time
0000          $TRAN   TPA$_DECIMAL,TPA$_EXIT,BOO$ADAPTER_NAME ; Take number as end
0000
0000   ;
0000   ;      Recognize the TERMINAL command
0000   ;
0000          $STATE  TERMINALCMD
0000          $TRAN   'TERMINAL'
0000          $STATE
0000          $TRAN   '/'
0000          $STATE
0000          $TRAN   'ECHO',TPA$_EXIT,SYSG$LOAD_TT_STR ; /ECHO only qualifier
0000   ;
0000   ;      Recognize WRITE Command
0000   ;
0000          $STATE  WRTCMD                  ;
0000          $TRAN   'WRITE'                 ; Command verb
0000          $STATE                          ;
0000          $TRAN   !WRTCUR,TPA$_EXIT       ;
0000          $TRAN   !WRTACT,TPA$_EXIT       ;
0000          $TRAN   !FILESPEC,TPA$_EXIT,BOO$WRTFILE ;
0000
0000          $STATE  WRTCUR                  ; WRITE CURRENT
0000          $TRAN   'CURRENT'
0000          $STATE
0000          $TRAN   TPA$_EOS,TPA$_EXIT,BOO$WRTCUR
0000
0000          $STATE  WRTACT                  ; WRITE ACTIVE
0000          $TRAN   'ACTIVE'
0000          $STATE
0000          $TRAN   TPA$_EOS,TPA$_EXIT,BOO$WRTACT
0000   ;
0000   ; RECOGNIZE CONNECT OPTIONS
0000   ;
0000          $STATE  CONECTOPT               ;
0000          $TRAN   '/'                     ; Switch introducer
0000          $STATE                          ;
0000          $TRAN   !ADAPTER,TPA$_EXIT,BOO$CONADP           ; Adapter number
0000          $TRAN   'NOADAPTER',TPA$_EXIT,BOO$CONNLADP      ; Use null adapter
0000          $TRAN   !CONCREG,TPA$_EXIT,BOO$CONCREG          ; Control register (UBA)
0000          $TRAN   !CONCVECTOR,TPA$_EXIT,BOO$CONCVEC       ; Vector (UBA)
0000          $TRAN   !CONCNUMVEC,TPA$_EXIT,BOO$CONCNUM       ; Number of vectors
0000          $TRAN   !CONAUNIT,TPA$_EXIT,BOO$CONAUNIT        ; Adapter unit
0000          $TRAN   !CONUNITS,TPA$_EXIT,BOO$CONUNITS        ; Maximum units
0000          $TRAN   !CONSYSID_LO,TPA$_EXIT,BOO$CONSYSID_LOW  ; System ID (low)
0000          $TRAN   !CONSYSID_HI,TPA$_EXIT,BOO$CONSYSID_HIGH ; System ID (high)
0000          $TRAN   !CONVECOFF,TPA$_EXIT,BOO$CONVECOFFSET   ; Offset to vector(combo dev
0000          $TRAN   !CONCSROFF,TPA$_EXIT,BOO$CONCSROFFSET   ; Offset to CSR(combo device
0000          $TRAN   'DRIVERNAME'            ;
0000          $STATE                          ;
0000          $TRAN   !SEPARATOR              ;
0000          $STATE  LOAD1                   ;
0000          $TRAN   !FILESPEC,TPA$_EXIT,BOO$CONDRVNAM       ; Driver name
0000
0000          $STATE  CONCREG                 ; Control register address
0000          $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        $TRAN   !SHRCEFCNT,TPA$_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          $STATE  SHRMBXMAX                 ; Port maximum Mailboxes
0000          $TRAN   'MAXMAILBOXES',VALUE      ;
0000
0000          $STATE  SHRCEFMAX                 ; Port maximum Common Ev Flag Clusters
0000          $TRAN   'MAXCEFCLUSTERS',VALUE    ;
0000
0000          $STATE  POOLCNT                   ; Total pool blocks count
0000          $TRAN   'POOLBCOUNT',VALUE        ;
0000          $TRAN   'POOLBCNT',VALUE          ;
0000
0000          $STATE  POOLSIZE                  ; Pool block size
0000          $TRAN   'POOLBSIZE',VALUE         ;
0000
0000          $STATE  PRQCNT                    ; Total PRQ blocks count
0000          $TRAN   'PRQCOUNT',VALUE          ;
0000          $TRAN   'PRQCNT',VALUE   ;
0000
0000          $STATE  SHRSTART                  ; Starting relative PFN
0000          $TRAN   'START',VALUE             ;
0000
0000          .ENDC                             ; End SYSGEN specific command
0000          $END_STATE
0000      ;
```

```
                                  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$GT_SYSNAME::                           ; System name string
3A 4D 45 54 53 59 53 24 53 59 53 00'  0538   869         .ASCIC  \SYS$SYSTEM:SYS.EXE\   ; Name of sytem image
               45 58 45 2E 53 59 53  0544
                                  12  0538
                                  054B   870 BOO$GT_SYSPARNAME::
3A 4D 45 54 53 59 53 24 53 59 53 00'  054B   871         .ASCIC  \SYS$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  @!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  @!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     882
21 43 41 35 31 21 000005D6'010E0000'  05CE     883 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     884
31 28 23 21 20 20 00000616'010E0000'  060E     885 NCTRLSTR:        .ASCID  @  !#(17AC) @
                     20 29 43 41 37  061C
                                      0621     886
72 61 74 53 20 20 00000629'010E0000'  0621     887 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     888
61 72 61 50 2F 21 0000064D'010E0000'  0645     889 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     890
                                      0665     891 SDVHDR:
61 4E 20 72 65 74 65 6D 61 72 61 50  0665     892         .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     893         .ASCII  \ Unit  Dynamic\
                               63 69  06B1
                               0A 0D  06B3     894         .ASCII  <CR><LF>                  ;
2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D  06B5     895         .ASCII  \--------------              -------   -------   -------   -------\
20 20 20 20 20 20 20 20 20 20 2D 2D  06C1
20 20 2D 2D 2D 2D 2D 2D 2D 20 20 20  06CD
2D 20 20 20 2D 2D 2D 2D 2D 2D 2D 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 2D 2D 2D 2D 20  06F5     896         .ASCII  \ ----  -------\
                               2D 2D  0701
                          0000009E  0703     897 SDVHDRLEN=.-SDVHDR
                                      0703     898
                                      0703     899 .IF     NDF,CMDSW                         ; SYSBOOCMD definitions for RIO$OUTPUT_LINE
                                      0703     900
                          00000100  0703     901 RIO$AB_OUTBUF::          .long   BUFFER_SIZE
                          0000070B' 0707     902                          .long   RIO$AB_BUFFER
                          0000080B  070B     903 RIO$AB_BUFFER::          .blkb   BUFFER_SIZE
                              0000  080B     904 RIO$GW_OUTLEN::          .word   0
                                      080D     905
                                00  080D     906 BOO$GB_FILELEN:          .byte   0
                          00000000  080E     907 BOO$GL_FILEADDR:         .long   0
                                      0812     908 BOO$GT_CURRENT:
                                      0812     909 BOO$GT_DEFAULT:
                          00000000  0812     910 BOO$GL_PARINUSE:         .long   0
                                      0816     911
                                      0816     912 .ENDC
```

```
                                  0816          .SBTTL
                                  0816  ;++
                                  0816  ;
                                  0816  ; Functional Description:
                                  0816  ;
                                  0816  ;
                                  0816  ; Calling Sequence:
                                  0816  ;       NONE
                                  0816  ;
                                  0816  ; Input Parameters:
                                  0816  ;       NONE
                                  0816  ;
                                  0816  ; Implicit Inputs:
                                  0816  ;       NONE
                                  0816  ;
                                  0816  ; Output Parameters:
                                  0816  ;       NONE
                                  0816  ;
                                  0816  ; Implicit Outputs:
                                  0816  ;       NONE
                                  0816  ;
                                  0816  ; Side Effects:
                                  0816  ;       NONE
                                  0816  ;
                                  0816  ;--
                                  0816          .LIST   MEB                     ; Show macro expansions
                                  0816
                                  0816
                            0FFC  0816  BOO$GETPARAM::  .WORD   ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>     ; Save all registers
                                  0818
                                  0818          .IF     NDF,CMDSW       ; SYSBOOCMD only
                                  0818  ;
                                  0818  ; Make descriptors PIC (only needed in SYSBOOCMD)
                                  0818  ;
                                  0818
  FD63 CF    FD6A CF       9E     0818          MOVAB   CTRLSTR+8,CTRLSTR+4     ; Set address in descriptor
  FD84 CF    FD8B CF       9E     081F          MOVAB   HEXSTR+8,HEXSTR+4       ; Set address in descriptor
  FDA5 CF    FDAC CF       9E     0826          MOVAB   ASCSTR+8,ASCSTR+4       ; Set address in descriptor
  FDDE CF    FDE5 CF       9E     082D          MOVAB   NCTRLSTR+8,NCTRLSTR+4   ; Set address in descriptor
  FDEA CF    FDF1 CF       9E     0834          MOVAB   SCTRLSTR+8,SCTRLSTR+4   ; Set address in descriptor
             FECC CF       9E     083B          MOVAB   RIO$AB_BUFFER,-
             FEC5 CF              083F                  RIO$AB_OUTBUF+4         ; Set address in descriptor
                                  0842
                                  0842          .ENDC
                                  0842
                                  0842  READCMD:                                ;
        57   F7BA CF       DE     0842          MOVAL   PARMBLK,R7              ; Get address of parameter block
                08 A7      D4     0847          CLRL    TPA$L_STRINGCNT(R7)     ; Initialize string length
     0C A7   F8EA CF       9E     084A          MOVAB   BOO$GT_COMSTR,TPA$L_STRINGPTR(R7)       ; And address
                                  0850  READLINE:                               ;
        52   F81C CF       9E     0850          MOVAB   BOO$GT_COMBUF,R2        ; Set address of buffer
                      52   DD     0855          PUSHL   R2                      ; Set buffer address into argument list
         7E    C8 8F       9A     0857          MOVZBL  #BOO$C_COMBUFSZ,-(SP)   ;  and maximum size for read
       00000000'EF         9F     085B          PUSHAB  BOO$GT_PROMPT           ; Address of prompt string
    00000000'EF    03      FB     0861          CALLS   #3,L^BOO$READPROMPT     ; Prompt for and accept command
                01 50      E8     0868          BLBS    R0,5$                   ; Exit if end of file.
                           04     086B          RET                             ;
```

