

CCCCCCCCCCCC	LLL	IIIIIIII	UUU	UUU	TTTTTTTTTTTTTTTT	LLL
CCCCCCCCCCCC	LLL	IIIIIIII	UUU	UUU	TTTTTTTTTTTTTTTT	LLL
CCCCCCCCCCCC	LLL	IIIIIIII	UUU	UUU	TTTTTTTTTTTTTTTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCCCCCCCCCCC	LLLLLLLLLLLLLLLL	IIIIIIII	UUUUUUUUUUUUUU	UUUUUUUUUUUUUU	TTTT	LLLLLLLLLLLLLLLL
CCCCCCCCCCCC	LLLLLLLLLLLLLLLL	IIIIIIII	UUUUUUUUUUUUUU	UUUUUUUUUUUUUU	TTTT	LLLLLLLLLLLLLLLL
CCCCCCCCCCCC	LLLLLLLLLLLLLLLL	IIIIIIII	UUUUUUUUUUUUUU	UUUUUUUUUUUUUU	TTTT	LLLLLLLLLLLLLLLL

```
SSSSSSSS HH HH 000000 WW WW MM MM SSSSSSSS CCCCCCCC PPPPPPPP
SSSSSSSS HH HH 000000 WW WW MM MM SSSSSSSS CCCCCCCC PPPPPPPP
SS HH HH 00 00 WW WW MMMM MMMM SS CC PP PP PP
SS HH HH 00 00 WW WW MMMM MMMM SS CC PP PP PP
SS HH HH 00 00 WW WW MM MM SS CC PP PP PP
SSSSSS HH HH 00 00 WW WW MM MM SSSSSS CC PPPPPPPP
SSSSSS HH HH 00 00 WW WW MM MM SSSSSS CC PPPPPPPP
SS HH HH 00 00 WW WW MM MM SS CC PP
SS HH HH 00 00 WW WW MM MM SS CC PP
SS HH HH 00 00 WWW WW MM MM SS CC PP
SSSSSSS HH HH 000000 WW WW MM MM SSSSSSSS CCCCCCCC PP
SSSSSSSS HH HH 000000 WW WW MM MM SSSSSSSS CCCCCCCC PP
                                                                ....
                                                                ....
                                                                ....
                                                                ....
```

```
LL          IIIIII SSSSSSSS
LL          IIIIII SSSSSSSS
LL          II      SS
LL          II      SS
LL          II      SS
LL          II      SS
LL          II      SSSSSS
LL          II      SSSSSS
LL          II      SS
LL          II      SS
LL          II      SS
LL          II      SS
LLLLLLLLLLL IIIIII SSSSSSSS
LLLLLLLLLLL IIIIII SSSSSSSS
```

SHOWSMSCP
Table of contents

- MSCP Server Information

J 7

15-SEP-1984 23:50:53 VAX/VMS Macro V04-00

Page 0

(1) 28
(1) 58

PROGRAM DESCRIPTION
DECLARATIONS

```
0000 1 .TITLE SHOW$MSCP - 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 SHOW Utility
0000 33 :
0000 34 ABSTRACT
0000 35 :
0000 36 This module contains the routines to show
0000 37 information about the MSCP server.
0000 38 :
0000 39 ENVIRONMENT
0000 40 :
0000 41 NATIVE MODE, USER MODE
0000 42 :
0000 43 AUTHOR
0000 44 :
0000 45 Anne E. Warner, 10-Apr-1984
0000 46 :
0000 47 MODIFIED BY
0000 48 :
0000 49 V03-002 AEW0002 Anne Warner 07-Aug-1984
0000 50 Make a check to see if the return status from MSCP$SHOW
0000 51 is ss$_devoffline and if it is return the code show$_mscpnotld
0000 52 stating that the MSCP-Server code is not loaded.
0000 53 :
0000 54 V03-001 AEW0001 Anne Warner 03-Jul-1984
0000 55 Make changes for the reformat of the display.
0000 56 :--
```

```

0000 58      .SBTTL  DECLARATIONS
0000 59
0000 60  :
0000 61  :      SYMBOL DEFINITIONS
0000 62  :
0000 63
0000 64      .nocross
0000 65      $CDTDEF
0000 66      $CLIDF      ; CLI Utility definitions
0000 67      $IPLDEF
0000 68      $MSCPDEF
0000 69      $PBDEF
0000 70      $SSDEF      ; System service condition code def's
0000 71      $SYIDF      ; System-wide Information ($GETSYI)
0000 72
0000 73      .cross
0000 74
0000 75      .DEFAULT DISPLACEMENT LONG
0000 76
0000 77
0000 78
0000 79
0000 80  :
0000 81  :      STORAGE DEFINITIONS
0000 82  :
0000 83
00000000 84      .PSECT  SHOW$RWDATA,NOEXE,WRT,LONG
0000 85
0000 86  :
0000 87  :      HEADINGS AND HEADING DESCRIPTORS
0000 88  :
0000 89
0000 90 BAN_MAIN:
20 50 43 53 40 20 20 20 20 20 20 20 0000 91      .ASCII  /      MSCP-Served Devices on !AD !%D/
63 69 76 65 44 20 64 65 76 72 65 53 000C
25 21 20 44 41 21 20 6E 6F 20 73 65 0018
      44 0024
      00000025 0025 92 BAN_MAIN_LEN=-BAN_MAIN
      0025 93 BAN_MAIN_SHOW:
00000025 0025 94      .LONG BAN_MAIN_LEN
00000000 0029 95      .ADDRESS BAN_MAIN
002D 96
002D 97
20 20 20 20 20 20 20 20 20 20 20 20 0D 0A 002D 98 BAN_U:  .ASCII  <10><13>
20 20 20 20 20 20 20 20 20 20 20 20 002F 99      .ASCII  /
20 20 20 20 20 20 20 20 20 20 20 20 003B
20 20 20 20 20 20 20 20 20 20 20 20 0047
65 75 51 20 20 20 20 20 20 20 20 20 0053
      73 74 73 65 75 71 65 52 20 65 75 005F
      0D 0A 006A
20 20 20 20 20 3A 65 63 69 76 65 44 006C 100      .ASCII  <10><13>
73 75 74 61 74 53 20 20 20 20 20 20 0078 101      .ASCII  /Device:
20 6C 61 74 6F 54 20 20 20 20 20 20 0084
20 6C 61 74 6F 54 20 20 20 20 65 7A 69 53 0090
20 78 61 4D 20 20 20 20 74 6E 65 72 009C
      73 74 73 6F 48 20 20 20 20 00A8
00000084 00B1 102 BAN_U_LEN=-BAN_U

