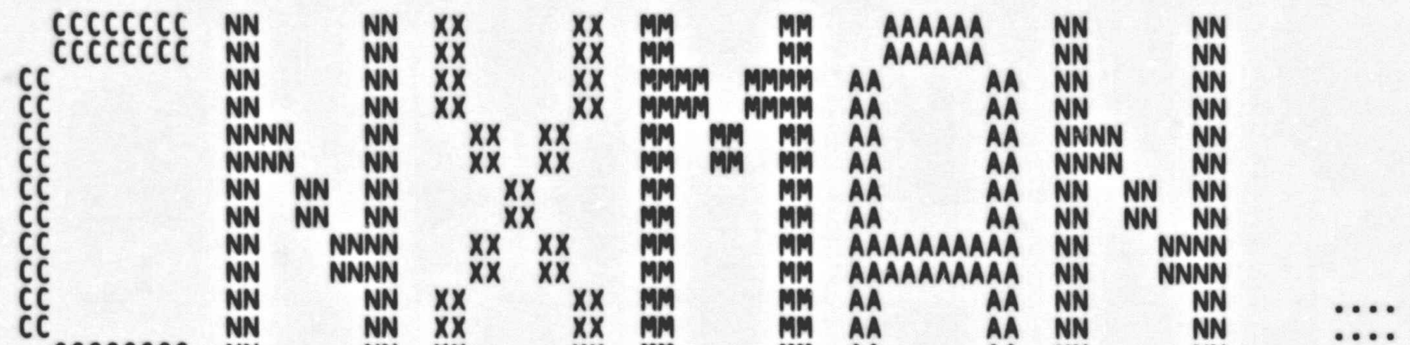
```
   SSSSSSSSSSS  YYY         YYY    SSSSSSSSSSS  LLL              OOOOOOOOO       AAAAAAAAA
   SSSSSSSSSSS  YYY         YYY    SSSSSSSSSSS  LLL              OOOOOOOOO       AAAAAAAAA
   SSSSSSSSSSS  YYY         YYY    SSSSSSSSSSS  LLL              OOOOOOOOO       AAAAAAAAA
SSS             YYY         YYY SSS             LLL           OOO         OOO   AAA         AAA
SSS             YYY         YYY SSS             LLL           OOO         OOO   AAA         AAA
SSS             YYY         YYY SSS             LLL           OOO         OOO   AAA         AAA
SSS                YYY   YYY    SSS             LLL           OOO         OOO   AAA         AAA
SSS                YYY   YYY    SSS             LLL           OOO         OOO   AAA         AAA
SSS                YYY   YYY    SSS             LLL           OOO         OOO   AAA         AAA
   SSSSSSSSS          YYY          SSSSSSSSS     LLL           OOO         OOO   AAA         AAA
   SSSSSSSSS          YYY          SSSSSSSSS     LLL           OOO         OOO   AAA         AAA
   SSSSSSSSS          YYY          SSSSSSSSS     LLL           OOO         OOO   AAA         AAA
            SSS       YYY                   SSS  LLL           OOO         OOO   AAAAAAAAAAAAAAA
            SSS       YYY                   SSS  LLL           OOO         OOO   AAAAAAAAAAAAAAA
            SSS       YYY                   SSS  LLL           OOO         OOO   AAAAAAAAAAAAAAA
            SSS       YYY                   SSS  LLL           OOO         OOO   AAA         AAA
            SSS       YYY                   SSS  LLL           OOO         OOO   AAA         AAA
            SSS       YYY                   SSS  LLL           OOO         OOO   AAA         AAA
SSSSSSSSSSS           YYY SSSSSSSSSSS     LLLLLLLLLLLLLLL        OOOOOOOOO     AAA         AAA
SSSSSSSSSSS           YYY SSSSSSSSSSS     LLLLLLLLLLLLLLL        OOOOOOOOO     AAA         AAA
SSSSSSSSSSS           YYY SSSSSSSSSSS     LLLLLLLLLLLLLLL        OOOOOOOOO     AAA         AAA
```

FILEID**CNXMAN



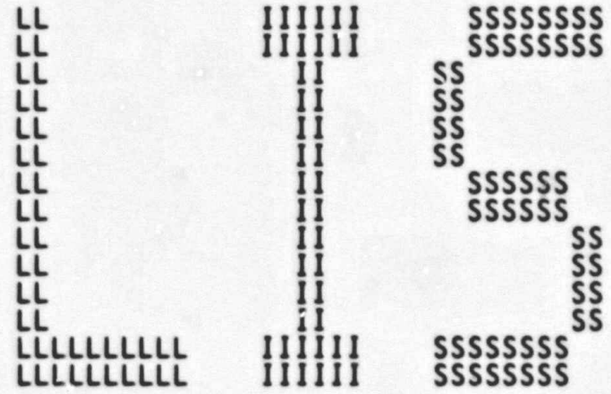


Table of contents

```
0000     1          .TITLE  CNXMAN - Cluster Connection Manager
0000     2          .IDENT  'V04-000'
0000     3 ;
0000     4 ;*****************************************************************************
0000     5 ;*                                                                           *
0000     6 ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
0000     7 ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
0000     8 ;*  ALL RIGHTS RESERVED.                                                     *
0000     9 ;*                                                                           *
0000    10 ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED    *
0000    11 ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0000    12 ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER    *
0000    13 ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY    *
0000    14 ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY    *
0000    15 ;*  TRANSFERRED.                                                             *
0000    16 ;*                                                                           *
0000    17 ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE    *
0000    18 ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0000    19 ;*  CORPORATION.                                                             *
0000    20 ;*                                                                           *
0000    21 ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS    *
0000    22 ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
0000    23 ;*                                                                           *
0000    24 ;*                                                                           *
0000    25 ;*****************************************************************************
0000    26 ;
0000    27
0000    28 ;++
0000    29 ; FACILITY: EXECUTIVE, CLUSTER MANAGEMENT
0000    30 ;
0000    31 ; ABSTRACT:
0000    32 ;       This module creates and manages connections to the other systems
0000    33 ;       in the cluster.
0000    34 ;
0000    35 ; ENVIRONMENT: VAX/VMS
0000    36 ;
0000    37 ; AUTHOR: Steve Beckhardt,       CREATION DATE: 17-Aug-1982
0000    38 ;
0000    39 ; MODIFIED BY:
0000    40 ;
0000    41 ;       V03-028 DWT0235         David W. Thiel          7-Aug-1984
0000    42 ;               Correct source of SCS message size for testing.
0000    43 ;
0000    44 ;       V03-027 DWT0228         David W. Thiel          24-Jul-1984
0000    45 ;               Modify send credits from 10 to 5.  Modify limit
0000    46 ;               on unacknowledged messages from 7 to 4.
0000    47 ;
0000    48 ;       V03-026 DWT0216         David W. Thiel          30-Apr-1984
0000    49 ;               Correct sequencing of events for "lasp gasp"
0000    50 ;               messages.
0000    51 ;
0000    52 ;       V03-025 DWT0206         David W. Thiel          07-Apr-1984
0000    53 ;               Add support for "Last Gasp" from a failing system.
0000    54 ;               Initialize CLUBPWF block in CLUB.
0000    55 ;
0000    56 ;       V03-024 DWT0202         David W. Thiel          25-Mar-1984
0000    57 ;               Remove all references to CNCT$V_QOURUM and
```

```
0000     58 ;                  CNCT$V_TRANSITION.
0000     59 ;
0000     60 ;       V03-023 DWT0191         David W. Thiel          21-Mar-1984
0000     61 ;               Update to support new ACKMSG.  Reinstate improved
0000     62 ;               version number checking.
0000     63 ;
0000     64 ;       V03-022 DWT0176         David W. Thiel          23-Feb-1984
0000     65 ;               Initialize CLUB$W_QDVOTES to largest integer when
0000     66 ;               creating CLUB.  Maintain SB$L_CSB as a pointer to
0000     67 ;               the newest CSB for a system.
0000     68 ;
0000     69 ;       V03-021 DWT0163         David W. Thiel          19-Jan-1984
0000     70 ;               Correct CNX$DISC_* routines.  Support forced
0000     71 ;               disconnection in the general case.  Rename
0000     72 ;               CNX_ERROR to CNX$ERROR.
0000     73 ;
0000     74 ;       V03-020 DWT0148         David W. Thiel          13-Dec-1983
0000     75 ;               Store SYSGEN parameters LCKDIRWT and QDSKVOTES in
0000     76 ;               the local CSB.  Restructure code.  Use CNX$ALLOZMEM
0000     77 ;               to allocate and zero pool.  Correct disabling of
0000     78 ;               polling once a node is firmly discovered.
0000     79 ;
0000     80 ;       V03-019 DWT0142         David W. Thiel          07-Nov-1983
0000     81 ;               Use symbolic protocol level (CNCT$K_PROTOCOL).
0000     82 ;
0000     83 ;       V03-018 DWT0127         David W. Thiel          30-Aug-1983
0000     84 ;               Pull console message routines out into new module
0000     85 ;               CLUMESSAG.MAR.
0000     86 ;               Disable process polling after accepting a connection.
0000     87 ;               Check more carefully for a fatal disconnect.
0000     88 ;
0000     89 ;       V03-017 DWT0117         David W. Thiel          24-Aug-1983
0000     90 ;               Replace systemid with node name in all messages.
0000     91 ;               Change CONFIG_CHANGE to CNX$CONFIG_CHANGE.  Update
0000     92 ;               protocol version level to 6 to mark incompatibility
0000     93 ;               with previous versions.
0000     94 ;
0000     95 ;       V03-016 DWT0116         David W. Thiel          2-Aug-1983
0000     96 ;               Increment protocol level to mark incompatibility
0000     97 ;               with previous versions.
0000     98 ;
0000     99 ;       V03-015 DWT0109         David W. Thiel          16-Jul-1983
0000    100 ;               Use CNX$CHECK_QUORUM to hang on lose of quorum.
0000    101 ;               Tolerate repeating software incarnation numbers.
0000    102 ;               Correct cleanup after an ACCEPT fails.  Clean up
0000    103 ;               code a little bit.  Improve some messages.
0000    104 ;
0000    105 ;       V03-014 ROW0185         Ralph O. Weber          21-JUN-1983
0000    106 ;               Change CSB SEL queue to block transfer partners BTX queue, to
0000    107 ;               support connection manager block transfers.  Remove CLUB
0000    108 ;               references to SEL queue.
0000    109 ;
0000    110 ;       V03-013 DWT0105         David W. Thiel          16-Jun-1983
0000    111 ;               Fail message on any call to LONG_BREAK.
0000    112 ;               Refuse to open connection when LONG_BREAK is set.
0000    113 ;
0000    114 ;       V03-012 DWT0098         David W. Thiel          14-May-1983
```

```
0000   115 ;               If incompatibile connection manager's see each other,
0000   116 ;               BUGCHECK one of them.
0000   117 ;               Remove temporary configuration management code and
0000   118 ;               integrate use of CONMAN module.
0000   119 ;               Move CNX$DISPATCH to CONMAN.
0000   120 ;               Add CSB$L_SB as pointer to SB.
0000   121 ;               More initialization of local CSB.
0000   122 ;               Dynamically allocate CLUB structure.
0000   123 ;
0000   124 ;       V03-011 ROW0186         Ralph O. Weber          25-APR-1983
0000   125 ;               Bump protocol version number to indicate use of two level
0000   126 ;               dispatching.  Add setup for CLUB$L_JNL_DISPT in CNX$INIT.
0000   127 ;               Change LCK$GL_DIRSYSCSB to LCK$GL_DIRSYSCSID.  Change setup to
0000   128 ;               put directory node CSID in there.  Add CNX$DISPATCH, the
0000   129 ;               target of the first level input dispatcher for FAC_CNX
0000   130 ;               messages.
0000   131 ;
0000   132 ;       V03-010 DWT0093         David W. Thiel          15-Apr-1983
0000   133 ;               Track changes in $CLUBDEF.
0000   134 ;
0000   135 ;       V03-009 DWT0090         David W. Thiel          31-Mar-1983
0000   136 ;               Add reconnection data to detect partitioned clusters.
0000   137 ;               Extend CSB and CLUB.
0000   138 ;               Change protocol version to 2.
0000   139 ;
0000   140 ;       V03-008 DWT0085         David W. Thiel          14-Mar-1983
0000   141 ;               Avoid attempt to output message during initialization.
0000   142 ;               Correct misuse of stack for connect data.
0000   143 ;
0000   144 ;       V03-007 DWT0084         David W. Thiel          12-Mar-1983
0000   145 ;               Correct bug that allows a reconnect to be sent to
0000   146 ;               a recently rebooted system.  Log CSB creations.
0000   147 ;
0000   148 ;       V03-006 SRB0070         Steve Beckhardt         10-Mar-1983
0000   149 ;               Added routine to send the job controller a message
0000   150 ;               when a system is removed from the cluster.  This is
0000   151 ;               a temporary change.
0000   152 ;
0000   153 ;       V03-005 DWT0083         David W. Thiel          10-Mar-1983
0000   154 ;               Replace HALTs with generic connection manager
0000   155 ;               BUG_CHECKs.  Create and use CLUster Block.  Change
0000   156 ;               SYSAP name.
0000   157 ;
0000   158 ;       V03-004 DWT0082         David W. Thiel          3-Mar-1983
0000   159 ;               Correct use of disconnect/reject status codes.
0000   160 ;
0000   161 ;       V03-003 DWT0070         David W. Thiel          21-Feb-1983
0000   162 ;               Major revision which includes:
0000   163 ;               Initialize automatically on being loaded.
0000   164 ;               Use SCA Process Poller to find new systems.
0000   165 ;               More state oriented structure.
0000   166 ;               Split out acknowledged message services as ACKMSG.
0000   167 ;
0000   168 ;       V03-002 SRB0064         Steve Beckhardt         21-Jan-1983
0000   169 ;               Removed cell LCK$GL_DIRSYSCSB as it now resides in the
0000   170 ;               EXEC (in module SYSCOMMON).
0000   171 ;
```

```
0000   172 ;--
0000   173
```

```
                        .SBTTL  DECLARATIONS
                ;
                ; INCLUDE FILES:
                ;
                        $CDTDEF                         ; CDT Offsets
                        $CLMDRSDEF                      ; Cluster disconnect/reject codes
                        $CLSMSGDEF                      ; Cluster message definitions
                        $CLUBDEF                        ; CLUster Block offsets
                        $CNCTDEF                        ; CONNECT data offsets
                        $CSBDEF                         ; CSB Offsets
                        $DYNDEF                         ; Data structure type codes
                        $FKBDEF                         ; Fork block offsets
                        $IPLDEF                         ; IPL definitions
                        $MSGDEF                         ; Mailbox message type codes
                        $PBDEF                          ; PB Offsets
                        $PDTDEF                         ; PDT Offsets
                        $SBDEF                          ; SB Offsets
                        $SCSCMGDEF                      ; SCS CONNECT Message definitions
                        $TQEDEF                         ; TQE offsets

                ;
                ; MACROS:
                ;

                        .MACRO  STATE_DISP      LIST
                        BSBW    DISPATCH
                        .IRP    ITEM,<LIST>
                        STATE_ENTRY     ITEM
                        .ENDR
                        .BYTE   0                       ; Flag end
                        .ENDM   STATE_DISP

                        .MACRO  STATE_ENTRY     CODE,ADDR,?LABEL
                        .BYTE   CSB$K_'CODE
                        .SIGNED_WORD    ADDR-LABEL
                LABEL:  .ENDM   STATE_ENTRY

                ;
                ; EQUATED SYMBOLS:
                ;

00000005                SEND_CREDITS = 5                        ; Connection send credits
                ;
                ;       The unacknowledged message limit is set to SEND_CREDITS-1
                ;       in order to force an explicit acknowledgement message to
                ;       be sent BEFORE SCS will ever consider sending an explicit
                ;       credit message.  This is done because the acknowledgement
                ;       message costs the same as the credit message and does
                ;       more (it completes sending threads).  This formula for
                ;       choosing an unacknowledged message limit is conservative
                ;       in that message buffers cached in the warm CDRP cache
                ;       instead of being deallocated cannot cause explicit SCS
                ;       credit messages.
                ;

                ;
                ; OWN STORAGE:
```

```
                                0000   232 ;
                                0000   233
                           00000000    234         .PSECT  $$$040,LONG             ; R/W Data PSECT
                                0000   235
                                0000   236 PROC_NAME:                             ; SYSAP name
74 73 75 6C 63 58 41 56 24 53 4D 56     237         .ASCII  "VMS$VAXcluster"
                                   72 65
                                000C
                                000E   238         .REPT   16-<.-PROC_NAME>
                                000E   239         .ASCII  " "
                                20     240         .ENDR
                                0010   241
                                0010   242 ;****************************************************************************
                                0010   243 ;
                                0010   244 ; NOTE: The following assumptions are in effect for this entire module.
                                0010   245 ;
                                0010   246 ;****************************************************************************
                                0010   247
                                0010   248         ASSUME  IPL$_SYNCH  EQ  IPL$_SCS
                                0010   249         ASSUME  IPL$_SYNCH  EQ  IPL$_TIMER
                                0010   250
                                0010   251         .DEFAULT        DISPLACEMENT,WORD
```

```
0010   253 ;++
0010   254 ;
0010   255 ; STATE-ORIENTED DESCRIPTION OF CONNECTION MANAGEMENT LAYER:
0010   256 ;
0010   257 ;
0010   258 ;       The connection manager is organized as a state machine.
0010   259 ;       Each connection has its own independent state machine.  Each
0010   260 ;       connection is represented by a Connection Status Block (CSB).
0010   261 ;       The state of a connection is defined by the contents of the
0010   262 ;       CSB$B_STATE field of the CSB.  The states are defined by
0010   263 ;       symbols of the format:  CSB$K_state where "state" is the
0010   264 ;       state name.
0010   265 ;
0010   266 ; STATES:
0010   267 ;
0010   268 ;    NEW
0010   269 ;       Brand new connection block created as the result of a
0010   270 ;       reference to a node id/software incarnation for which no
0010   271 ;       CSB existed.
0010   272 ;
0010   273 ;    CONNECT
0010   274 ;       Initial connect request to a newly discovered system in
0010   275 ;       progress.
0010   276 ;
0010   277 ;    ACCEPT
0010   278 ;       Initial connection from a newly discovered system being
0010   279 ;       accepted.
0010   280 ;
0010   281 ;    OPEN
0010   282 ;       Connection to a system exists and is available for use.
0010   283 ;       This is the "normal" state of a CSB.
0010   284 ;
0010   285 ;    DISCONNECT
0010   286 ;       Disconnect of an open connection in progress.
0010   287 ;
0010   288 ;    WAIT
0010   289 ;       Timeout in progress.  On conclusion of the timeout, an
0010   290 ;       attempt will be made to reconnect to the remote system.
0010   291 ;
0010   292 ;    RECONNECT
0010   293 ;       Connect in progress to a system to which a previous
0010   294 ;       connection broke.
0010   295 ;
0010   296 ;    REACCEPT
0010   297 ;       Accept in progress to a system to which a previous
0010   298 ;       connection broke.
0010   299 ;
0010   300 ;    DEAD
0010   301 ;       A new incarnation of the node has been seen.
0010   302 ;       No connection new connection to the incarnation specified
0010   303 ;       by the CSB is possible, for obvious reasons.
0010   304 ;
0010   305 ;    LOCAL
0010   306 ;       Special state only found for the local node.
0010   307 ;
0010   308 ;--
```

```
                                0010   310          .SBTTL  CNX$INIT - Initialize connection manager
                                0010   311
                                0010   312 ;++
                                0010   313 ;
                                0010   314 ; FUNCTIONAL DESCRIPTION:
                                0010   315 ;
                                0010   316 ;       This routine is called during system booting to initialize
                                0010   317 ;       the connection manager.
                                0010   318 ;
                                0010   319 ; CALLING SEQUENCE:
                                0010   320 ;
                                0010   321 ;       JSB     CNX$INIT
                                0010   322 ;       IPL is 31
                                0010   323 ;
                                0010   324 ; INPUT PARAMETERS:
                                0010   325 ;
                                0010   326 ;       NONE
                                0010   327 ;
                                0010   328 ; OUTPUT PARAMETERS:
                                0010   329 ;
                                0010   330 ;       NONE
                                0010   331 ;
                                0010   332 ; COMPLETION CODES:
                                0010   333 ;
                                0010   334 ;       NONE
                                0010   335 ;
                                0010   336 ; SIDE EFFECTS:
                                0010   337 ;
                                0010   338 ;       R0-R5 are destroyed
                                0010   339 ;
                                0010   340 ;--
                                0010   341
                    00000000    0000   342          .PSECT  $$$002                  ; Initialization section
                                0000   343
                                0000   344 CNX$INIT::
    51   01A8  8F    3C         0000   345          MOVZWL  #CLUB$K_LENGTH,R1       ; Length of CLUB
                FFF8'   30         0005   346          BSBW    CNX$ALLOZMEM            ; Allocate and zero memory
              3D  50    E9         0008   347          BLBC    R0,5$                   ; Branch if memory not available
00000000'GF     52    D0         000B   348          MOVL    R2,G^CLU$GL_CLUB        ; Store for the world to see
          62    62    7E         0012   349          MOVAQ   CLUB$L_CSBQFL(R2), -    ; Initialize CSB queue
                                0015   350                  CLUB$L_CSBQFL(R2)
        04  A2    62    7E         0015   351          MOVAQ   CLUB$L_CSBQFL(R2), -
                                0019   352                  CLUB$L_CSBQBL(R2)
        0B  A2    03    90         0019   353          MOVB    #DYN$C_CLU_CLUB, -      ; Block subtype
                                001D   354                  CLUB$B_SUBTYPE(R2)
 14 A2    0000'CF    DE         001D   355          MOVAL   W^CJF$MIN_JOURNAL,CLUB$L_JNL_DISPT(R2) ; Init journal dispatch
      66  A2    01    B0         0023   356          MOVW    #1,CLUB$W_FIRST_INDEX(R2)       ; Next available CSID slot
    00AE  C2    00    B2         0027   357          MCOMW   #0,CLUB$W_QDVOTES(R2)   ; Initialize to maximum possible votes
                                002C   358          ASSUME  CLUBFKB$S_FORK_BLOCK GE FKB$K_LENGTH
    50   00CC  C2    9E         002C   359          MOVAB   CLUB$B_FORK_BLOCK(R2),R0        ; Address of fork block
      0B  A0    08    90         0031   360          MOVB    #IPL$_SCS,FKB$B_FIPL(R0) ; Store IPL in fork block
                                0035   361          ASSUME  CLUBPWF$S_FORK_BLOCK GE FKB$K_LENGTH
    50   018C  C2    9E         0035   362          MOVAB   CLUB$B_CLUBPWFT(R2),R0  ; Power recovery fork block
      0B  A0    08    90         003A   363          MOVB    #IPL$_SCS,FKB$B_FIPL(R0) ; Store IPL in fork block
51   00000000'GF    DE         003E   364          MOVAL   G^SCS$GA_LOCALSB,R1
            0598'   30         0045   365          BSBW    CNX$CREATE_CSB          ; Create CSB for local system
          63  50    E9         0048   366 5$:      BLBC    R0,7$                   ; Branch on error
```

```
                51    64 A5   D0  004B   367          MOVL    CSB$L_CLUB(R5),R1          ; Get address of cluster block
                10 A1    55   D0  004F   368          MOVL    R5,CLUB$L_LOCAL_CSB(R1)    ; Local system CSB address
                50    7C A5   9E  0053   369          MOVAB   CSB$B_CNCT(R5),R0          ; Connect data template
                40 A5    60   90  0057   370          MOVB    CNCT$B_ECOLVL(R0), -       ; Fill in protocol ECO level
                                 005B   371                  CSB$B_ECOLVL(R5)
             41 A5    01 A0   90  005B   372          MOVB    CNCT$B_VERNUM(R0), -       ; Fill in protocol version
                                 0060   373                  CSB$B_VERNUM(R5)
          50 A5    00000000'GF   B0  0060   374          MOVW    G^CLU$GW_VOTES, -          ; Votes held by local system
                                 0068   375                  CSB$W_VOTES(R5)
          52 A5    00000000'GF   B0  0068   376          MOVW    G^CLU$GW_QUORUM, -         ; Local system proposed quorum
                                 0070   377                  CSB$W_QUORUM(R5)
          56 A5    00000000'GF   B0  0070   378          MOVW    G^CLU$GW_QDSKVOTES, -      ; Local system proposed quorum disk votes
                                 0078   379                  CSB$W_QDVOTES(R5)
          54 A5    00000000'GF   B0  0078   380          MOVW    G^CLU$GW_LCKDIRWT, -       ; Lock manager directory system weight
                                 0080   381                  CSB$W_LCKDIRWT(R5)
          60 A5    01000000 8F   C8  0080   382          BISL2   #CSB$M_LOCAL, -            ; Mark this the local CSB
                                 0088   383                  CSB$L_STATUS(R5)
                43 A5    0B    90  0088   384          MOVB    #CSB$K_LOCAL, -            ; Set state of local CSB
                                 008C   385                  CSB$B_STATE(R5)
                                 008C   386 ;
                                 008C   387 ; LISTEN for incoming CONNECTS.
                                 008C   388 ;
                                 008C   389          LISTEN  MSGADR = CNX$RCV_CNCT_MSG, -    ; Listen for incoming CONNECTs
                                 008C   390                  LPRNAM = PROC_NAME, -
                                 008C   391                  PRINFO = (R0), -           ; Use data in local CSB
                                 008C   392                  ERRADR = LISTEN_ERROR
                   49 50   E9  00A3   393          BLBC    R0,10$                     ; Branch on failure
             51    040C 8F   3C  00A6   394          MOVZWL  #12+<256*4>,R1             ; Length of cluster vector
                      FF52'  30  00AB   395          BSBW    CNX$ALLOZMEM               ; Allocate and zero memory
                   3E 50   E9  00AE   396 7$:       BLBC    R0,10$                     ; Branch on failure
                0B A2    02   90  00B1   397          MOVB    #DYN$C_CLU_CLUVEC,11(R2)   ; Store sub-type
       00000000'GF    0C A2   9E  00B5   398          MOVAB   12(R2),G^CLU$GL_CLUSVEC    ; Store vector address
    00000000'GF    0100 8F   B0  00BD   399          MOVW    #256,G^CLU$GW_MAXINDEX     ; Maximum cluster vector index+1
