

BBBBBBBBBBBB		00000000		00000000		TTTTTTTTTTTT		SSSSSSSSSS
BBBBBBBBBBBB		00000000		00000000		TTTTTTTTTTTT		SSSSSSSSSS
BBBBBBBBBBBB		00000000		00000000		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	
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		00000000		00000000		TTTTTTTTTTTT		SSSSSSSSSS
BBBBBBBBBBBB		00000000		00000000		TTTTTTTTTTTT		SSSSSSSSSS
BBBBBBBBBBBB		00000000		00000000		TTTTTTTTTTTT		SSSSSSSSSS

```
LL      000000      AAAAAA      DDDDDDDD      DDDDDDDD      RRRRRRRR      IIIIII      VV      VV
LL      000000      AAAAAA      DDDDDDDD      DDDDDDDD      RRRRRRRR      IIIIII      VV      VV
LL      00      00      AA      AA      DD      DD      DD      DD      RR      RR      II      VV      VV
LL      00      00      AA      AA      DD      DD      DD      DD      RR      RR      II      VV      VV
LL      00      00      AA      AA      DD      DD      DD      DD      RR      RR      II      VV      VV
LL      00      00      AA      AA      DD      DD      DD      DD      RR      RR      II      VV      VV
LL      00      00      AA      AA      DD      DD      DD      DD      RRRRRRRR      II      VV      VV
LL      00      00      AA      AA      DD      DD      DD      DD      RRRRRRRR      II      VV      VV
LL      00      00      AAAAAAAAAA      DD      DD      DD      DD      RR      RR      II      VV      VV
LL      00      00      AAAAAAAAAA      DD      DD      DD      DD      RR      RR      II      VV      VV
LL      00      00      AA      AA      DD      DD      DD      DD      RR      RR      II      VV      VV
LL      00      00      AA      AA      DD      DD      DD      DD      RR      RR      II      VV      VV
LL      00      00      AA      AA      DD      DD      DD      DD      RR      RR      II      VV      VV
LLLLLLLLLLLL      000000      AA      AA      DDDDDDDD      DDDDDDDD      RR      RR      IIIIII      VV      VV
LLLLLLLLLLLL      000000      AA      AA      DDDDDDDD      DDDDDDDD      RR      RR      IIIIII      VV      VV
```

```
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
LLLLLLLLLLLL      IIIIII      SSSSSSSS
LLLLLLLLLLLL      IIIIII      SSSSSSSS
```

LOADDRIV
Table of contents

- LOAD A DEVICE DRIVER

N 12

15-SEP-1984 23:53:35 VAX/VMS Macro V04-00

Page 0

(1) 58
(1) 104
(1) 231

DECLARATIONS
LOAD DRIVER
LOADER - LOAD THE DRIVER

```

0000 1 .TITLE LOADDRIV - LOAD A DEVICE DRIVER
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: DEVICE DRIVER LOADER
0000 31 :
0000 32 : ABSTRACT: LOAD A DRIVER
0000 33 :
0000 34 :
0000 35 : ENVIRONMENT: USER MODE PRIVILEGED CODE
0000 36 :
0000 37 : AUTHOR: LEN KAWELL, CREATION DATE:8-JUN-78
0000 38 :
0000 39 : MODIFICATION HISTORY:
0000 40 :
0000 41 : V03-005 WHM0001 Bill Matthews 01-Feb-1984
0000 42 : Set AP correctly before calling IOGEN$CNTRL_INI.
0000 43 :
0000 44 : V03-004 BLS0223 Benn Schreiber 13-May-1983
0000 45 : Fix truncation error
0000 46 :
0000 47 : V03-003 MSH0003 Maryann Hinden 15-Feb-1983
0000 48 : Check earlier for nounload bit set in DPT.
0000 49 :
0000 50 : V03-002 MSH0002 Maryann Hinden 09-Feb-1983
0000 51 : Change references to IOGEN$GL_DPT to ACF$GL_DPT.
0000 52 : Move SGN$GL_DDB, et. al., to ACF$GL_xxx.
0000 53 :
0000 54 : V03-001 MSH0001 Maryann Hinden 07-Jan-1982
0000 55 : If zero UCB pointer in DDB, disallow reload.
0000 56 :--

