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

FILEID**SHOWADAP

```
  SSSSSSSS  HH      HH    000000    WW      WW    AAAAAA    DDDDDDDD      AAAAAA    PPPPPPPP
  SSSSSSSS  HH      HH    000000    WW      WW    AAAAAA    DDDDDDDD      AAAAAA    PPPPPPPP
SS          HH      HH  00      00  WW      WW  AA      AA  DD      DD  AA      AA  PP      PP
SS          HH      HH  00      00  WW      WW  AA      AA  DD      DD  AA      AA  PP      PP
SS          HH      HH  00      00  WW      WW  AA      AA  DD      DD  AA      AA  PP      PP
SS          HH      HH  00      00  WW      WW  AA      AA  DD      DD  AA      AA  PP      PP
  SSSSSS    HHHHHHHHHH  00      00  WW      WW  AA      AA  DD      DD  AA      AA  PPPPPPPP
  SSSSSS    HHHHHHHHHH  00      00  WW      WW  AA      AA  DD      DD  AA      AA  PPPPPPPP
        SS  HH      HH  00      00  WW  WW  WW  AAAAAAAAAA  DD      DD  AAAAAAAAAA  PP
        SS  HH      HH  00      00  WW  WW  WW  AAAAAAAAAA  DD      DD  AAAAAAAAAA  PP
        SS  HH      HH  00      00  WWWW  WWWW  AA      AA  DD      DD  AA      AA  PP
        SS  HH      HH  00      00  WWWW  WWWW  AA      AA  DD      DD  AA      AA  PP
SSSSSSSS    HH      HH    000000    WW      WW  AA      AA  DDDDDDDD    AA      AA  PP
SSSSSSSS    HH      HH    000000    WW      WW  AA      AA  DDDDDDDD    AA      AA  PP

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

Table of contents

```
0000     1          .TITLE  SHOWADAP - SHOW ADAPTER and GENERIC ADAPTER NAMES
0000     2          .IDENT  'V04-000'
0000     3
0000     4 ;
0000     5 ;*****************************************************************************
0000     6 ;*                                                                           *
0000     7 ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
0000     8 ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
0000     9 ;*  ALL RIGHTS RESERVED.                                                     *
0000    10 ;*                                                                           *
0000    11 ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED    *
0000    12 ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0000    13 ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER    *
0000    14 ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY    *
0000    15 ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY    *
0000    16 ;*  TRANSFERRED.                                                             *
0000    17 ;*                                                                           *
0000    18 ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE    *
0000    19 ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0000    20 ;*  CORPORATION.                                                             *
0000    21 ;*                                                                           *
0000    22 ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS    *
0000    23 ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
0000    24 ;*                                                                           *
0000    25 ;*                                                                           *
0000    26 ;*****************************************************************************
0000    27 ;
0000    28
0000    29 ;++
0000    30 ; FACILITY:     SYSGEN
0000    31 ;
0000    32 ; ABSTRACT:
0000    33 ;       THIS ROUTINE PROVIDES GENERIC ADAPTER NAMES FOR SYSGEN
0000    34 ;
0000    35 ; ENVIRONMENT:  USER, EXEC MODE
0000    36 ;
0000    37 ; AUTHOR:  Jake VanNoy          CREATION DATE: 30-APR-1981
0000    38 ;
0000    39 ; MODIFICATION HISTORY:
0000    40 ;
0000    41 ;       V03-006 TCM0002         Trudy C. Matthews       25-Jul-1984
0000    42 ;               Change venus cpu model number from 11/790 to 8600.
0000    43 ;
0000    44 ;       V03-005 KPL0100         Peter Lieberwirth       10-Feb-1984
0000    45 ;               Change CONFREG to CONFREGL, a longword-array of adapter
0000    46 ;               types.
0000    47 ;
0000    48 ;               Permit undefined adapter types without signalling an error,
0000    49 ;               foreign adapters are anticipated on the BI.
0000    50 ;
0000    51 ;       V03-004 WHM0002         Bill Matthews           01-Feb-1984
0000    52 ;               No adapter default is now -1 not 0. BOO$RESET_ADAP was modified.
0000    53 ;
0000    54 ;       V03-003 WHM0001         Bill Matthews           31-Jan-1984
0000    55 ;               Add support for mixed 16k and 64k memory display.
0000    56 ;
0000    57 ;       V03-002 KDM0084         Kathleen D. Morse       23-Sep-1983
```

```
0000     58 ;                  Add MicroVAX I and MicroVAX II to CPUDISP.
0000     59 ;
0000     60 ;       V03-001 TCM0001          Trudy C. Matthews       03-Aug-1983
0000     61 ;                Re-write code that displays CPU model number to use the
0000     62 ;                new format CPUDISP macro.  Add support for 785 model display.
0000     63 ;
0000     64 ;       V02-007 JLV0139          Jake VanNoy             2-Jan-1981
0000     65 ;                Remove revision number code because of problems
0000     66 ;                with formatting of data, will wait for GETSYI
0000     67 ;                calls to do this. Replace calls to LIB$PUT_OUTPUT
0000     68 ;                with calls to RIO$OUTPUT_LINE.
0000     69 ;
0000     70 ;       V02-006 JLV0118          Jake VanNoy             9-Nov-1981
0000     71 ;                Added code to report errors in BOO$ADAPTER_NAME.
0000     72 ;
0000     73 ;       V02-005 JLV0091          Jake VanNoy             22-Sep-1981
0000     74 ;                Expand number of bytes in boo$ab_count_blk.
0000     75 ;                Also removed MPM's and DR's from "generic"
0000     76 ;                classification.
0000     77 ;
0000     78 ;       V02-004 JLV0086          Jake VanNoy             15-Sep-1981
0000     79 ;                Added 64 bit memory support and changed the way lookups
0000     80 ;                are done.
0000     81 ;
0000     82 ;       V02-003 JLV0041          Jake VanNoy             13-Jul-1981
0000     83 ;                Added G^ to LIB$ call.
0000     84 ;
0000     85 ;       V02-002 JLV0035          Jake VanNoy             6-Jul-1981
0000     86 ;                Added CI definition.
0000     87 ;
```

