

[illegible]

```

SSSSSSSS HH HH 000000 WW WW MM MM I I I I I SSSSSSSS CCCCCCCC
SSSSSSSS HH HH 000000 WW WW MM MM I I I I I SSSSSSSS CCCCCCCC
SS SS HH HH 00 00 WW WW MMMM MMMM I I I I I SS SSSSSSSS CC CCCCCC
SS SS HH HH 00 00 WW WW MM MM I I I I I SS SSSSSS CC CCCCCC
SSSSSSS HH HH 00 00 WW WW MM MM I I I I I SSSSSS SS CC CCCCCC
SSSSSSS HH HH 00 00 WW WW MM MM I I I I I SSSSSS SS CC CCCCCC
SS SS HH HH 00 00 WW WW MM MM I I I I I SSSSSS SS CC CCCCCC
SS SS HH HH 00 00 WWW WWW MM MM I I I I I SSSSSSSS CCCCCCCC
SSSSSSSS HH HH 000000 WW WW MM MM I I I I I SSSSSSSS CCCCCCCC
SSSSSSSS HH HH 000000 WW WW MM MM I I I I I SSSSSSSS CCCCCCCC

LL LL I I I I I SSSSSSSS
LL LL I I I I I SSSSSSSS
LL LL I I SS SSSSSS
LL LL I I SS SSSSSS
LL LL I I SS SSSSSS
LL LL I I SS SSSSSS
LL LL I I SS SSSSSS
LL LL I I SS SSSSSS
LL LL I I SS SSSSSS
LLLLLLLLLL I I I I I SSSSSSSS
LLLLLLLLLL I I I I I SSSSSSSS
```



```
1 0001 0 MODULE showmisc (IDENT = 'V04-000',
2 0002 0 ADDRESSING_MODE (EXTERNAL = GENERAL)) =
3 0003 0
4 0004 1 BEGIN
5 0005 1
6 0006 1
7 0007 1 *****
8 0008 1 *
9 0009 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
10 0010 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
11 0011 1 * ALL RIGHTS RESERVED.
12 0012 1 *
13 0013 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
14 0014 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
15 0015 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
16 0016 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
17 0017 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
18 0018 1 * TRANSFERRED.
19 0019 1 *
20 0020 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
21 0021 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
22 0022 1 * CORPORATION.
23 0023 1 *
24 0024 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
25 0025 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
26 0026 1 *
27 0027 1 *
28 0028 1 *****
29 0029 1
30 0030 1
31 0031 1 ++
32 0032 1
33 0033 1 FACILITY: SHOW utility
34 0034 1
35 0035 1 ABSTRACT:
36 0036 1 This module contains the option routines for SHOW RMS, SHOW
37 0037 1 WORKING_SET, SHOW ACCOUNTING
38 0038 1
39 0039 1 ENVIRONMENT:
40 0040 1 VAX native, user mode.
41 0041 1
42 0042 1 AUTHOR: Gerry Smith CREATION DATE: 25-Jun-1982
43 0043 1
44 0044 1 MODIFIED BY:
45 0045 1
46 0046 1 V03-007 RAS0281 Ron Schaefer 27-Mar-1984
47 0047 1 Add Network Block Count display to SHOW RMS;
48 0048 1 adjust display alignments.
49 0049 1
50 0050 1 V03-006 MCN0155 Maria del C. Nasr 01-Mar-1984
51 0051 1 The offset to the disallow flags is from the start of
52 0052 1 the PCB structure, and not the longword.
53 0053 1
54 0054 1 V03-005 MCN0150 Maria del C. Nasr 09-Feb-1984
55 0055 1 Make display for SHOW WORKING_SET values larger.
56 0056 1
57 0057 1 V03-004 GAS0148 Gerry Smith 27-Jun-1983
```

SHOWMISC
V04-000

6 6
16-Sep-1984 01:21:13
14-Sep-1984 12:09:39

VAX-11 Bliss-32 V4.0-742
[CLIUTL.SRC]SHOWMISC.B32;1

Page 2
(1)

: 58 0058 1 :
: 59 0059 1 :
: 60 0060 1 :
: 61 0061 1 :
: 62 0062 1 :
: 63 0063 1 :
: 64 0064 1 :
: 65 0065 1 :
: 66 0066 1 :
: 67 0067 1 :
: 68 0068 1 :
: 69 0069 1 :
: 70 0070 1 :
: 71 0071 1 :--

Update SHOW ACCOUNTING to display all classes
of accounting.

V03-003 GAS0125 Gerry Smith 20-Apr-1983
Fix displays for SHOW RMS_DEFAULT

V03-002 GAS0107 Gerry Smith 8-Feb-1983
Extend the output field width for SHOW WORKING_SET.

V03-001 GAS0105 Gerry Smith 31-Jan-1983
Fix the output of SHOW RMS so that the multi-buffer
counts come out correctly.

SHOWMISC
V04-000

16-Sep-1984 01:21:13
14-Sep-1984 12:09:39

```
VAX-11 Bliss-32 v4.0-742
[CLIUTL.SRC]SHOWMISC.B32;1
```

Page 3
(2)

```

: 73          0072 1
: 74          0073 1 :
: 75          0074 1 : Include files
: 76          0075 1 :
: 77          0076 1
: 78          0077 1 LIBRARY 'SYSS$LIBRARY:LIB';
: 79          0078 1 REQUIRE 'SRC$:SHOWDEF';
: 80          0177 1

```

```
! VAX/VMS system definitions
! SHOW common definitions
```

```
82 0178 1 1
83 0179 1 1 Table of contents
84 0180 1 1
85 0181 1 1
86 0182 1 1 FORWARD ROUTINE
87 0183 1 1 show$accounting : NOVALUE;
88 0184 1 1 show$rms_default : NOVALUE;
89 0185 1 1 show$working_set : NOVALUE;
90 0186 1 1
91 0187 1 1 EXTERNAL
92 0188 1 1 exe$gl_acmflags,
93 0189 1 1 sys$gb_dmbfsur,
94 0190 1 1 sys$gb_dmbfsmt,
95 0191 1 1 sys$gb_dmbfsdk,
96 0192 1 1 sys$gb_dmbfrel,
97 0193 1 1 sys$gb_dmbfidx,
98 0194 1 1 sys$gb_dmbfhsh,
99 0195 1 1 sys$gb_dmbc,
100 0196 1 1 sys$gb_dfnbc,
101 0197 1 1 sys$gw_rmsextend,
102 0198 1 1 sys$gb_rmsprolog,
103 0199 1 1 pio$gb_dmbfsur,
104 0200 1 1 pio$gb_dmbfsmt,
105 0201 1 1 pio$gb_dmbfsdk,
106 0202 1 1 pio$gb_dmbfrel,
107 0203 1 1 pio$gb_dmbfidx,
108 0204 1 1 pio$gb_dmbfhsh,
109 0205 1 1 pio$gb_dmbc,
110 0206 1 1 pio$gb_dfnbc,
111 0207 1 1 pio$gw_rmsextend,
112 0208 1 1 pio$gb_rmsprolog;
113 0209 1 1
114 0210 1 1 EXTERNAL ROUTINE
115 0211 1 1 show$write_line : NOVALUE;
```



```
.. 117      0212 1 |
.. 118      0213 1 | Declare some OWN storage, string descriptors which describe the various
.. 119      0214 1 | types of accounting.
.. 120      0215 1 |
.. 121      0216 1 | OWN
.. 122      0217 1 |   acc_type : VECTOR[10] INITIAL
.. 123      0218 1 |       (%ASCID : PROCESS      any process termination',
.. 124      0219 1 |         %ASCID : IMAGE      image execution',
.. 125      0220 1 |         %ASCID : INTERACTIVE interactive job termination',
.. 126      0221 1 |         %ASCID : LOGIN_FAILURE login failures',
.. 127      0222 1 |         %ASCID : SUBPROCESS subprocess termination',
.. 128      0223 1 |         %ASCID : DETACHED   detached job termination',
.. 129      0224 1 |         %ASCID : BATCH      batch job termination',
.. 130      0225 1 |         %ASCID : NETWORK    network job termination',
.. 131      0226 1 |         %ASCID : PRINT      all print jobs',
.. 132      0227 1 |         %ASCID : MESSAGE    user messages'
.. 133      0228 1 |   );
.. 134      0229 1 |
```

```
136 0230 1 GLOBAL ROUTINE show$accounting : NOVALUE =
137 0231 2 BEGIN
138 0232 2
139 0233 2 ---
140 0234 2
141 0235 2 This routine performs the SHOW ACCOUNTING function. A copy of the
142 0236 2 accounting flags is brought into local memory, and based on the
143 0237 2 bit setting, the current accounting options are displayed.
144 0238 2
145 0239 2 ---
146 0240 2
147 0241 2 LOCAL
148 0242 2 acmflags : BITVECTOR[32]; ! Temp storage for accounting flags
149 0243 2
150 0244 2
151 0245 2 Get a copy of the accounting flags, so that the flag settings are consistent
152 0246 2 for the duration of this invocation of SHOW ACCOUNTING.
153 0247 2
154 0248 2 acmflags = .exe$gl_acmflags;
155 0249 2
156 0250 2
157 0251 2 Determine if any accounting is turned on. If not, then simply print a
158 0252 2 line saying that, and return.
159 0253 2
160 0254 2 IF .acmflags EQL 0
161 0255 2 THEN show$write_line(%ASCII 'Accounting is not currently enabled.', 0)
162 0256 2
163 0257 2
164 0258 2 If something is set, then write a header line and then print whatever
165 0259 2 activities are shown in ACMFLAGS.
166 0260 2
167 0261 2 ELSE
168 0262 2 BEGIN
169 0263 2 show$write_line(%ASCII 'Accounting is currently enabled to log the following activities:!/ ', 0);
170 0264 2 INCR i FROM 0 TO 9 DO
171 0265 2 IF .acmflags[i]
172 0266 2 THEN show$write_line(.acc_type[i], 0);
173 0267 2 END;
174 0268 2
175 0269 2 RETURN;
176 0270 1 END;
```

																.TITLE	SHOWMISC			
																.IDENT	\V04-000\			
																.PSECT	\$SPLITS,NOWRT,NOEXE,2			
20	20	53	53	45	43	4F	52	50	20	20	20	20	20	20	00000	P.AAB:	.ASCII	\	PROCESS	any process terminal\
65	63	6F	72	70	20	79	6E	61	20	20	20	20	20	20	0000F					
					61	6E	69	6D	72	65	74	20	73	73	0001E					
											6E	6F	69	74	00028					
													010E002C	0002C	P.AAA:	.ASCII	\tion\			
													00000000	00030		.LONG	17694764			
20	20	20	20	45	47	41	4D	49	20	20	20	20	20	20	00034	P.AAD:	.ADDRESS	P.AAB		
65	78	65	20	65	67	61	6D	69	20	20	20	20	20	20	00043		.ASCII	\	IMAGE	image execution\
									6E	6F	69	74	75	63	00052					

[illegible]

SHOWMISC
V04-000

M 6
16-Sep-1984 01:21:13 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:09:39 [CLIUTL.SRC]SHOWMISC.B32;1

Page 8
(5)

00 00 2F 21 3A 73 65 69 74 69 76 69 74 00253
010E0042 00260 P.AAW: .LONG 17694786
00000000 00264 .ADDRESS P.AAX

.PSECT \$OWNS\$,NOEXE,2

00000000' 00000000' 00000000' 00000000' 00000000' 00000000' 00000000' 00000000' ACC_TYPE:

.ADDRESS P.AAA, P.AAC, P.AAE, P.AAG, P.AAI, -
P.AAK, P.AAM, P.AAO, P.AAQ, P.AAS

.EXTRN EX\$GL_ACMFLAGS
.EXTRN SY\$GB_DFMBFSUR
.EXTRN SY\$GB_DFMBFSMT
.EXTRN SY\$GB_DFMBFSDK
.EXTRN SY\$GB_DFMBFREL
.EXTRN SY\$GB_DFMBFIDX
.EXTRN SY\$GB_DFMBFHS
.EXTRN SY\$GB_DFMBF, SY\$GB_DFNBC
.EXTRN SY\$GW_RMSEXTEND
.EXTRN SY\$GB_RMSPROLOG
.EXTRN PIO\$GB_DFMBFSUR
.EXTRN PIO\$GB_DFMBFSMT
.EXTRN PIO\$GB_DFMBFSDK
.EXTRN PIO\$GB_DFMBFREL
.EXTRN PIO\$GB_DFMBFIDX
.EXTRN PIO\$GB_DFMBFHS
.EXTRN PIO\$GB_DFMBF, PIO\$GB_DFNBC
.EXTRN PIO\$GW_RMSEXTEND
.EXTRN PIO\$GB_RMSPROLOG
.EXTRN SHOW\$WRITE_LINE

.PSECT \$CODE\$,NOWRT,2

001C 00000
54 00000000G 00 9E 00002
53 00000000G 00 D0 00009
0A 12 00010
7E D4 00012
64 0000' CF 9F 00014
02 FB 00018
04 0001B
7E D4 0001C 1\$:
0000' CF 9F 0001E
64 02 FB 00022
52 D4 00025
0A 53 52 E1 00027 2\$:
7E D4 0002B
0000' CF 42 DD 0002D
64 02 FB 00032
EE 52 09 F3 00035 3\$:
04 00039

.ENTRY SHOW\$ACCOUNTING, Save R2,R3,R4
MOVAB SHOW\$WRITE_LINE, R4
MOVL EX\$GL_ACMFLAGS, ACMFLAGS
BNEQ 1\$
CLRL -(SP)
PUSHAB P.AAU
CALLS #2, SHOW\$WRITE_LINE
RET
CLRL -(SP)
PUSHAB P.AAW
CALLS #2, SHOW\$WRITE_LINE
CLRL 1
BBC 1, ACMFLAGS, 3\$
CLRL -(SP)
PUSHL ACC_TYPE[1]
CALLS #2, SHOW\$WRITE_LINE
AOBLEQ #9, 1, 2\$
RET

: 0230
: 0248
: 0254
: 0255
: 0263
: 0264
: 0265
: 0266
: 0265
: 0270

; Routine Size: 58 bytes, Routine Base: \$CODE\$ + 0000


```
178 0271 1 GLOBAL ROUTINE show$rms_default : NOVALUE =
179 0272 2 BEGIN
180 0273 2
181 0274 2 ---
182 0275 2
183 0276 2 This routine performs the SHOW RMS_DEFAULT funtion. The current settings
184 0277 2 of the RMS default parameters are output.
185 0278 2
186 0279 2 ---
187 0280 2
188 0281 2 LOCAL
189 0282 2     status,
190 0283 2     desc : VECTOR[2],
191 0284 2     arglist : VECTOR[8],
192 0285 2     buffer : VECTOR[128,BYTE];
193 0286 2
194 0287 2
195 0288 2 Print the heading lines
196 0289 2
197 0290 2 show$write_line (%ASCID '          MULTI-      MULTIBUFFER COUNTS      NETWORK', 0,
198 0291 2                   %ASCID '          BLOCK      Indexed Relative      Sequential      BLOCK', 0,
199 0292 2                   %ASCID '          COUNT      COUNT      Disk      Magtape      Unit      Record      COUNT', 0);
200 0293 2
201 0294 2
202 0295 2 Format and print the process RMS defaults
203 0296 2
204 0297 2 arglist[0] = .pio$gb_dfmbc;
205 0298 2 arglist[1] = .pio$gb_dfmbfidx;
206 0299 2 arglist[2] = .pio$gb_dfmbfrel;
207 0300 2 arglist[3] = .pio$gb_dfmbfsdk;
208 0301 2 arglist[4] = .pio$gb_dfmbfsmt;
209 0302 2 arglist[5] = .pio$gb_dfmbfsur;
210 0303 2 arglist[6] = .pio$gb_dfnbc;
211 0304 2
212 0305 2 show$write_line(%ASCID 'Process  !4SB      ! !4SB      !4SB      !4SB      !4SB      !4SB      ! !4SB',
213 0306 2                   arglist);
214 0307 2
215 0308 2
216 0309 2 Format and print the system RMS defaults
217 0310 2
218 0311 2 arglist[0] = .sys$gb_dfmbc;
219 0312 2 arglist[1] = .sys$gb_dfmbfidx;
220 0313 2 arglist[2] = .sys$gb_dfmbfrel;
221 0314 2 arglist[3] = .sys$gb_dfmbfsdk;
222 0315 2 arglist[4] = .sys$gb_dfmbfsmt;
223 0316 2 arglist[5] = .sys$gb_dfmbfsur;
224 0317 2 arglist[6] = .sys$gb_dfnbc;
225 0318 2
226 0319 2 show$write_line(%ASCID 'System  !4SB      ! !4SB      !4SB      !4SB      !4SB      !4SB      ! !4SB',
227 0320 2                   arglist);
228 0321 2
229 0322 2
230 0323 2 Now display the prologue and extend default quantities
231 0324 2
232 0325 2 arglist[0] = .pio$gb_rmsprolog;
233 0326 2 arglist[1] = .pio$gw_rmsextend;
234 0327 2 show$write_line(%ASCID '!/      Prolog      Extend Quantity !/Process  !4SB      !5UW', arglist);
```



```
.. 235      0328 2
.. 236      0329 2
.. 237      0330 2 arglist[0] = .sys$gb_rmsprolog;
.. 238      0331 2 arglist[1] = .sys$gw_rmsextend;
.. 239      0332 2 show$write_line(%ASCII'D 'System !4SB !5UW', arglist);
.. 240      0333 2 RETURN;
.. 241      0334 1 END; ! End of show$rms
```

```
.PSECT $SPLITS,NOWRT,NOEXE,2
49 54 4C 55 4D 20 20 20 20 20 20 20 20 20 20 00268 P.AAZ: .ASCII \ MULTI- ; MULTI\
20 20 20 20 20 20 20 20 20 20 20 20 20 20 00277
20 20 20 20 20 20 20 20 20 20 20 20 20 20 00286
20 20 53 54 4E 49 54 4C 55 4D 20 20 20 20 00290 .ASCII \BUFFER COUNTS ; NETWORK\<0>
20 7C 20 20 20 20 20 20 20 20 20 20 20 20 0029F
00 4B 52 4F 57 54 45 4E 002AE
00 00 002B6
010E004D 002B8 P.AAY: .ASCII <0><0>
00000000 002BC .LONG 17694797
002C0 P.ABB: .ADDRESS P.AAZ
52 43 4F 4C 42 20 20 20 20 20 20 20 20 20 002CF .ASCII \ BLOCK ; Indexed Relative \
6E 65 75 71 65 53 20 20 20 20 20 20 20 20 002DE .ASCII \ Sequential ; BLOCK\
20 7C 20 20 20 20 20 20 20 20 20 20 20 20 002F7
4B 43 4F 4C 42 20 20 20 20 20 20 20 20 20 00306
010E004C 0030C P.ABA: .LONG 17694796
00000000 00310 .ADDRESS P.ABB
54 4E 55 4F 43 20 20 20 20 20 20 20 20 20 00314 P.ABD: .ASCII \ COUNT ; \
20 20 20 20 20 20 20 20 20 20 20 20 20 20 00323
20 65 70 61 74 67 61 4D 20 20 20 20 20 20 00332
20 7C 20 64 72 6F 63 65 52 20 74 69 6E 55 20 0033C .ASCII \Disk Magtape Unit Record ; COUNT\
54 4E 55 4F 43 20 0034B
010E004C 00360 P.ABC: .LONG 17694796
00000000 00364 .ADDRESS P.ABD
20 20 42 53 34 21 20 20 73 73 65 63 6F 72 50 00368 P.ABF: .ASCII \Process !4SB ; !4SB !4SB !\
20 20 20 20 20 20 42 53 34 21 20 20 7C 20 20 00377
20 20 20 20 42 53 34 21 20 20 20 42 53 34 21 00386
20 7C 20 20 20 42 53 34 21 20 20 20 42 53 34 21 00390 .ASCII \4SB !4SB !4SB ; !4SB\<0>
20 20 20 20 20 20 20 20 42 53 34 21 20 20 0039F
00 42 53 34 21 003AE
00 00 003B3
010E004A 003B4 P.ABE: .ASCII <0>
00000000 003B8 .LONG 17694794
003BC P.ABH: .ADDRESS P.ABF
20 20 20 20 20 20 42 53 34 21 20 20 7C 20 20 003CB .ASCII \System !4SB ; !4SB !4SB !\
20 20 20 20 42 53 34 21 20 20 20 42 53 34 21 003DA
20 7C 20 20 20 42 53 34 21 20 20 20 42 53 34 21 003E4
20 20 20 20 20 20 20 20 42 53 34 21 20 20 003F3
00 42 53 34 21 00402
00 00 00407
010E004A 00408 P.ABG: .ASCII <0>
00000000 0040C .LONG 17694794
00410 P.ABJ: .ADDRESS P.ABH
6F 72 50 20 20 20 20 20 20 20 20 20 20 20 20 0041F .ASCII \!/ Prolog Extend Quantity !/\
51 20 64 6E 65 74 78 45 20 20 20 20 20 20 20 20 67 6F 6C
```



```
20 20 42 53 34 2F 21 20 79 74 69 74 6E 61 75 0042E
   00 57 55 35 21 20 20 73 73 65 63 6F 72 50 00438
                                010E0043 00447
                                00000000 00454
                                00000000 00458
20 20 42 53 34 21 20 20 20 6D 65 74 73 79 53 0045C
   00 57 55 35 21 20 20 20 20 20 20 20 20 20 0046B
                                010E001B 00478
                                00000000 0047C
```

```
.ASCII \Process !4SB !5UW\<0>
P.ABI: .LONG 17694787
        .ADDRESS P.ABJ
P.ABL: .ASCII \System !4SB !5UW\<0>
P.ABK: .LONG 17694747
        .ADDRESS P.ABL
```

```
0004 00000
52 00000000G 00 9E 00002
5E FF58 CE 9E 00009
      0000' 7E D4 0000E
      0000' CF 9F 00010
      0000' 7E D4 00014
      0000' CF 9F 00016
      0000' 7E D4 0001A
      0000' CF 9F 0001C
62 06 FB 00020
D8 AD 00000000G 00 D0 00023
DC AD 00000000G 00 D0 0002B
E0 AD 00000000G 00 D0 00033
E4 AD 00000000G 00 D0 0003B
E8 AD 00000000G 00 D0 00043
EC AD 00000000G 00 D0 0004B
F0 AD 00000000G 00 D0 00053
      D8 AD 9F 0005B
      0000' CF 9F 0005E
62 02 FB 00062
D8 AD 00000000G 00 D0 00065
DC AD 00000000G 00 D0 0006D
E0 AD 00000000G 00 D0 C0075
E4 AD 00000000G 00 D0 0007D
E8 AD 00000000G 00 D0 00085
EC AD 00000000G 00 D0 0008D
F0 AD 00000000G 00 D0 00095
      D8 AD 9F 0009D
      0000' CF 9F 000A0
62 02 FB 000A4
D8 AD 00000000G 00 D0 000A7
DC AD 00000000G 00 D0 000AF
      D8 AD 9F 000B7
      0000' CF 9F 000BA
62 02 FB 000BE
D8 AD 00000000G 00 D0 000C1
DC AD 00000000G 00 D0 000C9
      D8 AD 9F 000D1
      0000' CF 9F 000D4
62 02 FB 000D8
      04 000DB
```

.PSECT \$CODE\$,NOWRT,2

```
.ENTRY SHOW$RMS DEFAULT, Save R2
MOVAB SHOW$WRITE_LINE, R2
MOVAB -168(SP), SP
CLRL -(SP)
PUSHAB P.ABC
CLRL -(SP)
PUSHAB P.ABA
CLRL -(SP)
PUSHAB P.AAY
CALLS #6, SHOW$WRITE_LINE
MOVL PIOSGB_DFMBBC, ARGLIST
MOVL PIOSGB_DFMBFIDX, ARGLIST+4
MOVL PIOSGB_DFMBFREL, ARGLIST+8
MOVL PIOSGB_DFMBFSKD, ARGLIST+12
MOVL PIOSGB_DFMBFSMT, ARGLIST+16
MOVL PIOSGB_DFMBFSUR, ARGLIST+20
MOVL PIOSGB_DFNBC, ARGLIST+24
PUSHAB ARGLIST
PUSHAB P.ABE
CALLS #2, SHOW$WRITE_LINE
MOVL SYSSGB_DFMBBC, ARGLIST
MOVL SYSSGB_DFMBFIDX, ARGLIST+4
MOVL SYSSGB_DFMBFREL, ARGLIST+8
MOVL SYSSGB_DFMBFSKD, ARGLIST+12
MOVL SYSSGB_DFMBFSMT, ARGLIST+16
MOVL SYSSGB_DFMBFSUR, ARGLIST+20
MOVL SYSSGB_DFNBC, ARGLIST+24
PUSHAB ARGLIST
PUSHAB P.ABG
CALLS #2, SHOW$WRITE_LINE
MOVL PIOSGB_RMSPROLOG, ARGLIST
MOVL PIOSGW_RMSEXTEND, ARGLIST+4
PUSHAB ARGLIST
PUSHAB P.ABI
CALLS #2, SHOW$WRITE_LINE
MOVL SYSSGB_RMSPROLOG, ARGLIST
MOVL SYSSGW_RMSEXTEND, ARGLIST+4
PUSHAB ARGLIST
PUSHAB P.ABK
CALLS #2, SHOW$WRITE_LINE
RET
```

