

DDDDDDDDDDDD		CCCCCCCCCCCC	LLL
DDDDDDDDDDDD		CCCCCCCCCCCC	LLL
DDDDDDDDDDDD		CCCCCCCCCCCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDDDDDDDDDDD		CCCCCCCCCCCC	LLLLLLLLLLLLLLLL
DDDDDDDDDDDD		CCCCCCCCCCCC	LLLLLLLLLLLLLLLL
DDDDDDDDDDDD		CCCCCCCCCCCC	LLLLLLLLLLLLLLLL

```

SSSSSSSS HH HH 000000 WW WW
SSSSSSSS HH HH 000000 WW WW
SS HH HH 00 00 WW WW
SS HH HH 00 00 WW WW
SS HH HH 00 00 WW WW
SS HH HH 00 00 WW WW
SSSSSS HH HH HH HH HH HH WW WW
SSSSSS HH HH HH HH HH HH WW WW
SS HH HH 00 00 WW WW WW WW
SS HH HH 00 00 WW WW WW WW
SS HH HH 00 00 WW WW WW WW
SSSSSSS HH HH 000000 WW WW
SSSSSSS HH HH 000000 WW WW

```

```

....
....
....
....

```

```

LL IIIIII SSSSSSSS
LL IIIIII SSSSSSSS
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 IIIIII SSSSSSSS
LLLLLLLLL IIIIII SSSSSSSS
LLLLLLLLL IIIIII SSSSSSSS

```

SHOW  
Table of contents

- SHOW COMMAND EXECUTION

C 10

16-SEP-1984 00:16:12 VAX/VMS Macro V04-00

Page 0

(3)	150	SHOW DEFAULT
(4)	193	SHOW PROTECTION
(5)	247	SHOW SYMBOL TABLE ENTRIES
(6)	401	SHOW STATUS
(7)	490	SHOW DAYTIME
(8)	520	SHOW DISK QUOTA

00000000

```

0000 1 .TITLE SHOW - SHOW COMMAND EXECUTION
0000 2 .IDENT 'V04-000'
0000 3
0000 4
0000 5 *****
0000 6 *
0000 7 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 8 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 9 * ALL RIGHTS RESERVED.
0000 10 *
0000 11 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 12 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 13 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 14 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 15 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 16 * TRANSFERRED.
0000 17 *
0000 18 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 19 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 20 * CORPORATION.
0000 21 *
0000 22 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 23 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 24 *
0000 25 *
0000 26 *****
0000 27
0000 28 SHOW INFORMATION DCLS COMMAND EXECUTION
0000 29
0000 30 SHOW DAYTIME
0000 31 SHOW DEFAULT
0000 32 SHOW PROTECTION
0000 33 SHOW STATUS
0000 34 SHOW DISK QUOTA AND USAGE
0000 35
0000 36 D. N. CUTLER 15-APR-77
0000 37
0000 38 MODIFIED BY:
0000 39
0000 40 V03-013 HWS0084 Harold Schultz 19-Jul-1984
0000 41 Fix formatting of SHOW STATUS output. (make it non-TAB
0000 42 dependent).
0000 43
0000 44 V03-012 HWS0054 Harold Schultz 09-Apr-1984
0000 45 Change SHOW SYMBOL default from /NOLOG to /LOG.
0000 46
0000 47 V03-011 HWS0032 Harold Schultz 14-Mar-1984
0000 48 Report truncation on symbol definition output. Add
0000 49 SHOW SYMBOL/LOG.
0000 50
0000 51 V03-010 HWS0012 Harold Schultz 13-Feb-1984
0000 52 Fix broken branch.
0000 53
0000 54 V03-009 PCG0008 Peter George 18-Aug-1983
0000 55 Change the way that default protection is retrieved.
0000 56
0000 57 V03-008 PCG0007 Peter George 30-Jun-1983

```

```
0000 58 : Handle event flags more intelligently.
0000 59 : Reformat call to DCL$CVTUIIC.
0000 60 : Remove SHOW TRANSLATION.
0000 61 :
0000 62 : V03-007 PCG0006 Peter George 15-Jun-1983
0000 63 : Display octal longword in SHOW SYMBOL.
0000 64 : Fix broken offset.
0000 65 :
0000 66 : V03-006 PCG0005 Peter George 17-Feb-1983
0000 67 : Remove reference to $CLIDEFQUALSHOW.
0000 68 : Remove SETBIT WRK_V_NOSTAT from SHOW SYMBOL.
0000 69 : Call DCL$GETNVAL.
0000 70 :
0000 71 : V03-005 PCG0004 Peter George 07-Jan-1983
0000 72 : Modularize SHOW DEF and SHOW PROT so the F$ENVIRONMENT
0000 73 : can call them.
0000 74 :
0000 75 : V03-004 PCG0003 Peter George 16-Nov-1982
0000 76 : Reference WRK_C_SCRSTKSIZ instead of WRK_C_MSGBUFSIZ.
0000 77 :
0000 78 : V03-003 PCG0002 Peter George 21-Oct-1982
0000 79 : Make SHOW SYMBOL work with the second version keyword
0000 80 : parsing code.
0000 81 :
0000 82 : V03-002 PCG0001 Peter George 15-Oct-1982
0000 83 : Modify SHOW SYMBOL to work with the new keyword parsing code.
0000 84 :
0000 85 : V03-001 MTR0001 Mike Rhodes 17-Mar-1982
0000 86 : Fix continuation if .ASCID string defined at FAO_STRING:.
0000 87 :---
```

```

0000 39 :
0000 90 : MACRO LIBRARY CALLS
0000 91 :
0000 92 : PRCDEF ;DEFINE PROCESS WORK AREA
0000 93 : WRKDEF ;DEFINE COMMAND WORK AREA
0000 94 : PTRDEF ;DEFINE RESULT PARSE DESCRIPTOR FORMAT
0000 95 : SYMDEF ;DEFINE SYMBOL ENTRY OFFSETS
0000 96 : $CLMSGDEF ;DEFINE ERROR/STATUS VALUES
0000 97 : $JPIDEF ;GET JOB PROCESS INFORMATION DEFINITIONS
0000 98 :
0000 99 :
0000 100 : Define system structures used
0000 101 :
0000 102 : $DQFDEF ; format of disk quota file record
0000 103 : $FIBDEF ; format of FIB (ACP interface block)
0000 104 :
0000 105 :
0000 106 : LOCAL DATA
0000 107 :
0000 108 : .PSECT DCL$ZCODE,BYTE,RD,NOWRT
0000 109 ACCESS: ;ALLOWED ACCESS DESIGNATORS
44 45 57 52 0000 110 .ASCII /RWED/ ;
0004 111 STRINGMSG: ;TABLE DISPLAY CONTROL STRING
53 41 21 43 41 21 53 41 21 20 20 00' 0004 112 .ASCIC ' !AS!AC!AS = '!AS''
22 53 41 21 22 20 3D 20 0010
13 0004
0018 113 BINARYMSG:
53 41 21 43 41 21 53 41 21 20 20 00' 0018 114 .ASCIC ' !AS!AC!AS = !SL Hex = !-!XL Octal = !-!OL'
78 65 48 20 20 20 4C 53 21 20 3D 20 0024
63 4F 20 20 4C 58 21 2D 21 20 3D 20 0030
4C 4F 21 2D 21 20 3D 20 6C 61 74 003C
2E 0018
0047 115 NOACCESS: ;NO ACCESS ALLOWED DESIGNATOR
53 53 45 43 43 41 20 4F 4E 0047 116 .ASCII /NO ACCESS/
0050 117 NOACCESSEND: ;
0050 118 PROTECTMSG: ;DEFAULT PROTECTION CONTROL STRING
44 41 21 3D 4D 45 54 53 59 53 20 20 0050 119 .ASCII / SYSTEM=!AD, OWNER=!AD, GROUP=!AD, WORLD=!AD/ ;
2C 44 41 21 3D 52 45 4E 57 4F 20 2C 005C
20 2C 44 41 21 3D 50 55 4F 52 47 20 0068
44 41 21 3D 44 4C 52 4F 57 0074
007D 120 PROTECTEND: ;
007D 121 STATUS_MSG: ;
007D 122 .ASCII - ;
20 6E 6F 20 73 75 74 61 74 53 20 20 007D 123 & Status on !XD!5* Elapsed CPU :!XD&;
70 61 6C 45 20 2A 35 21 44 25 21 20 0089
44 25 21 3A 20 55 50 43 20 64 65 73 0095
00A1 124 .ASCII - ;
43 20 20 20 20 4C 55 39 21 3A 20 4F 00AD 125 &!/ Buff. I/O :!9UL Cur. ws. : !5UW!4* Open files : !5UW&;
20 20 20 3A 20 2E 73 77 20 2E 72 75 00B9
6E 65 70 4F 20 2A 34 21 57 55 35 21 00C5
20 20 20 20 3A 20 73 65 6C 69 66 20 00D1
57 55 35 21 20 00DD
00E2 126 .ASCII - ;
4F 2F 49 20 2E 72 69 44 20 20 2F 21 00E2 127 &!/ Dir. I/O : !9UL Phys. Mem. : !5UW!4* Page Faults :!9UL&;
50 20 20 20 20 4C 55 39 21 20 3A 20 00EE
20 3A 20 2E 6D 65 4D 20 2E 73 79 68 00FA

