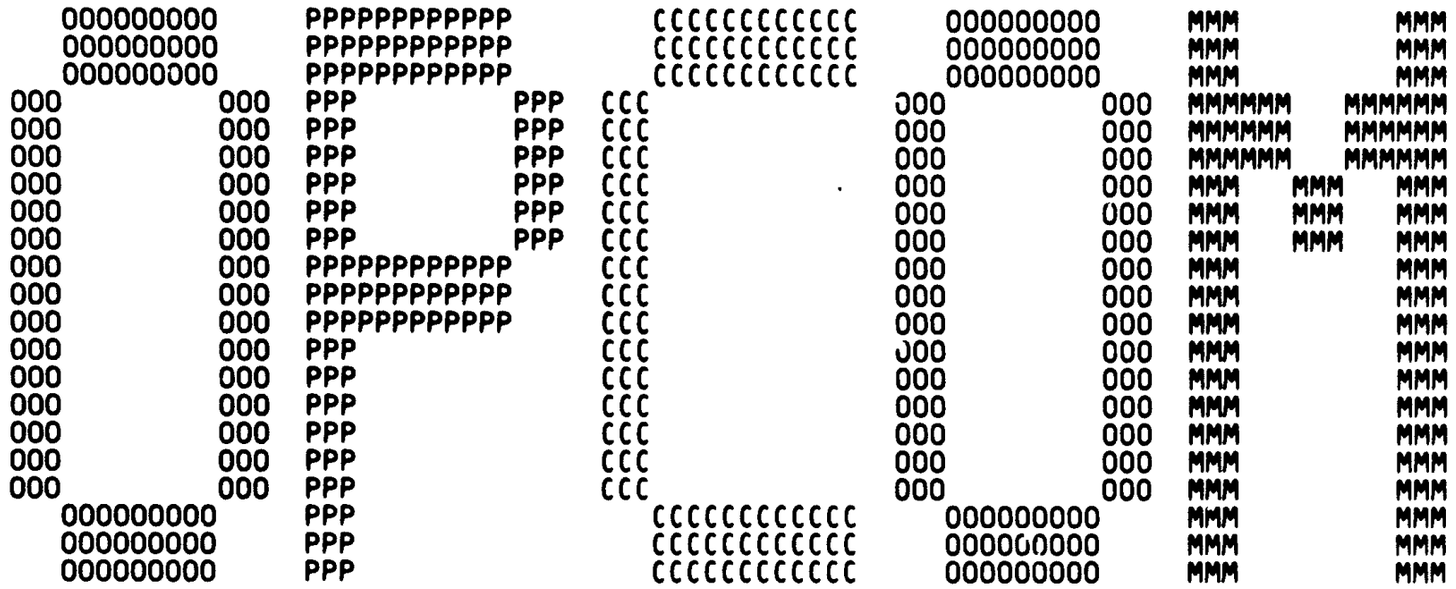

_$2

Sym

ALL

ASC

BOD
BOD
BOD
BOD
BOD
BOD
BOD
BOD
BUG
BYP
CAN
CAN
CAN
CHE
CHE

CLU
CLU
CLU
CLU
CLU
CLU
CLU

CLU
CLU
CLU
CLU
CLU
CLU
CLU
CLU

CLU
CLU

FILEID**CLUSUTIL

```
  CCCCCCCC  LL          UU      UU    SSSSSSSS  UU      UU  TTTTTTTTTT    IIIIII    LL
  CCCCCCCC  LL          UU      UU    SSSSSSSS  UU      UU  TTTTTTTT'T    IIIIII    LL
CC          LL          UU      UU  SS          UU      UU      TT          II      LL
CC          LL          UU      UU  SS          UU      UU      TT          II      LL
CC          LL          UU      UU  SS          UU      UU      TT          II      LL
CC          LL          UU      UU  SS          UU      UU      TT          II      LL
CC          LL          UU      UU    SSSSSS    UU      UU      TT          II      LL
CC          LL          UU      UU    SSSSSS    UU      UU      TT          II      LL
CC          LL          UU      UU          SS  UU      UU      TT          II      LL
CC          LL          UU      UU          SS  UU      UU      TT          II      LL
CC          LL          UU      UU          SS  UU      UU      TT          II      LL          ....
CC          LL          UU      UU          SS  UU      UU      TT          II      LL          ....
  CCCCCCCC  LLLLLLLLLL  UUUUUUUUUU  SSSSSSSS    UUUUUUUUUU      TT        IIIIII    LLLLLLLLLL  ....
  CCCCCCCC  LLLLLLLLLL  UUUUUUUUUU  SSSSSSSS    UUUUUUUUUU      TT        IIIIII    LLLLLLLLLL  ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
;     1          0001  0 MODULE  OPC$CLUSUTIL    (
;     2          0002  0                          LANGUAGE (BLISS32),
;     3          0003  0                          IDENT = 'V04-000'
;     4          0004  0                         ) =
;     5          0005  0
;     6          0006  0 !*****************************************************************************
;     7          0007  0 !*                                                                           *
;     8          0008  0 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
;     9          0009  0 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
;    10          0010  0 !*  ALL RIGHTS RESERVED.                                                     *
;    11          0011  0 !*                                                                           *
;    12          0012  0 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
;    13          0013  0 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
;    14          0014  0 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
;    15          0015  0 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
;    16          0016  0 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
;    17          0017  0 !*  TRANSFERRED.                                                             *
;    18          0018  0 !*                                                                           *
;    19          0019  0 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
;    20          0020  0 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
;    21          0021  0 !*  CORPORATION.                                                             *
;    22          0022  0 !*                                                                           *
;    23          0023  0 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
;    24          0024  0 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
;    25          0025  0 !*                                                                           *
;    26          0026  0 !*                                                                           *
;    27          0027  0 !*****************************************************************************
;    28          0028  0
;    29          0029  0 !++
;    30          0030  0 ! FACILITY:
;    31          0031  0 !
;    32          0032  0 !       OPCOM
;    33          0033  0 !
;    34          0034  0 ! ABSTRACT:
;    35          0035  0 !
;    36          0036  0 !       This module contains all the various and sundry general
;    37          0037  0 !       purpose utility routines used by cluster functions within OPCOM.
;    38          0038  0 !
;    39          0039  0 ! Environment:
;    40          0040  0 !
;    41          0041  0 !       VAX/VMS operating system.
;    42          0042  0 !
;    43          0043  0 ! Author:
;    44          0044  0 !
;    45          0045  0 !       CW Hobbs
;    46          0046  0 !
;    47          0047  0 ! Creation date:
;    48          0048  0 !
;    49          0049  0 !       8 July 1983
;    50          0050  0 !
;    51          0051  0 ! Revision history:
;    52          0052  0 !
;    53          0053  0 !       V03-004 CWH3004         CW Hobbs                        21-May-1984
;    54          0054  0 !               Allow wildcard $GETSYI to return SS$_NOSUCHNODE, as it will
;    55          0055  0 !               do this if a node disappears while $GETSYI is working on
;    56          0056  0 !               getting the info.
;    57          0057  0 !
```

```
;    58          0058  0 !       V03-003 CWH3169         CW Hobbs                        5-May-1984
;    59          0059  0 !               Second pass for cluster-wide OPCOM:
;    60          0060  0 !                - Change CLUSUTIL CONFIGURE to have a value - true if the
;    61          0061  0 !                  configuration changed, false if not.
;    62          0062  0 !                - Do not request ACK's when a node appears, wait for it to
;    63          0063  0 !                  ask us for the ACK.  This avoids sending a message to
;    64          0064  0 !                  a node before it is ready to listen.
;    65          0065  0 !                - Remove a check for NET0: being around, not necessary
;    66          0066  0 !                  now that CSP does not use decnet.
;    67          0067  0 !
;    68          0068  0 !       V03-002 CWH3002         CW Hobbs                16-Sep-1983
;    69          0069  0 !               Change error message for cluster errors
;    70          0070  0 !
;    71          0071  0 !
;    72          0072  0 !--
;    73          0073  0
;    74          0074  1 BEGIN                                   ! Start of CLUSUTIL
```

```
;    76          0075  1 LIBRARY 'SYS$LIBRARY:LIB.L32';
;    77          0076  1 LIBRARY 'LIB$:OPCOMLIB';
;    78          0077  1
;    79          0078  1 FORWARD ROUTINE
;    80          0079  1         CLUSUTIL_CONFIGURE,                     ! Reconfigure cluster systems
;    81          0080  1         CLUSUTIL_FIND_NOD_BY_CSID,              ! Find the NOD for a given CSID
;    82          0081  1         CLUSUTIL_FIND_NOD_BY_NAME,              ! Find the NOD for a given nodename
;    83          0082  1         CLUSUTIL_FIND_NOD_BY_SYSTEMID,          ! Find the NOD for a given SYSTEMID
;    84          0083  1         CLUSUTIL_INCR_SEQUENCE,                 ! Increment a sequence number, cluster unique
;    85          0084  1         CLUSUTIL_INIT           : NOVALUE,      ! Perform initialization functions related to clusters
;    86          0085  1         CLUSUTIL_NEXT_SEQUENCE,                 ! Increment global NEXT_SEQUENCE number, cluster unique
;    87          0086  1         CLUSUTIL_NODE_ACTIVATE  : NOVALUE,      ! Activate a node which has responded to our acknowledge req
;    88          0087  1         CLUSUTIL_NODE_INACTIVATE : NOVALUE,     ! Inactivate a node which has disappeared
;    89          0088  1         CLUSUTIL_NODE_MESSAGE   : NOVALUE,      ! Tell cluster operators about node changes
;    90          0089  1         CLUSUTIL_NODE_START     : NOVALUE,      ! Initialize a node to the START state
;    91          0090  1         CLUSUTIL_SYSTEMID_EQUAL : JSB_R0R1;     ! Compare SCS system ids, return equivalence
;    92          0091  1
;    93          0092  1 EXTERNAL ROUTINE
;    94          0093  1         ALLOCATE_DS,
;    95          0094  1         CLUSMSG_RQCB_SEND,                      ! Send an RQCB to remote nodes
;    96          0095  1         DEALLOCATE_DS,
;    97          0096  1         DEALLOCATE_RQCB : NOVALUE,              ! Dispose of an RQCB
;    98          0097  1         FORMAT_MESSAGE,                         ! Format a message
;    99          0098  1         LOG_MESSAGE,                            ! Log an event
;   100          0099  1         NOTIFY_LISTED_OPERATORS;                ! Notify interested operators
;   101          0100  1
;   102          0101  1 EXTERNAL LITERAL
;   103          0102  1         RQCB_K_TYPE,                            ! RQCB structure type
;   104          0103  1         MIN_SCOPE,                              ! Minimum scope value
;   105          0104  1         MAX_SCOPE,                              ! Maximum scope value
;   106          0105  1         NOD_K_TYPE;
;   107          0106  1
;   108          0107  1 EXTERNAL
;   109          0108  1         OCD_VECTOR      : VECTOR,               ! OCD list heads
;   110          0109  1         SEQ_WIDTH_DEF   : LONG,                 ! Width of node information when cluster is active
;   111          0110  1         SEQ_WIDTH       : LONG,                 ! Width of node information
;   112          0111  1         SEQ_SEED        : LONG,                 ! Some bits of local node info
;   113          0112  1         NEXT_SEQUENCE   : LONG,                 ! Next sequence number for data structures, etc
;   114          0113  1         GLOBAL_STATUS   : BITVECTOR [32],
;   115          0114  1         LCL_CSID        : LONG,
;   116          0115  1         LCL_NOD         : $ref_bblock,
;   117          0116  1         NOD_HEAD        : VECTOR [2, LONG];
;   118          0117  1
;   119          0118  1 BUILTIN
;   120          0119  1         INSQUE,
;   121          0120  1         REMQUE;
;   122          0121  1
;   123          0122  1 OWN
;   124          0123  1         NODE_CSID       : LONG,
;   125          0124  1         SYSTEMID        : VECTOR [6, BYTE],
;   126          0125  1         SWINCARN        : VECTOR [2, LONG],
;   127          0126  1         NAME_BUF        : VECTOR [16, BYTE],
;   128          0127  1         NAME_LEN        : LONG,
;   129          0128  1         CLUSTER_FLAG    : LONG,
;   130          0129  1         SYI_CSID        : VECTOR [4, LONG]      ! GETSYI list to get CSID and MEMBER items only
;   131          0130  1                         INITIAL  ((SYI$_NODE_CSID^16 + 4),
;   132          0131  1                                   NODE_CSID,
```

```
;  133          0132  1                                   0,
;  134          0133  1                                   0),
;  135          0134  1          SYI_NODE         : VECTOR [16, LONG]     ! GETSYI list
;  136          0135  1                           INITIAL  ((SYI$_NODE_CSID^16 + 4),
;  137          0136  1                                    NODE_CSID,
;  138          0137  1                                    0,
;  139          0138  1                                    (SYI$_CLUSTER_MEMBER^16 OR 4),
;  140          0139  1                                    CLUSTER_FLAG,
;  141          0140  1                                    0,
;  142          0141  1                                    (SYI$_NODE_SYSTEMID^16 + 6),
;  143          0142  1                                    SYSTEMID,
;  144          0143  1                                    0,
;  145          0144  1                                    (SYI$_NODE_SWINCARN^16 + 8),
;  146          0145  1                                    SWINCARN,
;  147          0146  1                                    0,
;  148          0147  1                                    (SYI$_NODENAME^16 + 16),
;  149          0148  1                                    NAME_BUF,
;  150          0149  1                                    NAME_LEN,
;  151          0150  1                                    0);
```

```
;  153          0151  1 GLOBAL ROUTINE CLUSUTIL_CONFIGURE =                %SBTTL 'clusutil_configure'
;  154          0152  1
;  155          0153  1 !++
;  156          0154  1 ! Functional description:
;  157          0155  1 !
;  158          0156  1 !       Compare cluster configuration database with reality, and make any adjustments
;  159          0157  1 !       which are necessary.
;  160          0158  1 !
;  161          0159  1 ! Input:
;  162          0160  1 !       None.
;  163          0161  1 !
;  164          0162  1 ! Implicit Input:
;  165          0163  1 !       None.
;  166          0164  1 !
;  167          0165  1 ! Output:
;  168          0166  1 !       None.
;  169          0167  1 !
;  170          0168  1 ! Implict output:
;  171          0169  1 !       Global data may be altered
;  172          0170  1 !
;  173          0171  1 ! Side effects:
