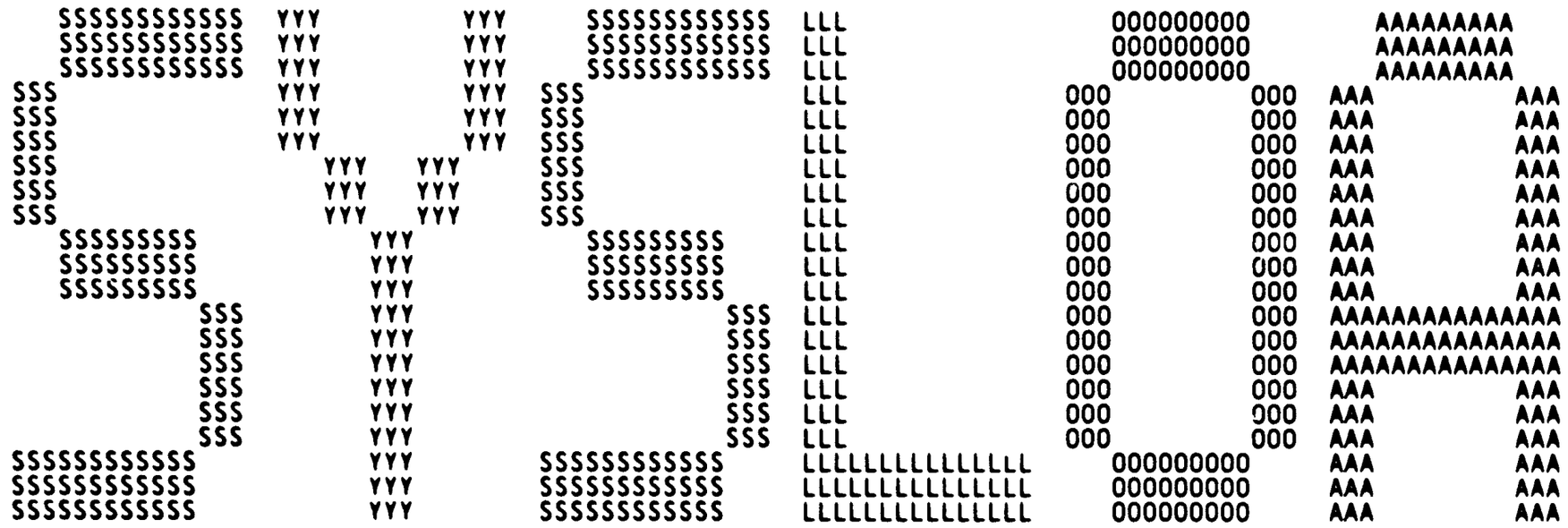
SSSSSSSSSSS YYY YYY SSSSSSSSSSS LLL 000000000 AAAAAAAAA
SSSSSSSSSSS YYY YYY SSSSSSSSSSS LLL 000000000 AAAAAAAAA
SSSSSSSSSSS YYY YYY SSSSSSSSSSS LLL 000000000 AAAAAAAAA
SSS YYY YYY SSS LLL 000 000 AAA AAA
SSS YYY YYY SSS LLL 000 000 AAA AAA
SSS YYY YYY SSS LLL 000 000 AAA AAA
SSS YYY YYY SSS LLL 000 000 AAA AAA
SSS YYY YYY SSS LLL 000 000 AAA AAA
SSS YYY YYY SSS LLL 000 000 AAA AAA
SSSSSSSSS YYY SSSSSSSSS LLL 000 000 AAA AAA
SSSSSSSSS YYY SSSSSSSSS LLL 000 000 AAA AAA
SSSSSSSSS YYY SSSSSSSSS LLL 000 000 AAA AAA
SSS YYY SSS LLL 000 000 AAAAAAAAAAAAAAA
SSS YYY SSS LLL 000 000 AAAAAAAAAAAAAAA
SSS YYY SSS LLL 000 000 AAAAAAAAAAAAAAA
SSS YYY SSS LLL 000 000 AAA AAA
SSS YYY SSS LLL 000 000 AAA AAA
SSS YYY SSS LLL 000 000 AAA AAA
SSSSSSSSSSS YYY SSSSSSSSSSS LLLLLLLLLLLLLLL 000000000 AAA AAA
SSSSSSSSSSS YYY SSSSSSSSSSS LLLLLLLLLLLLLLL 000000000 AAA AAA
SSSSSSSSSSS YYY SSSSSSSSSSS LLLLLLLLLLLLLLL 000000000 AAA AAA

```
  CCCCCCCC    SSSSSSSS  PPPPPPPP      QQQQQQ    UU      UU    OOOOOO    RRRRRRRR    UU      UU  MM      MM
  CCCCCCCC    SSSSSSSS  PPPPPPPP      QQQQQQ    UU      UU    OOOOOO    RRRRRRRR    UU      UU  MM      MM
CC           SS         PP      PP  QQ      QQ  UU      UU  OO      OO  RR      RR  UU      UU  MMMM  MMMM
CC           SS         PP      PP  QQ      QQ  UU      UU  OO      OO  RR      RR  UU      UU  MMMM  MMMM
CC           SS         PP      PP  QQ      QQ  UU      UU  OO      OO  RR      RR  UU      UU  MM  MM  MM
CC           SS         PP      PP  QQ      QQ  UU      UU  OO      OO  RR      RR  UU      UU  MM  MM  MM
CC             SSSSSS   PPPPPPPP    QQ      QQ  UU      UU  OO      OO  RRRRRRRR    UU      UU  MM      MM
CC             SSSSSS   PPPPPPPP    QQ      QQ  UU      UU  OO      OO  RRRRRRRR    UU      UU  MM      MM
CC                   SS PP          QQ  QQ  QQ  UU      UU  OO      OO  RR  RR      UU      UU  MM      MM
CC                   SS PP          QQ  QQ  QQ  UU      UU  OO      OO  RR  RR      UU      UU  MM      MM
CC                   SS PP          QQ    QQ    UU      UU  OO      OO  RR    RR    UU      UU  MM      MM
CC                   SS PP          QQ    QQ    UU      UU  OO      OO  RR    RR    UU      UU  MM      MM
  CCCCCCCC  SSSSSSSS   PP            QQQQ  QQ  UUUUUUUUUU    OOOOOO    RR      RR  UUUUUUUUUU  MM      MM
  CCCCCCCC  SSSSSSSS   PP            QQQQ  QQ  UUUUUUUUUU    OOOOOO    RR      RR  UUUUUUUUUU  MM      MM

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
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

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

Table of contents

(2) 79 Declarations
(3) 104 Own storage
(4) 174 CSP$QUORUM - Disk quorum action routine
(5) 292 REQUEST_INIT - Request initialization
(6) 332 GET_QDNAME - Get the quorum disk name
(7) 384 OPEN_FILE - Open the quorum file
(8) 444 GET_LBN - Get the quorum file logical block number
(9) 486 VALIDATE_FILE - Validate the quorum file
(10) 545 WRITE_FILE - Write the quorum file
(11) 596 CLOSE_FILE - Close the quorum file
(12) 626 REQUEST_COMPLETE - Request completion
(13) 661 CALCULATE_CHECKSUM - Calculate the quorum file checksum