```
                                                G 4
                     0000     89          .SBTTL  DECLARATIONS
                     0000     90 ;
                     0000     91 ; INCLUDE FILES:
                     0000     92 ;
                     0000     93
                     0000     94 ;
                     0000     95 ; CONSTANTS:
                     0000     96 ;
                     0000     97
0000000A             0000     98 LF                = 10
00000014             0000     99 boo$c_count_blk = 20
                     0000    100
                     0000    101 ;
                     0000    102 ; MACROS:
                     0000    103 ;
                     0000    104
                     0000    105          $NDTDEF
                     0000    106          $PRDEF
                     0000    107          $SYSGMSGDEF
                     0000    108          $TPADEF
                     0000    109
                     0000    110 $VIELD   BOO,0,<-
                     0000    111          <GENERIC    ,,M>,-
                     0000    112          >
00000000             0000    113 L_CONSTANT       = 0             ; CONSTANT = NDT adapter type code
00000004             0000    114 W_FLAGS          = 4             ; only flag is GENERIC, means memory or DR32
00000006             0000    115 W_INDEX          = 6             ; INDEX is sysgen-specific, used to associate
                     0000    116                                  ; occurance counts with adapters,
                     0000    117                                  ; MBA=0, UBA=1, CI=2
00000008             0000    118 L_NAME           = 8             ; Offset to ASCID string containing adapter name
                     0000    119
                     0000    120 .Macro   Adapter constant,string=<>,Flags = 0,index
                     0000    121
                     0000    122          .PSECT  PAGED_DATA_2    rd,wrt,noexe,quad
                     0000    123          $$$ = .
                     0000    124          .LONG   CONSTANT
                     0000    125          .WORD   FLAGS
                     0000    126          .WORD   INDEX
                     0000    127          .ASCID  /STRING/
                     0000    128
                     0000    129          .PSECT  PAGED_DATA      rd,wrt,noexe,quad
                     0000    130          .LONG   $$$
                     0000    131
                     0000    132 .Endm    adapter
                     0000    133
                     0000    134 ;
                     0000    135 ; OWN STORAGE:
                     0000    136 ;
                     0000    137
                00000000     138 .PSECT  PAGED_DATA      rd,wrt,noexe,quad
                     0000    139
00000040             0000    140 Maxnexus = 64                                    ; maximum is 4 BIs
                     0000    141
00000100             0000    142 Boo$ab_confreg_blk:     .blkl   maxnexus        ;
00000140             0100    143 Boo$ab_adap_idx:        .blkb   maxnexus        ;
00000240             0140    144 Boo$ab_adap_txt:        .blkl   maxnexus        ;
                     0240    145
```

```
                                00000254  0240    146 Boo$ab_Count_blk:       .blkb   boo$c_count_blk ; This many adapter types need
                                          0254    147                                                  ; occurance counts.
                                          0254    148
79 54 20 55 50 43 0000025C'010E0000'      0254    149 C780:                   .ascid  @CPU Type: 11/780@
      30 38 37 2F 31 31 20 3A 65 70       0262
79 54 20 55 50 43 00000274'010E0000'      026C    150 C750:                   .ascid  @CPU Type: 11/750@
      30 35 37 2F 31 31 20 3A 65 70       027A
79 54 20 55 50 43 0000028C'010E0000'      0284    151 C730:                   .ascid  @CPU Type: 11/730@
      30 33 37 2F 31 31 20 3A 65 70       0292
79 54 20 55 50 43 000002A4'010E0000'      029C    152 C790:                   .ascid  @CPU Type:   8600@
      30 30 36 38 20 20 20 3A 65 70       02AA
79 54 20 55 50 43 000002BC'010E0000'      02B4    153 C785:                   .ascid  @CPU Type: 11/785@
      35 38 37 2F 31 31 20 3A 65 70       02C2
79 54 20 55 50 43 000002D4'010E0000'      02CC    154 CUV1:                   .ascid  @CPU Type: MicroVAX I@
58 41 56 6F 72 63 69 4D 20 3A 65 70       02DA
                              49 20       02E6
79 54 20 55 50 43 000002F0'010E0000'      02E8    155 CUV2:                   .ascid  @CPU Type: MicroVAX II@
58 41 56 6F 72 63 69 4D 20 3A 65 70       02F6
                              49 49 20    0302
                                          0305    156
                                00000000  0305    157 str_size:               .long   0
                                00000000  0309    158 str_addr:               .long   0
                                0000032D  030D    159 str_start:              .blkb   32
                                          032D    160
3C 33 21 20 20 20 00000335'010E0000'      032D    161 ctr_generic:            .ascid  /   !3<!UL!>!_   !AS!UB/
41 21 20 20 20 5F 21 3E 21 4C 55 21       033B
                           42 55 21 53    0347
3C 33 21 20 20 20 00000353'010E0000'      034B    162 ctr_memory:             .ascid  /   !3<!UL!>!_   !AS/
41 21 20 20 20 5F 21 3E 21 4C 55 21       0359
                                 53       0365
                                          0366    163
75 78 65 4E 20 0A 0000036E'010E0000'      0366    164 header:                 .ascid  <LF>/ Nexus        Generic Name or Description/
63 69 72 65 6E 65 47 20 20 20 09 73       0374
73 65 44 20 72 6F 20 65 6D 61 4E 20       0380
              6E 6F 69 74 70 69 72 63     038C
                    0000039C'010E0000'    0394    165 trailer:                .ascid  //
                                          039C    166
77 6F 6E 6B 6E 55 000003A4'010E0000'      039C    167 unk_adap:               .ascid  /Unknown/
                                    6E    03AA
                                          03AB    168
                                00000000  03AB    169 called_flag:            .long   0
                                          03AF    170
```

