

ADDUNIT
Table of contents

H 11
- MSCP Library Routine for SET/SERVED

16-SEP-1984 02:04:01 VAX/VMS Macro V04-00

Page 0

(1)	28	PROGRAM DESCRIPTION
(1)	63	DECLARATIONS
(1)	89	MSCP\$ADDUNIT Routine
(1)	387	DEVTBL - Table of device specific parameters

ADD
Pse

PSE

\$AB
MSC

Pha

Ini
Com
Pas
Sym
Pas
Sym
Pse
Cro
Ass

The
106
The
440
23

Mac

_S2
-S2
-S2
TOT

223

The

MAC

```
0000 1 .TITLE ADDUNIT - MSCP L brary Routine for SET/SERVED
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 28 .SBTTL PROGRAM DESCRIPTION
0000 29 :++
0000 30 : FACILITY
0000 31 :
0000 32 : MSCP Server
0000 33 :
0000 34 : ABSTRACT
0000 35 :
0000 36 : This module is called from the SET utility to implement the
0000 37 : SET/SERVED command. It handles the interface to the MSCP Server.
0000 38 :
0000 39 : ENVIRONMENT
0000 40 :
0000 41 : NATIVE MODE, KERNEL MODE
0000 42 :
0000 43 : AUTHOR
0000 44 :
0000 45 : Kerbey T. Altmann, 24-Mar-1984
0000 46 :
0000 47 : MODIFIED BY
0000 48 :
0000 49 : V03-004 DAS0001 David Solomon 09-Jul-1984
0000 50 : Make MSCP$CODE psect read-only.
0000 51 :
0000 52 : V03-003 KTA3130 Kerbey T. Altmann 22-May-1984
0000 53 : Fix a bug in KTA3126.
0000 54 :
0000 55 : V03-002 KTA3126 Kerbey T. Altmann 23-Apr-1984
0000 56 : Fix problem with the DD device.
0000 57 :
0000 58 : V03-001 NPK3051 N. Kronenberg 17-Apr-1984
0000 59 : Change SSS_INDEVNAM to SSS_IVDEVNAM.
0000 60 :
0000 61 :--
```

```
0000 63      .SBTTL  DECLARATIONS
0000 64
0000 65  :
0000 66  :      SYMBOL DEFINITIONS
0000 67  :
0000 68
0000 69      .nocross
0000 70      $CDBDEF
0000 71      $DCDEF
0000 72      $DDBDEF
0000 73      $DEVDEF
0000 74      $DYNDEF
0000 75      $IPLDEF
0000 76      $MSCPDEF
0000 77      $PRDEF
0000 78      $$SDEF
0000 79      $UCBDEF
0000 80      $VCBDEF
0000 81  :
0000 82  :      MSCPDEF.MLB
0000 83  :
0000 84      $HQBDEF
0000 85      $MSUDEF
0000 86      $UQBDEF
0000 87      .cross
```

```

0000 89      .SBTTL  MSCP$ADDUNIT Routine
0000 90
0000 91      :++
0000 92
0000 93      : Functional description:
0000 94
0000 95      : This routine implements the functions needed by the SET DEVICE/SERVED
0000 96      : DCL command. It receives an argument from SET, validates it, allocates
0000 97      : and fills in the proper unit control block for the MSCP server.
0000 98
0000 99      : Calling sequence:
0000 100
0000 101      : KERNEL MODE, IPL = 2
0000 102      : IO database locked for write access
0000 103      : CALLS #3,G^MSCP$ADDUNIT
0000 104
0000 105      : Inputs:
0000 106
0000 107
00000004 0000 108      : UCBADDR = 4      ; Address of the device's UCB
00000008 0000 109      : OPTIONS = 8      ; Option flags (1 = /NOWRITE)
0000000C 0000 110      : CONTROL = 12     ; Controller letter, if present
0000 111
0000 112      : Outputs:
0000 113
0000 114      : R0 = Status
0000 115      : $$$_NORMAL - Success
0000 116      : $$$_DEVACTIVE - Unit is already served
0000 117      : $$$_DEVOFFLINE - MSCP server code not loaded
0000 118      : $$$_INSMFEM - Insufficient memory to allocate control blocks
0000 119      : $$$_DEVICEFULL - No space in MSCP server tables to add unit
0000 120      : $$$_IVDEVNAM - CONTROL letter is not A-G
0000 121
0000 122      :--
0000 123
00000000 0000 124      .PSECT  MSCP$CODE, LONG, NOWRT
0000 125
07FC 0000 126      .ENTRY  MSCP$ADDUNIT, ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10>
0002 127
58 00000000'9F  D0 0002 128      MOVL   @#SCS$GL_MSCP,R8      ; Pick up pointer to MSCP process
06 19 0009 129      BLSS   10$                  ; Valid system VA
50 0084 8F 3C 000B 130      MOVZWL #$$$_DEVOFFLINE,R0    ; Nothing there, error
04 0010 131      RET
0011 132
55 04 AC D0 0011 133 10$: MOVL   UCBADDR(AP),R5      ; Transfer UCB address
57 54 A5 3C 0015 134      MOVZWL UCBSW_UNIT(R5),R7    ; Pick up unit number
53 D4 0019 135      CLRL   R3                  ; Initialize
5A 8000 8F B0 001B 136      MOVW   #MSCP$M UF REPLC,R10 ; Default unit flags
53 41 A5 90 0020 137      MOVB   UCBSB_DEVTYPE(R5),R3 ; Pick up device type
53 81 8F 82 0024 138      BGEQ   20$                  ; A normal DIGITAL device
50 53 30 A1 002A 139      SUBB   #DT$ FD1,R3         ; Put in range of 0-n
002E 140      ADDW3  #<MSCP$K_EMS FD1-    ; Set correct mscp type
002E 141      @MSCP$V_EU_SUBC>,R3,R0
0030 142      BRB   50$                  ; Join common code
59 53 10 78 0030 143      ASHL   #16,R3,R9          ; Set device type
59 57 88 0034 144 20$: BISB   R7,R9                ; OR in unit number

