

[illegible]

MSC

\$C
\$C
\$C
\$C
\$C
\$C
\$C

\$C
\$C
\$C

\$C
\$C

\$C
\$C

```
MM      MM      SSSSSSSS      CCCCCCCC      P P P P P P P      SSSSSSSS      HH      HH      000000      WW      WW
MM      MM      SSSSSSSS      CCCCCCCC      P P P P P P P      SSSSSSSS      HH      HH      000000      WW      WW
MMM     MMM     SS      CC      PP      PP      SS      HH      HH      00      00      WW      WW
MMM     MMM     SS      CC      PP      PP      SS      HH      HH      00      00      WW      WW
MM      MM      SS      CC      PP      PP      SS      HH      HH      00      00      WW      WW
MM      MM      SSSSSS      CC      P P P P P P P      SSSSSS      H H H H H H H H H H      00      00      WW      WW
MM      MM      SSSSSS      CC      P P P P P P P      SSSSSS      H H H H H H H H H H      00      00      WW      WW
MM      MM      SS      CC      PP      PP      SS      HH      HH      00      00      WW      WW
MM      MM      SS      CC      PP      PP      SS      HH      HH      00      00      WW      WW
MM      MM      SS      CC      PP      PP      SS      HH      HH      00      00      WW      WW
MM      MM      SSSSSSSS      CCCCCCCC      PP      SSSSSSSS      HH      HH      00      00      WWW      WWW
MM      MM      SSSSSSSS      CCCCCCCC      PP      SSSSSSSS      HH      HH      00      00      WWW      WWW
MM      MM      SSSSSSSS      CCCCCCCC      PP      SSSSSSSS      HH      HH      000000      WW      WW
MM      MM      SSSSSSSS      CCCCCCCC      PP      SSSSSSSS      HH      HH      000000      WW      WW
                                     ....
                                     ....
                                     ....
                                     ....
```

```
LL      I I I I I      SSSSSSSS
LL      I I I I I      SSSSSSSS
LL      I I      SS
LL      I I      SS
LL      I I      SS
LL      I I      SS
LL      I I      SSSSSS
LL      I I      SSSSSS
LL      I I      SS
LL      I I      SS
LL      I I      SS
LL      I I      SS
LLLLLLLLLLLL      I I I I I      SSSSSSSS
LLLLLLLLLLLL      I I I I I      SSSSSSSS
```


(1)	28	PROGRAM DESCRIPTION
(1)	68	DECLARATIONS
(1)	111	MSCP\$GET_UNIT - Return information on units
(1)	196	MSCP\$GET_HOST - Return information on hosts
(1)	261	MSCP\$GET_RESO - Return information on resources
(1)	311	MSCP\$GET_COU1 - Return information request counts
(1)	360	MSCP\$GET_COU2 - Return the count from MSCP operations

```

0000 1      .TITLE  MSCP$SHOW - MSCP Server Information
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 MSU Utility
0000 33
0000 34 ABSTRACT
0000 35
0000 36 This module contains the routines to gather and pass
0000 37 information to SHOW$MSCP which displays this information
0000 38 about the MSCP server.
0000 39
0000 40 ENVIRONMENT
0000 41
0000 42 NATIVE MODE, USER MODE
0000 43
0000 44 AUTHOR
0000 45
0000 46 Anne E. Warner, 10-Apr-1984
0000 47
0000 48 MODIFIED BY
0000 49
0000 50 V03-004 AEW0004 Anne E. Warner 02-Aug-1984
0000 51 Change MSUSW_MIN_PKT to MSUSW_INI_PKT in GET_RESO to
0000 52 correct mistake in the display
0000 53
0000 54 V03-003 AEW0003 Anne E. Warner 10-Jul-1984
0000 55 Make changes within the routines for the reformat of
0000 56 SHOW MSCP which is now SHOW DEVICE/SERVED.
0000 57
0000 58 V03-002 AEW0002 Anne E. Warner 24-May-1984
0000 59 Change where the address of the 'DUMMY1' stack is moved
0000 60 into R10 to avoid Access Violating on certain disks. This
0000 61 change is made in the module MSCP$GET_UNIT.
0000 62
0000 63 V03-001 AEW0001 Anne E. Warner 27-Apr-1984
0000 64 Made bug fixes so that this code will work with
0000 65 SHOW$MSCP.
0000 66 :--