```
                             086C
                             086C      ; Upcase input
                             086C
           50   52   D0      086C      5$:      MOVL     R2,R0                          ; Set address of string
           51   80   9A      086F               MOVZBL   (R0)+,R1                       ; Get address and count
        61 8F   60   91      0872      7$:      CMPB     (R0),#^A/a/                    ; Lower case possible ?
                09   1F      0876               BLSSU    8$                             ; No, Branch
        7A 8F   60   91      0878               CMPB     (R0),#^A/z/                    ; Lower case possible ?
                03   1A      087C               BGTRU    8$                             ; No, Branch
           60   20   8A      087E               BICB2    #^X20,(R0)                     ; Clear bit, make character upper case
                50   D6      0881      8$:      INCL     R0                             ; Increment pointer
             EC 51   F5      0883               SOBGTR   R1,7$                          ; Loop
                             0886
 53    08 A7    0C A7   C1   0886               ADDL3    TPA$L_STRINGPTR(R7),TPA$L_STRINGCNT(R7),R3 ; Get current pointer
           50   82   9A      088C               MOVZBL   (R2)+,R0                       ; Get length of input line
                BF   13      088F               BEQL     READLINE                       ; Ignore null input
           51   52   D0      0891               MOVL     R2,R1                          ; Move to LOCC address register
           52   81   9A      0894      30$:     MOVZBL   (R1)+,R2                       ; Get a character
           52   2D   91      0897               CMPB     #^A/-/,R2                      ; Is this a possible continuation?
                1D   13      089A               BEQL     50$                            ; Branch if yes
           52   21   91      089C               CMPB     #^A/!/,R2                      ; Is this the start of a comment?
                0B   13      089F               BEQL     40$                            ; Branch if yes
             08 A7   D6      08A1               INCL     TPA$L_STRINGCNT(R7)            ; Bump characters in command string
           83   52   90      08A4               MOVB     R2,(R3)+                       ; Copy character to command string
             EA 50   F5      08A7      35$:     SOBGTR   R0,30$                         ; Continue for all characters in put
                4B   11      08AA               BRB      PARSE                          ; Done, parse command
                50   D7      08AC      40$:     DECL     R0                             ; One less character
     01 A1     50   21   3A  08AE               LOCC     #^A/!/,R0,1(R1)                ; Scan remaining string for !
                42   13      08B3               BEQL     PARSE                          ; None end of line first
                51   D6      08B5               INCL     R1                             ; Advance to next character
                EE   11      08B7               BRB      35$                            ; Continue with line scan
           54   53   D0      08B9      50$:     MOVL     R3,R4                          ; Save string insertion pointer
        55   08 A7   D0      08BC               MOVL     TPA$L_STRINGCNT(R7),R5         ; and current length
           83   52   90      08C0               MOVB     R2,(R3)+                       ; Copy to buffer anyway
             08 A7   D6      08C3               INCL     TPA$L_STRINGCNT(R7)            ; Advance counter
                15   11      08C6               BRB      65$                            ; And check for end of string
           52   81   9A      08C8      60$:     MOVZBL   (R1)+,R2                       ; Get another character
           83   52   90      08CB               MOVB     R2,(R3)+                       ; Copy to buffer
             08 A7   D6      08CE               INCL     TPA$L_STRINGCNT(R7)            ; Bump string count
           52   20   91      08D1               CMPB     #^A/ /,R2                      ; Blank?
                07   13      08D4               BEQL     65$                            ; Yes, still might be a continuation
           52   21   91      08D6               CMPB     #^A/!/,R2                      ; Is this a comment?
                0F   13      08D9               BEQL     80$                            ; Branch if yes
                CA   11      08DB               BRB      35$                            ; Not a continuation
             E8 50   F5      08DD      65$:     SOBGTR   R0,60$                         ; Continue to end of line
           53   54   D0      08E0      70$:     MOVL     R4,R3                          ; Drop everything after continuation
        08 A7   55   D0      08E3               MOVL     R5,TPA$L_STRINGCNT(R7)         ; By restoring count
              FF66   31      08E7               BRW      READLINE                       ; Read another line
                50   D7      08EA      80$:     DECL     R0                             ; One less character
     01 A1     50   21   3A  08EC               LOCC     #^A/!/,R0,1(R1)                ; Scan for end of comment
                ED   13      08F1               BEQL     70$                            ; None
                51   D6      08F3               INCL     R1                             ; Skip trailing !
                E6   11      08F5               BRB      65$                            ; and continue scan for end of line
           67   08   D0      08F7      PARSE:   MOVL     #TPA$K_COUNT0,TPA$L_COUNT(R7)   ; Init count field
        04 A7   02   C8      08FA               BISL     #TPA$M_ABBREV,TPA$L_OPTIONS(R7) ; Permit abbreviations
                     CA      08FE               BICL2    #^C<BOOCMD$M_NOCHECK!-
                             08FF                        BOOCMD$M_SETOUTPUT!-
```

```
                                  08FF                  BOOCMD$M_TERMINAL>,-
      00000000'EF   FFFE7FFE 8F   08FF                  BOO$GL_CMDOPT            ; Clear all options but specified
                        18 A7  94 0909          CLRB    TPA$B_CHAR(R7)           ; Last character parsed
                  00000000'EF  9F 090C          PUSHAB  KEYTBL                   ; Pass address of key table
                  00000000'EF  9F 0912          PUSHAB  STATE1                   ;  and state table
                           57  DD 0918          PUSHL   R7                       ; Set address of parameter block
          00000000'GF   03     FB 091A          CALLS   #3,G^LIB$TPARSE          ; Parse input
                        19 50  E8 0921          BLBS    R0,20$                   ; Branch if no syntax error
                  13 50   1F   E0 0924          BBS     #31,R0,15$               ; Branch if error already given
                                  0928
                                  0928  .IF     NDF,CMDSW               ; SYSBOOCMD
                                  0928
                                  0928          MSG     <-E-Syntax error>        ; SYSBOOT error message
                        F6D5'  30 0928          BSBW    BOO$FACMSG               ;
72 65 20 78 61 74 6E 79 53 2D 45 2D 0928          .ASCIZ  \-E-Syntax error\                       ;
                  00 72 6F 72     0937
                                  093B
                                  093B  .IFF                            ; SYSGENCMD
                                  093B
                                  093B          CMPL    #LIB$_SYNTAXERR,R0       ; Tparse Syntax error ?
                                  093B          BEQLU   10$                      ; Branch if yes
                                  093B          TSTL    R0                       ; Zero ?
                                  093B          BEQL    10$                      ; Branch if yes
                                  093B          PUSHL   R0                       ; Push REAL error code
                                  093B          CALLS   #1,G^LIB$SIGNAL          ; Signal Error
                                  093B          BRW     30$                      ; Continue
                                  093B
                                  093B  ; Heuristically determine where syntax error occured
                                  093B
                                  093B  10$:    MOVZBL  TPA$B_CHAR(R7),R4        ; Was there a character parsed ?
                                  093B          BNEQ    12$                      ; Branch if yes
                                  093B          MOVQ    TPA$L_STRINGCNT(R7),-(SP) ; Push entire read-in string
                                  093B          BRB     14$                      ; Branch
                                  093B
                                  093B  12$:    SUBL3   TPA$L_TOKENCNT(R7),TPA$L_STRINGCNT(R7),R2 ; Length
                                  093B          ADDL3   TPA$L_TOKENCNT(R7),TPA$L_STRINGPTR(R7),R3 ; Address
                                  093B          CMPB    #^A'/',R4                ; Was it a qualifier error ?
                                  093B          BEQL    13$                      ; No
                                  093B          MOVQ    R2,-(SP)
                                  093B          BRB     14$
                                  093B
                                  093B  13$:    LOCC    TPA$B_CHAR(R7),R2,(R3)   ; Find it then
                                  093B          MOVQ    R0,-(SP)                 ; Push length and address
                                  093B  14$:    PUSHL   #2                       ; Number of FAO params
                                  093B          PUSHL   #SYSG$_SYNTAX            ; Error message
                                  093B          CALLS   #4,G^LIB$SIGNAL          ; Signal the error
                                  093B
                                  093B  .ENDC
                                  093B
                           5C  11 093B  15$:    BRB     30$                      ; and get another command
          56      00000000'EF  D0 093D  20$:    MOVL    BOO$GL_CMDOPT,R6         ; Get command option flags
                  54 56   08   E0 0944          BBS     #BOOCMD$V_CONT,R6,EXIT   ; Exit if continue flag
                  4D 56   09   E1 0948          BBC     #BOOCMD$V_DEFAULT,R6,30$ ; Read another command if Help
                                  094C  ;
                                  094C  ; The Default values for system parameters are selected and must be copied to
                                  094C  ; the current system parameter area.
                                  094C  ;
```

```
 00000004'EF  00000000'EF  7D  094C             MOVQ    EXE$GQ_TODCBASE,SAVE_TODCBASE ; Save time base register
 0000000C'EF  00000000'EF  D0  0957             MOVL    EXE$GL_TODR,SAVE_TODR         ; Save time register
                               0962
                  0000'8F  28  0962             MOVC3   #EXE$C_SYSPARSZ,-
              00000000'EF      0966                     BOO$A_SYSPARAM,-
              00000000'EF      096B                     EXE$A_SYSPARAM          ; Copy defaults
                               0970
 00000000'EF  00000004'EF  7D  0970             MOVQ    SAVE_TODCBASE,EXE$GQ_TODCBASE ; Restore
 00000000'EF  0000000C'EF  D0  097B             MOVL    SAVE_TODR,EXE$GL_TODR
                               0986
              00000000'8F  E2  0986             BBSS    #EXE$V_WRITESYSPARAMS,- ; Use default => write current needed
           00 00000000'GF      098C                     G^EXE$GL_DYNAMIC_FLAGS,1$;
                               0992        1$:
                  FE7C CF  DE  0992             MOVAL   BOO$GT_DEFAULT,-
                  FE79 CF      0996                     BOO$GL_PARINUSE         ; Set default in use
                     FEA6  31  0999        30$: BRW     READCMD                 ; Read more commands
                               099C
                  50   01  D0  099C        EXIT: MOVL   #1,R0                   ; Return success
                           04  099F             RET                             ;
                               09A0
```

```
                              09A0  1103         .SBTTL  BOO$FILESPEC - Parse file spec
                              09A0  1104 ;+
                              09A0  1105 ;
                              09A0  1106 ; CALLING SEQUENCE:
                              09A0  1107 ;
                              09A0  1108 ;       called as a TPARSE action routine
                              09A0  1109 ;
                              09A0  1110 ; INPUT:
                              09A0  1111 ;
                              09A0  1112 ;       The tparse parameter block (AP)
                              09A0  1113 ;
                              09A0  1114 ; OUTPUT:
                              09A0  1115 ;
                              09A0  1116 ;       A possible file spec is found.
                              09A0  1117 ;
                              09A0  1118 ; SIDE EFFECTS:
                              09A0  1119 ;
                              09A0  1120 ;       The tparse parameter block is updated.
                              09A0  1121 ;
                              09A0  1122 ;-
                              09A0  1123
                        00FC  09A0  1124 .Entry  BOO$FILESPEC, ^M<R2,R3,R4,R5,R6,R7>
                              09A2  1125
       52   0C AC   D0        09A2  1126         MOVL    TPA$L_STRINGPTR(AP),R2   ; Get address of current parse
       14 AC   52   D0        09A6  1127         MOVL    R2,TPA$L_TOKENPTR(AP)    ; Set token pointer
     FE5F CF   52   D0        09AA  1128         MOVL    R2,BOO$GL_FILEADDR       ; Set file spec pointer
       53   08 AC   D0        09AF  1129         MOVL    TPA$L_STRINGCNT(AP),R3   ; Remainder of parse string length
            23      13        09B3  1130         BEQL    100$                     ; Error if zero
                              09B5  1131
     62   53   20   3A        09B5  1132         LOCC    #^A/ /,R3,(R2)           ; is there a blank?
                 04   12      09B9  1133         BNEQ    50$                      ; Branch if yes
     62   53   2F   3A        09BB  1134         LOCC    #^Aa/a,R3,(R2)           ; is there a slash?
                              09BF  1135
       08 AC   50   D0        09BF  1136 50$:    MOVL    R0,TPA$L_STRINGCNT(AP)   ; Remaining length
       0C AC   51   D0        09C3  1137         MOVL    R1,TPA$L_STRINGPTR(AP)   ; Address of blank or slash
          51   52   C2        09C7  1138         SUBL2   R2,R1                    ; Calculate length
       10 AC   51   D0        09CA  1139         MOVL    R1,TPA$L_TOKENCNT(AP)    ; Set length of file spec
     FE3A CF   51   90        09CE  1140         MOVB    R1,BOO$GB_FILELEN        ; Set length of file spec
          50   01   D0        09D3  1141 60$:    MOVL    #SS$_NORMAL,R0           ; Set success
               07   11        09D6  1142         BRB     110$                     ; Exit
                              09D8  1143
  50    00000000'8F   D0      09D8  1144 100$:   MOVL    #LIB$_SYNTAXERR,R0
                      04      09DF  1145 110$:   RET
                              09E0  1146
```