```
                     03AF    172 ; ADAPTER TABLE
                     03AF    173
                     03AF    174 Boo$al_adap_table:
                     03AF    175         adapter NDT$_MEM4NI,    <4K memory, non-interleaved>
                     03B3    176         adapter NDT$_MEM4I,     <4K memory, interleaved>
                     03B7    177         adapter NDT$_MEM16NI,   <16K memory, non-interleaved>
                     03BB    178         adapter NDT$_MEM16I,    <16K memory, interleaved>
                     03BF    179         adapter NDT$_MEM1664NI, <Mixed 16K and 64K memory, non-interleaved>
                     03C3    180         adapter NDT$_MB,        <MB>,   Boo$m_generic,  0
                     03C7    181         adapter NDT$_UB0,       <UB>,   Boo$m_generic,  1
                     03CB    182         adapter NDT$_UB1,       <UB>,   Boo$m_generic,  1
                     03CF    183         adapter NDT$_UB2,       <UB>,   Boo$m_generic,  1
                     03D3    184         adapter NDT$_UB3,       <UB>,   Boo$m_generic,  1
                     03D7    185         adapter NDT$_CI,        <CI>,   Boo$m_generic,  2
                     03DB    186         adapter NDT$_MPM0,      <MPM0>
                     03DF    187         adapter NDT$_MPM1,      <MPM1>
                     03E3    188         adapter NDT$_MPM2,      <MPM2>
                     03E7    189         adapter NDT$_MPM3,      <MPM3>
                     03EB    190         adapter NDT$_DR32,      <DR32>
                     03EF    191         adapter NDT$_MEM64NIL,  <64K non-interleaved memory, lower controller>
                     03F3    192         adapter NDT$_MEM64EIL,  <64K externally interleaved memory, lower controller
                     03F7    193         adapter NDT$_MEM64NIU,  <64K non-interleaved memory, upper controller>
                     03FB    194         adapter NDT$_MEM64EIU,  <64K externally interleaved memory, upper controller
                     03FF    195         adapter NDT$_MEM64I,    <64K internally interleaved memory>
                     0403    196
00000000             0403    197         .long   0       ; End of table
                     0407    198 ;
                     0407    199 ; Note: The maximum index for generic adapters above must less than or equal to
                     0407    200 ; the constant boo$c_count_blk.
                     0407    201 ;
                     0407    202
```

```
0407   204          .Sbttl  BOO$SHOWADAP - SHOW/ADAPTERS routine
0407   205 ;++
0407   206 ; FUNCTIONAL DESCRIPTION:
0407   207 ;
0407   208 ;       Scan CONFREG and output text associated with each adapter.
0407   209 ;       Text is to match exactly what a user would type for the
0407   210 ;       /ADAPTER qualifier or the AUTOCONFIGURE "adapter" command.
0407   211 ;
0407   212 ; CALLING SEQUENCE:
0407   213 ;
0407   214 ;       Called from TPARSE as an action routine
0407   215 ;
0407   216 ; INPUT PARAMETERS:
0407   217 ;
0407   218 ;       None.
0407   219 ;
0407   220 ; IMPLICIT INPUTS:
0407   221 ;
0407   222 ;       CONFREG.
0407   223 ;
0407   224 ; OUTPUT PARAMETERS:
0407   225 ;
0407   226 ;       R0      Completion code
0407   227 ;
0407   228 ; IMPLICIT OUTPUTS:
0407   229 ;
0407   230 ;       NONE
0407   231 ;
0407   232 ;--
0407   233
0407   234 ; Register usage
0407   235 ;
0407   236
0407   237
0407   238
00000000   239          .PSECT  PAGED_CODE      rd,nowrt,exe,long
0000   240
07FC 0000   241 .Entry  BOO$SHOW_ADAPTER, ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10>
0002   242
00B9'CF   00 FB 0002   243          calls   #0,w^Show_Cpu           ; Show CPU specific data
        09 50 E8 0007   244          blbs    R0,5$                   ; Branch if OK
        50 DD 000A      245          pushl   R0                      ; Push error code
00000000'GF   01 FB 000C   246          calls   #1,G^Lib$Signal         ; Signal and continue
0013   247
00000366'EF   DF 0013   248 5$:     pushal  header                  ; Set up header
000000A4'EF   01 FB 0019   249          calls   #1,Boo$Output_Desc      ; Output header
0020   250
000001A1'EF   00 FB 0020   251 10$:     calls   #0,Get_All_Adap         ; Fill in Adap_txt and Adap_idx
        03 50 E8 0027   252          blbs    R0,15$                  ; Branch if ok
         0076 31 002A   253          brw     80$                     ; Exit
002D   254
        52 D4 002D   255 15$:    clrl    R2                      ; initialize index
53   00000000'EF   D0 002F   256          movl    exe$gl_numnexus,R3      ; Set count (User readable location)
    54   0100'CF   9E 0036   257          movab   w^boo$ab_adap_idx,R4    ; Adapter index table
    55   0140'CF   9E 003B   258          movab   w^boo$ab_adap_txt,R5    ; Adapter text table
0040   259
   57   6542   D0 0040   260 20$:    movl    (R5)[R2],R7             ; Get address of text descriptor
```