```

```
0000 68 .SBTTL DECLARATIONS
0000 69
0000 70 :
0000 71 : SYMBOL DEFINITIONS
0000 72 :
0000 73 :
0000 74 .nocross
0000 75 $CDTDEF
0000 76 $IPLDEF
0000 77 $MSCPDEF
0000 78 $PBDEF
0000 79 $PRDEF
0000 80 $SBDEF
0000 81 $SSDEF
0000 82 :
0000 83 : MSCPDEF.MLB
0000 84 :
0000 85 $HQBDEF
0000 86 $MSUDEF
0000 87 $UQBDEF
0000 88 .cross
0000 89
0000 90 .DEFAULT DISPLACEMENT LONG
0000 91
0000 92
0000 93 :
0000 94 : STORAGE DEFINITIONS
0000 95 :
0000 96
00000000 97 .PSECT SHOW$RWDATA,NOEXE,WRT,LONG
0000 98
0000 99 STATUS_ARY:
45 4E 49 4C 4E 4F 20 20 0000 100 .ASCII / ONLINE/ ; ONLINE
20 20 20 20 20 20 20 20 0008 101 .ASCII / /
20 20 20 20 20 20 20 20 0010 102 .ASCII / /
45 4E 49 4C 46 46 4F 20 0018 103 .ASCII / OFFLINE/ ; OFFLINE
4C 49 41 56 41 20 20 20 0020 104 .ASCII / AVAIL/ ; AVAILABLE
0028 105
54 52 57 4F 4E 2F 0028 106 NOWRT: .ASCII ?/NOWRT?
00000006 002E 107 NOWRT_LEN=-NOWRT
002E 108
00000036 002E 109 TIME: .BLKL 2
```



```
0036 111 .SBTTL MSCP$GET_UNIT - Return information on units
0036 112
00000000 113 .PSECT SHOW$CODE,EXE,WRT,LONG
0000 114
0000 115 .ENABLE LSB
0000 116
07FC 0000 117 .ENTRY MSCP$GET_UNIT ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10>
0002 118
58 50 0084 8F 3C 0002 119 MOVZWL #SS$ DEVOFFLINE,R0 ; Set for failure to find MSCP
00000000'9F D0 0007 120 MOVL @#SC$GL_MSCP,R8 ; Pick up the pointer to MSCP area
01 19 000E 121 BLSS 10$ ; All okay
04 0010 122 RET ; Return in disgrace
0011 123
0C BC 20 9A 0011 124 10$: MOVZBL #2*MSUSK_MAX_UNIT,@12(AP)
59 00000100'EF 9E 0015 125 MOVAB DEVNAM,R9 ; Save buffer address in non-volatile
56 08 AC D0 001C 126 MOVL 8(AP),R6 ; Grab unit slot number
57 0090 C846 D0 0020 127 20$: MOVL MSUSL_UNIT_VEC(R8)[R6],R7 ; UQB pointer
05 12 0026 128 BNEQ 21$ ; Found one
5A D4 0028 129 CLRL R10
008F 31 002A 130 BRW 60$ ; None
002D 131
002D 132 21$: SETIPL 70$ ; Synch access
B3 0034 133 BITW #<MSCP$M_UF_WRTPS!- ; Check write protect status
0035 134 MSCP$M_UF_WRTPH>,-
0035 135 UQBSW_UNIT_FLGS(R7)
12 A7 3000 8F 02 13 003A 136 BEQL 22$ ; Not write protected
003C 137 ; PUSHAL NOWRT ; Set up write protect message
003C 138 ; PUSHL #NOWRT_LEN
02 11 003C 139 BRB 23$
003E 140
7E 7C 003E 141 22$: CLRQ -(SP) ; Null message
0040 142 ;
0040 143 ; Process actual device name from UCB
0040 144 ;
50 10 3C 0040 145 23$: MOVZWL #16,R0 ; Max device name size
51 59 D0 0043 146 MOVL R9,R1 ; Use scratch area on stack
5A 00FC'CF DE 0046 147 MOVAL W^DUMMY1,R10 ; Set up register to mimic a stack
54 D4 004B 148 CLRL R4 ; Want nodename$devname
55 30 A7 D0 004D 149 MOVL UQBSL_UCB(R7),R5 ; Point to UCB
00000000'9F 16 0051 150 JSB @#IOC$CVT_DEVNAM ; Do it
55 51 D0 0057 151 MOVL R1,R5 ; Transfer length
69 55 24 3A 005A 152 LOCC #^A/$/,R5,(R9) ; Find the end of nodename
14 13 005E 153 BEQL 40$ ; None
00000110'EF 01 A1 9E 0060 154 MOVAB 1(R1),DEV_NAME ; Start of devname
00000114'EF 50 02 C3 0068 155 SUBL3 #2,R0,DEV_CNT ; Size
11 11 0070 156 BRB 50$
0072 157
AC 11 0072 158 39$: BRB 20$ ; BRB cannot fully reach
0074 159
00000110'EF 59 D0 0074 160 40$: MOVL R9,DEV_NAME ; Start of name
00000114'EF 55 01 C3 007B 161 SUBL3 #1,R5,DEV_CNT ; Set length
0083 162 ;
0083 163 ; Set up rest of line
0083 164 ;
52 0C A7 98 0083 165 50$: CVTBL UQBSB_HOST_CNT(R7),R2 ; Number of host online
7A 52 01 C1 0087 166 ADDL3 #1,R2,-(R10) ; Correct for offset
7A 52 A7 3C 008B 167 MOVZWL UQBSW_MAX_QUE(R7),-(R10) ; Que parameters
```

```
7A 50 A7 3C 008F 168      MOVZWL  UQBSW_NUM QUE(R7),-(R10)      ; Ditto
7A 48 A7 D0 0093 169      MOVL     UQBSL_MAXBLOCK(R7),-(R10)    ; Max block (size)
    05 00 EF 0097 170      EXTZV   #MSCP$V ST_MASK,#MSCP$S-ST_MASK,-
52 0E A7 009A 171      UQBSW_STATUS(R7),R2      ; Get the device status
    009D 172      SETIPL  #0      ; Drop IPL - finished with UQB
7A 52 02 C4 00A0 173      MULL2   #2,R2      ; Compute correct offset
7A 00000000'EF42 DE 00A3 174      MOVAL  STATUS_ARY[R2],-(R10)    ; ASCII text for status
    7A 08 D0 00AB 175      MOVL     #8,-(R10)      ; Length of text
7A 00000110'EF D0 00AE 176      MOVL     DEV_NAME,-(R10)      ; Move stored device name to stack
7A 00000114'EF D0 00B5 177      MOVL     DEV_CNT,-(R10)      ; Move device name count to stack
    04 BC 5A D0 00BC 178 60$: MOVL     R10,R4(AP)      ; Provide top of argument stack to calling r
    50 01 3C 00C0 179      MOVZWL  #SS$_NORMAL,R0
    04 00C3 180 65$: RET
    00C4 181
    00C4 182      ; Argument list - placed here since it must be filled in at IPL 8.
    00C4 183
    00C4 184      .ALIGN  LONG
000000FC 00C4 185      .BLKL  14
00000000 00FC 186 DUMMY1: .LONG  0
00000110 0100 187 DEVNAM: .BLKB  16
    0110 188 DEV_NAME:
00000000 0110 189      .LONG  0
00000000 0114 190 DEV_CNT: .LONG  0
    0118 191
00000008 0118 192 70$: .LONG  IPL$_SCS      ; End of locked down code
    011C 193
    011C 194      .DISABLE LSB
```



```
011C 196 .SBTTL MSCP$GET_HOST - Return information on hosts
011C 197
011C 198 .ENABLE LSB
011C 199
053C 011C 200 .ENTRY MSCP$GET_HOST ^M<R2,R3,R4,R5,R8,R10>
011E 201
58 50 0084 8F 3C 011E 202 MOVZWL #SS$ DEVOFFLINE,R0 ; Set for failure to find MSCP
00000000'9F D0 0123 203 MOVL @#SC$GL_MSCP,R8 ; Pick up the pointer to MSCP area
01 19 012A 204 BLSS 10$ ; All okay
04 012C 205 RET ; Return in disgrace
012D 206
0C BC 20 9A 012D 207 10$: MOVZBL #2*MSUSK_MAX_HOST,@12(AP) ; Return the maximum host count
55 08 AC D0 0131 208 MOVL 8(AP),R5 ; Pick up the host slot
54 0110 C845 D0 0135 209 20$: MOVL MSUSL_HOST_VEC(R8)[R5],R4 ; HQB pointer
05 12 013B 210 BNEQ 21$ ; Found
5A D4 013D 211 CLRL R10 ; Show no host
006A 31 013F 212 BRW 60$ ; None
0142 213
51 7C 0142 214 21$: CLRQ R1 ; Set up
0144 215 SETIPL 70$ ; Synch access
50 14 A4 D0 014B 216 MOVL HQBSL_CDT(R4),R0 ; Pick up CDT pointer
15 13 014F 217 BEQL 22$ ; Ooops!
50 1C A0 D0 0151 218 MOVL CDT$ _PB(R0),R0 ; Pick up PB pointer
0F 13 0155 219 BEQL 22$ ; Ooops!
50 30 A0 D0 0157 220 MOVL PB$ _SBLINK(R0),R0 ; Pick up SB pointer
09 13 015B 221 BEQL 22$
000001CC'EF 44 A0 7DFD 015D 222 MOVO SB$T_NODENAME(R0),NNAME ; Finally we get the host name
53 10 A4 D0 0166 223 22$: MOVL HQBS[_UNIT_ONLN(R4)],R3 ; Bitmap of units
016A 224 SETIPL #0 ; Finished
5A 01C8'CF DE 016D 225 MOVAL W^DUMMY2,R10 ; Set up register to mimic stack
50 D4 0172 226 CLRL R0 ; Set up
52 1F D0 0174 227 MOVL #31,R2
02 53 E9 0177 228 25$: BLBC R3,30$ ; None
50 D6 017A 229 INCL R0
53 53 01 9C 017C 230 30$: ROTL #1,R3,R3 ; Check next bit
F4 52 F5 0180 231 SOBGTR R2,25$
7A 50 D0 0183 232 MOVL R0,-(R10) ; number of devices on host
0186 233
: At high IPL pick up the queue numbers
0186 234
0186 235
0186 236 SETIPL B^70$ ; Synch access
7A 2A A4 3C 018A 237 MOVZWL HQBSW_MAX_QUE(R4),-(R10) ; Pick up value from data block
7A 28 A4 3C 018E 238 MOVZWL HQBSW_NUM_QUE(R4),-(R10) ; Ditto
0192 239 SETIPL #0 ; Finished with block
0000002E'EF 20 A4 7D 0195 240 MOVQ HQBSQ_TIME(R4),TIME ; Pick up host connect time
7A 0000002E'EF 9E 019D 241 MOVAB TIME,-(R10) ; Get the time
7A 000001CC'EF 7EFD 01A4 242 MOVAO NNAME,-(R10) ; Move in the host name from above
04 BC 5A D0 01AC 243 60$: MOVL R10,@4(AP) ; Copy the arg list pointer
50 01 3C 01B0 244 MOVZWL #SS$_NORMAL,R0
04 01B3 245 65$: RET
01B4 246
: Argument List - placed here since it must be filled in at IPL 8.
01B4 247
01B4 248
01B4 249 .ALIGN LONG
000001C8 01B4 250 .BLKL 5
00000000 01C8 251 DUMMY2: .LONG 0
000001DC 01CC 252 NNAME: .BLKL 4
```

Page 7
(1)

```

00000000 01DC 253 CDT: .LONG 0
00000000 01E0 254 PB: .LONG 0
00000000 01E4 255 SB: .LONG 0
          01E8 256
00000008 01E8 257 70$: .LONG IPL$_SCS
          01EC 258
          01EC 259 .DISABLE LSB

```



```
01EC 261 .SBTTL MSCP$GET_RESO - Return information on resources
01EC 262
01EC 263 .ENABLE LSB
01EC 264
0538 01EC 265 .ENTRY MSCP$GET_RESO ^M<R3,R4,R5,R8,R10>
01EE 266
58 50 0084 8F 3C 01EE 267 MOVZWL #SS$ DEVOFFLINE,R0 ; Set for failure to find MSCP
00000000 9F D0 01F3 268 MOVL @#SC$SGL_MSCP,R8 ; Pick up the pointer to MSCP area
01 19 01FA 269 BLSS 10$ ; All okay
04 01FC 270 RET ; Return in disgrace
01FD 271
5A 0298 CF DE 01FD 272 10$: MOVAL W^DUMMY3,R10 ; Set up register to mimic stack
0202 273 SETIPL 40$ ; Synch access to data blocks
7A 62 A8 3C 0209 274 MOVZWL MSUSW_MAX_QUE(R8),-(R10); Grab statistics
7A 60 A8 3C 020D 275 MOVZWL MSUSW_NUM_QUE(R8),-(R10);
7A 48 A8 3C 0211 276 MOVZWL MSUSW_NUM_PKT(R8),-(R10);
7A 46 A8 3C 0215 277 MOVZWL MSUSW_INI_PKT(R8),-(R10);
7A 5C A8 D0 0219 278 MOVL MSUSL_FRACTION(R8),-(R10);
6A 00000200 8F C6 021D 279 DIVL2 #512,(R10) ; TEMP DIVIDE UNTIL SYSGEN PARAMS CHANGE
7A 58 A8 D0 0224 280 MOVL MSUSL_SMALL(R8),-(R10) ;
6A 00000200 8F C6 0228 281 DIVL2 #512,(R10) ; TEMP DIVIDE
53 7C 022F 282 CLRQ R3 ; Zero temp accumulators
55 50 A8 D0 0231 283 MOVL MSUSL_BUFF_HEAD(R8),R5 ; Pick up head of buffer pool
0B 13 0235 284 BEQL 30$ ; None
54 04 A5 C0 0237 285 20$: ADDL 4(R5),R4 ; Yes, add the size
53 D6 023B 286 INCL R3 ; Incr number of buffers
55 65 D0 023D 287 MOVL (R5),R5 ; Next block
F5 12 0240 288 BNEQ 20$ ; Okay, go one
55 54 A8 D0 0242 289 30$: MOVL MSUSL_MAX_BUF(R8),R5 ; Original size
0246 290 SETIPL #0 ; Finished with the data block
7A 55 54 C3 0249 291 SUBL3 R4,R5, -(R10) ; In use
6A 00000200 8F C6 024D 292 DIVL2 #512,(R10) ; TEMP DIVIDE
7A 54 D0 0254 293 MOVL R4, -(R10) ; Free
6A 00000200 8F C6 0257 294 DIVL2 #512,(R10) ; TEMP DIVIDE
7A 55 D0 025E 295 MOVL R5, -(R10) ; Total
6A 00000200 8F C6 0261 296 DIVL2 #512,(R10) ; TEMP DIVIDE
04 BC 5A D0 0268 297 MOVL R10,@4(AP) ; Pass back where start of info is
50 01 3C 026C 298 MOVZWL #SS$_NORMAL,R0
04 026F 299 RET
0270 300 ;
0270 301 ; Argument list - placed here since it must be filled in at IPL 8.
0270 302 ;
0270 303 .ALIGN LONG
00000298 0270 304 .BLKL 10
0298 305 DUMMY3:
0298 306
00000008 0298 307 40$: .LONG IPL$_SCS
029C 308
029C 309 .DISABLE LSB
```



```
029C 311 .SBTTL MSCP$GET_COU1 - Return information request counts
029C 312
029C 313 .ENABLE LSB
029C 314
06F0 029C 315 .ENTRY MSCP$GET_COU1 ^M<R4,R5,R6,R7,R9,R10>
029E 316
59 50 0084 8F 3C 029E 317 MOVZWL #SS$ DEVOFFLINE,R0 ; Set for failure to find MSCP
00000000 9F D0 02A3 318 MOVL @#SC$GL_MSCP,R9 ; Pick up the pointer to MSCP area
01 19 02AA 319 BLSS 10$ ; All okay
04 02AC 320 RET ; Return in disgrace
02AD 321
02AD 322 10$: SETIPL 70$ ; Synch access
5A 0330 CF DE 02B4 323 MOVAL W^DUMMY4,R10 ; Set up register to mimic stack
7A D4 02B9 324 CLRL -(R10)
7A D4 02BB 325 CLRL -(R10) ; Need two zeros on bottom of stack
6A D4 02BD 326 CLRL (R10) ; for null arguments in display
59 04B0 C9 9E 02BF 327 MOVAB MSU$L,BLKCOUNT+512(R9),R9 ; Point to the count
000002EE EF 9E 02C4 328 MOVAB REP_TBL,R5 ; Point to table of repetitions
55 57 85 9A 02CB 329 30$: MOVZBL (R5)+,R7
13 02CE 330 BEQL 60$ ; Ended
56 85 9A 02D0 331 MOVZBL (R5)+,R6
54 56 D0 02D3 332 40$: MOVL R6,R4 ; Copy
7A D4 02D6 333 CLRL -(R10) ; Init counter
6A 79 C0 02D8 334 50$: ADDL -(R9),(R10) ; Tally up the counts
FA 54 F5 02DB 335 SOBGTR R4,50$ ; Loop until done
F2 57 F5 02DE 336 SOBGTR R7,40$
E8 11 02E1 337 BRB 30$ ; Look for next entry
02E3 338
02E3 339 60$: SETIPL #0 ; Done
04 BC 5A D0 02E6 340 MOVL R10,@4(AP) ; Beginning of arglist needed for output
50 01 3C 02EA 341 MOVZWL #SS$_NORMAL,R0
04 02ED 342 RET
02EE 343
02EE 344 ; Tables & argument list - placed here since they must be accessed at IPL 8.
02EE 345
02EE 346 REP_TBL:
18 01 02EE 347 .BYTE 1,24 ; 105-127
10 04 02F0 348 .BYTE 4,16 ; 40-55,...88-104
08 05 02F2 349 .BYTE 5,8 ; 0-7,...32-39
0000 02F4 350 .WORD 0
02F6 351
02F6 352 .ALIGN LONG
00000330 02F8 353 .BLKL 14
00000000 0330 354 DUMMY4: .LONG
0334 355
00000008 0334 356 70$: .LONG IPL$_SCS
0338 357
0338 358 .DISABLE LSB
```



```
0338 360 .SBTTL MSCP$GET_COU2 - Return the count from MSCP operations
0338 361
0338 362 .ENABLE LSB
0338 363
0700 0338 364 .ENTRY MSCP$GET_COU2 ^M<R8,R9,R10>
033A 365
58 50 0084 8F 3C 033A 366 MOVZWL #SS$ DEVOFFLINE,R0 ; Set for failure to find MSCP
00000000'9F D0 033F 367 MOVL @#SC$SGL_MSCP,R8 ; Pick up the pointer to MSCP area
01 19 0346 368 BLSS 10$ ; All okay
04 04 0348 369 RET ; Return in disgrace
0349 370
59 01B0 C8 9E 0349 371 10$: MOVAB MSU$L_OPCOUNT(R8),R9 ; Point at counts
00 11 034E 372 BRB 21$
0350 373
5A 000003F4'EF DE 0350 374 21$: SETIPL 70$ ; Synch access to data block
0357 375 MOVAL DUMMY5,R10 ; Set up register to mimic stack
035E 376
035E 377 : Get all operations by category and build 'stack'
035E 378 :
035E 379 :
7A 01B0 C8 D0 035E 379 MOVL MSU$L_OPCOUNT(R8),-(R10); TOTAL
7A 0238 C8 D0 0363 380 MOVL MSU$L_WRITE(R8),-(R10); WRITE
7A 01D4 C8 D0 0368 381 MOVL MSU$L_ONLINE(R8),-(R10); ONLINE
7A 0080 C8 D0 036D 382 MOVL 32*4(R9),-(R10); CMP HST DAT
7A 01D8 C8 D0 0372 383 MOVL MSU$L_SETUNIT(R8),-(R10); SET UNT CHR
7A 01BC C8 D0 0377 384 MOVL MSU$L_GETUNIT(R8),-(R10); GET UNT STS
7A 44 A9 D0 037C 385 MOVL 17*4(R9),-(R10); CMP CTL DAT
7A 01C0 C8 D0 0380 386 MOVL MSU$L_SETCTRL(R8),-(R10); SET CTL CHR
7A 01B8 C8 D0 0385 387 MOVL MSU$L_GETCMD(R8),-(R10); GET CMD STS
7A 01D0 C8 D0 038A 388 MOVL MSU$L_AVAIL(R8),-(R10); AVAILABLE
7A 50 A9 D0 038F 389 MOVL 20*4(R9),-(R10); REPLACE
7A 4C A9 D0 0393 390 MOVL 19*4(R9),-(R10); FLUSH
7A 40 A9 D0 0397 391 MOVL 16*4(R9),-(R10); ACCESS
7A 0234 C8 D0 039B 392 MOVL MSU$L_READ(R8),-(R10); READ
7A 48 A9 D0 03A0 393 MOVL 18*4(R9),-(R10); ERASE
7A 01B4 C8 D0 03A4 394 MOVL MSU$L_ABORT(R8),-(R10); ABORT
04 BC 5A D0 03A9 395 SETIPL #0 ; Finished with block
50 01 3C 03AC 396 MOVL R10,@4(AP)
04 04 03B0 397 MOVZWL #SS$_NORMAL,R0
03B3 398 RET
03B4 399
03B4 400 : Argument list - placed here since it must be filled in at IPL 8.
03B4 401 :
000003F4 03B4 402 .ALIGN LONG
03B4 403 .BLKL 16
03F4 404 DUMMY5:
03F4 405
00000008 03F4 406 70$: .LONG IPL$_SCS
03F8 407
03F8 408 .DISABLE LSB
03F8 409
03F8 410 .END
```

MSCP\$SHOW
Symbol table

- MSCP Server Information

E 5

16-SEP-1984 02:05:50 VAX/VMS Macro V04-00
5-SEP-1984 02:08:39 [MSCP.SRC]MSCP\$SHOW.MAR;1

Page 11
(1)

CDT	= 000001DC	R	03	MSUSL_FRACTION	0000005C		
CDT\$L_PB	= 0000001C			MSUSL_GETCMD	000001B8		
DEVNAM	00000100	R	03	MSUSL_GETUNIT	000001BC		
DEV_CNT	00000114	R	03	MSUSL_HOST_DSBL	00000088		
DEV_NAME	00000110	R	03	MSUSL_HOST_VEC	00000110		
DUMMY1	000000FC	R	03	MSUSL_MAX_BUF	00000054		
DUMMY2	000001C8	R	03	MSUSL_MEM_WAIT	00000078		
DUMMY3	00000298	R	03	MSUSL_ONLINE	000001D4		
DUMMY4	00000330	R	03	MSUSL_OPCOUNT	000001B0		
DUMMY5	000003F4	R	03	MSUSL_P1	0000000C		
HQBSB_FLAG	0000001F			MSUSL_P2	00000010		
HQBSB_PORT	0000000B			MSUSL_READ	00000234		
HQBSB_TYPE	0000000A			MSUSL_SETCTRL	000001C0		
HQBSB_VEC	0000001E			MSUSL_SETUNIT	000001D8		
HQBSK_LEN	00000048			MSUSL_SMALL	00000058		
HQBSK_LEN	00000048			MSUSL_STATE	00000040		
HQBSL_BLINK	00000004			MSUSL_UNIT_VEC	00000090		
HQBSL_CDT	00000014			MSUSL_UQB_LIST	00000080		
HQBSL_FLINK	00000000			MSUSL_WRITE	00000238		
HQBSL_UNIT_ONLN	00000010			MSUSQ_CTRL_ID	00000190		
HQBSQ_PENDING	00000030			MSUSQ_CTRL_INFO	00000198		
HQBSQ_TIME	00000020			MSUSQ_HQBLIST	000004B4		
HQBSW_CNT_FLGS	00000018			MSUSQ_QUEUE	00000070		
HQBSW_HST_TMO	0000001A			MSUST_NAME	00000014		
HQBSW_MAX_QUE	0000002A			MSUSW_INI_HOST	0000004C		
HQBSW_NUM_QUE	00000028			MSUSW_INI_PKT	00000046		
HQBSW_REFCT	0000002C			MSUSW_MAX_PKT	0000004A		
HQBSW_SIZE	00000008			MSUSW_MAX_QUE	00000062		
HQBSW_USE_FRAC	0000001C			MSUSW_MIN_PKT	0000004A		
IOC\$CVT_DEVNAM	*****	X	03	MSUSW_NUM_HOST	0000004E		
IPL\$ SCSI	= 00000008			MSUSW_NUM_PKT	00000048		
MSCP\$GET_COU1	0000029C	RG	03	MSUSW_NUM_QUE	00000060		
MSCP\$GET_COU2	00000338	RG	03	MSUSW_PACKET	00000044		
MSCP\$GET_HOST	0000011C	RG	03	MSUSW_SIZE	00000008		
MSCP\$GET_RESO	000001EC	RG	03	NNAME	000001CC	R	03
MSCP\$GET_UNIT	00000000	RG	03	NOWRT	00000028	R	02
MSCP\$M_UF_WRTPH	= 00002000			NOWRT_LEN	= 00000006		
MSCP\$M_UF_WRTPS	= 00001000			PB	= 000001E0	R	03
MSCP\$S_ST_MASK	= 00000005			PB\$S_SBLINK	= 00000030		
MSCP\$V_ST_MASK	= 00000000			PR\$ IPL	= 00000012		
MSUSB_CTRL_DAT	000001A0			REP_TBL	000002EE	R	03
MSUSB_SUBTYPE	0000000B			SB	000001E4	R	03
MSUSB_TYPE	0000000A			SB\$T_NODENAME	= 00000044		
MSUSB_VEC	00000024			SC\$SGL_MSCP	*****	X	03
MSUSC_LENGTH	000004CC			SS\$_DEVOFFLINE	= 00000084		
MSUSK_LENGTH	000004CC			SS\$_NORMAL	= 00000001		
MSUSK_MAX_HOST	= 00000010			STATUS_ARY	00000000	R	02
MSUSK_MAX_UNIT	= 00000010			TIME	0000002E	R	02
MSUSL_ABORT	000001B4			UQBSB_CMD_PEND	0000000D		
MSUSL_AVAIL	000001D0			UQBSB_FLAGS	0000002B		
MSUSL_BLKCOUNT	000002B0			UQBSB_HOST_CNT	0000000C		
MSUSL_BUFF_HEAD	00000050			UQBSB_SLOT	0000002A		
MSUSL_CDRP_LIST	00000070			UQBSB_TYPE	0000000A		
MSUSL_CDRP_SAVE	0000006C			UQBSK_LEN	0000009C		
MSUSL_CDSV_SIZE	00000068			UQBSK_LEN	0000009C		
MSUSL_D1	00000000			UQBSL_CDRP_BL	00000064		
MSUSL_D2	00000004			UQBSL_CDRP_FL	00000060		

MSCP\$SHOW
Symbol table

- MSCP Server Information

F 5

16-SEP-1984 02:05:50
5-SEP-1984 02:08:39

VAX/VMS Macro V04-00
[MSCP.SRC]MSCP\$SHOW.MAR;1

Page 12
(1)

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

! 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	000004CC (1228.)	01 (1.)	NOPIC	USR	CON	ABS	LCL	NOSHR	EXE	RD	WRT	NOVEC	BYTE				
SHOW\$RWDATA	00000036 (54.)	02 (2.)	NOPIC	USR	CON	REL	LCL	NOSHR	NOEXE	RD	WRT	NOVEC	LONG				
SHOW\$CODE	000003F8 (1016.)	03 (3.)	NOPIC	USR	CON	REL	LCL	NOSHR	EXE	RD	WRT	NOVEC	LONG				

! Performance indicators !

Phase	Page faults	CPU Time	Elapsed Time
Initialization	11	00:00:00.03	00:00:01.48
Command processing	75	00:00:00.57	00:00:05.81
Pass 1	303	00:00:12.30	00:00:36.48
Symbol table sort	0	00:00:01.86	00:00:04.20
Pass 2	71	00:00:02.15	00:00:06.95
Symbol table output	17	00:00:00.15	00:00:01.05
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 479 00:00:17.08 00:00:56.00

The working set limit was 1350 pages.

66802 bytes (131 pages) of virtual memory were used to buffer the intermediate code.

There were 70 pages of symbol table space allocated to hold 1231 non-local and 33 local symbols.

410 source lines were read in Pass 1, producing 30 object records in Pass 2.

18 pages of virtual memory were used to define 17 macros.

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

Macro library name

Macros defined

_ \$255\$DUA28:[SYS.OBJ]LIB.MLB;1
_ \$255\$DUA28:[MSCP.OBJ]MSCPDEF.MLB;1
_ \$255\$DUA28:[SYSLIB]STARLET.MLB;2
TOTALS (all libraries)

6
3
5
14

1344 GETS were required to define 14 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LISS:MSCPSHOW/OBJ=OBJ\$:MSCPSHOW MSRC\$:MSCPSHOW/UPDATE=(ENH\$:MSCPSHOW)+LIB\$:MSCPDEF/LIB+EXECML\$/LIB

AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY