

(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

:DYC0001
-1

```

0000 1 .TITLE CSPQUORUM - CSP DISK QUORUM MODULE
0000 .1 .IDENT 'V04-001'
0000 3
0000 4
0000 5
0000 6
0000 7 *
0000 8 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY *
0000 9 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. *
0000 10 * ALL RIGHTS RESERVED. *
0000 11 *
0000 12 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
0000 13 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE *
0000 14 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER *
0000 15 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
0000 16 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY *
0000 17 * TRANSFERRED. *
0000 18 *
0000 19 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE *
0000 20 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT *
0000 21 * CORPORATION. *
0000 22 *
0000 23 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS *
0000 24 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL. *
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 contains corrupt data.
0000 38
0000 39 Environment:
0000 40 CSP process
0000 41 --
0000 42
0000 43 Author:
0000 44 R. Scott Hanna, CREATION DATE: 23-Aug-1984
0000 45
0000 46 Modified by:
0000 47
0000 .1 V04-001 DYC0001 Dennis Y. Chan 21-Nov-1984
0000 .2 Added check in action routine for SSS_FORCEDERROR return
0000 .3 while trying to validate quorum file and rewrite the
0000 .4 file with template.
0000 .5
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 R0 failure indication is
0000 51 thought to be unpredictable.
0000 52

```

:DYC0001
:DYC0001
:DYC0001
:DYC0001
:DYC0001

```
0000 53 : V03-005 DWT0211 David W. Thiel 09-Apr-1984
0000 54 : Call CNX$DISK_CHANGE when CLUBST_QDNAME is filled in.
0000 55 :
0000 56 : V03-004 RSH0133 R. Scott Hanna 03-Apr-1984
0000 57 : Modify CLOSE_FILE to only call CSP$SWAIT when the file close
0000 58 : I/O is successfully queued.
0000 59 :
0000 60 : V03-003 RSH0126 R. Scott Hanna 21-Feb-1984
0000 61 : Incorporate changes to make this algorithm a "client" of
0000 62 : the CSP. It is now requested by the QUORUM code rather than
0000 63 : periodically running to check for work. In addition this
0000 64 : routine can now create a quorum file if none exists and
0000 65 : repair one that does but contains corrupt data.
0000 66 :
0000 67 : V03-002 ADE0001 Alan D. Eldridge 28-Feb-1984
0000 68 : Change name of CSP$QUORUM to CSP$QUORUM_INIT, add new
0000 69 : CSP$QUORUM entry point which is used now only as a place
0000 70 : holder.
0000 71 :
0000 72 : V03-001 RSH0079 R. Scott Hanna 10-Nov-1983
0000 73 : Modify algorithm to check every 2 minutes to see if the
0000 74 : "connection" to the quorum disk has been lost. If so it
0000 75 : does the quorum file lookup again.
0000 76 :--
```

```
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.
0000000A 0000 100
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,DVIS_FULLDEVNAM ; $GETDVI item list
00000009' 004D 116 .LONG QD_NAME
00000001' 0051 117 .LONG QD_DESCR
00000000 0055 118 .LONG 0
0059 119
00000061'010E0000' 0059 120 QF_DESCR: .ASCID /QUORUM.DAT;1/ ; Quorum file descriptor
4D 55 52 4F 55 51 0061
31 3B 54 41 44 2E 0067
006D 121
00000000 006D 122 CLUDCB_LBN: .LONG 0 ; Quorum file LBN from the CLUDCB
0071 123
00000000 0071 124 LOOKUP_LBN: .LONG 0 ; Quorum file LBN from file lookup
0075 125
00000000 0075 126 CHANNEL: .LONG 0 ; Quorum disk channel number
0079 127
00000000 00000000 0079 128 IOSB: .QUAD 0 ; I/O status block
0081 129
00 0081 130 ERROR_COUNT: .BYTE 0 ; Error reported counter
0082 131
00000000 0082 132 ERROR_MESSAGE: .LONG 0 ; Descr addr of last error message
0086 133
0040 0086 134 FIB_DESCR: .WORD FIB$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 01CE 153 FPRO: .WORD ^XFF00 ; File protection (S:RWED,O:RWED)
0110 154
4D 55 52 4F 55 51 0110 155 IDENT_STRING: .ASCII /QUORUM FILE/ ; Quorum file ID string
45 4C 49 46 20 20 0116
011C 156
011C 157 ASSUME CLUQF$S_IDENT EQ .-IDENT_STRING