```

```

51 0224'CF43 DE 0037 146 MOVAL W^DEVTBL[R3],R1 ; Pick up entry in table
    SA 02 A1 3C 003D 147 MOVZWL 2(R1),R10 ; and unit flags
    50 61 32 0041 148 CVTWL (R1),R0 ; Pick up part of emulated unit
    28 18 0044 149 BGEQ 50$ ; Nothing special
    0046 150 ;
    0046 151 ; Unit is a DSA disk. Now find out the controller model in order to
    0046 152 ; form the correct unit number.
    0046 153 ;
52 00BC C5 D0 0046 154 MOVL UCBSL_CDDDB(R5),R2 ; Pick up pointer to Clas. Driver
    21 13 004B 155 BEQL 50$ ; None, treat as normal
    52 26 A2 9A 004D 156 MOVZBL CDDBSB_CNTRLMDL(R2),R2 ; Get the controller model byte
    01 52 91 0051 157 CMPB R2,#MSCP$K_CM_HSC50 ; Is it HSC50?
    07 12 0054 158 BNEQ 30$ ; No
50 0200 8F 3C 0056 159 MOVZWL #MSCP$K_EMD_HSC- ; Yes, make it HSC disk
    005B 160 @MSCP$V_EU_CTYPE,R0
    11 11 005B 161 BRB 50$ ; Continue
    005D 162 ;
    04 52 91 005D 163 30$: CMPB R2,#MSCP$K_CM_EMULA ; Is it an emulated disk?
    07 12 0060 164 BNEQ 40$ ; No
50 0500 8F 3C 0062 165 MOVZWL #MSCP$K_EMD_EMUL- ; Yes, make it emulated disk
    0067 166 @MSCP$V_EU_CTYPE,R0
    05 11 0067 167 BRB 50$ ; Continue
    0069 168 ;
50 0100 8F 3C 0069 169 40$: MOVZWL #MSCP$K_EMD_UDA- ; Make it UDA disk
    006E 170 @MSCP$V_EU_CTYPE,R0
    57 50 A8 006E 171 50$: BISW2 R0,R7 ; Set in part of emulated unit
53 0C AC 9A 0071 172 MOVZBL CONTROL(AP),R3 ; Get special controller
    08 12 0075 173 BNEQ 70$ ; User supplied something
53 28 A5 D0 0077 174 60$: MOVL UCBSL_DDB(R5),R3 ; Point to DDB
53 17 A3 9A 007B 175 MOVZBL DDB$T_NAME+3(R3),R3 ; Pick up controller
53 40 8F 82 007F 176 70$: SUBB2 #^X40,R3 ; Make 'A'-->1
    07 53 91 0083 177 CMPB R3,#7 ; Is it in range 'A-G'
    06 1B 0086 178 BLEQU 80$ ; Yes, all is okay
50 0144 8F 3C 0088 179 MOVZWL #SS$_IVDEVNAM,R0 ; No, error out
    04 008D 180 RET
    008E 181 ;
59 08 08 53 F0 008E 182 80$: INSV R3,#8,#8,R9 ; Insert controller designator
    53 53 0C 78 0093 183 ASHL #MSCP$V_EU_DESIG,R3,R3 ; Shift into position
    57 53 C8 0097 184 BISL R3,R7 ; Complete unit number
56 57 04 00 EF 009A 185 EXTZV #0,#MSUS$_UNIT_SIZ,R7,R6; Slot index
    009F 186 ;
    009F 187 ; ADD a disk unit - first search the unit table in MSCP using a hashing
    009F 188 ; algorithm. If a entry is found, check for unit number match. If
    009F 189 ; there is a match, we have an error. If not, continue to look for
    009F 190 ; an empty slot. If there are none, error out with a full status.
    009F 191 ;
    009F 192 ; R5 --> UCB
    009F 193 ; R6 = Slot #
    009F 194 ; R7 = Unit #
    009F 195 ; R8 --> MSCP server
    009F 196 ; R9 = Possible unit id
    009F 197 ; R10 = Device unit flags
    009F 198 ;
    009F 199 SEARCHTABLE:
    009F 200 SETIPL B^60$ ; Synch access
52 0090 C846 D0 00A3 201 MOVL MSUS$_UNIT_VEC(R8)[R6],R2 ; Look for unit block in slot
    3D 13 00A9 202 BEQL 30$ ; Empty