;  174          0172  1 !       Messages will be sent to cluster operators if there are any changes.
;  175          0173  1 !
;  176          0174  1 ! Routine value:
;  177          0175  1 !       True if change in configuration, false otherwise
;  178          0176  1 !--
;  179          0177  1
;  180          0178  2 BEGIN                                               ! Start of CLUSUTIL_CONFIGURE
;  181          0179  2
;  182          0180  2 ROUTINE REMOVE_NODE (SCS_ID : REF VECTOR [3, WORD], QUEUE : REF VECTOR [2,LONG]) =
;  183          0181  3 BEGIN
;  184          0182  3 BUILTIN
;  185          0183  3     REMQUE;
;  186          0184  3 LOCAL
;  187          0185  3     PTR : $ref_bblock;
;  188          0186  3 !
;  189          0187  3 ! Loop through all the nodes on the queue, remove an entry if it matches the SYSTEMID
;  190          0188  3 !
;  191          0189  3 PTR = .QUEUE [0];
;  192          0190  3 WHILE .PTR NEQ QUEUE [0]
;  193          0191  3 DO
;  194          0192  4     BEGIN
;  195          0193  4     IF CLUSUTIL_SYSTEMID_EQUAL (.SCS_ID, PTR [NOD_T_NODE_SYSTEMID])
;  196          0194  4     THEN
;  197          0195  5         BEGIN
;  198          0196  5         REMQUE (.PTR, PTR);
;  199          0197  5         RETURN .PTR;
;  200          0198  4         END;
;  201          0199  4     PTR = .PTR [NOD_L_FLINK];
;  202          0200  3     END;
;  203          0201  3 RETURN 0;
;  204          0202  2 END;


                                                                .TITLE  OPC$CLUSUTIL
                                                                .IDENT  \V04-000\
```

```
                                        .PSECT  $OWN$,NOEXE,2

                                00000 NODE_CSID:
                                        .BLKB   4
                                00004 SYSTEMID:
                                        .BLKB   6
                                0000A   .BLKB   2
                                0000C SWINCARN:
                                        .BLKB   8
                                00014 NAME_BUF:
                                        .BLKB   16
                                00024 NAME_LEN:
                                        .BLKB   4
                                00028 CLUSTER_FLAG
                                        .BLKB   4
                    10D00004    0002C SYI_CSID:
                                        .LONG   282066948
                    00000000'   00030   .ADDRESS NODE_CSID
                00000000        00034   .LONG   0, 0
                    10D00004    0003C SYI_NODE:
                                        .LONG   282066948
                    00000000'   00040   .ADDRESS NODE_CSID
                10CF0004        00044   .LONG   0, 282001412
                    00000000'   0004C   .ADDRESS CLUSTER_FLAG
                10D30006        00050   .LONG   0, 282263558
                    00000000'   00058   .ADDRESS SYSTEMID
                10D40008        0005C   .LONG   0, 282329096
                    00000000'   00064   .ADDRESS SWINCARN
                10D90010        00068   .LONG   0, 282656784
            00000000'   00000000'   00070   .ADDRESS NAME_BUF, NAME_LEN
                    00000000    00078   .LONG   0

                                        .EXTRN  ALLOCATE_DS, CLUSMSG_RQCB_SEND
                                        .EXTRN  DEALLOCATE_DS, DEALLOCATE_RQCB
                                        .EXTRN  FORMAT_MESSAGE, LOG_MESSAGE
                                        .EXTRN  NOTIFY_LISTED_OPERATORS
                                        .EXTRN  RQCB_K_TYPE, MIN_SCOPE
                                        .EXTRN  MAX_SCOPE, NOD_K_TYPE
                                        .EXTRN  OCD_VECTOR, SEQ_WIDTH_DEF
                                        .EXTRN  SEQ_WIDTH, SEQ_SEED
                                        .EXTRN  NEXT_SEQUENCE, GLOBAL_STATUS
                                        .EXTRN  LCL_CSID, LCL_NOD
                                        .EXTRN  NOD_HEAD

                                        .PSECT  $CODE$,NOWRT,2

                                0004 00000 REMOVE_NODE:
                                        .WORD   Save R2                         ; 0180
        52      08      BC  D0  00002   MOVL    @QUEUE, PTR                     ; 0189
08      AC              52  D1  00006 1$:       CMPL    PTR, QUEUE              ; 0190
                        1A  13  0000A           BEQL    3$                      ;
        51      50      A2  9E  0000C           MOVAB   8(PTR), R1              ; 0193
        50      04      AC  D0  00010           MOVL    SCS_ID, R0              ;
                    0000V   30  00014           BSBW    CLUSUTIL_SYSTEMID_EQUAL ;
                07          50  E9  00017       BLBC    R0, 2$                  ;
                52          62  0F  0001A       REMQUE  (PTR), PTR              ; 0196
                50          52  D0  0001D       MOVL    PTR, R0                 ; 0197
```

```
                    04 00020        RET                              ;
      52       6?   D0 00021 2$:    MOVL    (PTR), PTR               ; 0199
               E0   11 00024        BRB     1$                       ; 0190
               50   D4 00026 3$:    CLRL    R0                       ; 0201
                    04 00028        RET                              ; 0202

; Routine Size:  41 bytes,    Routine Base:  $CODE$ + 0000
```

```
;  206          0203  2 LOCAL
;  207          0204  2         CHANGE,
;  208          0205  2         NOD             : $ref_bblock,          ! Local pointer
;  209          0206  2         WILD            : LONG,
;  210          0207  2         TEMP_Q          : VECTOR [2,LONG]
;  211          0208  2                         INITIAL (TEMP_Q, TEMP_Q),
;  212          0209  2         STATUS          : LONG;
;  213          0210  2
;  214          0211  2 CHANGE = FALSE;                                 ! Assume no change in the configuration
;  215          0212  2 !
;  216          0213  2 ! If not in a cluster we are done.
;  217          0214  2 !
;  218          0215  3 IF (NOT .GLOBAL_STATUS [GBLSTS_K_IN_VAXcluster])
;  219          0216  2 THEN
;  220          0217  2     RETURN .CHANGE;
;  221          0218  2 !
;  222          0219  2 ! Move all the node entries to our temporary queue, making sure that the nodes are still active
;  223          0220  2 !
;  224          0221  2 WHILE NOT REMQUE (.NOD_HEAD [0], NOD)
;  225          0222  2 DO
;  226          0223  3     BEGIN
;  227          0224  3     !
;  228          0225  3     ! Get cluster information for this node.  Looking for CSID is enough.
;  229          0226  3     !
;  230          0227  3     STATUS = $GETSYIW (CSIDADR=NOD [NOD_L_NODE_CSID], ITMLST=SYI_CSID);
;  231          0228  3     IF NOT .STATUS
;  232          0229  3     THEN
;  233          0230  4         BEGIN
;  234          0231  4         !
;  235          0232  4         ! Place the node in the "departed" state, and all that that entails
;  236          0233  4         !
;  237          0234  4         CLUSUTIL_NODE_INACTIVATE (.NOD);
;  238          0235  4         CHANGE = TRUE;
;  239          0236  3         END;
;  240          0237  3     !
;  241          0238  3     ! Put it on the temporary queue
;  242          0239  3     !
;  243          0240  3     INSQUE (.NOD, TEMP_Q);
;  244          0241  2     END;
;  245          0242  2 !
;  246          0243  2 ! Build a list of all the nodes in the cluster
;  247          0244  2 !
;  248          0245  2 WILD = -1;
;  249          0246  2 WHILE TRUE
;  250          0247  2 DO
;  251          0248  3     BEGIN
;  252          0249  3     !
;  253          0250  3     ! Get cluster information for wild nodes.  Loop until success, end, or
;  254          0251  3     ! serious failure.  $GETSYI will return NOSUCHNODE if a node happens to
;  255          0252  3     ! disappear while the $GETSYI call is processing the CSID.
;  256          0253  3     !
;  257          0254  3     WHILE TRUE
;  258          0255  3     DO
;  259          0256  4         BEGIN
;  260          0257  4         STATUS = $GETSYIW (CSIDADR=WILD, ITMLST=SYI_NODE);
;  261          0258  4         IF .STATUS EQL SS$_NOMORENODE           ! Found the end
;  262          0259  4           OR
```

```
;  263     0260  4            .STATUS                                 ! Found a live one
;  264     0261  4        THEN
;  265     0262  4            EXITLOOP;
;  266     0263  4        IF NOT .STATUS                              ! Ooops
;  267     0264  4        THEN
;  268     0265  4            IF .STATUS NEQ SS$_NOSUCHNODE           ! NOSUCHNODE is ok, try next
;  269     0266  4            THEN
;  270     0267  4                $signal_stop (.STATUS);
;  271     0268  3            END;
;  272     0269  3        IF .STATUS EQL SS$_NOMORENODE
;  273     0270  3        THEN
;  274     0271  3            EXITLOOP;
;  275     0272  3        !
;  276     0273  3        ! See if this node is in the temporary queue.  If so, it will be removed.
;  277     0274  3        ! Otherwise, 0 will be returned.
;  278     0275  3        !
;  279     0276  3        NOD = REMOVE_NODE (SYSTEMID, TEMP_Q);
;  280     C277  3        !
;  281     0278  3        ! If the node is 0, then we have a brand new node to add
;  282     0279  3        !
;  283     0280  3        IF .NOD EQL 0
;  284     0281  3        THEN
;  285     0282  4            BEGIN
;  286     0283  4            !
;  287     0284  4            ! Allocate and start the NOD
;  288     0285  4            !
;  289     0286  5            IF NOT (STATUS = ALLOCATE_DS (NOD_K_TYPE, NOD))
;  290     0287  4            THEN
;  291     0288  4                $signal_stop (.STATUS);
;  292     0289  4            NOD [NOD_B_STATE] = NOD_K_STATE_DEPARTED;       ! Pass through "departed" state briefly, the next
;  293     0290  3            END;                                            !  clause will move us to "started"
;  294     0291  3        !
;  295     0292  3        ! If the node is present but "departed", then start the node
;  296     0293  3        !
;  297     0294  3        IF .NOD [NOD_B_STATE] EQL NOD_K_STATE_DEPARTED
;  298     0295  3        THEN
;  299     0296  4            BEGIN
;  300     0297  4            CLUSUTIL_NODE_START (.NOD);
;  301     0298  4            CLUSUTIL_NODE_MESSAGE (.NOD, OPC$_NODE_START, FALSE);
;  302     0299  4            CHANGE = TRUE;
;  303     0300  3            END;
;  304     0301  3        !
;  305     0302  3        ! Put it back on the real queue
;  306     0303  3        !
;  307     0304  3        INSQUE (.NOD, NOD_HEAD);
;  308     0305  2        END;
;  309     0306  2
;  310     0307  2 !
;  311     0308  2 ! OK, now if there are any nodes left on the temporary queue, that means that
;  312     0309  2 ! those nodes are no longer with us.  (They vaporized while we were in the loop.)
;  313     0310  2 !
;  314     0311  2 WHILE NOT REMQUE (.TEMP_Q [0], NOD)
;  315     0312  2 DO
;  316     0313  3     BEGIN
;  317     0314  3     !
;  318     0315  3     ! Place the node in the "departed" state, and all that that entails
;  319     0316  3     !
```

```
;  320          0317  3      CLUSUTIL_NODE_INACTIVATE (.NOD);
;  321          0318  3      !
;  322          0319  3      ! Put it back on the real queue
;  323          0320  3      !
;  324          0321  3      INSQUE (.NOD, NOD_HEAD);
;  325          0322  3      CHANGE = TRUE;
;  326          0323  2      END;
;  327          0324  2
;  328          0325  2  RETURN .CHANGE;
;  329          0326  1  END;                                   ! End of CLUSUTIL_CONFIGURE


                                                                .EXTRN  SYS$GETSYIW, LIB$STOP

                                                    003C 00000  .ENTRY  CLUSUTIL_CONFIGURE, Save R2,R3,R4,R5        ; 0151
                                55 00000000G    00  9E 00002  MOVAB  SYS$GETSYIW, R5                             ;
                                5E              10  C2 00009  SUBL2  #16, SP                                    ;
                             08 AE        08    AE  9E 0000C  MOVAB  TEMP_Q, TEMP_Q                             ; 0178
                             0C AE        08    AE  9E 00011  MOVAB  TEMP_Q, TEMP_Q+4                           ;
                                                54  D4 00016  CLRL   CHANGE                                     ; 0211
                                03      0000G   CF  E8 00018  BLBS   GLOBAL_STATUS+1, 1$                        ; 0215
                                                    31 0001D  BRW    11$                                        ;
                             04 AE      0000G   DF  0F 00020  1$:    REMQUE @NOD_HEAD, NOD                      ; 0221
