

BBBBBBBBBBBB		000000000		000000000		TTTTTTTTTTTT		SSSSSSSSSS
BBBBBBBBBBBB		000000000		000000000		TTTTTTTTTTTT		SSSSSSSSSS
BBBBBBBBBBBB		000000000		000000000		TTTTTTTTTTTT		SSSSSSSSSS
BBB	BBB	000	000	000	000	TTT	SSS	
BBB	BBB	000	000	000	000	TTT	SSS	
BBB	BBB	000	000	000	000	TTT	SSS	
BBB	BBB	000	000	000	000	TTT	SSS	
BBB	BBB	000	000	000	000	TTT	SSS	
BBB	BBB	000	000	000	000	TTT	SSS	
BBB	BBB	000	000	000	000	TTT	SSS	
BBBBBBBBBBBB		000	000	000	000	TTT	SSS	
BBBBBBBBBBBB		000	000	000	000	TTT	SSS	
BBBBBBBBBBBB		000	000	000	000	TTT	SSS	
BBB	BBB	000	000	000	000	TTT	SSS	
BBB	BBB	000	000	000	000	TTT	SSS	
BBB	BBB	000	000	000	000	TTT	SSS	
BBB	BBB	000	000	000	000	TTT	SSS	
BBB	BBB	000	000	000	000	TTT	SSS	
BBB	BBB	000	000	000	000	TTT	SSS	
BBBBBBBBBBBB		000000000		000000000		TTT	SSSSSSSSSS	
BBBBBBBBBBBB		000000000		000000000		TTT	SSSSSSSSSS	
BBBBBBBBBBBB		000000000		000000000		TTT	SSSSSSSSSS	

(1) 127

DISPLAY DEVICE DATABASE TOPOLOGY

```
0000 1 .TITLE SHODEV - DISPLAY DEVICE DATABASE TOPOLOGY
0000 2 .IDENT 'V04-000'
0000 3
0000 4 *****
0000 5
0000 6 COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 7 DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 8 ALL RIGHTS RESERVED.
0000 9
0000 10 THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 11 ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 12 INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 13 COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 14 OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 15 TRANSFERRED.
0000 16
0000 17 THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 18 AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 19 CORPORATION.
0000 20
0000 21 DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 22 SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 23
0000 24 *****
0000 25
0000 26
0000 27 ++
0000 28
0000 29 Facility: System generation and initialization
0000 30
0000 31 Abstract: This module contains routines to show the device database
0000 32 topology.
0000 33
0000 34 Environment:
0000 35
0000 36 Author: Len Kawell, Creation date: 14-JUL-1978
0000 37
0000 38 Modification History:
0000 39
0000 40 V03-005 WHM0003 Bill Matthews 25-Jun-1984
0000 41 Use the ucb chain in the ddb to display the first 63 units
0000 42 if there is no idb for a device. Fix register usage across
0000 43 calls to SCH$IOUNLOCK.
0000 44
0000 45 V03-004 WHM0002 Bill Matthews 15-Mar-1984
0000 46 Rewrite SHOW/DEVICE algorithm to have the IO database MUTEX
0000 47 for read when accessing UCB's or pointers to UCB's.
0000 48
0000 49 V03-003 MSH0003 Maryann Hinden 31-Jan-1983
0000 50 Changes for cluster device names.
0000 51
0000 52 V03-002 MSH0002 Maryann Hinden 19-Oct-1982
0000 53 Modify display format for 0 UCB pointer.
0000 54
0000 55 V03-001 MSH0001 Maryann Hinden 30-Sep-1982
0000 56 Check for DDB$L_UCB = 0.
0000 57 --
```



```
0000 58
0000 59 :
0000 60 : LOCAL MACROS
0000 61 :
0000 62 : .MACRO STRING_DESC STRING,?L1,?L2
0000 63 : .LONG L2-L1
0000 64 : .LONG L1
0000 65 L1:
0000 66 : .ASCII \STRING\
0000 67 L2:
0000 68 : .ENDM STRING_DESC
0000 69
0000 70 :
0000 71 : EXTERNAL SYMBOLS
0000 72 :
0000 73 : $DPTDEF ;DRIVER PROLOGUE DEFINITIONS
0000 74 : $DDBDEF ;DDB DEFINITIONS
0000 75 : $DYNDDEF ;DEFINE TYPE CODES
0000 76 : $IDBDEF ;IDB DEFINITIONS
0000 77 : $UCBDEF ;UCB DEFINITIONS
0000 78 : $CRBDEF ;CRB DEFINITIONS
0000 79 : $VECDEF ;INTERRUPT VECTOR DEFINITIONS
0000 80 : $TPADEF ;TPARSE DEFINITIONS
0000 81 :
0000 82 : EQUATED VALUES:
0000 83 :
0000000D 0000 84 : CR=13 ; CHARACTER CODE FOR CARRIAGE RETURN
0000000A 0000 85 : LF=10 ; CHARACTER CODE FOR LINEFEED
0000 86 :
0000 87 : OWN STORAGE
0000 88 :
00000000 0000 89 : .PSECT PAGED_DATA rd,wrt,noexe,quad
0000 90
0000 91 HEADER: ;PAGE HEADER
0000 92 : .ASCII <CR><LF>
0002 93 : .ASCII /--Driver--/
000D 94 : .ASCII /--Start--7--/
0016 95 : .ASCII /--End--7--/
001E 96 HEADER2:
001E 97 : .ASCII /-Dev-/
0023 98 : .ASCII /-DDB----/
002C 99 : .ASCII /-CRB----/
0035 100 : .ASCII /-IDB----/
003E 101 : .ASCII /-Unit7---/
0043 102 : .ASCII /-UCB----/
004C 103 HEADER_LEN=-HEADER
004C 104 HEADER_LEN2=HEADER2-HEADER
004C 105
004C 106 DRIVER_FAO: ;DRIVER FAO DESCRIPTOR
004C 107 : STRING_DESC <!10AC !XL !XL>
0061 108 DDB_FAO: ;DDB FAO DESCRIPTOR
0061 109 : STRING_DESC <!3( ) !AC !XL !XL !XL>
0082 110 UCB_FAO: ;UCB FAO DESCRIPTOR
0082 111 : STRING_DESC <!7( ) !4UB !XL>
009B 112 DDB_NF_FAO: ;DDB FAO DESCRIPTOR (WITH NULL FIELDS)
009B 113 : STRING_DESC <!3( ) !AC !XL>
00B4 114
```