```



```

28 A2 57 B1 00AB 203      CMPW   R7,UQBSW_UNIT(R2)      ; Hit, is it a match?
      3C 13 00AF 204      BEQL   20$                    ; Yes
      56 10 C0 00B1 205      ADDL   #MSUSK_MAX_UNIT,R6     ; Try for second probe
52 0090 C846 D0 00B4 206      MOVL   MSUSL_UNIT_VEC(R8)[R6],R2 ; Look for unit block in slot
      2C 13 00BA 207      BEQL   30$                    ; Empty
      28 A2 57 B1 00BC 208      CMPW   R7,UQBSW_UNIT(R2)      ; Hit, is it a match?
      1F 13 00C0 209      BEQL   20$                    ; Yes
      56 1F D0 00C2 210      MOVL   #<2*MSUSK_MAX_UNIT>-1,R6 ; Sequential search
52 0090 C846 D0 00C5 211 10$: MOVL   MSUSL_UNIT_VEC(R8)[R6],R2 ; Pick up unit
      1B 13 00CB 212      BEQL   30$                    ; Empty
      28 A2 57 B1 00CD 213      CMPW   R7,UQBSW_UNIT(R2)      ; Hit, is it a match?
      0E 13 00D1 214      BEQL   20$                    ; Yes
      10 56 D7 00D3 215      DECL   R6                    ; Loop thru
      56 56 D1 00D5 216      CMPL   R6,#MSUSK_MAX_UNIT ; Reached end?
      EB 18 00D8 217      BGEQ   10$                    ; No, loop for another probe
50 0850 8F 3C 00DA 218      MOVZWL #SS$_DEVICEFULL,R0     ; Yes, table is full
      19 11 00DF 219      BRB    40$
      00E1 220
50 02C4 8F 3C 00E1 221 20$: MOVZWL #SS$_DEACTIVE,R0     ; ERROR - existing device with unit
      12 11 00E6 222      BRB    40$                    ; Clean up
      00E8 223
      00E8 224 ; Found an empty slot. Allocate the UQB and fill it in.
      00E8 225
      51 9C 8F 9A 00E8 226 30$: MOVZBL #UQBS$_LEN,R1      ; Set size
      00000000 GF 16 00EC 227      JSB    G^EXES$ALONONPAGED ; Grab some pool
      09 50 C8 00F2 228      BLBS   R0,50$                ; All okay
50 0124 8F 3C 00F5 229      MOVZWL #SS$_INSMEM,R0       ; Sho error
      00FA 230 40$: SETIPL #IPL$_ASTDEL ; Lower IPL
      04 00FD 231      RET                    ; and out
      00FE 232
      C8 00FE 233 50$: BISL   #<DEVSM_CLU!- ; Set the 'Avail CLUSTER wide' bit
      00FF 234      DEVSM_SRV>,- ; and 'MSCP SERVER' bits
      00FF 235      UCBSL DEVCHAR2(R5)
3C A5 00000081 8F AE 0106 236      MNEGW #1,UQBS$_HOST_CNT(R2) ; Show no hosts
      OC A2 01 80 010A 237      MOVW   #MSCP$_ST_OFFLN,- ; Set STATUS to OFFLINE
      OE A2 010C 238      UQBSW STATUS(R2)
0090 C846 52 D0 010E 239      MOVL   R2,MSUSL_UNIT_VEC(R8)[R6]; Stick this UQB in MSCP table
      0114 240      SETIPL #IPL$_ASTDEL ; Lower IPL to ASTDEL
      04 11 0117 241      BRB    FILLUQB ; Skip around the data
      0119 242
      00000008 0119 243 60$: .LONG IPL$_SCS ; End of locked code
      011D 244
      011D 245 ; Now fill in the newly allocated UQB
      011D 246
      011D 247 FILLUQB:
      54 52 D0 011D 248      MOVL   R2,R4
      82 54 D0 0120 249      MOVL   R4,(R2)+ ; Set shadow que
      82 54 D0 0123 250      MOVL   R4,(R2)+ ; Ditto
      82 51 B0 0126 251      MOVW   R1,(R2)+ ; Set size field
      82 60 8F 9B 0129 252      MOVZBW #DYN$_SCS,(R2)+ ; Set type
      012D 253      ASSUME UQBS$_HOST_CNT EQ 12
      82 01 AE 012D 254      MNEGW #1,(R2)+ ; Set initial HOST CNT, AVL
      82 04 B0 0130 255      MOVW   #MSCP$_ST_AVLBL,(R2)+ ; Set STATUS to AVAILABLE
      0133 256      ASSUME UQBSW_MOLT_UNT EQ UQBS$_HOST_CNT+4
82 0101 8F 56 A5 0133 257      MULW3 R6,#^X101,(R2)+ ; Set unit
      82 5A B0 0139 258      MOVW   R10,(R2)+ ; Set unit flags
      013C 259 ;