; Routine Size: 220 bytes, Routine Base: \$CODE\$ + 003A

SHOWMISC
V04-000

D 7
16-Sep-1984 01:21:13
14-Sep-1984 12:09:39

VAX-11 Bliss-32 V4.0-742
[CLIUTL.SRC]SHOWMISC.B32;1

Page 12
(6)


```
243 0335 1 GLOBAL ROUTINE show$working_set : NOVALUE =
244 0336 2 BEGIN
245 0337
246 0338 |---
247 0339 |
248 0340 | The current working set parameters are output.
249 0341 |
250 0342 |---
251 0343
252 0344 LOCAL
253 0345     status,
254 0346     default : VOLATILE,
255 0347     quota : VOLATILE,
256 0348     extent : VOLATILE,
257 0349     max_quota : VOLATILE,
258 0350     max_extent : VOLATILE,
259 0351     pcb_flags : BITVECTOR[32],
260 0352     item_list : $BLOCK[7*4*4],
261 0353     iosb : VECTOR[2],
262 0354     arglist : VECTOR[3],
263 0355     desc : VECTOR[2],
264 0356     buffer : VECTOR[128,BYTE];
265 0357
266 0358 LITERAL
267 0359     disallow_bit = $BITPOSITION (pcb$v_disaws);
268 0360
269 P 0361     setup_list( item_list,
270 P 0362                 |pi$sts,      4, pcb_flags,      0,
271 P 0363                 |pi$dfwscnt,  4, default,      0,
272 P 0364                 |pi$wsquota,  4, quota,        0,
273 P 0365                 |pi$wsextent, 4, extent,        0,
274 P 0366                 |pi$wsauth,  4, max_quota,     0,
275 P 0367                 |pi$wsauthext, 4, max_extent,    0,
276 0368                 0,0,0,0);
277 0369
278 P 0370     status = $GETJPIW(ITMLST = item_list,
279 0371                 IOSB = iosb);
280 0372
281 0373     IF .status THEN status = (.iosb[0])<0,16>;
282 0374     IF NOT .status
283 0375     THEN
284 0376         SIGNAL_STOP(.status);
285 0377
286 0378     arglist[0] = .default;
287 0379     arglist[1] = .quota;
288 0380     arglist[2] = .extent;
289 0381
290 0382     show$write_line(%ASCII ' Working Set      /Limit= !UL   /Quota= !UL   /Extent= !UL',
291 0383                 arglist);
292 0384
293 0385     arglist[0] = .max_quota;
294 0386     arglist[1] = .max_extent;
295 0387
296 0388     IF .pcb_flags[disallow_bit]
297 0389     THEN show$write_line(%ASCII ' Adjustment disabled   Authorized Quota= !UL   Authorized Extent= !UL',
298 0390                 arglist);
299 0391     ELSE show$write_line(%ASCII ' Adjustment enabled   Authorized Quota= !UL   Authorized Extent= !UL',
                arglist);
```



```
: 300      0392 2  
: 301      0393 2 RETURN;  
: 302      0394 1 END;
```

! End of show\$work

```
20 20 74 65 53 20 67 6E 69 6B 72 6F 57 20 20 00480 P.ABN: .PSECT $SPLITS,NOWRT,NOEXE,2  
4C 55 21 20 3D 74 69 6D 69 4C 2F 20 20 20 0048F .ASCII \ Working Set /Limit= !UL /Quota=\  
74 6E 65 74 78 45 2F 20 20 20 51 2F 20 20 20 0049E  
010E003C 004A8 .ASCII \ !UL /Extent= !UL\  
00000000 004B7  
69 64 20 74 6E 65 6D 74 73 75 6A 64 41 20 20 004BC P.ABM: .LONG 17694780  
72 6F 68 74 75 41 20 20 20 64 65 6C 62 61 73 004C0 .ADDRESS P.ABN  
7A 69 72 6F 68 74 75 41 20 20 4C 55 21 20 3D 004C4 P.ABP: .ASCII \ Adjustment disabled Authorized Quota\  
00 4C 55 21 20 3D 74 6E 65 74 78 45 20 64 65 004D3 .ASCII \= !UL Authorized Extent= !UL<0><0>  
00 00 0050A  
00 0050B  
010E0045 0050C P.ABO: .ASCII <0>  
00000000 00510 .LONG 17694789  
6E 65 20 74 6E 65 6D 74 73 75 6A 64 41 20 20 00514 P.ABR: .ADDRESS P.ABP  
72 6F 68 74 75 41 20 20 20 64 65 6C 62 61 00523 .ASCII \ Adjustment enabled Authorized Quota\  
7A 69 72 6F 68 74 75 41 20 20 4C 55 21 20 3D 00532  
00 4C 55 21 20 3D 74 6E 65 74 78 45 20 64 65 0053C .ASCII \= !UL Authorized Extent= !UL<0><0>  
00 00 0054B  
00 0055A  
00 0055B  
010E0045 0055C P.ABQ: .ASCII <0>  
00000000 00560 .LONG 17694789  
.ADDRESS P.ABR
```

.EXTRN SYSS\$GETJPIW

.PSECT \$CODE\$,NOWRT,2

```
0004 00000  
52 00000000G 00 9E 00002  
5E FEDC CE 9E 00009  
50 FF7C CD 9E 0000E  
80 03050004 8F D0 00013  
80 6E 9E 0001A  
80 04030004 80 D4 0001D  
80 FC 8F D0 0001F  
80 04020004 80 D4 0002A  
80 F8 8F D0 0002C  
80 04160004 80 D4 00037  
80 F4 8F D0 00039  
80 04010004 80 D4 00044  
80 F0 8F D0 00046  
80 AD 9E 0004D  
80 D4 00051
```

```
.ENTRY SHOW$WORKING SET, Save R2  
MOVAB SHOW$WRITE LINE, R2  
MOVAB -292(SP), SP  
MOVAB ITEM LIST, PTR  
MOVL #50659332, (PTR)+  
MOVAB PCB FLAGS, (PTR)+  
CLRL (PTR)+  
MOVL #67305476, (PTR)+  
MOVAB DEFAULT, (PTR)+  
CLRL (PTR)+  
MOVL #67239940, (PTR)+  
MOVAB QUOTA, (PTR)+  
CLRL (PTR)+  
MOVL #68550660, (PTR)+  
MOVAB EXTENT, (PTR)+  
CLRL (PTR)+  
MOVL #67174404, (PTR)+  
MOVAB MAX QUOTA, (PTR)+  
CLRL (PTR)+
```

0335

0368

80	04170004	8F	D0	00053	MOVL	#68616196, (PTR)+	
80	EC	AD	9E	0005A	MOVAB	MAX_EXTENT, (PTR)+	
		80	D4	0005E	CLRL	(PTR)+	
		80	7C	00060	CLRQ	(PTR)+	
		80	D4	00062	CLRL	(PTR)+	
		7E	7C	00064	CLRQ	-(SP)	0371
	FF74	CD	9F	00066	PUSHAB	IOSB	
	FF7C	CD	9F	0006A	PUSHAB	ITEM_LIST	
		7E	7C	0006E	CLRQ	-(SP)	
		7E	D4	00070	CLRL	-(SP)	
00000000G	00	07	FB	00072	CALLS	#7, SYSSGETJPIW	
	08	50	E9	00079	BLBC	STATUS, 1\$	0372
	50	CD	D0	0007C	MOVL	IOSB, STATUS	
	09	50	E8	00081	BLBS	STATUS, 2\$	0373
		50	DD	00084	PUSHL	STATUS	0375
00000000G	00	01	FB	00086	CALLS	#1, LIB\$STOP	
	008C	AD	D0	0008D	MOVL	DEFAULT, ARGLIST	0377
	0090	AD	D0	00093	MOVL	QUOTA, ARGLIST+4	0378
	FF70	AD	D0	00099	MOVL	EXTENT, ARGLIST+8	0379
		CE	9F	0009F	PUSHAB	ARGLIST	0381
		CF	9F	000A3	PUSHAB	P.ABM	
	62	02	FB	000A7	CALLS	#2, SHOW\$WRITE_LINE	
	008C	AD	D0	000AA	MOVL	MAX_QUOTA, ARGLIST	0384
	0090	AD	D0	000B0	MOVL	MAX_EXTENT, ARGLIST+4	0385
	0A	AE	E9	000B6	BLBC	PCB_FLAGS+3, 3\$	0387
		CE	9F	000BA	PUSHAB	ARGLIST	0388
		CF	9F	000BE	PUSHAB	P.ABO	
		08	11	000C2	BRB	4\$	
	008C	CE	9F	000C4	PUSHAB	ARGLIST	0390
	0000	CF	9F	000C8	PUSHAB	P.ABO	
	62	02	FB	000CC	CALLS	#2, SHOW\$WRITE_LINE	
		04	000CF	RET			0394

; Routine Size: 208 bytes, Routine Base: \$CODE\$ + 0116

SHOWMISC
V04-000

H 7
16-Sep-1984 01:21:13
14-Sep-1984 12:09:39

VAX-11 Bliss-32 V4.0-742
[CLIUTL.SRC]SHOWMISC.B32;1

Page 16
(8)

: 304 0395 1 END
: 305 0396 0 ELUDOM

.EXTRN LIB\$STOP

PSECT SUMMARY

Name	Bytes	Attributes
\$PLITS	1380	NOVEC,NOWRT, RD ,NOEXE,NOSHR, LCL, REL, CON,NOPI,ALIGN(2)
\$OWNS	40	NOVEC, WRT, RD ,NOEXE,NOSHR, LCL, REL, CON,NOPI,ALIGN(2)
\$CODES	486	NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPI,ALIGN(2)

Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
_\$255\$DUA28:[SYSLIB]LIB.L32;1	18619	22	0	1000	00:01.9

COMMAND QUALIFIERS

: BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LISS:SHOWMISC/OBJ=OBJ\$:SHOWMISC MSRC\$:SHOWMISC/UPDATE=(ENH\$:SHOWMISC)

: Size: 486 code + 1420 data bytes
: Run Time: 00:14.1
: Elapsed Time: 00:41.5
: Lines/CPU Min: 1689
: Lexemes/CPU-Min: 18772
: Memory Used: 130 pages
: Compilation Complete

0057

**DIGITAL
CONFIDENTIAL**