```
0000     1          .TITLE  CSPQUORUM - CSP DISK QUORUM MODULE
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 ;++
0000    29 ; Facility: CSP
0000    30 ;
0000    31 ; Abstract:
0000    32 ;       This module is a "client" thread of the CSP. It is requested by the
0000    33 ;       quorum disk code whenever:
0000    34 ;
0000    35 ;               - SYSINIT did not find the quorum file,
0000    36 ;               - an I/O error occurred when reading or writing the quorum file,
0000    37 ;               - or the quorum file containes corrupt data.
0000    38 ;
0000    39 ; Enviornment:
0000    40 ;       CSP process
0000    41 ;--
0000    42 ;
0000    43 ; Author:
0000    44 ;       R. Scott Hanna, CREATION DATE: 23-Aug-1983
0000    45 ;
0000    46 ; Modified by:
0000    47 ;
0000    48 ;       V03-006 ADE0002         Alan D. Eldridge        25-Apr-1984
0000    49 ;               Use $GETDVIW_S instead of $GETDVI_S with an AST since
0000    50 ;               when an AST is delivered upon RO failure indication is
0000    51 ;               thought to be unpredictable.
0000    52 ;
0000    53 ;       V03-005 DWT0211         David W. Thiel          09-Apr-1984
0000    54 ;               Call CNX$DISK_CHANGE when CLUB$T_QDNAME is filled in.
0000    55 ;
0000    56 ;       V03-004 RSH0133         R. Scott Hanna          03-Apr-1984
0000    57 ;               Modify CLOSE_FILE to only call CSP$$WAIT when the file close
```

```
                        I/O is successfully queued.
;
;       V03-003 RSH0126         R. Scott Hanna          21-Feb-1984
;               Incorporate changes to make this algorithm a "client" of
;               the CSP. It is now requested by the QUORUM code rather than
;               periodically running to check for work. In addition this
;               routine can now create a quorum file if none exists and
;               repair one that does but contains corrupt data.
;
;       V03-002 ADE0001         Alan D. Eldridge        28-Feb-1984
;               Change name of CSP$QUORUM to CSP$QUORUM_INIT, add new
;               CSP$QUORUM entry point which is used now only as a place
;               holder.
;
;       V03-001 RSH0079         R. Scott Hanna          10-Nov-1983
;               Modify algorithm to check every 2 minutes to see if the
;               "connection" to the quorum disk has been lost. If so it
;               does the quorum file lookup again.
;--
```

```
                     0000     78
                     0000     79 .SBTTL  Declarations
                     0000     80 ;
                     0000     81 ;       Define Symbols
                     0000     82 ;
                     0000     83
                     0000     84         $ATRDEF                                 ; Attribute control block
                     0000     85         $CCBDEF                                 ; Channel control block
                     0000     86         $CLUBDEF                                ; Cluster block
                     0000     87         $CLUDCBDEF                              ; Cluster quorum disk control block
                     0000     88         $CLUQFDEF                               ; Cluster quorum file
                     0000     89         $DSCDEF                                 ; Descriptor definitions
                     0000     90         $DVIDEF                                 ; $GETDVI item list codes
                     0000     91         $FIBDEF                                 ; File information block
                     0000     92         $FIDDEF                                 ; File ID codes
                     0000     93         $IODEF                                  ; I/O function codes
                     0000     94         $IPLDEF                                 ; Interrupt priority levels
                     0000     95         $SBKDEF                                 ; Attributes statistics block
                     0000     96         $SSDEF                                  ; System service status codes
                     0000     97
                     0000     98 ; Error messages will no longer be reported after ERROR_COUNT reaches
                     0000     99 ; ERROR_THRESHOLD.
                     0000    100
0000000A             0000    101         ERROR_THRESHOLD = 10
                     0000    102
```

```
                                0000    104 .SBTTL  Own storage
                                0000    105
                             00 0000    106 THREAD_ACTIVE:  .BYTE   0                       ; Thread active flag
                                0001    107
                           0000 0001    108 QD_DESCR:       .WORD   0                       ; Quorum disk descriptor
                             0E 0003    109                 .BYTE   DSC$K_DTYPE_T
                             02 0004    110                 .BYTE   DSC$K_CLASS_D
                  00000009' 0005    111                 .LONG   QD_NAME
                                0009    112
                   00000049 0009    113 QD_NAME:        .BLKB   64                      ; Quorum disk name
                                0049    114
                  00E8 0040 0049    115 QD_ITMLST:      .WORD   64,DVI$_FULLDEVNAM      ; $GETDVI item list
                   00000009' 004D    116                 .LONG   QD_NAME
                   00000001' 0051    117                 .LONG   QD_DESCR
                   00000000 0055    118                 .LONG   0
                                0059    119
4D 55 52 4F 55 51 00000061'010E0000' 0059    120 QF_DESCR:       .ASCID  /QUORUM.DAT;1/          ; Quorum file descriptor
                 31 3B 54 41 44 2E  0067
                                006D    121
                   00000000 006D    122 CLUDCB_LBN:     .LONG   0                       ; Quorum file LBN from the CLUDCB
                                0071    123
                   00000000 0071    124 LOOKUP_LBN:     .LONG   0                       ; Quorum file LBN from file lookup
                                0075    125
                   00000000 0075    126 CHANNEL:        .LONG   0                       ; Quorum disk channel number
                                0079    127
          00000000 00000000 0079    128 IOSB:           .QUAD   0                       ; I/O status block
                                0081    129
                         00 0081    130 ERROR_COUNT:    .BYTE   0                       ; Error reported counter
                                0082    131
                   00000000 0082    132 ERROR_MESSAGE:  .LONG   0                       ; Descr addr of last error message
                                0086    133
                       0040 0086    134 FIB_DESCR:      .WORD   FIB$K_LENGTH            ; FIB descriptor
                         0E 0088    135                 .BYTE   DSC$K_DTYPE_T
                         01 0089    136                 .BYTE   DSC$K_CLASS_S
                   0000008E' 008A    137                 .LONG   FIB
                                008E    138
                   000000CE 008E    139 FIB:            .BLKB   FIB$K_LENGTH            ; File information block
                                00CE    140
                  0009 0020 00CE    141 ATTRIB_BLOCK:   .WORD   ATR$S_STATBLK,ATR$C_STATBLK ; Attribute control block
                   000000EA' 00D2    142                 .LONG   STATBLK
                  0015 0004 00D6    143                 .WORD   ATR$S_UIC,ATR$C_UIC
                   0000010A' 00DA    144                 .LONG   UIC
                  0016 0002 00DE    145                 .WORD   ATR$S_FPRO,ATR$C_FPRO
                   0000010E' 00E2    146                 .LONG   FPRO
                   00000000 00E6    147                 .LONG   0
                                00EA    148
                   0000010A 00EA    149 STATBLK:        .BLKB   ATR$S_STATBLK           ; Statistics block
                                010A    150
                  0001 0004 010A    151 UIC:            .WORD   4,1                     ; File owner UIC ([1,4])
                                010E    152
                       FF00 010E    153 FPRO:           .WORD   ^XFF00                  ; File protection (S:RWED,O:RWED)
                                0110    154
45 4C 49 46 20 20 4D 55 52 4F 55 51 0110    155 IDENT_STRING:   .ASCII  /QUORUM  FILE/          ; Quorum file ID string
                                011C    156                 ASSUME  CLUQF$S_IDENT EQ .-IDENT_STRING
                                011C    157
          00000000 00000000 011C    158 RESCHEDULE_TIMER: .QUAD 0                       ; Reschedule interval
                                0124    159
```

```
                                00000524  0124    160 QF_BUFFER:      .BLKB   CLUQF$K_BLOCKS*512      ; Quorum file buffer
                                          0524    161
57 2D 50 53 43 25 0000052C'010E0000'      0524    162 MSSG1:  .ASCID  \%CSP-W-QFNOTFOUND, Previously existing quorum file not found\
2C 44 4E 55 4F 46 54 4F 4E 46 51 2D       0532
20 79 6C 73 75 6F 69 76 65 72 50 20       053E
6F 75 71 20 67 6E 69 74 73 69 78 65       054A
74 6F 6E 20 65 6C 69 66 20 6D 75 72       0556
                  64 6E 75 6F 66 20       0562
49 2D 50 53 43 25 00000570'010E0000'      0568    163 MSSG2:  .ASCID  \%CSP-I-QFCREATED, Quorum file created\
20 2C 44 45 54 41 45 52 43 46 51 2D       0576
20 65 6C 69 66 20 6D 75 72 6F 75 51       0582
                  64 65 74 61 65 72 63    058E
57 2D 50 53 43 25 0000059D'010E0000'      0595    164 MSSG3:  .ASCID  \%CSP-W-QFCHANGED, Quorum file location has changed\
20 2C 44 45 47 4E 41 48 43 46 51 2D       05A3
20 65 6C 69 66 20 6D 75 72 6F 75 51       05AF
73 61 68 20 6E 6F 69 74 61 63 6F 6C       05BB
                  64 65 67 6E 61 68 63 20 05C7
49 2D 50 53 43 25 000005D7'010E0000'      05CF    165 MSSG4:  .ASCID  \%CSP-I-QFINIT, Quorum file initialized\
6F 75 51 20 2C 54 49 4E 49 46 51 2D       05DD
69 6E 69 20 65 6C 69 66 20 6D 75 72       05E9
                  64 65 7A 69 6C 61 69 74 05F5
45 2D 50 53 43 25 00000605'010E0000'      05FD    166 MSSG5:  .ASCID  \%CSP-E-QDASSIGN, Quorum disk assign error\
51 20 2C 4E 47 49 53 53 41 44 51 2D       060B
61 20 6B 73 69 64 20 6D 75 72 6F 75       0617
   72 6F 72 72 65 20 6E 67 69 73 73       0623
45 2D 50 53 43 25 00000636'010E0000'      062E    167 MSSG6:  .ASCID  \%CSP-E-QFOPEN, Quorum file open/create error\
6F 75 51 20 2C 4E 45 50 4F 46 51 2D       063C
65 70 6F 20 65 6C 69 66 20 6D 75 72       0648
72 72 65 20 65 74 61 65 72 63 2F 6E       0654
                              72 6F       0660
45 2D 50 53 43 25 0000066A'010E0000'      0662    168 MSSG7:  .ASCID  \%CSP-E-QFRATT, Quorum file read attributes error\
6F 75 51 20 2C 54 54 41 52 46 51 2D       0670
61 65 72 20 65 6C 69 66 20 6D 75 72       067C
73 65 74 75 62 69 72 74 74 61 20 64       0688
                  72 6F 72 72 65 20       0694
45 2D 50 53 43 25 000006A2'010E0000'      069A    169 MSSG8:  .ASCID  \%CSP-E-QFREAD, Quorum file read error\
6F 75 51 20 2C 44 41 45 52 46 51 2D       06A8
61 65 72 20 65 6C 69 66 20 6D 75 72       06B4
                  72 6F 72 72 65 20 64    06C0
45 2D 50 53 43 25 000006CF'010E0000'      06C7    170 MSSG9:  .ASCID  \%CSP-E-QFWRITE, Quorum file write error\
75 51 20 2C 45 54 49 52 57 46 51 2D       06D5
72 77 20 65 6C 69 66 20 6D 75 72 6F       06E1
          72 6F 72 72 65 20 65 74 69    06ED
45 2D 50 53 43 25 000006FE'010E0000'      06F6    171 MSSG10: .ASCID  \%CSP-E-QDGETDVI, Quorum disk $GETDVI failed\
51 20 2C 49 56 44 54 45 47 44 51 2D       0704
24 20 6B 73 69 64 20 6D 75 72 6F 75       0710
65 6C 69 61 66 20 49 56 44 54 45 47       071C
                                 64       0728
                                          0729    172
```

```
                                0729   174 .SBTTL  CSP$QUORUM - Disk quorum action routine
                                0729   175
                                0729   176 ;++
                                0729   177 ;       This routine is requested when some type of error has occured with the
                                0729   178 ;       quorum disk. It acknowledges the request and gets the quorum disk name
                                0729   179 ;       and logical block number from the CLUDCB. It then does a access with a
                                0729   180 ;       create modifier QIO to the ACP. (i.e. If the quorum file does not exist
                                0729   181 ;       it creates one.) If a quorum file exists the data in the file is
                                0729   182 ;       validated. If the quorum file contents are invalid or the quorum file
                                0729   183 ;       was created, a template quorum block is written back to the file. The
                                0729   184 ;       request is completed by updating the logical block number, the request
                                0729   185 ;       and acknowledge bits, and the state field in the CLUDCB.
                                0729   186 ;
                                0729   187 ; CALLING SEQUENCE:
                                0729   188 ;
                                0729   189 ;       JSB     CSP$QUORUM
                                0729   190 ;
                                0729   191 ; INPUTS:
                                0729   192 ;
                                0729   193 ;       NONE
                                0729   194 ;
                                0729   195 ; OUTPUT:
                                0729   196 ;
                                0729   197 ;       NONE
                                0729   198 ;--
                                0729   199 CSP$QUORUM::
03 F8D2 CF    00    E3          0729   200         BBCS    #0,THREAD_ACTIVE,1$             ; Br if thread not active
              0153    31        072F   201         BRW     17$                             ; Thread active, ignore request
          00FC 8F     BB        0732   202 1$:     PUSHR   #^M<R2,R3,R4,R5,R6,R7>          ; Save registers
                                0736   203                                                 ;
                                0736   204         $CMKRNL_S       REQUEST_INIT            ; Do the initial work
                03 50   E8      0745   205         BLBS    R0,2$                           ; Br if request necessary
                 0132   31      0748   206         BRW     16$
                                074B   207
                F8B2 CF   B5    074B   208 2$:     TSTW    QD_DESCR                        ; Do we have the quorum disk name?
                      10   12    074F   209         BNEQU   3$                              ; Br if yes
 000008BF'EF          16        0751   210         JSB     GET_QDNAME                      ; Get quorum disk name
                07 50   E8      0757   211         BLBS    R0,3$                           ; Br if success
          51    99 AF   DE      075A   212         MOVAL   MSSG10,R1                       ; Quorum disk assign error
                 009D   31      075E   213         BRW     11$
                                0761   214
                                0761   215 3$:     $ASSIGN_S       DEVNAM  = QD_DESCR,-    ; Assign a channel to
                                0761   216                             CHAN    = CHANNEL       ; the quorum disk
                08 50   E8      0772   217         BLBS    R0,4$                           ; Br if success
          51    FE84 CF   DE    0775   218         MOVAL   MSSG5,R1                        ; Quorum disk assign error
                 0081   31      077A   219         BRW     11$
                                077D   220
 0000093E'EF          16        077D   221 4$:     JSB     OPEN_FILE                       ; Open the quorum file
                07 50   E8      0783   222         BLBS    R0,5$                           ; Br if success
          51    FEA4 CF   DE    0786   223         MOVAL   MSSG6,R1                        ; Quorum disk open/create error
                   71   11      078B   224         BRB     11$
                                078D   225
          0619 8F     50   B1    078D   226 5$:     CMPW    R0,#SS$_CREATED                 ; Did we create the file?
                      2E   12    0792   227         BNEQU   8$                              ; Br if not
 000009BC'EF          16        0794   228         JSB     GET_LBN                         ; Get the quorum file LBN
                07 50   E8      079A   229         BLBS    R0,6$                           ; Br if error
          51    FEC1 CF   DE    079D   230         MOVAL   MSSG7,R1                        ; Quorum disk read attributes error
```