```

Queue Requests/

Status Total Size Current Max Hos

												103	BAN_U_SHOW:							
												00000084	00B1	104	.LONG BAN U LEN					
												0000002D	00B5	105	.ADDRESS BAN_U					
												0D 0A	00B9	106						
20	20	20	2	20	20	20	20	20	20	20	20	00B9	107	BAN_H:	.ASCII <10><13>					
20	20	20	20	20	20	20	20	20	20	20	20	00B8	108		.ASCII /				Queue Requests/	
20	20	20	20	20	20	20	20	20	20	20	20	00C7								
20	20	20	20	20	20	20	20	20	20	20	20	00D3								
65	75	51	20	20	20	20	20	20	20	20	20	00DF								
	73	74	73	65	75	71	65	52	20	65	75	00EB								
												0D 0A	00F6	109	.ASCII <10><13>					
20	20	20	20	20	20	20	3A	74	73	6F	48	00F8	110		.ASCII /Host:	Time of Connection	Current	Max	Devic	
65	6D	69	54	20	20	20	20	20	20	20	20	0104								
69	74	63	65	6E	6E	6F	43	20	66	6F	20	0110								
72	75	43	20	20	20	20	20	20	20	6E	6F	011C								
20	78	61	4D	20	20	20	20	74	6E	65	72	0128								
			73	65	63	69	76	65	44	20	20	0134								
												00000084	013D	111	BAN_H_LEN=-BAN_H					
												013D	013D	112	BAN_H_SHOW:					
												00000084	013D	113	.LONG BAN H LEN					
												000000B9	0141	114	.ADDRESS BAN_H					
												0145	0145	115						
												0D 0A	0145	116	BAN_R:	.ASCII <10><13>				Fragment Size/
20	20	20	20	20	20	20	20	20	20	20	20	0147	117		.ASCII /					
20	20	20	20	20	20	20	20	20	20	20	20	0153								
20	20	20	20	20	20	20	20	20	20	20	20	015F								
20	20	20	20	20	20	20	20	20	20	20	20	016B								
6E	65	6D	67	61	72	46	20	20	20	20	20	0177								
						65	7A	69	53	20	74	0183								
												0D 0A	0189	118	.ASCII <10><13>					
20	20	3A	73	65	63	72	75	6F	73	65	52	018B	119		.ASCII /Resources:	Total	Free	In Use	Minimum	Maxim
6C	61	74	6F	54	20	20	20	20	20	20	20	0197								
20	20	65	65	72	46	20	20	20	20	20	20	01A3								
20	20	65	73	55	20	6E	49	20	20	20	20	01AF								
20	20	6D	75	6D	69	6E	69	4D	20	20	20	01BB								
			6D	75	6D	69	78	61	4D	20	20	01C7								
												0000008B	01D0	120	BAN_R_LEN=-BAN_R					
												01D0	01D0	121						
												0000008B	01D0	122	BAN_R_SHOW:					
												00000145	01D0	123	.LONG BAN R LEN					
												01D8	01D4	124	.ADDRESS BAN_R					
												01D8	01D8	125						
20	20	20	20	20	20	20	20	20	20	0D	0A	01D8	126	BAN_M:	.ASCII <10><13>/	Current	Maximum/			
65	72	72	75	43	20	20	20	20	20	20	20	01E4								
6D	75	6D	69	78	61	4D	20	20	20	74	6E	01F0								
												00000024	01FC	127	BAN_M_LEN=-BAN_M					
												01FC	01FC	128	BAN_M_SHOW:					
												00000024	01FC	129	.LONG BAN M LEN					
												000001D8	0200	130	.ADDRESS BAN_M					
												0204	0204	131						
6F	43	20	74	73	65	75	71	65	52	0D	0A	0204	132	BAN_B:	.ASCII <10><13>/Request Count:/					
								3A	74	6E	75	0210								
												00000010	0214	133	BAN_B_LEN=-BAN_B					
												0214	0214	134	BAN_B_SHOW:					
												00000010	0214	135	.LONG BAN B LEN					
												00000204	0218	136	.ADDRESS BAN_B					
												021C	021C	137						

```

73 6E 6F 69 74 61 72 65 70 4F 0D 0A 021C 138 BAN_0: .ASCII <10><13>/Operations Count:/
      3A 74 6E 75 6F 43 20 0228
      00000013 022F 139 BAN_0_LEN=-BAN_0
      00000013 022F 140 BAN_0_SHOW:
      0000021C 022F 141 .LONG BAN_0_LEN
      0233 142 .ADDRESS BAN_0
      0237 143
      0237 144 :
      0237 145 :
      0237 146 :
      0237 147 :
      0237 148 CTRSTR_U:
      0237 149 .ASCID / !12AD!AD !8UL .2UW !2UW !2UB/
31 21 20 20 20 20 0000023F 010E0000 0237
20 20 20 20 20 20 44 41 21 44 41 32 0245
20 20 20 20 20 20 4C 55 38 21 20 20 0251
20 20 20 20 20 20 57 55 32 21 20 20 025D
20 20 20 20 20 20 57 55 32 21 20 0269
      42 55 32 21 20 0275
      027A 150
      027A 151 CTRSTR_H:
      027A 152 .ASCID / !12AC !XD !2UW !2UW !2UW/
20 20 20 20 20 20 44 25 21 20 43 41 32 0288
20 20 20 20 20 20 57 55 32 21 20 20 0294
20 20 20 20 20 20 57 55 32 21 20 20 02A0
      57 55 32 21 20 20 02AC
      02B2 153
      02B2 154 CTRSTR_BA:
      02B2 155 .ASCID / Buffer Area: !3UL !3UL !3UL !3UL
75 42 20 20 20 20 000002BA 010E0000 02B2
20 20 3A 61 65 72 41 20 72 65 66 66 02C0
20 20 20 20 20 20 4C 55 33 21 20 20 20 02CC
20 20 20 20 20 20 4C 55 33 21 20 20 20 02D8
20 20 20 20 20 20 4C 55 33 21 20 20 20 02E4
20 20 20 20 4C 55 33 21 20 20 20 20 02F0
      4C 55 33 21 20 20 20 20 02FC
      0304 156
      0304 157 CTRSTR_IOP:
      0304 158 .ASCID ? I/O Packets: !3UL !3UL?
2F 49 20 20 20 20 0000030C 010E0000 0304
20 20 3A 73 74 65 68 63 61 50 20 4F 0312
20 20 20 20 20 20 4C 55 33 21 20 20 20 031E
      4C 55 33 21 20 20 032A
      0330 159
      0330 160 CTRSTR_BW:
      0330 161 .ASCID / Buffer Wait: !2UW !2UW/
75 42 20 20 20 20 00000338 010E0000 0330
20 20 3A 74 69 61 57 20 72 65 66 66 033E
20 20 20 20 57 55 32 21 20 20 20 20 034A
      57 55 32 21 20 20 20 20 0356
      035E 162
      035E 163 REQX: .LONG REQ1
      0362 164 .LONG REQ2
      0366 165 .LONG REQ3
      036A 166 .LONG REQ4
      036E 167 REQ1: .ASCID / 0-7: !8UL 32-39: !8UL 88-103: !8UL/
20 20 20 20 20 20 00000376 010E0000 037C
20 20 4C 55 38 21 20 3A 37 20 30 20 0388
3A 39 33 20 32 33 20 20 20 20 20 20 0394
20 20 20 20 20 20 20 4C 55 38 21 20 03A0
55 38 21 20 3A 33 30 31 20 38 38 20 03AC
      4C

```


20 20 20 20 20 20	000003B5'	010E0000'	03AD	168 REQ2:	.ASCID /	8-15: !8UL	40-55: !8UL	104-127: !8UL/
20 20 4C 55 38 21	20 3A 35 31 2D 38		038B					
3A 35 35 2D 30 34	20 20 20 20 20 20		03C7					
20 20 20 20 20 20	20 4C 55 38 21 20		03D3					
55 38 21 20 3A 37	32 31 2D 34 30 31		03DF					
			03EB					
31 20 20 20 20 20	000003F4'	010E0000'	03EC	169 REQ3:	.ASCID /	16-23: !8UL	56-71: !8UL/	
20 20 4C 55 38 21	20 3A 33 32 2D 36		03FA					
3A 31 37 2D 36 35	20 20 20 20 20 20		0406					
	4C 55 38 21 20		0412					
32 20 20 20 20 20	0000041F'	010E0000'	0417	170 REQ4:	.ASCID /	24-31: !8UL	72-87: !8UL/	
20 20 4C 55 38 21	20 3A 31 33 2D 34		0425					
3A 37 38 2D 32 37	20 20 20 20 20 20		0431					
	4C 55 38 21 20		043D					
			0442	171				
	0000045A'		0442	172 CSLX:	.LONG CSL1			
	000004A0'		0446	173	.LONG CSL2			
	000004E6'		044A	174	.LONG CSL3			
	0000052C'		044E	175	.LONG CSL4			
	00000572'		0452	176	.LONG CSL5			
	000005B8'		0456	177	.LONG CSL6			
41 20 20 20 20 20	00000462'	010E0000'	045A	178 CSL1:	.ASCID /	ABORT !6UL	ERASE !6UL	READ !6UL/
21 20 20 20 20 20	20 20 54 52 4F 42		0468					
53 41 52 45 20 20	20 20 20 4C 55 36		0474					
4C 55 36 21 20 20	20 20 20 20 20 45		0480					
20 20 20 20 44 41	45 52 20 20 20 20		048C					
	4C 55 36 21 20 20		0498					
41 20 20 20 20 20	000004A8'	010E0000'	04A0	179 CSL2:	.ASCID /	ACCESS !6UL	FLUSH !6UL	REPLACE !6UL/
21 20 20 20 20 20	20 53 53 45 43 43		04AE					
53 55 4C 46 20 20	20 20 20 4C 55 36		04BA					
4C 55 36 21 20 20	20 20 20 20 20 48		04C6					
20 45 43 41 4C 50	45 52 20 20 20 20		04D2					
	4C 55 36 21 20 20		04DE					
41 20 20 20 20 20	000004EE'	010E0000'	04E6	180 CSL3:	.ASCID /	AVAILABLE !6UL	GET COM STS !6UL	SET CTL CHR !6UL/
21 20 20 20 45 4C	42 41 4C 49 41 56		04F4					
20 54 45 47 20 20	20 20 20 4C 55 36		0500					
4C 55 36 21 20 53	54 53 20 4D 4F 43		050C					
20 4C 54 43 20 54	45 53 20 20 20 20		0518					
	4C 55 36 21 20 52		0524					
43 20 20 20 20 20	00000534'	010E0000'	052C	181 CSL4:	.ASCID /	CMP CTL DAT !6UL	GET UNT STS !6UL	SET UNT CHR !6UL/
21 20 54 41 44 20	4C 54 43 20 50 4D		053A					
20 54 45 47 20 20	20 20 20 4C 55 36		0546					
4C 55 36 21 20 53	54 53 20 54 4E 55		0552					
20 54 4E 55 20 54	45 53 20 20 20 20		055E					
	4C 55 36 21 20 52		056A					
43 20 20 20 20 20	0000057A'	010E0000'	0572	182 CSL5:	.ASCID /	CMP HST DAT !6UL	ONLINE !6UL	WRITE !6UL/
21 20 54 41 44 20	54 53 48 20 50 4D		0580					
49 4C 4E 4F 20 20	20 20 20 4C 55 36		058C					
4C 55 36 21 20 20	20 20 20 20 45 4E		0598					
20 20 20 45 54 49	52 57 20 20 20 20		05A4					
	4C 55 36 21 20 20		05B0					
74 6F 54 20 20 20	0C0005C0'	010E0000'	05B8	183 CSL6:	.ASCID /	Total !6UL/		
21 20 20 20 20 20	20 20 20 20 6C 61		05C6					
	4C 55 36		05D2					
			05D5	184				
			05D5	185 :				
			05D5	186 :	QUALIFIERS			

```

05D5 187 ;
05D5 188 ;
54 4E 55 4F 43 000005DD'010E0000' 05D5 189 ALL: .ASCID /ALL/
54 53 4F 48 000005EB'010E0000' 05E0 190 COUNT: .ASCID /COUNT/
52 55 4F 53 45 52 00000601'010E0000' 05ED 191 HOST: .ASCID /HOST/
45 43 0607 192 RESOURCE:
0609 193 .ASCID /RESOURCE/

```

VARIABLES

```

00000000 0609 199 NUMNODE: .LONG 0 ; Null argument for buffer address in GET_NODE
0000061D 060D 200 NODE: .BLKB 16 ; ASCII string to contain the node name for header
; after call to $GETSYIW
061D 201 ; ITMLST for $GETSYIW
0010 061D 202 GET_NODE: ; buffer length defined to be 16 for nodename
10D9 061F 203 .WORD 16 ; Item code to get the node name
0000060D' 0621 204 .WORD SYIS NODENAME ; Buffer address. Not used with this item code
00000609' 0625 205 .ADDRESS NODE ; return length address
00000000 0629 206 .ADDRESS NUMNODE ; signifies end of item list
062D 207 .LONG 0 ; Argument list for main heading
00000000 062D 208 MAIN_ARG: ; Will contain number of characters in node name (NUMNODE)
0000060D' 0631 209 .LONG 0 ; contains the address of the nodename from $GETSYIW
00000000 0635 210 .ADDRESS NODE ; and a zero to specify current date and time wanted
00000000 0639 211 .LONG 0 ; The maximum number of times to loop through a routine
063D 212 MAX: .LONG 0 ; determined in MSCP routines.
00000000 063D 213 .LONG 0 ; The count of how many times a routine has been executed
00000000 0641 214 CNT: .LONG 0 ; This variable contains the address of the first parameter
0645 215 TOP_ARG: .LONG 0 ; to be printed. This is determined in the MSCP routines.
00000655 0645 216 .LONG 0 ; Holds the address of TOP_ARG so the arglist may be
0655 217 HOLD: .BLKL 4 ; separated for output
0655 218 ;
0655 219 ;
0655 220 ;

```

```

00000000 222
0000 0000 223 .PSECT SHOW$CODE,EXE,NOVRT,LONG
0000 0000 224 .ENTRY SHOW$MSCP,*M<>
0002 225
0002 226 :++
0002 227 :
0002 228 : FUNCTIONAL DESCRIPTION
0002 229 :
0002 230 : SHOW$MSCP controls what information is displayed, determined by
0002 231 : the user input. It first checks if any qualifiers were specified
0002 232 : and if so the corresponding information is displayed. If the /ALL
0002 233 : qualifier was specified, all of the information is displayed. If no
0002 234 : qualifiers were specified then only the device information is displayed.
0002 235 : This routine determines this information and dispatched to the proper
0002 236 : routine(s).
0002 237 :
0002 238 : CALLING SEQUENCE
0002 239 :
0002 240 : SHODEV ----> SHOW$MSCP
0002 241 :
0002 242 : INPUT PARAMETERS
0002 243 :
0002 244 : None
0002 245 :
0002 246 : OUTPUT PARAMETERS
0002 247 :
0002 248 : None
0002 249 :
0002 250 :--
0002 251 :
00000A2'EF 00 FB 0002 252 CALLS #0,SHO_UNIT : The device info is always displayed
      7C 50 E9 0009 253 BLBC RO,10$
000005D5'EF 01 DF 000C 254 PUSHAL ALL : Find if ALL information is requested
00000000'GF 01 FB 0012 255 CALLS #1,G^CLISPRESENT
      1E 50 E9 0019 256 BLBC RO,2$ : Branch if not
00000145'EF 00 FB 001C 257 CALLS #0,SHO_HOST : If so then call all routines to display
      62 50 E9 0023 258 BLBC RO,10$ : the HOST, RESOURCE, and COUNT info
000001B0'EF 00 FB 0026 259 CALLS #0,SHO_RESO
      58 50 E9 002D 260 BLBC RO,10$
0000023E'EF 00 FB 0030 261 CALLS #0,SHO_COUN
      4E 50 E9 0037 262 BLBC RO,10$
000005ED'EF 01 DF 003A 263 2$: PUSHAL HOST : If the /ALL qualifier was not set
00000000'GF 01 FB 0040 264 CALLS #1,G^CLISPRESENT : then /HOST, /RESOURCE, and /COUNT
      0A 50 E9 0047 265 BLBC RO,3$ : must be checked individually.
00000145'EF 00 FB 004A 266 CALLS #0,SHO_HOST : Call the CLI routine to check if
      34 50 E9 0051 267 BLBC RO,10$ : present, if not try the next qualifier.
000005F9'EF 01 DF 0054 268 3$: PUSHAL RESOURCE : if it is then display info for the
00000000'GF 01 FB 005A 269 CALLS #1,G^CLISPRESENT : qualifier and then check for the next
      0A 50 E9 0061 270 BLBC RO,4$ : one.
000001B0'EF 00 FB 0064 271 CALLS #0,SHO_RESO
      1A 50 E9 006B 272 BLBC RO,10$
000005E0'EF 01 DF 006E 273 4$: PUSHAL COUNT
00000000'GF 01 FB 0074 274 CALLS #1,G^CLISPRESENT : After all qualifiers checked and
      23 50 E9 007B 275 BLBC RO,20$ : processed return to caller.
0000023E'EF 00 FB 007E 276 CALLS #0,SHO_COUN
      19 50 E8 0085 277 BLBC RO,20$
00000084 8F 50 D1 0088 278 10$: CML RO,#SS$_DEVOffline

```

50	00000000	07	12	008F	279	BNEQ	15\$		
		8F	DO	0091	280	MOVL	#SHOWS_MSCPNOTLD,R0		
		50	DD	0098	281	PUSHL	R0		
00000000	'GF	01	FB	009A	282	CALLS	#1,G^LIB\$SIGNAL		; If there was an error along the way
			04	00A1	283	RET			; call the signal routine to tell user.

```

0100 0042 85 .ENTRY SHO_UNIT,^M<R8>
      00A4 286
      00A4 287 :++
      00A4 288
      00A4 289 FUNCTIONAL DESCRIPTION
      00A4 290
      00A4 291 SHO UNIT controls the display of information units in the data
      00A4 292 base. It calls MSCP$GET_UNIT to get the actual information then
      00A4 293 sends the information and its format to SHOW$WRITE_LINE which prints
      00A4 294 it. SHO_UNITS repeats this until all possible units (MAX) have been
      00A4 295 processed.
      00A4 296
      00A4 297 CALLING SEQUENCE
      00A4 298
      00A4 299 CALLS #0,SHO_UNIT
      00A4 300
      00A4 301 :--
      00A4 302
      00A4 303 $GETSYIW S ITMLST = GET_NODE ; Get the node name for heading
      00B8 304 BLBC R0,7$
      00BE 305 MOVL NUMNODE,MAIN_ARG ; Number of characters in node name
      00C9 306 PUSHAL MAIN_ARG ; Contains node name and zero to
      00CF 307 PUSHAQ BAN MAIN SHOW ; specify date and time wanted
      00D5 308 CALLS #2,G^SHOW$WRITE_LINE
      00DC 309 PUSHL #0 ; Set up argument list for output
      00DE 310 PUSHAQ BAN U SHOW ; Heading. Routine in module
      00E4 311 CALLS #2,C^SHOW$WRITE_LINE ; [cliutl.src]showio does output
      00EB 312 CLRL CNT ; Clear the counter to start
      00F1 313 5' PUSHAL MAX ; Want to get maximum number of units
      00F7 314 PUSHL CNT ; Supply the count
      00FD 315 PUSHAL TOP_ARG ; Want to get back address of argument list
      0103 316 PUSHL #3
      0105 317 MOVL SP,R8 ; R8 will contain addr. of arg list for call
      0108 318 $CMKRNLS ROUTIN = G^MSCP$GET_UNIT,- ; Execute kernel routine
      0108 319 ARGST = (R8)
      0117 320 7$: BLBC R0,8$ ; Check return status
      011A 321 ADDL #16,SP ; Reclaim 4 longwords of stack space
      011D 322 TSTL TOP_ARG ; If passed back as zero there is no
      0123 323 BLEQ 6$ ; information to print.
      0125 324 PUSHL TOP_ARG ; Address of argument list of items
      0128 325 PUSHAQ CTRSTR U ; Format for argument list
      0131 326 CALLS #2,G^SHOW$WRITE_LINE ; Call routine in SHOWIO to output
      0138 327 6$: AOBLS MAX,CNT,5$ ; Loop through all units
      0144 328 8$: RET
      0145 329

```

```

          59 50
0000062D'EF 00000609'EF D0 00BB 304
          0000062D'EF DF 00C9 305
          00000025'EF 7F 00CF 306
00000000'GF 02 FB 00D5 307
          00 DD 00DC 308
          000000B1'EF 7F 00DE 309
00000000'GF 02 FB 00E4 310
          0000063D'EF D4 00EB 311
          00000639'EF DF 00F1 312
          0000063D'EF DD 00F7 313
          00000641'EF DF 00FD 314
          03 DD 0103 315
          58 SE D0 0105 316
          0108 317
          0108 318
          0108 319
          2A 50 E9 0117 320
          SE 10 C0 011A 321
          00000641'EF D5 011D 322
          13 15 0123 323
          00000641'EF DD 0125 324
          00000237'EF 7F 0128 325
          00000000'GF 02 FB 0131 326
AD 0000063D'EF 00000639'EF F2 0138 327
          04 0144 328
          0145 329

```

```

0100 0145 331 .ENTRY SHO_HOST,^M<R8>
      0145 332
      0147 333
      0147 334
      0147 335
      0147 336
      0147 337
      0147 338
      0147 339
      0147 340
      0147 341
      0147 342
      0147 343
      0147 344
      0147 345
      0147 346
      0147 347
      0147 348
      0147 349
      0149 350
      014F 351
      0156 352
      015C 353
      0162 354
      0168 355
      016E 356
      0170 357
      0173 358
      0173 359
      0182 360
      0185 361
      0188 362
      018E 363
      0190 364
      0196 365
      019C 366
      01A3 367
      01AF 368
      01B0 369

```

FUNCTIONAL DESCRIPTION

SHO_HOST controls the display of information concerning the status of MSCP hosts. It calls MSCP\$GET_HOST to get the information and SHOW\$WRITE_LINE to display the information. MSCP\$GET_HOST is called until it all hosts are processed.

CALLING SEQUENCE

CALLS #0,SHO_UNIT

```

DD 0147 349 PUSHL #0
7F 0149 350 PUSHAQ BAN_H_SHOW ; Argument list for header
FB 014F 351 CALLS #2,-G^SHOW$WRITE_LINE ; Print header
D4 0156 352 CLR  CNT ; Initialize counter
DF 015C 353 5$: PUSHAL MAX ; Need to get maximum number of hosts
DD 0162 354 PUSHL CNT ; Supply the count
DF 0168 355 PUSHAL TOP_ARG ; Need to get back address of arg list
DD 016E 356 PUSHL #3 ; Number of arguments
DO 0170 357 MOVL SP,R8 ; R8 will contain arg list for call
0173 358 $CMKRNLS_ROUTIN = G^MSCP$GET_HOST,- ; Execute kernel routine
0173 359 ARGST = (R8)
E9 0182 360 BLBC R0,7$ ; Check return status
CO 0185 361 ADDL #16,SP ; reclaim 4 longwords of stack space
D5 0188 362 TSTL TOP_ARG ; If zero there is nothing to print
15 018E 363 BLEQ 6$
DD 0190 364 PUSHL TOP_ARG ; Start of argument list
7F 0196 365 PUSHAQ CTRSTR_H ; Format for argument list
FB 019C 366 CALLS #2,G^SHOW$WRITE_LINE ; Print information from SHOWIO
F2 01A3 367 6$: AOBLS MAX,CNT,5$ ; Loop through all hosts
04 01AF 368 7$: RET
01B0 369

```

```

0000013D'EF 00 DD
00000000'GF 02 FB
0000063D'EF 04 D4
00000639'EF DF
0000063D'EF DD
00000641'EF DF
58 SE DD
2A 50 E9
SE 10 CO
00000641'EF D5
13 15
00000641'EF DD
0000027A'EF 7F
00000000'GF 02 FB
AD 0000063D'EF 00000639'EF F2
04

```

```

0300 01B0 371      .ENTRY SHO_RESO,^M<R8,R9>
      01B2 372
      01B2 373      :++
      01B2 374      :
      01B2 375      FUNCTIONAL DESCRIPTION
      01B2 376
      01B2 377      SHO_RESO controls the display of information about buffers and
      01B2 378      packets (resources) of MSCP served devices. The routine MSCP$GET_RESO
      01B2 379      is called to get the information and SHOWWRITE_LINE to print it.
      01B2 380
      01B2 381      CALLING SEQUENCE
      01B2 382
      01B2 383      CALLS #0,SHO_RESO
      01B2 384      :--
      01B2 385
      01B2 386
      000001D0'EF DD 01B2 387      PUSHL #0
      00000000'GF 02 DF 01B4 388      PUSHAL BAN_R_SHOW ; Print header
      00000641'EF 01 DF 01BA 389      CALLS #2,G^SHOWWRITE_LINE
      01 01 DD 01C7 390      PUSHAL TOP_ARG ; Need address of argument list
      58 SE DO 01C9 391      PUSHL #1
      01CC 392      MOVL SP,R8 ; R8 will contain address of arg list for ca
      01CC 393      $CMKRNLS_ROUTIN = G^MSCP$GET_RESO,- ; Execute kernel routine
      01CC 394      ARGST = (R8)
      5F 50 E9 01DB 395      BLBC R0,58 ; Check return status
      59 00000641'EF DO 01DE 396      MOVL TOP_ARG,R9
      00000641'EF DD 01E5 397      PUSHL TOP_ARG ; Start of argument list
      000002B2'EF 7F 01EB 398      PUSHAQ CTRSTR BA ; Format of output for buffer area
      00000000'GF 02 FB 01F1 399      CALLS #2,G^SHOWWRITE_LINE ; Print information
      00000641'EF 14 A9 DE 01F8 400      MOVAL 5*4(R9),TOP_ARG ; find start of I/O Packet output
      00000641'EF DD 0200 401      PUSHL TOP_ARG ; Start of argument list
      00000304'EF 7F 0206 402      PUSHAQ CTRSTR IOP ; Format of output for I/O Packet
      00000000'GF 02 FB 020C 403      CALLS #2,G^SHOWWRITE_LINE ; Print information
      00 00 DD 0213 404      PUSHL #0
      000001FC'EF 7F 0215 405      PUSHAQ BAN_M_SHOW ; Header for Buffer Wait
      00000000'GF 02 FB 021B 406      CALLS #2,G^SHOWWRITE_LINE
      00000641'EF 1C A9 DE 0222 407      MOVAL 7*4(R9),TOP_ARG ; find start of Buffer Wait output
      00000641'EF DD 022A 408      PUSHL TOP_ARG ; Start of argument list
      00000330'EF 7F 0230 409      PUSHAQ CTRSTR BW ; Format of output for Buffer Wait
      00000000'GF 02 FB 0236 410      CALLS #2,G^SHOWWRITE_LINE ; Print information
      04 023D 411 58:      RET
      023E 412

```

```

03C0 023E 414 .ENTRY SHO_COUN,^M<R6,R7,R8,R9>
      0240 415
      0240 416 :++
      0240 417 :
      0240 418 FUNCTIONAL DESCRIPTION
      0240 419
      0240 420 SHO_COUN controls the display of information about the current
      0240 421 counts of commands and I/O's. The information is gathered in two
      0240 422 routines, MSCP$GET_COU1 and MSCP$GET_COU2, and output by
      0240 423 SHOW$WRITE_LINE.
      0240 424
      0240 425 CALLING SEQUENCE
      0240 426
      0240 427 CALLS #0,SHO_COUN
      0240 428
      0240 429 :--
      56 D4 0240 430 CLRRL R6
      00 DD 0242 431 PUSHL #0
      00000214'EF DF 0244 432 PUSHAL BAN,B,SHOW ; Print count heading
      00000000'GF 02 FB 024A 433 CALLS #2,G^SHOW$WRITE_LINE
      00000641'EF DF 0251 434 PUSHAL TOP_ARG ; Get address of argument list
      01 DD 0257 435 PUSHL #1
      58 5E DO 0259 436 MOVL SP,R8 ; Call arg list in R8
      025C 437 $CMKRNLS_ROUTIN = G^MSCP$GET_COU1,-
      025C 438 ARGST = (R8)
      59 38 50 E9 026B 439 BLBC R0,4$ ; Return status
      00000641'EF DO 026E 440 MOVL TOP_ARG,R9
      57 00000645'EF DE 0275 441 2$: MOVAL HOLD,R7 ; The display will split the arglist
      87 69 DO 027C 442 MOVL (R9),(R7)+ ; into four lines. To do this the
      87 10 A9 DO 027F 443 MOVL 4*4(R9),(R7)+ ; correct arguments are moved into a
      87 20 A9 DO 0283 444 MOVL 8*4(R9),(R7)+ ; list with starting address HOLD. Four
      67 30 A9 DO 0287 445 MOVL 12*4(R9),(R7) ; arguments are moved and outputted, then
      00000645'EF DF 028B 446 PUSHAL HOLD ; the next four until all arguments have
      0000035E'EF 46 DD 0291 447 PUSHL REQX[R6] ; output. R6 is used as an index to
      00000000'GF 02 FB 0298 448 CALLS #2,G^SHOW$WRITE_LINE ; select the correct format line
      59 04 A9 DE 029F 449 MOVAL 4(R9),R9
      CE 56 03 F3 02A3 450 AOBLEQ #3,R6,2$
      03 11 02A7 451 BRB 5$
      004D 31 02A9 452 4$: BRW 7$ ; Branch won't reach
      02AC 453 :
      02AC 454 :
      02AC 455 :
      02AC 456 :
      00 DD 02AC 457 5$: PUSHL #0
      0000022F'EF DF 02AE 458 PUSHAL BAN,O,SHOW
      00000000'GF 02 FB 02B4 459 CALLS #2,G^SHOW$WRITE_LINE ; Print the heading
      56 D4 02BB 460 CLRRL R6 ; R6 is the counter in this routine
      00000641'EF DF 02BD 461 PUSHA TOP_ARG ; Need to get address of argument list.
      01 DD 02C3 462 PUSHL #1 ; Number of arguments
      58 5E DO 02C5 463 MOVL SP,R8 ; R8 contains addr of arglist for call
      02C8 464 $CMKRNLS_ROUTIN = G^MSCP$GET_COU2,- ; Execute kernel routine
      02C8 465 ARGST = (R8)
      59 1F 50 E9 02D7 466 BLBC R0,7$ ; Check return status
      00000641'EF DO 02DA 467 MOVL TOP_ARG,R9
      59 DD 02E1 468 6$: PUSHL R9 ; Start of argument list
      00000442'EF 46 DD 02E3 469 PUSHL CSLX[R6] ; Format of argument list
      00000000'GF 02 FB 02EA 470 CALLS #2,G^SHOW$WRITE_LINE ; Print information

```


- MSCP Server Information
DECLARATIONS

K 8

15-SEP-1984 23:50:53 VAX/VMS Macro V04-00
4-SEP-1984 23:23:04 [CLIUTL.SRC]SHOWMSCP.MAR;1

59	OC	A9	DE	02F1	471	MOVAL	3*4(R9),R9	
E8	56	05	F3	02F5	472	AOBLEQ	#5,R6,6\$	
			04	02F9	473	RET		: Loop thru all commands
				02FA	474			
				02FA	475	.END		

SHOWSMSCP
Symbol table

- MSCP Server Information

L 8

15-SEP-1984 23:50:53 VAX/VMS Macro V04-00
4-SEP-1984 23:23:04 [CLIUTL.SRC]SHOWMSCP.MAR;1

\$ST1	= 00000001		
ALL	000005D5	R	02
BAN_B	00000204	R	02
BAN_B_LEN	= 00000010		
BAN_B_SHOW	00000214	R	02
BAN_H	000000B9	R	02
BAN_H_LEN	= 00000084		
BAN_H_SHOW	0000013D	R	02
BAN_M	000001D8	R	02
BAN_MAIN	00000000	R	02
BAN_MAIN_LEN	= 00000025		
BAN_MAIN_SHOW	00000025	R	02
BAN_M_LEN	= 00000024		
BAN_M_SHOW	000001FC	R	02
BAN_O	0000021C	R	02
BAN_O_LEN	= 00000013		
BAN_O_SHOW	0000022F	R	02
BAN_R	00000145	R	02
BAN_R_LEN	= 00000088		
BAN_R_SHOW	000001D0	R	02
BAN_U	0000002D	R	02
BAN_U_LEN	= 00000084		
BAN_U_SHOW	000000B1	R	02
CLIPRESENT	*****	X	03
CNT	0000063D	R	02
COUNT	000005E0	R	02
CSL1	0000045A	R	02
CSL2	000004A0	R	02
CSL3	000004E6	R	02
CSL4	0000052C	R	02
CSL5	00000572	R	02
CSL6	000005B8	R	02
CSLX	00000442	R	02
CTRSTR_BA	000002B2	R	02
CTRSTR_BW	00000330	R	02
CTRSTR_H	0000027A	R	02
CTRSTR_IOP	00000304	R	02
CTRSTR_U	00000237	R	02
GET_NODE	0000061D	R	02
HOLD	00000645	R	02
HOST	000005ED	R	02
LIBSSIGNAL	*****	X	03
MAIN_ARG	0000062D	R	02
MAX	00000639	R	02
MSCP\$GET_COU1	*****	X	03
MSCP\$GET_COU2	*****	X	03
MSCP\$GET_HOST	*****	X	03
MSCP\$GET_RESO	*****	X	03
MSCP\$GET_UNIT	*****	X	03
NODE	0000060D	R	02
NUMNODE	00000609	R	02
REQ1	0000036E	R	02
REQ2	000003AD	R	02
REQ3	000003EC	R	02
REQ4	00000417	R	02
REQX	0000035E	R	02
RESOURCE	000005F9	R	02

SHOWSMSCP	00000000	RG	03
SHOW\$WRITE LINE	*****	X	03
SHOW\$MSCPROTLD	*****	X	03
SHO_COUN	0000023E	RG	03
SHO_HOST	00000145	RG	03
SHO_RESO	000001B0	RG	03
SHO_UNIT	000000A2	RG	03
SS\$DEVOFFLINE	= 00000084		
SY\$NODENAME	= 000010D9		
SY\$CMKRNL	*****	GX	03
SY\$GETSYIW	*****	GX	03
TOP_ARG	00000641	R	02

! 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\$\$	00000000 (0.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
SHOWSRWDATA	00000655 (1621.)	02 (2.)	NOPIC USR CON REL LCL NOSHR NOEXE RD WRT NOVEC LONG
SHOWSCODE	000002FA (762.)	03 (3.)	NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC LONG

! Performance indicators !

Phase	Page faults	CPU Time	Elapsed Time
Initialization	16	00:00:00.10	00:00:01.51
Command processing	102	00:00:00.88	00:00:04.80
Pass 1	377	00:00:13.38	00:00:41.97
Symbol table sort	0	00:00:02.31	00:00:06.36
Pass 2	110	00:00:02.74	00:00:08.45
Symbol table output	10	00:00:00.10	00:00:00.39
Psect synopsis output	1	00:00:00.02	00:00:00.04
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	618	00:00:19.53	00:01:03.52

The working set limit was 1500 pages.
80807 bytes (158 pages) of virtual memory were used to buffer the intermediate code.
There were 90 pages of symbol table space allocated to hold 1523 non-local and 19 local symbols.
475 source lines were read in Pass 1, producing 34 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:[CLIUTL OBJ]CLIUTL.MLB;1	0
_\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	4
_\$255\$DUA28:[SYSLIB]STARLET.MLB;2	10
TOTALS (all libraries)	14

1584 GETS were required to define 14 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:SHOWMSCP/OBJ=OBJ\$:SHOWMSCP MSRC\$:SHOWMSCP/UPDATE=(ENH\$:SHOWMSCP)+EXECML\$/LIB+LIB\$:CLIUTL/LIB

This image displays a grid of 100 terminal window screenshots, arranged in 10 rows and 10 columns. Each window shows a different system utility or diagnostic tool. The most prominent and clearly legible windows include:

- SHOWTERM LIS**: Located in the upper right quadrant, showing terminal status information.
- SHOWMISC LIS**: Located in the middle left, showing miscellaneous system information.
- SHOWPROC LIS**: Located in the middle center, showing process status.
- SHOWSYS LIS**: Located in the middle right, showing system status.
- SHOWMAIN LIS**: Located in the lower middle left, showing main system information.
- SHOWMSCP LIS**: Located in the lower middle center, showing message control panel status.
- SHOWMSG LIS**: Located in the lower middle right, showing message status.
- SHOWQUE LIS**: Located in the lower right, showing queue status.

The remaining windows in the grid show various other system utilities, including file management, device control, and system configuration tools, though their text is less legible due to the image's resolution and the small size of the individual windows.