```

```

0000 58          .SBTTL  DECLARATIONS
0000 59
0000 60
0000 61  :
0000 62  : INCLUDE FILES:
0000 63  :
0000 64  :
0000 65  :
0000 66  : MACROS:
0000 67  :
0000 68  :
0000 69  :
0000 70  : EQUATED SYMBOLS:
0000 71  :
00000004 0000 72  FILE_NAME =      4          ;OFFSET TO FILE NAME ARG
0000 73
0000 74          $IHDEF          ;DEFINE IMAGE HEADER OFFSETS
0000 75          $IHPDEF         ;DEFINE IMAGE HEADER OFFSETS
0000 76          $IHDEF         ;DEFINE IMAGE HEADER OFFSETS
0000 77          $JPIDEF        ;DEFINE SYSSGETJPI VALUES
0000 78          $DPTDEF        ;DEFINE DRIVER PROLOGUE TABLE
0000 79          $DYNDDEF       ;DEFINE DYNAMIC MEMORY TYPES
0000 80          $DDBDEF        ;DEFINE DEVICE DATA BLOCK
0000 81          $UCBDEF        ;DEFINE UNIT CONTROL BLOCK
0000 82          $PRDEF         ;DEFINE PROCESSOR REGISTERS
0000 83          $SSDEF         ;DEFINE SYSTEM MESSAGES
0000 84          $SYSGMSGDEF    ;DEFINE SYSGEN MESSAGES
0000 85
0000 86  :
0000 87  : OWN STORAGE:
0000 88  :
00000000 0000 89          .PSECT  NONPAGED_DATA   rd,wrt,noexe,quad
0000 90
00000008 0000 91  RET_ADDR:          ;MAP RANGE ARRAY RETURNED BY SYS
0000 92          .BLKL   2
0008 93
0008 94  CHANNEL:          ;LOCATION TO STORE CHANNEL NUMBER
0000000A 0008 95          .BLKW   1
000A 96
000A 97  LINK_TIME:
00000012 000A 98          .BLKL   2          ;LINK DATE TEMPORARY
0012 99  ECO_LEVEL:
00000016 0012 100         .BLKL   1          ;ECO TEMPORARY
0016 101
0016 102

```

```

0016 104 .SBTTL LOAD DRIVER
0016 105
0016 106 :++
0016 107 : FUNCTIONAL DESCRIPTION:
0016 108 :
0016 109 : This routine will load a device driver into the system pool
0016 110 : and if the same name driver was already loaded, it will
0016 111 : reinitialize all the control-blocks with info from the new
0016 112 : driver and unload the old driver.
0016 113 :
0016 114 : CALLING SEQUENCE:
0016 115 :
0016 116 : CALL IOGEN$LOAD(FILE_NAME)
0016 117 :
0016 118 : INPUT PARAMETERS:
0016 119 :
0016 120 : FILE_NAME(AP) = address of file name counted string
0016 121 :
0016 122 : IMPLICIT INPUTS:
0016 123 :
0016 124 : NONE
0016 125 :
0016 126 : OUTPUT PARAMETERS:
0016 127 :
0016 128 : ACF$GL_DPT = ADDR OF NEW DRIVER PROLOGUE TABLE
0016 129 :
0016 130 : IMPLICIT OUTPUTS:
0016 131 :
0016 132 : NONE
0016 133 :
0016 134 : COMPLETION CODES:
0016 135 :
0016 136 : R0 = STATUS OF LOAD OPERATION
0016 137 :
0016 138 : SIDE EFFECTS:
0016 139 :
0016 140 : NONE
0016 141 :
0016 142 :--
00000000 143 .PSECT NONPAGED_CODE rd,nowrt,exe,long
0000 144
0000 145 IOGEN$LOADDRIV::
003C 0000 146 .WORD ^M<R2,R3,R4,R5>
0002 147
50 04 AC D0 0002 148 MOVL FILE_NAME(AP),R0 ;GET ADDR OF FILE NAME STRING
53 0008'CF 3E 0006 149 MOVAW W^CHANNEL,R3 ;GET ADDR TO STORE CHANNEL #
54 0000'CF 7E 000B 150 MOVAG W^RET_ADDR,R4 ;GET ADDR OF RETURN ADDRESS ARRAY
00000000'GF 16 0010 151 JSB G^IOGEN$READDRIV ;READ IN DRIVER
03 50 E8 0016 152 BLBS R0,3$ ;BRANCH IF SUCCESS
00C6 31 0019 153 BRW 70$ ;BR IF ERROR
001C 154 :
001C 155 : CHECK IF DRIVER IS ASSEMBLED AND LINKED WITH CORRECT VERSION OF VMS,
001C 156 : AFTER VALIDATING DPT
001C 157 :
55 64 D0 001C 158 3$: MOVL (R4),R5 ;GET ADDR OF IMAGE HEADER
50 10 A5 9A 001F 159 MOVZBL IHD$B HDRBLKCNT(R5),R0 ;GET IMAGE HEADER BLOCK COUNT
50 00000200 8F C4 0023 160 MULL #512,R0 ;COMPUTE HEADER BYTE COUNT