```

```
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
0000052C'010E0000' 0524 162 MSSG1: .ASCID \XCSP-W-QFNOTFOUND, Previously existing quorum file not found\
57 2D 50 53 43 25 052C
54 4F 4E 46 51 2D 0532
2C 44 4E 55 4F 46 0538
69 76 65 72 50 20 053E
20 79 6C 73 75 6F 0544
69 74 73 69 78 65 054A
6F 75 71 20 67 6E 0550
69 66 20 6D 75 72 0556
74 6F 6E 20 65 6C 055C
64 6E 75 6F 66 20 0562
00000570'010E0000' 0568 163 MSSG2: .ASCID \XCSP-I-QFCREATED, Quorum file created\
49 2D 50 53 43 25 0570
45 52 43 46 51 2D 0576
20 2C 44 45 54 41 057C
6D 75 72 6F 75 51 0582
20 65 6C 69 66 20 0588
65 74 61 65 72 63 058E
                                64 0594
0000059D'010E0000' 0595 164 MSSG3: .ASCID \XCSP-W-QFCHANGED, Quorum file location has changed\
57 2D 50 53 43 25 059D
41 48 43 46 51 2D 05A3
20 2C 44 45 47 4E 05A9
6D 75 72 6F 75 51 05AF
20 65 6C 69 66 20 05B5
69 74 61 63 6F 6C 05BB
73 61 68 20 6E 6F 05C1
67 6E 61 68 63 20 05C7
                                64 65 05CD
000005D7'010E0000' 05CF 165 MSSG4: .ASCID \XCSP-I-QFINIT, Quorum file initialized\
49 2D 50 53 43 25 05D7
49 4E 49 46 51 2D 05DD
6F 75 51 20 2C 54 05E3
69 66 20 6D 75 72 05E9
69 6E 69 20 65 6C 05EF
7A 69 6C 61 69 74 05F5
                                64 65 05FB
00000605'010E0000' 05FD 166 MSSG5: .ASCID \XCSP-E-QDASSIGN, Quorum disk assign error\
45 2D 50 53 43 25 0605
53 53 41 44 51 2D 060B
51 20 2C 4E 47 49 0611
20 6D 75 72 6F 75 0617
61 20 6B 73 69 64 061D
20 6E 67 69 73 73 0623
                                72 6F 72 72 65 0629
00000636'010E0000' 062E 167 MSSG6: .ASCID \XCSP-E-QFOPEN, Quorum file open/create error\
45 2D 50 53 43 25 0636
45 50 4F 46 51 2D 063C
6F 75 51 20 2C 4E 0642
69 66 20 6D 75 72 0648
65 70 6F 20 65 6C 064E
61 65 72 63 2F 6E 0654
72 72 65 20 65 74 065A
```



```
0000066A'010E0000' 72 6F 0660
45 2D 50 53 43 25 0662 168 MSSG7: .ASCID \XCSP-E-QFRATT, Quorum file read attributes error\
54 41 52 46 51 2D 066A
6F 75 51 20 2C 54 0670
69 66 20 6D 75 72 0676
61 65 72 20 65 72 067C
72 74 74 61 20 64 0682
73 65 74 75 62 69 0688
72 6F 72 72 65 20 068E
000006A2'010E0000' 0694
45 2D 50 53 43 25 069A 169 MSSG8: .ASCID \XCSP-E-QFREAD, Quorum file read error\
41 45 52 46 51 2D 06A2
6F 75 51 20 2C 44 06AB
69 66 20 6D 75 72 06AE
61 65 72 20 65 6C 06B4
6F 72 72 65 20 64 06BA
72 06C0
000006CF'010E0000' 72 06C6
45 2D 50 53 43 25 06C7 170 MSSG9: .ASCID \XCSP-E-QFWRITE, Quorum file write error\
49 52 57 46 51 2D 06CF
75 51 20 2C 45 54 06D5
66 20 6D 75 72 6F 06DB
72 77 20 65 6C 69 06E1
72 65 20 65 74 69 06E7
72 6F 72 06ED
000006FE'010E0000' 06F3
45 2D 50 53 43 25 06F6 171 MSSG10: .ASCID \XCSP-E-QDGETDVI, Quorum disk $GETDVI failed\
54 45 47 44 51 2D 06FE
51 20 2C 49 56 44 0704
20 6D 75 72 6F 75 070A
24 20 68 73 69 64 0710
49 56 44 54 45 47 0716
65 6C 69 61 66 20 0722
64 0728
0729 172
```

```

0729 174 .SBTTL CSP$QUORUM - Disk quorum action routine
0729 175
0729 176 :++
0729 177 : This routine is requested when some type of error has occurred with the
0729 178 : quorum disk. It acknowledges the request and gets the quorum disk name
0729 179 : and logical block number from the CLUDCB. It then does a access with a
0729 180 : create modifier QIO to the ACP. (i.e. If the quorum file does not exist
0729 181 : it creates one.) If a quorum file exists the data in the file is
0729 182 : validated. If the quorum file contents are invalid or the quorum file
0729 183 : was created, a template quorum block is written back to the file. The
0729 184 : request is completed by updating the logical block number, the request
0729 185 : and acknowledge bits, and the state field in the CLUDCB.
0729 186
0729 187 CALLING SEQUENCE:
0729 188
0729 189 JSB CSP$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$QUORUM::
0729 200 BBS #0,THREAD_ACTIVE,1$ ; Br if thread not active
0729 201
0729 202 1$: BRW 17$ ; Thread active, ignore request
0729 203 PUSHR #^M<R2,R3,R4,R5,R6,R7> ; Save registers
0729 204
0729 205 $CMKRNLS REQUEST_INIT ; Do the initial work
0729 206 BLBS R0,2$ ; Br if request necessary
0729 207 BRW 16$
0729 208 2$: TSTW QD_DESCR ; Do we have the quorum disk name?
0729 209 BNEQU 3$ ; Br if yes
0729 210 JSB GET_QDNAME ; Get quorum disk name
0729 211 BLBS R0,3$ ; Br if success
0729 212 MOVAL MSG10,R1 ; Quorum disk assign error
0729 213 BRW 11$
0729 214
0729 215 3$: $ASSIGN_S DEVNAM = QD_DESCR,- ; Assign a channel to
0729 216 CHAN = CHANNEL ; the quorum disk
0729 217 BLBS R0,4$ ; Br if success
0729 218 MOVAL MSG5,R1 ; Quorum disk assign error
0729 219 BRW 11$
0729 220
0729 221 4$: JSB OPEN_FILE ; Open the quorum file
0729 222 BLBS R0,5$ ; Br if success
0729 223 MOVAL MSG6,R1 ; Quorum disk open/create error
0729 224 BRB 11$
0729 225
0729 226 5$: CMPW R0,#SS$_CREATED ; Did we create the file?
0729 227 BNEQU 8$ ; Br if not
0729 228 JSB GET_LBN ; Get the quorum file LBN
0729 229 BLBS R0,6$ ; Br if error
  
```


		0852	279		ASTADR = CSP\$\$RESUME,-	
		0852	280		REQIDT = CSP\$GL_CURCTX	
00000000	'EF	FB	086B	281	CALLS #0,CSP\$\$WAIT	; Wait for completion
	FED6	31	0872	282	BRW 2\$	
			0875	283		
			0875	284	15\$: \$CMKRNLS REQUEST_COMPLETE	; Complete the request
f778	CF	94	0884	285	16\$: CLRBT THREAD_ACTIVE	; Clear thread active flag
			0888	286		
00FC	8F	BA	0888	287	POPR #^M<R2,R3,R4,R5,R6,R7>	; Restore registers
50	01	D0	088C	288	17\$: MOVL #SS\$NORMAL,R0	; Return success
		05	088F	289	RSB	
			0890	290		

CS
PS

PS
--
.
SA

Ph
--
In
Co
Pa
Syn
Pa
Syn
PS
Cr
As

The
95
The
69
32

Mac
--
-S
-S
-S
TO
18
The
MA

```

0890 292 .SBTTL REQUEST_INIT - Request initialization
0890 293
0890 294 :++
0890 295 :
0890 296 :       This routine acknowledges the request, gets the quorum disk LBN
0890 297 :       and name, and initializes the reschedule timer.
0890 298 :
0890 299 : CALLING SEQUENCE:
0890 300 :
0890 301 :       $CMKRNL_S       REQUEST_INIT
0890 302 :
0890 303 : INPUTS:
0890 304 :
0890 305 :       NONE
0890 306 :
0890 307 : OUTPUT:
0890 308 :
0890 309 :       R0 = Status
0890 310 :           SSS_NORMAL - Request is necessary (Quorum disk state is NOT_READY)
0890 311 :           0 - Request not necessary
0890 312 :       R1,R3 Destroyed
0890 313 :--
0890 314
0890 315 REQUEST_INIT:
0890 316 .WORD 0
00000000 50 D4 0892 317 CLRL R0 ; Assume request is not necessary
53 00B4 C3 D0 0894 318 MOVL G^CLUSGL_CLUB,R3 ; Get CLUB address
53 00B4 C3 D0 0898 319 MOVL CLUB$L CLUDCB(R3),R3 ; Get CLUDCB address
20 01 B1 08A0 320 CMPW #CLUDCB$M QS NOT_READY,- ; Is state NOT_READY?
20 A3 08A2 321 CLUDCB$W_STATE(R3)
1F 12 08A4 322 BNEQU 1$ ; Br if not
22 A3 AB 08A6 323 BISW #CLUDCB$M OF CSPACK,- ; Ack the request
1C A3 D0 08A8 324 CLUDCB$W_FLAGS(R3)
F7BD CF D0 08AA 325 MOVL CLUDCB$L_QFLBN(R3),CLUDCB_LBN ; Get the LBN
00000000 51 3C 08B0 326 MOVZWL G^CLUSGW_QDSKINTERVAL,R1 ; Get quorum disk timeout value
51 7A 08B7 327 EMUL R1,#-10000000,#0,- ; Convert to seconds (Delta form)
FF676980 8F 08B9
00 08BE
F85A CF 08BF 328 RESCHEDULE TIMER
50 01 D0 08C2 329 MOVL #SS$_NORMAL,R0 ; Request is necessary
04 08C5 330 1$: RET

```

```

08C6 332 .SBTTL GET_QDNAME - Get the quorum disk name
08C6 333
08C6 334 : **
08C6 335 : This routine gets the full quorum disk name and makes sure that a
08C6 336 : copy is in the CLUB.
08C6 337 :
08C6 338 : CALLING SEQUENCE:
08C6 339 :
08C6 340 : JSB GET_QDNAME
08C6 341 :
08C6 342 : INPUTS:
08C6 343 :
08C6 344 : NONE
08C6 345 :
08C6 346 : OUTPUT:
08C6 347 :
08C6 348 : RO = Status
08C6 349 : R1-R7 Destroyed
08C6 350 :--
08C6 351
08C6 352 GET_QDNAME:
00000000'GF 28 08C6 353 MOV C3 #CLUDCBSS_DISK_QUORUM,- ; Get copy of quorum disk name
F739 CF 08C8 354 G^CLUSGB_QDISK,QD_NAME
10 20 3A 08D0 355 LOCC #^A/ /,#CLUDCBSS_DISK_QUORUM,- ; Locate end of name
F733 CF 08D3 356 QD_NAME
10 50 A3 08D6 357 SUBW3 RO,#CLUDCBSS_DISK_QUORUM,QD_DESCR ; Store name length
F725 CF 08D9
08DC 358 $GETDVIW_S DEVNAM = QD_DESCR,- ; Get full device name
08DC 359 ITMLST = QD_ITMLST,-
08DC 360 -; ASTADR = CSP$$RESUME,-
08DC 361 -; ASTPRM = CSP$GL_CURCIX,-
08DC 362 IOSB = IOSB
1C 50 E9 08FB 363 BLBC RO,1$ ; Br if error
50 F77A CF 3C 08FB 364 -; CALLS #0,CSP$$WAIT ; Wait for completion
14 50 E9 0900 365 MOVZWL IOSB,RO ; Get completion status
50 01 D0 0903 366 BLBC RO,1$ ; Br if error
04 11 0912 367 $CMKRNL_S GET_QDNAME1 ; Put name in CLUB
F6E6 CF B4 0915 368 MOVL #1,RO ; Return success
05 0917 369 BRB 2$
091C 370 1$: CLRW QD_DESCR ; Zero name size
091C 371 2$: RSB
091C 372
091C 373 GET_QDNAME1:
00000000'GF 0000 091C 374 .WORD 0
0088 C6 56 D0 091E 375 MOVL G^CLUSGL_CLUB,R6 ; Get CLUB address
0088 C6 95 0924 376 TSTB CLUBST_QDNAME(R6) ; Is name in CLUB already?
F6D1 CF 19 12 0925 377 BNEQU 1$ ; Br if yes
57 A3 0929 378 SUBW3 #2,QD_DESCR,R7 ; Get adjusted name size
F6D4 CF 57 28 092B 379 MOV C3 R7,QD_NAME+1,CLUBST_QDNAME+1(R6) ; Put name in CLUB
0089 C6 57 0930 380
0088 C6 57 90 0931 381 MOVB R7,CLUBST_QDNAME(R6) ; Put name size in CLUB
00000000'GF 16 0932 382 JSB G^CNX$DISK_CHANGE ; Tell connection manager
04 0944 382 1$: RET

```

```

0945 384 .SBTTL OPEN_FILE - Open the quorum file
0945 385
0945 386 :++
0945 387 : This routine "opens" the quorum file and obtains its logical block
0945 388 : number. It will first attempt to find any existing quorum file but
0945 389 : if unsuccessful, it will create a new one. Note that the logical
0945 390 : block number is only returned if the file was not created. This is
0945 391 : due to the fact that the statistics block is not returned on a create.
0945 392
0945 393 CALLING SEQUENCE:
0945 394 JSB OPEN_FILE
0945 395
0945 396 :
0945 397 INPUTS:
0945 398
0945 399 NONE
0945 400
0945 401 OUTPUT:
0945 402
0945 403 R0 = Status of file open
0945 404
0945 405 R1-R6 Destroyed
0945 406 :--
0945 407
0945 408 OPEN_FILE:
0945 409
0945 410 :
0945 411 : First we initialize the FIB.
0945 412 :
0945 413 MOVAL FIB,R6 ; Get FIB pointer
0945 414 MOVCS #0,(SP),#0,#FIBSK_LENGTH,(R6) ; Init FIB to all zeros
0945 415
0945 416 MOVL #FIBSM_WRITE!FIBSM_NOREAD!FIBSM_NOWRITE!FIBSM_WRITETHRU,-
0945 417 FIB$L_ACCTL(R6) ; Access bits
0945 418 MOVW #FIDSC_MFD,FIB$W_DID_NUM(R6) ; Directory is the MFD
0945 419 MOVW #FIDSC_MFD,FIB$W_DID_SEQ(R6)
0945 420 MOVW #FIBSM_EXTEND!FIBSM_ALCON!FIBSM_FILCON,- ; Extend control bits
0945 421 FIB$W_EXCTL(R6)
0945 422 MOVL #CLUQFSK_BLOCKS,FIB$L_EXSZ(R6) ; Allocation size
0945 423 :
0945 424 : Attempt to lookup/create the quorum file and access for read/write
0945 425 :
0945 426 $QIO_S CHAN = CHANNEL,-
0945 427 FUNC = #IOS_ACCESS!IOSM_ACCESS!IOSM_CREATE,-
0945 428 IOSB = IOSB,-
0945 429 ASTADR = CSP$$RESUME,-
0945 430 ASTPRM = CSP$GL_CURCTX,-
0945 431 P1 = FIB_DESCR,-
0945 432 P2 = #QF_DESCR,-
0945 433 P5 = #ATTRIB_BLOCK
0945 434 BLGC R0,1$ ; Br if error
0945 435 CALLS #0,CSP$$WAIT ; Wait for completion
0945 436
0945 437 MOVZWL IOSB,R0 ; Get I/O completion status
0945 438 BLBC R0,1$ ; Br if error
0945 439 :
0945 440 : Get the quorum file LBN.