                                 00C6   400
             50    0007'CF   9E  00C6   401          MOVAB   W^CNX$NEWSYSTEM,R0         ; Address of new system routine
             52    0000'CF   9E  00CB   402          MOVAB   PROC_NAME,R2               ; Address of process name
              00000000'GF     16  00D0   403          JSB     G^SCS$POLL_PROC            ; Poll for copies of self
                      16 50   E9  00D6   404          BLBC    R0,10$                     ; Branch on failure
          55  00000000'GF     D0  00D9   405          MOVL    G^CLU$GL_CLUB,R5           ; Get address of cluster block
              0C A5    51     D0  00E0   406          MOVL    R1,CLUB$L_POLL_CTX(R5)     ; Save context for later calls
                                 00E4   407                                             ; R0 is odd -- enable polling
                                 00E4   408                                             ; R1 is address of SPPB
                      52     D4  00E4   409          CLRL    R2                         ; All systems now and forever
              00000000'GF     16  00E6   410          JSB     G^SCS$POLL_MODE            ; Enable polling
                         FF11' 30  00EC   411          BSBW    CNX$CON_INIT               ; Initialize configuration manager
                               05  00EF   412 10$:     RSB                                ; Return status
                                 00F0   413
                               00000000   414          .PSECT  $$$100,LONG
                                 0000   415 LISTEN_ERROR:
                                 0000   416          DISCONNECT                         ; Clean up error
                               05  0006   417          RSB                                ;  and return
```

```
                                                B 9
                                .SBTTL  CNX$NEWSYSTEM - New system heard from

                                ;++
                                ;
                                ; FUNCTIONAL DESCRIPTION:
                                ;
                                ;       This routine is called when the SCA Process Poller detects
                                ;       a possible new system.
                                ;
                                ; CALLING SEQUENCE:
                                ;
                                ;       JSB     CNX$NEWSYSTEM
                                ;       IPL must be at IPL$_SYNCH
                                ;
                                ; INPUT PARAMETERS:
                                ;
                                ;       R0 is context (nothing interesting)
                                ;       R1 is address of process name
                                ;       R2 is address of system ID
                                ;       R3 is address of process information
                                ;
                                ; OUTPUT PARAMETERS:
                                ;
                                ;       NONE
                                ;
                                ; COMPLETION CODES:
                                ;
                                ;       R0 odd = success
                                ;              Disable polling on the system
                                ;       R0 even = failure
                                ;              Continue polling the system
                                ;
                                ; SIDE EFFECTS:
                                ;
                                ;       R1-R5 are destroyed
                                ;
                                ;--
                                .PSECT  $$$100,LONG

                                CNX$NEWSYSTEM::
51    52    D0                  MOVL    R2,R1                   ; Address of System ID
      52    D4                  CLRL    R2                      ; No buffer
00000000'GF 16                  JSB     G^SCS$CONFIG_SYS        ; Find SB
     19 50  E9                  BLBC    R0,30$                  ; Branch on not found
      057E  30                  BSBW    CNX$LOOKUP_CSB          ; Find or allocate a CSB
     13 50  E9                  BLBC    R0,30$                  ; Branch on invalid system ID
                                STATE_DISP      <<NEW,20$>,<CONNECT,30$>,<ACCEPT,30$>>
50    01    D0                  MOVL    #1,R0                   ; Disable polling
      05                        RSB                             ; Return

      03    10                  BSBB    CNX$NEW_CSB             ; Do initial connect to new CSB
      50    D4                  CLRL    R0                      ; Continue polling
      05                        RSB                             ; Unable to allocate memory
```

```
                     0031   474          .SBTTL  CNX$NEW_CSB - Connect to new system
                     0031   475
                     0031   476 ;++
                     0031   477 ;
                     0031   478 ; FUNCTIONAL DESCRIPTION:
                     0031   479 ;
                     0031   480 ;       This routine is called to do the initial connect to a
                     0031   481 ;       new system.
                     0031   482 ;
                     0031   483 ; CALLING SEQUENCE:
                     0031   484 ;
                     0031   485 ;       JSB     CNX$NEW_CSB
                     0031   486 ;       IPL must be at IPL$_SYNCH
                     0031   487 ;
                     0031   488 ; INPUT PARAMETERS:
                     0031   489 ;
                     0031   490 ;       R5 is the CSB address
                     0031   491 ;
                     0031   492 ; OUTPUT PARAMETERS:
                     0031   493 ;
                     0031   494 ;       NONE
                     0031   495 ;
                     0031   496 ; COMPLETION CODES:
                     0031   497 ;
                     0031   498 ;       NONE
                     0031   499 ;
                     0031   500 ; SIDE EFFECTS:
                     0031   501 ;
                     0031   502 ;       R0-R4 are destroyed
                     0031   503 ;
                     0031   504 ;--
                     0031   505
                     0031   506 CNX$NEW_CSB::
   05      90        0031   507          MOVB    #CSB$K_CONNECT,-        ; Set state
43 A5                0033   508                  CSB$B_STATE(R5)
   01      10        0035   509          BSBB    CNX$CONNECT             ; Request connection
           05        0037   510          RSB                             ; Unable to allocate memory
```

```
                        0038   512          .SBTTL  CNX$CONNECT - Connect to remote system
                        0038   513
                        0038   514 ;++
                        0038   515 ;
                        0038   516 ; FUNCTIONAL DESCRIPTION:
                        0038   517 ;
                        0038   518 ;       This routine is called to initiate a connection to the
                        0038   519 ;       connection manager on a remote system.
                        0038   520 ;
                        0038   521 ; CALLING SEQUENCE:
                        0038   522 ;
                        0038   523 ;       JSB     CNX$CONNECT
                        0038   524 ;       IPL must be at IPL$_SYNCH
                        0038   525 ;
                        0038   526 ; INPUT PARAMETERS:
                        0038   527 ;
                        0038   528 ;       R5 is address of initialized CSB
                        0038   529 ;
                        0038   530 ; OUTPUT PARAMETERS:
                        0038   531 ;
                        0038   532 ;       NONE
                        0038   533 ;
                        0038   534 ; COMPLETION CODES:
                        0038   535 ;
                        0038   536 ;       NONE
                        0038   537 ;
                        0038   538 ; SIDE EFFECTS:
                        0038   539 ;
                        0038   540 ;       R0-R5 are destroyed
                        0038   541 ;
                        0038   542 ;--
                        0038   543
                        0038   544 CNX$CONNECT::
                        0038   545 ;
                        0038   546 ; Try to connect
                        0038   547 ; This thread may be suspended here
                        0038   548 ;
          03C4   30     0038   549          BSBW    CNCT_DATA                       ; Set up connect data
    50    68 A5  D0     003B   550          MOVL    CSB$L_SB(R5),R0                 ; Address of SB
                        003F   551          CONNECT MSGADR = CNX$RCV_MSG,-          ; Connect to system
                        003F   552                  ERRADR = CNX$ERROR,-
                        003F   553                  LPRNAM = PROC_NAME,-
                        003F   554                  RPRNAM = PROC_NAME,-
                        003F   555                  RSYSID = SB$B_SYSTEMID(R0),-
                        003F   556                  INITCR = #SEND_CREDITS,-
                        003F   557                  CONDAT = CSB$B_CNCT(R5),-
                        003F   558                  AUXSTR = (R5)
        03 50   E8      0072   559          BLBS    R0,5$
         04DE   30      0075   560          BSBW    CNX_STATUS_CHECK                ; Check for bugcheck request
                        0078   561 5$:
                        0078   562          STATE_DISP      <<CONNECT,100$>,<RECONNECT,200$>,<DEAD,300$>>
        0B 50   E9      0085   563 10$:     BLBC    R0,20$                          ; OK if connect failed
    50  8004 8F   3C    0088   564          MOVZWL  #<CLMDRS$M_DRS ! -              ; Disconnect status
                        008D   565                  CLMDRS$C_PROTOCOL>,R0
                        008D   566 15$:     DISCONNECT                              ; Break anomalous connection
                05      0093   567 20$:     RSB                                     ; Drop thread
                        0094   568
```

```
                50   8104 8F    3C   0094    569 90$:    MOVZWL  #<CLMDRS$M_DRS ! -          ; Disconnect status
                                     0099    570                   CLMDRS$M_FATAL ! -      ; Bugcheck request
                                     0099    571                   CLMDRS$C_PROTOCOL>,R0
                          F2    11   0099    572         BRB     15$
                                     009B    573
                                     009B    574 ;
                                     009B    575 ; Initial connect attempt completed
                                     009B    576 ;
                                     009B    577 ;       R2 is address of connection message
                                     009B    578 ;       R3 is address of CDT
                                     009B    579 ;       R4 is address of PDT
                                     009B    580 ;       R5 is address of CSB
                                     009B    581 ;
                             43 50   E9   009B    582 100$:   BLBC    R0,150$                 ; Connection error
                                     009E    583         ASSUME  CNCT$B_VERNUM  EQ  CNCT$B_ECOLVL+1
                                     009E    584         ASSUME  CSB$B_VERNUM  EQ CSB$B_ECOLVL+1
                             20 A2   B0   009E    585         MOVW    SCSCMG$B_SNDDAT+CNCT$B_ECOLVL(R2),- ; Store remote side's
                             40 A5        00A1    586                 CSB$B_ECOLVL(R5)        ; protocol version number and ECO level
                             23 A2   90   00A3    587         MOVB    SCSCMG$B_SNDDAT+CNCT$B_ACKLIM(R2),- ; Store remote side's
                             33 A5        00A6    588                 CSB$B_REMACKLIM(R5)     ; ACK limit
                              037F   30   00A8    589         BSBW    CNCT_CHECK              ; Check connect data
                             E6 50   E9   00AB    590         BLBC    R0,90$                  ; Bugcheck remote node
                                     00AE    591         ASSUME  CSB$L_PDT  EQ  CSB$L_CDT+4
                          0C A5   53   7D   00AE    592         MOVQ    R3,CSB$L_CDT(R5)        ; Store CDT and PDT address
                                01   90   00B2    593         MOVB    #CSB$K_OPEN,-           ; Mark connection open
                                43 A5        00B4    594                 CSB$B_STATE(R5)
                                52   DD   00B6    595         PUSHL   R2                      ; Save connection message addr
                           52   68 A5   D0   00B8    596         MOVL    CSB$L_SB(R5),R2         ; Address of System Block
                           52   18 A2   9E   00BC    597         MOVAB   SB$B_SYSTEMID(R2),R2    ; Address of destination system ID
                           51   64 A5   D0   00C0    598         MOVL    CSB$L_CLUB(R5),R1       ; Address of cluster block
                           51   0C A1   D0   00C4    599         MOVL    CLUB$L_POLL_CTX(R1),R1  ; Address of SPPB
                                50   D4   00C8    600         CLRL    R0                      ; Disable polling
                         00000000'GF    16   00CA    601         JSB     G^SCS$POLL_MODE         ; Disable polling this system
                                04   BA   00D0    602         POPR    #^M<R2>                 ; Restore connection message addr
                         50   0000'CF    9E   00D2    603         MOVAB   CNCT_MSG,R0             ; Connect message address
                                FF26'   30   00D7    604         BSBW    CNX$CONFIG_CHANGE       ; Note configuration change
                                FF23'   30   00DA    605         BSBW    CNX$RESEND_MSGS         ; Initialize connection
                                FF20'   30   00DD    606         BSBW    CNX$CON_NEWSYS          ; Inform configuration manager of new system
                                05   00E0    607         RSB                             ; Return
                                     00E1    608
                                     00E1    609 150$:   ; Come here on failure to make a connection.
                                     00E1    610
                                0424 30   00E1    611         BSBW    CNX$DECREFCNT           ; Deallocate CSB and return
                                05   00E4    612         RSB
                                     00E5    613
                                     00E5    614 ;
                                     00E5    615 ; Reconnect completed
                                     00E5    616 ;
                                3A 50   E9   00E5    617 200$:   BLBC    R0,220$
                                     00E8    618         ASSUME  CSB$L_PDT  EQ  CSB$L_CDT+4
                          0C A5   53   7D   00E8    619         MOVQ    R3,CSB$L_CDT(R5)        ; Store CDT and PDT address
                                0389 30   00EC    620         BSBW    RECNCT_CHECK            ; Check data following reconnect
                                59 50   E9   00EF    621         BLBC    R0,280$                 ; Other node should crash
                       4D 60 A5   00   E0   00F2    622         BBS     #CSB$V_LONG_BREAK,-     ; Disconnect if long break
                                     00F7    623                 CSB$L_STATUS(R5),270$   ;   has been seen
                                23 A2   90   00F7    624         MOVB    SCSCMG$B_SNDDAT+CNCT$B_ACKLIM(R2),- ; Store remote side's
                                33 A5        00FA    625                 CSB$B_REMACKLIM(R5)     ; ACK limit
```

```
                    01   90   00FC   626          MOVB    #CSB$K_OPEN,-                   ; Mark connection open
                 43 A5        00FE   627                  CSB$B_STATE(R5)
                   FEFD'      30   0100   628          BSBW    CNX$CHECK_QUORUM            ; Resume activity on if quorum
              50    0000'CF   9E   0103   629          MOVAB   RECNCT_MSG,R0               ; Address of reconnect message
                   FEF5'      30   0108   630          BSBW    CNX$CONFIG_CHANGE           ; Note configuration change
                              0108   631          MOVW    -                               ; Get last received sequence number
                50   2C A2    B0   010B   632                  SCSCMG$B_SNDDAT+CNCT$W_RCVDSEQNM(R2),R0
                50   30 A5    A3   010F   633          SUBW3   CSB$W_ACKRSEQNM(R5), -      ; Is it .ge. last one?
                        51         0113   634                  R0,R1
                        08   19   0114   635          BLSS    210$                        ; Branch if not and bugcheck
                30 A5    50   B0   0116   636          MOVW    R0,CSB$W_ACKRSEQNM(R5)      ; Store updated number
                   FEE3'      30   011A   637          BSBW    CNX$RESEND_MSGS             ; Send pending message, initialize connectio
                              05   011D   638          RSB
                                   011E   639
                                   011E   640 210$:     BUG_CHECK       CNXMGRERR,FATAL ; Invalid acknowledged sequence number
                                   0122   641
             0D 60 A5    00    E0   0122   642 220$:     BBS     #CSB$V_LONG_BREAK, -        ; Branch if long break has already
                                   0127   643                  CSB$L_STATUS(R5),240$       ;  been seen
          48 A5    00000000'GF   D1   0127   644          CMPL    G^EXE$GL_ABSTIM, -          ; Have we retried for long enough?
                                   012F   645                  CSB$L_TIMEOUT(R5)
                        0A   1F   012F   646          BLSSU   250$                        ; Not timeout out yet
                      03BA   30   0131   647          BSBW    LONG_BREAK                  ; Long break seen
                54    2710   8F   3C   0134   648 240$:     MOVZWL  #10000,R4               ; 10 sec = 10000 ms timeout
                        05   11   0139   649          BRB     260$
                                   013B   650
                54    03E8   8F   3C   013B   651 250$:     MOVZWL  #1000,R4                ; 1 sec = 1000 ms
                      0227   30   0140   652 260$:     BSBW    CNX$WAIT                    ; Start timeout
                        05   0143   653          RSB
                                   0144   654
                50    8004   8F   3C   0144   655 270$:     MOVZWL  #<CLMDRS$M_DRS ! -      ; Disconnect status
                                   0149   656                  CLMDRS$C_PROTOCOL>,R0
                        05   11   0149   657          BRB     290$                        ; Branch to common code
                                   014B   658
                             3C   014B   659 280$:     MOVZWL  #< -                        ; Other node should withdraw
                                   014C   660                      CLMDRS$M_DRS ! -
                                   014C   661                      CLMDRS$C_REMOVED -
                50    800A   8F        014C   662                  >, R0
                      01DE   30   0150   663 290$:     BSBW    CNX$DISCONNECT              ; Break connection
                        05   0153   664          RSB
                                   0154   665
                      FF2E   30   0154   666 300$:     BSBW    10$                         ; Drop connection
                      03F5   30   0157   667          BSBW    DEAD_NODE                   ; Make node die
                             05   015A   668          RSB                                 ; Recurn
```

```
                     015B   670          .SBTTL  CNX$RCV_CNCT_MSG - Receive CONNECT message
                     015B   671
                     015B   672 ;++
                     015B   673 ;
                     015B   674 ; FUNCTIONAL DESCRIPTION:
                     015B   675 ;
                     015B   676 ;       This routine is called by SCS when a incoming CONNECT occurs
                     015B   677 ;       for us.  First the list of CSBs is scanned to see if we had a
                     015B   678 ;       previous connection to that system.  If we did then if the
                     015B   679 ;       software incarnation is the same we have to resend any messages
                     015B   680 ;       that haven't been received.  If the software incarnation changed,
                     015B   681 ;       then we have to do a failover.  If we don't have a CSB for that
                     015B   682 ;       system then one is created.
                     015B   683 ;
                     015B   684 ; CALLING SEQUENCE:
                     015B   685 ;
                     015B   686 ;       JSB     CNX$RCV_CNCT_MSG
                     015B   687 ;
                     015B   688 ; INPUT PARAMETERS:
                     015B   689 ;
                     015B   690 ;       R2      Address of connect request message
                     015B   691 ;       R3      Address of CDT
                     015B   692 ;       R4      Address of PDT
                     015B   693 ;
                     015B   694 ; IMPLICIT INPUTS:
                     015B   695 ;
                     015B   696 ;       None
                     015B   697 ;
                     015B   698 ; OUTPUT PARAMETERS:
                     015B   699 ;
                     015B   700 ;       None
                     015B   701 ;
                     015B   702 ; IMPLICIT OUTPUTS:
                     015B   703 ;
                     015B   704 ;       Completion codes returned to remote system if connection
                     015B   705 ;       is rejected:
                     015B   706 ;
                     015B   707 ;       SS$_REJECT      Connection rejected.   R1 is in CLMDRS format.
                     015B   708 ;       SS$_INSFMEM     Unable to allocate memory
                     015B   709 ;
                     015B   710 ; COMPLETION CODES:
                     015B   711 ;
                     015B   712 ;       None
                     015B   713 ;
                     015B   714 ; SIDE EFFECTS:
                     015B   715 ;
                     015B   716 ;       None
                     015B   717 ;
                     015B   718 ;--
                     015B   719
                     015B   720 CNX$RCV_CNCT_MSG:
51   1C A3   D0      015B   721 10$:    MOVL    CDT$L_PB(R3),R1         ; Get address of path block
51   30 A1   D0      015F   722         MOVL    PB$L_SBLINK(R1),R1      ; Get address of system block
      0430   30      0163   723         BSBW    CNX$LOOKUP_CSB          ; Find CSB
     1B 50   E9      0166   724         BLBC    R0,30$                  ; Branch on error
                     0169   725         STATE_DISP      <<NEW,200$>,<CONNECT,100$>,<RECONNECT,300$>,<WAIT,400$>>
                     0179   726         BUG_CHECK       CNXMGRERR,FATAL ; Unexpected connect received
```

```
                                017D   727 ;       MOVZWL  #<CLMDRS$M_DRS ! -      ; Protocol error
                                017D   728 ;               CLMDRS$C_PROTOCOL ! -
                                017D   729 ;               CLMDRS$M_FATAL>,R0
                                017D   730 ;       BRB     50$
                                017D   731
                 50   8006 8F   3C   017D   732 20$:    MOVZWL  #<CLMDRS$M_DRS ! -      ; Protocol version error
                                0182   733               CLMDRS$C_VERSION>,R0
                           0C   11   0182   734         BRB     50$
                                0184   735
                 50   8004 8F   3C   0184   736 30$:    MOVZWL  #<CLMDRS$M_DRS ! -      ; Nonfatal protocol error
                                0189   737               CLMDRS$C_PROTOCOL>,R0
                           05   11   0189   738         BRB     50$
                                018B   739
                 50   8104 8F   3C   018B   740 40$:    MOVZWL  #<CLMDRS$M_DRS ! -      ; Fatal protocol error