```
                  49   13  0044    261          beql    60$                         ; Branch if zero
              56  6442   98  0046    262          cvtbl   (R4)[R2],R6                 ; Get index count
                  1B   19  0048    263          blss    30$                         ; Branch if non-generic
                       004C    264
                       004C    265          $FAO_S  Ctrstr = w^ctr_generic,- ; Format string
                       004C    266                  Outbuf = w^rio$ab_outbuf,-
                       004C    267                  Outlen = w^rio$gw_outlen,-
                       004C    268                  P1     = R2,-               ; Nexus number
                       004C    269                  P2     = R7,-               ; Adapter text prefix address
                       004C    270                  P3     = R6                 ; Count
                  17   11  0065    271          brb     40$
                       0067    272
                       0067    273 30$:     $FAO_S  Ctrstr = w^ctr_memory,- ; Format string
                       0067    274                  Outbuf = w^rio$ab_outbuf,-
                       0067    275                  Outlen = w^rio$gw_outlen,-
                       0067    276                  P1     = R2,-               ; Nexus number
                       0067    277                  P2     = R7                 ; Adapter text address
                       007E    278
               0B 50   E8  007E    279 40$:     blbs    R0,50$                      ; branch if no error from fao
                       0081    280
                  50   DD  0081    281 45$:     pushl   R0                          ; Set error code
    00000000'GF   01   FB  0083    282          calls   #1,G^Lib$Signal             ; Signal Error
                  03   11  008A    283          brb     60$                         ; goto end of loop
                       008C    284
              FF71'    30  008C    285 50$:     bsbw    rio$output_line             ; Output line
           AD 52   53   F2  008F    286 60$:     aoblss  R3,R2,20$                   ; Loop
                       0093    287
         00000394'EF   DF  0093    288          pushal  trailer                     ; Set up blank line trailer
    000000A4'EF   01   FB  0099    289          calls   #1,BOO$Output_Desc          ; Output header
                       00A0    290
    50   01   D0  00A0    291          movl    #1,R0
                  04   00A3    292 80$:     ret
```

```
                     00A4   294 .sbttl  Boo$Output_Desc - Output a line
                     00A4   295
                     00A4   296 ;+
                     00A4   297 ; BOO$OUTPUT_DESC
                     00A4   298 ;
                     00A4   299 ; Calling sequence
                     00A4   300 ;
                     00A4   301 ; CALLS #1,BOO$OUTPUT_DESC
                     00A4   302 ;
                     00A4   303 ; Input:
                     00A4   304 ;       4(AP) = Address of descriptor of line to output
                     00A4   305 ;
                     00A4   306 ; Output:
                     00A4   307 ;       RIO$OUTPUT_LINE is called.
                     00A4   308 ;
                     00A4   309 ;-
                     00A4   310
                003C 00A4   311 .Entry  BOO$OUTPUT_DESC,^M<R2,R3,R4,R5>
                     00A6   312
      50   04 BC    7D  00A6   313         movq    @4(AP),R0               ; Fetch Descriptor into R0,R1
    0000'CF   50    B0  00AA   314         movw    R0,W^Rio$gw_outlen      ; Set Length
0000'CF   61   50   28  00AF   315         movc3   R0,(R1),W^Rio$ab_buffer ; Move into buffer to be written
           FF48'    30  00B5   316         bsbw    Rio$Output_Line         ; Output line
                04  00B8   317         ret
                     00B9   318
```