```

```

65 67 61 50 20 2A 34 21 57 55 35 21 0106
55 39 21 3A 20 73 74 6C 75 61 46 20 0112
                                4C 011E
                                011F
00000000 011F 128 STATUS_END:
                                011F 129 TEXT_T_NULLSTR:
                                0123 130 .LONG 0
00000000 0123 131 JPI_CODES:
0407 0123 132 FAO_CPUTIM=<.-JPI_CODES>*2
040C 0123 133 .WORD JPI$CPUTIM
00000008 0127 134 FAO_WSSIZE=<.-JPI_CODES>*2
0411 0127 135 .WORD JPI$WSSIZE
0000000C 0129 136 FAO_FILLM=<.-JPI_CODES>*2
040F 0129 137 .WORD JPI$FILLM
040B 012B 138 .WORD JPI$DIRIO
00000014 012D 139 FAO_GPGCNT=<.-JPI_CODES>*2
030C 012D 140 .WORD JPI$GPGCNT
040A 012F 141 .WORD JPI$PAGEFLTS
0000001C 0131 142 FAO_FILCNT=<.-JPI_CODES>*2
0000001C 0131 143 .WORD JPI$FILCNT
0314 0131 144 FAO_Q_CPU=FAO_FILCNT
00000020 0133 145 .WORD JPI$Q_CPU
030D 0133 146 FAO_PPGCNT=<.-JPI_CODES>*2
00000009 0135 147 .WORD JPI$PPGCNT
                                148 JPI_ARGS = <.-JPI_CODES>/2

```

```

:NULL STRING (COUNTED & DESCRIPTOR)
:LIST OF JPI ITEM CODES
:FOR SHOW STATUS
:
:
:
:USE THIS LOCATION TWICE
:
:COMPUTE NUMBER OF JPI CODES

```

```

0135 150      .SBTTL  SHOW DEFAULT
0135 151      :+
0135 152      :DCL$SHOWDEF - SHOW DEFAULT
0135 153      :
0135 154      :THIS ROUTINE IS CALLED AS AN INTERNAL COMMAND TO EXECUTE THE SHOW DEFAULT
0135 155      :DCLS COMMAND.
0135 156      :
0135 157      :INPUTS:
0135 158      :
0135 159      :R8 = ADDRESS OF SCRATCH BUFFER DESCRIPTOR.
0135 160      :R9 = ADDRESS OF SCRATCH STACK.
0135 161      :R10 = BASE ADDRESS OF COMMAND WORK AREA.
0135 162      :R11 = BASE ADDRESS OF PROCESS WORK AREA.
0135 163      :
0135 164      :OUTPUTS:
0135 165      :
0135 166      :THE CURRENT DEFAULT DEVICE AND DIRECTORY IS WRITTEN TO THE OUTPUT STREAM.
0135 167      :-
0135 168      :
0135 169      DCL$SHOWDEF::          :SHOW DEVICE AND DIRECTORY INFORMATION
0135 170      BSBB          DCL$SHOWDEFINT      :CALL ROUTINE THAT DOES THE WORK
0137 171      BRW          EXTMSG              :
013A 172      :
013A 173      DCL$SHOWDEFINT::
013A 174      ADDL          #2,4(R8)           :MAKE ROOM FOR LEADING SPACES
013E 175      MOVAB        W^DCL$T_DSKNAM,R1  :ADDRESS OF DISK NAME COUNTED STRING
0143 176      MOVZBL       (R1)+,R0           :GET QUAD WORD DESCRIPTOR
0146 177      PUSHR        #^M<R0,R1>        :PUT QUAD WORD DESCRIPTOR IN STACK
0148 178      CLRQ         -(SP)              :FIRST TWO ARGS ARE ZERO
014A 179      CLRL         -(SP)              :ALSO THIRD IS ZERO
014C 180      PUSHAQ       (R8)               :ADDRESS OF BUFFER DESCRIPTOR
014E 181      PUSHAQ       (R8)               :PLACE TO RESTORE LENGTH
0150 182      PUSHAQ       20(SP)             :ADDRESS OF DESCRIPTOR IN STACK
0153 183      CALLS        #8,@#SYS$STRNLOG   :TRANSLATE AND CLEAR THE STACK
015A 184      ADDL3        (R8)+,(R8),R2      :FIND FIRST BYTE AFTER DEVICE NAME
015E 185      MOVAB        @#PIO$GT_DDSTRING,R1 :GET ADDRESS OF DEFAULT DIRECTORY STRING
0165 186      MOVZBL       (R1)+,R0           :GET LENGTH OF DEFAULT DIRECTORY STRING
0168 187      MOVQ         R0,(R1),(R2)       :INSERT DEFAULT DIRECTORY STRING
016C 188      MOVL        (R8),R2            :POINT AT START OF DEVICE NAME
016F 189      MOVW        #^A/ / -(R2)       :INSERT LEADER FOR NEATNESS
0174 190      SUBL3        R2,R3,R1           :FIND LENGTH OF STRING
0178 191      RSB          :

```

```

03 10
02A7 31
04 AB 02 C0
51 0000'CF 9E 013E 175
50 81 9A 0143 176
03 BB 0146 177
7E 7C 0148 178
7E D4 014A 179
68 7F 014C 180
68 3F 014E 181
14 AE 7F 0150 182
00000000'9F 08 FB 0153 183
52 68 88 C1 015A 184
51 00000000'9F 9E 015E 185
50 81 9A 0165 186
62 61 50 28 0168 187
52 68 D0 016C 188
72 2020 8F B0 016F 189
51 53 52 C3 0174 190
05 0178 191

```



```

0179 193 .SBTTL SHOW PROTECTION
0179 194 :+
0179 195 : DCL$SHOWPROT - SHOW PROTECTION
0179 196 :
0179 197 : THIS ROUTINE IS CALLED AS AN INTERNAL COMMAND TO EXECUTE THE SHOW PROTECTION
0179 198 : DCLS COMMAND.
0179 199 :
0179 200 : INPUTS:
0179 201 :
0179 202 : R8 = ADDRESS OF SCRATCH BUFFER DESCRIPTOR.
0179 203 : R9 = ADDRESS OF SCRATCH STACK.
0179 204 : R10 = BASE ADDRESS OF COMMAND WORK AREA.
0179 205 : R11 = BASE ADDRESS OF PROCESS WORK AREA.
0179 206 :
0179 207 : OUTPUTS:
0179 208 :
0179 209 : THE CURRENT DEFAULT PROTECTION IS CONVERTED TO ASCII AND WRITTEN TO
0179 210 : THE OUTPUT STREAM.
0179 211 : -
0179 212 :
0179 213 DCL$SHOWPROT:: :SHOW PROTECTION INFORMATION
0179 214 BSBB DCL$SHOWPROTINT :CALL ROUTINE THAT DOES THE WORK
0179 215 BRW EXTMSG :
0179 216
0179 217 DCL$SHOWPROTINT:: :
0179 218 CLRL -(SP) :WHERE TO RETURN PROTECTION
0179 219 MOVL SP, -(SP) :NOTE WHERE PROTECTION IS TO BE PUT
0179 220 CLRL -(SP) :DON'T WANT TO SET PROTECTION
0179 221 CALLS #2, @#SYSS$SETDFPROT :GET DEFAULT PROTECTION
0179 222 MOVL (SP)+, R0 :COPY PROTECTION TO USEFUL REG
0179 223 MOVZBL #12, R7 :SET OUTER LOOP INDEX
0179 224 10$: CLRL -(SP) :ALLOCATE SPACE FOR ACCESS DESIGNATORS
0179 225 MOVAB NOACCESS, -(R9) :ASSUME NO ACCESS ALLOWED
0179 226 MOVZBL #NOACCESS$END-NOACCESS, -(R9) :
0179 227 EXTZV R7, #4, R0, R1 :EXTRACT NEXT PROTECTION FIELD
0179 228 MCOML R1, R2 :COMPLEMENT PROTECTION FIELD
0179 229 BITL #^XF, R2 :ALL ACCESS DENIED?
0179 230 BEQL 40$ :IF EQL YES
0179 231 CLRL R6 :CLEAR INNER LOOP INDEX
0179 232 MOVAB (SP), 4(R9) :SET ADDRESS OF ACCESS DESIGNATORS
0179 233 CLRL (R9) :CLEAR COUNT OF ACCESS DESIGNATORS
0179 234 20$: BBS R6, R1, 30$ :IF SET, ACCESS DENIED
0179 235 MOVAB ACCESS[R6], @ (R9)[SP] :INSERT ACCESS DESIGNATOR
0179 236 INCL (R9) :INCREMENT COUNT OF ACCESS DESIGNATORS
0179 237 30$: AOBLSS #4, R6, 20$ :ANY MORE TO CHECK?
0179 238 40$: ACBB #0, #-4, R7, 10$ :ANY MORE FIELDS TO CHECK?
0179 239 PUSHAB PROTECTMSG :BUILD FORMAT CONTROL STRING DESCRIPTOR
0179 240 PUSHL #PROTECTEND-PROTECTMSG :
0179 241 MOVL SP, R0 :COPY ADDRESS OF CONTROL STRING DESCRIPTOR
0179 242 $FAOL_S (R0), (R8), (R8), (R9) :FORMAT PROTECTION MESSAGE
0179 243 MOVQ (R8), R1 :RETRIEVE OUTPUT MESSAGE PARAMETERS
0179 244 ADDL #6*4, SP :CLEAN STACK
0179 245 RSB :