```
                                 09E0  1148          .SBTTL  BOO$USECUR - Use parameters from current image
                                 09E0  1149 BOO$USECUR::                                  ; Set to current system values
                           03FC  09E0  1150          .WORD   ^M<R2,R3,R4,R5,R6,R7,R8,R9>;
      50   0000054B'GF      9E   09E2  1151          MOVAB   G^BOO$GT_SYSPARNAME,R0   ; Get address of system .PAR file name
         5C    F613 CF      DE   09E9  1152          MOVAL   PARMBLK,AP               ; Get address of the TPARSE parameter block
            10 AC   80      9A   09EE  1153          MOVZBL  (R0)+,TPA$L_TOKENCNT(AP); Set up for call to BOO$USEFILE
            14 AC   50      D0   09F2  1154          MOVL    R0,TPA$L_TOKENPTR(AP)    ;
      00000000'GF   6C      FA   09F6  1155          CALLG   (AP),G^BOO$USEFILE       ; Call routine to process the .PAR file
                 13 50      E9   09FD  1156          BLBC    R0,10$                   ; Branch to failure code
           00000000'8F      E5   0A00  1157          BBCC    #EXE$V_WRITESYSPARAMS,-  ; Use current => no write current needed
        00 00000000'GF           0A06  1158                  G^EXE$GL_DYNAMIC_FLAGS,5$;
               FE02 CF      DE   0A0C  1159 5$:      MOVAL   BOO$GT_CURRENT,-
               FDFF CF           0A10  1160                  BOO$GL_PARINUSE          ; Set parameters in use
               50   01      D0   0A13  1161 10$:     MOVL    #1,R0                    ; Return success
                            04   0A16  1162          RET                              ;
                                 0A17  1163
```

```
                                        0A17             .SBTTL  BOO$SHOWV - Routine to show one parameter value
                                        0A17       ;
                                        0A17       ; Input Parameters:
                                        0A17       ;        R4 - Pointer to PRM block to be displayed.
                                        0A17       ;
                                        0A17       ; Output Parameters:
                                        0A17       ;        Content of parameter block is displayed by calling RIO$OUTPUT_LINE
                                        0A17       ;
                                        0A17       BOO$SHOWV:                                              ;
                      50    64     D0   0A17             MOVL    (R4),R0                          ; Get address of value
          50   00000000'EF40       9E   0A1A             MOVAB   BOO$A_SYSPARAM[R0],R0            ; Add current base address
            50   00000000'8F       C2   0A22             SUBL    #BOO$A_SYSPARAM,R0               ;  and subtract link-time value
                 53   FB79 CF      7E   0A29             MOVAQ   HEXSTR,R3                        ; Assume hex display
                            0B     E0   0A2E             BBS     #BOOCMD$V_DISHEX,-
             0A 00000000'EF             0A30                     BOO$GL_CMDOPT,1$                 ; If set, then display hex
                 53   FB44 CF      7E   0A36             MOVAQ   CTRLSTR,R3                       ; Assume not ascii data
               03 10 A4    10      E0   0A3B             BBS     #PRM$V_ASCII,PRM$L_FLAGS(R4),2$ ; Branch if ascii
                         0085      31   0A40       1$:   BRW     15$                              ;
                                        0A43
                                        0A43       ;
                                        0A43       ; ASCII data
                                        0A43       ;
                            55     DD   0A43       2$:   PUSHL   R5                               ; Save a register
                       5E   10     C2   0A45             SUBL2   #16,SP                           ; Allocate a buffer on the stack
                  52    14 A4      9A   0A48             MOVZBL  PRM$B_SIZE(R4),R2                ; Get size (in bits)
             52   52    FD 8F      78   0A4C             ASHL    #-3,R2,R2                        ; Convert size from bit to byte count
                       51   52     D0   0A51             MOVL    R2,R1                            ; Make a copy of the size
                       04   52     D1   0A54             CMPL    R2,#4                            ; Size > 4?
                            03     15   0A57             BLEQ    3$                               ; If geq yes
                       51   04     D0   0A59             MOVL    #4,R1                            ; Max of 4 for default, max and min
          FB17 CF      05   51     83   0A5C       3$:   SUBB3   R1,#5,BLANKS                     ; Calculate number of blank spaces needed
                            16     BB   0A62             PUSHR   #^M<R1,R2,R4>                    ; Save some registers
   0C AE    10    20   60   52     2C   0A64             MOVC5   R2,(R0),#^A/ /,#16,<3*4>(SP) ; Move the parameter value into the buf
                            16     BA   0A6B             POPR    #^M<R1,R2,R4>                    ; Restore the registers
                       53   5E     D0   0A6D             MOVL    SP,R3                            ; Save a pointer to the buffer
          FAF1 CF      11   52     83   0A70             SUBB3   R2,#17,CUR_BLANKS                ; Calculate number of pad blanks
                      FAEC CF      DF   0A76             PUSHAL  BOO$T_NODYNAMIC                  ; Assume not dynamic
               05 10 A4    00      E1   0A7A             BBC     #PRM$V_DYNAMIC,PRM$L_FLAGS(R4),10$ ; Branch if not
                 6E   FAE1 CF      DE   0A7F             MOVAL   BOO$T_DYNAMIC,(SP)               ; Change to dynamic string
                        26 A4      9F   0A84       10$:  PUSHAB  PRM$T_UNIT(R4)                   ; Stack address of unit name string
                        0C A4      9F   0A87             PUSHAB  PRM$L_MAX(R4)                    ; Stack maximum value
                            51     DD   0A8A             PUSHL   R1                               ;  and size
                      FAE9 CF      9F   0A8C             PUSHAB  BLANKS                           ; Blanks for padding
                        08 A4      9F   0A90             PUSHAB  PRM$L_MIN(R4)                    ; Minimum value
                            51     DD   0A93             PUSHL   R1                               ;  and size
                      FAE0 CF      9F   0A95             PUSHAB  BLANKS                           ; Blanks for padding
                        04 A4      9F   0A99             PUSHAB  PRM$L_DEFAULT(R4)                ; Default value
                            51     DD   0A9C             PUSHL   R1                               ;  and size
                      FAD7 CF      9F   0A9E             PUSHAB  BLANKS                           ; Blanks for padding
                            53     DD   0AA2             PUSHL   R3                               ; Address of current value
                            52     DD   0AA4             PUSHL   R2                               ;  and size
                      FABD CF      9F   0AA6             PUSHAB  CUR_BLANKS                       ; Blanks for padding
                        16 A4      9F   0AAA             PUSHAB  PRM$T_NAME(R4)                   ; Stack address of parameter name
                      FC52 CF      7F   0AAD             PUSHAQ  RIO$AB_OUTBUF                    ; Stack address of buffer descriptor
                      FD56 CF      DF   0AB1             PUSHAL  RIO$GW_OUTLEN                    ; Set address of loc to receive size
                      FB15 CF      7F   0AB5             PUSHAQ  ASCSTR                           ; Control string for ascii
                                        0AB9
```

```
     00000000'EF    12    FB  0AB9  1222          CALLS    #18,SYS$FAO                ; Format value for output
              5E    10    C0  0AC0  1223          ADDL2    #16,SP                     ; Remove the buffer from the stack
                    55  8ED0  0AC3  1224          POPL     R5                         ; Restore a register
                    3D    11  0AC6  1225          BRB      55$                        ; and join common code
                              0AC8  1226
                              0AC8  1227 ;
                              0AC8  1228 ; Decimal or hex display - R3 contains address of control string
                              0AC8  1229 ;
                    42    10  0AC8  1230 15$:    BSBB     GETDATA                    ; Get data item according to size
              FA98 CF     DF  0ACA  1231          PUSHAL   BOO$T_NODYNAMIC            ; Assume not dynamic
        05 10 A4    00    E1  0ACE  1232          BBC      #PRM$V_DYNAMIC,PRM$L_FLAGS(R4),20$ ; Branch if not
         6E   FA8D CF     DE  0AD3  1233          MOVAL    BOO$T_DYNAMIC,(SP)         ; Change to dynamic string
                              0AD8  1234
                 26 A4    9F  0AD8  1235 20$:     PUSHAB   PRM$T_UNIT(R4)             ; Stack address of unit name string
                 0C A4    DD  0ADB  1236          PUSHL    PRM$L_MAX(R4)              ; Stack maximum value
                 08 A4    DD  0ADE  1237          PUSHL    PRM$L_MIN(R4)              ;  and minimum value
                 04 A4    DD  0AE1  1238          PUSHL    PRM$L_DEFAULT(R4)          ; Default value
       10 A4    1000 8F   B3  0AE4  1239          BITW     #PRM$M_NEG,PRM$L_FLAGS(R4)        ; check for negated value
                    03    13  0AEA  1240          BEQL     30$                        ; Branch if not
              6E    6E    CE  0AEC  1241          MNEGL    (SP),(SP)                  ; Make absolute value
                    51    DD  0AEF  1242 30$.     PUSHL    R1                         ; Current value
                 16 A4    9F  0AF1  1243          PUSHAB   PRM$T_NAME(R4)             ; Stack address of parameter name
              FC0B CF     7F  0AF4  1244          PUSHAQ   RIO$AB_OUTBUF              ; Stack address of buffer descriptor
              FD0F CF     DF  0AF8  1245          PUSHAL   RIO$GW_OUTLEN              ; Set address of loc to receive size
                    53    DD  0AFC  1246          PUSHL    R3                         ; Push address of control string
                              0AFE  1247
     00000000'EF    0A    FB  0AFE  1248          CALLS    #10,SYS$FAO                ; Format value for output
                 03 50    E9  0B05  1249 55$:     BLBC     R0,60$
                              0B08  1250
                  0405    30  0B08  1251          BSBW     RIO$OUTPUT_LINE            ; Output the line
                    05        0B0B  1252 60$:     RSB                                 ; and return
                              0B0C  1253
                              0B0C  1254 GETDATA:                                     ;
           51    15 A4    9A  0B0C  1255          MOVZBL   PRM$B_POS(R4),R1           ; GET SIZE OF DATUM
   51    60    14 A4    51    EF  0B10  1256          EXTZV    R1,PRM$B_SIZE(R4),(R0),R1        ; GET DATUM
       10 A4    1000 8F   B3  0B16  1257          BITW     #PRM$M_NEG,PRM$L_FLAGS(R4)       ; CHECK FOR NEGATED VALUE
                    0D    13  0B1C  1258          BEQL     10$                        ; NO
           51    15 A4    9A  0B1E  1259          MOVZBL   PRM$B_POS(R4),R1           ; GET POSITION AGAIN
   51    60    14 A4    51    EE  0B22  1260          EXTV     R1,PRM$B_SIZE(R4),(R0),R1        ; CONVERT TO SIGNED NUMBER
              51    51    CE  0B28  1261          MNEGL    R1,R1                      ; ABSOLUTE VALUE
                    05        0B2B  1262 10$:     RSB                                 ;
                              0B2C  1263
                              0B2C  1264 ;
                              0B2C  1265 ;        Show names of parameters
                              0B2C  1266 ;
                              0B2C  1267 BOO$SHONAMES:                                ;
                        00FC  0B2C  1268          .WORD    ^M<R2,R3,R4,R5,R6,R7>      ;
     56    00000000'EF    9E  0B2E  1269          MOVAB    BOO$A_PRMBLK,R6            ; Set base of parameter blocks
                              0B35  1270
              57    05    D0  0B35  1271 10$:     MOVL     #5,R7                      ; Init argument count
                    66    D5  0B38  1272          TSTL     (R6)                       ; At end of list?
                    53    13  0B3A  1273          BEQL     90$                        ; Yes, finished
           51    16 A6    DE  0B3C  1274          MOVAL    PRM$T_NAME(R6),R1          ; Set parameter name address
           56    32 A6    9E  0B40  1275          MOVAB    PRM$C_LENGTH(R6),R6        ; Next parameter block
                    66    D5  0B44  1276          TSTL     (R6)                       ; At end of list?
                    26    13  0B46  1277          BEQL     70$                        ; yes
           52    16 A6    DE  0B48  1278          MOVAL    PRM$T_NAME(R6),R2          ; Set second address
```