```
                                00B9   320 .Sbttl  Show_CPU - Show CPU specfic data
                                00B9   321
                           003C 00B9   322 .Entry  Show_CPU, ^M<R2,R3,R4,R5>
                                00BB   323
                 00000394'EF    DF   00BB   324         pushal  trailer                 ; Set up blank line
                  DF AF    01    FB   00C1   325         calls   #1,Boo$Output_Desc      ; Output
                        6D 50    E9   00C5   326         blbc    R0,End_show_cpu         ; branch on error
                                      00C8   327
                                      00C8   328 ;
                                      00C8   329 ; This cumbersome way of picking up the CPU model number display is used so
                                      00C8   330 ; that all CPU-dependent code is flagged by the use of the CPUDISP macro.
                                      00C8   331 ;
                                      00C8   332         .list   meb
                                      00C8   333         cpudisp -
                                      00C8   334                 <<780,c780_model>,-
                                      00C8   335                  <750,c750_model>,-
                                      00C8   336                  <730,c730_model>,-
                                      00C8   337                  <790,c790_model>,-
                                      00C8   338                  <UV1,cUV1_model>,-
                                      00C8   339                  <UV2,cUV2_model>,-
                                      00C8   340                  <785,c785_model>>
          01    00000000'GF    91   00C8                CMPB    G^EXE$GB_CPUTYPE, -
                         0B    12   00CF                BNEQ    30014$
       03 00000000'GF    17   E1   00D1                BBC     #23,G^EXE$GB_CPUDATA,30014$
                      004E    31   00D9                BRW     c785_model
                                      00DC        30014$:
    08  01    00000000'GF    8F   00DC                CASEB   G^EXE$GB_CPUTYPE,#$$BASE,#$$LIMIT
                                      00E4        30015$:
                      0016' 00E4                .SIGNED_WORD    c780_model-30015$
                      001E' 00E6                .SIGNED_WORD    c750_model-30015$
                      0026' 00E8                .SIGNED_WORD    c730_model-30015$
                      002E' 00EA                .SIGNED_WORD    c790_model-30015$
                      0012  00EC                .IIF    EQ  $$GENSW,    .WORD   2*<$$LIMIT+1>
                      0012  00EE                .IIF    EQ  $$GENSW,    .WORD   2*<$$LIMIT+1>
                      0036' 00F0                .SIGNED_WORD    cUV1_model-30015$
                      003E' 00F2                .SIGNED_WORD    cUV2_model-30015$
                      0046' 00F4                .SIGNED_WORD    c785_model-30015$
                      FEFF  00F6                        .WORD   ^XFEFF
                      0004' 00F8                        .IIF IDN <FATAL>,<FATAL> , .WORD        BUG$_UNSUPRTCPU!4
                                00FA   341         .nlist  meb
                                00FA   342 c780_model:
                 00000254'EF    DF   00FA   343         pushal  c780
                         2E    11   0100   344         brb     output_model
                                0102   345 c750_model:
                 0000026C'EF    DF   0102   346         pushal  c750
                         26    11   0108   347         brb     output_model
                                010A   348 c730_model:
                 00000284'EF    DF   010A   349         pushal  c730
                         1E    11   0110   350         brb     output_model
                                0112   351 c790_model:
                 0000029C'EF    DF   0112   352         pushal  c790
                         16    11   0118   353         brb     output_model
                                011A   354 cUV1_model:
                 000002CC'EF    DF   011A   355         pushal  cUV1
                         0E    11   0120   356         brb     output_model
                                0122   357 cUV2_model:
                 000002E8'EF    DF   0122   358         pushal  cUV2
```

```
                 06   11  0128   359          brb     output_model
                          012A   360 c785_model:
 000002B4'EF     DF       012A   361          pushal  c785
                          0130   362
                          0130   363 output_model:
 FF6F CF         01   FB  0130   364          calls   #1,Boo$Output_Desc      ; output
                          0135   365 End_show_cpu:
                      04  0135   366          ret
                          0136   367
                          0136   368
```

```
                     0136   370 .sbttl - BOO$ADAPTER_NAME - generic adapter name parsing
                     0136   371
                     0136   372 ;++
                     0136   373 ; FUNCTIONAL DESCRIPTION:
                     0136   374 ;
                     0136   375 ;       Scan CONFREG and output text associated with each adapter.
                     0136   376 ;       Text is to match exactly what a user would type for the
                     0136   377 ;       /ADAPTER qualifier or the AUTOCONFIGURE "adapter" command.
                     0136   378 ;
                     0136   379 ; CALLING SEQUENCE:
                     0136   380 ;
                     0136   381 ;       Called from TPARSE as an action routine
                     0136   382 ;
                     0136   383 ; INPUT PARAMETERS:
                     0136   384 ;
                     0136   385 ;       TPA$L_NUMBER(AP) - Number in generic specifier (e.g. 0 if "UBO")
                     0136   386 ;
                     0136   387 ; IMPLICIT INPUTS:
                     0136   388 ;
                     0136   389 ;       CONFREG.
                     0136   390 ;       str_size and str_start - Set up by previous TPARSE routines as
                     0136   391 ;               length and character string in generic adapter type.
                     0136   392 ;
                     0136   393 ; OUTPUT PARAMETERS:
                     0136   394 ;
                     0136   395 ;       R0      Completion code
                     0136   396 ;
                     0136   397 ; IMPLICIT OUTPUTS:
                     0136   398 ;
                     0136   399 ;       TPA$L_NUMBER(AP) is set to appropriate nexus number.
                     0136   400 ;
                     0136   401 ;--
                     0136   402
                     0136   403 ; Register usage
                     0136   404 ;
                     0136   405 ; R4 - Base address of adap_idx array
                     0136   406 ; R5 - Base address of adap_txt array
                     0136   407 ; R6 - Index through loop
                     0136   408 ; R7 - addr(adapter text)
                     0136   409 ; R8 - occurance of this type adapter
                     0136   410 ; R9 - Size in bytes of input adapter string
                     0136   411 ; R10 - Address of input adapter string
                     0136   412 ; R11 - Size in bytes of array (16 for 1 SBI, 32 for 2)
                     0136   413 ;
                     0136   414
                 0FFC 0136   415 .Entry Boo$Adapter_Name, ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
                     0138   416
    59   0305'CF   D0 0138   417         movl    w^str_size,R9           ; Size of adapter name string
           4A      13 013D   418         beql    35$                     ; Branch to invalid name error if zero
    5A   030D'CF   9E 013F   419 10$:    movab   w^str_start,R10         ; Address of adapter name
                     0144   420
000001A1'EF    00   FB 0144   421         calls   #0,Get_All_Adap         ; Fill in Adap_txt and Adap_idx
         46 50     E9 014B   422         blbc    R0,40$                  ; Branch if error
                     014E   423
           56      D4 014E   424         clrl    R6                      ; initialize index
5B  00000000'EF    D0 0150   425         movl    exe$gl_numnexus,R11     ; Set count (User readable location)
    54   0100'CF   9E 0157   426         movab   w^boo$ab_adap_idx,R4    ; Adapter index table
```

```
                5   0140'CF    9E   015C    427          movab   w^boo$ab_adap_txt,R5     ; Adapter text table
                                0161    428
               57     6546     D0   0161    429 20$:     movl    (R5)[R6],R7             ; Get address of text descriptor
                       1E      13   0165    430          beql    30$                     ; Branch if zero
               58     6446     98   0167    431          cvtbl   (R4)[R6],R8             ; Get index count
                       18      19   016B    432          blss    30$                     ; Branch if non-generic
                                016D    433
               59     67       B1   016D    434          cmpw    (R7),R9                 ; Compare lengths
                       13      12   0170    435          bneq    30$                     ; Branch if not equal
                                0172    436
            1C AC      58       D1   0172    437          cmpl    R8,Tpa$l_number(AP)     ; Check occurance number
                       0D      12   0176    438          bneq    30$                     ; Branch if not equal
                                0178    439
         04 B7     6A    59       29   0178    440          cmpc3   R9,(R10),a4(R7)         ; Comapre actual strings
                       06      12   017D    441          bneq    30$                     ; Branch if not equal
                                017F    442
            1C AC      56       D0   017F    443          movl    R6,Tpa$l_number(AP)     ; Set adapter number
                       18      11   0183    444          brb     50$
                                0185    445
            D8 56      5B       F2   0185    446 30$:     aoblss  R11,R6,20$              ; Loop R11 times, incrementing R6
                                0189    447
      50    007C80BA   8F       D0   0189    448 35$:     movl    #sysg$_invadap,R0       ; Set error
            1C AC      01       CE   0190    449          mnegl   #1,Tpa$l_number(AP)     ; Set adapter number
                                0194    450
                       50       DD   0194    451 40$:     pushl   R0                      ; push error status
      00000000'GF      01       FB   0196    452          calls   #1,G^Lib$Signal         ; Signal error
                                019D    453 50$:
               50      01       D0   019D    454          movl    #1,R0                   ; Set success
                       04      01A0    455          ret                             ; return
                               01A1    456
```

```
                                                D 5
                                01A1    458 .sbttl  Get_All_Adap - Get all adapters into readable format
                                01A1    459
                          03FC  01A1    460 .Entry  Get_all_adap,^M<R2,R3,R4,R5,R6,R7,R8,R9>
                                01A3    461
  03 000003AB'EF        E9      01A3    462         blbc    called_flag,10$         ; Branch if first call here
             008B       31      01AA    463         brw     110$                    ; Exit, no work necessary
                                01AD    464
                                01AD    465 10$:    $CMEXEC_S w^Read_Confreg        ; Read confreg into user readable area
           01 50        E8      01BA    466         blbs    R0,15$                  ; Branch on success
                             04 01BD    467         ret                             ; return with error
                                01BE    468
   53   0240'CF        9E      01BE    469 15$:    movab   w^boo$ab_count_blk,R3   ; Count block
        54    14        9A      01C3    470         movzbl  #boo$c_count_blk,R4     ; size
              83        94      01C6    471 20$:    clrb    (R3)+                   ; zero it out
           FB 54        F5      01C8    472         sobgtr  R4,20$                  ; loop
                                01CB    473
              50        D4      01CB    474         clrl    R0                      ; initialize index
51   00000000'EF        D0      01CD    475         movl    exe$gl_numnexus,R1      ; Set count (User readable location)
   52    0000'CF        9E      01D4    476         movab   w^boo$ab_confreg_blk,R2 ; Set address of output block
   53    0240'CF        9E      01D9    477         movab   w^boo$ab_count_blk,R3   ; Count block
   54    0100'CF        9E      01DE    478         movab   w^boo$ab_adap_idx,R4    ; Adapter index table
   55    0140'CF        9E      01E3    479         movab   w^boo$ab_adap_txt,R5    ; Adapter text table
                                01E8    480
        56    82        D0      01E8    481 40$:    movl    (R2)+,R6                ; Get adapter ($NDTDEF) type
              3A        13      01EB    482         beql    90$                     ; Branch if nothing on nexus
                                01ED    483
57   000003AF'EF        9E      01ED    484         movab   boo$al_adap_table,R7    ; Set address of descriptor array
                                01F4    485 ;
                                01F4    486 ; arrays adap_idx and adap_txt is now filled in for this nexus.
                                01F4    487 ; adap_txt will always be non-zero for a nexus with a known adapter
                                01F4    488 ; on it. adap_idx will be a positive integer (0 through n) indicating
                                01F4    489 ; its occurance count, or negative indicating that it is a memory
                                01F4    490 ; adapter with no generic name.
                                01F4    491 ;
                                01F4    492
        58    87        D0      01F4    493 50$:    movl    (R7)+,R8                ; Get next block (defined by Adapter)
              21        13      01F7    494         beql    80$                     ; Adapter type not found
        68    56        D1      01F9    495         cmpl    R6,l_constant(R8)       ; Adapter type match ?
              F6        12      01FC    496         bneq    50$                     ; Loop if not
                                01FE    497
      6440    01        8E      01FE    498         mnegb   #1,(R4)[R0]             ; Assume not generic
              00        E1      0202    499         bbc     #boo$v_generic,-
         0C 04 A8               0204    500                 w_flags(R8),60$     ; Branch if not
     59    06 A8        3C      0207    501         movzwl  w_index(R8),R9          ; Set adapter type index
   6440     6349        90      020B    502         movb    (R3)[R9],(R4)[R0]       ; Move occurance count to adap_idx array
            6349        96      0210    503         incb    (R3)[R9]                  Increment occurance count
                                0213    504
  6540    08 A8         9E      0213    505 60$:    movab   l_name(R8),(R5)[R0]     ; Set text descriptor address
              13        11      0218    506         brb     100$                    ; end of loop
                                021A    507
                                021A    508 80$:                                    ; Unrecognized adapter type
6540 0000039C'EF        9E      021A    509         movab   unk_adap,(R5)[R0]       ; Set "Unknown" text descriptor address
            6440        94      0222    510         clrb    (R4)[R0]                ; No adapter count
              06        11      0225    511         brb     100$
                                0227    512
            6440        94      0227    513 90$:    clrb    (R4)[R0]                ; No adapter count
            6540        D4      022A    514         clrl    (R5)[R0]                ; No adapter text
```