```

```

03 10
0263 31
7E D4
7E DO
7E D4
00000000'9F 02 FB
50 8E DO
57 0C 9A
79 FEAF CF 9E
79 09 9A
51 50 04 57 EF
52 51 D2
52 0F D3
1A 13
04 A9 6E 9E
0A 51 69 D4
00 B94E FE46 CF46 90
EE 56 04 F2
FFC8 57 FC 8F 00 9D
FE82 CF 9F
50 2D DD
50 5E DO
51 68 7D
5E 18 C0
05 01E8

```

```

01E9 247 .SBTTL SHOW SYMBOL TABLE ENTRIES
01E9 248
01E9 249 :+ DCL$SHOWSYMBOL - SHOW SYMBOL TABLE ENTRIES
01E9 250 :
01E9 251 : THIS ROUTINE IS CALLED AS AN INTERNAL COMMAND TO EXECUTE THE SHOW SYMBOL
01E9 252 : TABLE ENTRIES DCLS COMMAND.
01E9 253 :
01E9 254 : INPUTS:
01E9 255 :
01E9 256 : R8 = ADDRESS OF SCRATCH BUFFER DESCRIPTOR.
01E9 257 : R9 = ADDRESS OF SCRATCH STACK.
01E9 258 : R10 = BASE ADDRESS OF COMMAND WORK AREA.
01E9 259 : R11 = BASE ADDRESS OF PROCESS WORK AREA.
01E9 260 :
01E9 261 : OUTPUTS:
01E9 262 :
01E9 263 : THE SPECIFIED SYMBOL TABLE ENTRY OR ALL SYMBOL TABLE ENTRIES FROM
01E9 264 : EITHER THE LOCAL OR GLOBAL SYMBOL TABLE ARE WRITTEN TO THE OUTPUT
01E9 265 : STREAM.
01E9 266 :
01E9 267 :
01E9 268 DCL$SHOWSYMBOL::
01E9 269 CLRL -(SP) ;SHOW SYMBOL TABLE ENTRIES
01EB 270 CLRB @4(R8) ;ZERO TABLE LISTHEAD ADDRESS
01EE 271 CLRQ R6 ;ASSUME /LOG
10$: BSBW DCL$GETDVAL ;ZERO DESCRIPTOR OF SYMBOL NAME
01F3 272 CMPB #PTR_K_ENDLINE,R5 ;GET NEXT DESCRIPTOR VALUE
273 BEQL 30$ ;END OF LINE?
01F6 274 CMPB #PTR_K_PARAMETR,R5 ;BRANCH IF SO
275 BNEQ 25$ ;PARAMETER?
01FD 276 ASSUME PTR_V_KEYWORD EQ 21 ;BRANCH IF QUALIFIER
01FD 277 BBS #1,R3,10$ ;SKIP IF OPTION KEYWORD
56 51 7D 0201 279 MOVQ R1,R6 ;SAVE DESCRIPTOR OF SYMBOL NAME
EA 11 0204 280 BRB 10$
00'8F 51 91 0209 282 25$: BSBW DCL$GETNVAL ;GET QUALIFIER NUMBER
1D 13 020D 283 CMPB R1,#CLISK_SHSY_GLOB ;/GLOBAL?
00'8F 51 91 020F 284 BEQL 20$ ;BRANCH IF SO
11 13 0213 285 CMPB R1,#CLISK_SHSY_LOCA ;/LOCAL?
00'8F 51 91 0215 286 BEQL 15$ ;BRANCH IF SO
D5 12 0219 287 CMPB R1,#CLISK_SHSY_LOG ;/LOG?
04 B8 94 021B 288 BNEQ 10$ ;IF NOT, IGNORE IT
CF 53 E9 021E 289 CLRB @4(R8) ;ASSUME /LOG
04 B8 96 0221 290 BLBC R3,10$ ;IT WAS /LOG. FLAG OK AS IS
CA 11 0224 291 INCB @4(R8) ;IT WAS /NOLOG. CHANGE FLAG
6E 38 AB 7E 0226 292 15$: BRB 10$ ;GET NEXT DESC. VALUE
C4 11 022A 293 MOVQ PRC_Q_LOCAL(R11),(SP) ;SET ADDRESS OF LOCAL SYMBOL TABLE
6E 28 AB 7E 022C 294 20$: BRB 10$ ;SET ADDRESS OF GLOBAL SYMBOL TABLE
BE 11 0230 295 BRB 10$
68 B7 0232 296 30$: DECB (R8) ;REMOVE /LOG FLAG FROM SCRATCH BUFFER
04 A8 D6 0234 297 INCL 4(R8)
51 56 7D 0237 298 MOVQ R6,R1 ;GET DESCRIPTOR OF SYMBOL NAME (IF ANY)
56 8ED0 023A 299 POPL R6 ;GET ADDRESS OF SYMBOL TABLE LISTHEAD
51 D5 023D 300 TSTL R1 ;ANY SYMBOL NAME SPECIFIED?
28 12 023F 301 BNEQ 40$ ;IF SO, DISPLAY IT
0241 302 ;OTHERWISE, ASSUME /ALL
0241 303

```

```

0241 304 :
0241 305 : DISPLAY ALL SYMBOL ENTRIES
0241 306 :
0241 307 :
56 56 D5 0241 308 TSTL R6 ;ANY SYMBOL TABLE SPECIFIED?
04 12 0243 309 BNEQ 32$ ;BR IF TABLE ADDRESS PRESENT
56 38 AB 7E 0245 310 MOVAQ PRC Q_LOCAL(R11),R6 ;ASSUME /LOCAL
56 56 D0 0249 311 32$: MOVL R6,AP- ;COPY ADDRESS OF NAME TABLE LISTHEAD
56 66 D0 024C 312 36$: MOVL (R6),R6 ;GET ADDRESS OF NEXT ENTRY
5C 56 D1 024F 313 CMPL R6,AP ;END OF TABLE?
14 13 0252 314 BEQL 38$ ;IF EQL YES
68 DD 0254 315 PUSHL (R8) ;SAVE SIZE OF SCRATCH BUFFER
0240 8F BB 0256 316 PUSHR #^M<R6,R9> ;SAVE REGISTERS
53 56 D0 025A 317 MOVL R6,R3 ;COPY SYMBOL POINTER
37 10 025D 318 BSBB DISPSYMB ;FORMAT AND OUTPUT ENTRY
0240 8F BA 025F 319 POPR #^M<R6,R9> ;RESTORE REGISTERS
68 8E D0 0263 320 MOVL (SP)+,(R8) ;RESET SCRATCH BUFFER DESCRIPTOR SIZE
E4 11 0266 321 BRB 36$ :
05 0268 322 38$: RSB :
0269 323 :
0269 324 : DISPLAY SPECIFIED SYMBOL VALUE
0269 325 :
0269 326 :
0269 327 :
79 51 7D 0269 328 40$: MOVQ R1,-(R9) ;SAVE SYMBOL ENTRY DESCRIPTOR
79 79 7C 026C 329 CLRQ -(R9) ;GUESS AT UNDEFINED
50 56 D0 026E 330 MOVL R6,R0 ;GET ADDRESS OF SYMBOL TABLE LISTHEAD
08 12 0271 331 BNEQ 50$ ;IF NEQ SPECIFIC
00000000'EF 16 0273 332 JSB DCL$SEARCH ;SEARCH ALL LOCAL AND GLOBAL SYMBOL TABLES
06 11 0279 333 BRB 60$ :
00000000'EF 16 027B 334 50$: JSB DCL$SEARCHT ;SEARCH SPECIFIC SYMBOL TABLE
12 50 E8 0281 335 60$: BLBS R0,DISPSYMB ;BRANCH IF FOUND
0284 336 ERRMSG UNDSYM ;OUTPUT UNDEFINED SYMBOL MESSAGE
028E 337 STATUS NORMAL ;RETURN SUCCESSFUL
05 0295 338 RSB :
0296 339 :
0296 340 :+
0296 341 : DISPSYMB - DISPLAY THE VALUE OF A GIVEN SYMBOL
0296 342 :
0296 343 : INPUTS:
0296 344 :
0296 345 : R3 = ADDRESS OF SYMBOL TABLE ENTRY
0296 346 :
0296 347 DISPSYMB: ;FORMAT A SYMBOL
52 0C A3 9E 0296 348 MOVAB SYM T_SYMBOL(R3),R2 ;POINT TO SYMBOL NAME
51 82 9A 029A 349 MOVZBL (R2)+,R1 ;GET NAME LENGTH
79 51 7D 029D 350 MOVQ R1,-(R9) ;BUILD NAME DESCRIPTOR
57 59 D0 02A0 351 MOVL R9,R7 ;COPY SCRATCH STACK POINTER
79 51 7D 02A3 352 MOVQ R1,-(R9) ;BUILD NAME DESCRIPTOR AGAIN
67 0B A3 9A 02A6 353 MOVZBL SYM B_NONUNIQUE(R3),(R7) ;SET LENGTH OF REMAINDER
11 13 02AA 354 BEQL DISP$NORSYM ;IF EQL NOT ABBREVIATED
69 0B A3 82 02AC 355 SUBB SYM B_NONUNIQUE(R3),(R9) ;SHORTEN NAME TO ONLY UNIQUE PART
04 A7 69 C0 02B0 356 ADDL (R9),R7 ;SKIP UNIQUE PART IN REMAINDER
54 00000000'EF 9E 02B4 357 MOVAB L^DCL$GT_SYMBR,R4 ;SET THE ABBREVIATION FLAG
08 11 02BB 358 BRB DISPABR :
02BD 359 :
02BD 360 : R3 = ADDRESS OF SYMBOL TABLE ENTRY