```

```

64 50 C0 002A 161 ADDL R0,(R4) ;SET ADDRESS OF FIRST IMAGE SECTION (DPT)
51 64 D0 002D 162 MOVL (R4),R1 ;GET ADDRESS OF DRIVER PROLOGUE TABLE
0030 163
1E 0A A1 91 0030 164 CMPB DPT$B_TYPE(R1),#DYN$C_DPT ;IS IT A VALID DRIVER PROLOGUE
08 13 C034 165 BEQLU 5$ ;BR IF YES
007C8032 8F DD 0036 166 PUSHL #SYSG$ _INVDPT ;SET INVALID DPT ERROR
74 11 003C 167 BRB 40$ ;BRANCH TO EXIT
003E 168
10 A1 18 B1 003E 169 5$: CMPW #DPT$W_VERSION,DPT$W_INITTAB(R1) ;DOES DPT HAVE A VERSION NUMBER?
18 1E 0042 170 BGEQU 10$ ;BR IF NO - DEFINITELY NOT RIGHT
04 18 A1 B1 0044 171 CMPW DPT$W_VERSION(R1),#DPT$C_VERSION ;DOES VERSION MATCH OUR VERSION?
15 12 0048 172 BNEQ 10$ ;BR IF NO - NOT ASSEMBLED WITH RIGHT SYS
02 A5 28 B1 004A 173 CMPW #IHDS$L_SYSVER,IHDS$W_ACTIVOFF(R5) ;DOES HEADER HAVE A SYS VERSION?
0F 1E 004E 174 BGEQU 10$ ;BR IF NO - DEFINITELY NOT RIGHT
50 28 A5 D0 0050 175 MOVL IHDS$L_SYSVER(R5),R0 ;GET SYSTEM VERSION OF IMAGE
28 13 0054 176 BEQL 30$ ;BR IF NONE - NOT LINKED WITH EXEC! ?
00000000'8F 50 D1 0056 177 CML R0,#SYSS$K_VERSION ;DOES IMAGE VERSION MATCH OUR VERSION?
1F 13 005D 178 BEQL 30$ ;BR IF YES
04 AC DD 005F 179 10$: PUSHL FILE_NAME(AP)
01 DD 0062 180 PUSHL #1
007CA023 8F DD 0064 181 PUSHL #SYSG$ _DRIVENAM
7E D4 006A 182 CLRL -(SP)
007C806A 8F DD 006C 183 PUSHL #SYSG$ _SYSVERDIF ;SET FAILURE STATUS
00000000'GF 05 FB 0072 184 CALLS #5,G*LIB$SIGNAL ;SIGNAL
7E 02 CE 0079 185 MNEGL #2,-(SP) ;SET 'ERROR ALREADY SIGNALLED'
34 11 007C 186 BRB 40$ ;EXIT
007E 187
007E 188 ; DRIVER IS OK, FILL IN DPT FIELDS
007E 189
007E 190 30$:
50 06 A5 3C 007E 191 MOVZWL IHDS$W_IMGIDOFF(R5),R0 ; GET IMAGE IHI OFFSET
50 55 C0 0082 192 ADDL2 R5,R0 ; CALCULATE ADDRESS OF IHI
0000000A'EF 38 A0 7D 0085 193 MOVQ IHI$Q_LINKTIME(R0), - ; STORE IN TEMPORARY
008D 194 LINK_TIME
008D 195
00000012'EF D4 008D 196 CLRL ECO_LEVEL ; ASSUME NO PATCHES
50 08 A5 3C 0093 197 MOVZWL IHDS$W_PATCHOFF(R5),R0 ; GET IMAGE IHP OFFSET
0A 13 0097 198 BEQL 35$ ; BRANCH IF NONE
50 55 C0 0099 199 ADDL2 R5,R0 ; CALCULATE ADDRESS OF IHP
00000012'EF 60 D0 009C 200 MOVL IHP$L_ECO1(R0), - ; STORE IN TEMPORARY
00A3 201 ECO_LEVEL
00A3 202
00A3 203 ; CHANGE MODE TO KERNEL TO LOAD
00A3 204
00A3 205 35$:
50 DD 00A3 206 $CMKRNL_S W^LOADER,(AP) ;
00B0 207 PUSHL -R0 ;SAVE STATUS
00B2 208
00B2 209 ; DELETE THE INPUT FILE VIRTUAL ADDRESSES, DEASSIGN THE CHANNEL, AND RETURN
00B2 210
64 55 D0 00B2 211 40$: MOVL R5,(R4) ;RESET ADDRESS RANGE TO INCLUDE HEADER
00B5 212 $DELTVA_S (R4) ;DELETE THE CREATED ADDRESS RANGE
03 50 E8 00C2 213 BLBS -R0,50$ ;BR IF SUCCESS
6E 50 D0 00C5 214 MOVL R0,(SP) ;SAVE ERROR STATUS
00C8 215 50$: $DASSGN_S (R3) ;DEASSIGN THE CHANNEL
03 50 E8 00D2 216 BLBS -R0,60$ ;BR IF SUCCESS
6E 50 D0 00D5 217 MOVL R0,(SP) ;SAVE ERROR STATUS