```
                                022D    515
                     B7 50  51  F2  022D    516 100$:   aoblss  R1,R0,40$               ; Loop R1 times
                                    0231    517
000003AB'                EF  01  CE  0231    518         mnegl   #1,called_flag         ; Set flag indicating routine called
                            50  01  D0  0238    519 110$:   movl    #1,R0              ; Set success
                                04  023B    520         ret
                                    023C    521
```

```
                                023C    523 .sbttl  Read_Confreg - Read adapter configuration array
                                023C    524
                          001C  023C    525 .Entry  READ_CONFREG, ^M<R2,R3,R4>
                                023E    526
                                023E    527 ;
                                023E    528 ; EXEC mode routine to read CONFREG into user-mode readable area
                                023E    529 ;
                                023E    530
52   00000000'EF    D0          023E    531         movl    exe$gl_confregl,R2      ; Set address of confreg
54   00000000'EF    D0          0245    532         movl    exe$gl_numnexus,R4      ; Set count
     53    0000'CF  9E          024C    533         movab   w^boo$ab_confreg_blk,R3 ; Set address of output block
                                0251    534
          83    82  D0          0251    535 10$:    movl    (R2)+,(R3)+             ; 4 bytes (1 CONFREGL entry) at a time
            FA  54  F5          0254    536         sobgtr  r4,10$                  ; Loop until done
                                0257    537
          50    01  D0          0257    538         movl    #1,R0                   ; Set success
                    04          025A    539         ret                             ; Return
                                025B    540
                                025B    541
```

```
                                025B   543 .sbttl  TPARSE adapter name parsing routines
                                025B   544
                           0000 025B   545 .Entry  Boo$Reset_Adap, ^M<>
                                025D   546
                    0305'CF   D4 025D   547         clrl    w^str_size              ; Zero size
0309'CF             030D'CF   9E 0261   548         movab   w^str_start,w^str_addr  ; Set start address
                 1C AC    01   CE 0268   549         mnegl   #1,tpa$l_number(AP)     ; Assume adapter zero
                                026C   550
                           04   026C   551         ret                             ; Return
                                026D   552
                           0000 026D   553 .Entry  Boo$Adap_Letter, ^M<>
                                026F   554
                 51    0309'CF   D0 026F   555         movl    w^str_addr,R1           ; Current string pointer
                   81     18 AC   90 0274   556         movb    tpa$b_char(AP),(R1)+    ; Move and increment address
                 0309'CF    51   D0 0278   557         movl    R1,w^str_addr           ; Set new address
                    0305'CF   D6 027D   558         incl    w^str_size              ; Increment size
                           04   0281   559         ret
                                0282   560
                                0282   561         .END
```