```

```

02BD 361 : (R9) = DESCRIPTOR OF UNIQUE PORTION OF SYMBOL NAME
02BD 362 :
02BD 363 DISPNSYMSYM: :DISPLAY SYMBOL W/ NO ABBREVIATION
54 FESE CF 9E 02BD 364 MOVAB TEXT_T_NULLSTR,R4 :WRITE NULL AS NONUNIQUE DELIMITER
57 54 D0 02C2 365 MOVL R4,R7 :WRITE NULL AS NONUNIQUE PORTION
02C5 366 :
02C5 367 : R3 = ADDRESS OF SYMBOL TABLE ENTRY
02C5 368 : R4 = ADDRESS OF ASCII STRING BETWEEN UNIQUE AND NON-UNIQUE PORTIONS OF NAME
02C5 369 : R7 = ADDRESS OF DESCRIPTOR OF NONUNIQUE PORTION OF SYMBOL NAME
02C5 370 : (R9) = DESCRIPTOR OF UNIQUE PORTION OF SYMBOL NAME
02C5 371 :
02C5 372 DISPABR: :DISPLAY W/ ABBREVIATION
51 55 59 D0 02C5 373 MOVL R9,R5 :GET ADDRESS OF NAME DESCRIPTOR
51 0C A3 9A 02C8 374 MOVZBL SYM_T_SYMBOL(R3),R1 :GET LENGTH OF SYMBOL NAME
51 0D A341 9E 02CC 375 MOVAB SYM_T_SYMBOL+1(R3)[R1],R1 :GET ADDRESS OF SYMBOL VALUE
02 0A A3 91 02D1 376 CMPB SYM_B_TYPE(R3),#SYM_K_BINARY :IS SYMBOL A STRING VALUE?
0A 12 02D5 377 BNEQ 10$ :BRANCH IF SO
51 52 61 D0 02D7 378 MOVL (R1),R2 :GET BINARY VALUE
FD3A CF 9E 02DA 379 MOVAB BINARYMSG,R1 :DESCRIBE SYMBOL AS BINARY
0E 11 02DF 380 BRB 50$
50 81 3C 02E1 381 10$: MOVZWL (R1)+,R0 :CONSTRUCT DESCRIPTOR OF STRING VALUE
79 50 7D 02E4 382 MOVQ R0,-(R9) :PUSH DESCRIPTOR ONTO STACK
51 52 59 D0 02E7 383 MOVL R9,R2 :GET ADDRESS OF EQUIV DESCRIPTOR
FD16 CF 9E 02EA 384 MOVAB STRINGMSG,R1 :DESCRIBE SYMBOL AS A STRING
50 81 9A 02EF 385 50$: MOVZBL (R1)+,R0 :MAKE INTO DESCRIPTOR
79 50 7D 02F2 386 MOVQ R0,-(R9) :AND PUSH ONTO STACK
02F5 387 $FAO_S (R9),(R8),(R8),R5,R4,R7,R2 :FORMAT OUTPUT MESSAGE
59 04 A8 D0 030A 388 MOVL 4(R8),R9 :SETUP TO RETREIVE /LOG FLAG
50 0000'8F B1 030E 389 CMPW #SS$_BUFFEROVF,R0 :WAS SYMBOL TRUNCATED?
04 13 0313 390 BEQL 55$ :YES, LEAVE /LOG FLAG ALONE.
FF A9 01 90 0315 391 MOVB #1,-1(R9) :INSURE NO TRUNCATION MESSAGE
51 68 7D 0319 392 55$: MOVQ (R8),R1 :GET OUTPUT MESSAGE PARAMETERS
FCE1' 30 031C 393 BSBW DCL$MSGOUT :OUTPUT SYMBOL
OD FF A9 E8 031F 394 BLBS -1(R9),60$ :SKIP IF /NOLOG
51 00 D0 0323 395 MOVL #0,R1 :SET FAO COUNT
50 0003DE2B 8F D0 0326 396 MOVL #CLIS$SYMTRUNC,R0 :SET MESSAGE CODE
FCDO' 30 032D 397 BSBW DCL$FORMMSG :OUTPUT TRUNCATION MESSAGE
0330 398 60$: STATUS NORMAL :SET NORMAL COMPLETION STATUS
0337 399 RSB

```

```

0338 401 .SBTTL SHOW STATUS
0338 402 :+
0338 403 : DCL$SHOWSTAT - SHOW STATUS
0338 404 :
0338 405 : THIS ROUTINE IS CALLED AS AN INTERNAL COMMAND TO EXECUTE THE SHOW STATUS
0338 406 : DCLS COMMAND.
0338 407 :
0338 408 : INPUTS:
0338 409 :
0338 410 : R8 = ADDRESS OF SCRATCH BUFFER DESCRIPTOR.
0338 411 : R9 = ADDRESS OF SCRATCH STACK.
0338 412 : R10 = BASE ADDRESS OF COMMAND WORK AREA.
0338 413 : R11 = BASE ADDRESS OF PROCESS WORK AREA.
0338 414 :
0338 415 : OUTPUTS:
0338 416 :
0338 417 : VALUES CHARACTERIZING THE CURRENT PROCESS'S STATUS
0338 418 : ARE FORMATTED AND WRITTEN TO THE OUTPUT STREAM.
0338 419 :
0338 420 : SIDE EFFECTS:
0338 421 :
0338 422 : THIS ROUTINE IS USING THE COMMAND BUFFER INSTEAD OF THE SCRATCH BUFFER
0338 423 : (THE LATTER IS USED AS SCRATCH STACK)
0338 424 : -
0338 425 :
0338 426 DCL$SHOWSTAT::
0338 427 :
0338 428 :
0338 429 : BUILD DESCRIPTOR OF EXPANSION BUFFER
0338 430 :
0338 431 :
0338 432 :      58  04  A8  D0  MOVL      4(R8),R8          :BUILD DESCRIPTOR TO EXPANSION BUFFER
0338 433 :                               :IN SCRATCH BUFFER
0338 434 :      68  0400 8F  3C  MOVZWL  #WRK_C_CMDBUFSIZ,(R8)  :LENGTH OF BUFFER
0338 435 :      04  A8  F492 CA  DE  MOVAL   WRK_G_BUFFER(R10),4(R8) :POINTER TO START OF EXPANSION BUFFER
0338 436 :
0338 437 :
0338 438 : INITIALIZE POINTERS AND INDEXES
0338 439 :
0338 440 :
0338 441 :      50  04  A8  D0  MOVL      4(R8),R0          :POINTER TO SCRATCH BUFFER
0338 442 :                               :INDEX INTO JPI RESULTANT LIST
0338 443 :      59  24  C2  D4  SUBL     #JPI_ARGS*4,R9      :ALLOCATE JPI RESULTANT LIST
0338 444 :
0338 445 :
0338 446 : CONSTRUCT LIST OF JPI ITEM DESCRIPTOR BLOCKS IN SCRATCH BUFFER
0338 447 :
0338 448 :
0338 449 :      80      80  04  B0  10$:  MOVW     #4,(R0)+      :LENGTH OF RESULT (=LONGWORD)
0338 450 :      FDCB  CF41  B0  0353  450 :      MOVW     JPI_CODES[R1],(R0)+      :JPI CODE
0338 451 :      80      6941 DE  0359  451 :      MOVAL   (R9)[R1],(R0)+      :POINTER TO RESULT BUFFER (=LONGWORD)
0338 452 :      80      D4  035D  452 :      CLRL    (R0)+      :NO NEED FOR RESULTANT LENGTH
0338 453 :      ED 51  08  F3  035F  453 :      AOBLEQ  #JPI_ARGS-1,R1,10$      :REPEAT FOR EACH ITEM IN LIST
0338 454 :      60      D4  0363  454 :      CLRL    (R0)-      :END ITEM LIST
0338 455 :
0338 456 :
0338 457 : GET JOB PROCESS PARAMETERS