00000080	00B4	115	OUTBUF_DESC:			
000000C0	00B4	116	.LONG	OUTBUF_LEN		
	00B8	117	.LONG	OUTBUF		
	00BC	118				
	00BC	119				
	00BC	120				
000000C0	00BC	121	OUTBUF_SIZE:			
	00BC	122	.BLKL	1		
00000140	00C0	123	OUTBUF:			
00000080	00C0	124	.BLKB	128		
	0140	125	OUTBUF_LEN=.	OUTBUF		

;OUTPUT BUFFER DESCRIPTOR

;OUTPUT BUFFER SIZE

;OUTPUT BUFFER


```
0140 127 .SBTTL DISPLAY DEVICE DATABASE TOPOLOGY
0140 128 :++
0140 129 :
0140 130 : FUNCTIONAL DESCRIPTION:
0140 131 :
0140 132 : This routine displays the names and addresses of one or all the
0140 133 : drivers, DDBs, CRBs, IDBs, and UCBs.
0140 134 :
0140 135 : CALLING SEQUENCE:
0140 136 :
0140 137 : CALL BOO$SHODEV(PARMBLK) ;Called as a TPARSE action routine
0140 138 : ; TPASL_TOKENPTR(AP) is driver name descriptor
0140 139 :
0140 140 : CALL BOO$SHODEV_ALL(PARMBLK) ;Called as a TPARSE action routine
0140 141 :
0140 142 : SHOW/DEVICE is called with 0 in TPASL_PARAM(AP)
0140 143 : SHOW/DRIVER is called with 1 in TPASL_PARAM(AP)
0140 144 :
0140 145 :--
00000000 146 .PSECT PAGED_CODE rd,nowrt,exe,long
0000 147 .ENABL LSB
0000 148 BOO$SHODEV_ALL::
0000 149 .WORD 0
10 AC D4 0002 150 CLRL TPASL_TOKENCNT(AP) ;MAKE ANYTHING MATCH
02 11 0005 151 BRB 10$
0007 152
0007 153 BOO$SHODEV::
0000 0007 154 .WORD 0
5E 00000200 8F C2 0009 155 10$: SUBL2 #<64*8>,SP ;ALLOCATE A BUFFER ON THE USER STACK
59 5E D0 0010 156 MOVL SP,R9 ;GET PTR TO THE BUFFER
09 50 E8 0020 157 $CMEXEC S W^DEVTOP_DISP,(AP) ;DO IT ALL IN EXEC MODE
50 DD 0023 158 BLBS R0,20$ ;BR IF SUCCESS
00000000'GF 01 FB 0025 159 PUSHL R0 ;SET ERROR ARG
50 01 D0 002C 160 20$: CALLS #1,G^LIB$SIGNAL ;SIGNAL THE ERROR
04 002F 161 MOVL #1,R0 ;SET SUCCESS FOR PARSER
0030 162 RET
0030 163
0030 164 .DSABL LSB
0030 165
0030 166
0030 167 :
0030 168 : DISPLAY THE HEADER
0030 169 :
0030 170 DEVTOP_DISP:
0000'CF 004C 8F 28 0030 171 .WORD ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
0000'CF 0000'CF 0032 172 MOVC #HEADER_LEN,W^HEADER,-
0000'CF 4C 8F 9B 0039 173 W^RIOSAB_BUFFER ;PUT HEADER IN OUTPUT BUFFER
05 20 AC E9 003C 174 MOVZBW #HEADER_LEN,W^RIOS$GW_OUTLEN ;SET SIZE OF STRING
0000'CF 1E 9B 0042 175 BLBC TPASL_PARAM(AP),10$ ;BRANCH IF SHOW/DEVICE
FFB2' 30 0046 176 MOVZBW #HEADER_LEN2,W^RIOS$GW_OUTLEN ;SET SIZE OF STRING FOR SHOW/DRIVER
004B 177
004B 178 10$: BSBW RIOS$OUTPUT_LINE ;WRITE IT TO OUTPUT DEVICE
004E 179 :
004E 180 : DISPLAY THE DRIVER INFO
004E 181 :
58 00000000'GF D0 004E 182 MOVL G^IOC$GL_DPTLIST,R8 ;GET ADDR OF FIRST DRIVER
0055 183