                                                2C  1D 00026  BVS    3$                                         ;
                                                7E  7C 00028  CLRQ   -(SP)                                      ; 0227
                                                7E  D4 0002A  CLRL   -(SP)                                      ;
                                        0000'   CF  9F 0002C  PUSHAB SYI_CSID                                   ;
                                                7E  D4 00030  CLRL   -(SP)                                      ;
                          7E    18 AE           2C  C1 00032  ADDL3  #44, NOD, -(SP)                            ;
                                                7E  D4 00037  CLRL   -(SP)                                      ;
                                65              07  FB 00039  CALLS  #7, SYS$GETSYIW                            ;
                                53              50  D0 0003C  MOVL   R0, STATUS                                ;
                                0B              53  E8 0003F  BLBS   STATUS, 2$                                 ; 0228
                                      04 AE     DD  00042     PUSHL  NOD                                         ; 0234
                          0000V CF              01  FB 00045  CALLS  #1, CLUSUTIL_NODE_INACTIVATE               ;
                                54              01  D0 0004A  MOVL   #1, CHANGE                                 ; 0235
                             08 AE      04      BE  0E 0004D  2$:    INSQUE @NOD, TEMP_Q                        ; 0240
                                                CC  11 00052  BRB    1$                                         ; 0221
                                6E              01  CE 00054  3$:    MNEGL  #1, WILD                            ; 0245
                                                7E  7C 00057  4$:    CLRQ   -(SP)                               ; 0257
                                                7E  D4 00059  CLRL   -(SP)                                      ;
                                        0000'   CF  9F 0005B  PUSHAB SYI_NODE                                   ;
                                                7E  D4 0005F  CLRL   -(SP)                                      ;
                                      14 AE     9F  00061     PUSHAB WILD                                        ;
                                                7E  D4 00064  CLRL   -(SP)                                      ;
                                65              07  FB 00066  CALLS  #7, SYS$GETSYIW                            ;
                                53              50  D0 00069  MOVL   R0, STATUS                                ;
                          00000A00 8F           53  D1 0006C  CMPL   STATUS, #2560                              ; 0258
                                                73  13 00073  BEQL   10$                                        ;
                                0B              53  E8 00075  BLBS   STATUS, 5$                                 ; 0260
                          0000028C 8F           53  D1 00078  CMPL   STATUS, #652                               ; 0265
                                                D6  13 0007F  BEQL   4$                                         ;
                                                28  11 00081  BRB    6$                                         ; 0267
                                                63  13 00083  5$:    BEQL   10$                                 ; 0269
                                      08 AE     9F  00085     PUSHAB TEMP_Q                                      ; 0276
                                        0000'   CF  9F 00088  PUSHAB SYSTEMID                                   ;
```

```
                FF46   CF            02  FB 0008C          CALLS   #2, REMOVE_NODE                    ;
                  04   AE            50  D0 00091          MOVL    R0, NOD                           ;
                                     26  12 00095          BNEQ    8$                                ; 0280
                              04     AE  9F 00097          PUSHAB  NOD                               ; 0286
                       00000000G     8F  DD 0009A          PUSHL   #NOD_K_TYPE                       ;
                0000G  CF            02  FB 000A0          CALLS   #2, ALLOCATE_DS                   ;
                       53            50  D0 000A5          MOVL    R0, STATUS                        ;
                       0A            53  E8 000A8          BLBS    STATUS, 7$                        ;
                                     53  DD 000AB 6$:      PUSHL   STATUS                            ; 0288
            00000000G  00            01  FB 000AD          CALLS   #1, LIB$STOP                      ;
                                     04  000B4             RET                                       ;
                       50      04    AE  D0 000B5 7$:      MOVL    NOD, R0                           ; 0289
                  22   A0            04  90 000B9          MOVB    #4, 34(R0)                        ;
                       52      04    AE  D0 000BD 8$:      MOVL    NOD, R2                           ; 0294
                       04      22    A2  91 000C1          CMPB    34(R2), #4                        ;
                                     19  12 000C5          BNEQ    9$                                ;
                                     52  DD 000C7          PUSHL   R2                                ; 0297
                0000V  CF            01  FB 000C9          CALLS   #1, CLUSUTIL_NODE_START           ;
                                     7E  D4 000CE          CLRL    -(SP)                             ; 0298
                          00058243   8F  DD 000D0          PUSHL   #361027                           ;
                                     52  DD 000D6          PUSHL   R2                                ;
                0000V  CF            03  FB 000D8          CALLS   #3, CLUSUTIL_NODE_MESSAGE         ;
                       54            01  D0 000DD          MOVL    #1, CHANGE                        ; 0299
                0000G  CF            62  0E 000E0 9$:      INSQUE  (R2), NOD_HEAD                    ; 0304
                              FF6F   31  000E5             BRW     4$                                ; 0246
                  04   AE      08    BE  0F 000E8 10$:     REMQUE  @TEMP_Q, NOD                      ; 0311
                                     13  1D 000ED          BVS     11$                               ;
                              04     AE  DD 000EF          PUSHL   NOD                               ; 0317
                0000V  CF            01  FB 000F2          CALLS   #1, CLUSUTIL_NODE_INACTIVATE      ;
                0000G  CF      04    BE  0E 000F7          INSQUE  @NOD, NOD_HEAD                    ; 0321
                       54            01  D0 000FD          MOVL    #1, CHANGE                        ; 0322
                                     E6  11 00100          BRB     10$                               ; 0311
                       50            54  D0 00102 11$:     MOVL    CHANGE, R0                        ; 0325
                                         04 00105          RET                                       ; 0326

; Routine Size:  262 bytes,    Routine Base:  $CODE$ + 0029
```

```
;  33'          0327  1 GLOBAL ROUTINE CLUSUTIL_FIND_NOD_BY_CSID (CSID) =                %SBTTL 'clusutil_find_nod_by_csid'
;  332          0328  1
;  333          0329  1 !++
;  334          0330  1 ! Functional description:
;  335          0331  1 !
;  336          0332  1 !       Find a cluster NOD block, given the CSID of the node.
;  337          0333  1 !
;  338          0334  1 ! Input:
;  339          0335  1 !
;  340          0336  1 !       CSID - Longword csid of system desired
;  341          0337  1 !
;  342          0338  1 ! Implicit Input:
;  343          0339  1 !
;  344          0340  1 !       None.
;  345          0341  1 !
;  346          0342  1 ! Output:
;  347          0343  1 !
;  348          0344  1 !       None.
;  349          0345  1 !
;  350          0346  1 ! Implict output:
;  351          0347  1 !
;  352          0348  1 !       None.
;  353          0349  1 !
;  354          0350  1 ! Side effects:
;  355          0351  1 !
;  356          0352  1 !       None.
;  357          0353  1 !
;  358          0354  1 ! Routine value:
;  359          0355  1 !
;  360          0356  1 !       Address of node block, or 0 if not found
;  361          0357  1 !--
;  362          0358  1
;  363          0359  2 BEGIN                                                   ! Start of CLUSUTIL_FIND_NOD_BY_CSID
;  364          0360  2
;  365          0361  2 LOCAL
;  366          0362  2     PTR : $ref_bblock;
;  367          0363  2
;  368          0364  2 !
;  369          0365  2 ! Loop through all the nodes on the queue, remove an entry if it matches the CSID
;  370          0366  2 !
;  371          0367  2 PTR = .NOD_HEAD [0];
;  372          0368  2 WHILE .PTR NEQ NOD_HEAD [0]
;  373          0369  2 DO
;  374          0370  3     BEGIN
;  375          0371  3     IF .PTR [NOD_L_NODE_CSID] EQL .CSID
;  376          0372  3     THEN
;  377          0373  3         RETURN .PTR;
;  378          0374  3     PTR = .PTR [NOD_L_FLINK];
;  379          0375  2     END;
;  380          0376  2
;  381          0377  2 RETURN 0;
;  382          0378  1 END;                                                    ! End of CLUSUTIL_FIND_NOD_BY_CSID
```

```
                                    0000 00000          .ENTRY  CLUSUTIL_FIND_NOD_BY_CSID, Save nothing     ; 0327
                          51    0000G CF   D0 00002           MOVL    NOD_HEAD, PTR                         ; 0367
                          50    0000G CF   9E 00007 1$:       MOVAB   NOD_HEAD, R0                          ; 0368
                          50          51   D1 0000C           CMPL    PTR, R0                               ;
                                      10   13 0000F           BEQL    3$                                    ;
                       04 AC      2C  A1   D1 00011           CMPL    44(PTR), CSID                         ; 0371
                                      04   12 00016           BNEQ    2$                                    ;
                          50          51   D0 00018           MOVL    PTR, R0                               ; 0373
                                      04   04 0001B           RET                                           ;
                          51          61   D0 0001C 2$:      MOVL    (PTR), PTR                            ; 0374
                                      E6   11 0001F           BRB     1$                                    ; 0368
                                      50   D4 00021 3$:      CLRL    R0                                    ; 0377
                                           04 00023           RET                                           ; 0378

; Routine Size:  36 bytes,    Routine Base:  $CODE$ + 012F
```

```
;  384          0379  1 GLOBAL ROUTINE CLUSUTIL_FIND_NOD_BY_NAME (NAME : $ref_bblock) =          %SBTTL 'clusutil_find_nod_by_name'
;  385          0380  1
;  386          0381  1 !++
;  387          0382  1 ! Functional description:
;  388          0383  1 !
;  389          0384  1 !       Find a cluster NOD block, given the nodename of the node.
;  390          0385  1 !
;  391          0386  1 ! Input:
;  392          0387  1 !
;  393          0388  1 !       NAME - Pointer to name descriptor
;  394          0389  1 !
;  395          0390  1 ! Implicit Input:
;  396          0391  1 !
;  397          0392  1 !       None.
;  398          0393  1 !
;  399          0394  1 ! Output:
;  400          0395  1 !
;  401          0396  1 !       None.
;  402          0397  1 !
;  403          0398  1 ! Implict output:
;  404          0399  1 !
;  405          0400  1 !       None.
;  406          0401  1 !
;  407          0402  1 ! Side effects:
;  408          0403  1 !
;  409          0404  1 !       None.
;  410          0405  1 !
;  411          0406  1 ! Routine value:
;  412          0407  1 !
;  413          0408  1 !       Address of node block, or 0 if not found
;  414          0409  1 !--
;  415          0410  1
;  416          0411  2 BEGIN                                           ! Start of CLUSUTIL_FIND_NOD_BY_NAME
;  417          0412  2
;  418          0413  2 LOCAL
;  419          0414  2     PTR : $ref_bblock;
;  420          0415  2
;  421          0416  2 !
;  422          0417  2 ! Loop through all the nodes on the queue, remove an entry if it matches the NAME
;  423          0418  2 !
;  424          0419  2 PTR = .NOD_HEAD [0];
;  425          0420  2 WHILE .PTR NEQ NOD_HEAD [0]