```

				0365	458	:			
	7E	7C		0365	459	:	CLRQ	-(SP)	:ALLOCATE AN IOSB
	50	5E	D0	0367	460	:	MOVL	SP,R0	
				036A	461	:	\$GETJPIW	S ITMLST=@4(R8),-	:GET PROCESS INFORMATION
				036A	462	:		Efn=#EXESC_SYSEFN,-	
	5E	08	C0	036A	463	:		IOSB=(R0)	
				0382	464	:	ADDL	#8,SP	:POP THE IOSB
				0385	465	:			
				0385	466	:			
				0385	467	:			
				0385	468	:			
				0385	469	:			
	0C	A9	1C	A9	C2	0385	470	SUBL	FAO_FILCNT(R9),FAO_FILLM(R9) ;COMPUTE COUNT OF OPEN FILES
	14	A9	20	A9	C0	038A	471	ADDL	FAO_PPGCNT(R9),FAO_GPGCNT(R9) ;COMPUTE TOTAL PHYSICAL MEMORY OCCUPIE
69	FFFE7960	8F	7A	038F	472	EMUL	#-100000,FAO_CPUTIM(R9),-	;CALCULATE TIME IN 100NS UNITS	
	1C	A9	00	0396	473		#0,FAO_Q_CPL(R9)		
	69	1C	A9	DE	0399	474	MOVAL	FAO_Q_CPO(R9),FAO_CPUTIM(R9) ;REPLACE BY POINTER TO QUADWORD	
			79	D4	039D	475	CLRL	-(R9)-	:INSERT SYSTEM TIME AND DATE AT TOP
				039F	476	:			
				039F	477	:			
				039F	478	:			
				039F	479	:			
				039F	480	:			
	7E	FCDA	CF	9F	039F	481	PUSHAB	STATUS MSG	:FAO MESSAGE TEXT
		A2	8F	9A	03A3	482	MOVZBL	#STATUS_END-STATUS_MSG,-(SP)	:LENGTH OF MESSAGE
	50	5E	D0	03A7	483	MOVL	SP,R0		
				03AA	484	\$FAOL_S	(R0),(R8),(R8),(R9)		:BUILD MESSAGE IN SCRATCH BUFFER
	51	68	7D	03B9	485	MOVQ	(R8),R1		:DESCRIPTOR OF MESSAGE
	5E	08	C0	03BC	486	ADDL	#2*4,SP		:CLEAN STACK
		001F	31	03BF	487	BRW	EXTMSG		:GO OUTPUT MESSAGE
				03C2	488				

```

03C2 490 .SBTTL SHOW DAYTIME
03C2 491 :+
03C2 492 : DCL$SHOWTIME - SHOW DAYTIME
03C2 493 :
03C2 494 : THIS ROUTINE IS CALLED AS AN INTERNAL COMMAND TO EXECUTE THE SHOW DAYTIME
03C2 495 : DCLS COMMAND.
03C2 496 :
03C2 497 : INPUTS:
03C2 498 :
03C2 499 : R8 = ADDRESS OF SCRATCH BUFFER DESCRIPTOR.
03C2 500 : R9 = ADDRESS OF SCRATCH STACK.
03C2 501 : R10 = BASE ADDRESS OF COMMAND WORK AREA.
03C2 502 : R11 = BASE ADDRESS OF PROCESS WORK AREA.
03C2 503 :
03C2 504 : OUTPUTS:
03C2 505 :
03C2 506 : THE CURRENT TIME AND DATE ARE CONVERTED TO ASCII AND WRITTEN TO THE
03C2 507 : OUTPUT STREAM.
03C2 508 : -
03C2 509 :
03C2 510 DCL$SHOWTIME::
52 04 A8 D0 03C2 511 MOVL 4(R8),R2 ;SHOW TIME AND DATE INFORMATION
62 2020 BF B0 03C6 512 MOVW #^A/ /,(R2) ;GET ADDRESS OF SCRATCH BUFFER
04 A8 02 C0 03CB 513 ADDL #2,4(R8) ;INSERT LEADING BLANKS
51 16 3C 03CF 514 $ASCTIM,S,(R8) ;POINT PAST LEADING BLANKS
FC1C' 30 03DE 515 MOVZWL #2,R1 ;CONVERT CURRENT TIME TO ASCII
03E1 516 EXTMSG: BSBW DCL$MSGOUT ;SET LENGTH OF OUTPUT MESSAGE
03E4 517 STATUS NORMAL ;OUTPUT MESSAGE TEXT
05 03EB 518 RSB ;SET NORMAL COMPLETION STATUS

```

S  
V

31

5

5

```

03EC 520 .SBTTL SHOW DISK QUOTA
03EC 521 :++
03EC 522 : DCL$SHOWQUOTA - SHOW DISK QUOTA
03EC 523 :
03EC 524 : THIS ROUTINE ASSIGNS A CHANNEL TO THE SPECIFIED DISK AND EXAMINES
03EC 525 : THIS QUOTA FILE ENTRY BELONGING TO THE PROCESS' UIC, AND OUTPUTS
03EC 526 : THE FORMATTED RESULT TO SYSS$OUTPUT.
03EC 527 :
03EC 528 : INPUTS:
03EC 529 :
03EC 530 : R8 = ADDRESS OF SCRATCH BUFFER DESCRIPTOR.
03EC 531 : R9 = ADDRESS OF SCRATCH STACK.
03EC 532 : R10 = BASE ADDRESS OF COMMAND WORK AREA.
03EC 533 : R11 = BASE ADDRESS OF PROCESS WORK AREA.
03EC 534 :
03EC 535 : OUTPUTS:
03EC 536 :
03EC 537 : VALUES FOR SPECIFIED OR DEFAULT DISK QUOTA ARE FORMATTED AND
03EC 538 : WRITTEN TO THE OUTPUT STREAM.
03EC 539 : --
03EC 540 :
03EC 541 :
03EC 542 : LOCAL STACK USAGE
03EC 543 :
03EC 544 : .PSECT DCL$ABS,ABS
03EC 545 :
0000 0000
0000000C 0000 546 UIC_LIST: .BLKL 3 ; GETJPI item list to read UIC
00000014 000C 547 IO STATUS: .BLKQ 1 ; I/O status block
00000054 0014 548 FIB: .BLKB FIB$C_LENGTH ; FIB for ACP function
0000005C 0054 549 FIB_DESC: .BLKQ 1 ; descriptor for FIB
00000064 005C 550 RECORD_DESC: .BLKQ 1 ; quota record buffer descriptor
00000064 0064 551 ; re-use for FAO string descriptor
00000084 0064 552 QUOTA_RECORD: .BLKB DQF$C_LENGTH ; quota record buffer
00000088 0084 553 CHANNEL: .BLKL 1 ; channel number
00000090 0088 554 DEVNAM_DESC: .BLKQ 1 ; device name descriptor
00000080 0090 555 DEVNAM: .BLKB 32 ; device name buffer
0080 556 IMPURE_SIZE:
0080 557
0080 558 :
0080 559 : FAO control strings
0080 560 :
0000 03EC 561 : .PSECT DCL$ZCODE
03EC 562 :
03EC 563 : NOTE - ONLY SINGLE BYTE UIC FIELDS
03EC 564 :
20 49 25 21 20 72 65 73 55 20 20 00' 03EC 565 FAO_STRING: .ASCIC ' User !%I has !SL blocks used, !SL !AC,'-

```

```

63 6F 6C 62 20 4C 53 21 20 73 61 68 03F8
4C 53 21 20 2C 64 65 73 75 20 73 68 0404
20 66 6F 20 20 2F 21 2C 43 41 21 20 0410
7A 69 72 6F 68 74 75 61 20 4C 53 21 041C
69 6D 72 65 70 20 64 6E 61 20 64 65 0428
61 72 64 72 65 76 6F 20 64 65 74 74 0434
6C 62 20 4C 53 21 20 66 6F 20 74 66 0440
53 41 21 20 6E 6F 20 73 68 63 6F 044C
6A 03EC
65 6C 62 61 6C 69 61 76 61 00' 0457

```

```

566 '!/ of !SL authorized and permitted overdraft of !SL blocks on !AS'
567 AVAIL: .ASCIC /available/

```



```

4E 57 41 52 44 52 45 56 4F 09 0457
00' 0461 568 OVER: .ASCIC /OVERDRAWN/
09 0461
046B 569
046B 570 DCL$SHOWQUOTA::
046B 571 ASSUME IMPURE SIZE LE WRK C_SCRSTKSIZ ;MAKE SURE SCRATCH STACK BIG ENOUGH
69 00B0 8F 59 FF50 C9 9E 046B 572 MOVAB -IMPURE_SIZE(R9),R9 ;ALLOCATE LOCAL STACK STORAGE
00 69 00 2C 0470 573 MOVCS #0,(R9),#0,#IMPURE_SIZE,(R9) ;INITIALLY ZERO
5C A9 20 D0 0478 574 MOVL #DQF$C_LENGTH,RECORD_DESC(R9) ;SET QUOTA RECORD
54 A9 60 A9 64 A9 9E 047C 575 MOVAB QUOTA_RECORD(R9),RECORD_DESC+4(R9) ;DESCRIPTOR
00000040 8F D0 0481 576 MOVL #FIB$C_LENGTH,FIB_DESC(R9) ;AND DESCRIPTOR FOR ACP
58 A9 14 A9 9E 0489 577 MOVAB FIB(R9),FIB_DESC+4(R9) ;FUNCTION
7E D4 048E 578 CLRL -(SP) ;ASSUME NO UIC SPECIFIED
0490 579
0490 580 ; GET NEXT TOKEN
0490 581
0490 582 10$:
FB6D' 30 0490 583 BSBW DCL$GETDVAL ;GET NEXT RESULT PARSE DESCRIPTOR
0493 584 20$:
55 04 91 0493 585 CMPB #PTR_K_ENDLINE,R5 ;END OF LINE?
42 13 0496 586 BEQL 100$ ;IF EQL YES
55 00 91 0498 587 CMPB #PTR_K_COMDQUAL,R5 ;COMMAND QUALIFIER?
F3 12 049B 588 BNEQ 10$ ;IF NEQ NO, IGNORE IT
049D 589
049D 590 ; PARSE DISK NAME QUALIFIER VALUE
049D 591
51 FB60' 30 049D 592 BSBW DCL$GETNVAL ;GET QUALIFIER NUMBER
00'8F 91 04A0 593 CMPB #CLISK_SHQO_DISK,R1 ;DISK QUALIFIER?
OF 12 04A4 594 BNEQ 40$ ;IF NEQ NO
FB57' 30 04A6 595 BSBW DCL$GETDVAL ;ATTEMPT TO GET VALUE
55 02 91 04A9 596 CMPB #PTR_K_QUALVALU,R5 ;IS IT A VALUE DESCRIPTOR
E5 12 04AC 597 BNEQ 20$ ;IF NEQ NO
0088 C9 51 7D 04AE 598 MOVQ R1,DEVNAM_DESC(R9) ;STORE DEVICE DESCRIPTOR
DB 11 04B3 599 BRB 10$ ;AND GO AGAIN
04B5 600
04B5 601 ; PARSE UIC QUALIFIER VALUE
04B5 602
51 00'8F 91 04B5 603 40$: CMPB #CLISK_SHQO_USER,R1 ;USER QUALIFIER?
D5 12 04B9 604 BNEQ 10$ ;IF NEQ NO, IGNORE IT
FB42' 30 04BB 605 BSBW DCL$GETDVAL ;ATTEMPT TO GET VALUE
55 02 91 04BE 606 CMPB #PTR_K_QUALVALU,R5 ;QUALIFIER VALUE?
D0 12 04C1 607 BNEQ 20$ ;GO TRY SOMETHING ELSE
6E D6 04C3 608 INCL (SP) ;FLAG UIC WAS SPECIFIED
54 51 7D 04C5 609 MOVQ R1,R4 ;SAVE DESCRIPTOR OF UIC STRING
FB35' 30 04C8 610 BSBW DCL$CVTUIC ;CONVERT UIC TO LONGWORD
07 50 E9 04CB 611 BLBC R0,50$ ;BRANCH IF ERROR
68 A9 51 D0 04CE 612 MOVL R1,QUOTA_RECORD+DQF$UIC(R9);SAVE UIC NUMBER
FFBB 31 04D2 613 BRW 10$ ;GO FOR ANOTHER RESULT PARSE PIECE
04D5 614
04D5 615
04D5 616 ; RETURN ERROR PARSING UIC
04D5 617
8E D5 04D5 618 50$: TSTL (SP)+ ;RESTORE STACK
0124 31 04D7 619 BRW 99$ ;EXIT WITH ERROR
04DA 620
04DA 621
04DA 622 ; IF NO UIC SPECIFIED ON COMMAND, GET CURRENT PROCESS UIC

```

```

04DA 623
04DA 624
69 03040004 8E D5 04DA 625
04 A9 68 A9 2B 12 04DA 626
8F D0 04DC 627
7E 7C 04DE 628
50 5E D0 04E5 629
04EF 630
04EF 631
04EF 632
5E 08 C0 04EF 633
0506 634
0509 635
0509 636
0509 637
0509 638
0509 639
0509 640
52 0088 C9 9E 0509 641
62 B5 050E 642
2B 12 0510 643
62 20 D0 0512 644
04 A2 0090 C9 9E 0515 645
51 0000 CF 9E 051B 646
50 81 9A 0520 647
03 BB 0523 648
7E 7C 0525 649
7E D4 0527 650
62 7F 0529 651
62 3F 052B 652
14 AE 7F 052D 653
00000000 9F 08 FB 0530 654
03 50 EB 0537 655
00C1 31 053A 656
053D 657
53 04 A2 62 C1 053D 658
73 3A 91 0542 659
02 12 0545 660
62 D7 0547 661
0549 662
0549 663
0549 664
0549 665
0549 666
03 50 EB 0549 667
00A0 31 0558 668
055B 669
055E 670
055E 671
2A A9 0C B0 055E 672
50 5C A9 9E 0562 673
0566 674
0566 675
0566 676
0566 677
0566 678
0566 679

100$:
TSTL (SP)+ ;WAS UIC SPECIFIED?
BNEQ 110$ ;IF NEQ YES
MOVL #4+<JPI$ UIC@16>,UIC_LIST(R9);SET UP GETJPI FOR
MOVAB QUOTA_RECORD+DQF$$_UIC(R9),UIC_LIST+4(R9);FOR GETTING UIC
CLRQ -(SP) ;ALLOCATE AN IOSB
MOVL SP,R0
$GETJPIW S ITMLST=UIC_LIST(R9),-;GET PROCESS INFORMATION
EFN=#EXESC_SYSEFN,-
IOSB=(R0)
ADDL #8,SP ;POP THE IOSB

: IF NO DISK NAME SPECIFIED ON COMMAND, TRANSLATE SYSSDISK TO GET CURRENT
: DEFAULT DISK NAME

110$:
MOVAB DEVNAM_DESC(R9),R2 ;GET ADDRESS OF DEVICE DESCRIPTOR
TSTW (R2) ;DEVICE SPECIFIED?
BNEQ 115$ ;IF NEQ YES
MOVL #32,(R2) ;SET CHARACTER DESCRIPTOR
MOVAB DEVNAM(R9),4(R2) ;FOR TRANSLATION OF SYSSDISK
MOVAB W^DCLST_DSKNAM,R1 ;ADDRESS OF DISK NAME COUNTED STRING
MOVZBL (R1)+,R0 ;GET QUAD WORD DESCRIPTOR
PUSHR #^M<R0,R1> ;PUT QUAD WORD DESCRIPTOR IN STACK
CLRQ -(SP) ;FIRST TWO ARGS ARE ZERO
CLRL -(SP) ;ALSO THIRD IS ZERO
PUSHAQ (R2) ;ADDRESS OF BUFFER DESCRIPTOR
PUSHAQ (R2) ;PLACE TO RESTORE LENGTH
PUSHAQ 20(SP) ;ADDRESS OF DESCRIPTOR IN STACK
CALLS #8,#SYSSTRNLOG ;TRANSLATE AND CLEAR THE STACK
BLBS R0,115$ ;IF LBS SUCCESSFUL
BRW 99$ ;ELSE EXIT WITH ERROR

115$:
ADDL3 (R2),4(R2),R3 ;GET ADDRESS OF LAST CHARACTER
CMPB #^A/:/,-(R3) ;OF DEVICE. IS IT COLON?
BNEQ 117$ ;IF NEQ NO
DECL (R2) ;DISCOUNT TRAILING COLON

: ASSIGN A CHANNEL TO THE DISK AND GET DISK QUOTA RECORD FROM ACP

117$:
$ASSIGN_S DEVNAM = (R2),- ;ASSIGN A CHANNEL TO SPECIFIED
CHAN = CHANNEL(R9) ;DEVICE
BLBS R0,120$ ;IF LBS SUCCESSFUL
BRW 99$ ;ELSE EXIT WITH ERROR

120$:
MOVW #FIB$C_EXA_QUOTA,FIB+FIB$W_CNTRLFUNC(R9) ; issue ACP function to read quota record
MOVAB RECORD_DESC(R9),R0
$QIOW_S CHAN = CHANNEL(R9),-
FUNC = #IOS_ACPCONTROL,-
IOSB = IO_STATUS(R9),-
EFN = #EXESC_SYSEFN,-
P1 = FIB_DESC(R9),-
P2 = R0,-