```

```

013C 260 ; Check on the write protection status
013C 261 ;
05 38 19 E1 013C 262 BBC #DEV$V_SWL,-
013E 263 UCB$$_DEV$CHAR(R5),10$ ; Jump if not writelocked
00 FE 0D E2 0141 264 BBSS #MSCP$V_UF_WRTPH,-
00 FE A2 0143 265 -2(R2),10$ ; Set the bit
05 08 AC E9 0146 266 10$: BLBC OPTIONS(AP),20$ ; Check for /NOWRITE
00 FE 0C E2 014A 267 BBSS #MSCP$V_UF_WRTPS,-
00 FE A2 014C 268 -2(R2),20$ ; Set the bit
82 D4 014F 269 20$: CLRL (R2)+ ; Clear out reserved
0151 270 ;
0151 271 ; Set the unit id
0151 272 ;
0151 273 ASSUME UQB$$_UNIT_ID EQ UQB$$_MULT_UNIT+8
82 D4 0151 274 CLRL (R2)+ ; Clear low 32 bits of unit id
82 59 D0 0153 275 MOVL R9,(R2)+ ; upper 32 bits
FF A2 04 90 0156 276 MOVB #MSCP$K_CL_D144,-1(R2) ; Set MSCP class code
015A 277 ;
015A 278 ; Set the media id & unit/slot numbers
015A 279 ;
015A 280 ASSUME UQB$$_MEDIA_ID EQ 32
82 008C C5 D0 015A 281 ASSUME UQB$$_MEDIA_ID EQ UQB$$_UNIT_ID+8
82 D4 015F 282 MOVL UCB$$_MEDIA_ID(R5),(R2)+ ; Transfer it
0161 283 CLRL (R2)+ ; Clear SHDW
82 57 B0 0161 284 ASSUME UQB$$_UNIT EQ UQB$$_SHDW_STS+2
82 56 B0 0161 285 MOVW R7,(R2)+ ; Set in unit number
0164 286 MOVW R6,(R2)+ ; Set in slot number + clear byte
0167 287 ;
0167 288 ; Set the volume serial number
0167 289 ;
0167 290 ASSUME UQB$$_VOL_SER EQ UQB$$_SHDW_UNIT+8
50 34 A5 D0 0167 291 MOVL UCB$$_VCB(R5),R0 ; Pick up pointer to VCB
06 13 016B 292 BEQL 30$ ; None, set up fake
62 64 A0 D0 016D 293 MOVL VCB$$_SERIALNUM(R0),(R2) ; Set the serial
07 12 0171 294 BNEQ 40$ ; All okay
62 00001234 8F D0 0173 295 30$: MOVL #*X1234,(R2) ; Set a fake one
18 A4 82 D0 017A 296 40$: MOVL (R2)+,UQB$$_UNIT_ID(R4) ; Copy to unit id field
82 55 D0 017E 297 ASSUME UQB$$_UCB EQ UQB$$_VOL_SER+4
017E 298 MOVL R5,(R2)+ ; Store UCB away
82 D4 0181 299 ASSUME UQB$$_HOST_ONLN EQ UQB$$_UCB+4
0181 300 CLRL (R2)+ ; Clear HOST_ONLN
0183 301 ;
0183 302 ; Set name if present
0183 303 ;
0183 304 ASSUME UQB$$_NAME EQ UQB$$_HOST_ONLN+4
04 6C 91 0183 305 CMPB (AP),#4 ; Check for optional paramters
22 1F 0186 306 BLSSU 60$ ; Not present
3A BB 0188 307 PUSHR #*M<R1,R3,R4,R5> ; Present, save registers
50 10 AC D0 018A 308 MOVL 16(AP),R0 ; Symbol descriptor
51 60 3C 018E 309 MOVZWL (R0),R1 ; Size
0F 51 D1 0191 310 CMPL R1,#15 ; Maximize with 15
51 03 15 0194 311 BLEQ 50$
82 51 90 0199 313 50$: MOVL #15,R1
04 B0 51 2C 019C 314 MOVCS R1,@(R0),#32,#15,(R2) ; Copy device name
52 53 D0 01A3 315 MOVL R3,R2 ; Set pointer to end of name
3A BA 01A6 316 POPR #*M<R1,R3,R4,R5> ; Restore registers