;  426          0421  2 DO
;  427          0422  3     BEGIN
;  428          0423  3     IF CH$EQL (.NAME [DSC$W_LENGTH], .NAME [DSC$A_POINTER], 0,
;  429          0424  3                .PTR [NOD_L_NAME_LEN], .PTR [NOD_L_NAME_PTR])
;  430          0425  3     THEN
;  431          0426  3         RETURN .PTR;
;  432          0427  3     PTR = .PTR [NOD_L_FLINK];
;  433          0428  2     END;
;  434          0429  2
;  435          0430  2 RETURN 0;
;  436          0431  1 END;                                            ! End of CLUSUTIL_FIND_NOD_BY_NAME
```

```
                                                003C 00000          .ENTRY  CLUSUTIL_FIND_NOD_BY_NAME, Save R2,R3,R4,R5 ; 0379
                                     54   0000G  CF   D0 00002      MOVL    NOD_HEAD, PTR                               ; 0419
                                     55     04   AC   D0 00007      MOVL    NAME, R5                                    ; 0423
                                     50   0000G  CF   9E 0000B 1$:   MOVAB   NOD_HEAD, R0                               ; 0420
                                     50          54   D1 00010      CMPL    PTR, R0                                     ;
                                                 15   13 00013      BEQL    3$                                          ;
          00        34    A4        04   B5     04   BC   2D 00015      CMPC5   @NAME, @4(R5), 52(PTR), #0, @48(PTR)     ; 0424
                                           30   B4        0001D                                                          ;
                                                 04   12 0001F      BNEQ    2$                                          ;
                                     50          54   D0 00021      MOVL    PTR, R0                                     ; 0426
                                                 04   00024      RET                                                     ;
                                     54          64   D0 00025 2$:   MOVL    (PTR), PTR                                  ; 0427
                                                 E1   11 00028      BRB     1$                                          ; 0420
                                                 50   D4 0002A 3$:   CLRL    R0                                          ; 0430
                                                      04 0002C      RET                                                 ; 0431

; Routine Size:  45 bytes,    Routine Base:  $CODE$ + 0153
```

```
;  438          0432  1 GLOBAL ROUTINE CLUSUTIL_FIND_NOD_BY_SYSTEMID (SYSTEMID : REF VECTOR [3,WORD]) =        %SBTTL 'clusutil_fin
;  439          0433  1
;  440          0434  1 !++
;  441          0435  1 ! Functional description:
;  442          0436  1 !
;  443          0437  1 !       Find a cluster NOD block, given the SYSTEMID of the node.
;  444          0438  1 !
;  445          0439  1 ! Input:
;  446          0440  1 !
;  447          0441  1 !       SYSTEMID - 48-bit id of system desired
;  448          0442  1 !
;  449          0443  1 ! Implicit Input:
;  450          0444  1 !
;  451          0445  1 !       None.
;  452          0446  1 !
;  453          0447  1 ! Output:
;  454          0448  1 !
;  455          0449  1 !       None.
;  456          0450  1 !
;  457          0451  1 ! Implict output:
;  458          0452  1 !
;  459          0453  1 !       None.
;  460          0454  1 !
;  461          0455  1 ! Side effects:
;  462          0456  1 !
;  463          0457  1 !       None.
;  464          0458  1 !
;  465          0459  1 ! Routine value:
;  466          0460  1 !
;  467          0461  1 !       Address of node block, or 0 if not found
;  468          0462  1 !--
;  469          0463  1
;  470          0464  2 BEGIN                                           ! Start of CLUSUTIL_FIND_NOD_BY_SYSTEMID
;  471          0465  2
;  472          0466  2 LOCAL
;  473          0467  2     PTR : $ref_bblock;
;  474          0468  2
;  475          0469  2 !
;  476          0470  2 ! Loop through all the nodes on the queue, remove an entry if it matches the SYSTEMID
;  477          0471  2 !
;  478          0472  2 PTR = .NOD_HEAD [0];
;  479          0473  2 WHILE .PTR NEQ NOD_HEAD [0]
;  480          0474  2 DO
;  481          0475  3     BEGIN
;  482          0476  3     IF CLUSUTIL_SYSTEMID_EQUAL (PTR [NOD_T_NODE_SYSTEMID], .SYSTEMID)
;  483          0477  3     THEN
;  484          0478  3         RETURN .PTR;
;  485          0479  3     PTR = .PTR [NOD_L_FLINK];
;  486          0480  2     END;
;  487          0481  2
;  488          0482  2 RETURN 0;
;  489          0483  1 END;                                            ! End of CLUSUTIL_FIND_NOD_BY_CSID
```

```
                                0004 00000          .ENTRY  CLUSUTIL_FIND_NOD_BY_SYSTEMID, Save R2      ; 0432
                    52  0000G  CF   D0 00002          MOVL    NOD_HEAD, PTR                               ; 0472
                    50  0000G  CF   9E 00007 1$:      MOVAB   NOD_HEAD, R0                                ; 0473
                    50          52   D1 0000C          CMPL    PTR, R0                                     ;
                                17   13 0000F          BEQL    3$                                          ;
                    50    50    A2   9E 00011          MOVAB   80(PTR), R0                                 ; 0476
                    51    04    AC   D0 00015          MOVL    SYSTEMID, R1                                ;
                            0000V 30   00019          BSBW    CLUSUTIL_SYSTEMID_EQUAL                     ;
                    04          50   E9 0001C          BLBC    R0, 2$                                      ;
                    50          52   D0 0001F          MOVL    PTR, R0                                     ; 0478
                                04   00022          RET                                                 ;
                    52          62   D0 00023 2$:      MOVL    (PTR), PTR                                  ; 0479
                                DF   11 00026          BRB     1$                                          ; 0473
                                50   D4 00028 3$:      CLRL    R0                                          ; 0482
                                     04 0002A          RET                                                 ; 0483

; Routine Size:  43 bytes,    Routine Base:  $CODE$ + 0180
```

```
;  491          0484  1 GLOBAL ROUTINE CLUSUTIL_INCR_SEQUENCE (OLD_SEQ) =
;  492          0485  1
;  493          0486  1 !++
;  494          0487  1 ! Functional description:
;  495          0488  1 !
;  496          0489  1 !       Take the number passed as input, return the number incremented with a cluster
;  497          0490  1 !       unique sequence number.
;  498          0491  1 !
;  499          0492  1 ! Input:
;  500          0493  1 !
;  501          0494  1 !       OLD_SEQ : Longword sequence number to be incremented
;  502          0495  1 !
;  503          0496  1 ! Implicit Input:
;  504          0497  1 !
;  505          0498  1 !       SEQ_WIDTH     : Width of node information field in sequence number
;  506          0499  1 !
;  507          0500  1 ! Output:
;  508          0501  1 !
;  509          0502  1 !       None.
;  510          0503  1 !
;  511          0504  1 ! Implict output:
;  512          0505  1 !
;  513          0506  1 !       None.
;  514          0507  1 !
;  515          0508  1 ! Side effects:
;  516          0509  1 !
;  517          0510  1 !       None.
;  518          0511  1 !
;  519          0512  1 ! Routine value:
;  520          0513  1 !
;  521          0514  1 !       Incremented sequence number
;  522          0515  1 !--
;  523          0516  1
;  524          0517  2 BEGIN                                           ! Start of CLUSUTIL_INCR_SEQUENCE
;  525          0518  2
;  526          0519  2
;  527          0520  2 REGISTER
;  528          0521  2         NEW_SEQ         : LONG;
;  529          0522  2
;  530          0523  2 !
;  531          0524  2 ! First, extract the cardinal number (high bits) from the sequence number.  Increment the value.
;  532          0525  2 ! (SEQ_WIDTH will be zero if not in a cluster)
;  533          0526  2 !
;  534          0527  2 NEW_SEQ = .OLD_SEQ<.SEQ_WIDTH,32-.SEQ_WIDTH,0> + 1;
;  535          0528  2 !
;  536          0529  2 ! Now, move the cardinal number over to the left and stick the fixed node
;  537          0530  2 ! identifier into the low bits  (SEQ_WIDTH will be zero if not in a cluster)
;  538          0531  2 NEW_SEQ = (.NEW_SEQ^.SEQ_WIDTH) + .SEQ_SEED;
;  539          0532  2 !
;  540          0533  2 ! Return the updated value
;  541          0534  2 !
;  542          0535  2 RETURN .NEW_SEQ;
;  543          0536  1 END;                                           ! End of CLUSUTIL_INCR_SEQUENCE
```

```
                                                0004 00000          .ENTRY  CLUSUTIL_INCR_SEQUENCE, Save R2         ; 0484
                                52      0000G   CF   D0 00002          MOVL    SEQ_WIDTH, R2                   ; 0527
                            50  20              52   C3 00007          SUBL3   R2, #32, R0                     ;
                51      04  AC  50              52   EF 0000B          EXTZV   R2, R0, OLD_SEQ, R1             ;
                                50      01      A1   9E 00011          MOVAB   1(R1), NEW_SEQ                  ;
                            51  50              52   78 00015          ASHL    R2, NEW_SEQ, R1                 ; 0531
                            50  51      0000G   CF   C1 00019          ADDL3   SEQ_SEED, R1, NEW_SEQ           ;
                                                04   04 0001F          RET                                     ; 0536

; Routine Size:  32 bytes,    Routine Base:  $CODE$ + 01AB
```

```
;  545          0537  1 GLOBAL ROUTINE CLUSUTIL_INIT : NOVALUE =                %SBTTL 'clusutil_init'
;  546          0538  1
;  547          0539  1 !++
;  548          0540  1 ! Functional description:
;  549          0541  1 !
;  550          0542  1 !       Perform process initialization activities related to cluster participation.
;  551          0543  1 !
;  552          0544  1 ! Input:
;  553          0545  1 !
;  554          0546  1 !       None.
;  555          0547  1 !
;  556          0548  1 ! Implicit Input:
;  557          0549  1 !
;  558          0550  1 !       None.
;  559          0551  1 !
;  560          0552  1 ! Output:
;  561          0553  1 !
;  562          0554  1 !       None.
;  563          0555  1 !
;  564          0556  1 ! Implict output:
;  565          0557  1 !
;  566          0558  1 !       Global data is initialized.
;  567          0559  1 !
;  568          0560  1 ! Side effects:
;  569          0561  1 !
;  570          0562  1 !       We will know if we are in a cluster, and if so, we will be ready to
;  571          0563  1 !       participate in cluster activities.
;  572          0564  1 !
;  573          0565  1 ! Routine value:
;  574          0566  1 !
;  575          0567  1 !       None.
;  576          0568  1 !--
;  577          0569  1
;  578          0570  2 BEGIN                                           ! Start of CLUSUTIL_INIT
;  579          0571  2
;  580          0572  2 LOCAL
;  581          0573  2         NOD     : $ref_bblock,
;  582          0574  2         STATUS  : LONG;
;  583          0575  2
;  584          0576  2 !
;  585          0577  2 ! If we are already in a cluster, leave without doing any more
;  586          0578  2 !
;  587          0579  2 IF .GLOBAL_STATUS [GBLSTS_K_IN_VAXcluster]
;  588          0580  2 THEN
;  589          0581  2     RETURN;
;  590          0582  2 !
;  591          0583  2 ! Get system information to see if we are in a cluster.
;  592          0584  2 ! Failure is fatal (there is no system?).
;  593          0585  2 !
;  594          0586  3 IF NOT (STATUS = $GETSYIW (ITMLST=SYI_NODE))
;  595          0587  2 THEN
;  596          0588  2     $signal_stop (.STATUS);
;  597          0589  2 !
;  598          0590  2 ! Save the membership flag
;  599          0591  2 !
;  600          0592  3 IF NOT (GLOBAL_STATUS [GBLSTS_K_IN_VAXcluster] = .CLUSTER_FLAG)
;  601          0593  2 THEN
```

```
;  602          0594  2     RETURN;
;  603          0595  2 !
;  604          0596  2 ! Save the CSID and the sequence number seed before we allocate data
;  605          0597  2 ! structures.  The default sequence width is held by the global SEQ_WIDTH_DEF
;  606          0598  2 ! to make it possible to increase the size of the cluster supported with a simple
;  607          0599  2 ! PATCH.  This helps balance the friendliness of having small request numbers against
;  608          0600  2 ! the need to be able to support larger clusters in the future.
;  609          0601  2 !