                                0190   741                 CLMDRS$M_FATAL ! -
                                0190   742               CLMDRS$C_PROTOCOL>,R0
                                0190   743 50$:    REJECT                          ; Reject the connection
                           05   0193   744         RSB
                                0194   745
                                0194   746 ;
                                0194   747 ; Connect request from system to which we are connecting
                                0194   748 ; Compare system ids and allow the system with the lower id to CONNECT
                                0194   749 ; and the one with the higher id to ACCEPT.
                                0194   750 ;
                           0C   BB   0194   751 100$:    PUSHR   #^M<R2,R3>              ; Save R2 and R3 from CMPC3
               52    68 A5   D0   0196   752         MOVL    CSB$L_SB(R5),R2         ; Address of System Block
00000018'GF    18 A2    06   29   019A   753         CMPC3   #SB$S_SYSTEMID, -       ; Compare system ids
                                01A3   754               SB$B_SYSTEMID(R2), -
                                01A3   755               G^SCS$GA_LOCALSB+SB$B_SYSTEMID
                           0C   BA   01A3   756         POPR    #^M<R2,R3>              ; Restore R2 and R3
                           DD   1E   01A5   757         BGEQU   30$                     ; Remote is higher - reject
                                01A7   758 ;
                                01A7   759 ; Connect request -- expected to be from a new system.
                                01A7   760 ; If the connect request is not from a newly booting system, we assume
                                01A7   761 ; that the local system has booted twice with the same incarnation
                                01A7   762 ; number.  In this case, we reject the connection and hope that the other
                                01A7   763 ; node will eventually see our attempts to connect and realize that this
                                01A7   764 ; node has duplicated its incarnation number.
                                01A7   765 ;
             22 A2    91   01A7   766 200$:    CMPB    SCSCMG$B_SNDDAT+CNCT$B_TYPE(R2).- ; Verify remote side is doing
                01    01AA   767               #CNCT$K_INITIAL         ; a new CONNECT
                D7    12   01AB   768         BNEQ    30$             ; It's not - don't allow a connect
                           01AD   769         ASSUME  CNCT$B_VERNUM  EQ  CNCT$B_ECOLVL+1
                           01AD   770         ASSUME  CSB$B_VERNUM  EQ CSB$B_ECOLVL+1
             20 A2    B0   01AD   771         MOVW    SCSCMG$B_SNDDAT+CNCT$B_ECOLVL(R2),- ; Store remote side's
             40 A5         01B0   772               CSB$B_ECOLVL(R5)        ; protocol version number and ECO level
             23 A2    90   01B2   773         MOVB    SCSCMG$B_SNDDAT+CNCT$B_ACKLIM(R2),- ; Store remote side's
             33 A5         01B5   774               CSB$B_REMACKLIM(R5)     ; ACK limit
              0270    30   01B7   775         BSBW    CNCT_CHECK              ; Check connect data
             CE 50    E9   01BA   776         BLBC    R0,40$                  ; Branch to bugcheck remote node
                06    90   01BD   777         MOVB    #CSB$K_ACCEPT,-         ; Set state
             43 A5         01BF   778               CSB$B_STATE(R5)
                5F    10   01C1   779         BSBB    CNX$ACCEPT              ; Accept connection
                      05   01C3   780         RSB
                                01C4   781
                                01C4   782 ;
                                01C4   783 ; Reconnect request from a system to which we are reconnecting
```

```
                                01C4    784 ;
                          0C    BB  01C4    785 300$:   PUSHR   #^M<R2,R3>                ; Save R2 and R3 from CMPC3
                    52    68 A5   D0  01C6    786         MOVL    CSB$L_SB(R5),R2         ; Address of System Block
          00000018'GF     18 A2   06  29  01CA    787         CMPC3   #SB$S_SYSTEMID, -       ; Compare system ids
                                01D3    788                 SB$B_SYSTEMID(R2), -
                                01D3    789                 G^SCS$GA_LOCALSB+SB$B_SYSTEMID
                          0C    BA  01D3    790         POPR    #^M<R2,R3>              ; Restore R2 and R3
                          AD    1E  01D5    791         BGEQU   30$                     ; Remote is higher - reject
                                01D7    792 ;
                                01D7    793 ; Connect request from a system that we are timing out
                                01D7    794 ;
                       22 A2    91  01D7    795 400$:   CMPB    SCSCMG$B_SNDDAT+CNCT$B_TYPE(R2),- ; Verify remote side is doing
                          02        01DA    796                 #CNCT$K_RECONNECT       ; a RECONNECT
                          35    12  01DB    797         BNEQ    430$                    ; It's not -- handle special situation
                        0298    30  01DD    798         BSBW    RECNCT_CHECK            ; Check data following reconnect
                9F 60 A5    00    E0  01E0    799         BBS     #CSB$V_LONG_BREAK, -    ; Reject if long break has been
                                01E5    800                 CSB$L_STATUS(R5),30$
                                01E5    801         ASSUME  CNCT$B_VERNUM EQ CNCT$B_ECOLVL+1
                                01E5    802         ASSUME  CSB$B_VERNUM EQ CSB$B_ECOLVL+1
                       20 A2    B1  01E5    803         CMPW    SCSCMG$B_SNDDAT+CNCT$B_ECOLVL(R2),- ; Are we speaking the
                          40 A5     01E8    804                 CSB$B_ECOLVL(R5)        ; same protocol as before?
                          91    12  01EA    805         BNEQ    20$                     ; No -- fatal error
                       23 A2    90  01EC    806         MOVB    SCSCMG$B_SNDDAT+CNCT$B_ACKLIM(R2),- ; Store remote side's
                          33 A5     01EF    807                 CSB$B_REMACKLIM(R5)     ; ACK limit
                          08    90  01F1    808         MOVB    #CSB$K_REACCEPT,-       ; Set state
                          43 A5     01F3    809                 CSB$B_STATE(R5)
                       2C A2    B0  01F5    810         MOVW    SCSCMG$B_SNDDAT+CNCT$W_RCVDSEQNM(R2),- ; Get last received
                          50        01F8    811                 R0                      ; sequence number (of ours)
                       30 A5    A3  01F9    812         SUBW3   CSB$W_ACKRSEQNM(R5),-   ; Verify it's greater than or equal to
                    51    50        01FC    813                 R0,R1                   ; the last one
                          04    18  01FE    814         BGEQ    420$                    ; It is
                                0200    815         BUG_CHECK       CNXMGRERR,FATAL ; *** Sequence number error
                                0204    816
                30 A5     50    B0  0204    817 420$:   MOVW    R0,CSB$W_ACKRSEQNM(R5)  ; It's ok - store it
                        032D    30  0208    818         BSBW    DELETE_TQE              ; Clean out TQE
                7E A5     02    90  020B    819         MOVB    #CNCT$K_RECONNECT, -    ; Flag this as a reconnect
                                02CF    820                 CSB$B_CNCT+CNCT$B_TYPE(R5)
                          11    10  020F    821         BSBB    CNX$ACCEPT              ; Accept connection
                                05  0211    822         RSB
                                0212    823
                                0212    824 ;
                                0212    825 ; Get here is this node expected a re-connection and instead received
                                0212    826 ; an initial connect request.  This happens if the remote node has rebooted
                                0212    827 ; with a duplicated software incarnation number.  To recover from this
                                0212    828 ; situation, the software incarnation number in the CSB is modified and
                                0212    829 ; this routine is re-entered.  The old CSB will be marked "DEAD".
                                0212    830 ; A new CSB will be formed and the connect request accepted.
                                0212    831 ;
                        0323    30  0212    832 430$:   BSBW    DELETE_TQE              ; Clean out TQE
                50    38 A5     7E  0215    833         MOVAQ   CSB$Q_SWINCARN(R5),R0   ; Address of stored software incarnation num
                   80    60     D2  0219    834         MCOML   (R0),(R0)+              ; Invalidate software incarnation number
                   60    60     D2  021C    835         MCOML   (R0),(R0)               ; so that a new CSB will be formed
                      FF39    31  021F    836         BRW     10$                     ; Re-enter this routine
```

```
                        0222   838          .SBTTL  CNX$ACCEPT - Accept connection from remote system
                        0222   839
                        0222   840 ;++
                        0222   841 ;
                        0222   842 ; FUNCTIONAL DESCRIPTION:
                        0222   843 ;
                        0222   844 ;       This routine is called to accept a connection from the
                        0222   845 ;       connection manager on a remote system.
                        0222   846 ;
                        0222   847 ; CALLING SEQUENCE:
                        0222   848 ;
                        0222   849 ;       JSB     CNX$ACCEPT
                        0222   850 ;       IPL must be at IPL$_SYNCH
                        0222   851 ;
                        0222   852 ; INPUT PARAMETERS:
                        0222   853 ;
                        0222   854 ;       R2 is address of connection message
                        0222   855 ;       R3 is address of CDT
                        0222   856 ;       R4 is address of PDT
                        0222   857 ;       R5 is address of initialized CSB
                        0222   858 ;
                        0222   859 ; OUTPUT PARAMETERS:
                        0222   860 ;
                        0222   861 ;       NONE
                        0222   862 ;
                        0222   863 ; COMPLETION CODES:
                        0222   864 ;
                        0222   865 ;       NONE
                        0222   866 ;
                        0222   867 ; SIDE EFFECTS:
                        0222   868 ;
                        0222   869 ;       R0-R5 are destroyed
                        0222   870 ;
                        0222   871 ;--
                        0222   872
                        0222   873 CNX$ACCEPT::
                        0222   874 ;
                        0222   875 ; Try to accept
                        0222   876 ; This thread may be suspended here
                        0222   877 ;
 01DA   30              0222   878          BSBW    CNCT_DATA                       ; Setup connect data
                        0225   879          ACCEPT  MSGADR = CNX$RCV_MSG,-
                        0225   880                  ERRADR = CNX$ERROR,-
                        0225   881                  INITCR = #SEND_CREDITS,-
                        0225   882                  CONDAT = CSB$B_CNCT(R5),-
                        0225   883                  AUXSTR = (R5)
06 50   E8            024B   884          BLBS    R0,10$                          ; Branch on success
   3F   BB            024E   885          PUSHR   #^M<R0,R1,R2,R3,R4,R5>          ; Save registers
   10   10            0250   886          BSBB    20$                             ; Clean up failed success
   3F   BA            0252   887          POPR    #^M<R0,R1,R2,R3,R4,R5>          ; Restore registers
                        0254   888 10$:
                        0254   889          STATE_DISP      <<ACCEPT,100$>,<REACCEPT,200$>>
                        025E   890          BUG_CHECK       CNXMGRERR,FATAL ; Bugcheck
                        0262   891
                        0262   892 ;
                        0262   893 ; Accept attempt failed.
                        0262   894 ; Clean up by rejecting a connection based on the listening CDT.
```

```
                        0262   895 ;
        53   52   D0     0262   896 20$:    MOVL    R2,R3                   ; Address of listening CDT
                        0265   897          REJECT                          ; Reject the request
                     05 0268   898          RSB                             ; Terminate the thread
                        0269   899
                        0269   900 ;
                        0269   901 ; Initial accept attempt completed
                        0269   902 ;
                        0269   903 ;       R3 is address of CDT
                        0269   904 ;       R4 is address of PDT
                        0269   905 ;       R5 is address of CSB
                        0269   906 ;
          2F 50   E9    0269   907 100$:    BLBC    R0,150$                 ; ACCEPT error
                        026C   908          ASSUME  CSB$L_PDT  EQ  CSB$L_CDT+4
    0C A5    53   7D    026C   909          MOVQ    R3,CSB$L_CDT(R5)        ; Store CDT and PDT address
    52   68 A5   D0    0270   910          MOVL    CSB$L_SB(R5),R2         ; Address of System Block
    52   18 A2   9E    0274   911          MOVAB   SB$B_SYSTEMID(R2),R2    ; Address of destination system ID
    51   64 A5   D0    0278   912          MOVL    CSB$L_CLUB(R5),R1       ; Address of CLUB
    51   0C A1   D0    027C   913          MOVL    CLUB$L_POLL_CTX(R1),R1  ; Address of SPPB
              50   D4    0280   914          CLRL    R0                      ; Disable polling
  00000000'GF   16    0282   915          JSB     G^SCS$POLL_MODE         ; Disable polling this system
              01   90    0288   916          MOVB    #CSB$K_OPEN,-           ; Mark connection open
          43 A5         028A   917                  CSB$B_STATE(R5)
    50  0000'CF   9E    028C   918          MOVAB   ACCPT_MSG,R0            ; Address of accept message
           FD6C'  30    0291   919          BSBW    CNX$CONFIG_CHANGE       ; Note configuration change
           FD69'  30    0294   920          BSBW    CNX$RESEND_MSGS         ; Initialize for sending messages
           FD66'  30    0297   921          BSBW    CNX$CON_NEWSYS          ; Inform configuration manager
                     05 029A   922          RSB
                        029B   923
           026A   30    029B   924 150$:    BSBW    CNX$DECREFCNT           ; Deallocate CSB
                     05 029E   925          RSB
                        029F   926
                        029F   927 ;
                        029F   928 ; Reaccept attempt completed
                        029F   929 ;
                        029F   930 ;       R3 is address of CDT
                        029F   931 ;       R4 is address of PDT
                        029F   932 ;       R5 is address of CSB
                        029F   933 ;
          1C 50   E9    029F   934 200$:    BLBC    R0,210$                 ; Branch on failure
                        02A2   935          ASSUME  CSB$L_PDT  EQ  CSB$L_CDT+4
    0C A5    53   7D    02A2   936          MOVQ    R3,CSB$L_CDT(R5)        ; Store CDT and PDT address
1C 60 A5    00   E0    02A6   937          BBS     #CSB$V_LONG_BREAK,-     ; Branch if long break in connection
                        02AB   938                  CSB$L_STATUS(R5),220$
              01   90    02AB   939          MOVB    #CSB$K_OPEN,-           ; Mark connection open
          43 A5         02AD   940                  CSB$B_STATE(R5)
           FD4E'  30    02AF   941          BSBW    CNX$CHECK_QUORUM        ; Resume activity on if quorum
    50  0000'CF   9E    02B2   942          MOVAB   REACCPT_MSG,R0          ; Address of reaccept message
           FD46'  30    02B7   943          BSBW    CNX$CONFIG_CHANGE       ; Note configuration change
           FD43'  30    02BA   944          BSBW    CNX$RESEND_MSGS         ; Resend outstanding messages
                     05 02BD   945          RSB
                        02BE   946
    54  03E8 8F   3C    02BE   947 210$:    MOVZWL  #1000,R4                ; Wait 1 sec = 1000 ms
           00A4   30    02C3   948          BSBW    CNX$WAIT                ; Enter wait state
                     05 02C6   949          RSB
                        02C7   950
    50  800A 8F   3C    02C7   951 220$:    MOVZWL  #<CLMDRS$M_DRS ! -
```

```
                02CC   952                      CLMDRS$C_REMOVED>,R0
63      10      02CC   953              BSBB    CNX$DISCONNECT          ; Break connection
        05      02CE   954              RSB
```

```
                        956          .SBTTL  CNX$DISC_BUGCHECK - Disconnect from Node and Request it to Bugcheck
                        957          .SBTTL  CNX$DISC_REMOVE - Disconnect from Node Removed from Cluster
                        958          .SBTTL  CNX$DISC_PROTOCOL - Disconnect from Node for protcol reasons
                        959 ;++
                        960 ;
                        961 ; FUNCTIONAL DESCRIPTION:
                        962 ;
                        963 ;       These routine are called to break a connection with a remote node and
                        964 ;       return some informational status.
                        965 ;
                        966 ;       CNX$DISC_BUGCHECK: Request the remote node to bugcheck
                        967 ;       CNX$DISK_REMOVE: Status indicates that the node was removed from the
                        968 ;                        cluster
                        969 ;       CNX$DISC_PROTOCOL: Disconnect for protocol reasons, reconnect as convenient
                        970 ;
                        971 ; CALLING SEQUENCE:
                        972 ;
                        973 ;       JSB     CNX$DISC_BUGCHECK
                        974 ;       JSB     CNX$DISC_REMOVE
                        975 ;       JSB     CNX$DISC_PROTOCOL
                        976 ;       IPL is at SCS fork level
                        977 ;
                        978 ; INPUT PARAMETERS:
                        979 ;
                        980 ;       R5:     Address of CSB of removed node
                        981 ;
                        982 ; IMPLICIT INPUTS:
                        983 ;
                        984 ;       None
                        985 ;
                        986 ; OUTPUT PARAMETERS:
                        987 ;
                        988 ;       None
                        989 ;
                        990 ; IMPLICIT OUTPUTS:
                        991 ;
                        992 ;       None
                        993 ;
                        994 ; SIDE EFFECTS:
                        995 ;
                        996 ;       R0-R1 destroyed
                        997 ;
                        998 ;--
                        999
                       1000          .ENABLE LSB
                       1001
                       1002 CNX$DISC_BUGCHECK::
50   8104 8F   3C       1003          MOVZWL  #CLMDRS$M_DRS ! -       ; Disconnect status
                       1004                  CLMDRS$M_FATAL ! -      ; Fatal
                       1005                  CLMDRS$C_PROTOCOL, R0
      05   11           1006          BRB     10$                     ; Join common code
                       1007
                       1008 CNX$DISC_REMOVE::
50   800A 8F   3C       1009          MOVZWL  #CLMDRS$M_DRS ! -       ; Disconnect status
                       1010                  CLMDRS$C_REMOVED,R0
      09   10           1011 10$:     BSBB    DISC_STATUS             ; Disconnect using status in R0
    020E   30           1012          BSBW    LONG_BREAK              ; Mark a long break (may already be done)
```

```
                 05  02E0  1013          RSB                                 ; Return to caller
                     02E1  1014
                     02E1  1015 CNX$DISC_PROTOCOL::
50   8004 8F     3C  02E1  1016          MOVZWL  #CLMDRS$M_DRS ! -           ; Disconnect status
                     02E6  1017                  CLMDRS$C_PROTOCOL, R0
                     02E6  1018 DISC_STATUS:                                 ; Disconnect status in R0
                 3C  BB  02E6  1019          PUSHR   #^M<R2,R3,R4,R5>        ; Save non-volatile registers
                     02E8  1020          STATE_DISP      <<OPEN,100$>>       ; Connection is currently open
                 02  11  02EF  1021          BRB     190$
                     02F1  1022
                 16  10  02F1  1023 100$:   BSBB    CNX$BREAK                 ; Break connection, status in R0
                 3C  BA  02F3  1024 190$:   POPR    #^M<R2,R3,R4,R5>          ; Restore registers
                 05  02F5  1025          RSB
                     02F6  1026
                     02F6  1027          .DISABLE        LSB
```

```
                        02F6  1029          .SBTTL  CNX$ERROR - Connection error
                        02F6  1030
                        02F6  1031 ;++
                        02F6  1032 ;
                        02F6  1033 ; FUNCTIONAL DESCRIPTION:
                        02F6  1034 ;
                        02F6  1035 ;       This routine is called by SCS when a connection breaks.
                        02F6  1036 ;       This routine calls CNX$CLEANUP to deal with outstanding
                        02F6  1037 ;       messages and then does a DISCONNECT.  A timeout is then
                        02F6  1038 ;       requested at the conclusion of which the connection will
                        02F6  1039 ;       be reattempted.
                        02F6  1040 ;
                        02F6  1041 ; CALLING SEQUENCE:
                        02F6  1042 ;
                        02F6  1043 ;       JSB     CNX$ERROR (Called from SCS)
                        02F6  1044 ;       IPL is at SCS fork level (8)
                        02F6  1045 ;
                        02F6  1046 ; INPUT PARAMETERS:
                        02F6  1047 ;
                        02F6  1048 ;       R0      Contains error status (SS$_DISCONNECT or SS$_VCBROKEN)
                        02F6  1049 ;       R1      Additional status (disconnect reason or virtual circuit
                        02F6  1050 ;               broken reason)
                        02F6  1051 ;       R3      Address of CDT
                        02F6  1052 ;       R4      Address of PDT
                        02F6  1053 ;
                        02F6  1054 ; IMPLICIT INPUTS:
                        02F6  1055 ;
                        02F6  1056 ;       None
                        02F6  1057 ;
                        02F6  1058 ; OUTPUT PARAMETERS:
                        02F6  1059 ;
                        02F6  1060 ;       None
                        02F6  1061 ;
                        02F6  1062 ; IMPLICIT OUTPUTS:
                        02F6  1063 ;
                        02F6  1064 ;       None
                        02F6  1065 ;
                        02F6  1066 ; SIDE EFFECTS:
                        02F6  1067 ;
                        02F6  1068 ;       R0-R5 destroyed
                        02F6  1069 ;
                        02F6  1070 ;--
                        02F6  1071
                        02F6  1072 CNX$ERROR::
  55    5C A3    D0     02F6  1073          MOVL    CDT$L_AUXSTRUC(R3),R5   ; CSB address
          23    BB     02FA  1074          PUSHR   #^M<R0,R1,R5>           ; Save registers
50    8004 8F    3C     02FC  1075          MOVZWL  #CLMDRS$M_DRS ! -       ; Disconnect status
                        0301  1076                  CLMDRS$C_PROTOCOL,R0
          06    10     0301  1077          BSBB    CNX$BREAR               ; Use common disconnect code
          23    BA     0303  1078          POPR    #^M<R0,R1,R5>           ; Restore registers
        024E    30     0305  1079          BSBW    CNX_STATUS_CHECK        ; Check for bugcheck request