```

```
14 BC 10 AC 29 0055 184 DISP_DRIV:
      21 AB 0055 185 CMPC TPA$L_TOKENCNT(AP),@TPA$L_TOKENPTR(AP),- ;NAMES MATCH?
      2B 12 005A 186 DPT$T_NAME+1(R8)
50 08 AB 3C 005C 187 BNEQ NEXT_DRIV ;BR IF NOT
      50 58 C0 005E 188 MOVZWL DPT$Q_SIZE(R8),R0 ;GET SIZE OF DRIVER
51 20 AB DE 0062 189 ADDL2 R8,R0 ;SET ADDR OF END OF DRIVER
      0065 190 MOVAL DPT$T_NAME(R8),R1 ;SET ADDR OF DRIVER NAME
      0069 191
      0069 192 $FAO_S CTRSTR = W^DRIVER_FAO,-
      0069 193 OUTBUF = W^RIOSAB_OUTBUF,-
      0069 194 OUTLEN = W^RIOSGW_OUTLEN,-
      0069 195 P1 = R1,-
      0069 196 P2 = R8,-
      0069 197 P3 = R0
      0082 198
      FF7B' 30 0082 199 BSBW RIOSOUTPUT LINE ;WRITE IT TO OUTPUT DEVICE
10 20 AC E9 0085 200 BLBC TPA$L_PARAM(AP),DISP_DEV ;BRANCH IF SHOW/DEVICE
      58 68 D0 0089 201 NEXT_DRIV: ;FIND NEXT DRIVER
00000000'8F 58 D1 0089 202 MOVL DPT$L_FLINK(R8),R8 ;GET ADDR OF NEXT DRIVER
      C0 12 008C 203 CMPL R8,#IOC$GL_DPTLIST ;END OF DRIVER LIST?
      50 01 D0 0093 204 BNEQ DISP_DRIV ;BR IF NOT
      04 0095 205 MOVL #1,R0 ;SET SUCCESS
      0098 206 RET
      0099 207
      0099 208
      0099 209 ;
      0099 210 ; DISPLAY THE DEVICE INFO
      0099 211 ;
      0099 212 DISP_DEV:
      5B D4 0099 213 CLRL R11 ;INITIALIZE
      009B 214 NEXT_DDB:
      009B 215 $CMKRNLS 1$(AP) ;CALL ROUTINE IN KERNAL MODE
      5B D5 00AA 216 TSTL R11 ;IF NO DDB CHECK FOR ANOTHER DRIVER
      DB 13 00AC 217 BEQL NEXT_DRIV ;R0 = 0 IMPLIED DONE WITH THIS DEVICE
      009D 31 00AE 218 BRW DISP_UCB ;DISPLAY THE UCB INFO
      00B1 219
      0000 00B1 220 1$: .WORD ^M<>
54 00000000'GF D0 00B3 221 MOVL G^CTL$GL_PCB,R4 ;GET PCB POINTER
      00000000'GF 16 00BA 222 JSB G^SCH$IOLOCKR ;GET IO DATABASE MUTEX FOR READ
      5A D4 00C0 223 10$: CLRL R10 ;ONLY WANT DDB'S
      00000000'GF 16 00C2 224 JSB G^IOC$SCAN_IDDB ;GET ADDR OF FIRST DDB
      39 50 E9 00C8 225 BLBC R0,30$ ;FINISHED THE ENTIRE CHAIN?
      51 24 AB DE 00CB 226 MOVAL DDB$T_DRVNAME(R11),R1 ;GET ADDR OF DRIVER NAME
      50 81 9A 00CF 227 MOVZBL (R1)+,R0 ;GET SIZE OF DRIVER NAME
      53 20 AB 9E 00D2 228 MOVAB DPT$T_NAME(R8),R3 ;GET ADDR OF DRIVER NAME
      52 83 9A 00D6 229 MOVZBL (R3)+,R2 ;GET SIZE OF DRIVER NAME
63 52 00 61 50 2D 00D9 230 CMPC5 R0,(R1),#0,R2,(R3) ;DOES DRIVER NAME MATCH?
      DF 12 00DF 231 BNEQ 10$ ;ELSE - GET ADDR OF NEXT DDB
      57 14 AB DE 00E1 232 MOVAL DDB$T_NAME(R11),R7 ;GET ADDR OF DEV NAME
      5A 04 AB D0 00E5 233 MOVL DDB$L_UCB(R11),R10 ;GET ADDR OF FIRST UCB