```

```

56 F745 CF DE
00 6E 00 2C
66 0040 8F
00080501 8F DO
0A A6 04 B0
0C A6 04 B0
0085 8F B0
16 A6
18 A6 02 DO

```

```

1D 50 E9
00 FB
00000000'EF
50 F6C9 CF 3C
0E 50 E9

```

F734	CF	B0	0984	439 ;		
F686	CF		0984	440	MOVW	STATBLK+SBK\$W_STLBNL,LOOKUP_LBN ; Get the Low-order LBN
F728	CF	B0	0988	441	MOVW	STATBLK+SBK\$W_STLBNH,LOOKUP_LBN+2 ; Get the High-order LBN
F681	CF	05	098F			
			09C2	442 1\$:	RSB	


```

09C3 444 .SBTTL GET_LBN - Get the quorum file logical block number
09C3 445
09C3 446 :++
09C3 447 : This routine reads the quorum file attributes and gets the quorum
09C3 448 : file logical block number.
09C3 449 :
09C3 450 : CALLING SEQUENCE:
09C3 451 :
09C3 452 :     JSB     GET_LBN
09C3 453 :
09C3 454 : INPUTS:
09C3 455 :
09C3 456 :     NONE
09C3 457 :
09C3 458 : OUTPUT:
09C3 459 :
09C3 460 :     R0 = Status of file open
09C3 461 :
09C3 462 :     R1 Destroyed
09C3 463 :--
09C3 464
09C3 465 GET_LBN.
09C3 466
09C3 467     $QIO_S      CHAN      = CHANNEL,-      ; Read the file attributes
09C3 468                FUNC      = #IOS_ACCESS,-
09C3 469                IOSB      = IOSB,-
09C3 470                ASTADR    = CSP$$RESUME,-
09C3 471                ASTPRM   = CSP$GL_CURCTX,-
09C3 472                P1       = FIB_DESCR,-
09C3 473                P2       = #OF_DESCR,-
09C3 474                P5       = #ATTRIB_BLOCK
09C3 475                BLBC    RO,1$      ; Br if error
09F8 476                CALLS  #0,CSP$$WAIT ; Wait for completion
09FD
1D 50  E9 09FB 477                MOVZWL IOSB,R0      ; Get I/O completion status
00000000'EF  FB 09FB 478                BLBC    RO,1$      ; Br if error
50  F673 CF 3C 0A02 479 :
OE 50  E9 0A07 480 : Get the quorum file LBN.
0A0A 481 :
F6DE CF B0 0A0A 482                MOVW   STATBLK+SBK$W_STLBNL,LOOKUP_LBN ; Get the Low-order LBN
F660 CF 0A0E
F6D5 CF B0 0A11 483                MOVW   STATBLK+SBK$W_STLBNH,LOOKUP_LBN+2 ; Get the High-order LBN
F65B CF 0A15
05 0A18 484 1$: RSB
  
```

```

0A19 486 .SBTTL VALIDATE_FILE - Validate the quorum file
0A19 487
0A19 488 :++
0A19 489 :       This routine reads the quorum file and validates its contents.
0A19 490 :
0A19 491 : CALLING SEQUENCE:
0A19 492 :
0A19 493 :       ISB     VALIDATE_FILE
0A19 494 :
0A19 495 : INPUTS:
0A19 496 :
0A19 497 :       NONE
0A19 498 :
0A19 499 : OUTPUT:
0A19 500 :
0A19 501 :       R0 = Status of validate
0A19 502 :
0A19 503 :       If R0 = SSS_NORMAL the file is valid. If R0 = 0 the file is
0A19 504 :       invalid. Otherwise R0 contains an I/O status error.
0A19 505 :
0A19 506 :       R1-R3,R6,R7 Destroyed
0A19 507 :--
0A19 508
0A19 509 VALIDATE_FILE:
0A19 510
0A19 511 :
0A19 512 : Queue a read request to the quorum file
0A19 513 :
0A19 514 :       $QIO_S           CHAN     = CHANNEL,-
0A19 515 :                       FUNC     = #IOS_READLBLK,-
0A19 516 :                       IOSB     = IOSB,-
0A19 517 :                       ASTADR   = CSP$$RESUME,-
0A19 518 :                       ASTPRM   = CSP$GL_CURCTX,-
0A19 519 :                       P1       = QF_BUFFER,-
0A19 520 :                       P2       = #CLUQFSK_BLOCKS*512,-
0A19 521 :                       P3       = LOOKUP_LBN
0A19 522 :
0A4C 522 BLBC     R0,2$           ; Br if error
0A4F 523 CALLS   #0,CSP$$WAIT      ; Wait for completion
0A51
0A56 524 MOVZWL IOSB,R0           ; Get I/O completion status
0A5B 525 BLBC     R0,2$           ; Br if error
0A5E
0A5E 526 :
0A5E 527 : Validate the data in the quorum file
0A5E 528 :
0A5E 529 CLRL    -(SP)             ; Assume file not valid
0A60 530 MOVAL  QF_BUFFER,R6     ; Get buffer pointer
0A65 531 JSB    CALCULATE_CHECKSUM ; Get the checksum
0A68 532 TSTL   R7                ; Is checksum valid?
0A6D 533 BNEQU  1$                ; Br if not
0A6F 534 CMPC3  #CLUQFSK_IDENT,-  ; Validate ID area
0A71 535 CLUQFS IDENT(R6),-
0A72 536 IDENT_STRING
0A75 537 BNEQU  1$                ; Br if file invalid
0A77 538 CMPW   #CLUQFSK_VERSION,- ; Is version correct?
0A79 539 CLUQFSW_VERSION(R6)
0A7B 540 BNEQU  1$                ; Br if not
0A7D 541 MOVZBL #SSS_NORMAL,(SP)  ; File is valid

```

```

34 50 E9 0A4C 522
00 00 FB 0A4F 523
00000J00'EF 0A51
50 F61F CF 3C 0A56 524
25 50 E9 0A5B 525
0A5E 526
0A5E 527
0A5E 528
56 F6C0 CF D4 0A5E 529
00000B53'EF 16 0A60 530
57 D5 0A65 531
11 12 0A68 532
OC 29 0A6D 533
66 0A6F 534
F69B CF 0A71 535
09 12 0A72 536
02 B1 0A75 537
OC A6 0A77 538
03 12 0A79 539
6E 01 9A 0A7B 540
01 9A 0A7D 541

```

CSPQUORUM
V04-001

N 4
- CSP DISK QUORUM MODULE
VALIDATE_FILE - Validate the quorum file

8-JAN-1985 18:40:11
5-SEP-1984 04:08:58

VAX/VMS Macro V04-00 Page 16
[SYSLOA.BUGSRC]CSPQUORUM.MAR;1 (9)

50 8E D0 0AB0 542 1\$: MOVL (SP)+,R0
05 0AB3 543 2\$: RSB

; Return status

OPD
V04

```

0A84 545 .SBTTL WRITE_FILE - Write the quorum file
0A84 546
0A84 547 :++
0A84 548 :   This routine builds a template quorum file and writes it to the disk.
0A84 549 :
0A84 550 : CALLING SEQUENCE:
0A84 551 :
0A84 552 :   JSB   WRITE_FILE
0A84 553 :
0A84 554 : INPUTS:
0A84 555 :
0A84 556 :   NONE
0A84 557 :
0A84 558 : OUTPUT:
0A84 559 :
0A84 560 :   R0 = Status of the write
0A84 561 :
0A84 562 :   R1-R7 Destroyed
0A84 563 :--
0A84 564
0A84 565 WRITE_FILE:
0A84 566
0A84 567 :
0A84 568 :   Build a template quorum file
0A84 569 :
56   F69C CF   DE 0A84 570   MOVAL   QF_BUFFER,R6           ; Get buffer pointer
00   6E   00   2C 0A89 571   MOVCS   #0,(SP),#0,-           ; Zero buffer
66   0400 8F   0A8D 572   #CLUQF$K_BLOCKS*512,(R6)
        OC   28 0A91 573   MOVCS   #CLUQF$S_IDENT,-       ; Store ident string
        F67A CF   0A93 574   IDENT_STRING,-
        63   02   0A96 575   CLUQF$T_IDENT(R6)
00000B53'Ef 16 0A97 576   MOVW   #CLUQF$R_VERSION,(R3)   ; Store version number
44 A6   57   0A9A 577   JSB    CALCULATE_CHECKSUM     ; Get the checksum
48 A6   01   0AA0 578   MOVL   R7,CLUQF$C_CHECKSUM(R6) ; Store checksum
0AA8 579   MOVB   #1,CLUQF$B_IGNORE(R6) ; Set ignore flag
0AA8 580 :
0AA8 581 :   Write the template quorum file.
0AA8 582 :
0AA8 583 :   $QIO_S           CHAN   = CHANNEL,-
0AA8 584 :                   FUNC   = #IOS_WRITELBLK,-
0AA8 585 :                   IOSB   = IOSB,-
0AA8 586 :                   ASTADR = CSP$$RESUME,-
0AA8 587 :                   ASTPRM = CSP$GL_CURCTX,-
0AA8 588 :                   P1     = QF_BUFFER,-
0AA8 589 :                   P2     = #CLUQF$K_BLOCKS*512,-
0AA8 590 :                   P3     = LOOKUP_LBN
        OC 50   E9 0ADB 591   BLBC   R0,1$                 ; Br if error
        00   FB 0ADE 592   CALLS  #0,CSP$$WAIT         ; Wait for completion
00000000'EF 0AE0
50   F590 CF   3C 0AE5 593   MOVZWL IOSB,R0             ; Get I/O completion status
        05   05 0AEA 594 1$:  RSB

```

```

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

```

07 50 E9 OB14 622 BLBC RO,1\$; Br if error
00 FB OB17 623 CALLS #0,CSP\$\$WAIT ; Wait for completion
00000000'FF OB19
05 OB1E 624 1\$: RSB

```

OB1F 626 .SBTTL REQUEST_COMPLETE - Request completion
OB1F 627
OB1F 628 :++
OB1F 629 :      This routine completes the request by updating the CLUDCB fields.
OB1F 630 :
OB1F 631 :      CALLING SEQUENCE:
OB1F 632 :
OB1F 633 :      $CMKRNLS      REQUEST_COMPLETE
OB1F 634 :
OB1F 635 :      INPUTS:
OB1F 636 :
OB1F 637 :      NONE
OB1F 638 :
OB1F 639 :      OUTPUT:
OB1F 640 :
OB1F 641 :      R0,R1 Destroyed
OB1F 642 :--
OB1F 643
OB1F 644 REQUEST_COMPLETE:
OB1F 645
OB1F 646      .WORD      0
00000000'GF 0000 OB21 647      MOVL      G^CLUSGL_CLUB,R0      ; Get CLUB address
                    DO OB27
50 00B4 C0 DO OB28 648      MOVL      CLUB$ CLUDCB(R0),R0      ; Get CLUDCB address
51 F544 CF DO OB2D 649      MOVL      CHANNEC,R1      ; Get channel number
                    C3 OB32 650      SUBL3     R1,@#CTL$GL_CCBBASE,R1      ; Form CCB address
00000000'9F
                    OB34
                    OB39
OC A0 61 DO OB3A 651      SETIPL   #!PLS_TIMER      ; Synchronize access to CLUDCB
                    DO OB3D 652      MOVL     CCB$UCB(R1),CLUDCB$UCB(R0) ; Store UCB address in CLUDCB
                    DO OB41 653      MOVL     LOOKUP_LBN,CLUDCB$L_OF[BN(R0) ; Put LBN in CLUDCB
                    OB45
                    AA OB47 654      BICW    #CLUDCB$M_OF(CSPACK,-      ; Clear the in progress bit
                    OB49 655      CLUDCB$M_FLAGS(R0)
                    BO OB4B 656      MOVW   #CLUDCB$M_QS_READY,-      ; Set state to ready
                    OB4D 657      CLUDCB$M_STATE(R0)
                    OB4F 658      SETIPL  #0      ; Restore IPL
                    04 OB52 659      RET

```

```

    OB53 661 .SBTTL CALCULATE_CHECKSUM - Calculate the quorum file checksum
    OB53 662
    OB53 663 :++
    OB53 664 :      This routine calculates the checksum of the quorum block pointed to
    OB53 665 :      by R6. It includes the field CLUQFSL_CHECKSUM in the checksum
    OB53 666 :      calculation.
    OB53 667 :
    OB53 668 : CALLING SEQUENCE:
    OB53 669 :
    OB53 670 :     JSB     CALCULATE_CHECKSUM
    OB53 671 :
    OB53 672 : INPUTS:
    OB53 673 :
    OB53 674 :     R6 = Pointer to the quorum block
    OB53 675 :
    OB53 676 : OUTPUT:
    OB53 677 :
    OB53 678 :     R7 = Quorum block checksum
    OB53 679 :
    OB53 680 :     R2,R3  Destroyed
    OB53 681 :--
    OB53 682
    OB53 683 CALCULATE_CHECKSUM:
    OB53 684
    52 12 D0 OB53 685      MOVL    #CLUQFSK_CHECK_LENGTH/4,R2      ; R2 = checksum longword count
    53 56 D0 OB56 686      MOVL    R6,R3                          ; Copy buffer address
    57 57 D4 OB59 687      CLRL    R7                            ; Form checksum in R7
    FA 83 CC OB5B 688 1$: XORL2  (R3)+,R7                        ; Accumulate checksum
    05 F5 OB5E 689      SOBGTR  R2,1$                            ; Br if more
    OB61 690      RSB
    OB62 691
    OB62 692      .END
    
```

CSPQUORUM
Symbol table

- CSP DISK QUORUM MODULE

F 5

SST1	= 00000000			FIBSM_WRITE	= 00000100		
ATRSC_FPRO	= 000C0016			FIBSM_WRITETHRU	= 00080000		
ATRSC_STATBLK	= 00000009			FIBSW_DID_NUM	= 0000000A		
ATRSC_UIC	= 00000015			FIBSW_DID_SEQ	= 0000000C		
ATRSS_FPRO	= 00000002			FIBSW_EXCTL	= 00000016		
ATRSS_STATBLK	= 00000020			FIB_DESCR	= 00000086	R	01
ATRSS_UIC	= 00000004			FIDSC_MFD	= 00000004		
ATTRIB_BLOCK	000000CE	R	01	FPRO	0000010E	R	01
CALCULATE_CHECKSUM	00000B53	R	01	GET_LBN	000009C3	R	01
CCBSL_UCB	= 00000000			GET_QDNAME	000008C6	R	01
CHANNEL	00000075	R	01	GET_QDNAME1	0000091C	R	01
CLOSE_FILE	00000AEB	R	01	IDENT_STRING	00000110	R	01
CLUSGB_QDISK	*****	X	01	IOSM_ACCESS	= 00000040		
CLUSGL_CLUB	*****	X	01	IOSM_CREATE	= 00000080		
CLUSGW_QDSKINTERVAL	*****	X	01	IOS_ACCESS	= 00000032		
CLUBSL_CLUDCB	= 000000B4			IOS_DEACCESS	= 00000034		
CLUBST_QDNAME	= 000000B8			IOS_READBLK	= 00000021		
CLUDCBSL_QFLBN	= 0000001C			IOS_WRITEBLK	= 00000020		
CLUDCBSL_UCB	= 0000000C			IOSB	00000079	R	01
CLUDCBSM_QF_CSPACK	= 00000010			IPLS_TIMER	= 00000008		
CLUDCBSM_QS_NOT_READY	= 00000001			LOOKUP_LBN	00000071	R	01
CLUDCBSM_QS_READY	= 00000002			MSSG1	00000524	R	01
CLUDCBSS_DISK_QUORUM	= 00000010			MSSG10	000006F6	R	01
CLUDCBSW_FLAGS	= 00000022			MSSG2	00000568	R	01
CLUDCBSW_STATE	= 00000020			MSSG3	00000595	R	01
CLUDCB_LBN	= 0000006D	R	01	MSSG4	000005CF	R	01
CLUQFSB_IGNORE	= 00000048			MSSG5	000005FD	R	01
CLUQFSK_BLOCKS	= 00000002			MSSG6	0000062E	R	01
CLUQFSK_CHECK_LENGTH	= 00000048			MSSG7	00000662	R	01
CLUQFSK_VERSION	= 00000002			MSSG8	0000069A	R	01
CLUQFSL_CHECKSUM	= 00000044			MSSG9	000006C7	R	01
CLUQFSS_IDENT	= 0000000C			OPEN_FILE	00000945	R	01
CLUQFST_IDENT	= 00000000			PR\$ IPL	*****	X	01
CLUQFSW_VERSION	= 0000000C			QD_DESCR	00000001	R	01
CNX\$DISK_CHANGE	*****	X	01	QD_ITPLST	00000049	R	01
CSP\$RESOME	*****	X	01	QD_NAME	00000009	R	01
CSP\$SWAIT	*****	X	01	QF_BUFFER	00000124	R	01
CSP\$GL_CURCTX	*****	X	01	QF_DESCR	00000059	R	01
CSP\$QUORUM	00000729	RG	01	REQUEST_COMPLETE	00000B1F	R	01
CSP\$TELL_OPCOM	*****	X	01	REQUEST_INIT	00000890	R	01
CTL\$GL_CCBASE	*****	X	01	RESCHEDULE_TIMER	0000011C	R	01
DSC\$K_CLASS_D	= 00000002			SBK\$W_STLBNH	= 00000000		
DSC\$K_CLASS_S	= 00000001			SBK\$W_STLBNL	= 00000002		
DSC\$K_DTYPE_T	= 0000000E			SS\$_CREATED	= 00000619		
DVIS_FULLDEVNAM	= 000000E8			SS\$_FORCEDERROR	= 00002144		
ERROR_COUNT	00000081	R	01	SS\$_NORMAL	= 00000001		
ERROR_MESSAGE	00000082	R	01	STATBLK	000000EA	R	01
ERROR_THRESHOLD	= 0000000A			SYSS\$ASSIGN	*****	GX	01
FIB	0000008E	R	01	SYSS\$CMKRNL	*****	GX	01
FIB\$K_LENGTH	= 00000040			SYSS\$DASSGN	*****	GX	01
FIB\$L_ACCTL	= 00000000			SYSS\$GETDVIW	*****	GX	01
FIB\$L_EXSZ	= 00000018			SYSS\$QIO	*****	GX	01
FIB\$M_ALCON	= 00000001			SYSS\$SETIMR	*****	GX	01
FIB\$M_EXTEND	= 00000080			THREAD_ACTIVE	00000000	R	01
FIB\$M_FILCON	= 00000004			UIC	0000010A	R	01
FIB\$M_NOREAD	= 000000400			VALIDATE_FILE	00000A19	R	01
FIB\$M_NOWRITE	= 00000001			WRITE_FILE	00000A84	R	01

! 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 .	00000862 (2914.)	01 (1.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE
\$ABSS	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	63	00:00:00.29	00:00:01.19
Command processing	87	00:00:00.76	00:00:02.03
Pass 1	455	00:00:20.35	00:00:48.63
Symbol table sort	0	00:00:02.75	00:00:06.44
Pass 2	146	00:00:03.65	00:00:13.00
Symbol table output	15	00:00:00.10	00:00:00.31
Psect synopsis output	1	00:00:00.03	00:00:00.03
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	770	00:00:27.93	00:01:11.64

The working set limit was 1950 pages.
95950 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.
699 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
_\$255\$DUA18:[SYSLOA.OBJ]CLUSTER.MLB;2	1
-\$255\$DUA18:[SYS.OBJ]LIB.MLB;1	5
-\$255\$DUA18:[SYSLIB]STARLET.MLB;3	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=(BUG\$:CSPQUORUM)+EXECMLS/LIB+LIB\$:CLUSTER/LIB

0448 AH-EF71A-SE
VAX/VMS V4.1 SRC LST MCRF UPD

CSP
LIS

TTYCHAR1
LIS

TTDRVR
MAP

YCDRIVER
MAP

OPDRWS
LIS

CSPQUORUM
LIS

TTYCHAR0
LIS

The image displays a grid of 16 columns and 16 rows of source code listings. Each cell contains a small window of text, likely representing a single line or a small block of code from a larger file. The text is monospaced and appears to be a mix of comments and code. Some cells contain larger, more prominent text, such as the labels 'TTYCHAR1 LIS', 'TTDRVR MAP', 'YCDRIVER MAP', 'OPDRWS LIS', and 'CSPQUORUM LIS', which likely identify the source files being listed. The overall layout is a dense, structured grid of code snippets.