```
$$$                     = 000002A9 R      03
$$BASE                  = 00000001
$$DISPL                 = 0000000A
$$GENSW                 = 00000001
$$HIGH                  = 00000009
$$LIMIT                 = 00000008
$$LOW                   = 00000001
$$MNSW                  = 00000001
$$MXSW                  = 00000001
$$T2                    = 00000005
BIT...                  = 00000001
BOO$AB_ADAP_IDX           00000100 R      02
BOO$AB_ADAP_TXT           00000140 R      02
BOO$AB_CONFREG_BLK        00000000 R      02
BOO$AB_COUNT_BLK          00000240 R      02
BOO$ADAPTER_NAME          00000136 RG     04
BOO$ADAP_LETTER           0000026D RG     04
BOO$AL_ADAP_TABLE         000003AF R      02
BOO$C_COUNT_BLK         = 00000014
BOO$M_GENERIC           = 00000001
BOO$OUTPUT_DESC           000000A4 RG     04
BOO$RESET_ADAP            0000025B RG     04
BOO$SHOW_ADAPTER          00000000 RG     04
BOO$V_GENERIC           = 00000000
BUG$_UNSUPRTCPU           ******** X      04
C730                      00000284 R      02
C730_MODEL                0000010A R      04
C750                      0000026C R      02
C750_MODEL                00000102 R      04
C780                      00000254 R      02
C780_MODEL                000000FA R      04
C785                      000002B4 R      02
C785_MODEL                0000012A R      04
C790                      0000029C R      02
C790_MODEL                00000112 R      04
CALLED_FLAG               000003AB R      02
CTR_GENERIC               0000032D R      02
CTR_MEMORY                0000034B R      02
CUVT                      000002CC R      02
CUV1_MODEL                0000011A R      04
CUV2                      000002E8 R      02
CUV2_MODEL                00000122 R      04
END_SHOW_CPU              00000135 R      04
EXE$GB_CPUDATA            ******** X      04
EXE$GB_CPUTYPE            ******** X      04
EXE$GL_CONFREGL           ******** X      04
EXE$GL_NUMNEXUS           ******** X      04
GET_ALL_ADAP              000001A1 RG     04
HEADER                    00000366 R      02
LF                      = 0000000A
LIB$SIGNAL                ******** X      04
L_CONSTANT              = 00000000
L_NAME                  = 00000008
MAXNEXUS                = 00000040
NDT$_CI                 = 00000038
NDT$_DR32               = 00000030
NDT$_MB                 = 00000020
NDT$_MEM1664NI          = 00000012
NDT$_MEM16I             = 00000011
NDT$_MEM16NI            = 00000010
NDT$_MEM4I              = 00000009
NDT$_MEM4NI             = 00000008
NDT$_MEM64EIL           = 0C000069
NDT$_MEM64EIU           = 00U0006B
NDT$_MEM64I             = 0000006C
NDT$_MEM64NIL           = 00000068
NDT$_MEM64NIU           = 0000006A
NDT$_MPM0               = 00000040
NDT$_MPM1               = 00000041
NDT$_MPM2               = 00000042
NDT$_MPM3               = 00000043
NDT$_UB0                = 00000028
NDT$_UB1                = 00000029
NDT$_UB2                = 0000002A
NDT$_UB3                = 0000002B
OUTPUT_MODEL              00000130 R      04
PR$_SID_TYP730          = 00000003
PR$_SID_TYP750          = 00000002
PR$_SID_TYP780          = 00000001
PR$_SID_TYP785          = 00000009
PR$_SID_TYP790          = 00000004
PR$_SID_TYPMAX          = 00000008
PR$_SID_TYPUV1          = 00000007
PR$_SID_TYPUV2          = 00000008
READ_CONFREG              0000023C RG     04
RIO$AB_BUFFER             ******** X      04
RIO$AB_OUTBUF             ******** X      04
RIO$GW_OUTLEN             ******** X      04
RIO$OUTPUT_LINE           ******** X      04
SHOW_CPU                  000000B9 RG     04
SIZ...                  = 00000001
STR_ADDR                  00000309 R      02
STR_SIZE                  00000305 R      02
STR_START                 0000030D R      02
SYS$CMEXEC                ******** GX     04
SYS$FAO                   ******** X      04
SYSG$_INVADAP           = 007C80BA
TPA$B_CHAR              = 00000018
TPA$L_NUMBER            = 0000001C
TRAILER                   00000394 R      02
UNK_ADAP                  0000039C R      02
W_FLAGS                 = 00000004
W_INDEX                 = 00000006
```

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
PAGED_DATA	00000407 (1031.)	02 (2.)	NOPIC USR CON REL LCL NOSHR NOEXE RD WRT NOVEC QUAD
PAGED_DATA_2	000002DA (730.)	03 (3.)	NOPIC USR CON REL LCL NOSHR NOEXE RD WRT NOVEC QUAD
PAGED_CODE	00000282 (642.)	04 (4.)	NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC LONG

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	33	00:00:00.09	00:00:00.91
Command processing	116	00:00:00.72	00:00:04.00
Pass 1	270	00:00:06.52	00:00:13.07
Symbol table sort	0	00:00:00.41	00:00:00.70
Pass 2	115	00:00:01.88	00:00:03.88
Symbol table output	13	00:00:00.08	00:00:00.10
Psect synopsis output	2	00:00:00.03	00:00:00.03
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	551	00:00:09.73	00:00:22.70

The working set limit was 1500 pages.
56346 bytes (111 pages) of virtual memory were used to buffer the intermediate code.
There were 20 pages of symbol table space allocated to hold 274 non-local and 37 local symbols.
561 source lines were read in Pass 1, producing 45 object records in Pass 2.
27 pages of virtual memory were used to define 23 macros.

Macro library statistics

Macro library name	Macros defined
_255DUA28:[BOOTS.OBJ]BOOTS.MLB;1	0
_255DUA28:[SYS.OBJ]LIB.MLB;1	5
_255DUA28:[SYSLIB]STARLET.MLB;2	9
TOTALS (all libraries)	14

404 GETS were required to define 14 macros.

There were no errors, warnings or information messages.

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

SYSBOOCMD
LIS

SHOWADAP
LIS

STANDCONF
LIS

SHODEV
LIS

STAPUTERR
LIS

STALOCK
LIS

STASGNMSG
LIS

STACONFIG
LIS

STASYSGEN
LIS

SYSBOOT
LIS

STARDBRIV
LIS