                05     0308  1080          RSB
```

```
                                0309  1082          .SBTTL  CNX$BREAK - Cleanup and Disconnect SCS Connection
                                0309  1083
                                0309  1084 ;++
                                0309  1085 ;
                                0309  1086 ; FUNCTIONAL DESCRIPTION:
                                0309  1087 ;
                                0309  1088 ;       This routine is called when a connection breaks or when a
                                0309  1089 ;       connection is to be broken.  It calls CNX$CLEANUP to deal
                                0309  1090 ;       with outstanding messages and then does a DISCONNECT.
                                0309  1091 ;       A timeout is then requested at the conclusion of which the
                                0309  1092 ;       connection will be reattempted.
                                0309  1093 ;
                                0309  1094 ; CALLING SEQUENCE:
                                0309  1095 ;
                                0309  1096 ;       JSB     CNX$BREAK
                                0309  1097 ;       IPL is at IPL$_SYNCH = IPL$_SCS
                                0309  1098 ;
                                0309  1099 ; INPUT PARAMETERS:
                                0309  1100 ;
                                0309  1101 ;       R0      Contains disconnect code
                                0309  1102 ;       R5      Address of CSB
                                0309  1103 ;
                                0309  1104 ; IMPLICIT INPUTS:
                                0309  1105 ;
                                0309  1106 ;       None
                                0309  1107 ;
                                0309  1108 ; OUTPUT PARAMETERS:
                                0309  1109 ;
                                0309  1110 ;       None
                                0309  1111 ;
                                0309  1112 ; IMPLICIT OUTPUTS:
                                0309  1113 ;
                                0309  1114 ;       None
                                0309  1115 ;
                                0309  1116 ; SIDE EFFECTS:
                                0309  1117 ;
                                0309  1118 ;       R0-R4 Destroyed.
                                0309  1119 ;
                                0309  1120 ;--
                                0309  1121
                                0309  1122 CNX$BREAK::
                     50   DD    0309  1123          PUSHL   R0                      ; Save disconnect status
           18 60 A5   00   E0    030B  1124          BBS     #CSB$V_LONG_BREAK, -    ; Branch if long break
                                0310  1125                  CSB$L_STATUS(R5),20$
      50   00000000'GF   3C    0310  1126          MOVZWL  G^CLU$GW_RECNXINT,R0    ; Max. retry in seconds
48 A5 00000000'GF   50   C1    0317  1127          ADDL3   R0,G^EXE$GL_ABSTIM, -   ; Time at which to stop retries
                                0320  1128                  CSB$L_TIMEOUT(R5)
      50   0000'CF   9E    0320  1129          MOVAB   CNXERROR_MSG,R0         ; Address of message
                FCD8'   30    0325  1130          BSBW    CNX$CONFIG_CHANGE       ; Note configuration change
                                0328  1131 20$:
                     01   BA    0328  1132          POPR    #^M<R0>                 ; Restore disconnect status
                                032A  1133          ASSUME  CSB$L_PDT  EQ CSB$L_CDT+4
           53   0C A5   7D    032A  1134          MOVQ    CSB$L_CDT(R5),R3        ; Fetch CDT and PDT addresses
                     01   10    032E  1135          BSBB    CNX$DISCONNECT          ; Disconnect, status in R0
                          05    0330  1136          RSB
```

```
                        0331  1138          .SBTTL  CNX$DISCONNECT - Disconnect from remote system
                        0331  1139
                        0331  1140 ;++
                        0331  1141 ;
                        0331  1142 ; FUNCTIONAL DESCRIPTION:
                        0331  1143 ;
                        0331  1144 ;       This routine is called to disconnect from the
                        0331  1145 ;       connection manager on a remote system.
                        0331  1146 ;
                        0331  1147 ; CALLING SEQUENCE:
                        0331  1148 ;
                        0331  1149 ;       JSB     CNX$DISCONNECT
                        0331  1150 ;       IPL must be at IPL$_SYNCH
                        0331  1151 ;
                        0331  1152 ; INPUT PARAMETERS:
                        0331  1153 ;
                        0331  1154 ;       R0 is disconnect status
                        0331  1155 ;       R5 is address of initialized CSB
                        0331  1156 ;
                        0331  1157 ; OUTPUT PARAMETERS:
                        0331  1158 ;
                        0331  1159 ;       NONE
                        0331  1160 ;
                        0331  1161 ; COMPLETION CODES:
                        0331  1162 ;
                        0331  1163 ;       NONE
                        0331  1164 ;
                        0331  1165 ; SIDE EFFECTS:
                        0331  1166 ;
                        0331  1167 ;       R0-R5 are destroyed
                        0331  1168 ;
                        0331  1169 ;--
                        0331  1170
                        0331  1171 CNX$DISCONNECT::
 43 A5   07   90        0331  1172          MOVB    #CSB$K_DISCONNECT, -    ; Set disconnect state
                        0335  1173                  CSB$B_STATE(R5)
      50   DD           0335  1174          PUSHL   R0                      ; Save status
    FCC6'  30           0337  1175          BSBW    CNX$CHECK_QUORUM        ; Block activity if quorum lost
    FCC3'  30           033A  1176          BSBW    CNX$PRE_CLEANUP         ; Cleanup outstanding messages
      01   BA           033D  1177          POPR    #^M<R0>                 ; Restore status
                        033F  1178 ;
                        033F  1179 ; Try to disconnect
                        033F  1180 ; This thread may be suspended here
                        033F  1181 ;
                        033F  1182          ASSUME  CSB$L_PDT  EQ CSB$L_CDT+4
 53   0C A5   7D        033F  1183          MOVQ    CSB$L_CDT(R5),R3        ; Fetch CDT and PDT addresses
                        0343  1184          DISCONNECT                      ; Status in R0, always succeeds
                        0349  1185          STATE_DISP      <<DISCONNECT,100$>>
                        0350  1186          BUG_CHECK       CNXMGRERR,FATAL ; Invalid state
                        0354  1187
                        0354  1188 ;
                        0354  1189 ; Soft disconnect attempt completed
                        0354  1190 ;
    FCA9'  30           0354  1191 100$:    BSBW    CNX$POST_CLEANUP        ; Finish cleanup of outstanding messages
                        0357  1192          ASSUME  CSB$L_PDT  EQ CSB$L_CDT+4
   0C A5   7C           0357  1193          CLRQ    CSB$L_CDT(R5)           ; Clear CDT and PDT address in CSB
03 60 A5   00   E1      035A  1194          BBC     #CSB$V_LONG_BREAK, -    ; Branch if no long break yet
```

```
                 035F  1195                    CSB$L_STATUS(R5),110$
          FC9E'  30    035F  1196              BSBW    CNX$FAIL_MSG            ; Fail outstanding messages
54  01F4  8F     3C    0362  1197  110$:       MOVZWL  #500,R4                 ; Delay 500 milli-seconds
        01       10    0367  1198              BSBB    CNX$WAIT
                 05    0369  1199              RSB
```

```
                        036A  1201          .SBTTL  CNX$WAIT - Initiate timeout
                        036A  1202
                        036A  1203 ;++
                        036A  1204 ;
                        036A  1205 ; FUNCTIONAL DESCRIPTION:
                        036A  1206 ;
                        036A  1207 ;       This routine is called to begin a timeout before trying
                        036A  1208 ;       to reconnect to the connection manager on a remote system.
                        036A  1209 ;
                        036A  1210 ; CALLING SEQUENCE:
                        036A  1211 ;
                        036A  1212 ;       JSB     CNX$WAIT
                        036A  1213 ;       IPL must be at IPL$_SYNCH
                        036A  1214 ;
                        036A  1215 ; INPUT PARAMETERS:
                        036A  1216 ;
                        036A  1217 ;       R4 is the timeout period in milli-seconds
                        036A  1218 ;       R5 is address of initialized CSB
                        036A  1219 ;
                        036A  1220 ; OUTPUT PARAMETERS:
                        036A  1221 ;
                        036A  1222 ;       NONE
                        036A  1223 ;
                        036A  1224 ; COMPLETION CODES:
                        036A  1225 ;
                        036A  1226 ;       NONE
                        036A  1227 ;
                        036A  1228 ; SIDE EFFECTS:
                        036A  1229 ;
                        036A  1230 ;       R0-R4 are destroyed
                        036A  1231 ;
                        036A  1232 ;--
                        036A  1233
                        036A  1234 CNX$WAIT::
                    09  90  036A  1235          MOVB    #CSB$K_WAIT,-           ; Set connection state WAITing
                43  A5        036C  1236                  CSB$B_STATE(R5)
                51  30  3C  036E  1237          MOVZWL  #TQE$K_LENGTH,R1        ; Size of timer queue entry
        00000000'GF  16  0371  1238          JSB     G^EXE$ALONONPAGED       ; Allocate one
                60  50  E9  0377  1239          BLBC    R0,RETRY_CONNECT        ; No memory, so forget timeout
                55  DD  037A  1240          PUSHL   R5                      ; Save CSB address
            08  A2  51  B0  037C  1241          MOVW    R1,TQE$W_SIZE(R2)       ; Store size
            0A  A2  0F  90  0380  1242          MOVB    #DYN$C_TQE,TQE$B_TYPE(R2) ; Store type
            10  A2  55  D0  0384  1243          MOVL    R5,TQE$L_FR3(R2)        ; Store CSB address as fork reg. R3
              14  A2  D4  0388  1244          CLRL    TQE$L_FR4(R2)           ; Save zero as fork reg. R4
            44  A5  52  D0  038B  1245          MOVL    R2,CSB$L_TQE(R5)        ; Save TQE address in CSB
                55  52  D0  038F  1246          MOVL    R2,R5                   ; Move address of TQE
            0B  A5  01  90  0392  1247          MOVB    #TQE$C_SSSNGL,TQE$B_RQTYPE(R5) ; Store type of timer queue entry
        0C  A5    BB'AF  9E  0396  1248          MOVAB   B^TIMEOUT,TQE$L_FPC(R5) ; Store address of timer fork process
53  00  00002710  8F  54  7A  039B  1249          EMUL    R4,#10*1000,#0,R3       ; Get milli-seconds and cvt to 100 ns. units
    50  00000000'GF  7D  03A4  1250          MOVQ    G^EXE$GQ_SYSTIME,R0     ; Get current time
                50  53  C0  03AB  1251          ADDL    R3,R0                   ; Add to current time
                51  54  D8  03AE  1252          ADWC    R4,R1
        00000000'GF  16  03B1  1253          JSB     G^EXE$INSTIMQ           ; Insert in timer queue
                55  8ED0  03B7  1254          POPL    R5                      ; Restore CSB address
                    05  03BA  1255          RSB
                        03BB  1256
                        03BB  1257 ;++
```

```
                        03BB  1258 ; Come here as a timer fork process to retry the CONNECT
                        03BB  1259 ; Inputs:
                        03BB  1260 ;       R3      CSB address
                        03BB  1261 ;       R5      TQE address
                        03BB  1262 ;
                        03BB  1263 ;--
                        03BB  1264
                        03BB  1265 TIMEOUT:
          50    55  D0    03BB  1266         MOVL    R5,R0                   ; Address of timer queue entry
          55    53  D0    03BE  1267         MOVL    R3,R5                   ; Address of CSB
  00000000'GF     16    03C1  1268         JSB     G^EXE$DEANONPAGED       ; Deallocate it
       44 A5      D4    03C7  1269         CLRL    CSB$L_TQE(R5)           ; Clear pointer to TQE
       43 A5      91    03CA  1270         CMPB    CSB$B_STATE(R5),-       ; Are we in wait state?
                  09    03CD  1271                 #CSB$K_WAIT
                  12    03CE  1272         BNEQ    10$                     ; No, just return
                  10    03D0  1273         BSBB    RETRY_CONNECT           ; Do the rest in a subroutine so that
                        03D2  1274                                         ; CONNECT can return here
  55  00000000'GF     DE    03D2  1275 10$:    MOVAL   G^EXE$AL_TQENOREPT,R5   ; Use non-repeating timer queue entry
                  05    03D9  1276         RSB
                        03DA  1277
                        03DA  1278 ;
                        03DA  1279 ; Must check for change to remote system that may have occurred during
                        03DA  1280 ; the timeout.  Unlike other situations, there is no connection to break
                        03DA  1281 ; to give notification of such an event.
                        03DA  1282 ;
                        03DA  1283 RETRY_CONNECT:
       51   68 A5      D0    03DA  1284         MOVL    CSB$L_SB(R5),R1         ; Address of System Block
             01B5      30    03DE  1285         BSBW    CNX$LOOKUP_CSB          ; Find or allocate a CSB and
                        03E1  1286                                             ;  as a side effect, handle old CSB
            0E 50      E9    03E1  1287         BLBC    R0,20$                  ; Can't allocate CSB
                        03E4  1288         STATE_DISP      <<NEW,CNX$NEW_CSB>,<WAIT,100$>>
                        03EE  1289         BUG_CHECK       CNXMGRERR,FATAL
                        03F2  1290
                  05    03F2  1291 20$:    RSB
                        03F3  1292
          03    90    03F3  1293 100$:   MOVB    #CSB$K_RECONNECT,-      ; Change to RECONNECT state
       43 A5          03F5  1294                 CSB$B_STATE(R5)
  7E A5     02    90    03F7  1295         MOVB    #CNCT$K_RECONNECT, -    ; Flag this as a reconnect
                        03FB  1296                 CSB$B_CNCT+CNCT$B_TYPE(R5)
             FC3A   30    03FB  1297         BSBW    CNX$CONNECT             ; Request connection
                  05    03FE  1298         RSB                             ; Unable to allocate memory
```

```
                                03FF  1300          .SBTTL  CNCT_DATA - Setup Connect Data in CSB
                                03FF  1301 ;++
                                03FF  1302 ; FUNCTIONAL DESCRIPTION:
                                03FF  1303 ;
                                03FF  1304 ;       Set up the CSB$B_CNCT area in preparation for requesting
                                03FF  1305 ;       or accepting a connection.
                                03FF  1306 ;
                                03FF  1307 ; CALLING SEQUENCE:
                                03FF  1308 ;
                                03FF  1309 ;       BSBW    CNCT_DATA
                                03FF  1310 ;       IPL must be at IPL$_SCS
                                03FF  1311 ;
                                03FF  1312 ; INPUT PARAMETERS:
                                03FF  1313 ;
                                03FF  1314 ;       R5      Address of CSB
                                03FF  1315 ;
                                03FF  1316 ; OUTPUT PARAMETERS:
                                03FF  1317 ;
                                03FF  1318 ;       None
                                03FF  1319 ;
                                03FF  1320 ; SIDE EFFECTS:
                                03FF  1321 ;
                                03FF  1322 ;       R0 and R1 are destroyed.
                                03FF  1323 ;--
                                03FF  1324
                                03FF  1325 CNCT_DATA:
     50    7C A5    9E          03FF  1326          MOVAB   CSB$B_CNCT(R5),R0       ; Point to connect data area
     51    64 A5    D0          0403  1327          MOVL    CSB$L_CLUB(R5),R1       ; Address of CLUB
04 A0    20 A1    B0          0407  1328          MOVW    CLUB$W_QUORUM(R1), -    ; Cluster quorum
                                040C  1329                  CNCT$W_QUORUM(R0)
06 A0    22 A1    B0          040C  1330          MOVW    CLUB$W_VOTES(R1), -     ; Cluster votes
                                0411  1331                  CNCT$W_VOTES(R0)
08 A0    24 A1    B0          0411  1332          MOVW    CLUB$W_NODES(R1), -     ; Cluster nodes
                                0416  1333                  CNCT$W_NODES(R0)
                                0416  1334          ASSUME  CSB$M_LONG_BREAK EQ CNCT$M_LONG_BREAK
                                0416  1335          ASSUME  CSB$M_MEMBER EQ CNCT$M_MEMBER
                                0416  1336          ASSUME  CSB$M_REMOVED EQ CNCT$M_REMOVED
0B A0    60 A5    F8 8F    8B  0416  1337          BICB3   #^C<CSB$M_LONG_BREAK ! -
                                041D  1338                      CSB$M_MEMBER ! -
                                041D  1339                      CSB$M_REMOVED>, -
                                041D  1340                  CSB$L_STATUS(R5), -     ; Fill in status bits from CSB
                                041D  1341                  CNCT$B_CNXSTS(R0)
                                041D  1342          ASSUME  CLUB$M_CLUSTER EQ CNCT$M_CLUSTER
0A A0    1C A1    FE 8F    8B  041D  1343          BICB3   #^C<CLUB$M_CLUSTER>, -
                                0424  1344                  CLUB$L_FLAGS(R1), -     ; Fill in status bits from CLUB
                                0424  1345                  CNCT$B_CLSSTS(R0)
     0C A0    2E A5    B0      0424  1346          MOVW    CSB$W_RCVDSEQNM(R5), -  ; Last message received
                                0429  1347                  CNCT$W_RCVDSEQNM(R0)
                          05    0429  1348          RSB
```

```
                     042A  1350          .SBTTL  CNCT_CHECK - Verify Connect Data
                     042A  1351 ;++
                     042A  1352 ; FUNCTIONAL DESCRIPTION:
                     042A  1353 ;
                     042A  1354 ;       Evaluate received connect data vs. connect data
                     042A  1355 ;       sent or about to be sent.
                     042A  1356 ;
                     042A  1357 ; CALLING SEQUENCE:
                     042A  1358 ;
                     042A  1359 ;       BSBW    CNCT_CHECK
                     042A  1360 ;       IPL must be at IPL$_SCS
                     042A  1361 ;
                     042A  1362 ; INPUT PARAMETERS:
                     042A  1363 ;
                     042A  1364 ;       R2      Address of received connect data message
                     042A  1365 ;       R5      Address of CSB
                     042A  1366 ;
                     042A  1367 ; OUTPUT PARAMETERS:
                     042A  1368 ;
                     042A  1369 ;       R0 is status
                     042A  1370 ;               TRUE implies all is well, ACCEPT or proceed with connection
                     042A  1371 ;               FALSE implies incompatibility, REJECT or break connection
                     042A  1372 ;                     requesting remote node to BUGCHECK
                     042A  1373 ;
                     042A  1374 ; SIDE EFFECTS:
                     042A  1375 ;
                     042A  1376 ;       This node will BUGCHECK if incompatible with the remote node and
                     042A  1377 ;       it appears "best" that this node exit.
                     042A  1378 ;
                     042A  1379 ;       R1 is destroyed.
                     042A  1380 ;--
                     042A  1381
                     042A  1382 CNCT_CHECK:
          1C   BB    042A  1383          PUSHR   #^M<R2,R3,R4>           ; Save registers
                     042C  1384 ;
                     042C  1385 ; First, check message size of remote system against size required by
                     042C  1386 ; clusters.
                     042C  1387 ;
    54   64 A3   D0  042C  1388          MOVL    CSB$L_CLUB(R3),R4       ; Address of CLUB
    53   7C A5   9E  0430  1389          MOVAB   CSB$B_CNCT(R5),R3       ; Address of my connect data
    52   20 A2   9E  0434  1390          MOVAB   SCSCMG$B_SNDDAT(R2),R2  ; Address of remote connect data
                     0438  1391
01 A2    41 A5   91  0438  1392          CMPB    CSB$B_VERNUM(R5), -     ; Compare remote version number to
                     043D  1393                  CNCT$B_VERNUM(R2)       ;  local version number
          07   1E    043D  1394          BGEQU   10$                     ; Branch if remote is >= local
                     043F  1395 ;
                     043F  1396 ; Get here is local node has high version number than remote node
                     043F  1397 ; If the versions are compatible, branch to 50$
                     043F  1398 ; As of now, all different versions are incompatible.
                     043F  1399 ; If the versions are incompatible, decide who should crash.
                     043F  1400 ;
28 0A A2   00   E0   043F  1401          BBS     #CNCT$V_CLUSTER, -      ; Branch if remote node is a
                     0444  1402                  CNCT$B_CLSSTS(R2),70$   ;   cluster member
          1C   11    0444  1403          BRB     40$                     ; Branch to failure exit
                     0446  1404 ;
                     0446  1405 ; Get here if version are identical or if remote is a newer (higher)
                     0446  1406 ; protocol than local.
```

```
                                0446  1407 ; Check compatibility of message buffer sizes.
                                0446  1408 ;
006B 8F    00000000'GF    B1    0446  1409 10$:    CMPW    G^SCS$GW_MAXMSG, -      ; Is local system's maximium message
                                044F  1410                  #CLSMSG$K_MAXMSG        ;   size big enough?
                          23    1F    044F  1411          BLSSU   90$               ; Branch if too small
                     50   68 A5    D0    0451  1412          MOVL    CSB$L_SB(R5),R0         ; SB address of remote system
                006B 8F   22 A0    B1    0455  1413          CMPW    SB$W_MAXMSG(R0), -      ; Compare against maximum cluster
                                045B  1414                  #CLSMSG$K_MAXMSG
                          09    1E    045B  1415          BGEQU   50$               ; Branch if it is big enough
                    0E 0A A2    00    E0    045D  1416          BBS     #CNCT$V_CLUSTER, -      ; Branch if remote node is
                                0462  1417                  CNCT$B_CLSSTS(R2),80$   ;   a cluster member
                                0462  1418
                          50    D4    0462  1419 40$:    CLRL    R0                      ; Form failure status
                          03    11    0464  1420          BRB     60$                     ; Branch to common exit
                                0466  1421
                     50   00'   D0    0466  1422 50$:    MOVL    S^#SS$_NORMAL,R0        ; Success status
                          1C    BA    0469  1423 60$:    POPR    #^M<R2,R3,R4>           ; Restore registers
                          05    046B  1424          RSB
                                046C  1425
                                046C  1426 ;
                                046C  1427 ; Get here when node must leave cluster
                                046C  1428 ;
                                046C  1429 70$:    BUG_CHECK       CLUEXIT,FATAL   ; Leave cluster because of incompatible
                                0470  1430                                          ; protocol levels
                                0470  1431
                                0470  1432 80$:    BUG_CHECK       CLUEXIT,FATAL   ; Remote node is a cluster member with an
                                0474  1433                                          ; insufficient message buffer size.  This
                                0474  1434                                          ; node may never join, so die.
                                0474  1435
                                0474  1436 90$:    BUG_CHECK       CLUEXIT,FATAL   ; Local node has too small value of SYSGEN
                                0478  1437                                          ; parameter SCSMAXMSG.
```

```
                     0478  1439          .SBTTL  RECNCT_CHECK - Verify Reconnect Data
                     0478  1440 ;++
                     0478  1441 ; FUNCTIONAL DESCRIPTION:
                     0478  1442 ;
                     0478  1443 ;       Evaluate received reconnect data vs. connect data
                     0478  1444 ;       sent or about to be sent.
                     0478  1445 ;
                     0478  1446 ; CALLING SEQUENCE:
                     0478  1447 ;
                     0478  1448 ;       BSBW    RECNCT_CHECK
                     0478  1449 ;       IPL must be at IPL$_SCS
                     0478  1450 ;
                     0478  1451 ; INPUT PARAMETERS:
                     0478  1452 ;
                     0478  1453 ;       R5      Address of CSB
                     0478  1454 ;       R2      Address of SCS connect message
                     0478  1455 ;
                     0478  1456 ; OUTPUT PARAMETERS:
                     0478  1457 ;
                     0478  1458 ;       None
                     0478  1459 ;
                     0478  1460 ; SIDE EFFECTS:
                     0478  1461 ;
                     0478  1462 ;       R0 and R1 are destroyed.
                     0478  1463 ;--
                     0478  1464
                     0478  1465 ; Legend:
                     0478  1466 ;
                     0478  1467 ;       C:      CLUB CLUSTER bit: this node is cluster member
                     0478  1468 ;       c:      CNCT CLUSTER bit: other node is cluster member
                     0478  1469 ;       Q:      CLUB QUORUM bit: cluster containing this node has quorum
                     0478  1470 ;       q:      CNCT QUORUM bit: cluster containing other node has quorum
                     0478  1471 ;       M:      CSB MEMBER bit: connection is to local cluster member
                     0478  1472 ;       m:      CNCT MEMBER bit: connection is to local cluster member
                     0478  1473 ;       L:      CSB LONG_BREAK bit: this node has seen long cnx break
                     0478  1474 ;       l:      CNCT LONG_BREAK bit: other node has seen long cnx break
                     0478  1475 ;       R:      CSB REMOVED bit: this node has removed other from cluster
                     0478  1476 ;       r:      CNCT REMOVED bit: other node has removed this node from cluster
                     0478  1477 ;
                     0478  1478 RECNCT_CHECK:
                  0C BB  0478  1479          PUSHR   #^M<R2,R3>                      ; Save registers
            53 64 A5  D0  047A  1480          MOVL    CSB$L_CLUB(R5),R3               ; Get CLUB address
            52 20 A2  9E  047E  1481          MOVAB   SCSCMG$B_SNDDAT(R2),R2          ; Address of received connect data
                     0482  1482 ;
                     0482  1483 ; If other node has seen long break,  make sure this node counts it as a
                     0482  1484 ; long break also.
                     0482  1485 ;
         02 0B A2    00  E1  0482  1486          BBC     #CNCT$V_LONG_BREAK, -           ; Branch if (NOT L) & (NOT l)
                     0487  1487                          CNCT$B_CNXSTS(R2),10$
                  65 10  0487  1488          BSBB    LONG_BREAK                      ; Treat as though a long break
         07 1C A3    00  E0  0489  1489 10$:  BBS     #CLUB$V_CLUSTER, -              ; Branch if local node is cluster
                     048E  1490                          CLUB$L_FLAGS(R3),15$            ;   member
         57 60 A5    00  E0  048E  1491          BBS     #CSB$V_LONG_BREAK, -            ; Branch if long break and bugcheck
                     0493  1492                          CSB$L_STATUS(R5),80$            ;  (NOT C) & L
                  4B 11  0493  1493          BRB     50$                             ; All seems well
                     0495  1494
         07 0B A2    02  E1  0495  1495 15$:  BBC     #CNCT$V_REMOVED, -              ; Branch if other has not removed us
```

```
                        049A  1496                    CNCT$B_CNXSTS(R2),20$   ;
04 A2    06 A2    B1    049A  1497            CMPW    CNCT$W_VOTES(R2), -     ; Does remote cluster have a quorum?
                        049F  1498                    CNCT$W_QUORUM(R2)
             45    1E    049F  1499            BGEQU   70$                     ; Branch if r & q and bugcheck
07 60 A5     02    E1    04A1  1500 20$:       BBC     #CSB$V_REMOVED, -       ; Branch if we have not removed other
                        04A6  1501                    CSB$L_STATUS(R5),30$    ;
20 A3    22 A3    B1    04A6  1502            CMPW    CLUB$W_VOTES(R3), -     ; Does local cluster have a quorum?
                        04AB  1503                    CLUB$W_QUORUM(R3)
             2F    1E    04AB  1504            BGEQU   40$                     ; Branch if we have quorum --
                        04AD  1505                                            ;  other should go: R & Q
                        04AD  1506 30$:
                        04AD  1507 ;
                        04AD  1508 ; In the following two instructions, consider the case where
                        04AD  1509 ; one node has removed the other, but not vice-versa.  In these cases,
                        04AD  1510 ; it seems necessary that outgoing messages to the other node be blocked.
                        04AD  1511 ; In actuality, LONG_BREAK should inhibit outgoing messages from all
                        04AD  1512 ; parties except the connection manager.  The inhibited messages should
                        04AD  1513 ; be immediately returned with error status.
                        04AD  1514 ;
2E 60 A5     02    E1    04AD  1515            BBC     #CSB$V_REMOVED, -       ; Branch if we have not removed
                        04B2  1516                    CSB$L_STATUS(R5),50$    ;  other node
29 0B A2     02    E1    04B2  1517            BBC     #CNCT$V_REMOVED, -      ; Branch if other node has not removed
                        04B7  1518                    CNCT$B_CNXSTS(R2),50$   ;  this node
                        04B7  1519 ;
                        04B7  1520 ; Each node has removed the other
                        04B7  1521 ; Neither node has quorum: R & r & (NOT Q) & (NOT q)
                        04B7  1522 ;
22 A3    06 A2    B1    04B7  1523            CMPW    CNCT$W_VOTES(R2), -     ; Compare available votes
                        04BC  1524                    CLUB$W_VOTES(R3)
             28    1A    04BC  1525            BGTRU   70$                     ; Other has more votes, we crash
             1C    1F    04BE  1526            BLSSU   40$                     ; This node has more votes
24 A3    08 A2    B1    04C0  1527            CMPW    CNCT$W_NODES(R2), -     ; Compare number of nodes in cluster
                        04C5  1528                    CLUB$W_NODES(R3)
             1F    1A    04C5  1529            BGTRU   70$                     ; Other has more nodes, crash
             13    1F    04C7  1530            BLSSU   40$                     ; Other nodes has more nodes
             0C    BB    04C9  1531            PUSHR   #^M<R2,R3>
       52    68 A5    D0    04CB  1532            MOVL    CSB$L_SB(R5),R2         ; Remote System Block
00000018'GF     06    29    04CF  1533            CMPC3   #SB$S_SYSTEMID, -       ; Compare system ID's
              18 A2         04D6  1534                    G^SCS$GA_LOCALSB+SB$B_SYSTEMID, -       ;  local system ID
                        04D8  1535                    SB$B_SYSTEMID(R2)       ;   remote system ID
             0C    BA    04D8  1536            POPR    #^M<R2,R3>
             0A    1A    04DA  1537            BGTRU   70$
             50    D4    04DC  1538 40$:       CLRL    R0                      ; Failure status means other
             03    11    04DE  1539            BRB     60$                     ;  node should bugcheck
                        04E0  1540
          50    01    D0    04E0  1541 50$:       MOVL    #1,R0
             0C    BA    04E3  1542 60$:       POPR    #^M<R2,R3>              ; Restore registers
             05    04E5  1543            RSB
                        04E6  1544
                        04E6  1545 ;
                        04E6  1546 ; Get here when node must leave cluster
                        04E6  1547 ;
                        04E6  1548 70$:       BUG_CHECK       CLUEXIT,FATAL   ; Leave cluster
                        04EA  1549
                        04EA  1550 ;
                        04EA  1551 ; Get here when two nodes not part of a cluster regain a connection after
                        04EA  1552 ; a long break.
```

```
                ;
80$:    BUG_CHECK       CLUEXIT,FATAL   ; Reboot to avoid inconsistency
```

```
                                04EE  1556          .SBTTL  LONG_BREAK - Long Break in Connection
                                04EE  1557 ;++
                                04EE  1558 ; FUNCTIONAL DESCRIPTION:
                                04EE  1559 ;
                                04EE  1560 ;       Connection has remained broken for a long time.
                                04EE  1561 ;       All un-acked messages have their fork process resumed with
                                04EE  1562 ;       a failure status.
                                04EE  1563 ;
                                04EE  1564 ; CALLING SEQUENCE:
                                04EE  1565 ;
                                04EE  1566 ;       BSBW    LONG_BREAK
                                04EE  1567 ;       IPL must be at IPL$_SCS
                                04EE  1568 ;
                                04EE  1569 ; INPUT PARAMETERS:
                                04EE  1570 ;
                                04EE  1571 ;       R5      Address of CSB
                                04EE  1572 ;
                                04EE  1573 ; OUTPUT PARAMETERS:
                                04EE  1574 ;
                                04EE  1575 ;       None
                                04EE  1576 ;
                                04EE  1577 ; SIDE EFFECTS:
                                04EE  1578 ;
                                04EE  1579 ;       R0 and R1 are destroyed.
                                04EE  1580 ;--
                                04EE  1581
                                04EE  1582 LONG_BREAK:
0B 60 A5    00    E2            04EE  1583          BBSS    #CSB$V_LONG_BREAK, -    ; Mark long break seen and
                                04F3  1584                  CSB$L_STATUS(R5), 10$   ;   return if already set
 50    0000'CF    9E            04F3  1585          MOVAB   FAILIO_MSG,R0           ; Address of message
           FB05'   30            04F8  1586          BSBW    CNX$CONFIG_CHANGE       ; Note configuration change
           FB02'   30            04FB  1587          BSBW    CNX$CON_BREAK           ; Report long break
  07    43 A5    91            04FE  1588 10$:      CMPB    CSB$B_STATE(R5), -      ; Are we disconnecting?
                                0502  1589                  #CSB$K_DISCONNECT
             03    13            0502  1590          BEQL    20$                     ; Branch if yes
           FAF9'   30            0504  1591          BSBW    CNX$FAIL_MSG            ; Complete outstanding messages
                   05            0507  1592 20$:      RSB
```

```
                        0508  1594          .SBTTL  CNX$DECREFCNT - Decrement CSB Reference Count
                        0508  1595
                        0508  1596 ;++
                        0508  1597 ; FUNCTIONAL DESCRIPTION:
                        0508  1598 ;
                        0508  1599 ;       This routine decrements a CSB reference count and deletes
                        0508  1600 ;       the CSB when the reference count goes to 0.
                        0508  1601 ;
                        0508  1602 ; CALLING SEQUENCE:
                        0508  1603 ;
                        0508  1604 ;       BSBB    CNX$DECREFCNT
                        0508  1605 ;
                        0508  1606 ; INPUT PARAMETERS:
                        0508  1607 ;
                        0508  1608 ;       R5      Address of CSB
                        0508  1609 ;
                        0508  1610 ; IMPLICIT INPUTS:
                        0508  1611 ;
                        0508  1612 ;       NONE
                        0508  1613 ;
                        0508  1614 ; OUTPUT PARAMETERS:
                        0508  1615 ;       R5:     Address of CSB, if not deleted
                        0508  1616 ;       R5:     Contents of CSB$L_SYSQBL, if CSB deleted
                        0508  1617 ;
                        0508  1618 ; IMPLICIT OUTPUTS:
                        0508  1619 ;       NONE
                        0508  1620 ;
                        0508  1621 ; COMPLETION CODES:
                        0508  1622 ;       NONE
                        0508  1623 ;
                        0508  1624 ; SIDE EFFECTS:
                        0508  1625 ;       R2-R4 preserved
                        0508  1626 ;
                        0508  1627 ;--
                        0508  1628
                        0508  1629 CNX$DECREFCNT::
        6C A5     97    0508  1630          DECB    CSB$B_REF_CNT(R5)       ; Decrement reference count
          2A      12    050B  1631          BNEQ    20$                     ; Branch if non-zero
         FAF0'    30    050D  1632          BSBW    CNX$FAIL_MSG            ; Fail any outstanding messages
          26      10    0510  1633          BSBB    DELETE_TQE              ; Flush timer queue entry
50   0000'CF     9E    0512  1634          MOVAB   DEAD_MSG,R0             ; Address of dead node message
         FAE6'    30    0517  1635          BSBW    CNX$CONFIG_CHANGE       ; Report configuration change
50    68 A5      D0    051A  1636          MOVL    CSB$L_SB(R5),R0         ; SB address
5C A0    55      D1    051E  1637          CMPL    R5,SB$L_CSB(R0)         ; Is this CSB pointed to?
          03      12    0522  1638          BNEQ    10$
        5C A0     D4    0524  1639          CLRL    SB$L_CSB(R0)            ; Invalidate back pointer
        04 A5     DD    0527  1640 10$:     PUSHL   CSB$L_SYSQBL(R5)        ; Backward link
          0C      BB    052A  1641          PUSHR   #^M<R2,R3>              ; Save registers
     50   65      0F    052C  1642          REMQUE  (R5),R0                 ; Unlink CSB
00000000'GF       16    052F  1643          JSB     G^EXE$DEANONPAGED       ; Deallocate it
          2C      BA    0535  1644          POPR    #^M<R2,R3,R5>           ; Restore registers
                  05    0537  1645 20$:     RSB
```

```
                     0538  1647          .SBTTL  DELETE_TQE - Delete a TQE linked to a CSB
                     0538  1648
                     0538  1649 ;++
                     0538  1650 ; FUNCTIONAL DESCRIPTION:
                     0538  1651 ;
                     0538  1652 ;       This routine deallocates a TQE linked to a CSB.
                     0538  1653 ;
                     0538  1654 ; CALLING SEQUENCE:
                     0538  1655 ;
                     0538  1656 ;       BSBB    DELETE_TQE
                     0538  1657 ;
                     0538  1658 ; INPUT PARAMETERS:
                     0538  1659 ;
                     0538  1660 ;       R5      Address of CSB
                     0538  1661 ;
                     0538  1662 ; IMPLICIT INPUTS:
                     0538  1663 ;
                     0538  1664 ;       CSB$L_TQE is 0 or address of TQE
                     0538  1665 ;
                     0538  1666 ; OUTPUT PARAMETERS:
                     0538  1667 ;       NONE
                     0538  1668 ;
                     0538  1669 ; IMPLICIT OUTPUTS:
                     0538  1670 ;       NONE
                     0538  1671 ;
                     0538  1672 ; COMPLETION CODES:
                     0538  1673 ;       NONE
                     0538  1674 ;
                     0538  1675 ; SIDE EFFECTS:
                     0538  1676 ;       R0 and R1 destroyed.
                     0538  1677 ;
                     0538  1678 ;--
                     0538  1679
                     0538  1680 DELETE_TQE:
50   44 A5   D0      0538  1681          MOVL    CSB$L_TQE(R5),R0        ; Get address of TQE
        10   13      053C  1682          BEQL    10$                     ; There isn't one
        0C   BB      053E  1683          PUSHR   #^M<R2,R3>              ; Save registers
   50   60   0F      0540  1684          REMQUE  (R0),R0                 ; Remove from timer queue
00000000'GF  16      0543  1685          JSB     G^EXE$DEANONPAGED       ; Deallocate it
   44   A5   D4      0549  1686          CLRL    CSB$L_TQE(R5)           ; Clear pointer
        0C   BA      054C  1687          POPR    #^M<R2,R3>              ; Restore registers
             05      054E  1688 10$:     RSB
```

```
                        054F  1690          .SBTTL  DEAD_NODE - Manage death of a node
                        054F  1691 ;++
                        054F  1692 ; FUNCTIONAL DESCRIPTION:
                        054F  1693 ;
                        054F  1694 ;       Called when a new incarnation of a system has been seen to
                        054F  1695 ;       remove the last vestiges of knowledge of the old incarnation.
                        054F  1696 ;       No connection to the system exists.  The CSB is deleted.
                        054F  1697 ;
                        054F  1698 ; CALLING SEQUENCE:
                        054F  1699 ;
                        054F  1700 ;       BSBB    DEAD_NODE
                        054F  1701 ;
                        054F  1702 ; INPUT PARAMETERS:
                        054F  1703 ;
                        054F  1704 ;       R5      Address of CSB
                        054F  1705 ;
                        054F  1706 ; IMPLICIT INPUTS:
                        054F  1707 ;
                        054F  1708 ;       NONE
                        054F  1709 ;
                        054F  1710 ; OUTPUT PARAMETERS:
                        054F  1711 ;
                        054F  1712 ;       NONE
                        054F  1713 ;
                        054F  1714 ; IMPLICIT OUTPUTS:
                        054F  1715 ;       NONE
                        054F  1716 ;
                        054F  1717 ; COMPLETION CODES:
                        054F  1718 ;       NONE
                        054F  1719 ;
                        054F  1720 ; SIDE EFFECTS:
                        054F  1721 ;       R0 and R1 are destroyed.  R5 is invalidated.
                        054F  1722 ;
                        054F  1723 ;--
                        054F  1724
                        054F  1725 DEAD_NODE:
9D   10                 054F  1726          BSBB    LONG_BREAK              ; Simulate connection broken for long time
E5   10                 0551  1727          BSBB    DELETE_TQE              ; Get rid of timer
B3   10                 0553  1728          BSBB    CNX$DECREFCNT           ; Get rid of CSB (eventually)
     05                 0555  1729          RSB
```

```
                        0556  1731          .SBTTL  CNX_STATUS_CHECK - Check SCS failure message
                        0556  1732 ;++
                        0556  1733 ; FUNCTIONAL DESCRIPTION:
                        0556  1734 ;
                        0556  1735 ;       Check SCS failure message and BUGCHECK if the remote node has requested
                        0556  1736 ;       it.
                        0556  1737 ;
                        0556  1738 ; CALLING SEQUENCE:
                        0556  1739 ;
                        0556  1740 ;       JSB     CNX_STATUS_CHECK
                        0556  1741 ;
                        0556  1742 ; INPUT PARAMETERS:
                        0556  1743 ;
                        0556  1744 ;       R5      CSB address
                        0556  1745 ;       R0      SCS Reason code
                        0556  1746 ;       R1      SYSAP reason (if R0=SS$_REJECT)
                        0556  1747 ;
                        0556  1748 ; IMPLICIT INPUTS:
                        0556  1749 ;
                        0556  1750 ;       NONE
                        0556  1751 ;
                        0556  1752 ; OUTPUT PARAMETERS:
                        0556  1753 ;
                        0556  1754 ;       NONE
                        0556  1755 ;
                        0556  1756 ; IMPLICIT OUTPUTS:
                        0556  1757 ;
                        0556  1758 ;       NONE
                        0556  1759 ;
                        0556  1760 ; COMPLETION CODES:
                        0556  1761 ;
                        0556  1762 ;       NONE
                        0556  1763 ;
                        0556  1764 ; SIDE EFFECTS:
                        0556  1765 ;
                        0556  1766 ;       NONE
                        0556  1767 ;
                        0556  1768 ;--
                        0556  1769
                        0556  1770 CNX_STATUS_CHECK:
          03    BB      0556  1771          PUSHR   #^M<R0,R1>                      ; Save registers
0000'8F   50    B1      0558  1772          CMPW    R0,#SS$_REJECT                  ; Is this a connection request reject?
          07    13      055D  1773          BEQL    10$                             ; Branch if yes
0000'8F   50    B1      055F  1774          CMPW    R0,#SS$_DISCONNECT              ; Is this a requested disconnect?
          14    12      0564  1775          BNEQ    20$                             ; Branch if no
  10 51   0F    E1      0566  1776 10$:     BBC     #CLMDRS$V_DRS,R1,20$            ; Branch if not a cluster disconnect code
  24 51   08    E0      056A  1777          BBS     #CLMDRS$V_FATAL,R1,50$          ; Branch if bugcheck requested
0000'8F   50    B1      056E  1778          CMPW    R0,#SS$_DISCONNECT              ; Is this a requested disconnect?
          05    12      0573  1779          BNEQ    20$                             ; Branch if no
       51   0A    91      0575  1780          CMPB    #CLMDRS$C_REMOVED,R1            ; Is this node removed from cluster?
          14    13      0578  1781          BEQL    40$                             ; Branch if local node removed and exit
0000'8F   50    B1      057A  1782 20$:     CMPW    R0,#SS$_VCBROKEN                ; Is this a circuit failure?
          0A    12      057F  1783          BNEQ    30$                             ; Branch if no
0000'8F   51    B1      0581  1784          CMPW    R1,#SS$_NOSUCHNODE              ; Is this a result of a "last gasp"?
          03    12      0586  1785          BNEQ    30$                             ; Branch if no
        FF63    30      0588  1786          BSBW    LONG_BREAK                      ; Declare a long break
          03    BA      058B  1787 30$:     POPR    #^M<R0,R1>                      ; Restore registers
```

```
05  058D  1788          RSB                                 ; Return to caller
    058E  1789
    058E  1790 40$:      BUG_CHECK       CLUEXIT,FATAL       ; This node removed from cluster
    0592  1791
    0592  1792 50$:      BUG_CHECK       CNXMGRERR,FATAL     ; Bugcheck requested by disconnecting remote
    0596  1793
```

```
                     0596  1795          .SBTTL  CNX$LOOKUP_CSB - Lookup a CSB given a SB address
                     0596  1796 ;++
                     0596  1797 ; FUNCTIONAL DESCRIPTION:
                     0596  1798 ;
                     0596  1799 ;       CNX$LOOKUP_CSB find a CSB with matching System ID and
                     0596  1800 ;               software incarnation number given an SB address.
                     0596  1801 ;
                     0596  1802 ; CALLING SEQUENCE:
                     0596  1803 ;
                     0596  1804 ;       JSB     CNX$LOOKUP_CSB
                     0596  1805 ;
                     0596  1806 ; INPUT PARAMETERS:
                     0596  1807 ;
                     0596  1808 ;       R1      Address of SB
                     0596  1809 ;
                     0596  1810 ; IMPLICIT INPUTS:
                     0596  1811 ;
                     0596  1812 ;       NONE
                     0596  1813 ;
                     0596  1814 ; OUTPUT PARAMETERS:
                     0596  1815 ;
                     0596  1816 ;       R5 is address of CSB
                     0596  1817 ;
                     0596  1818 ; IMPLICIT OUTPUTS:
                     0596  1819 ;       NONE
                     0596  1820 ;
                     0596  1821 ; COMPLETION CODES:
                     0596  1822 ;       R0 contains status
                     0596  1823 ;
                     0596  1824 ; SIDE EFFECTS:
                     0596  1825 ;       R1 is destroyed
                     0596  1826 ;
                     0596  1827 ;--
                     0596  1828
                     0596  1829 CNX$LOOKUP_CSB::                             ; Lookup given SB address
          1C   BB    0596  1830          PUSHR   #^M<R2,R3,R4>               ; Save registers
     54   51   D0    0598  1831          MOVL    R1,R4                       ; SB address
55   5C   A4   D0    059B  1832          MOVL    SB$L_CSB(R4),R5             ; Get CSB for this SB
          2D   13    059F  1833          BEQL    30$                         ; Branch if no CSB for this SB
                     05A1  1834
                     05A1  1835          ; Check software incarnation.
                     05A1  1836
34 60 A5  18   E0    05A1  1837          BBS     #CSB$V_LOCAL,-              ; Skip if local system
                     05A6  1838                  CSB$L_STATUS(R5),50$
          08   29    05A6  1839          CMPC3   #CSB$S_SWINCARN,-           ; Software incarnations match?
     38   A5         05A8  1840                  CSB$Q_SWINCARN(R5),-
     2C   A4         05AA  1841                  SB$Q_SWINCARN(R4)
          2C   13    05AC  1842          BEQL    50$                         ; Branch if yes and exit
                     05AE  1843          ;
                     05AE  1844          ; There is an existing CSB with a different software incarnation.
                     05AE  1845          ; Get rid of it and fail over that node (perahps for the second time!)
                     05AE  1846          ;
                     05AE  1847          STATE_DISP      <<NEW,40$>,<DEAD,30$>,<WAIT,20$>,<RECONNECT,10$>>
                     05BE  1848          BUG_CHECK       CNXMGRERR,FATAL ; Temporary Bugcheck
                     05C2  1849
          0A   90    05C2  1850 10$:     MOVB    #CSB$K_DEAD,-               ; Set state=DEAD
     43   A5         05C4  1851                  CSB$B_STATE(R5)
```

```
                06   11  05C6  1852          BRB     30$                     ; Branch to allocate new block
                                05C8  1853
                0A   90  05C8  1854  20$:     MOVB    #CSB$K_DEAD,-           ; Set state=DEAD
             43 A5       05CA  1855                  CSB$B_STATE(R5)
                81   10  05CC  1856          BSBB    DEAD_NODE               ; Handle dead node
          51    54   D0  05CE  1857  30$:     MOVL    R4,R7                   ; SB address
                0D   10  05D1  1858          BSBB    CNX$CREATE_CSB          ; Create new CSB
                08   11  05D3  1859          BRB     60$                     ; Return with status
                                05D5  1860
    38 A5    2C A4   7D  05D5  1861  40$:     MOVQ    SB$Q_SWINCARN(R4), -    ; Update software incarnation and
                                05DA  1862                  CSB$Q_SWINCARN(R5)      ;   continue
          50    00'  3C  05DA  1863  50$:     MOVZWL  S^#SS$_NORMAL,R0        ; Found CSB, in R5
                1C   BA  05DD  1864  60$:     POPR    #^M<R2,R3,R4>           ; Restore nonvolatile registers
                     05  05DF  1865          RSB
```

```
                        05E0  1867          .SBTTL  CNX$CREATE_CSB - Create a new CSB given a SB address
                        05E0  1868 ;++
                        05E0  1869 ; FUNCTIONAL DESCRIPTION:
                        05E0  1870 ;
                        05E0  1871 ;       CNX$CREATE_CSB creates a CSB with matching System ID and
                        05E0  1872 ;               software incarnation number given an SB address.
                        05E0  1873 ;       It is assumed that no similar CSB already exists.
                        05E0  1874 ;
                        05E0  1875 ; CALLING SEQUENCE:
                        05E0  1876 ;
                        05E0  1877 ;       JSB     CNX$CREATE_CSB
                        05E0  1878 ;
                        05E0  1879 ; INPUT PARAMETERS:
                        05E0  1880 ;
                        05E0  1881 ;       R1      Address of SB
                        05E0  1882 ;
                        05E0  1883 ; IMPLICIT INPUTS:
                        05E0  1884 ;
                        05E0  1885 ;       NONE
                        05E0  1886 ;
                        05E0  1887 ; OUTPUT PARAMETERS:
                        05E0  1888 ;
                        05E0  1889 ;       R5 is address of CSB
                        05E0  1890 ;
                        05E0  1891 ; IMPLICIT OUTPUTS:
                        05E0  1892 ;       NONE
                        05E0  1893 ;
                        05E0  1894 ; COMPLETION CODES:
                        05E0  1895 ;       R0 contains status
                        05E0  1896 ;
                        05E0  1897 ; SIDE EFFECTS:
                        05E0  1898 ;       R1 is destroyed
                        05E0  1899 ;
                        05E0  1900 ;--
                        05E0  1901
                        05E0  1902 CNX$CREATE_CSB::                             ; Lookup given SB address
     00DC 8F    BB      05E0  1903          PUSHR   #^M<R2,R3,R4,R6,R7>        ; Save registers
     57   51    D0      05E4  1904          MOVL    R1,R7                      ; SB address
51   00AC 8F    3C      05E7  1905          MOVZWL  #CSB$K_LENGTH,R1           ; Size of CSB
        FA11'   30      05EC  1906          BSBW    CNX$ALLOZMEM               ; Allocate and zero memory
       03 50    E8      05EF  1907          BLBS    R0,10$                     ; Branch if successful
        009C    31      05F2  1908          BRW     50$                        ; Exit, status in R0
                        05F5  1909
     56   52    D0      05F5  1910 10$:     MOVL    R2,R6                      ; CSB address
5C   A7   56    D0      05F8  1911          MOVL    R6,SB$L_CSB(R7)            ; Update SB to point to newest CSB
           01    90      05FC  1912          MOVB    #DYN$C_CLU_CSB,-           ; Store subtype
        0B A6            05FE  1913                  CSB$B_SUBTYPE(R6)
        14 A6    DE      0600  1914          MOVAL   CSB$L_SENTQFL(R6),-        ; Initialize sent list
        18 A6            0603  1915                  CSB$L_SENTQBL(R6)
        14 A6    D4      0605  1916          CLRL    CSB$L_SENTQFL(R6)
        1C A6    DE      0608  1917          MOVAL   CSB$L_RESENDQFL(R6),-      ; Initialize resend list
        20 A6            060B  1918                  CSB$L_RESENDQBL(R6)
        1C A6    D4      060D  1919          CLRL    CSB$L_RESENDQFL(R6)
34 A6      01    D0      0610  1920          MOVL    #1,CSB$L_CURRCDRP(R6)      ; Block critical section in SEND_MSG
        24 A6    9E      0614  1921          MOVAB   CSB$L_WARMCDRPQFL(R6),-    ; Initialize warm CDRP queue
        24 A6            0617  1922                  CSB$L_WARMCDRPQFL(R6)
        24 A6    9E      061?  1923          MOVAB   CSB$L_WARMCDRPQFL(R6),-
```

```
                        28 A6          061C  1924                    CSB$L_WARMCDRPQBL(R6)
                                       061E  1925
                                       061E  1926            ; Store remote side's software incarnation number and system id.
                                       061E  1927            ; so that if this connection breaks and another is established,
                                       061E  1928            ; we can determine if it's the same system and software
                                       061E  1929            ; incarnation at the other end.
                                       061E  1930
                        2C A7    7D    061E  1931    MOVQ    SB$Q_SWINCARN(R7),-     ; Store software incarnation number
                        38 A6          0621  1932            CSB$Q_SWINCARN(R6)
                           04    90    0623  1933    MOVB    #CSB$K_NEW,-            ; Set state to NEW
                        43 A6          0625  1934            CSB$B_STATE(R6)         ;
                58 A6   58 A6    DE    0627  1935    MOVAL   CSB$L_PARTNERQFL(R6),- ; Initialize block transfer
                                       062C  1936            CSB$L_PARTNERQFL(R6)    ; partners queue.
                5C A6   58 A6    DE    062C  1937    MOVAL   CSB$L_PARTNERQFL(R6),-
                                       0631  1938            CSB$L_PARTNERQBL(R6)
            74 A6   00000000'GF   7D    0631  1939    MOVQ    G^EXE$GQ_SYSTIME,-      ; Stamp reference time in CSB
                                       0639  1940            CSB$Q_REFTIME(R6)
            64 A6   00000000'GF   D0    0639  1941    MOVL    G^CLU$GL_CLUB,-         ; Address of CLUB
                                       0641  1942            CSB$L_CLUB(R6)
                    6C A6    01    90    0641  1943    MOVB    #1,CSB$B_REF_CNT(R6)    ; Initialize reference count
                    68 A6    57    D0    0645  1944    MOVL    R7,CSB$L_SB(R6)         ; Address of SB
                    50   7C A6    9E    0649  1945    MOVAB   CSB$B_CNCT(R6),R0       ; Connect data block
                    60   00'8F    90    064D  1946    MOVB    I^#0,CNCT$B_ECOLVL(R0)  ; ECO level, set for easy patching
                    01 A0    0C    90    0651  1947    MOVB    #CNCT$K_PROTOCOL,-      ; Protocol level
                                       0655  1948            CNCT$B_VERNUM(R0)
                    02 A0    01    90    0655  1949    MOVB    #CNCT$K_INITIAL,-       ; Initial connect
                                       0659  1950            CNCT$B_TYPE(R0)
                    03 A0    04    90    0659  1951    MOVB    #SEND_CREDITS-1,-       ; Unacknowledged message limit is
                                       065D  1952            CNCT$B_ACKLIM(R0)       ;   send credits - 1.
                                       065D  1953    ASSUME  CLUB$L_CSBQFL EQ 0
                                       065D  1954    ASSUME  CSB$L_SYSQFL EQ 0
                55   64 A6    D0    065D  1955    MOVL    CSB$L_CLUB(R6),R5       ; Get address of CSB queue header
                     55   65  D0    0661  1956 20$: MOVL    CSB$L_SYSQFL(R5),R5     ; Get address of next CSB
                64 A6   55    D1    0664  1957    CMPL    R5,CSB$L_CLUB(R6)       ; Reached end of list?
                           0C 13    0668  1958    BEQL    30$                     ; Yes
                50   68 A5    D0    066A  1959    MOVL    CSB$L_SB(R5),R0         ; This CSB's SB address
        18 A0   18 A7    06    29    066E  1960    CMPC3   #SB$S_SYSTEMID,-        ; Compare system IDs
                                       0674  1961            SB$B_SYSTEMID(R7),-
                                       0674  1962            SB$B_SYSTEMID(R0)
                           EB 14    0674  1963    BGTR    20$                     ; Branch if no match
                04 B5      66    0E    0676  1964 30$: INSQUE  CSB$L_SYSQFL(R6),-
                                       067A  1965            @CSB$L_SYSQBL(R5)
                     55    56    D0    067A  1966    MOVL    R6,R5                   ; Set up result register
            00000000'8F    57    D1    067D  1967    CMPL    R7,#SCS$GA_LOCALSB      ; Is this the local SB?
                           08 13    0684  1968    BEQL    40$                     ; Skip message output
                                       0686  1969                                    ;  because we are at IPL 31!
                50   0000'CF    9E    0686  1970    MOVAB   CSB_MSG,R0              ; Address of new CSB message
                        F972'  30    068B  1971    BSBW    CNX$CONFIG_CHANGE       ; Log CSB creation
                     50   00'  3C    068E  1972 40$: MOVZWL  S^#SS$_NORMAL,R0        ; Found CSB, in R5
                     00DC 8F    BA    0691  1973 50$: POPR    #^M<R2,R3,R4,R6,R7>     ; Restore nonvolatile registers
                                05    0695  1974    RSB
```

```
                        0696  1976          .SBTTL  DISPATCH - Dispatch on CSB state
                        0696  1977 ;++
                        0696  1978 ; FUNCTIONAL DESCRIPTION:
                        0696  1979 ;
                        0696  1980 ;       This routine dispatches based upon the CSB state
                        0696  1981 ;       using a table assembled in-line at the call site.
                        0696  1982 ;       The STATE_DISP macro defines and builds the table.
                        0696  1983 ;
                        0696  1984 ; INPUT PARAMETERS:
                        0696  1985 ;
                        0696  1986 ;       (SP)    First byte of dispatch table
                        0696  1987 ;       R5      Address of CSB
                        0696  1988 ;
                        0696  1989 ; OUTPUT PARAMETERS:
                        0696  1990 ;
                        0696  1991 ;       NONE
                        0696  1992 ;
                        0696  1993 ; SIDE EFFECTS:
                        0696  1994 ;
                        0696  1995 ;       All registers preserved.
                        0696  1996 ;--
                        0696  1997
                        0696  1998 DISPATCH::
        03      BB      0696  1999          PUSHR   #^M<R0,R1>              ; Save R0 and R1
50  08  AE      D0      0698  2000          MOVL    8(SP),R0                ; Fetch table address
    51  80      9A      069C  2001 10$:     MOVZBL  (R0)+,R1                ; Fetch state code
        0E      13      069F  2002          BEQL    30$                     ; End of table
43  A5  51      91      06A1  2003          CMPB    R1,CSB$B_STATE(R5)      ; Match?
        05      13      06A5  2004          BEQL    20$                     ; Branch on match
    50  02      C0      06A7  2005          ADDL2   #2,R0                   ; Bump over word offset
        F0      11      06AA  2006          BRB     10$
                        06AC  2007
    51  80      32      06AC  2008 20$:     CVTWL   (R0)+,R1                ; Fetch word offset
08  AE  51      50      C1    06AF  2009 30$:     ADDL3   R0,R1,8(SP)       ; Store return address
        03      BA      06B4  2010          POPR    #^M<R0,R1>              ; Restore registers
                05      06B6  2011          RSB
                        06B7  2012
                        06B7  2013
                        06B7  2014          .END
```

```
ACCPT_MSG                       ********  X   04
BUG$_CLUEXIT                    ********  X   04
BUG$_CNXMGRERR                  ********  X   04
CDT$C_AUXSTRUC                  =  0000005C
CDT$L_PB                        =  0000001C
CJF$MIN_JOURNAL                 ********  X   03
CLMDRS$C_PROTOCOL               =  00000004
CLMDRS$C_REMOVED                =  0000000A
CLMDRS$C_VERSION                =  00000006
CLMDRS$M_DRS                    =  00008000
CLMDRS$M_FATAL                  =  00000100
CLMDRS$V_DRS                    =  0000000F
CLMDRS$V_FATAL                  =  00000008
CLSMSG$K_MAXMSG                 =  0000006B
CLU$GL_CLUB                     ********  X   03
CLU$GL_CLUSVEC                  ********  X   03
CLU$GW_LCKDIRWT                 ********  X   03
CLU$GW_MAXINDEX                 ********  X   03
CLU$GW_QDSKVOTES                ********  X   03
CLU$GW_QUORUM                   ********  X   03
CLU$GW_RECNXINT                 ********  X   04
CLU$GW_VOTES                    ********  X   03
CLUB$B_CLUBPWF                  =  0000018C
CLUB$B_FORK_BLOCK               =  000000CC
CLUB$B_SUBTYPE                  =  0000000B
CLUB$K_LENGTH                   =  000001A8
CLUB$L_CSBQBL                   =  00000004
CLUB$L_CSBQFL                   =  00000000
CLUB$L_FLAGS                    =  0000001C
CLUB$L_JNL_DISPT                =  00000014
CLUB$L_LOCAL_CSB                =  00000010
CLUB$L_POLL_CTX                 =  0000000C
CLUB$M_CLUSTER                  =  00000001
CLUB$V_CLUSTER                  =  00000000
CLUB$W_FIRST_INDEX              =  00000066
CLUB$W_NODES                    =  00000024
CLUB$W_QDVOTES                  =  000000AE
CLUB$W_QUORUM                   =  00000020
CLUB$W_VOTES                    =  00000022
CLUBFKB$S_FORK_BLOCK            =  00000018
CLUBPWF$S_FORK_BLOCK            =  00000018
CNCT$B_ACRLIM                   =  00000003
CNCT$B_CLSSTS                   =  0000000A
CNCT$B_CNXSTS                   =  0000000B
CNCT$B_ECOLVL                   =  00000000
CNCT$B_TYPE                     =  00000002
CNCT$B_VERNUM                   =  00000001
CNCT$K_INITIAL                  =  00000001
CNCT$K_PROTOCOL                 =  0000000C
CNCT$K_RECONNECT                =  00000002
CNCT$M_CLUSTER                  =  00000001
CNCT$M_LONG_BREAK               =  00000001
CNCT$M_MEMBER                   =  00000002
CNCT$M_REMOVED                  =  00000004
CNCT$V_CLUSTER                  =  00000000
CNCT$V_LONG_BREAK               =  00000000
CNCT$V_REMOVED                  =  00000002
CNCT$W_NODES                    =  00000008
CNCT$W_QUORUM                   =  00000004
CNCT$W_RCVDSEQNM                =  0000000C
CNCT$W_VOTES                    =  0C000006
CNCT_CHECK                      0000042A R      04
CNCT_DATA                       000003FF R      04
CNCT_MSG                        ********    X   04
CNX$ACCEPT                      00000222 RG     04
CNX$ALLOZMEM                    ********    X   03
CNX$BREAK                       00000309 RG     04
CNX$CHECK_QUORUM                ********    X   04
CNX$CONFIG_CHANGE               ********    X   04
CNX$CONNECT                     00000038 RG     04
CNX$CON_BREAK                   ********    X   04
CNX$CON_INIT                    ********    X   03
CNX$CON_NEWSYS                  ********    X   04
CNX$CREATE_CSB                  000005E0 RG     04
CNX$DECREFCNT                   00000508 RG     04
CNX$DISCONNECT                  00000331 RG     04
CNX$DISC_BUGCHECK               000002CF RG     04
CNX$DISC_PROTOCOL               000002E1 RG     04
CNX$DISC_REMOVE                 000002D6 RG     04
CNX$ERROR                       000002F6 RG     04
CNX$FAIL_MSG                    ********    X   04
CNX$INIT                        00000000 RG     03
CNX$LOOKUP_CSB                  00000596 RG     04
CNX$NEWSYSTEM                   00000007 RG     04
CNX$NEW_CSB                     00000031 RG     04
CNX$POST_CLEANUP                ********    X   04
CNX$PRE_CLEANUP                 ********    X   04
CNX$RCV_CNCT_MSG                0000015B R      04
CNX$RCV_MSG                     ********    X   04
CNX$RESEND_MSGS                 ********    X   04
CNX$WAIT                        0000036A RG     04
CNXERROR_MSG                    ********    X   04
CNX_STATUS_CHECK                00000556 R      04
CSB$B_CNCT                      =  0000007C
CSB$B_ECOLVL                    =  0000C040
CSB$B_REF_CNT                   =  0000006C
CSB$B_REMACKLIM                 =  00000033
CSB$B_STATE                     =  00000043
CSB$B_SUBTYPE                   =  0000000B
CSB$B_VERNUM                    =  00000041
CSB$K_ACCEPT                    =  00000006
CSB$K_CONNECT                   =  00000005
CSB$K_DEAD                      =  0000000A
CSB$K_DISCONNECT                =  00000007
CSB$K_LENGTH                    =  000000AC
CSB$K_LOCAL                     =  0000000B
CSB$K_NEW                       =  00000004
CSB$K_OPEN                      =  00000001
CSB$K_REACCEPT                  =  00000008
CSB$K_RECONNECT                 =  00000003
CSB$K_WAIT                      =  00000009
CSB$L_CDT                       =  0000000C
CSB$L_CLUB                      =  00000064
CSB$L_CURRCDRP                  =  00000034
```

```
CSB$L_PARTNERQBL        = 0000005C
CSB$L_PARTNERQFL        = 00000058
CSB$L_PDT               = 00000010
CSB$L_RESENDQBL         = 00000020
CSB$L_RESENDQFL         = 0000001C
CSB$L_SB                = 00000068
CSB$L_SENTQBL           = 00000018
CSB$L_SENTQFL           = 00000014
CSB$L_STATUS            = 00000060
CSB$L_SYSQBL            = 00000004
CSB$L_SYSQFL            = 00000000
CSB$L_TIMEOUT           = 00000048
CSB$L_TQE               = 00000044
CSB$L_WARMCDRPQBL       = 00000028
CSB$L_WARMCDRPQFL       = 00000024
CSB$M_LOCAL             = 01000000
CSB$M_LONG_BREAK        = 00000001
CSB$M_MEMBER            = 00000002
CSB$M_REMOVED           = 00000004
CSB$Q_REFTIME           = 00000074
CSB$Q_SWINCARN          = 00000038
CSB$S_SWINCARN          = 00000008
CSB$V_LOCAL             = 00000018
CSB$V_LONG_BREAK        = 00000000
CSB$V_REMOVED           = 00000002
CSB$W_ACKRSEQNM         = 00000030
CSB$W_LCKDIRWT          = 00000054
CSB$W_QDVOTES           = 00000056
CSB$W_QUORUM            = 00000052
CSB$W_RCVDSEQNM         = 0000002E
CSB$W_VOTES             = 00000050
CSB_MSG                 ******** X    04
DEAD_MSG                ******** X    04
DEAD_NODE               0000054F R    04
DELETE_TQE              00000538 R    04
DISC_STATUS             000002E6 R    04
DISPATCH                00000696 RG   04
DYN$C_CLU_CLUB          = 00000003
DYN$C_CLU_CLUVEC        = 00000002
DYN$C_CLU_CSB           = 00000001
DYN$C_TQE               = 0000000F
EXE$ALONONPAGED         ******** X    04
EXE$AL_TQENOREPT        ******** X    04
EXE$DEANONPAGED         ******** X    04
EXE$GL_ABSTIM           ******** X    04
EXE$GQ_SYSTIME          ******** X    04
EXE$INSTIMQ             ******** X    04
FAILIO_MSG              ******** X    04
FKB$B_FIPL              = 0000000B
FKB$K_LENGTH            = 00000018
IPL$_SCS                = 00000008
IPL$_SYNCH              = 00000008
IPL$_TIMER              = 00000008
LISTEN_ERROR            00000000 R    04
LONG_BREAK              000004EE R    04
PB$L_SBLINK             = 00000030
PDT$C_REJECT            = 0000004C
PROC_NAME               00000000 R    02
REACCPT_MSG             ******** X    04
RECNCT_CHECK            00000478 R    04
RECNCT_MSG              ******** X    04
RETRY_CONNECT           000003DA R    04
SB$B_SYSTEMID           = 00000018
SB$L_CSB                = 0000005C
SB$Q_SWINCARN           = 0000002C
SB$S_SYSTEMID           = 00000006
SB$W_MAXMSG             = 00000022
SCS$ACCEPT              ******** GX   04
SCS$CONFIG_SYS          ******** X    04
SCS$CONNECT             ******** X    04
SCS$DISCONNECT          ******** X    04
SCS$GA_LOCALSB          ******** X    03
SCS$GW_MAXMSG           ******** X    04
SCS$LISTEN              ******** GX   03
SCS$POLL_MODE           ******** X    03
SCS$POLL_PROC           ******** X    03
SCSCMG$B_SNDDAT         = 00000020
SEND_CREDITS            = 00000005
SS$_DISCONNECT          ******** X    04
SS$_NORMAL              ******** X    04
SS$_NOSUCHNODE          ******** X    04
SS$_REJECT              ******** X    04
SS$_VCBROKEN            ******** X    04
TIMEOUT                 000003BB R    04
TQE$B_RQTYPE            = 0000000B
TQE$B_TYPE              = 0000000A
TQE$C_SSSNGL            = 00000001
TQE$K_LENGTH            = 00000030
TQE$L_FPC               = 0000000C
TQE$L_FR3               = 00000010
TQE$L_FR4               = 00000014
TQE$W_SIZE              = 00000008
```

Psect synopsis

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
ABS	00000000 (0.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
$$$040	00000010 (16.)	02 (2.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC LONG
$$$002	000000F0 (240.)	03 (3.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE
$$$100	000006B7 (1719.)	04 (4.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC LONG

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	32	00:00:00.04	00:00:02.20
Command processing	110	00:00:00.40	00:00:02.23
Pass 1	412	00:00:10.56	00:00:34.95
Symbol table sort	0	00:00:01.38	00:00:03.59
Pass 2	346	00:00:03.37	00:00:09.25
Symbol table output	25	00:00:00.13	00:00:00.34
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	929	00:00:15.90	00:00:52.58

The working set limit was 1800 pages.
91243 bytes (179 pages) of virtual memory were used to buffer the intermediate code.
There were 80 pages of symbol table space allocated to hold 1267 non-local and 113 local symbols.
2014 source lines were read in Pass 1, producing 26 object records in Pass 2.
34 pages of virtual memory were used to define 31 macros.

Macro library statistics

Macro library name	Macros defined
_255DUA28:[SYSLOA.OBJ]CLUSTER.MLB;1	3
_255DUA28:[SYS.OBJ]LIB.MLB;1	17
_255DUA28:[SYSLIB]STARLET.MLB;2	6
TOTALS (all libraries)	26

1394 GETS were required to define 26 macros.

There were no errors, warnings or information messages.

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

0392 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

CNXOPT
LIS

ADPSUBUV1
LIS

CNXMAN
LIS

CJFCLUSTR
LIS

CLUSTRLOA
LIS

CONMAN
LIS

ADPSUB790
LIS

CLUMESSAG
LIS