;  610          0602  2 LCL_CSID = .NODE_CSID;
;  611          0603  2 SEQ_WIDTH = .SEQ_WIDTH_DEF;
;  612          0604  2 SEQ_SEED = ((.NODE_CSID<16,2,0>)^(.SEQ_WIDTH_DEF-2)) + .NODE_CSID<0,.SEQ_WIDTH_DEF-2,0>;
;  613          0605  2 !
;  614          0606  2 ! Allocate and initialize the NOD, and add it to the list of nodes, also make
;  615          0607  2 ! it the local node
;  616          0608  2 !
;  617          0609  3 IF NOT (STATUS = ALLOCATE_DS (NOD_K_TYPE, NOD))
;  618          0610  2 THEN
;  619          0611  2     $signal_stop (.STATUS);
;  620          0612  2 CLUSUTIL_NODE_START (.NOD);
;  621          0613  2 NOD [NOD_B_STATE] = NOD_K_STATE_LOCAL;
;  622          0614  2 INSQUE (.NOD, NOD_HEAD);
;  623          0615  2 LCL_NOD = .NOD;
;  624          0616  2
;  625          0617  2 RETURN;
;  626          0618  1 END;                                    ! End of CLUSUTIL_INIT


                                                001C 00000          .ENTRY  CLUSUTIL_INIT, Save R2,R3,R4                ; 0537
                                    54      0000' CF   9E 00002     MOVAB   NODE_CSID, R4                               ;
                                    5E            04   C2 00007     SUBL2   #4, SP                                      ;
                                    7C      0000G CF   E8 0000A     BLBS    GLOBAL_STATUS+1, 3$                         ; 0579
                                                  7E   7C 0000F     CLRQ    -(SP)                                       ; 0586
                                                  7E   D4 00011     CLRL    -(SP)                                       ;
                                      3C          A4   9F 00013     PUSHAB  SYI_NODE                                    ;
                                                  7E   7C 00016     CLRQ    -(SP)                                       ;
                                                  7E   D4 00018     CLRL    -(SP)                                       ;
                                00000000G         00   FB 0001A     CALLS   #7, SYS$GETSYIW                             ;
                                                  50   E9 00021     BLBC    STATUS, 1$                                  ;
                                    51            28   A4 00024     MOVL    CLUSTER_FLAG, R1                            ; 0592
0000G   CF          01                00   F0 00028     INSV    R1, #0, #1, GLOBAL_STATUS+1                         ;
                                                  51   E9 0002F     BLBC    R1, 3$                                      ;
                                0000G   CF        64   D0 00032     MOVL    NODE_CSID, LCL_CSID                         ; 0602
                                0000G   CF  0000G CF   D0 00037     MOVL    SEQ_WIDTH_DEF, SEQ_WIDTH                    ; 0603
                            52  0000G   CF        02   C3 0003E     SUBL3   #2, SEQ_WIDTH_DEF, R2                       ; 0604
        53          02      A4                00   EF 00044     EXTZV   #0, #2, NODE_CSID+2, R3                     ;
                            53                53   78 0004A     ASHL    R2, R3, R3                                  ;
        51                  64                52   00 EF 0004E EXTZV   #0, R2, NODE_CSID, R1                       ;
                0000G       CF                53   C1 00053     ADDL3   R1, R3, SEQ_SEED                            ;
                                                  5E   DD 00059     PUSHL   SP                                          ; 0609
                                    00000000G     8F   DD 0005B     PUSHL   #NOD_K_TYPE                                 ;
                                0000G   CF        02   FB 00061     CALLS   #2, ALLOCATE_DS                             ;
                                                  50   E8 00066     BLBS    STATUS, 2$                                  ;
                                                  50   DD 00069 1$: PUSHL   STATUS                                      ; 0611
                                00000000G         00   FB 0006B     CALLS   #1, LIB$STOP                                ;
```

```
                                04 00072         RET                                    ;
              52             6E DO 00073 2$:     MOVL    NOD, R2                         ; 0612
                             52 DD 00076         PUSHL   R2                              ;
       0000V  CF             01 FB 00078         CALLS   #1, CLUSUTIL_NODE_START         ;
          22  A2             01 90 0007D         MOVB    #1, 34(R2)                      ; 0613
       0000G  CF             62 0E 00081         INSQUE  (R2), NOD_HEAD                  ; 0614
       0000G  CF             6E D0 00086         MOVL    NOD, LCL_NOD                    ; 0615
                                04 0008B 3$:     RET                                     ; 0618

; Routine Size:  140 bytes,    Routine Base:  $CODE$ + 01CB
```

```
;  628          0619  1 GLOBAL ROUTINE CLUSUTIL_NEXT_SEQUENCE =
;  629          0620  1
;  630          0621  1 !++
;  631          0622  1 ! Functional description:
;  632          0623  1 !
;  633          0624  1 !       Increment and return the global variable NEXT_SEQUENCE.
;  634          0625  1 !
;  635          0626  1 ! Input:
;  636          0627  1 !
;  637          0628  1 !       None.
;  638          0629  1 !
;  639          0630  1 ! Implicit Input:
;  640          0631  1 !
;  641          0632  1 !       None.
;  642          0633  1 !
;  643          0634  1 ! Output:
;  644          0635  1 !
;  645          0636  1 !       None.
;  646          0637  1 !
;  647          0638  1 ! Implict output:
;  648          0639  1 !
;  649          0640  1 !       Global cell NEXT_SEQUENCE is incremented.
;  650          0641  1 !
;  651          0642  1 ! Side effects:
;  652          0643  1 !
;  653          0644  1 !       None.
;  654          0645  1 !
;  655          0646  1 ! Routine value:
;  656          0647  1 !
;  657          0648  1 !       Incremented sequence number
;  658          0649  1 !--
;  659          0650  1
;  660          0651  2 BEGIN                                           ! Start of CLUSUTIL_NEXT_SEQUENCE
;  661          0652  2
;  662          0653  2 REGISTER
;  663          0654  2     SEQ : LONG;
;  664          0655  2 !
;  665          0656  2 ! Get, store and return the updated value
;  666          0657  2 !
;  667          0658  2 SEQ = CLUSUTIL_INCR_SEQUENCE (.NEXT_SEQUENCE);
;  668          0659  2 NEXT_SEQUENCE = .SEQ;
;  669          0660  2
;  670          0661  2 RETURN .SEQ;
;  671          0662  1 END;                                            ! End of CLUSUTIL_NEXT_SEQUENCE


                                        0000 00000          .ENTRY  CLUSUTIL_NEXT_SEQUENCE, Save nothing    ; 0619
                                 0000G  CF   DD 00002       PUSHL   NEXT_SEQUENCE                           ; 0658
                          FF49   CF         01   FB 00006   CALLS   #1, CLUSUTIL_INCR_SEQUENCE              ;
                          0000G  CF         50   D0 0000B   MOVL    SEQ, NEXT_SEQUENCE                      ; 0659
                                            04   00010     RET                                             ; 0662

; Routine Size:  17 bytes,    Routine Base:  $CODE$ + 0257
```

```
;  673          0663  1 GLOBAL ROUTINE CLUSUTIL_NODE_ACTIVATE (NOD : $ref_bblock) : NOVALUE =          %SBTTL 'CLUSUTIL_NODE_activa
;  674          0664  1
;  675          0665  1 !++
;  676          0666  1 ! Functional description:
;  677          0667  1 !
;  678          0668  1 !       Place a NOD into ACTIVE state.
;  679          0669  1 !
;  680          0670  1 ! Input:
;  681          0671  1 !
;  682          0672  1 !       None.
;  683          0673  1 !
;  684          0674  1 ! Implicit Input:
;  685          0675  1 !
;  686          0676  1 !       None.
;  687          0677  1 !
;  688          0678  1 ! Output:
;  689          0679  1 !
;  690          0680  1 !       None.
;  691          0681  1 !
;  692          0682  1 ! Implict output:
;  693          0683  1 !
;  694          0684  1 !       Global data may be altered
;  695          0685  1 !
;  696          0686  1 ! Side effects:
;  697          0687  1 !
;  698          0688  1 !       Messages will be sent to cluster operators if there are any changes.
;  699          0689  1 !
;  700          0690  1 ! Routine value:
;  701          0691  1 !
;  702          0692  1 !       None.
;  703          0693  1 !--
;  704          0694  1
;  705          0695  2 BEGIN                                                   ! Start of CLUSUTIL_NODE_ACTIVATE
;  706          0696  2
;  707          0697  2 !
;  708          0698  2 ! If the node is already active, return
;  709          0699  2 !
;  710          0700  2 IF .NOD [NOD_B_STATE] EQL NOD_K_STATE_ACTIVE
;  711          0701  2 THEN
;  712          0702  2     RETURN;
;  713          0703  2 !
;  714          0704  2 ! Set the state of the node to active
;  715          0705  2 !
;  716          0706  2 NOD [NOD_B_STATE] = NOD_K_STATE_ACTIVE;
;  717          0707  2 NOD [NOD_V_ACK_PEND] = FALSE;
;  718          0708  2 !
;  719          0709  2 ! Tell cluster operators that we have activated this node
;  720          0710  2 !
;  721          0711  2 CLUSUTIL_NODE_MESSAGE (.NOD, OPC$_NODE_ACTIVE, FALSE);
;  722          0712  2
;  723          0713  2 RETURN;
;  724          0714  1 END;                                                   ! End of CLUSUTIL_NODE_ACTIVATE
```

```
                                    0000 00000          .ENTRY  CLUSUTIL_NODE_ACTIVATE, Save nothing        ; 0663
                          50    04  AC  D0 00002          MOVL    NOD, R0                                     ; 0700
                          03    22  A0  91 00006          CMPB    34(R0), #3                                  ;
                                    17  13 0000A          BEQL    1$                                          ;
                       22   A0       03  90 0000C          MOVB    #3, 34(R0)                                  ; 0706
                       2A   A0       01  8A 00010          BICB2   #1, 42(R0)                                  ; 0707
                                    7E  D4 00014          CLRL    -(SP)                                       ; 0711
                            0005821B 8F  DD 00016          PUSHL   #360987                                     ;
                                    50  DD 0001C          PUSHL   R0                                          ;
                     0000V  CF      03  FB 0001E          CALLS   #3, CLUSUTIL_NODE_MESSAGE                   ;
                                        04 00023 1$:      RET                                                 ; 0714

; Routine Size:  36 bytes,    Routine Base:  $CODE$ + 0268
```

```
;  726     0715  1 GLOBAL ROUTINE CLUSUTIL_NODE_INACTIVATE (NOD : $ref_bblock) : NOVALUE =          %SBTTL 'CLUSUTIL_NODE_INacti
;  727     0716  1
;  728     0717  1 !++
;  729     0718  1 ! Functional description:
;  730     0719  1 !
;  731     0720  1 !       Place a NOD into "departed" state.
;  732     0721  1 !
;  733     0722  1 ! Input:
;  734     0723  1 !
;  735     0724  1 !       None.
;  736     0725  1 !
;  737     0726  1 ! Implicit Input:
;  738     0727  1 !
;  739     0728  1 !       None.
;  740     0729  1 !
;  741     0730  1 ! Output:
;  742     0731  1 !
;  743     0732  1 !       None.
;  744     0733  1 !
;  745     0734  1 ! Implict output:
;  746     0735  1 !
;  747     0736  1 !       Global data may be altered
;  748     0737  1 !
;  749     0738  1 ! Side effects:
;  750     0739  1 !
;  751     0740  1 !       Messages will be sent to cluster operators if there are any changes.
;  752     0741  1 !
;  753     0742  1 ! Routine value:
;  754     0743  1 !
;  755     0744  1 !       None.
;  756     0745  1 !--
;  757     0746  1
;  758     0747  2 BEGIN                                           ! Start of CLUSUTIL_NODE_INACTIVATE
;  759     0748  2
;  760     0749  2 LOCAL
;  761     0750  2     OCD_INDEX,
;  762     0751  2     OCD_COUNT,
;  763     0752  2     OCD         : $ref_bblock,
;  764     0753  2     RQST_RQCB   : $ref_bblock;
;  765     0754  2 !
;  766     0755  2 ! If the node is already "departed", return
;  767     0756  2 !
;  768     0757  2 IF .NOD [NOD_B_STATE] EQL NOD_K_STATE_DEPARTED
;  769     0758  2 THEN
;  770     0759  2     RETURN;
;  771     0760  2 !
;  772     0761  2 ! Set the state of the node to "departed"
;  773     0762  2 !
;  774     0763  2 NOD [NOD_B_STATE] = NOD_K_STATE_DEPARTED;
;  775     0764  2 !
;  776     0765  2 ! Tell cluster operators that we have removed this node
;  777     0766  2 !