```
                 5A   11  07A2   231          BRB     11$
                                  07A4   232
             F8C5 CF   D5  07A4   233 6$:      TSTL    CLUDCB_LBN                      ; Did the file exist before?
                  0B   13  07A8   234          BEQLU   7$                              ; Br if not
             FD76 CF   7F  07AA   235          PUSHAQ  MSSG1
 00000000'EF      01   FB  07AE   236          CALLS   #1,CSP$TELL_OPCOM               ; Output message via opcom
                                  07B5   237
             FDAF CF   7F  07B5   238 7$:      PUSHAQ  MSSG2
 00000000'EF      01   FB  07B9   239          CALLS   #1,CSP$TELL_OPCOM               ; Output message via opcom
                  2E   11  07C0   240          BRB     10$
                                  07C2   241
             F8A7 CF   D5  07C2   242 8$:      TSTL    CLUDCB_LBN                      ; Did the file exist before?
                  14   13  07C6   243          BEQLU   9$                              ; Br if not
 F8A2 CF     F8A1 CF   D1  07C8   244          CMPL    CLUDCB_LBN,LOOKUP_LBN           ; Do the LBN's match?
                  0B   13  07CF   245          BEQLU   9$                              ; Br if they do
             FDC0 CF   7F  07D1   246          PUSHAQ  MSSG3
 00000000'EF      01   FB  07D5   247          CALLS   #1,CSP$TELL_OPCOM               ; Output message via opcom
                                  07DC   248
      00000A12'EF     16   07DC   249 9$:      JSB     VALIDATE_FILE                   ; Validate the file contents
               46 50  E8  07E2   250          BLBS    R0,13$                          ; Br if quorum file valid
                  50   D5  07E5   251          TSTL    R0                              ; I/O error?
                  07   13  07E7   252          BEQLU   10$                             ; Br if not
          51     FEAD CF   DE  07E9   253          MOVAL   MSSG8,R1                        ; Quorum file read error
                  0E   11  07EE   254          BRB     11$
                                  07F0   255
       00000A7D'EF     16   07F0   256 10$:     JSB     WRITE_FILE                      ; Write a template quorum file
               27 50  E8  07F6   257          BLBS    R0,12$                          ; Br if success
          51     FECA CF   DE  07F9   258          MOVAL   MSSG9,R1                        ; Quorum file write error
                                  07FE   259
       F87F CF    51   D1  07FE   260 11$:     CMPL    R1,ERROR_MESSAGE                ; Same error as last reported?
                  2E   13  0803   261          BEQLU   14$                             ; Br if yes
       F878 CF    51   D0  0805   262          MOVL    R1,ERROR_MESSAGE                ; Save address of last error message
       F872 CF    0A   91  080A   263          CMPB    #ERROR_THRESHOLD,ERROR_COUNT    ; Should we report this one?
                  22   13  080F   264          BEQLU   14$                             ; Br if not
             F86C CF   96  0811   265          INCB    ERROR_COUNT                     ; Increment error count
                  61   DF  0815   266          PUSHAL  (R1)                            ; Push address of message descr
 00000000'EF      01   FB  0817   267          CALLS   #1,CSP$TELL_OPCOM               ; Output message via opcom
                  13   11  081E   268          BRB     14$
                                  0820   269
             FDAB CF   7F  0820   270 12$:     PUSHAQ  MSSG4
 00000000'EF      01   FB  0824   271          CALLS   #1,CSP$TELL_OPCOM               ; Output message via opcom
             F852 CF   94  082B   272 13$:     CLRB    ERROR_COUNT                     ; Reset error count and...
             F84F CF   D4  082F   273          CLRL    ERROR_MESSAGE                   ; ...error pointer.
      00000AE4'EF     16   0833   274 14$:     JSB     CLOSE_FILE                      ; Close the quorum file
                                  0839   275          $DASSGN_S       CHAN = CHANNEL          ; Deassign the quorum disk channel
             F838 CF   95  0845   276          TSTB    ERROR_COUNT                     ; Have we completed the request?
                  23   13  0849   277          BEQLU   15$                             ; Br if yes
                                  084B   278          $SETIMR_S       DAYTIM = RESCHEDULE_TIMER,- ; Delay and try again
                                  084B   279                          ASTADR  = CSP$$RESUME,-
                                  084B   280                          REQIDT  = CSP$GL_CURCTX
 00000000'EF     00   FB  0864   281          CALLS   #0,CSP$$WAIT                    ; Wait for completion
               FEDD   31  086B   282          BRW     2$
                                  086E   283
                                  086E   284 15$:     $CMKRNL_S       REQUEST_COMPLETE        ; Complete the request
             F77F CF   94  087D   285 16$:     CLRB    THREAD_ACTIVE                   ; Clear thread active flag
                                  0881   286                                          ;
             00FC 8F   BA  0881   287          POPR    #^M<R2,R3,R4,R5,R6,R7>          ; Restore registers
```

```
                50   01    D0  0885    288 17$:    MOVL    #SS$_NORMAL,R0                  ; Return success
                           05  0888    289         RSB
                               0889    290
```

```
                                0889   292 .SBTTL  REQUEST_INIT - Request initialization
                                0889   293
                                0889   294 ;++
                                0889   295 ;
                                0889   296 ;       This routine acknowledges the request, gets the quorum disk LBN
                                0889   297 ;       and name, and initializes the reschedule timer.
                                0889   298 ;
                                0889   299 ; CALLING SEQUENCE:
                                0889   300 ;
                                0889   301 ;       $CMKRNL_S       REQUEST_INIT
                                0889   302 ;
                                0889   303 ; INPUTS:
                                0889   304 ;
                                0889   305 ;       NONE
                                0889   306 ;
                                0889   307 ; OUTPUT:
                                0889   308 ;
                                0889   309 ;       R0 = Status
                                0889   310 ;                SS$_NORMAL - Request is necessary (Quorum disk state is NOT_READY)
                                0889   311 ;                         0 - Request not necessary
                                0889   312 ;       R1,R3 Destroyed
                                0889   313 ;--
                                0889   314
                                0889   315 REQUEST_INIT:
                           0000 0889   316         .WORD   0
                             50  D4    088B   317         CLRL    R0                              ; Assume request is not necessary
                 53  00000000'GF  D0    088D   318         MOVL    G^CLU$GL_CLUB,R3                ; Get CLUB address
                     53  00B4 C3  D0    0894   319         MOVL    CLUB$L_CLUDCB(R3),R3            ; Get CLUDCB address
                             01  B1    0899   320         CMPW    #CLUDCB$M_QS_NOT_READY,-        ; Is state NOT_READY?
                           20 A3         089B   321                 CLUDCB$W_STATE(R3)
                             1F  12    089D   322         BNEQU   1$                              ; Br if not
                             10  A8    089F   323         BISW    #CLUDCB$M_QF_CSPACK,-           ; Ack the request
                           22 A3         08A1   324                 CLUDCB$W_FLAGS(R3)
                    F7C4 CF  1C A3  D0  08A3   325         MOVL    CLUDCB$L_QFLBN(R3),CLUDCB_LBN   ; Get the LBN
                 51  00000000'GF  3C    08A9   326         MOVZWL  G^CLU$GW_QDSKINTERVAL,R1        ; Get quorum disk timeout value
              00 FF676980 8F  51  7A    08B0   327         EMUL    R1,#-10000000,#0,-              ; Convert to seconds (Delta form)
                         F861 CF        08B8   328                 RESCHEDULE_TIMER
                         50  01  D0    08BB   329         MOVL    #SS$_NORMAL,R0                  ; Request is necessary
                                 04    08BE   330 1$:      RET
```

```
                                08BF    332 .SBTTL  GET_QDNAME - Get the quorum disk name
                                08BF    333
                                08BF    334 ;++
                                08BF    335 ;       This routine gets the full quorum disk name and makes sure that a
                                08BF    336 ;       copy is in the CLUB.
                                08BF    337 ;
                                08BF    338 ; CALLING SEQUENCE:
                                08BF    339 ;
                                08BF    340 ;       JSB     GET_QDNAME
                                08BF    341 ;
                                08BF    342 ; INPUTS:
                                08BF    343 ;
                                08BF    344 ;       NONE
                                08BF    345 ;
                                08BF    346 ; OUTPUT:
                                08BF    347 ;
                                08BF    348 ;       R0 = Status
                                08BF    349 ;       R1-R7 Destroyed
                                08BF    350 ;--
                                08BF    351
                                08BF    352 GET_QDNAME:
                          10    28    08BF    353         MOVC3   #CLUDCB$S_DISK_QUORUM,-         ; Get copy of quorum disk name
F740 CF    00000000'GF                 08C1    354                 G^CLU$GB_QDISK,QD_NAME
                       10   20    3A   08C9    355         LOCC    #^A/ /,#CLUDCB$S_DISK_QUORUM,-  ; Locate end of name
                     F73A CF          08CC    356                 QD_NAME
                F72C CF    10   50    A3   08CF    357         SUBW3   R0,#CLUDCB$S_DISK_QUORUM,QD_DESCR ; Store name length
                                    08D5    358         $GETDVIW_S      DEVNAM = QD_DESCR,-     ; Get full device name
                                    08D5    359                         ITMLST = QD_ITMLST,-
                                    08D5    360         -;              ASTADR = CSP$$RESUME,-
                                    08D5    361         -;              ASTPRM = CSP$GL_CURCTX,-
                                    08D5    362                         IOSB = IOSB
                          1C 50     E9    08F1    363         BLBC    R0,1$                           ; Br if error
                                    08F4    364 -;      CALLS   #0,CSP$$WAIT                    ; Wait for completion
                    50    F781 CF   3C    08F4    365         MOVZWL  IOSB,R0                         ; Get completion status
                          14 50     E9    08F9    366         BLBC    R0,1$                           ; Br if error
                                    08FC    367         $CMKRNL_S       GET_QDNAME1             ; Put name in CLUB
                          50   01   D0    090B    368         MOVL    #1,R0                           ; Return success
                               04   11    090E    369         BRB     2$
                          F6ED CF   B4    0910    370 1$:     CLRW    QD_DESCR                        ; Zero name size
                               05   05    0914    371 2$:     RSB
                                    0915    372
                                    0915    373 GET_QDNAME1:
                               0000 .WORD   0915    374         .WORD   0
                    56    00000000'GF   D0    0917    375         MOVL    G^CLU$GL_CLUB,R6                ; Get CLUB address
                               00B8 C6   95    091E    376         TSTB    CLUB$T_QDNAME(R6)               ; Is name in CLUB already?
                                    19   12    0922    377         BNEQU   1$                      ; Br if yes
                    57    F6D8 CF   02    A3    0924    378         SUBW3   #2,QD_DESCR,R7            ; Get adjusted name size
00B9 C6             F6DB CF   57    28    092A    379         MOVC3   R7,QD_NAME+1,CLUB$T_QDNAME+1(R6) ; Put name in CLUB
                               00B8 C6   57    90    0932    380         MOVB    R7,CLUB$T_QDNAME(R6)    ; Put name size in CLUB
                                00000000'GF   16    0937    381         JSB     G^CNX$DISK_CHANGE       ; Tell connection manager
                                    04   0938    382 1$:     RET
```

```
                                093E    384 .SBTTL  OPEN_FILE - Open the quorum file
                                093E    385
                                093E    386 ;++
                                093E    387 ;       This routine "opens" the quorum file and obtains its logical block
                                093E    388 ;       number. It will first attempt to find any existing quorum file but
                                093E    389 ;       if unsuccessful, it will create a new one. Note that the logical
                                093E    390 ;       block number is only returned if the file was not created. This is
                                093E    391 ;       due to the fact that the statistics block is not returned on a create.
                                093E    392 ;
                                093E    393 ; CALLING SEQUENCE:
                                093E    394 ;
                                093E    395 ;       JSB     OPEN_FILE
                                093E    396 ;
                                093E    397 ; INPUTS:
                                093E    398 ;
                                093E    399 ;       NONE
                                093E    400 ;
                                093E    401 ; OUTPUT:
                                093E    402 ;
                                093E    403 ;       R0 = Status of file open
                                093E    404 ;
                                093E    405 ;       R1-R6 Destroyed
                                093E    406 ;--
                                093E    407
                                093E    408 OPEN_FILE:
                                093E    409
                                093E    410 ;
                                093E    411 ; First we initialize the FIB.
                                093E    412 ;
                     56 F74C CF 093E    413         MOVAL   FIB,R6                          ; Get FIB pointer
66 0040 8F 00 6E 00 2C 0943    414         MOVC5   #0,(SP),#0,#FIB$K_LENGTH,(R6)   ; Init FIB to all zeros
                00080501 8F 00 094B    415         MOVL    #FIB$M_WRITE!FIB$M_NOREAD!FIB$M_NOWRITE!FIB$M_WRITETHRU,-
                      66    0951    416                 FIB$L_ACCTL(R6)                 ; Access bits
                   0A A6 04 B0 0952    417         MOVW    #FID$C_MFD,FIB$W_DID_NUM(R6)    ; Directory is the MFD
                   0C A6 04 B0 0956    418         MOVW    #FID$C_MFD,FIB$W_DID_SEQ(R6)
                      0085 8F B0 095A    419         MOVW    #FIB$M_EXTEND!FIB$M_ALCON!FIB$M_FILCON,- ; Extend control bits
                        16 A6       095E    420                 FIB$W_EXCTL(R6)
                   18 A6 02 D0 0960    421         MOVL    #CLUQF$K_BLOCKS,FIB$L_EXSZ(R6)  ; Allocation size
                                0964    422 ;
                                0964    423 ; Attempt to lookup/create the quorum file and access for read write
                                0964    424 ;
                                0964    425         $QIO_S          CHAN    = CHANNEL,-
                                0964    426                         FUNC    = #IO$_ACCESS!IO$M_ACCESS!IO$M_CREATE,-
                                0964    427                         IOSB    = IOSB,-
                                0964    428                         ASTADR  = CSP$$RESUME,-
                                0964    429                         ASTPRM  = CSP$GL_CURCTX,-
                                0964    430                         P1      = FIB_DESCR,-
                                0964    431                         P2      = #QF_DESCR,-
                                0964    432                         P5      = #ATTRIB_BLOCK
                      1D 50 E9 099B    433         BLBC    R0,1$                           ; Br if error
          00000000'EF 00 FB 099E    434         CALLS   #0,CSP$$WAIT                    ; Wait for completion
                50 F6D0 CF 3C 09A5    435         MOVZWL  IOSB,R0                         ; Get I/O completion status
                      0E 50 E9 09AA    436         BLBC    R0,1$                           ; Br if error
                                09AD    437 ;
                                09AD    438 ; Get the quorum file LBN.
                                09AD    439 ;
          F6BD CF F73B CF B0 09AD    440         MOVW    STATBLK+SBK$W_STLBNL,LOOKUP_LBN ; Get the Low-order LBN
```

```
                F6B8 CF    F732 CF    B0  09B4    441          MOVW    STATBLK+SBK$W_STLBNH,LOOKUP_LBN+2 ; Get the High-order LBN
                                      05  09BB    442 1$:      RSB
```

```
                     09BC    444 .SBTTL  GET_LBN - Get the quorum file logical block number
                     09BC    445
                     09BC    446 ;++
                     09BC    447 ;       This routine reads the quorum file attributes and gets the quorum
                     09BC    448 ;       file logical block number.
                     09BC    449 ;
                     09BC    450 ; CALLING SEQUENCE:
                     09BC    451 ;
                     09BC    452 ;       JSB     GET_LBN
                     09BC    453 ;
                     09BC    454 ; INPUTS:
                     09BC    455 ;
                     09BC    456 ;       NONE
                     09BC    457 ;
                     09BC    458 ; OUTPUT:
                     09BC    459 ;
                     09BC    460 ;       R0 = Status of file open
                     09BC    461 ;
                     09BC    462 ;       R1 Destroyed
                     09BC    463 ;--
                     09BC    464
                     09BC    465 GET_LBN:
                     09BC    466
                     09BC    467         $QIO_S          CHAN    = CHANNEL,-     ; Read the file attributes
                     09BC    468                         FUNC    = #IO$_ACCESS,-
                     09BC    469                         IOSB    = IOSB,-
                     09BC    470                         ASTADR  = CSP$$RESUME,-
                     09BC    471                         ASTPRM  = CSP$GL_CURCTX,-
                     09BC    472                         P1      = FIB_DESCR,-
                     09BC    473                         P2      = #QF_DESCR,-
                     09BC    474                         P5      = #ATTRIB_BLOCK
           1D 50     E9    09F1    475         BLBC    R0,1$                           ; Br if error
00000000'EF    00     FB    09F4    476         CALLS   #0,CSP$$WAIT                    ; Wait for completion
     50   F67A CF     3C    09FB    477         MOVZWL  IOSB,R0                         ; Get I/O completion status
           0E 50     E9    0A00    478         BLBC    R0,1$                           ; Br if error
                           0A03    479 ;
                           0A03    480 ; Get the quorum file LBN.
                           0A03    481 ;
F667 CF    F6E5 CF    B0    0A03    482         MOVW    STATBLK+SBK$W_STLBNL,LOOKUP_LBN ; Get the Low-order LBN
F662 CF    F6DC CF    B0    0A0A    483         MOVW    STATBLK+SBK$W_STLBNH,LOOKUP_LBN+2 ; Get the High-order LBN
                     05    0A11    484 1$:     RSB
```

```
                     0A12   486 .SBTTL  VALIDATE_FILE - Validate the quorum file
                     0A12   487
                     0A12   488 ;++
                     0A12   489 ;       This routine reads the quorum file and validates its contents.
                     0A12   490 ;
                     0A12   491 ; CALLING SEQUENCE:
                     0A12   492 ;
                     0A12   493 ;       JSB     VALIDATE_FILE
                     0A12   494 ;
                     0A12   495 ; INPUTS:
                     0A12   496 ;
                     0A12   497 ;       NONE
                     0A12   498 ;
                     0A12   499 ; OUTPUT:
                     0A12   500 ;
                     0A12   501 ;       R0 = Status of validate
                     0A12   502 ;
                     0A12   503 ;               If R0 = SS$_NORMAL the file is valid. If R0 = 0 the file is
                     0A12   504 ;               invalid. Otherwise R0 contains an I/O status error.
                     0A12   505 ;
                     0A12   506 ;       R1-R3,R6,R7 Destroyed
                     0A12   507 ;--
                     0A12   508
                     0A12   509 VALIDATE_FILE:
                     0A12   510
                     0A12   511 ;
                     0A12   512 ; Queue a read request to the quorum file
                     0A12   513 ;
                     0A12   514         $QIO_S          CHAN    = CHANNEL,-
                     0A12   515                         FUNC    = #IO$_READLBLK,-
                     0A12   516                         IOSB    = IOSB,-
                     0A12   517                         ASTADR  = CSP$$RESUME,-
                     0A12   518                         ASTPRM  = CSP$GL_CURCTX,-
                     0A12   519                         P1      = QF_BUFFER,-
                     0A12   520                         P2      = #CLUQF$K_BLOCKS*512,-
                     0A12   521                         P3      = LOOKUP_LBN
          34 50   E9 0A45   522         BLBC    R0,2$                           ; Br if error
00000000'EF   00   FB 0A48   523         CALLS   #0,CSP$$WAIT                    ; Wait for completion
   50 F626 CF   3C 0A4F   524         MOVZWL  IOSB,R0                         ; Get I/O completion status
          25 50   E9 0A54   525         BLBC    R0,2$                           ; Br if error
                     0A57   526 ;
                     0A57   527 ; Validate the data in the quorum file
                     0A57   528 ;
          7E   D4 0A57   529         CLRL    -(SP)                           ; Assume file not valid
   56 F6C7 CF   DE 0A59   530         MOVAL   QF_BUFFER,R6                    ; Get buffer pointer
 00000B4C'EF   16 0A5E   531         JSB     CALCULATE_CHECKSUM              ; Get the checksum
          57   D5 0A64   532         TSTL    R7                              ; Is checksum valid?
          11   12 0A66   533         BNEQU   1$                              ; Br if not
          0C   29 0A68   534         CMPC3   #CLUQF$S_IDENT,-                ; Validate ID area
                66 0A6A   535                 CLUQF$T_IDENT(R6),-
        F6A2 CF      0A6B   536                 IDENT_STRING
          09   12 0A6E   537         BNEQU   1$                              ; Br if file invalid
          02   B1 0A70   538         CMPW    #CLUQF$K_VERSION,-              ; Is version correct?
            0C A6      0A72   539                 CLUQF$W_VERSION(R6)
          03   12 0A74   540         BNEQU   1$                              ; Br if not
       6E   01   9A 0A76   541         MOVZBL  #SS$_NORMAL,(SP)                ; File is valid
       50   8E   D0 0A79   542 1$:     MOVL    (SP)+,R0                        ; Return status
```