```

```

          0566 680
50 5D 50 E9 058D 681          BLBC R0,98$          P4 = R0
          0590 682          MOVZWL IO,STATUS(R9),R0      :get I/O status and check it
50 0C A9 3C 0594 683          BLBC R0,98$
          0597 684          :
          0597 685          :
          0597 686          :
50 0088 C9 9E 0597 687          MOVAB DEVNAM,DESC(R9),R0      :get address of dev. name descriptor
51 FE4C CF 9E 059C 688          MOVAB FAO,STRING,R1          :point to FAO string
5C A9 81 9A 05A1 689          MOVZBL (R1)+,RECORD,DESC(R9)      :set its length
60 A9 51 D0 05A5 690          MOVL R1,RECORD,DESC+4(R9)      :and address
51 6C A9 D0 05A9 691          MOVL QUOTA,RECORD+DQFSL_USAGE(R9),R1 :GET BLOCKS IN USE
52 70 A9 D0 05AD 692          MOVL QUOTA,RECORD+DQFSL_PERMQUOTA(R9),R2 :AND PERMANENT QUOTA
54 FEA2 CF 9E 05B1 693          MOVAB AVAIL,R4          :ASSUME NOT OVERDRAWN
53 52 51 C3 05B6 694          SUBL3 R1,R2,R3          :COMPUTE NUMBER REMAINING
          05BA 695          BGEQ 130$          :IF GEQ THEN NOT OVERDRAWN
          05BC 696          MNEGL R3,R3          :MAKE OVERDRAFT POSITIVE
54 53 53 CE 05BF 697          MOVAB OVER,R4          :SET KEYWORD ADDRESS
          05C4 698 130$:
          05C4 699          MOVL QUOTA,RECORD+DQFSL_OVERDRAFT(R9),R5 :get overdraft limit
          05C8 700          SFAO_S CTRSTR = RECORD,DESC(R9),-      :call FAO to build the message
          05C8 701          OUTLEN = (R8),-
          05C8 702          OUTBUF = (R8),-
          05C8 703          P1 = QUOTA,RECORD+DQFSL_UIC(R9),-
          05C8 704          P2 = R1,-          :BLOCKS USED
          05C8 705          P3 = R3,-          :BLOCKS REMAINING OR OVERDRAWN
          05C8 706          P4 = R4,-          :"available" OR "OVERDRAWN"
          05C8 707          P5 = R2,-          :AUTHORIZED QUOTA
          05C8 708          P6 = R5,-          :AUTHORIZED OVERDRAFT
          05C8 709          P7 = R0          :DEVICE NAME
          05E5 710
          05E5 711
51 06 10 05E5 712          BSBB 98$          :deassign the channel
          05E7 713          MOVQ (R8),R1          :get string descriptor
          FDF4 31 05EA 714          BRW EXTM$G          :output and exit
          05ED 715          :
          05ED 716          : DEASSIGN THE CHANNEL TO THE DISK
          05ED 717          :
          05ED 718 98$:          PUSHL R0          :save return status
          05EF 719          $DASSGN_S CHAN = CHANNEL(R9)
50 8E D0 05FB 720          MOVL -(SP)+,R0          :restore return status
          05FE 721 99$:
          05FE 722          RSB
          05FF 723
          05FF 724          .END

```

SHOW  
Symbol table

- SHOW COMMAND EXECUTION

G 11

16-SEP-1984 00:16:12 VAX/VMS Macro V04-00  
4-SEP-1984 23:43:15 [DCL.SRC]SHOW.MAR;1

Page 17  
(8)

\$ST1	=	00000001			IMPURE SIZE	00000080		
\$ST2	=	0000000A			IOS_ACPCONTROL	*****	X	02
ACCESS		00000000	R	02	IO STATUS	0000000C		
AVAIL		00000457	R	02	JPI\$_BUFIO	=	0000040C	
BINARYMSG		00000018	R	02	JPI\$_CPUTIM	=	00000407	
CHANNEL		00000084			JPI\$_DIRIO	=	0000040B	
CLISK_SHQO_DISK		*****	X	02	JPI\$_FILCNT	=	00000314	
CLISK_SHQO_USER		*****	X	02	JPI\$_FILLM	=	0000040F	
CLISK_SHSY_GLOB		*****	X	02	JPI\$_GPGCNT	=	0000030C	
CLISK_SHSY_LOCA		*****	X	02	JPI\$_PAGEFLTS	=	0000040A	
CLISK_SHSY_LOG		*****	X	02	JPI\$_PPGCNT	=	0000030D	
CLIS_NORMAL	=	00030001			JPI\$_UIC	=	00000304	
CLIS_SYMTRUNC	=	0003DE2B			JPI\$_WSSIZE	=	00000411	
CLIS_UNDSYM	=	00038140			JPI_ARGS	=	00000009	
DCL\$CVTUIC		*****	X	02	JPI_CODES	00000123	R	02
DCL\$ERRORMSG		*****	X	02	NOACCESS	00000047	R	02
DCL\$FORMMSG		*****	X	02	NOACCESSEND	00000050	R	02
DCL\$GETDVAL		*****	X	02	OVER	00000461	R	02
DCL\$GETNVAL		*****	X	02	PIOSGT_DDSTRING	*****	X	02
DCL\$GT_SYMBR		*****	X	02	PRC_B_CONTINUE	000000F3		
DCL\$MSGOUT		*****	X	02	PRC_B_DEFRADIX	000000AE		
DCL\$SEARCH		*****	X	02	PRC_B_EXMDEPMOD	000000AD		
DCL\$SEARCHT		*****	X	02	PRC_B_EXMDEPWID	000000AC		
DCL\$SHOWDEF		00000135	RG	02	PRC_B_EXONLYL	0000012D		
DCL\$SHOWDEFINT		0000013A	RG	02	PRC_B_FLAGS2	000000AF		
DCL\$SHOWPROT		00000179	RG	02	PRC_B_IMGFLAG	00000078		
DCL\$SHOWPROTINT		0000017E	RG	02	PRC_B_OUTFLAGS	0000012C		
DCL\$SHOWQUOTA		0000046B	RG	02	PRC_B_PROMPTLEN	000000F0		
DCL\$SHOWSTAT		00000338	RG	02	PRC_C_LENGTH	00000534		
DCL\$SHOWSYMBL		000001E9	RG	02	PRC_G_COMMANDS	00000133		
DCL\$SHOWTIME		000003C2	RG	02	PRC_G_PROMPT	000000F4		
DCL\$T_DSKNAM		*****	X	02	PRC_K_LENGTH	00000534		
DEVNAM		00000090			PRC_L_CURRKEY	00000048		
DEVNAM_DESC		00000088			PRC_L_EXMDEPADR	000000A8		
DISPABR		000002C5	R	02	PRC_L_EXTARG	00000094		
DISPNORSYM		000002BD	R	02	PRC_L_EXTBLK	0000008C		
DISPSYMB		00000296	R	02	PRC_L_EXTCOD	0000009C		
DQF\$C_LENGTH	=	00000020			PRC_L_EXTHND	00000090		
DQF\$C_OVERDRAFT	=	00000010			PRC_L_EXTPRM	00000098		
DQF\$C_PERMQUOTA	=	0000000C			PRC_L_IDFLNK	0000008C		
DQF\$C_UIC	=	00000004			PRC_L_IMGACTSTS	00000080		
DQF\$C_USAGE	=	00000008			PRC_L_INDCLOCK	0000007C		
EXESC_SYSEFN		*****	X	02	PRC_L_INDEPTH	0000005C		
EXTMSG		000003E1	R	02	PRC_L_INDFAB	0000001C		
FAO_CPUTIM	=	00000000			PRC_L_INDIRAB	00000014		
FAO_FILCNT	=	0000001C			PRC_L_INDOURAB	00000018		
FAO_FILLM	=	0000000C			PRC_L_INPRAB	00000008		
FAO_GPGCNT	=	00000014			PRC_L_LASTKEY	0000004C		
FAO_PPGCNT	=	00000020			PRC_L_LSTSTATUS	00000080		
FAO_Q_CPU	=	0000001C			PRC_L_ONCTLY	00000088		
FAO_STRING		000003EC	R	02	PRC_L_ONERROR	0000006C		
FAO_WSSIZE	=	00000008			PRC_L_OUTOFBAND	00000084		
FIB		00000014			PRC_L_OUTRAB	0000000C		
FIB\$C_EXA_QUOTA	=	0000000C			PRC_L_OUTRABCTX	00000118		
FIB\$C_LENGTH	=	00000040			PRC_L_PPFLIST	00000070		
FIB\$W_CNTRLFUNC	=	00000016			PRC_L_RECALLPTR	0000012F		
FIB_DESC		00000054			PRC_L_RESTART	00000058		

SHOW  
Symbol table

- SHOW COMMAND EXECUTION

H 11

16-SEP-1984 00:16:12 VAX/VMS Macro V04-00  
4-SEP-1984 23:43:15 [DCL.SRC]SHOW.MAR;1

Page 18  
(8)

PRC_L_SAVAP	00000000		
PRC_L_SAVFP	00000004		
PRC_L_SEVERITY	00000050		
PRC_L_SPWN	00000000		
PRC_L_STACKLM	000000A4		
PRC_L_STACKPT	000000A0		
PRC_L_STATUS	00000054		
PRC_L_STS	00000084		
PRC_L_STV	00000088		
PRC_L_SYMBOL	00000060		
PRC_L_TMBX	00000074		
PRC_L_TRMLIST	00000010		
PRC_Q_ALLOCREG	00000020		
PRC_Q_COMMAND	000000E0		
PRC_Q_FLUSHTIME	00000000		
PRC_Q_GLOBAL	0000 028		
PRC_Q_IMAGENAME	0000LJD8		
PRC_Q_KEYPAD	00000040		
PRC_Q_LABEL	00000030		
PRC_Q_LOCAL	00000038		
PRC_Q_SAVEPRIV	000000E8		
PRC_T_OUTDVI	0000011C		
PRC_W_ASTIOSB	000000C6		
PRC_W_ASTRETN	000000C8		
PRC_W_ASTSTATUS	000000C4		
PRC_W_ATTMBX	0000007A		
PRC_W_FLAGS	00000068		
PRC_W_INPCHAN	00000064		
PRC_W_ONLEVEL	0000006A		
PRC_W_OUTIFI	00000114		
PRC_W_OUTISI	00000116		
PRC_W_OUTMBXCHN	000000CA		
PRC_W_OUTMBXREF	000000CE		
PRC_W_OUTMBXSIZ	000000CC		
PRC_W_PMPTCTRL	000000F1		
PRC_W_WAITIOSB	00000066		
PROTECTEND	0000007D	R	02
PROTECTMSG	00000050	R	02
PTR_B_LEVEL	00000004		
PTR_B_NUMBER	00000005		
PTR_B_PARMCNT	00000006		
PTR_B_VALUE	00000000		
PTR_C_LENGTH	0000000C		
PTR_K_CMDQUAL	= 00000000		
PTR_K_ENDLINE	= 00000004		
PTR_K_LENGTH	= 0000000C		
PTR_K_PARAMETR	= 00000003		
PTR_K_QUALVALU	= 00000002		
PTR_L_DESCR	00000000		
PTR_L_ENTITY	00000008		
PTR_V_KEYWORD	= 00000015		
QUOTA_RECORD	00000064		
RECORD_DESC	0000005C		
SS\$ BUFFEROVF	*****	X	02
STATUS_END	0000011F	R	02
STATUS_MSG	0000007D	R	02
STRINGMSG	00000004	R	02

SYM_B_FLAGS	0000000B		
SYM_B_NONUNIQUE	0000000B		
SYM_B_TYPE	0000000A		
SYM_K_BINARY	= 00000002		
SYM_L_BL	00000004		
SYM_L_FL	00000000		
SYM_T_SYMBOL	0000000C		
SYM_W_SIZE	00000008		
SYSSASCTIM	*****	GX	02
SYSSASSIGN	*****	GX	02
SYSSDASSGN	*****	GX	02
SYSSFAO	*****	X	02
SYSSFAOL	*****	GX	02
SYSSGETJPIW	*****	GX	02
SYSSQIOW	*****	GX	02
SYSSSETDFPROT	*****	X	02
SYSTRNLOG	*****	X	02
TEXT_T_NULLSTR	0000011F	R	02
UIC_LIST	00000000		
WRK_B_CMDOPT	FFFFFFFFC3		
WRK_B_MAXPARM	FFFFFFFFD0		
WRK_B_MINPARM	FFFFFFFFD1		
WRK_B_PARMCNT	FFFFFFFFCE		
WRK_B_PARMSUM	FFFFFFFFCF		
WRK_B_RECALLCNT	FFFFFFFFC5		
WRK_B_VALLEV	FFFFFFFFC4		
WRK_B_VERBTYP	FFFFFFFFC2		
WRK_C_CMDBUFSIZ	= 00000400		
WRK_C_LENGTH	FFFFFF486		
WRK_C_SCRSTKSIZ	= 00000200		
WRK_G_BUFFER	FFFFFF492		
WRK_G_INPBUF	FFFFFF896		
WRK_G_RESULT	FFFFFF9B6		
WRK_K_LENGTH	FFFFFF486		
WRK_L_CHARPTR	FFFFFF48E		
WRK_L_DISALLOW	FFFFFFE6		
WRK_L_ERRORRTN	FFFFFF9AE		
WRK_L_EXPANDPTR	FFFFFF486		
WRK_L_IMAGE	FFFFFFE2		
WRK_L_MARKPTR	FFFFFF48A		
WRK_L_PAROUT	FFFFFFD2		
WRK_L_PMPTADDR	FFFFFF9A2		
WRK_L_PROMPTRN	FFFFFF9A6		
WRK_L_PROPTR	FFFFFFC6		
WRK_L_QUABLK	FFFFFFCA		
WRK_L_READRTN	FFFFFF9AA		
WRK_L_RECALLPTR	FFFFFFEA		
WRK_L_RSLEND	FFFFFFB6		
WRK_L_RSLNXT	FFFFFFBA		
WRK_L_SAVAP	FFFFFFF8		
WRK_L_SAVFP	FFFFFFFC		
WRK_L_SAVSP	FFFFFFF4		
WRK_L_SIGNALRTN	FFFFFFD6		
WRK_L_SPECTRN	FFFFFF9B2		
WRK_L_TAB_VEC	FFFFFFDE		
WRK_L_VERB	FFFFFFBE		
WRK_W_FLAGS	FFFFFFF0		

SHOW  
Symbol table

- SHOW COMMAND EXECUTION

I 11

16-SEP-1984 00:16:12 VAX/VMS Macro V04-00  
4-SEP-1984 23:43:15 [DCL.SRC]SHOW.MAR;1

Page 19  
(8)

WRK\_W\_FLAGS2 FFFFFFFF2  
WRK\_W\_IMGCHAN FFFFFFFEE  
WRK\_W\_PMPLEN FFFFFFF99E

-----  
! 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	FFFFFFFC ( 0.)	01 ( 1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
DCL\$ZCODE	000005FF ( 1535.)	02 ( 2.)	NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC BYTE
DCL\$ABS	000000B0 ( 176.)	03 ( 3.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE

-----  
! Performance indicators !  
-----

Phase	Page faults	CPU Time	Elapsed Time
Initialization	9	00:00:00.07	00:00:01.03
Command processing	81	00:00:00.69	00:00:05.03
Pass 1	286	00:00:11.01	00:00:26.45
Symbol table sort	0	00:00:01.27	00:00:02.27
Pass 2	129	00:00:02.42	00:00:10.13
Symbol table output	27	00:00:00.17	00:00:00.19
Psect synopsis output	2	00:00:00.03	00:00:00.26
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	534	00:00:15.67	00:00:45.37

The working set limit was 1350 pages.  
56101 bytes (110 pages) of virtual memory were used to buffer the intermediate code.  
There were 50 pages of symbol table space allocated to hold 818 non-local and 32 local symbols.  
724 source lines were read in Pass 1, producing 19 object records in Pass 2.  
49 pages of virtual memory were used to define 33 macros.

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

Macro library name	Macros defined
-\$255\$DUA28:[SYSLIB]SYSBLDMLB.MLB;1	0
-\$255\$DUA28:[DCL.OBJ]DCL.MLB;1	8
-\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	1
-\$255\$DUA28:[SYSLIB]STARLET.MLB;2	17
TOTALS (all libraries)	26

1014 GETS were required to define 26 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:SHOW/OBJ=OBJ\$:SHOW MSRC\$:SHOW/UPDATE=(ENH\$:SHOW)+EXECMLS/LIB+LIB\$:DCL/LIB+SYSSLIBRARY:SYSBLDMLB/LIB