;  778     0767  2 CLUSUTIL_NODE_MESSAGE (.NOD, OPC$_NODE_DEPARTED, FALSE);
;  779     0768  2 !
;  780     0769  2 ! Search the entire database for requests owned by the disappearing node.
;  781     0770  2 !
;  782     0771  2 OCD_INDEX = MAX_SCOPE;
```

```
;  783          0772  2 WHILE .OCD_INDEX GEQ MIN_SCOPE
;  784          0773  2 DO
;  785          0774  3     BEGIN
;  786          0775  3     !
;  787          0776  3     ! Scan the OCD list for each class of operator
;  788          0777  3     !
;  789          0778  3     OCD = .OCD_VECTOR [(.OCD_INDEX - 1) * 2];                   ! Get first OCD address
;  790          0779  3     OCD_COUNT = .OCD_VECTOR [(.OCD_INDEX - 1) * 2 + 1];         ! Get # of OCDs in the list
;  791          0780  3     WHILE .OCD_COUNT GTR 0
;  792          0781  3     DO
;  793          0782  4         BEGIN
;  794          0783  4         !
;  795          0784  4         ! Scan the request list for each OCD.
;  796          0785  4         !
;  797          0786  4         RQST_RQCB  = .OCD [OCD_L_RQSTFLINK];                    ! Get first RQST_RQCB address
;  798          0787  4         WHILE .RQST_RQCB NEQ OCD [OCD_L_RQSTFLINK]
;  799          0788  4         DO
;  800          0789  5             BEGIN
;  801          0790  5             !
;  802          0791  5             ! If the ID matches the disappearing node, cancel the request
;  803          0792  5             !
;  804          0793  5             IF CLUSUTIL_SYSTEMID_EQUAL (RQST_RQCB [RQCB_T_SYSTEMID], NOD [NOD_T_NODE_SYSTEMID])
;  805          0794  5             THEN
;  806          0795  6                 BEGIN
;  807          0796  6                 LOCAL
;  808          0797  6                     MESSAGE_VECTOR : VECTOR [3, LONG],
;  809          0798  6                     RQCB;
;  810          0799  6                 !
;  811          0800  6                 ! Inform all interested operators that the request is canceled.
;  812          0801  6                 ! Log the cancelation notice, and remove the request from the data base.
;  813          0802  6                 !
;  814          0803  6                 MESSAGE_VECTOR [0] = OPC$_RQSTCAN;          ! Set message code
;  815          0804  6                 MESSAGE_VECTOR [1] = 0;                     ! Set # of message arguements
;  816          0805  6                 MESSAGE_VECTOR [2] = .RQST_RQCB [RQCB_L_RQSTNUM];! Set message arguement
;  817          0806  6                 REMQUE (.RQST_RQCB, RQST_RQCB);             ! Remove the request from the database
;  818          0807  6                 OCD [OCD_W_RQSTCOUNT] = .OCD [OCD_W_RQSTCOUNT] - 1;
;  819          0808  6                 FORMAT_MESSAGE (.RQST_RQCB, MESSAGE_VECTOR);
;  820          0809  6                 !
;  821          0810  6                 ! Inform all interested operators that the request is canceled.  Log the cancelation
;  822          0811  6                 ! notice.  No need to inform other nodes, they will be running in parallel with us.
;  823          0812  6                 !
;  824          0813  6                 NOTIFY_LISTED_OPERATORS (.RQST_RQCB);       ! Notify the interested operators
;  825          0814  6                 LOG_MESSAGE (.RQST_RQCB);                   ! Log the event
;  826          0815  6                 RQCB = .RQST_RQCB;                          ! Save the RQCB
;  827          0816  6                 RQST_RQCB = .RQST_RQCB [RQCB_L_FLINK];      ! Get address of next RQCB
;  828          0817  6                 DEALLOCATE_RQCB (.RQCB);                    ! Free the RQCB
;  829          0818  6                 END
;  830          0819  6             !
;  831          0820  6             ! Request doesn't belong to disappearing node, move to next request
;  832          0821  6             !
;  833          0822  5             ELSE
;  834          0823  5                 RQST_RQCB = .RQST_RQCB [RQCB_L_FLINK];      ! Get address of next RQCB
;  835          0824  4             END;
;  836          0825  4         OCD_COUNT = .OCD_COUNT - 1;                         ! Decrement OCD count
;  837          0826  4         OCD = .OCD [OCD_L_FLINK];                           ! Get address of next OCD
;  838          0827  3         END;
;  839          0828  3     OCD_INDEX = .OCD_INDEX - 1;                             ! Try next operator class
```

```
;  840          0829  2      END;
;  841          0830  2
;  842          0831  2 RETURN;
;  843          0832  1 END;

                                                . End of CLUSUTIL_NODE_INACTIVATE


                                    003C 00000          .ENTRY  CLUSUTIL_NODE_INACTIVATE, Save R2,R3,R4,R5 ; 0715
                                5E          0C  C2 00002        SUBL2   #12, SP                                 ;
                                50      04  AC  D0 00005        MOVL    NOD, R0                                 ; 0757
                                04      22  A0  91 00009        CMPB    34(R0), #4                              ;
                                            01  12 0000D        BNEQ    1$                                      ;
                                                04 0000F        RET                                             ;
                            22  A0          04  90 00010 1$:    MOVB    #4, 34(R0)                              ; 0763
                                            7E  D4 00014        CLRL    -(SP)                                   ; 0767
                                0005822B    8F  DD 00016        PUSHL   #361003                                 ;
                                            50  DD 0001C        PUSHL   R0                                      ;
                          0000V CF          03  FB 0001E        CALLS   #3, CLUSUTIL_NODE_MESSAGE               ;
                                53 00000000G  8F  D0 00023        MOVL    #MAX_SCOPE, OCD_INDEX                   ; 0771
                    00000000G   8F          53  D1 0002A 2$:    CMPL    OCD_INDEX, #MIN_SCOPE                   ; 0772
                                            01  18 00031        BGEQ    3$                                      ;
                                                04 00033        RET                                             ;
                50              53          01  78 00034 3$:    ASHL    #1, OCD_INDEX, R0                       ; 0776
                                52  0000GCF40   D0 00038        MOVL    OCD_VECTOR-8[R0], OCD                   ;
                                55  0000GCF40   D0 0003E        MOVL    OCD_VECTOR-4[R0], OCD_COUNT             ; 0779
                                            55  D5 00044 4$:    TSTL    OCD_COUNT                               ; 0780
                                            67  15 00046        BLEQ    8$                                      ;
                                54      3C  A2  D0 00048        MOVL    60(OCD), RQST_RQCB                      ; 0786
                                50      3C  A2  9E 0004C 5$:    MOVAB   60(R2), R0                              ; 0787
                                50          54  D1 00050        CMPL    RQST_RQCB, R0                           ;
                                            53  13 00053        BEQL    7$                                      ;
                51      04  AC 00000050 8F  C1 00055        ADDL3   #80, NOD, R1                            ; 0793
                                50      1C  A4  9E 0005E        MOVAB   28(RQST_RQCB), R0                       ;
                                        0000V 30 00062        BSBW    CLUSUTIL_SYSTEMID_EQUAL                 ;
                                3B          50  E9 00065        BLBC    R0, 6$                                  ;
                                6E 00058084 8F  D0 00068        MOVL    #360580, MESSAGE_VECTOR                 ; 0803
                                        04  AE  D4 0006F        CLRL    MESSAGE_VECTOR+4                        ; 0804
                            08  AE      70  A4  D0 00072        MOVL    112(RQST_RQCB), MESSAGE_VECTOR+8        ; 0805
                                54          64  0F 00077        REMQUE  (RQST_RQCB), RQST_RQCB                  ; 0806
                                        3A  A2  B7 0007A        DECW    58(OCD)                                 ; 0807
                                      4010  8F  BB 0007D        PUSHR   #^M<R4,SP>                              ; 0808
                        0000G   CF          02  FB 00081        CALLS   #2, FORMAT_MESSAGE                      ;
                                            54  DD 00086        PUSHL   RQST_RQCB                               ; 0813
                        0000G   CF          01  FB 00088        CALLS   #1, NOTIFY_LISTED_OPERATORS             ;
                                            54  DD 0008D        PUSHL   RQST_RQCB                               ; 0814
                        0000G   CF          01  FB 0008F        CALLS   #1, LOG_MESSAGE                         ;
                                50          54  D0 00094        MOVL    RQST_RQCB, RQCB                         ; 0815
                                54          64  D0 00097        MOVL    (RQST_RQCB), RQST_RQCB                  ; 0816
                                            50  DD 0009A        PUSHL   RQCB                                    ; 0817
                        0000G   CF          01  FB 0009C        CALLS   #1, DEALLOCATE_RQCB                     ;
                                            A9  11 000A1        BRB     5$                                      ; 0793
                                54          64  D0 000A3 6$:    MOVL    (RQST_RQCB), RQST_RQCB                  ; 0823
                                            A4  11 000A6        BRB     5$                                      ; 0787
                                            55  D7 000A8 7$:    DECL    OCD_COUNT                               ; 0825
                                52          62  D0 000AA        MOVL    (OCD), OCD                              ; 0826
```

```
                                95  11 000AD          BRB     4$                              ; 0780
                                53  D7 000AF 8$:      DECL    OCD_INDEX                       ; 0828
                              FF76  31 000B1          BRW     2$                              ; 0772
                                    04 000B4          RET                                     ; 0832

; Routine Size:  181 bytes,    Routine Base:  $CODE$ + 028C
```

```
;  845     0833  1 GLOBAL ROUTINE CLUSUTIL_NODE_MESSAGE (NOD : $ref_bblock, CODE, WORLD) : NOVALUE =
;  846     0834  1
;  847     0835  1 !++
;  848     0836  1 ! Functional description:
;  849     0837  1 !
;  850     0838  1 !       This routine notifies operators that the cluster configuration
;  851     0839  1 !       has changed.
;  852     0840  1 !
;  853     0841  1 !
;  854     0842  1 ! Inpu :
;  855     0843  1 !
;  856     0844  1 !       NOD     : Pointer to NOD data structure
;  857     0845  1 !       CODE    : OPCOM message code for the transition
;  858     0846  1 !       WORLD   : Flag - 1 send to rest of cluster, 0 to local node only
;  859     0847  1 !
;  860     0848  1 ! Implicit Input:
;  861     0849  1 !
;  862     0850  1 !       None.
;  863     0851  1 !
;  864     0852  1 ! Output:
;  865     0853  1 !
;  866     0854  1 !       None.
;  867     0855  1 !
;  868     0856  1 ! Implict output:
;  869     0857  1 !
;  870     0858  1 !       None.
;  871     0859  1 !
;  872     0860  1 ! Side effects:
;  873     0861  1 !
;  874     0862  1 !       Operators are notified.
;  875     0863  1 !
;  876     0864  1 ! Routine value:
;  877     0865  1 !
;  878     0866  1 !       None.
;  879     0867  1 !--
;  880     0868  1
;  881     0869  2 BEGIN                                           ! Start of CLUSUTIL_NODE_MESSAGE
;  882     0870  2
;  883     0871  2 LOCAL
;  884     0872  2         MESSAGE_VECTOR  : VECTOR [6,LONG],      ! Message info
;  885     0873  2         RQCB            : $ref_bblock,          ! RQCB data structure
;  886     0874  2         OCD             : $ref_bblock,          ! OCD data structure
;  887     0875  2         OCD_COUNT       : LONG,                 ! Count of OCDs in OCD list
;  888     0876  2         OCD_INDEX       : LONG,                 ! Index into OCD_VECTOR
;  889     0877  2         OPER_COUNT      : LONG,                 ! Count of operators in operator list
;  890     0878  2         STATUS          : LONG;
;  891     0879  2
;  892     0880  2 !
;  893     0881  2 ! Nothing to do if not in a cluster.
;  894     0882  2 !
;  895     0883  2 IF NOT .GLOBAL_STATUS [GBLSTS_K_IN_VAXcluster]
;  896     0884  2 THEN
;  897     0885  2     RETURN;
;  898     0886  2 !
;  899     0887  2 ! If we have printed an error message since the last timestamp, don't do another.
;  900     0888  2 !
;  901     0889  2 SELECTONE .CODE OF
```

```
;  902          0890  2 SET
;  903          0891  2     [OPC$_CLUSCOMM, OPC$_NODE_RETRY] :
;  904          0892  3                         BEGIN
;  905          0893  3                         IF .NOD [NOD_V_IOERR_DISPLAYED]         ! Have we already done one this timestamp?
;  906          0894  3                         THEN
;  907          0895  3                             RETURN;
;  908          0896  3                         NOD [NOD_V_IOERR_DISPLAYED] = TRUE;     ! Set the flag (cleared every timestamp)
;  909          0897  2                         END;
;  910          0898  2     [OTHERWISE] :       ;
;  911          0899  2 TES;
;  912          0900  2 !
;  913          0901  2 ! Allocate an RQCB.  This is necessary to format and later issue the message.
;  914          0902  2 !
;  915          0903  2 IF NOT ALLOCATE_DS (RQCB_K_TYPE, RQCB)
;  916          0904  2 THEN
;  917          0905  2     RETURN;
;  918          0906  2 !
;  919          0907  2 ! Set the operator interest mask to cluster
;  920          0908  2 !
;  921          0909  2 RQCB [RQCB_L_ATTNMASK1] = OPC$M_NM_CLUSTER;
;  922          0910  2 !
;  923          0911  2 ! Format the message, log it, and send it to all interested operators.
;  924          0912  2 ! Every operator in the data base is a candidate for the message.
;  925          0913  2 !
;  926          0914  2 MESSAGE_VECTOR [0] = .CODE;                    ! Set the message according to the flag.
;  927          0915  2 MESSAGE_VECTOR [1] = 0;                       ! Use current system time
;  928          0916  2 MESSAGE_VECTOR [2] = LCL_NOD [NOD_Q_NAME_DESC]; ! Use our name
;  929          0917  2 MESSAGE_VECTOR [3] = NOD [NOD_Q_NAME_DESC];    ! Set addr of node name descriptor
;  930          0918  2 MESSAGE_VECTOR [4] = .NOD [NOD_T_NODE_CSID];    ! Set node csid
;  931          0919  2 MESSAGE_VECTOR [5] = .(NOD [NOD_T_NODE_SYSTEMID])<0,16,0>;      ! Set node number
;  932          0920  2
;  933          0921  2 FORMAT_MESSAGE (.RQCB, MESSAGE_VECTOR);
;  934          0922  2 LOG_MESSAGE (.RQCB);                          ! Log the message
;  935          0923  2 !
;  936          0924  2 ! Send it to the world
;  937          0925  2 !
;  938          0926  2 IF .WORLD
;  939          0927  2 THEN
;  940          0928  2     CLUSMSG_RQCB_SEND (-1, CLM__CLUSTER, .RQCB);
;  941          0929  2 !
;  942          0930  2 ! Release the rqcb
;  943          0931  2 !
;  944          0932  2 DEALLOCATE_RQCB (.RQCB);
;  945          0933  2 RETURN;
;  946          0934  2
;  947          0935  1 END;                                           ! End of CLUSUTIL_NODE_MESSAGE


                                 000C 00000          .ENTRY  CLUSUTIL_NODE_MESSAGE, Save R2,R3          ; 0833
                      5E           1C   C2 00002          SUBL2   #28, SP                                    ;
                      01    0000G  CF   E8 00005          BLBS    GLOBAL_STATUS+1, 1$                        ; 0883
                                   04 0000A               RET                                                ;
                      53      08   AC   D0 0000B 1$:      MOVL    CODE, R3                                   ; 0889
          0005823B    8F           53   D1 0000F          CMPL    R3, #361019                                ; 0891
```

```
                                        09  13 00016       BEQL    2$                          ;
                00058253  8F            53  D1 00018       CMPL    R3, #361043                 ;
                                        0D  12 0001F       BNEQ    3$                          ;
                          50      04    AC  D0 00021 2$:   MOVL    NOD, R0                     ; 0893
               65   2A    A0            02  E0 00025       BBS     #2, 42(R0), 5$              ;
                    2A    A0            04  88 0002A       BISB2   #4, 42(R0)                  ; 0896
                                        5E  DD 0002E 3$:   PUSHL   SP                          ; 0903
                              00000000G 8F  DD 00030       PUSHL   #RQCB_K_TYPE                ;
                  0000G   CF            02  FB 00036       CALLS   #2, ALLOCATE_DS             ;
                          51            50  E9 0003B       BLBC    R0, 5$                      ;
                          52            6E  D0 0003E       MOVL    RQCB, R2                    ; 0909
                    5C    A2      80    8F  9A 00041       MOVZBL  #128, 92(R2)                ;
                    04    AE            53  D0 00046       MOVL    R3, MESSAGE_VECTOR          ; 0914
                                  08    AE  D4 0004A       CLRL    MESSAGE_VECTOR+4            ; 0915
          0C   AE  0000G  CF            30  C1 0004D       ADDL3   #48, LCL_NOD, MESSAGE_VECTOR+8 ; 0916
                          50      04    AC  D0 00054       MOVL    NOD, R0                     ; 0917
                    10    AE      30    A0  9E 00058       MOVAB   48(R0), MESSAGE_VECTOR+12   ;
                    14    AE      2C    A0  D0 0005D       MOVL    44(R0), MESSAGE_VECTOR+16   ; 0918
                    18    AE      50    A0  3C 00062       MOVZWL  80(R0), MESSAGE_VECTOR+20   ; 0919
                                  04    AE  9F 00067       PUSHAB  MESSAGE_VECTOR              ; 0921
                                        52  DD 0006A       PUSHL   R2                          ;
                  0000G   CF            02  FB 0006C       CALLS   #2, FORMAT_MESSAGE          ;
                                        52  DD 00071       PUSHL   R2                          ; 0922
                  0000G   CF            01  FB 00073       CALLS   #1, LOG_MESSAGE             ;
                          0C      0C    AC  E9 00078       BLBC    WORLD, 4$                   ; 0926
                                        52  DD 0007C       PUSHL   R2                          ; 0928
                                        07  DD 0007E       PUSHL   #7                          ;
                          7E            01  CE 00080       MNEGL   #1, -(SP)                   ;
                  0000G   CF            03  FB 00083       CALLS   #3, CLUSMSG_RQCB_SEND       ;
                                        52  DD 00088 4$:   PUSHL   R2                          ; 0932
                  0000G   CF            01  FB 0008A       CALLS   #1, DEALLOCATE_RQCB         ;
                                            04 0008F 5$:   RET                                 ; 0935

; Routine Size:  144 bytes,    Routine Base:  $CODE$ + 0341
```

```
;  949          0936  1 GLOBAL ROUTINE CLUSUTIL_NODE_START (NOD : $ref_bblock) : NOVALUE =              %SBTTL 'clusutil_node_start'
;  950          0937  1
;  951          0938  1 !++
;  952          0939  1 ! Functional description:
;  953          0940  1 !
;  954          0941  1 !       initialize a NOD block.
;  955          0942  1 !
;  956          0943  1 ! Input:
;  957          0944  1 !
;  958          0945  1 !       None.
;  959          0946  1 !
;  960          0947  1 ! Implicit Input:
;  961          0948  1 !
;  962          0949  1 !       Data in local storage from SYI call.
;  963          0950  1 !
;  964          0951  1 ! Output:
;  965          0952  1 !
;  966          0953  1 !       None.
;  967          0954  1 !
;  968          0955  1 ! Implict output:
;  969          0956  1 !
;  970          0957  1 !       None.
;  971          0958  1 !
;  972          0959  1 ! Side effects:
;  973          0960  1 !
;  974          0961  1 !       NOD block allocated.
;  975          0962  1 !
;  976          0963  1 ! Routine value:
;  977          0964  1 !
;  978          0965  1 !       None.
;  979          0966  1 !--
;  980          0967  1
;  981          0968  2 BEGIN                                           ! Start of CLUSUTIL_ADD_NOD
;  982          0969  2
;  983          0970  2 LOCAL
;  984          0971  2     STATUS;
;  985          0972  2 !
;  986          0973  2 ! Fill in the data from the $GETSYI buffers
;  987          0974  2 !
;  988          0975  2 NOD [NOD_B_STATE] = NOD_K_STATE_START;          ! Set to START state
;  989          0976  2 NOD [NOD_V_IOERR_DISPLAYED] = FALSE;
;  990          0977  2 NOD [NOD_V_NODE_LEAVING] = FALSE;
;  991          0978  2 NOD [NOD_L_NODE_CSID] = .NODE_CSID;
;  992          0979  2 NOD [NOD_L_NAME_LEN] = .NAME_LEN;
;  993          0980  2 NOD [NOD_L_NAME_PTR] = NOD [NOD_T_NAME_BUF];
;  994          0981  2 CH$MOVE (.NAME_LEN, NAME_BUF, NOD [NOD_T_NAME_BUF]);
;  995          0982  2 CH$MOVE (8, SWINCARN, NOD [NOD_Q_SWINCARN]);
;  996          0983  2 CH$MOVE (6, SYSTEMID, NOD [NOD_T_NODE_SYSTEMID]);
;  997          0984  2 !
;  998          0985  2 RETURN .NOD;
;  999          0986  1 END;


                                00FC 00000          .ENTRY  CLUSUTIL_NODE_START, Save R2,R3,R4,R5,R6,R7 ; 0936
```

```
                                 57      0000'  CF  9E 00002         MOVAB   NAME_LEN, R7                                ;
                                 56        04   AC  D0 00007         MOVL    NOD, R6                                    ; 0975
                           22    A6              02  90 0000B         MOVB    #2, 34(R6)                                 ;
                           2A    A6              0C  8A 0000F         BICB2   #12, 42(R6)                                ; 0977
                           2C    A6        DC    A7  D0 00013         MOVL    NODE_CSID, 44(R6)                          ; 0978
                           30    A6              67  D0 00018         MOVL    NAME_LEN, 48(R6)                           ; 0979
                           34    A6        38    A6  9E 0001C         MOVAB   56(R6), 52(R6)                             ; 0980
                     38    A6        F0    A7      67  28 00021         MOVC3   NAME_LEN, NAME_BUF, 56(R6)                 ; 0981
                     48    A6        E8    A7      08  28 00027         MOVC3   #8, SWINCARN, 72(R6)                       ; 0982
                     50    A6        E0    A7      06  28 0002D         MOVC3   #6, SYSTEMID, 80(R6)                       ; 0983
                                                     04  00033         RET                                                ; 0986

; Routine Size:  52 bytes,    Routine Base:  $CODE$ + 03D1
```

```
;1001     0987  1 GLOBAL ROUTINE CLUSUTIL_SYSTEMID_EQUAL (SYS_1 : $ref_bblock, SYS_2 : $ref_bblock) : JSB_R0R1 =
;1002     0988  1
;1003     0989  1 !++
;1004     0990  1 ! Functional description:
;1005     0991  1 !
;1006     0992  1 !       Compare two 48-bit SCS system ids for equivalence.
;1007     0993  1 !
;1008     0994  1 ! Input:
;1009     0995  1 !
;1010     0996  1 !       SYS_1   : Pointer to a 48-bit SCS id
;1011     0997  1 !       SYS_2   : Pointer to a 48-bit SCS id
;1012     0998  1 !
;1013     0999  1 ! Implicit Input:
;1014     1000  1 !
;1015     1001  1 !       None.
;1016     1002  1 !
;1017     1003  1 ! Output:
;1018     1004  1 !
;1019     1005  1 !       None.
;1020     1006  1 !
;1021     1007  1 ! Implict output:
;1022     1008  1 !
;1023     1009  1 !       None.
;1024     1010  1 !
;1025     1011  1 ! Side effects:
;1026     1012  1 !
;1027     1013  1 !       None.
;1028     1014  1 !
;1029     1015  1 ! Routine value:
;1030     1016  1 !
;1031     1017  1 !       True if IDs same, false if not
;1032     1018  1 !--
;1033     1019  1
;1034     1020  2 BEGIN                                           ! Start of CLUSUTIL_SYSTEMID_EQUAL
;1035     1021  2
;1036     1022  2 IF .SYS_1 [0,0,32,0] NEQ .SYS_2 [0,0,32,0]      ! First 32 bits
;1037     1023  2   OR
;1038     1024  2     .SYS_1 [4,0,16,0] NEQ .SYS_2 [4,0,16,0]     ! Next 16 bits
;1039     1025  2 THEN
;1040     1026  2     RETURN FALSE;
;1041     1027  2
;1042     1028  2 RETURN TRUE;
;1043     1029  1 END;                                            ! End of CLUSUTIL_SYSTEMID_EQUAL


                                61            60  D1 00000G CLUSUTIL_SYSTEMID_EQUAL::
                                                             CMPL    (SYS_1), (SYS_2)                ;1022
                                              0B  12 00003   BNEQ    1$
                            04  A1        04  A0  B1 00005   CMPW    4(SYS_1), 4(SYS_2)              ;1024
                                              04  12 0000A   BNEQ    1$
                                50            01  D0 0000C   MOVL    #1, R0                          ;1028
                                              05  0000F     RSB
                                                  50  D4 00010 1$:  CLRL    R0                      ;1029
                                                      05  00012   RSB
```

; Routine Size: 19 bytes,   Routine Base: $CODE$ + 0405

```
; 1045          1030  1 END
; 1046          1031  0 ELUDOM                        ! End of CLUSUTIL
```

```
;                                   PSECT SUMMARY
;
;       Name                    Bytes                           Attributes
;
;  $OWN$                          124  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $CODE$                        1048  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
```

```
;                                 Library Statistics
;
;                                      -------- Symbols --------      Pages       Processing
;       File                           Total    Loaded   Percent      Mapped      Time
;
;  _$255$DUA28:[SYSLIB]LIB.L32;1        18619        20         0        1000        00:01.9
;  _$255$DUA28:[OPCOM.OBJ]OPCOMLIB.L32;1  633        30         4          43        00:00.9
```

```
;                                   COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:CLUSUTIL/OBJ=OBJ$:CLUSUTIL MSRC$:CLUSUTIL/UPDATE=(ENH$:CLUSUTIL)

; Size:          1048 code + 124 data bytes
; Run Time:         00:22.7
; Elapsed Time:     01:09.8
; Lines/CPU Min:     2726
; Lexemes/CPU-Min: 14572
; Memory Used:  127 pages
; Compilation Complete
```

CLUMBX
LIS

DEVICE
LIS

LOGEVENT
LIS

CLUSREPLY
LIS

CLUSUTIL
LIS

DEBUG
LIS

CLUSCOMM
LIS

CLUSMSG
LIS