```

```
00000000'EF 00 FB 00D8 218 6J$:  
00D8 219 :  
00D8 220 ; CHECK TO SEE IF SCS CODE NEEDS TO BE LOADED AND LOAD IT IF NECESSARY.  
00D8 221 :  
00D8 222 CALLS #0,BOO$SCSLOADER ;CALL TO LOAD  
00DF 223 ;(ALWAYS RETURNS SUCCESS)  
00DF 224  
50 8ED0 00DF 225 65$: POPL R0 ;RESTORE THE STATUS  
00E2 226  
04 00E2 227 70$:  
00E2 228 RET  
00E3 229
```



```

OOE3 231 .SBTTL LOADER - LOAD THE DRIVER
OOE3 232 :++
OOE3 233 :
OOE3 234 : LOADER - LOAD THE NEW DRIVER
OOE3 235 :
OOE3 236 : Kernel mode routine to actually load the driver, initialize
OOE3 237 : any associated control blocks, and unload a previous driver.
OOE3 238 :
OOE3 239 : INPUTS:
OOE3 240 :
OOE3 241 : RET_ADDR = ADDRESS OF MAPPED DRIVER SECTION
OOE3 242 : LINK_TIME = QUADWORD LINK TIME FROM IMAGE HEADER
OOE3 243 : ECO_LEVEL = LONGWORD ECO LEVEL FROM IMAGE HEADER
OOE3 244 :
OOE3 245 : OUTPUTS:
OOE3 246 :
OOE3 247 : RO = STATUS OF OPERATION
OOE3 248 : ACF$GL_DPT = ADDRESS OF NEW DRIVER PROLOGUE TABLE
OOE3 249 :
OOE3 250 :--
OOE3 251 LOADER:
OFFC OOE3 252 .WORD ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
OOE5 253 :
OOE5 254 : MOVE DRIVER TO NON-PAGED POOL
OOE5 255 :
59 0000'CF D0 OOE5 256 MOVL W^RET_ADDR,R9 ;GET ADDR OF DRIVER PROLOGUE
OOEA 257
51 08 A9 3C OOE5 258 5$: MOVZWL DPT$W_SIZE(R9),R1 ;GET SIZE OF DPT
00000000'GF 16 OOE5 259 JSB G^EXE$ALONONPAGED ;ALLOCATE NON-PAGED MEMORY
06 50 E8 OOE5 260 BLBS RO,10$ ;BR IF SUCCESS
50 0124 8F 3C OOE5 261 MOVZWL #SS$_INSFMEM,R0 ;SET INSUFFICIENT MEMORY ERROR
04 OOE5 262 RET ;...EXIT
OOFD 263
OOFD 264 10$: PUSHL #SS$ NORMAL ;ASSUME SUCCESS
68 5A 51 7D OOE5 265 MOVQ R1,R10 ;SAVE SIZE AND ADDR OF BUFFER
08 AB 5A B0 OOE5 266 MOVQ DPT$W_SIZE(R9),(R9),(R11) ;MOVE DRIVER TO THE BUFFER
2C AB 0000000A'EF 7D OOE5 267 MOVW R10,DPT$W_SIZE(R11) ;SET ACTUAL SIZE OF BUFFER
OOE5 268 MOVQ LINK_TIME,-
OOE5 269 DPT$Q_LINKTIME(R11) ; SET LINK TIME
34 AB 00000012'EF D0 OOE5 270 MOVL ECO_LEVEL,-
OOE5 271 DPT$L_ECOLEVEL(R11) ; COPY FIRST LONGWORD OF ECO UPDATES
OOE5 272 :
OOE5 273 : SCAN DPT LIST TO CHECK IF DRIVER CURRENTLY LOADED
OOE5 274 :
OOE5 275 : BSBW IOGEN$LOCK_IODB ;LOCK THE I/O DATABASE
59 20 AB 9E OOE5 276 MOVAB DPT$T_NAME(R11),R9 ;GET ADDR OF DRIVER NAME
58 89 9A OOE5 277 MOVZBL (R9)+,R8 ;GET SIZE OF DRIVER NAME
5A 00000000'GF 9E OOE5 278 MOVAB G^IOC$GL_DPTLIST,R10 ;GET ADDR OF DPT LIST
54 5A D0 OOE5 279 MOVL R10,R4 ;SAVE IT
OOE5 280 20$:
OOE5 281 MOVL DPT$L_FLINK(R10),R10 ;GET ADDR OF NEXT DPT
OOE5 282 CMPL R10,R4 ;END OF LIST?
OOE5 283 BNEQ 30$ ;BR IF NOT
OOE5 284 BRW LINK_DPT ;ELSE NOT LOADED - JUST ADD TO LIST
51 20 AA 9E OOE5 285 30$: MOVAB DPT$T_NAME(R10),R1 ;GET ADDR OF DRIVER NAME
69 58 00 50 81 9A OOE5 286 MOVZBL (R1)+,R0 ;GET SIZE OF DRIVER NAME
61 50 2D OOE5 287 CMPC5 R0,(R1),#0,R8,(R9) ;COMPARE DRIVER NAMES

```

```

SA OD AA E6 12 0147 288 BNEQ 20$ ;BR IF NOT EQUAL
02 E0 0149 289 BBS #DPT$V NOUNLOAD,DPT$B_FLAGS(R10),NOUNLOAD ;BR IF NOT TO BE UNLOADED
OB AA 95 014E 290 TSTB DPT$B_REFC(R10) ;ANY DDB'S CONNECTED TO IT?
03 12 0151 291 BNEQ BUSY_CHECK ;BR IF YES
010C 31 0153 292 BRW UNLOAD ;ELSE - JUST UNLOAD IT
0156 293
0156 294 ; SCAN DEVICE DATABASE TO CHECK THAT ANY DEVICES THAT WILL USE
0156 295 ; THE NEW DRIVER ARE NOT BUSY
0156 296
0156 297 BUSY_CHECK:
0156 298 DSBINT ;DISABLE INTERRUPTS
56 00000000'GF DE 015C 299 MOVAL G^IOC$GL_DEVLIST,R6 ;GET ADDRESS OF DEVICE LISTHEAD
56 66 DO 0163 300 10$: MOVL DDB$L_LINK(R6),R6 ;GET ADDR OF NEXT DDB
2A 13 0166 301 BEQL UNLOAD_CHK ;BR IF NONE - NONE ARE BUSY
51 24 A6 9E 0168 302 MOVAB DDB$T_DRVNAME(R6),R1 ;GET ADDR OF DRIVER NAME
50 81 9A 016C 303 MOVZBL (R1)+,R0 ;GET SIZE OF DRIVER NAME
69 58 00 61 50 2D 016F 304 CMPC5 R0,(R1),#0,R8,(R9) ;DRIVER NAMES MATCH?
EC 12 0175 305 BNEQ 10$ ;BR IF NOT
57 04 A6 DO 0177 306 MOVL DDB$L_UCB(R6),R7 ;GET ADDR OF FIRST UCB
E6 13 017B 307 BEQL 10$ ;BR IF NONE
08 64 A7 08 E0 017D 308 20$: BBS #UCB$V_BSY,UCB$W_STS(R7),30$ ;BR IF DRIVER BUSY
57 30 A7 DO 0182 309 MOVL UCB$L_LINK(R7),R7 ;GET ADDR OF NEXT UCB
DB 13 0186 310 BEQL 10$ ;BR IF NONE
F3 11 0188 311 BRB 20$ ;CONTINUE CHECKING UCB'S
018A 312
56 02C4 8F 3C 018A 313 30$: MOVZWL #SS$_DEVACTIVE,R6 ;SET ERROR STATUS
00A2 31 018F 314 BRW ERROR ;...EXIT
0192 315
0192 316 ; CHECK IF CURRENT DRIVER CAN BE UNLOADED
0192 317
0192 318 UNLOAD_CHK:
50 14 AA B0 0192 319 MOVW DPT$W_UNLOAD(R10),R0 ;GET ADDR OF UNLOAD ACTION ROUTINE
1A 13 0196 320 BEQL REINIT ;BR IF NONE
50 5A C0 0198 321 ADDL R10,R0 ;GET ADDR OF UNLOAD ROUTINE
1FC0 8F BB 019B 322 PUSHB #*M<R6,R7,R8,R9,R10,R11,AP> ;SAVE REGISTERS
60 16 019F 323 JSB (R0) ;CALL THE UNLOAD ROUTINE
1FC0 8F BA 01A1 324 POPB #*M<R6,R7,R8,R9,R10,R11,AP> ;RESTORE REGISTERS
0A 50 E3 01A5 325 BLBS R0,REINIT ;BR IF OK TO UNLOAD
01A8 326
01A8 327 NOUNLOAD:
56 007C8012 8F DO 01A8 328 MOVL #SYSG$_NOUNLOAD,R6 ;SET ERROR STATUS
0082 31 01AF 329 BRW ERROR ;...EXIT
01B2 330
01B2 331 ; RE-SCAN DDB LIST TO FIND DDB'S CONNECTED TO CURRENT DRIVER
01B2 332 ; AND RE-INIT THE ASSOCIATED CONTROL BLOCKS
01B2 333
01B2 334 REINIT:
56 00000000'EF D4 01B2 335 CLRL ACF$GL_DDB ;CLEAR ADDRESS OF DDB
00000000'GF DE 01B8 336 MOVAL G^IOC$GL_DEVLIST,R6 ;GET ADDRESS OF DEVICE LIST
56 66 DO 01BF 337 10$: MOVL DDB$L_LINK(R6),R6 ;GET ADDR OF NEXT DDB
03 12 01C2 338 BNEQ 15$ ;BR IF ANOTHER
0082 31 01C4 339 BRW UNLOAD_ENBINT ;BRANCH IF NONE
51 24 A6 9E 01C7 340 15$: MOVAB DDB$T_DRVNAME(R6),R1 ;GET ADDR OF DRIVER NAME
50 81 9A 01CB 341 MOVZBL (R1)+,R0 ;GET SIZE OF DRIVER NAME
69 58 00 61 50 2D 01CE 342 CMPC5 R0,(R1),#0,R8,(R9) ;DRIVER NAMES MATCH?
E9 12 01D4 343 BNEQ 10$ ;BR IF NO
01D6 344

```

```

00000000'EF 56 DO 01D6 345      MOVL      R6,ACF$GL_DDB      ;SET ADDRESS OF DDB
      55 04 A6 DO 01DD 346      MOVL      DDB$L_UCB(R6),R5    ;SET ADDR OF UCB
      2F 13 01E1 347      BEQL      25$                ;IF ZERO POINTER, DISALLOW RELOAD
      0B AA 97 01E3 348      DECB      DPT$B_REFC(R10)    ;DEC REF COUNT OF CURRENT DRIVER
      0B AB 96 01E6 349      INCB      DPT$B_REFC(R11)    ;INC REF COUNT OF NEW DRIVER
      54 5B DO 01E9 350      MOVL      R11,R4            ;SET ADDR OF DPT
00000000'GF 16 01EC 351 20$: JSB      G^IOC$REINITDRV    ;RE-INIT DRIVER DATABASE
      27 50 E9 01F2 352      BLBC      R0,30$           ;BR IF ERROR
      0C A6 DO 01F5 353      MOVL      DDB$L_DDT(R6),-    ;SET DDT ADDRESS
      0088 C5 01F8 354      UCB$L_DDT(R5)              ;GET ADDR OF NEXT UCB
      55 30 A5 DO 01FB 355      MOVL      UCB$L_LINK(R5),R5 ;BR IF THERE IS ONE
      EB 12 01FF 356      BNEQ     20$                ;SAVE AP
      5C DD 0201 357      PUSHL   AP                  ;LOAD AUTOCONFIGURE CONTROL BLOCK ADDRESS
5C 00000000'EF 9E 0203 358      MOVAB   BOOSAL_ACF,AP        ;RE-INIT THE CONTROLLER
      FDF3' 30 020A 359      BSBW    IOGEN$CNTRL_INI     ;RESTORE AP
      5C 8ED0 020D 360      POPL    AP                  ;CHECK NEXT DDB
      AD 11 0210 361      BRB     10$
      0212 362
56 007C8012 8F DO 0212 363 25$: MOVL     #SYSG$_NOUNLOAD,R6    ;SET ERROR STATUS
      0018 31 0219 364      BRW     ERROR              ;...EXIT
      021C 365
56 007C800A 8F DO 021C 366 30$: MOVL     #SYSG$_INVDPTINI,R6    ;SET ERROR STATUS
      53 53 D7 0223 367      DECL    R3                  ;DECREMENT INIT TABLE POINTER
0000'CF 53 5B C3 0225 368      SUBL3   R11,R3,W^ACF$GL_DPT ;SAVE TABLE OFFSET FOR DEBUGGING
      54 5A DO 022B 369      MOVL    R10,R4            ;SET ADDR OF CURRENT UCB
      00000000'GF 16 022E 370      JSB     G^IOC$REINITDRV    ;TRY TO UNDO ANY DAMAGE
      0234 371      ;
      0234 372      ; ERROR LOADING NEW DRIVER - UNLOAD IT
      0234 373      ;
      0234 374      ; ERROR:
      0234 375
      50 5B DO 0237 376      ENBINT   ;ENABLE INTERRUPTS
00000000'GF 16 023A 377      MOVL    R11,R0            ;SET ADDR OF DRIVER
      FDBD' 30 0240 378      JSB     G^EXE$DEANONPAGED  ;DEALLOCATE ITS STORAGE
      50 56 DO 0243 379      BSBW    IOGEN$UNLK_IODB    ;UNLOCK THE I/O DATABASE
      8E D5 0246 380      MOVL    R6,R0            ;SET ERROR STATUS
      04 0248 381      TSTL   (SP)+          ;THROW AWAY SUCCESS STATUS
      0249 382      RET
      0249 383
      0249 384      ; UNLOAD CURRENTLY LOADED DRIVER
      0249 385      ;
      0249 386      UNLOAD_ENBINT:
      0249 387      ENBINT      ;RE-ENABLE INTERRUPTS
      024C 388      ;
      024C 389      ; See if there is a driver to relocate
      024C 390      ;
      0000'CF D5 024C 391      TSTL    W^ACF$GL_DDB    ;WAS AN ADDRESS SET?
      10 13 0250 392      BEQL    UNLOAD          ;BRANCH IF NO
      5B DD 0252 393      PUSHL   R11              ;SAVE DPT
      5B 0000'CF DO 0254 394      MOVL    W^ACF$GL_DDB,R11    ;SET ADDRESS OF DDB
      00000000'GF 16 0259 395      JSB     G^IOC$RELOC_DDT    ;RELOCATE DRIVER DDT and FDT
      5B 8ED0 025F 396      POPL    R11              ;RESTORE DPT
      C262 397      UNLOAD:
      0B AA 95 0262 398      TSTB   DPT$B_REFC(R10)    ;ANY REFERENCES LEFT?
      18 12 0265 399      BNEQ   LINK_DPT          ;BR IF YES
      5A 6A OF 0267 400      REMQUE  (R10),R10        ;REMOVE DPT FROM LIST
50 00000000'GF 9E 026A 401      MOVAB   G^EXE$INIT,R0     ;GET ADDR OF TOP OF EXEC

```

```
50 5A D1 0271 402      CMPL  R10,R0      ;DRIVER IN EXEC?
      09 1F 0274 403      BLSSU LINK_DPT    ;BR IF YES
50 5A D0 0276 404      MOVL  R10,R0      ;SET ADDR OF DPT
00000000'GF 16 0279 405      JSB   G^EXE$DEANONPAGED ;DEALLOCATE BUFFER
      027F 406      :
      027F 407      : LINK NEW DRIVER INTO DPT LIST
      027F 408      :
      027F 409      LINK_DPT:
00000000'GF 6B 0E 027F 410      INSQUE (R11),G^IOC$GL_DPTLIST ;INSERT NEW DRIVER INTO DPT LIST
      FD77' 30 0286 411      BSBW  IOGEN$UNLK IODB ;UNLOCK THE I/O DATABASE
0000'CF 5B D0 0289 412      MOVL  R11,W^ACF$GL_DPT ;SAVE ADDR OF PROLOGUE TABLE
      50 8ED0 028E 413      POPL  R0 ;SET STATUS
      04 0291 414      RET
      0292 415
      0292 416      .END
```

LOADDRIV
Symbol table

- LOAD A DEVICE DRIVER

K 13

15-SEP-1984 23:53:35 VAX/VMS Macro V04-00
4-SEP-1984 23:04:29 [BOOTS.SRC]LOADDRIV.MAR;1

ACF\$GL_DDB	*****	X	03
ACF\$GL_DPT	*****	X	03
BOO\$AL_ACF	*****	X	03
BOO\$SC\$LOADER	*****	X	03
BUSY_CHECK	00000156	R	03
CHANNEL	00000008	R	02
DDB\$SL_DDT	= 0000000C		
DDB\$SL_LINK	= 00000000		
DDB\$SL_UCB	= 00000004		
DDB\$T_DRVNAME	= 00000024		
DPT\$B_FLAGS	= 0000000D		
DPT\$B_REFC	= 0000000B		
DPT\$B_TYPE	= 0000000A		
DPT\$C_VERSION	= 00000004		
DPT\$L_ECOLEVEL	= 00000034		
DPT\$L_FLINK	= 00000000		
DPT\$Q_LINKTIME	= 0000002C		
DPT\$T_NAME	= 00000020		
DPT\$V_NOUNLOAD	= 00000002		
DPT\$W_INITTAB	= 00000010		
DPT\$W_SIZE	= 00000008		
DPT\$W_UNLOAD	= 00000014		
DPT\$W_VERSION	= 00000018		
DYN\$C_DPT	= 0000001E		
ECO_LEVEL	00000012	R	02
ERROR	00000234	R	03
EXE\$ALONONPAGED	*****	X	03
EXE\$DEANONPAGED	*****	X	03
EXE\$INIT	*****	X	03
FILE_NAME	= 00000004		
IHD\$B_HDRBLKCNT	= 00000010		
IHD\$L_SYSVER	= 00000028		
IHD\$W_ACTIVOFF	= 00000002		
IHD\$W_IMGIDOFF	= 00000006		
IHD\$W_PATCHOFF	= 00000008		
IHI\$Q_LINKTIME	= 00000038		
IHP\$L_ECO1	= 00000000		
IOC\$GC_DEVLIST	*****	X	03
IOC\$GL_DPTLIST	*****	X	03
IOC\$REINITDRV	*****	X	03
IOC\$RELOC_DDT	*****	X	03
IOGEN\$CNTRL_INI	*****	X	03
IOGEN\$LOADDRIV	00000000	RG	03
IOGEN\$LOCK_IODB	*****	X	03
IOGEN\$READDRIV	*****	X	03
IOGEN\$UNLK_IODB	*****	X	03
LIB\$SIGNAL	*****	X	03
LINK_DPT	0000027F	R	03
LINK_TIME	0000000A	R	02
LOADER	000000E3	R	03
NOUNLOAD	000001A8	R	03
PR\$ IPL	= 00000012		
REINIT	000001B2	R	03
RET_ADDR	00000000	R	02
SS\$DEVACTIVE	= 000002C4		
SS\$INSFMEM	= 00000124		
SS\$NORMAL	= 00000001		

SYSS\$CMKRNL	*****	GX	03
SYSS\$DASSGN	*****	GX	03
SYSS\$DELTV	*****	GX	03
SYSS\$K_VERSION	*****	X	03
SYSG\$DRIVENAM	= 007CA023		
SYSG\$INVDPT	= 007C8032		
SYSG\$INVDPTINI	= 007C800A		
SYSG\$NOUNLOAD	= 007C8012		
SYSG\$SY\$VERDIF	= 007C806A		
UCB\$SL_DDT	= 00000088		
UCB\$SL_LINK	= 00000030		
UCB\$V_BSY	= 00000008		
UCB\$W_STS	= 00000064		
UNLOAD	00000262	R	03
UNLOAD_CHK	00000192	R	03
UNLOAD_ENBINT	00000249	R	03

! Psect synopsis !

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
\$ABSS	00000000 (0.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
NONPAGED_DATA	00000016 (22.)	02 (2.)	NOPIC USR CON REL LCL NOSHR NOEXE RD WRT NOVEC QUAD
NONPAGED_CODE	00000292 (658.)	03 (3.)	NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC LONG

! Performance indicators !

Phase	Page faults	CPU Time	Elapsed Time
Initialization	30	00:00:00.08	00:00:00.35
Command processing	122	00:00:00.65	00:00:02.33
Pass 1	361	00:00:12.02	00:00:25.02
Symbol table sort	0	00:00:01.98	00:00:04.39
Pass 2	85	00:00:02.22	00:00:04.47
Symbol table output	10	00:00:00.09	00:00:00.49
Psect synopsis output	2	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	612	00:00:17.07	00:00:37.08

The working set limit was 1350 pages.
69201 bytes (136 pages) of virtual memory were used to buffer the intermediate code.
There were 70 pages of symbol table space allocated to hold 1325 non-local and 22 local symbols.
416 source lines were read in Pass 1, producing 17 object records in Pass 2.
24 pages of virtual memory were used to define 23 macros.

! Macro library statistics !

Macro library name	Macros defined
-\$255\$DUA28:[BOOTS.OBJ]BOOTS.MLB;1	0
-\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	10
-\$255\$DUA28:[SYSLIB]STARLET.MLB;2	10
TOTALS (all libraries)	20

1441 GETS were required to define 20 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LISS:LOADDRIV/OBJ=OBJ\$:LOADDRIV MSRC\$:LOADDRIV/UPDATE=(ENH\$:LOADDRIV)+EXECMLS/LIB+LIBS:BOOTS.MLB/LIB

0038 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