```
                57  D6  0B4C             INCL    R7                          ; Advance argument count
        56   32 A6  9E  0B4E             MOVAB   PRM$C_LENGTH(R6),R6         ; Next argument
                66  D5  0B52             TSTL    (R6)                        ; At end of list ?
                18  13  0B54             BEQL    70$                         ; yes
        53   16 A6  DE  0B56             MOVAL   PRM$T_NAME(R6),R3           ; Set third address
                57  D6  0B5A             INCL    R7                          ; Another argument
        56   32 A6  9E  0B5C             MOVAB   PRM$C_LENGTH(R6),R6         ; Next parameter block
                66  D5  0B60             TSTL    (R6)                        ; At end of list?
                0A  13  0B62             BEQL    70$                         ; Yes
                57  D6  0B64             INCL    R7                          ; another argument
        54   16 A6  9E  0B66             MOVAB   PRM$T_NAME(R6),R4           ; Set fourth address
        56   32 A6  DE  0B6A             MOVAL   PRM$C_LENGTH(R6),R6         ; Next parameter block
                1E  BB  0B6E      70$:   PUSHR   #^M<RT,R2,R3,R4>            ; Stack args
      7E    57  04  C3  0B70             SUBL3   #4,R7,-(SP)                 ; Set number of strings on line
           FB8B CF  7F  0B74             PUSHAQ  RIO$AB_OUTBUF               ; Stack address of buffer descriptor
           FC8F CF  DF  0B78             PUSHAL  RIO$GW_OUTLEN               ; Set address of loc to receive size
           FA8E CF  7F  0B7C             PUSHAQ  NCTRLSTR                    ; Stack address of control string descr
  00000000'EF   57  FB  0B80             CALLS   R7,SYS$FAO                  ; Format value for output
             08 50  E9  0B87             BLBC    R0,100$
                        0B8A
              0383  30  0B8A             BSBW    RIO$OUTPUT_LINE             ; Output line
                A6  11  0B8D             BRB     10$                         ; Loop
                        0B8F
           50   01  D0  0B8F      90$:   MOVL    #1,R0                       ; Success status
                04      0B92      100$:  RET                                 ; Return
                        0B93
                        0B93      ;
                        0B93      ; Show name of Startup command file
                        0B93      ;
                        0B93      BOO$SHOSTART:                              ;
                  00FC  0B93             .WORD   ^M<R2,R3,R4,R5,R6,R7>       ;
                        0B95
  50    00000000'EF  9E  0B95            MOVAB   L^EXE$GT_STARTUP,R0         ; Set address of string
                        0B9C             $FAO_S  -
                        0B9C                     CTRSTR  = SCTRLSTR,-          ; Stack address of control string descr
                        0B9C                     OUTLEN  = RIO$GW_OUTLEN,-     ; Set address of loc to receive size
                        0B9C                     OUTBUF  = RIO$AB_OUTBUF,-     ; Stack address of buffer descriptor
                        0B9C                     P1      = R0                  ; Set address of startup string
                50  DD  0B9C                     PUSHL    R0
           FB61 CF  7F  0B9E                     PUSHAQ   RIO$AB_OUTBUF
           FC65 CF  3F  0BA2                     PUSHAW   RIO$GW_OUTLEN
           FA77 CF  7F  0BA6                     PUSHAQ   SCTRLSTR
  00000000'GF   04  FB  0BAA                     CALLS    #$$T2,G^SYS$FAO
             03 50  E9  0BB1             BLBC    R0,10$
                        0BB4
              0359  30  0BB4             BSBW    RIO$OUTPUT_LINE
                04      0BB7      10$:   RET                                 ;
                        0BB8
```

```
                                  0BB8  1323            .SBTTL  BOO$NOCHECK - Disable value checking
                                  0BB8  1324 ;
                                  0BB8  1325 ; Disable Value Checking and Limiting
                                  0BB8  1326 ;
                            0000  0BB8  1327 BOO$NOCHECK:    .WORD   0                            ;
        00 00000000'EF   00   E3  0BBA  1328            BBCS    #BOOCMD$V_NOCHECK,BOO$GL_CMDOPT,10$ ; Set value check inhibit
                   50    01   D0  0BC2  1329 10$:       MOVL    #1,R0                        ; Return success
                              04  0BC5  1330            RET                                  ;
```

```
                             0BC6  1332         .SBTTL  BOO$NOCHECK - Disable value checking
                             0BC6  1333
                             0BC6  1334 ;
                             0BC6  1335 ; ENABLE VALUE CHECKING AND LIMITING
                             0BC6  1336 ;
                             0BC6  1337 BOO$CHECK:                                        ;
                       0000  0BC6  1338         .WORD   0                                 ; Null entry mask
 00 00000000'EF    00    E5  0BC8  1339         BBCC    #BOOCMD$V_NOCHECK,BOO$GL_CMDOPT,10$ ; Clear check flag
            50    01    D0  0BD0  1340 10$:    MOVL    #1,R0                             ; Return with success
                        04  0BD3  1341         RET                                       ;
                             0BD4  1342
```

```
                                    0BD4          .SBTTL  BOO$SEARCH - Lookup parameter name
                                    0BD4  ;
                                    0BD4  ; Input Parameters:
                                    0BD4  ;       TPA$L_TOKENCNT(AP) - Count of characters in token
                                    0BD4  ;       TPA$L_TOKENPTR(AP) - Address of token
                                    0BD4  ;
                                    0BD4  ; Output Parameters:
                                    0BD4  ;       TPA$L_PARAM(AP) - Address of PRM block for specified parameter
                                    0BD4  ;                          name if found.
                                    0BD4  ;       R0      - 0 => Name not found
                                    0BD4  ;                 1 => Name found
                                    0BD4  ;
                              003C  0BD4  BOO$SEARCH::    .WORD   ^M<R2,R3,R4,R5> ;
                                    0BD6
            00000000'EF    0A   E1  0BD6          BBC     #BOOCMD$V_USEFILE, -
                           06       0BDD                  BOO$GL_CMDOPT,5$          ; Skip count check if not USE <file>
                                    0BDE
                  10 AC    03   91  0BDE          CMPB    #3,TPA$L_TOKENCNT(AP)     ; Check for count of characters
                           32   18  0BE2          BGEQ    50$                       ; Exit if if not > 3
                                    0BE4
        54   00000000'EF   9E       0BE4  5$:     MOVAB   BOO$A_PRMBLK,R4           ; Set base of parameter blocks
                           64   D5  0BEB  10$:    TSTL    (R4)                      ; Check for end of list
                           02   12  0BED          BNEQ    30$                       ; Not yet
                           25   11  0BEF          BRB     50$                       ; Symbol not found error
              55     16 A4      9E  0BF1  30$:    MOVAB   PRM$T_NAME(R4),R5         ; Get pointer to name string
              85     10 AC      91  0BF5          CMPB    TPA$L_TOKENCNT(AP),(R5)+; Check for too many characters
                           08   14  0BF9          BGTR    35$                       ; Yes, cant be a match
                     10 AC      29  0BFB          CMPC3   TPA$L_TOKENCNT(AP),-      ;
              65     14 BC          0BFE                  @TPA$L_TOKENPTR(AP),(R5); Is this a match?
                           06   13  0C01          BEQL    40$                       ; Yes, return PRM pointer
              54     32 A4      9E  0C03  35$:    MOVAB   PRM$C_LENGTH(R4),R4       ; Advance to nex parameter descriptor
                           E2   11  0C07          BRB     10$                       ; and try another
                  20 AC    54   D0  0C09  40$:    MOVL    R4,TPA$L_PARAM(AP)        ; Return address of parameter block
                F412 CF    54   D0  0C0D          MOVL    R4,BOO$GL_DOT             ; And save as dot
                     50    01   D0  0C12          MOVL    #1,R0                     ; Indicate success
                           04       0C15          RET                               ;  and return
         18 00000000'EF    0A   E0  0C16  50$:    BBS     #BOOCMD$V_USEFILE,BOO$GL_CMDOPT,60$; No message on 'USE filename'
                                    0C1E
                                    0C1E  .IF     NDF,CMDSW                         ; SYSBOOCMD
                                    0C1E          MSG     <-E-No suc' parameter>
                        F3DF'   30  0C1E          BSBW    BOO$FACMSG                ;
 70 20 68 63 75 73 20 6F 4E 2D 45 2D  0C21          .ASCIZ  \-E-No such parameter\              ;
       00 72 65 74 65 6D 61 72 61     0C2D
                                    0C36  .IFF                                      ; SYSGENCMD
                                    0C36          PUSHL   #SYSG$_NOPARAM            ; Set message
                                    0C36          CALLS   #1,G^LIB$SIGNAL           ; Signal
                                    0C36  .ENDC
                                    0C36
                     50    02   CE  0C36  60$:    MNEGL   #2,R0                     ; Give unique error code
                           04       0C39          RET
                                    0C3A
                                    0C3A  ;
                                    0C3A  ; BOO$DOT - Use last parameter name if any
                                    0C3A  ;
                              0000  0C3A  BOO$DOT:.WORD   0                         ; Null entry mask
          20 AC    F3E4 CF     D0   0C3C          MOVL    BOO$GL_DOT,TPA$L_PARAM(AP)      ; Get dot address
                           02   12  0C42          BNEQ    10$                       ; Have pointer
```