```
 05  0A7C   543 2$:     RSB
```

```
                        0A7D   545 .SBTTL  WRITE_FILE - Write the quorum file
                        0A7D   546
                        0A7D   547 ;++
                        0A7D   548 ;       This routine builds a template quorum file and writes it to the disk.
                        0A7D   549 ;
                        0A7D   550 ; CALLING SEQUENCE:
                        0A7D   551 ;
                        0A7D   552 ;       JSB     WRITE_FILE
                        0A7D   553 ;
                        0A7D   554 ; INPUTS:
                        0A7D   555 ;
                        0A7D   556 ;       NONE
                        0A7D   557 ;
                        0A7D   558 ; OUTPUT:
                        0A7D   559 ;
                        0A7D   560 ;       R0 = Status of the write
                        0A7D   561 ;
                        0A7D   562 ;       R1-R7 Destroyed
                        0A7D   563 ;--
                        0A7D   564
                        0A7D   565 WRITE_FILE:
                        0A7D   566
                        0A7D   567 ;
                        0A7D   568 ; Build a template quorum file
                        0A7D   569 ;
56    F6A3 CF    DE     0A7D   570         MOVAL   QF_BUFFER,R6                    ; Get buffer pointer
00    6E   00    2C     0A82   571         MOVC5   #0,(SP),#0,-                    ; Zero buffer
66    0400 8F           0A86   572                 #CLUQF$K_BLOCKS*512,(R6)
           0C    28     0A8A   573         MOVC3   #CLUQF$S_IDENT,-                ; Store ident string
      F681 CF           0A8C   574                 IDENT_STRING,-
           66           0A8F   575                 CLUQF$T_IDENT(R6)
        63   02    B0   0A90   576         MOVW    #CLUQF$K_VERSION,(R3)           ; Store version number
00000B4C'EF    16     0A93   577         JSB     CALCULATE_CHECKSUM              ; Get the checksum
 44   A6    57    D0   0A99   578         MOVL    R7,CLUQF$L_CHECKSUM(R6)         ; Store checksum
 48   A6    01    90   0A9D   579         MOVB    #1,CLUQF$B_IGNORE(R6)           ; Set ignore flag
                        0AA1   580 ;
                        0AA1   581 ; Write the template quorum file.
                        0AA1   582 ;
                        0AA1   583         $QIO_S          CHAN    = CHANNEL,-
                        0AA1   584                         FUNC    = #IO$_WRITELBLK,-
                        0AA1   585                         IOSB    = IOSB,-
                        0AA1   586                         ASTADR  = CSP$$RESUME,-
                        0AA1   587                         ASTPRM  = CSP$GL_CURCTX,-
                        0AA1   588                         P1      = QF_BUFFER,-
                        0AA1   589                         P2      = #CLUQF$K_BLOCKS*512,-
                        0AA1   590                         P3      = LOOKUP_LBN
        0C   50    E9   0AD4   591         BLBC    R0,1$                           ; Br if error
00000000'EF    00     FB   0AD7   592         CALLS   #0,CSP$$WAIT                    ; Wait for completion
    50   F597 CF    3C   0ADE   593         MOVZWL  IOSB,R0                         ; Get I/O completion status
                        05   0AE3   594 1$:     RSB
```

```
                     0AE4    596 .SBTTL  CLOSE_FILE - Close the quorum file
                     0AE4    597
                     0AE4    598 ;++
                     0AE4    599 ;       This routine "closes" the quorum file by issuing a QIO with the
                     0AE4    600 ;       IO$_DEACCESS function code.
                     0AE4    601 ;
                     0AE4    602 ; CALLING SEQUENCE:
                     0AE4    603 ;
                     0AE4    604 ;       JSB     CLOSE_FILE
                     0AE4    605 ;
                     0AE4    606 ; INPUTS:
                     0AE4    607 ;
                     0AE4    608 ;       NONE
                     0AE4    609 ;
                     0AE4    610 ; OUTPUT:
                     0AE4    611 ;
                     0AE4    612 ;       R0,R1 Destroyed
                     0AE4    613 ;--
                     0AE4    614
                     0AE4    615 CLOSE_FILE:
                     0AE4    616
                     0AE4    617         $QIO_S          CHAN    = CHANNEL,-     ; Queue deaccess request
                     0AE4    618                         FUNC    = #IO$_DEACCESS,-
                     0AE4    619                         IOSB    = IOSB,-
                     0AE4    620                         ASTADR  = CSP$$RESUME,-
                     0AE4    621                         ASTPRM  = CSP$GL_CURCTX
           07 50     E9    0B0D    622         BLBC    R0,1$                           ; Br if error
00000000'EF  00     FB    0B10    623         CALLS   #0,CSP$$WAIT                    ; Wait for completion
                     05    0B17    624 1$:     RSB
```

```
                                0B18   626 .SBTTL  REQUEST_COMPLETE - Request completion
                                0B18   627
                                0B18   628 ;++
                                0B18   629 ;       This routine completes the request by updating the CLUDCB fields.
                                0B18   630 ;
                                0B18   631 ; CALLING SEQUENCE:
                                0B18   632 ;
                                0B18   633 ;       $CMKRNL_S       REQUEST_COMPLETE
                                0B18   634 ;
                                0B18   635 ; INPUTS:
                                0B18   636 ;
                                0B18   637 ;       NONE
                                0B18   638 ;
                                0B18   639 ; OUTPUT:
                                0B18   640 ;
                                0B18   641 ;       R0,R1 Destroyed
                                0B18   642 ;--
                                0B18   643
                                0B18   644 REQUEST_COMPLETE:
                                0B18   645
                           0000 0B18   646         .WORD   0
      50   00000000'GF      D0 0B1A   647         MOVL    G^CLU$GL_CLUB,R0                ; Get CLUB address
          50    00B4 C0      D0 0B21   648         MOVL    CLUB$L_CLUDCB(R0),R0            ; Get CLUDCB address
          51    F54B CF      D0 0B26   649         MOVL    CHANNEL,R1                      ; Get channel number
51    00000000'9F    51      C3 0B2B   650         SUBL3   R1,@#CTL$GL_CCBBASE,R1          ; Form CCB address
                               0B33   651         SETIPL  #IPL$_TIMER                     ; Synchronize access to CLUDCB
           0C A0    61      D0 0B36   652         MOVL    CCB$L_UCB(R1),CLUDCB$L_UCB(R0)  ; Store UCB add ess in CLUDCB
        1C A0    F533 CF      D0 0B3A   653         MOVL    LOOKUP_LBN,CLUDCB$L_QFLBN(R0)   ; Put LBN in CLUDCB
                    10      AA 0B40   654         BICW    #CLUDCB$M_QF_CSPACK,-           ; Clear the in progress bit
                 22 A0              0B42   655                 CLUDCB$W_FLAGS(R0)
                    02      B0 0B44   656         MOVW    #CLUDCB$M_QS_READY,-            ; Set state to ready
                 20 A0              0B46   657                 CLUDCB$W_STATE(R0)
                               0B48   658         SETIPL  #0                              ; Restore IPL
                            04 0B4B   659         RET
```

```
                     0B4C    661 .SBTTL  CALCULATE_CHECKSUM - Calculate the quorum file checksum
                     0B4C    662
                     0B4C    663 ;++
                     0B4C    664 ;       This routine calulates the checksum of the quorum block pointed to
                     0B4C    665 ;       by R6. It includes the field CLUQF$L_CHECKSUM in the checksum
                     0B4C    666 ;       calculation.
                     0B4C    667 ;
                     0B4C    668 ; CALLING SEQUENCE:
                     0B4C    669 ;
                     0B4C    670 ;       JSB     CALCULATE_CHECKSUM
                     0B4C    671 ;
                     0B4C    672 ; INPUTS:
                     0B4C    673 ;
                     0B4C    674 ;       R6 = Pointer to the quorum block
                     0B4C    675 ;
                     0B4C    676 ; OUTPUT:
                     0B4C    677 ;
                     0B4C    678 ;       R7 = Quorum block checksum
                     0B4C    679 ;
                     0B4C    680 ;       R2,R3   Destroyed
                     0B4C    681 ;--
                     0B4C    682
                     0B4C    683 CALCULATE_CHECKSUM:
                     0B4C    684
52    12    D0       0B4C    685         MOVL    #CLUQF$K_CHECK_LENGTH/4,R2      ; R2 = checksum longword count
53    56    D0       0B4F    686         MOVL    R6,R3                           ; Copy buffer address
      57    D4       0B52    687         CLRL    R7                              ; Form checksum in R7
57    83    CC       0B54    688 1$:     XORL2   (R3)+,R7                        ; Accumulate checksum
  FA  52    F5       0B57    689         SOBGTR  R2,1$                           ; Br if more
            05       0B5A    690         RSB
                     0B5B    691
                     0B5B    692         .END
```

```
$$T1                            = 00000000
ATR$C_FPRO                      = 00000016
ATR$C_STATBLK                   = 00000009
ATR$C_UIC                       = 00000015
ATR$S_FPRO                      = 00000002
ATR$S_STATBLK                   = 00000020
ATR$S_UIC                       = 00000004
ATTRIB_BLOCK                      000000CE R      01
CALCULATE_CHECKSUM                0000UB4C R      01
CCB$L_UCB                       = 00000000
CHANNEL                           00000075 R      01
CLOSE_FILE                        00000AE4 R      01
CLU$GB_QDISK                      ******** X    01
CLU$GL_CLUB                       ******** X    01
CLU$GW_QDSKINTERVAL               ******** X    01
CLUB$L_CLUDCB                   = 000000B4
CLUB$T_QDNAME                   = 000000B8
CLUDCB$L_QFLBN                  = 0000001C
CLUDCB$L_UCB                    = 0000000C
CLUDCB$M_QF_CSPACK              = 00000010
CLUDCB$M_QS_NOT_READY           = 00000001
CLUDCB$M_QS_READY               = 00000002
CLUDCB$S_DISK_QUORUM            = 00000010
CLUDCB$W_FLAGS                  = 00000022
CLUDCB$W_STATE                  = 00000020
CLUDCB_LBN                        0000006D R      01
CLUQF$B_IGNORE                  = 00000048
CLUQF$K_BLOCKS                  = 00000002
CLUQF$K_CHECK_LENGTH            = 00000048
CLUQF$K_VERSION                 = 00000002
CLUQF$L_CHECKSUM                = 00000044
CLUQF$S_IDENT                   = 0000000C
CLUQF$T_IDENT                   = 00000000
CLUQF$W_VERSION                 = 0000000C
CNX$DISR_CHANGE                   ******** X    01
CSP$$RESUME                       ******** X    01
CSP$$WAIT                         ******** X    01
CSP$GL_CURCTX                     ******** X    01
CSP$QUORUM                        00000729 RG     01
CSP$TELL_OPCOM                    ******** X    01
CTL$GL_CCBBASE                    ******** X    01
DSC$K_CLASS_D                   = 00000002
DSC$K_CLASS_S                   = 00000001
DSC$K_DTYPE_T                   = 0000000E
DVI$_FULLDEVNAM                 = 000000E8
ERROR_COUNT                       00000081 R      01
ERROR_MESSAGE                     00000082 R      01
ERROR_THRESHOLD                 = 0000000A
FIB                               0000008E R      01
FIB$K_LENGTH                    = 00000040
FIB$L_ACCTL                     = 00000000
FIB$L_EXSZ                      = 00000018
FIB$M_ALCON                     = 00000001
FIB$M_EXTEND                    = 00000080
FIB$M_FILCON                    = 00000004
FIB$M_NOREAD                    = 00000400
FIB$M_NOWRITE                   = 00000001
FIB$M_WRITE                     = 00000100
FIB$M_WRITETHRU                 = 00080000
FIB$W_DID_NUM                   = 0000000A
FIB$W_DID_SEQ                   = 0000000C
FIB$W_EXCTL                     = 00000016
FIB_DESCR                         00000086 R      01
FID$C_MFD                       = 00000004
FPRO                              0000010E R      01
GET_LBN                           000009BC R      01
GET_QDNAME                        000008BF R      01
GET_QDNAME1                       00000915 R      01
IDENT_STRING                      00000110 R      01
IO$M_ACCESS                     = 00000040
IO$M_CREATE                     = 00000080
IO$_ACCESS                      = 00000032
IO$_DEACCESS                    = 00000034
IO$_READLBLK                    = 00000021
IO$_WRITELBLK                   = 00000020
IOSB                              00000079 R      01
IPL$_TIMER                      = 00000008
LOOKUP_LBN                        00000071 R      01
MSSG1                             00000524 R      01
MSSG10                            000006F6 R      01
MSSG2                             00000568 R      01
MSSG3                             00000595 R      01
MSSG4                             000005CF R      01
MSSG5                             000005FD R      01
MSSG6                             0000062E R      01
MSSG7                             00000662 R      01
MSSG8                             0000069A R      01
MSSG9                             000006C7 R      01
OPEN_FILE                         0000093E R      01
PR$_IPL                           ********   X    01
QD_DESCR                          00000001 R      01
QD_ITMLST                         00000049 R      01
QD_NAME                           00000009 R      01
QF_BUFFER                         00000124 R      01
QF_DESCR                          00000059 R      01
REQUEST_COMPLETE                  00000B18 R      01
REQUEST_INIT                      00000889 R      01
RESCHEDULE_TIMER                  0000011C R      01
SBK$W_STLBNH                    = 00000000
SBK$W_STLBNL                    = 00000002
SS$_CREATED                     = 00000619
SS$_NORMAL                      = 00000001
STATBLK                           000000EA R      01
SYS$ASSIGN                        ********  GX    01
SYS$CMKRNL                        ********  GX    01
SYS$DASSGN                        ********  GX    01
SYS$GETDVIW                       ********  GX    01
SYS$QIO                           ********  GX    01
SYS$SETIMR                        ********  GX    01
THREAD_ACTIVE                     00000000 R      01
UIC                               0000010A R      01
VALIDATE_FILE                     00000A12 R      01
WRITE_FILE                        00000A7D R      01
```

**F

Psect synopsis

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
. BLANK .	00000B5B (2907.)	01 (1.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE
ABS	00000000 (0.)	02 (2.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.03	00:00:02.02
Command processing	119	00:00:00.44	00:00:02.91
Pass 1	460	00:00:11.23	00:00:48.42
Symbol table sort	0	00:00:01.81	00:00:04.72
Pass 2	136	00:00:02.33	00:00:08.22
Symbol table output	13	00:00:00.06	00:00:00.32
Psect synopsis output	2	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	761	00:00:15.92	00:01:06.63

The working set limit was 1650 pages.
95851 bytes (188 pages) of virtual memory were used to buffer the intermediate code.
There were 100 pages of symbol table space allocated to hold 1706 non-local and 28 local symbols.
692 source lines were read in Pass 1, producing 17 object records in Pass 2.
32 pages of virtual memory were used to define 31 macros.

Macro library statistics

Macro library name	Macros defined
_255DUA28:[SYSLOA.OBJ]CLUSTER.MLB;1	1
_255DUA28:[SYS.OBJ]LIB.MLB;1	5
_255DUA28:[SYSLIB]STARLET.MLB;2	22
TOTALS (all libraries)	28

1869 GETS were required to define 28 macros.

There were no errors, warnings or information messages.

```
MACRO/LIS=LIS$:CSPQUORUM/OBJ=OBJ$:CSPQUORUM MSRC$:CSPQUORUM/UPDATE=(ENH$:CSPQUORUM)+EXECML$/LIB+LIB$:CLUSTER/LIB
```

CSPOPCOM.LIS

CSPWAIT.LIS

CSPRCPCAC.LIS

CSPCJFRES.LIS

CSPQUORUM.LIS

DISTRLKI.LIS

CSPMOUNT.LIS

CSPVECTOR.LIS

CSPCLIENT.LIS

DSTRLOCK.LIS

DSTRLOCK.LIS