```

```

05 11 01A8 317 BRB 70$
01AA 318
52 62 D4 01AA 319 60$: CLRL (R2) ; Show noname
10 C0 01AC 320 ADDL #16,R2 ; Skip text
01AF 321
01AF 322 ; Set maximum block/start if present
01AF 323
01AF 324 70$: ASSUME UQB$$_MAXBLOCK EQ UQB$$_NAME+16
06 6C 91 01AF 325 CMPB (AP),#6 ; Check for optional paramters
OF 1F 01B2 326 BLSSU 80$ ; Not present
50 14 AC D0 01B4 327 MOVL 20(AP),R0 ; Pick up maximum extend
09 13 01B8 328 BEQL 80$ ; None use defaults
82 82 50 D0 01BA 329 MOVL R0,(R2)+ ; Use specified for max
18 AC D0 01BD 330 MOVL 24(AP),(R2)+ ; Ditto for start
07 11 01C1 331 BRB 90$
01C3 332
82 00B0 C5 D0 01C3 333 80$: MOVL UCBS$_MAXBLOCK(R5),(R2)+; Use full disk
82 D4 01C8 334 CLRL (R2)+ ; Start at zero
01CA 335 ASSUME UQB$$_START_LBN EQ UQB$$_MAXBLOCK+4
01CA 336 ASSUME UQB$$_NUM_QOE EQ UQB$$_START_LBN+4
01CA 337 ; Clear counters & init list heads
01CA 338
82 7C 01CA 340 90$: CLRQ (R2)+ ; Counts
01CC 341 ASSUME UQB$$_FENCEL EQ UQB$$_NUM_QOE+8
82 7C 01CC 342 CLRQ (R2)+ ; Shadow fence
01CE 343 ASSUME UQB$$_CDRP_FL EQ UQB$$_FENCEL+8
62 62 DE 01CE 344 MOVAL (R2),(R2) ; Executing CDRP list head
82 82 DE 01D1 345 MOVAL (R2)+,(R2)+
01D4 346 ASSUME UQB$$_BLOCKQ EQ UQB$$_CDRP_FL+8
62 62 DE 01D4 347 MOVAL (R2),(R2) ; Blocked CDRP list head
82 82 DE 01D7 348 MOVAL (R2)+,(R2)+
01DA 349 ASSUME UQB$$_SHQ EQ UQB$$_BLOCKQ+8
62 62 DE 01DA 350 MOVAL (R2),(R2) ; Special holding CDRP list head
82 82 DE 01DD 351 MOVAL (R2)+,(R2)+
01E0 352 ASSUME UQB$$_SHDW_LOW EQ UQB$$_SHQ+8
82 D4 01E0 353 CLRL (R2)+ ; Shadow master/low
01E2 354 ASSUME UQB$$_NUM_BLK EQ UQB$$_SHDW_LOW+4
82 D4 01E2 355 CLRL (R2)+ ; Blocking counts
01E4 356 ASSUME UQB$$_UNITQ EQ UQB$$_NUM_BLK+4
82 7C 01E4 357 CLRQ (R2)+ ; Unit list head
82 7C 01E6 358 CLRQ (R2)+
01E8 359
01E8 360 ; Finished setting up UQB, now link it into the unit que
01E8 361
01E8 362 SETIPL B^IPL_HI ; Synch access to MSCP
53 0080 C8 DE 01EC 363 MOVAL MSU$_UQB_LIST(R8),R3 ; UQB list head
52 53 C0 01F1 364 MOVL R3,R2 ; Copy
53 04 A3 D0 01F4 365 100$: MOVL 4(R3),R3 ; Get next entry, backwards
53 52 D1 01F8 366 CMPL R2,R3 ; End of the line?
06 13 01FB 367 BEQL 110$ ; Yes
A8 A3 57 B1 01FD 368 CMPW R7,UQB$$_UNIT- -
0201 369 UQB$$_UNITQ(R3) ; Compare unit numbers
63 0080 F1 1F 0201 370 BLSSU 100$ ; Current is still higher than target
C4 0E 0203 371 110$: INSQUE UQB$$_UNITQ(R4),(R3) ; Insert
0208 372
0208 373 ; Go off to MSCP to ADD a disk unit

```

```
40 A8 B6 0208 374 i20$: INCW MSUSL_STATE(R8) ; Up the unit count
    54 DD 0208 375 PUSHL R4 ; Copy UQB pointer
    03 DD 020B 376 PUSHL #MSUSK_AC_ADD ; Opcode for MSCP action routines
    02 DD 020F 377 PUSHL #2 ; Only 2 parameters
5C 5E D0 0211 378 MOVL SP,AP ; Set address of parameter area
04 B848 16 0214 379 JSB @4(R8)[R8] ; Go to the routine
50 01 D0 0218 380 MOVL #1,R0 ; Set success
    04 021B 381 SETIPL #IPL$_ASTDEL ; Drop IPL back
    021E 382 RET
    021F 383
00000008 021F 384
    021F 385 IPL_HI: .LONG IPL$_SCS
```

```

0223 387 .SBTTL DEVTBL - Table of device specific parameters
0223 388
00008000 0223 389 $NREM = <MSCP$M_UF_REPLC>
00008080 0223 390 $REM = <MSCP$M_UF_REPLC!MSCP$M_UF_RMVBL>
0223 391
0223 392 :: NOTE: DSA disks must have a first word of -1 to trigger special
0223 393 :: logic in the unit number routine.
0223 394 ::
0223 395 .ALIGN WORD
0224 396
0224 397 MSCP$GA_DEVTBL::
0000 0000 0224 398 DEVTBL: .WORD 0,0
8080 0018 0228 399 .WORD MSCP$K_EMS_RK@MSCP$V_EU_SUBC, $REM : DTS_RK06 1
8080 0018 022C 400 .WORD MSCP$K_EMS_RK@MSCP$V_EU_SUBC, $REM : DTS_RK07 2
8080 0008 0230 401 .WORD MSCP$K_EMS_RP@MSCP$V_EU_SUBC, $REM : DTS_RP04 3
8080 0008 0234 402 .WORD MSCP$K_EMS_RP@MSCP$V_EU_SUBC, $REM : DTS_RP05 4
8080 0008 0238 403 .WORD MSCP$K_EMS_RP@MSCP$V_EU_SUBC, $REM : DTS_RP06 5
8080 0010 023C 404 .WORD MSCP$K_EMS_RM@MSCP$V_EU_SUBC, $REM : DTS_RM03 6
8000 0010 0240 405 .WORD MSCP$K_EMS_RM@MSCP$V_EU_SUBC, $NREM : DTS_RP07 7
8000 0010 0244 406 .WORD MSCP$K_EMS_RM@MSCP$V_EU_SUBC, $NREM : DTS_RP07HT 8
8080 0020 0248 407 .WORD MSCP$K_EMS_RL@MSCP$V_EU_SUBC, $REM : DTS_RL01 9
8080 0020 024C 408 .WORD MSCP$K_EMS_RL@MSCP$V_EU_SUBC, $REM : DTS_RL02 10
8080 0028 0250 409 .WORD MSCP$K_EMS_RX@MSCP$V_EU_SUBC, $REM : DTS_RX02 11
8080 0028 0254 410 .WORD MSCP$K_EMS_RX@MSCP$V_EU_SUBC, $REM : DTS_RX04 12
8000 0010 0258 411 .WORD MSCP$K_EMS_RM@MSCP$V_EU_SUBC, $NREM : DTS_RM80 13
8080 0000 025C 412 .WORD MSCP$K_EMS_CN$M@MSCP$V_EU_SUBC, $REM : DTS_TU58 14
8080 0010 0260 413 .WORD MSCP$K_EMS_RM@MSCP$V_EU_SUBC, $REM : DTS_RM05 15
8080 0028 0264 414 .WORD MSCP$K_EMS_RX@MSCP$V_EU_SUBC, $REM : DTS_RX01 16
0000 0000 0268 415 .WORD 0,0 : DTS_ML11 17
8080 0010 026C 416 .WORD MSCP$K_EMS_RM@MSCP$V_EU_SUBC, $REM : DTS_RB02 18
8000 0010 0270 417 .WORD MSCP$K_EMS_RM@MSCP$V_EU_SUBC, $NREM : DTS_RB80 19
8000 FFFF 0274 418 .WORD -1, $NREM : DTS_RA80 20
8000 FFFF 0278 419 .WORD -1, $NREM : DTS_RA81 21
8080 FFFF 027C 420 .WORD -1, $REM : DTS_RA60 22
8080 0300 0280 421 .WORD MSCP$K_EMD_AZT@MSCP$V_EU_CTYPE, $REM : DTS_RC25 23
8000 0300 0284 422 .WORD MSCP$K_EMD_AZT@MSCP$V_EU_CTYPE, $NREM : DTS_RCF25 24
8000 0400 0288 423 .WORD MSCP$K_EMD_RDRX@MSCP$V_EU_CTYPE, $NREM : DTS_RD51 25
8080 0400 028C 424 .WORD MSCP$K_EMD_RDRX@MSCP$V_EU_CTYPE, $REM : DTS_RX50 26
8000 0400 0290 425 .WORD MSCP$K_EMD_RDRX@MSCP$V_EU_CTYPE, $NREM : DTS_RD52 27
8000 0400 0294 426 .WORD MSCP$K_EMD_RDRX@MSCP$V_EU_CTYPE, $NREM : DTS_RD53 28
8080 0400 0298 427 .WORD MSCP$K_EMD_RDRX@MSCP$V_EU_CTYPE, $REM : DTS_RD26 29
8000 FFFF 029C 428 .WORD -1, $NREM : DTS_RA82 30
8080 0300 02A0 429 .WORD MSCP$K_EMD_AZT@MSCP$V_EU_CTYPE, $REM : DTS_RC26 31
8000 0300 02A4 430 .WORD MSCP$K_EMD_AZT@MSCP$V_EU_CTYPE, $NREM : DTS_RCF26 32
02A8 431 ::
02A8 432 :: Patch space for future drives
02A8 433 ::
0000 0000 02A8 434 .WORD 0,0 : DTS_ 33
0000 0000 02AC 435 .WORD 0,0 : DTS_ 34
0000 0000 02B0 436 .WORD 0,0 : DTS_ 35
0000 0000 02B4 437 .WORD 0,0 : DTS_ 36
0000 0000 02B8 438 .WORD 0,0
02BC 439
02BC 440 .END

```

ADDUNIT
Symbol table

F 12
- MSCP Library Routine for SET/SERVED

16-SEP-1984 02:04:01 VAX/VMS Macro V04-00
5-SEP-1984 02:07:54 [MSCP.SRC]ADDUNIT.MAR;1

\$NREM	=	00008000			MSCPSV_EU_DESIG	=	0000000C
\$REM	=	00008080			MSCPSV_EU_SUBC	=	00000003
CDDBSB_CNTRMLDL	=	00000026			MSCPSV_UF_WRTPH	=	0000000D
CONTRC	=	0000000C			MSCPSV_UF_WRTPS	=	0000000C
DDBST_NAME	=	00000014			MSUSB_CTRC_DAT	=	000001A0
DEVSM_CLU	=	00000001			MSUSB_SUBTYPE	=	0000000B
DEVSM_SRV	=	00000080			MSUSB_TYPE	=	0000000A
DEVSV_SWL	=	00000019			MSUSB_VEC	=	00000024
DEVTBC	=	00000224	R	02	MSUSC_LENGTH	=	000004CC
DTS_FD1	=	00000081			MSUSK_AC_ADD	=	00000003
DYN\$C_SCS	=	00000060			MSUSK_LENGTH	=	000004CC
EXESA[ONONPAGED	*****				MSUSK_MAX_HOST	=	00000010
FILLUQB	0000011D		R	X	MSUSK_MAX_UNIT	=	00000010
HQBSB_FLAG	0000001F				MSUSK_UNIT_SIZ	=	00000004
HQBSB_PORT	0000000B				MSUSL_ABORT	=	000001B4
HQBSB_TYPE	0000000A				MSUSL_AVAIL	=	000001D0
HQBSB_VEC	0000001E				MSUSL_BLKCOUNT	=	000002B0
HQBSK_LEN	00000048				MSUSL_BUFF_HEAD	=	00000050
HQBSL_LEN	00000048				MSUSL_CDRP_LIST	=	00000070
HQBSL_BLINK	00000004				MSUSL_CDRP_SAVE	=	0000006C
HQBSL_CDT	00000014				MSUSL_CDSV_SIZE	=	00000068
HQBSL_FLINK	00000000				MSUSL_D1	=	00000000
HQBSL_UNIT_ONLN	00000010				MSUSL_D2	=	00000004
HQBSQ_PENDING	00000030				MSUSL_FRACTION	=	0000005C
HQBSQ_TIME	00000020				MSUSL_GETCMD	=	000001B8
HQBSW_CNT_FLGS	00000018				MSUSL_GETUNIT	=	000001BC
HQBSW_HST_TMO	0000001A				MSUSL_HOST_DSBL	=	00000088
HQBSW_MAX_QUE	0000002A				MSUSL_HOST_VEC	=	00000110
HQBSW_NUM_QUE	00000028				MSUSL_MAX_BUF	=	00000054
HQBSW_REFC	0000002C				MSUSL_MEM_WAIT	=	00000078
HQBSW_SIZE	00000008				MSUSL_ONLYNE	=	000001D4
HQBSW_USE_FRAC	0000001C				MSUSL_OPCOUNT	=	000001B0
IPL\$ASTDEL	=	00000002			MSUSL_P1	=	0000000C
IPL\$SCS	=	00000008			MSUSL_P2	=	00000010
IPL_RI	0000021F		R	02	MSUSL_READ	=	00000234
MSCPSADDUNIT	00000000		RG	02	MSUSL_SETCTRL	=	000001C0
MSCPSGA_DEVTBL	00000224		RG	02	MSUSL_SETUNIT	=	000001D8
MSCPSK_CL_D144	=	00000004			MSUSL_SMALL	=	00000058
MSCPSK_CM_EMULA	=	00000004			MSUSL_STATE	=	00000040
MSCPSK_CM_HSC50	=	00000001			MSUSL_UNIT_VEC	=	00000090
MSCPSK_EMD_AZT	=	00000003			MSUSL_UGB_CIST	=	00000080
MSCPSK_EMD_EMUL	=	00000005			MSUSL_WRITE	=	00000238
MSCPSK_EMD_HSC	=	00000002			MSUSQ_CTRL_ID	=	00000190
MSCPSK_EMD_RDRX	=	00000004			MSUSQ_CTRL_INFO	=	00000198
MSCPSK_EMD_UDA	=	00000001			MSUSQ_HQBLIST	=	000004B4
MSCPSK_EMS_CNSL	=	00000000			MSUSQ_QUEUE	=	00000070
MSCPSK_EMS_FD1	=	00000006			MSUST_NAME	=	00000014
MSCPSK_EMS_RK	=	00000003			MSUSW_INI_HOST	=	0000004C
MSCPSK_EMS_RL	=	00000004			MSUSW_INI_PKT	=	00000046
MSCPSK_EMS_RM	=	00000002			MSUSW_MAX_PKT	=	0000004A
MSCPSK_EMS_RP	=	00000001			MSUSW_MAX_QUE	=	00000062
MSCPSK_EMS_RX	=	00000005			MSUSW_MIN_PKT	=	0000004A
MSCPSK_ST_AVLBL	=	00000004			MSUSW_NUM_HOST	=	0000004E
MSCPSK_ST_OFFLN	=	00000003			MSUSW_NUM_PKT	=	00000048
MSCPSM_UF_REPLC	=	00008000			MSUSW_NUM_QUE	=	00000060
MSCPSM_UF_RMVBL	=	00000080			MSUSW_PACKET	=	00000044
MSCPSV_EU_CTYPE	=	00000008			MSUSW_SIZE	=	00000008

ADDUNIT
Symbol table

- MSCP Library Routine for SET/SERVED

G 12

16-SEP-1984 02:04:01 VAX/VMS Macro V04-00
5-SEP-1984 02:07:54 [MSCP.SRC]ADDUNIT.MAR;1

Page 12
(1)