      08 13 00E9 234 BEQL 20$ ;IF EQL, NO UCB LINK THERE
      55 24 AA D0 00EB 235 MOVL UCB$L_CRB(R10),R5 ;GET ADDR OF CRB
      56 2C A5 D0 00EF 236 MOVL CRB$L_INTD+VEC$L_IDB(R5),R6 ;GET ADDR OF IDB
54 00000000'GF D0 00F3 237 20$: MOVL G^CTL$GL_PCB,R4 ;GET PCB POINTER
      00000000'GF 16 00FA 238 JSB G^SCH$IOONLOCK ;RELEASE THE IO DATABASE MUTEX
      0100 239 SETIPL #0 ;AND LOWER IPL
      04 0103 240 RET ;AND RETURN TO EXEC MODE
```



```
5B D4 0104 241 30$: CLRL R11 ;SET END OF DDB CHAIN FLAG
EB 11 0106 242 BRB 20$ ;USE COMMON RETURN
    0108 243
    0108 244
    0108 245 ; DISPLAY THE UCB-RELATED DATA
    0108 246
    0108 247 OUTPUT_UCB INFO:
5A 02 C2 0108 248 SUBL2 #2,R10 ;BACK UP BUFFER INDEX
52 D4 010B 249 CLRL R2 ;CLEAR NEW BUFFER INDEX
    010D 250 10$: $FAO_S CTRSTR = W^UCB FAO,-
    010D 251 OUTBUF = W^RIO$AB_OUTBUF,-
    010D 252 OUTLEN = W^RIO$GW_OUTLEN,-
    010D 253 P1 = 4(R9)[R2],- ;SAVED UNIT NUMBER
    010D 254 P2 = (R9)[R2] ;SAVED UCB ADDRESS
    0127 255
FFDD 52 02 FED6' 30 0127 256 BSBW RIO$OUTPUT_LINE ;WRITE IT TO OUTPUT DEVICE
    F1 012A 257 ACBL R10,#2,R2,T0$ ;LOOP THROUGH THE BUFFER
    05 0130 258 RSB ;RETURN
    0131 259
    0131 260
    0131 261
    0131 262 ; NO UCB-RELATED DATA FOR DISPLAY
    0131 263
    0131 264 NO_UCB:
    0131 265 $FAO_S CTRSTR = W^DDB NF_FAO,-
    0131 266 OUTBUF = W^RIO$AB_OUTBUF,-
    0131 267 OUTLEN = W^RIO$GW_OUTLEN,-
    0131 268 P1 = R7,-
    0131 269 P2 = R11
    0148 270
    FEB5' 30 0148 271 BSBW RIO$OUTPUT_LINE ;WRITE IT TO OUTPUT DEVICE
    FF4D 31 014B 272 BRW NEXT_DDB
    014E 273
    014E 274
    014E 275 DISP_UCB:
    SA D5 014E 276 TSTL R10 ;ANY UCB'S?
    DF 13 0150 277 BEQL NO_UCB ;IF EQL NO
    0152 278
    0152 279 $FAO_S CTRSTR = W^DDB FAO,-
    0152 280 OUTBUF = W^RIO$AB_OUTBUF,-
    0152 281 OUTLEN = W^RIO$GW_OUTLEN,-
    0152 282 P1 = R7,-
    0152 283 P2 = R11,-
    0152 284 P3 = R5,-
    0152 285 P4 = R6 ;SET ADDR OF IDB
    016D 286
    FE90' 30 016D 287 BSBW RIO$OUTPUT_LINE ;WRITE IT TO OUTPUT DEVICE
    0170 288
55 OC A6 3C 0170 289 MOVZWL IDB$W_UNITS(R6),R5 ;GET THE MAXIMUM NUMBER OF UNITS
3F 55 D1 0174 290 CMPL R5,#63 ;MORE THAN 63 UNITS?
    06 15 0177 291 BLEQ DISP_UCB FROM DDB ;IF LEQ THEN USE UCB CHAIN IN DDB
09 OA A6 91 0179 292 CMPB IDB$B_TYPE(R6),#DYN$C_IDB ;IS THIS REALLY AN IDB?
    4B 13 017D 293 BEQL DISP_UCB_FROM_IDB ;IF EQL USE THE UCB LIST IN THE IDB
    017F 294
    017F 295
    017F 296 ; DISPLAY THE UCB INFO FROM THE DDB