```
                             50   D4  0C44   1398        CLRL    R0                           ; Give error status
                                  04  0C46   1399 10$:   RET                                  ;
                                  0C47  1401         .SBTTL  BOO$SETVALUE - Store parameter value
                                  0C47  1402 ;
                                  0C47  1403 ; Input Parameters:
                                  0C47  1404 ;       TPA$L_PARAM - Address of parameter descriptor
                                  0C47  1405 ;       TPA$L_NUMBER- Value to be checked and stored
                                  0C47  1406 ;
                                  0C47  1407 ; Output Parameters:
                                  0C47  1408 ;       If value is within bounds set by parameter descriptor, the
                                  0C47  1409 ;       value is moved to the address specified by the parameter descriptor
                                  0C47  1410 ;       R0 - Completion status 0 => value out of allowable range
                                  0C47  1411 ;                              1 => legal value successfully stored
                                  0C47  1412 ;
                                  0C47  1413 BOO$SETVALUE::                                 ;
                            0010  0C47  1414         .WORD   ^M<R4>                         ; Entry mask
                                  0C49  1415
           00000000'8F        E2  0C49  1416         BBSS    #EXE$V_WRITESYSPARAMS,-        ; Set a value => write current needed
        00 00000000'GF            0C4F  1417                 G^EXE$GL_DYNAMIC_FLAGS,1$;
                                  0C55  1418 1$:
             54   20 AC       D0  0C55  1419         MOVL    TPA$L_PARAM(AP),R4             ; Get pointer to parameter descriptor
                      10      E1  0C59  1420         BBC     #PRM$V_ASCII,-                 ; Ascii parameter?
                03 10 A4          0C5B  1421                 PRM$L_FLAGS(R4),10$            ; If BC no continue
                    009D      31  0C5E  1422         BRW     65$                            ; Branch to error
                      01      DD  0C61  1423 10$:    PUSHL   #1                             ; Assume good value
      25 00000000'EF  00      E0  0C63  1424         BBS     #BOOCMD$V_NOCHECK,BOO$GL_CMDOPT,30$ ; Should values be checked
             50   08 A4       D0  0C6B  1425         MOVL    PRM$L_MIN(R4),R0               ; Get minimum allowable value
                      0D      19  0C6F  1426         BLSS    20$                            ; No minimum
             50   1C AC       D1  0C71  1427         CMPL    TPA$L_NUMBER(AP),R0            ; Check input value
                      07      1E  0C75  1428         BGEQU   20$                            ; Branch if above minimum
             1C AC    50      D0  0C77  1429         MOVL    R0,TPA$L_NUMBER(AP)            ; Use minimum value
                6E    02      CE  0C7B  1430         MNEGL   #2,(SP)                        ; Note bad value
             50   0C A4       D0  0C7E  1431 20$:    MOVL    PRM$L_MAX(R4),R0               ; Get maximum allowable value
                      0C      19  0C82  1432         BLSS    30$                            ; Branch if no maximum
             1C AC    50      D1  0C84  1433         CMPL    R0,TPA$L_NUMBER(AP)            ; Check for maximum
                      06      1E  0C88  1434         BGEQU   30$                            ; Continue if value legal
             1C AC    50      D0  0C8A  1435         MOVL    R0,TPA$L_NUMBER(AP)            ; Limit to max value
                      6E      D4  0C8E  1436         CLRL    (SP)                           ; Indicate error
             50   20 AC       D0  0C90  1437 30$:    MOVL    TPA$L_PARAM(AP),R0             ; Get address at which to store
                   50 60      D0  0C94  1438         MOVL    PRM$L_ADDR(R0),R0              ;
         50   00000000'EF40   9E  0C97  1439         MOVAB   BOO$A_SYSPARAM[R0],R0          ; Add present base of parameters
          50   00000000'8F    C2  0C9F  1440         SUBL    #BOO$A_SYSPARAM,R0             ; And subtract link-time base
        10 A4    1000 8F      B3  0CA6  1441         BITW    #PRM$M_NEG,PRM$L_FLAGS(R4)        ; Check for negative
                      05      13  0CAC  1442         BEQL    35$                            ; No
          1C AC   1C AC       CE  0CAE  1443         MNEGL   TPA$L_NUMBER(AP),TPA$L_NUMBER(AP)        ; Complement
             51   15 A4       9A  0CB3  1444 35$:    MOVZBL  PRM$B_POS(R4),R1               ; Get position
    60    14 A4   51   1C AC  F0  0CB7  1445         INSV    TPA$L_NUMBER(AP),R1,PRM$B_SIZE(R4),(R0); Set value in field
                 50   8E      D0  0CBE  1446 40$:    MOVL    (SP)+,R0                       ; Get completion status
                      1F      19  0CC1  1447         BLSS    60$                            ; Low value limit
                   1B 50      E8  0CC3  1448         BLBS    R0,50$                         ; Success, return
                                  0CC6  1449
                                  0CC6  1450 .IF     NDF,CMDSW               ; SYSBOOCMD
                                  0CC6  1451
                                  0CC6  1452         MSG     <-W-Value set to maximum>
                   F337'      30  0CC6           BSBW    BOO$FACMSG                     ;
74 65 73 20 65 75 6C 61 56 2D 57 2D  0CC9           .ASCIZ  \-W-Value set to maximum\                   ;
00 6D 75 6D 69 78 61 6D 20 6F 74 20  0CD5
                      04  0CE1  1453 50$:    RET                                    ; and return
                                  0CE2  1454 60$:    MSG     <-W-Value set to minimum>
```

```
                         F31B'  30  0CE2                      BSBW    BOO$FACMSG                 ;
74 65 73 20 65 75 6C 61 56 2D 57 2D 0CE5                      .ASCIZ  \-W-Value set to minimum\                  ;
00 6D 75 6D 69 6E 69 6D 20 6F 74 20 0CF1
                                04  0CFD  1455                RET                                ;
                                    0CFE  1456 65$:           MSG     <-E-Parameter is not numeric type>
                         F2FF'  30  0CFE                      BSBW    BOO$FACMSG                 ;
72 65 74 65 6D 61 72 61 50 2D 45 2D 0D01                      .ASCIZ  \-E-Parameter is not numeric type\                ;
65 6D 75 6E 20 74 6F 6E 20 73 69 20 0D0D
         00 65 70 79 74 20 63 69 72 0D19
                                04  0D22  1457                RET
                                    0D23  1458
                                    0D23  1459 .IFF                                    ; SYSGENCMD
                                    0D23  1460
                                    0D23  1461                PUSHAB  PRM$T_NAME(R4)           ; Address of parameter name
                                    0D23  1462                PUSHL   #1                       ; Number of FAO param's
                                    0D23  1463                PUSHL   #SYSG$_SETMAX            ; Error status
                                    0D23  1464                BRB     70$
                                    0D23  1465 50$:           RET
                                    0D23  1466
                                    0D23  1467 60$:           PUSHAB  PRM$T_NAME(R4)           ; Address of parameter name
                                    0D23  1468                PUSHL   #1                       ; Number of FAO param's
                                    0D23  1469                PUSHL   #SYSG$_SETMIN            ; Error status
                                    0D23  1470                BRB     65$                      ;
                                    0D23  1471 65$:           PUSHAB  PRM$T_NAME(R4)           ; Address of parameter name
                                    0D23  1472                PUSHL   #1                       ; Number of FAO param's
                                    0D23  1473                PUSHL   #SYSG$_NOTASCII          ; Error status
                                    0D23  1474 70$:           CALLS   #3,G^LIB$SIGNAL          ; Signal
                                    0D23  1475                MOVL    #SS$_NORMAL,R0           ; Set success
                                    0D23  1476                RET                              ; and return
                                    0D23  1477
                                    0D23  1478 .ENDC
                                    0D23  1479
                                    0D23  1480 ;
                                    0D23  1481 ; Set to default value
                                    0D23  1482 ;
                                    0D23  1483 BOO$SETDEF:                                     ;
                          00FC      0D23  1484                .WORD   ^M<R2,R3,R4,R5,R6,R7>    ;
              54     20 AC   D0     0D25  1485                MOVL    TPA$L_PARAM(AP),R4       ; Get address of parameter block
                        10   E0     0D29  1486                BBS     #PRM$V_ASCII,-           ; Ascii parameter?
                  08 10 A4          0D2B  1487                        PRM$L_FLAGS(R4),10$      ; If BS yes
           1C AC     04 A4   D0     0D2E  1488                MOVL    PRM$L_DEFAULT(R4),TPA$L_NUMBER(AP); Set default as value
                      FF13   31     0D33  1489                BRW     BOO$SETVALUE+2           ; Call routine to set the value
           14 AC     04 A4   9E     0D36  1490 10$:           MOVAB   PRM$L_DEFAULT(R4),TPA$L_TOKENPTR(AP); Set ptr to default string
              54     14 A4   9A     0D3B  1491                MOVZBL  PRM$B_SIZE(R4),R4        ; Get size in bits
      10 AC   54     FD 8F   78     0D3F  1492                ASHL    #-3,R4,TPA$L_TOKENCNT(AP); Set size in bytes
                       09'   11     0D45  1493                BRB     BOO$SETASCII+2           ; Call routine to set the default string
                                    0D47  1494
                                    0D47  1495 ;
                                    0D47  1496 ; Set acsii parameter to all blanks
                                    0D47  1497 ;
                                    0D47  1498 BOO$SETBLANK:
                          00FC      0D47  1499                .WORD   ^M<R2,R3,R4,R5,R6,R7>
                     10 AC   D4     0D49  1500                CLRL    TPA$L_TOKENCNT(AP)       ; Set string count zero (null string)
                       02'   11     0D4C  1501                BRB     BOO$SETASCII+2           ; join common code
                                    0D4E  1502
                                    0D4E          .SBTTL  BOO$SETASCII - Action routine to set ASCII parameter type
                                    0D4E  ;
                                    0D4E  ; Input Parameters:
                                    0D4E  ;         TPA$L_PARAM(AP) - Address of parameter descriptor
                                    0D4E  ;         TPA$L_TOKENCNT(AP) - Length of parsed string
                                    0D4E  ;         TPA$L_TOKENPTR(AP) - Address of parsed string
                                    0D4E  ;
                                    0D4E  ; Output Parameters:
                                    0D4E  ;         The parameter is checked to ensure it is ASCII type, then length
                                    0D4E  ;         of the parsed string is compared to size of parameter.  If no
                                    0D4E  ;         error, then parameter is set to new string.
                                    0D4E  ;
                                    0D4E  BOO$SETASCII::
                              00FC  0D4E          .WORD   ^M<R2,R3,R4,R5,R6,R7>
                                    0D50
           00000000'8F        E2    0D50          BBSS    #EXE$V_WRITESYSPARAMS,- ; Set an value => write current needed
        00 00000000'GF              0D56                  G^EXE$GL_DYNAMIC_FLAGS,1$;
                                    0D5C  1$:
           00 04 AC   00      E5    0D5C          BBCC    #TPA$V_BLANKS,TPA$L_OPTIONS(AP),2$; Make blanks no longer significan
           56    20 AC        D0    0D61  2$:     MOVL    TPA$L_PARAM(AP),R6        ; Get address of parameter block
           10 A6    10        E0    0D65          BBS     #PRM$V_ASCII,PRM$L_FLAGS(R6),-
                    03              0D69                  5$                        ; If set, then ASCII type
                  00D7        31    0D6A          BRW     90$
           57    14 A6        9A    0D6D  5$:     MOVZBL  PRM$B_SIZE(R6),R7         ; Get size (in bits)
       57  57    FD 8F        78    0D71          ASHL    #-3,R7,R7                 ; Convert size from bit to byte count
           57    10 AC        D1    0D76          CMPL    TPA$L_TOKENCNT(AP),R7     ; Compare with parsed string size
                    03        1B    0D7A          BLEQU   10$                       ; If LEQU, then fits
                  00A1        31    0D7C          BRW     80$                       ; Else string too big
                    01        DD    0D7F  10$:    PUSHL   #1                        ; Assume success
              5E    10        C2    0D81          SUBL    #16,SP                    ; Make room for octaword buffer on stack
           53    14 AC        D0    0D84          MOVL    TPA$L_TOKENPTR(AP),R3     ; Get address of token
           54    10 AC        D0    0D88          MOVL    TPA$L_TOKENCNT(AP),R4     ; Get count of token
  6E    57    20    63    54    2C    0D8C          MOVC5   R4,(R3),#^A/ /,R7,(SP)    ; New value on stack temporarily
     2D 00000000'EF   00      E0    0D92          BBS     #BOOCMD$V_NOCHECK,BOO$GL_CMDOPT,30$ ; If checks disabled, branch
       08 A6    6E    57      2D    0D9A          CMPC5   R7,(SP),PRM$L_MIN(R6),-; Compare min value with parsed value
           08 A6    04              0D9F                  #4,PRM$L_MIN(R6)          ;
                    0E        1E    0DA2          BGEQU   20$                       ; Branch if input is greater
           08 A6    04        2C    0DA4          MOVC5   #4,PRM$L_MIN(R6),-        ; Set min value
       6E    57    08 A6            0DA8                  PRM$L_MIN(R6),R7,(SP)     ;
           10 AE    02        CE    0DAC          MNEGL   #2,16(SP)                 ; Ind error
                    15        11    0DB0          BRB     30$
           0C A6    04        2D    0DB2  20$:    CMPC5   #4,PRM$L_MAX(R6),-        ; Compare max value with parsed value
       6E    57    0C A6            0DB6                  PRM$L_MAX(R6),R7,(SP)     ;
                    0B        1E    0DBA          BGEQU   30$                       ; Branch if input is greater
           0C A6    04        2C    0DBC          MOVC5   #4,PRM$L_MAX(R6),-        ; Set max value
       6E    57    0C A6            0DC0                  PRM$L_MAX(R6),R7,(SP)     ;
                 10 AE        D4    0DC4          CLRL    16(SP)                    ; Ind error
              50    66        D0    0DC7  30$:    MOVL    PRM$L_ADDR(R6),R0         ; Get address parameter
     50    00000000'EF40      9E    0DCA          MOVAB   BOO$A_SYSPARAM[R0],R0     ; Add present base of parameters
       50    00000000'8F      C2    0DD2          SUBL    #BOO$A_SYSPARAM,R0        ; And subtract link-time base
          60    6E    57      28    0DD9          MOVC3   R7,(SP),(R0)              ; Set value in system
              5E    10        C0    0DDD          ADDL    #16,SP                    ; Remove value from stack
              50    8E        D0    0DE0          MOVL    (SP)+,R0                  ; Get status
                    1F        19    0DE3          BLSS    60$                       ; If neg, value set to min
                 1B 50        E8    0DE5          BLBS    R0,50$                    ; If LBS, success
                                    0DE8
                                    0DE8  .IF     NDF,CMDSW               ; SYSBOOCMD
                                    0DE8  1561
                                    0DE8  1562          MSG     <-W-Value set to maximum>
                        F215'  30   0DE8                BSBW    BOO$FACMSG              ;
74 65 73 20 65 75 6C 61 56 2D 57 2D 0DEB                .ASCIZ  \-W-Value set to maximum\                   ;
00 6D 75 6D 69 78 61 6D 20 6F 74 20 0DF7
                               04   0E03  1563 50$:     RET                     ; and return
                                    0E04  1564 60$:     MSG     <-W-Value set to minimum>
                        F1F9'  30   0E04                BSBW    BOO$FACMSG              ;
74 65 73 20 65 75 6C 61 56 2D 57 2D 0E07                .ASCIZ  \-W-Value set to minimum\                   ;
00 6D 75 6D 69 6E 69 6D 20 6F 74 20 0E13
                               04   0E1F  1565          RET                     ;
                                    0E20  1566
                                    0E20  1567 .IFF                             ; SYSGENCMD
                                    0E20  1568
                                    0E20  1569          PUSHAB  PRM$T_NAME(R4)           ; Address of parameter name
                                    0E20  1570          PUSHL   #1                       ; Number of FAO param's
                                    0E20  1571          PUSHL   #SYSG$_SETMAX            ; Error status
                                    0E20  1572          BRB     70$
                                    0E20  1573 50$:     RET
                                    0E20  1574
                                    0E20  1575 60$:     PUSHAB  PRM$T_NAME(R4)           ; Address of parameter name
                                    0E20  1576          PUSHL   #1                       ; Number of FAO param's
                                    0E20  1577          PUSHL   #SYSG$_SETMIN            ; Error status
                                    0E20  1578 70$:     CALLS   #3,G^LIB$SIGNAL          ; Signal
                                    0E20  1579 75$:     MOVL    #SS$_NORMAL,R0           ; Set success
                                    0E20  1580          RET                     ; and return
                                    0E20  1581
                                    0E20  1582 .ENDC
                                    0E20  1583
                                    0E20  1584          .IF NDF,CMDSW                    ; SYSBOOCMD
                                    0E20  1585
                                    0E20  1586 80$:
                                    0E20  1587          MSG     <-E-Specified string is too long>
                        F1DD'  30   0E20                BSBW    BOO$FACMSG              ;
64 65 69 66 69 63 65 70 53 2D 45 2D 0E23                .ASCIZ  \-E-Specified string is too long\                   ;
74 20 73 69 20 67 6E 69 72 74 73 20 0E2F
         00 67 6E 6F 6C 20 6F 6F    0E3B
                               04   0E43  1588          RET
                                    0E44  1589
                                    0E44  1590 90$:
                                    0E44  1591          MSG     <-E-Parameter is not ASCII type>
                        F1B9'  30   0E44                BSBW    BOO$FACMSG              ;
72 65 74 65 6D 61 72 61 50 2D 45 2D 0E47                .ASCIZ  \-E-Parameter is not ASCII type\                    ;
49 43 53 41 20 74 6F 6E 20 73 69 20 0E53
            00 65 70 79 74 20 49    0E5F
                               04   0E66  1592          RET
                                    0E67  1593
                                    0E67  1594          .IFF                             ; SYSGENCMD
                                    0E67  1595
                                    0E67  1596 80$:
                                    0E67  1597          PUSHL   #SYSG$_STRTOOLNG         ; Error status
                                    0E67  1598          CALLS   #1,G^LIB$SIGNAL          ; Output it
                                    0E67  1599          BRB     75$
                                    0E67  1600
                                    0E67  1601 90$:
                                    0E67  1602          PUSHAB  PRM$T_NAME(R4)           ; Address of parameter name
                                    0E67  1603          PUSHL   #1                       ; FAO arg count
```

```
0E67  1604        PUSHL   #SYSG$_NOTASCII         ; Error status
0E67  1605        CALLS   #3,G^LIB$SIGNAL         ; Output it
0E67  1606        BRB     75$
0E67  1607
0E67  1608        .ENDC
```

```
                              0E67          .SBTTL  BOO$SHOVALUE - Action routine to show single value
                              0E67     ;
                              0E67     ; Input Parameters:
                              0E67     ;        TPA$L_PARAMETER(AP) - Address of parameter block
                              0E67     ;
                              0E67     BOO$SHOVALUE:                                     ;
                        003C  0E67              .WORD   ^M<R2,R3,R4,R5>                   ;
                              0E69     ;
                              0E69     ; Output header
                              0E69     ;
                              0E69     .IF      DF,CMDSW                               ; SYSGEN Only
                              0E69              BBC     #BOOCMD$V_TERMINAL,-            ; Output header to terminals only
                              0E69                      BOO$GL_CMDOPT,10$               ;
                              0E69              .ENDC
                              0E69
           009E 8F        28  0E69              MOVC3   #SDVHDRLEN,-
  F898 CF  F7F5 CF            0E6D                      SDVHDR,RIO$AB_BUFFER            ; Move in string
  F991 CF  009E 8F        B0  0E73              MOVW    #SDVHDRLEN,RIO$GW_OUTLEN        ; Set length
              0093        30  0E7A              BSBW    RIO$OUTPUT_LINE                 ; Output line
                              0E7D
       54     20 AC       D0  0E7D     10$:     MOVL    TPA$L_PARAM(AP),R4              ; Get address of parameter block
              FB93        30  0E81              BSBW    BOO$SHOWV                       ; Show value
                          04  0E84              RET                                     ; Return with BOO$SHOWV status
                              0E85
                          0E85          .SBTTL  BOO$SHOALL - Action routine to show all parameter values
                          0E85    ;
                          0E85    ; Input Parameters:
                          0E85    ;       (AP)            Pointer to the TPARSE table
                          0E85    ;       TPA$L_PARAM(AP) The mask of acceptable types
                          0E85    ; Output Parameters:
                          0E85    ;       All parameters except special parameters are displayed.
                          0E85    ;
                          0E85    BOO$SHOALL:                                   ;
                   00FC   0E85          .WORD   ^M<R2,R3,R4,R5,R6,R7>   ;
                          0E87
                          0E87    .IF     DF,CMDSW                        ; SYSGEN ONLY
                          0E87
                          0E87          BBC     #BOOCMD$V_TERMINAL,-
                          0E87                  BOO$GL_CMDOPT,5$        ; If terminal,
                          0E87          CLRQ    -(SP)                   ;  clear the whole screen
                          0E87          CALLS   #2,G^SCR$ERASE_PAGE
                          0E87    5$:
                          0E87    ;
                          0E87    ; Format "Parameters in use message"
                          0E87    ;
                          0E87          $FAO_S  CTRSTR = CTR_PARINUSE,-
                          0E87                  OUTLEN = RIO$GW_OUTLEN,-
                          0E87                  OUTBUF = RIO$AB_OUTBUF,-
                          0E87                  P1 = BOO$GL_PARINUSE
                          0E87          BSBW    RIO$OUTPUT_LINE
                          0E87
                          0E87    .ENDC
                          0E87
          009E 8F     28  0E87          MOVC3   #SDVHDRLEN,-
 F87A CF  F7D7 CF         0E8B                  SDVHDR,RIO$AB_BUFFER    ; Move in string
 F973 CF  009E 8F     B0  0E91          MOVW    #SDVHDRLEN,RIO$GW_OUTLEN ; Set length
             0075     30  0E98          BSBW    RIO$OUTPUT_LINE         ; Output line
                          0E9B
                          0E9B    .IF     DF,CMDSW                        ; SYSGEN ONLY
                          0E9B          BBC     #BOOCMD$V_TERMINAL,-
                          0E9B                  BOO$GL_CMDOPT,7$        ; If terminal,
                          0E9B          PUSHL   #24                     ;  use only 24 lines
                          0E9B          PUSHL   #5                      ;  and scroll only the bottom portion
                          0E9B          CALLS   #2,G^SCR$SET_SCROLL     ;  and setup a scrolling region
                          0E9B    7$:
                          0E9B    .ENDC
                          0E9B
    54    00000000'EF     9E  0E9B          MOVAB   BOO$A_PRMBLK,R4         ; Set starting parameter block address
          55     20 AC    D0  0EA2          MOVL    TPA$L_PARAM(AP),R5      ; Set mask of acceptable types
                          0EA6    ;
                          0EA6    ; Loop through all parameters
                          0EA6    ;
                  64      D5  0EA6    10$:    TSTL    (R4)                    ; Check for end of list
                  1C      13  0EA8          BEQL    50$                     ;  yes, done
          06 55   1F      E0  0EAA          BBS     #PRM$V_ALL,R5,20$       ; Branch if SHOW/ALL
          10 A4   55      D3  0EAE          BITL    R5,PRM$L_FLAGS(R4)      ; Is this one to output?
                  0C      13  0EB2          BEQL    40$                     ; No, try another
                  07      E1  0EB4    20$:    BBC     #PRM$V_SPECIAL,-        ; Yes, is it a special parameter?
           04 10 A4           0EB6                  PRM$L_FLAGS(R4),30$     ; Branch if not
          03 55   07      E1  0EB9          BBC     #PRM$V_SPECIAL,R5,40$   ; It's special - branch if unasked for
                          0EBD    30$:
```

```
           FB57  30  0EBD            BSBW    BOO$SHOWV               ; Display values
    54    32 A4  9E  0EC0  40$:      MOVAB   PRM$C_LENGTH(R4),R4     ; Next parameter block
             E0  11  0EC4            BRB     10$                     ;
                     0EC6  50$:
                     0EC6
                     0EC6  .IF       DF,CMDSW
                     0EC6            BBC     #BOOCMD$V_TERMINAL,-
                     0EC6                    BOO$GL_CMDOPT,60$       ; If terminal,
                     0EC6            PUSHL   #24                     ;  Use only 24 lines
                     0EC6            PUSHL   #1                      ;  and scroll only the bottom portion
                     0EC6            CALLS   #2,G^SCR$SET_SCROLL     ;  and setup a scrolling region
                     0EC6  60$:
                     0EC6  .ENDC
                 04  0EC6            RET                             ;
```

```
                                 0EC7  ;
                                 0EC7  ; Set name of startup command file
                                 0EC7  ;
                                 0EC7  BOO$SETSTART:                                    ;
                          00FC   0EC7            .WORD    ^M<R2,R3,R4,R5,R6,R7>         ;
                                 0EC9
           00000000'8F     E2    0EC9            BBSS     #EXE$V_WRITESYSPARAMS,-      ; Set startup name => write current needed
        00 00000000'GF           0ECF                     G^EXE$GL_DYNAMIC_FLAGS,1$;
                                 0ED5  1$:
    56     00000000'EF     9E    0ED5            MOVAB    L^EXE$GT_STARTUP,R6          ; Point to slot for startup file name
                    50     D4    0EDC            CLRL     R0                           ; Assume error
        F92A CF     1F     91    0EDE            CMPB     #31,BOO$GB_FILELEN           ; Check for fit
                    01     18    0EE3            BGEQ     10$                          ; Continue if legal size
                           04    0EE5            RET                                   ;
        50    F923 CF      9A    0EE6  10$:      MOVZBL   BOO$GB_FILELEN,R0            ; Get count
              86    50     90    0EEB            MOVB     R0,(R6)+                     ; Set count for string
    66     F91B DF  50     28    0EEE            MOVC3    R0,@BOO$GL_FILEADDR,(R6);    ; Set file name
              50    01     3C    0EF4            MOVZWL   #1,R0                        ; Return success indication
                           04    0EF7            RET                                   ;
                                 0EF8
                          0EF8        .IF     NDF,CMDSW               ; SYSBOOCMD ONLY
                          0EF8
                          0EF8        .SBTTL  BOO$MSGOUT - Output message
                          0EF8   ;
                          0EF8   ; Calling Sequence:
                          0EF8   ;        BSBW    BOO$MSGOUT
                          0EF8   ;        .ASCIZ  message_string
                          0EF8   ;
                          0EF8   BOO$MSGOUT::                         ;
                  7E   7C 0EF8        CLRQ    -(SP)                   ; Null read buffer
               08 AE   DD 0EFA        PUSHL   8(SP)                   ; Address of string
      00000000'EF 03   FB 0EFD        CALLS   #3,L^BOO$READPROMPT     ; Output string
  9E    FA00 8F   00   3A 0F04        LOCC    #0,#64000,@(SP)+        ; Find end of string
               50 01   D0 0F0A        MOVL    #1,R0                   ; Set success code
            01 A1      17 0F0D        JMP     1(R1)                   ; Return to caller
                          0F10
                          0F10   ;+
                          0F10   ; This routine is in RMSCONIO for SYSGEN, is used here to map SYSBOOT
                          0F10   ; calls to this routine into calls to BOO$READPROMPT.
                          0F10   ;
                          0F10   ; Inputs:
                          0F10   ;        RIO$GW_OUTLEN - length of string to output
                          0F10   ;        RIO$AB_BUFFER - buffer to output
                          0F10   ;-
                          0F10
                          0F10   RIO$OUTPUT_LINE::
                          0F10
                  7E   51 7D 0F10     MOVQ    R1,-(SP)                ; Save R1,R2
          51   F8F4 CF    3C 0F13     MOVZWL  RIO$GW_OUTLEN,R1        ; Set length
          52   F7EF CF    9E 0F18     MOVAB   RIO$AB_BUFFER,R2        ; Set address
              51   6241   9E 0F1D     MOVAB   (R2)[R1],R1             ; Set address of end of string
  61    00000A0D 8F       D0 0F21     MOVL    #^X00000A0D,(R1)        ; Set CR, LF, zero byte at end
                          0F28
                  7E   7C 0F28        CLRQ    -(SP)                   ; Null read buffer
                  52   DD 0F2A        PUSHL   R2                      ; Address of string
      00000000'EF 03   FB 0F2C        CALLS   #3,L^BOO$READPROMPT     ; Output string
                          0F33
               51 8E   7D 0F33        MOVQ    (SP)+,R1                ; Restore R1,R2
                       05 0F36        RSB                             ; Return
                          0F37
```

```
                                  0F37  1769          .SBTTL  DUMMY COMMAND ROUTINES FOR COMMANDS NOT IN SYSBOOT
                                  0F37  1770 BOO$SET_OUTPUT::
                                  0F37  1771 BOO$USEACT::
                                  0F37  1772 SYS$ASCTOID::
                                  0F37  1773 SYS$FILESCAN::
                                  0F37  1774
                            0000  0F37  1775          .WORD   0                        ; Null entry mask
                                  0F39  1776          MSG     <-E-Syntax error>        ; SYSBOOT error message
                      F0C4'  30   0F39                BSBW    BOO$FACMSG               ;
72 65 20 78 61 74 6E 79 53 2D 45 2D  0F3C             .ASCIZ  \-E-Syntax error\                  ;
                   00 72 6F 72    0F48
               50    01    D0     0F4C  1777          MOVL    #1,R0                    ;
                            04    0F4F  1778          RET                              ;
                                  0F50  1779
                                  0F50  1780          .ENDC   ; End of SYSBOOT conditional code
                                  0F50  1781
                                  0F50  1782          .END                             ;
```

```
$$$CNT                   = 00000003                BOOCMD$M_DISHEX          = 00000800
$$$FLG                   = FFFFFFFF                BOOCMD$M_NOCHECK         = 00000001
$$$KEY                   = 00000021                BOOCMD$M_SETOUTPUT       = 00008000
$$$KFG                   = FFFFFFFF                BOOCMD$M_TERMINAL        = 00010000
$$$MOD                   = 00000002                BOOCMD$M_USEFILE         = 00000400
$$$TMP                   = 000000CE R      04      BOOCMD$V_CONT            = 00000008
$$KEYTAB                 = 00000000 R      03      BOOCMD$V_DEFAULT         = 00000009
$$T2                     = 00000004                BOOCMD$V_DISHEX          = 0000000B
ASCII                      0000008C R      02      BOOCMD$V_NOCHECK         = 00000000
ASCSTR                     000005CE R      06      BOOCMD$V_USEFILE         = 00000004
BLANKS                     00000579 R      06      BUFFER_SIZE              = 00000100
BOO$A_PRMBLK               ********   X    06      CR                       = 0000000D
BOO$A_SYSPARAM             ********   X    06      CTRLSTR                    0000057E R      06
BOO$CHECK                  00000BC6 R      06      CTR_PARINUSE               00000645 R      06
BOO$C_COMBUFSZ           = 000000C8  G             CUR_BLANKS                 00000567 R      06
BOO$C_COMSTRLEN          = 00000400  G             DISABLCMD                  00000042 R      02
BOO$DOT                    00000C3A R      06      ENABLCMD                   0000004C R      02
BOO$FACMSG                 ********   X    06      EXE$A_SYSPARAM             ********   X    06
BOO$FILESPEC               000009A0 RG     06      EXE$C_SYSPARSZ             ********   X    06
BOO$GB_FILELEN             0000080D R      06      EXE$GL_DYNAMIC_FLAGS       ********   X    06
BOO$GETPARAM               00000816 RG     06      EXE$GL_TODR                ********   X    06
BOO$GIVEHELP               ********   X    02      EXE$GQ_TODCBASE            ********   X    06
BOO$GL_CMDOPT              00000000 RG     05      EXE$GT_STARTUP             ********   X    06
BOO$GL_DOT                 00000024 RG     06      EXE$V_WRITESYSPARAMS       ********   X    06
BOO$GL_FILEADDR            0000080E R      06      EXIT                       0000099C R      06
BOO$GL_PARINUSE            00000812 R      06      FF                       = 0000000C
BOO$GQ_FILDESC             00000028 RG     06      FILESPEC                   000001EA R      02
BOO$GT_COMBUF              00000070 RG     06      GETDATA                    00000B0C R      06
BOO$GT_COMSTR              00000138 RG     06      HEXNUM                     00000202 R      02
BOO$GT_CURRENT             00000812 R      06      HEXQUAL                    00000194 R      02
BOO$GT_DEFAULT             00000812 R      06      HEXQUAL2                   0000018A R      02
BOO$GT_FILNAME             00000030 RG     06      HEXSTR                     000005A6 R      06
BOO$GT_PROMPT              ********   X    06      KEYTBL                     00000000 RG     03
BOO$GT_SYSNAME             00000538 RG     06      LF                       = 0000000A
BOO$GT_SYSPARNAME          0000054B RG     06      LIB$TPARSE                 ********   X    06
BOO$MSGOUT                 00000EF8 RG     06      LIB$_SYNTAXERR             ********   X    06
BOO$NOCHECK                00000BB8 R      06      NCTRLSTR                   0000060E R      06
BOO$READPROMPT             ********   X    06      NUMBER                     000001F2 R      02
BOO$SEARCH                 00000BD4 RG     06      OP$_ACBD                 = 0000006F
BOO$SETASCII               00000D4E RG     06      OP$_ACBF                 = 0000004F
BOO$SETBLANK               00000D47 R      06      OP$_ACBG                 = 00004FFD
BOO$SETDEF                 00000D23 R      06      OP$_ACBH                 = 00006FFD
BOO$SETSTART               00000EC7 R      06      OP$_ADDD2                = 00000060
BOO$SETVALUE               00000C47 RG     06      OP$_ADDD3                = 00000061
BOO$SET_OUTPUT             00000F37 RG     06      OP$_ADDF2                = 00000040
BOO$SHOALL                 00000E85 R      06      OP$_ADDF3                = 00000041
BOO$SHONAMES               00000B2C R      06      OP$_ADDG2                = 000040FD
BOO$SHOSTART               00000B93 R      06      OP$_ADDG3                = 000041FD
BOO$SHOVALUE               00000E67 R      06      OP$_ADDH2                = 000060FD
BOO$SHOWV                  00000A17 R      06      OP$_ADDH3                = 000061FD
BOO$T_DYNAMIC              00000564 R      06      OP$_ADDP4                = 00000020
BOO$T_NODYNAMIC            00000566 R      06      OP$_ADDP6                = 00000021
BOO$USEACT                 00000F37 RG     06      OP$_ASHP                 = 000000F8
BOO$USECUR                 000009E0 RG     06      OP$_CLRD                 = 0000007C
BOO$USEFILE                ********   X    02      OP$_CLRF                 = 000000D4
BOOCMD$M_CONT            = 00000100                OP$_CLRG                 = 0000007C
BOOCMD$M_DEFAULT         = 00000200                OP$_CLRH                 = 00007CFD
```

```
OP$_CMPD                         = 00000071              OP$_DIVH2                        = 000066FD
OP$_CMPF                         = 00000051              OP$_DIVH3                        = 000067FD
OP$_CMPG                         = 000051FD              OP$_DIVP                         = 00000027
OP$_CMPH                         = 000071FD              OP$_EDITPC                       = 00000038
OP$_CMPP3                        = 00000035              OP$_EMODD                        = 00000074
OP$_CMPP4                        = 00000037              OP$_EMODF                        = 00000054
OP$_CRC                          = 0000000B              OP$_EMODG                        = 000054FD
OP$_CVTBD                        = 0000006C              OP$_EMODH                        = 000074FD
OP$_CVTBF                        = 0000004C              OP$_MATCHC                       = 00000039
OP$_CVTBG                        = 00004CFD              OP$_MNEGD                        = 00000072
OP$_CVTBH                        = 00006CFD              OP$_MNEGF                        = 00000052
OP$_CVTDB                        = 00000068              OP$_MNEGG                        = 000052FD
OP$_CVTDF                        = 00000076              OP$_MNEGH                        = 000072FD
OP$_CVTDH                        = 000032FD              OP$_MOVD                         = 00000070
OP$_CVTDL                        = 0000006A              OP$_MOVF                         = 00000050
OP$_CVTDW                        = 00000069              OP$_MOVG                         = 000050FD
OP$_CVTFB                        = 00000048              OP$_MOVH                         = 000070FD
OP$_CVTFD                        = 00000056              OP$_MOVP                         = 00000034
OP$_CVTFG                        = 000099FD              OP$_MOVTC                        = 0000002E
OP$_CVTFH                        = 000098FD              OP$_MOVTUC                       = 0000002F
OP$_CVTFL                        = 0000004A              OP$_MULD2                        = 00000064
OP$_CVTFW                        = 00000049              OP$_MULD3                        = 00000065
OP$_CVTGB                        = 000048FD              OP$_MULF2                        = 00000044
OP$_CVTGF                        = 000033FD              OP$_MULF3                        = 00000045
OP$_CVTGH                        = 000056FD              OP$_MULG2                        = 000044FD
OP$_CVTGL                        = 00004AFD              OP$_MULG3                        = 000045FD
OP$_CVTGW                        = 000049FD              OP$_MULH2                        = 000064FD
OP$_CVTHB                        = 000068FD              OP$_MULH3                        = 000065FD
OP$_CVTHD                        = 0000F7FD              OP$_MULP                         = 00000025
OP$_CVTHF                        = 0000F6FD              OP$_POLYD                        = 00000075
OP$_CVTHG                        = 000076FD              OP$_POLYF                        = 00000055
OP$_CVTHL                        = 00006AFD              OP$_POLYG                        = 000055FD
OP$_CVTHW                        = 000069FD              OP$_POLYH                        = 000075FD
OP$_CVTLD                        = 0000006E              OP$_SCANC                        = 0000002A
OP$_CVTLF                        = 0000004E              OP$_SKPC                         = 0000003B
OP$_CVTLG                        = 00004EFD              OP$_SPANC                        = 0000002B
OP$_CVTLH                        = 00006EFD              OP$_SUBD2                        = 00000062
OP$_CVTLP                        = 000000F9              OP$_SUBD3                        = 00000063
OP$_CVTPL                        = 00000036              OP$_SUBF2                        = 00000042
OP$_CVTPS                        = 00000008              OP$_SUBF3                        = 00000043
OP$_CVTPT                        = 00000024              OP$_SUBG2                        = 000042FD
OP$_CVTRDL                       = 0000006B              OP$_SUBG3                        = 000043FD
OP$_CVTRFL                       = 0000004B              OP$_SUBH2                        = 000062FD
OP$_CVTRGL                       = 00004BFD              OP$_SUBH3                        = 000063FD
OP$_CVTRHL                       = 00006BFD              OP$_SUBP4                        = 00000022
OP$_CVTSP                        = 00000009              OP$_SUBP6                        = 00000023
OP$_CVTTP                        = 00000026              OP$_TSTD                         = 00000073
OP$_CVTWD                        = 0000006D              OP$_TSTF                         = 00000053
OP$_CVTWF                        = 0000004D              OP$_TSTG                         = 000053FD
OP$_CVTWG                        = 00004DFD              OP$_TSTH                         = 000073FD
OP$_CVTWH                        = 00006DFD              PARMBLK                            00000000 R      06
OP$_DIVD2                        = 00000066              PARSE                              000008F7 R      06
OP$_DIVD3                        = 00000067              PRM$B_POS                        = 00000015
OP$_DIVF2                        = 00000046              PRM$B_SIZE                       = 00000014
OP$_DIVF3                        = 00000047              PRM$C_LENGTH                     = 00000032
OP$_DIVG2                        = 000046FD              PRM$L_ADDR                       = 00000000
OP$_DIVG3                        = 000047FD              PRM$L_DEFAULT                    = 00000004
```

```
PRM$L_FLAGS                     = 00000010                  TPA$L_STRINGPTR              = 0000000C
PRM$L_MAX                       = 0000000C                  TPA$L_TOKENCNT               = 00000010
PRM$L_MIN                       = 00000008                  TPA$L_TOKENPTR               = 00000014
PRM$M_ACP                       = 00000008                  TPA$M_ABBREV                 = 00000002
PRM$M_ALL                       = 80000000                  TPA$M_BLANKS                 = 00000001
PRM$M_CLUSTER                   = 00008000                  TPA$V_BLANKS                 = 00000000
PRM$M_DYNAMIC                   = 00000001                  TPA$_ALPHA                   = 000001EE
PRM$M_JBC                       = 00000010                  TPA$_ANY                     = 000001ED
PRM$M_LGI                       = 00020000                  TPA$_BLANK                   = 000001F2
PRM$M_MAJOR                     = 00000400                  TPA$_DECIMAL                 = 000001F3
PRM$M_NEG                       = 00001000                  TPA$_DIGIT                   = 000001EF
PRM$M_PQL                       = 00000800                  TPA$_EOS                     = 000001F7
PRM$M_RMS                       = 00000020                  TPA$_EXIT                    = FFFFFFFF
PRM$M_SCS                       = 00004000                  TPA$_FAIL                    = FFFFFFFE
PRM$M_SPECIAL                   = 00000080                  TPA$_FILESPEC                = 000001EA
PRM$M_SYS                       = 00000040                  TPA$_HEX                     = 000001F5
PRM$M_SYSGEN                    = 00000004                  TPA$_IDENT                   = 000001EC
PRM$M_TTY                       = 00002000                  TPA$_KEYWORD                 = 00000100
PRM$T_NAME                      = 00000016                  TPA$_LAMBDA                  = 000001F6
PRM$T_UNIT                      = 00000026                  TPA$_MAXKEY                  = 000000DC
PRM$V_ALL                       = 0000001F                  TPA$_OCTAL                   = 000001F4
PRM$V_ASCII                     = 00000010                  TPA$_STRING                  = 000001F0
PRM$V_DYNAMIC                   = 00000000                  TPA$_SUBXPR                  = 000001F8
PRM$V_SPECIAL                   = 00000007                  TPA$_SYMBOL                  = 000001F1
READCMD                           00000842 R     06         TPA$_UIC                     = 000001EB
READLINE                          00000850 R     06         USEACT                         000001D2 R     02
RIO$AB_BUFFER                     0000070B RG    06         USECMD                         000001A6 R     02
RIO$AB_OUTBUF                     00000703 RG    06         USECUR                         000001C8 R     02
RIO$GW_OUTLEN                     0000080B RG    06         USEDEF                         000001DC R     02
RIO$OUTPUT_LINE                   00000F10 RG    06         VALUE                          0000020E R     02
SAVE_TODCBASE                     00000004 R     05
SAVE_TODR                         0000000C R     05
SCTRLSTR                          00000621 R     06
SDVHDR                            00000665 R     06
SDVHDRLEN                       = 0000009E
SEPARATOR                         00000206 R     02
SETCMD                            00000056 R     02
SETOUTPUT                         000000AC R     02
SETSPEC                           00000080 R     02
SETSTARTUP                        000000C0 R     02
SHOCMD                            000000D4 R     02
SHOSWITCH                         000000D6 R     02
SHOWONE                           00000182 R     02
SS$_NORMAL                      = 00000001
STATE1                            00000000 RG    02
SYMBOL                            00000096 R     02
SYS$ASCTOID                       00000F37 RG    06
SYS$FAO                           ********    X  06
SYS$FILESCAN                      00000F37 RG    06
TPA$B_CHAR                      = 00000018
TPA$K_COUNT0                    = 00000008
TPA$K_LENGTH0                   = 00000024
TPA$L_COUNT                     = 00000000
TPA$L_NUMBER                    = 0000001C
TPA$L_OPTIONS                   = 00000004
TPA$L_PARAM                     = 00000020
TPA$L_STRINGCNT                 = 00000008
```

```
                                        +----------------+
                                        ! Psect synopsis !
                                        +----------------+

PSECT name                       Allocation          PSECT No.  Attributes
----------                       ----------          ---------  ----------
.  ABS  .                        00000000  (     0.)  00 (  0.)  NOPIC   USR   CON   ABS   LCL NOSHR NOEXE NORD  NOWRT NOVEC BYTE
$ABS$                            00000000  (     0.)  01 (  1.)  NOPIC   USR   CON   ABS   LCL NOSHR   EXE   RD    WRT NOVEC BYTE
_LIB$STATE$                      00000218  (   536.)  02 (  2.)    PIC   USR   CON   REL   LCL   SHR   EXE   RD  NOWRT NOVEC BYTE
_LIB$KEY0$                       00000044  (    68.)  03 (  3.)    PIC   USR   CON   REL   LCL   SHR   EXE   RD  NOWRT NOVEC WORD
_LIB$KEY1$                       000000D7  (   215.)  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                        00000F50  (  3920.)  06 (  6.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD    WRT NOVEC LONG

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

Phase                       Page faults    CPU Time       Elapsed Time
-----                       -----------    --------       ------------
Initialization                      35    00:00:00.08    00:00:00.68
Command processing                 142    00:00:00.83    00:00:04.79
Pass 1                             941    00:00:56.21    00:01:48.03
Symbol table sort                    7    00:00:02.39    00:00:04.16
Pass 2                             461    00:00:11.15    00:00:22.36
Symbol table output                 38    00:00:00.32    00:00:00.61
Psect synopsis output                3    00:00:00.04    00:00:00.04
Cross-reference output               0    00:00:00.00    00:00:00.00
Assembler run totals              1629    00:01:11.03    00:02:20.67

The working set limit was 2000 pages.
220699 bytes (432 pages) of virtual memory were used to buffer the intermediate code.
There were 90 pages of symbol table space allocated to hold 1461 non-local and 74 local symbols.
4534 source lines were read in Pass 1, producing 40 object records in Pass 2.
167 pages of virtual memory were used to define 160 macros.

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

Macro library name                          Macros defined
------------------                          --------------
_$255$DUA28:[BOOTS.OBJ]BOOTS.MLB;1                1
_$255$DUA28:[SYS.OBJ]LIB.MLB;1                    5
_$255$DUA28:[SYSLIB]STARLET.MLB;2                14
TOTALS (all libraries)                           20

1730 GETS were required to define 20 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:SYSBOOCMD/OBJ=OBJ$:SYSBOOCMD MASD$:[EMULAT.SRC]MISSING/UPDATE=(MASD$:[EMULAT.ENH]MISSING)+MASD$:[BOOTS.SRC]SYSBOOCMD/
```

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

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

SYSBOOCMD
LIS

SHOWADAP
LIS

STANDCONF
LIS

SHODEV
LIS

STAPUTERR
LIS

STALOCK
LIS

STASGNMSG
LIS

STACONFIG
LIS

STASYSGEN
LIS

SYSBOOT
LIS

STARDDRIV
LIS