MS
V04

```

OPTIONS = 00000008
PRS IPL = 00000012
SCSSGL_MSCP ***** R X 02
SEARCHTABLE = 0000009F R X 02
SS$_DEACTIVE = 000002C4
SS$_DEVICEFULL = 00000850
SS$_DEVOFFLINE = 00000084
SS$_INS$MEM = 00000124
SS$_IVDEVNAM = 00000144
UCBSB_DEVTYPE = 00000041
UCBSL_Cddb = 0000008C
UCBSL_DDB = 00000028
UCBSL_DEVCHAR = 00000038
UCBSL_DEVCHAR2 = 0000003C
UCBSL_MAXBLOCK = 000000B0
UCBSL_MEDIA_ID = 0000008C
UCBSL_VCB = 00000034
UCBSW_UNIT = 00000054
UCBADDR = 00000004
UQBSB_CMD_PEND = 0000000D
UQBSB_FLAGS = 0000002B
UQBSB_HOST_CNT = 0000000C
UQBSB_SLOT = 0000002A
UQBSB_TYPE = 0000000A
UQBSL_LEN = 0000009C
UQBSK_LEN = 0000009C
UQBSL_CDRP_BL = 00000064
UQBSL_CDRP_FL = 00000060
UQBSL_CPY_CDRP = 00000088
UQBSL_FENCEH = 0000005C
UQBSL_FENCEL = 00000058
UQBSL_HOST_ONLN = 00000034
UQBSL_MAXBLOCK = 00000048
UQBSL_MEDIA_ID = 00000020
UQBSL_SHDW_BL = 00000004
UQBSL_SHDW_FL = 00000000
UQBSL_SHDW_LOW = 00000078
UQBSL_SHDW_MST = 00000078
UQBSL_START_LBN = 0000004C
UQBSL_UCB = 00000030
UQBSL_VOL_SER = 0000002C
UQBSQ_BLOCKQ = 00000068
UQBSQ_SHQ = 00000070
UQBSQ_UNITQ = 00000080
UQBSQ_UNIT_ID = 00000018
UQBST_NAME = 00000038
UQBSW_MAX_BLK = 0000007E
UQBSW_MAX_OPS = 00000056
UQBSW_MAX_QUE = 00000052
UQBSW_MULT_UNT = 00000010
UQBSW_NUM_BLK = 0000007C
UQBSW_NUM_OPS = 00000054
UQBSW_NUM_QUE = 00000050
UQBSW_SHDW_STS = 00000026
UQBSW_SHDW_UNT = 00000024
UQBSW_SIZE = 00000008
UQBSW_STATUS = 0000000E

```

```

UQBSW_UNIT = 00000028
UQBSW_UNT_FLGS = 00000012
VCBSL_SERIALNUM = 00000064

```

! Psect synopsis !

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
\$AB\$\$	000004CC (1228.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
MSCP\$CODE	000002BC (700.)	02 (2.)	NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC LONG

! Performance indicators !

Phase	Page faults	CPU Time	Elapsed Time
Initialization	17	00:00:00.06	00:00:01.96
Command processing	82	00:00:00.57	00:00:07.98
Pass 1	438	00:00:19.40	00:01:12.20
Symbol table sort	0	00:00:03.17	00:00:07.54
Pass 2	80	00:00:03.12	00:00:08.84
Symbol table output	21	00:00:00.20	00:00:01.52
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	640	00:00:26.54	00:01:40.07

The working set limit was 1500 pages.
106741 bytes (209 pages) of virtual memory were used to buffer the intermediate code.
There were 110 pages of symbol table space allocated to hold 2087 non-local and 26 local symbols.
440 source lines were read in Pass 1, producing 17 object records in Pass 2.
23 pages of virtual memory were used to define 22 macros.

! Macro library statistics !

Macro library name	Macros defined
-\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	8
-\$255\$DUA28:[MSCP.OBJ]MSCPDEF.MLB;1	3
-\$255\$DUA28:[SYSLIB]STARLET.MLB;2	8
TOTALS (all libraries)	19

2233 GEIS were required to define 19 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:ADDUNIT/OBJ=OBJ\$:ADDUNIT MSRC\$:ADDUNIT/UPDATE=(ENH\$:ADDUNIT)+LIB\$:MSCPDEF/LIB+EXECMLS/LIB

0249 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

This image displays a grid of 150 small terminal window screenshots, arranged in approximately 10 rows and 15 columns. Each window shows a different VAX/VMS command-line interface, likely generated by the 'D >' command which dumps all active windows. The windows contain various system utilities, diagnostic tools, and data displays. Some of the more prominent titles and content visible in the windows include:

- MSCP**: Multiple instances of the MSCP (Mass Storage Control Program) utility.
- MSCP MAP**: A display showing the mapping of mass storage devices.
- ADDUNIT LIS**: A list of added units.
- MSCP LIS**: A list of MSCP-related information.
- MPWATT LIS**: A list of MPWATT-related information.
- MPTIMER LIS**: A list of MPTIMER-related information.
- XDELTA LIS**: A list of XDELTA-related information.
- MSCPDEF MAR**: A display related to MSCP definitions.

The screenshots are densely packed, and each window shows a different stage or aspect of system operation, including command prompts, error messages, and data listings. The overall appearance is that of a complex, multi-tasking operating system environment.