    017F 297 ;
```

```
017F 298 DISP_UCB_FROM_DDB:
017F 299
017F 300 $CMKRNLS 1$, (AP) ;CALL ROUTINE IN KERNAL MODE
FF77 30 018E 301 BSBW OUTPUT_UCB_INFO ;OUTPUT THE UCB INFO GATHERED SO FAR
FF07 31 0191 302 BRW NEXT_DDB ;ELSE GET NEXT DDB
0194 303
0000 0194 304 1$: .WORD ^M<> ;
00000000'GF 16 0196 305 JSB G^SCH$IOLOCKR ;GET IO DATABASE MUTEX FOR READ
57 04 AB D0 019C 306 MOVL DDB$L_UCB(R11),R7 ;GET FIRST ADDRESS OF FIRST UCB
8F 13 01A0 307 BEQL NO_UCB ;IF EQL UCB LIST HAS GONE AWAY
5A D4 01A2 308 CLRL R10 ;SET BUFFER INDEX TO ZERO
04 A94A 54 A7 D0 01A4 309 10$: MOVL R7,(R9)[R10] ;LOAD UCB ADDRESS INTO BUFFER
5A 02 C0 01A8 310 MOVZWL UCB$W_UNIT(R7),4(R9)[R10] ;LOAD UNIT NUMBER INTO BUFFER
0000007E 8F 5A D1 01AE 311 ADDL2 #2,R10 ;INCREMENT INDEX INTO BUFFER
06 14 01B1 312 CMPL R10,#<63*2> ;EXIT LOOP IF BUFFER IS FULL
57 30 A7 D0 01B8 313 BGTR 20$ ;IF GTR BUFFER IS FULL
E4 12 01BA 314 MOVL UCB$L_LINK(R7),R7 ;GET ADDRESS OF NEXT UCB
00000000'GF 16 01BE 315 BNEQ 10$ ;IF NEQ THEN PROCESS NEXT UCB
01C0 316 20$: JSB G^SCH$IOUNLOCK ;RELEASE THE IO DATABASE MUTEX
01C6 317 SETIPL #0 ;AND LOWER IPL
04 01C9 318 RET ;RETURN TO EXEC MODE
01CA 319
01CA 320
01CA 321 ;
01CA 322 ; DISPLAY THE UCB INFO FROM THE IDB
01CA 323 ;
01CA 324 DISP_UCB_FROM_IDB:
01CA 325
57 D4 01CA 326 CLRL R7 ;SET UNIT NUMBER TO ZERO
FF2A 30 01CC 327 1$: $CMKRNLS 5$, (AP) ;CALL ROUTINE IN KERNAL MODE
55 57 D1 01DB 328 BSBW OUTPUT_UCB_INFO ;OUTPUT THE UCB INFO GATHERED SO FAR
E9 19 01DE 329 CMPL R7,R5 ;HAVE ALL UNITS BEEN PROCESSED?
FEB5 31 01E1 330 BLSS 1$ ;IF LEQ NO, CONTINUE PROCESSING UNITS
01E3 331 BRW NEXT_DDB ;ELSE GET NEXT DDB
01E6 332
0000 01E6 333 5$: .WORD ^M<> ;
5A D4 01E8 334 CLRL R10 ;SET BUFFER INDEX TO ZERO
00000000'GF 16 01EA 335 JSB G^SCH$IOLOCKR ;GET IO DATABASE MUTEX FOR READ
51 18 A647 D0 01F0 336 10$: MOVL IDB$L_UCBLST(R6)[R7],R1 ;UCB EXIST?
16 13 01F5 337 BEQL 20$ ;IF EQL NO
04 A94A 54 A1 D0 01F7 338 MOVL R1,(R9)[R10] ;IF UCB EXISTS LOAD ADDRESS INTO BUFFER
5A 02 C0 01FB 339 MOVZWL UCB$W_UNIT(R1),4(R9)[R10] ;LOAD UNIT NUMBER INTO BUFFER
0000007E 8F 5A D1 0201 340 ADDL2 #2,R10 ;INCREMENT INDEX INTO BUFFER
04 14 0204 341 CMPL R10,#<63*2> ;EXIT LOOP IF BUFFER IS FULL
DF 57 55 F2 020B 342 BGTR 30$ ;IF GEQ BUFFER IS FULL
00000000'GF 16 020D 343 20$: AOBLS R5,R7,10$ ;LOOP FOR ALL UNITS
0211 344 30$: JSB G^SCH$IOUNLOCK ;RELEASE THE IO DATABASE MUTEX
0217 345 SETIPL #0 ;AND LOWER IPL
04 021A 346 RET ;AND RETURN TO EXEC MODE
021B 347
021B 348 .END
```


SHODEV
Symbol table

- DISPLAY DEVICE DATABASE TOPOLOGY^{N 3}

16-SEP-1984 00:02:39 VAX/VMS Macro V04-00
4-SEP-1984 23:05:57 [BOOTS.SRC]SHODEV.MAR;1

Page 8
(1)

\$ST2	= 00000007		
BOOSSHODEV	00000007	RG	03
BOOSSHODEV_ALL	00000000	RG	03
CR	= 00000000		
CRBSL_INTD	= 00000024		
CTL\$GC_PCB	*****	X	03
DDB\$SL_UCB	= 00000004		
DDB\$T_DRVNAME	= 00000024		
DDB\$T_NAME	= 00000014		
DDB_FAO	00000061	R	02
DDB_NF_FAO	0000009B	R	02
DEVTOP_DISP	00000030	R	03
DISP_DEV	00000099	R	03
DISP_DRIV	00000055	R	03
DISP_UCB	0000014E	R	03
DISP_UCB_FROM_DDB	0000017F	R	03
DISP_UCB_FROM_IDB	000001CA	R	03
DPT\$C_FLINK	= 00000000		
DPT\$T_NAME	= 00000020		
DPT\$W_SIZE	= 00000008		
DRIVER_FAO	0000004C	R	02
DYN\$C_IDB	= 00000009		
HEADER	00000000	R	02
HEADER2	0000001E	R	02
HEADER_LEN	= 0000004C		
HEADER_LEN2	= 0000001E		
IDB\$B_TYPE	= 0000000A		
IDB\$SL_UCBLST	= 00000018		
IDB\$W_UNITS	= 0000000C		
IOC\$GC_DPTLIST	*****	X	03
IOC\$SCAN_IDB	*****	X	03
LF	= 0000000A		
LIB\$SIGNAL	*****	X	03
NEXT_DDB	0000009B	R	03
NEXT_DRIV	00000089	R	03
NO_UCB	00000131	R	03
OUTBUF	000000C0	R	02
OUTBUF_DESC	000000B4	R	02
OUTBUF_LEN	= 00000080		
OUTBUF_SIZE	000000BC	R	02
OUTPUT_UCB_INFO	00000108	R	03
PR\$IPC	*****	X	03
RIO\$AB_BUFFER	*****	X	03
RIO\$AB_OUTBUF	*****	X	03
RIO\$GW_OUTLEN	*****	X	03
RIO\$OUTPUT_LINE	*****	X	03
SCH\$IOLOCKR	*****	X	03
SCH\$IOUNLOCK	*****	X	03
SYSS\$CMEXEC	*****	GX	03
SYSS\$CMKRNL	*****	GX	03
SYSS\$FAO	*****	X	03
TPASL_PARAM	= 00000020		
TPASL_TOKENCNT	= 00000010		
TPASL_TOKENPTR	= 00000014		
UCBSL_CRB	= 00000024		
UCBSL_LINK	= 00000030		
UCBSW_UNIT	= 00000054		

UCB_FAO
VEC\$SL_IDB

00000082 R 02
= 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
PAGED_DATA	00000140 (320.)	02 (2.)	NOPIC USR CON REL LCL NOSHR NOEXE RD WRT NOVEC QUAD
PAGED_CODE	0000021B (539.)	03 (3.)	NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC LONG

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

Phase	Page faults	CPU Time	Elapsed Time
Initialization	37	00:00:00.09	00:00:00.90
Command processing	141	00:00:00.72	00:00:04.38
Pass 1	268	00:00:07.71	00:00:15.56
Symbol table sort	0	00:00:01.01	00:00:01.93
Pass 2	74	00:00:01.62	00:00:03.51
Symbol table output	8	00:00:00.07	00:00:00.07
Psect synopsis output	2	00:00:00.04	00:00:00.22
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	532	00:00:11.27	00:00:26.57

The working set limit was 1350 pages.
41850 bytes (82 pages) of virtual memory were used to buffer the intermediate code.
There were 40 pages of symbol table space allocated to hold 683 non-local and 24 local symbols.
348 source lines were read in Pass 1, producing 17 object records in Pass 2.
22 pages of virtual memory were used to define 20 macros.

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

Macro library name	Macros defined
\$255\$DUA28:[BOOTS.OBJ]BOOTS.MLB;1	0
\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	8
\$255\$DUA28:[SYSLIB]STARLET.MLB;2	8
TOTALS (all libraries)	16

770 GETS were required to define 16 macros.

There were no errors, warnings or information messages.

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

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

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY