C 8

```
   000000000     PPPPPPPPPPPP        CCCCCCCCCCCC     000000000     MMM         MMM
   000000000     PPPPPPPPPPPP        CCCCCCCCCCCC     000000000     MMM         MMM
   000000000     PPPPPPPPPPPP        CCCCCCCCCCCC     000000000     MMM         MMM
000         000  PPP         PPP  CCC              000         000  MMMMMM   MMMMMM
000         000  PPP         PPP  CCC              000         000  MMMMMM   MMMMMM
000         000  PPP         PPP  CCC              000         000  MMMMMM   MMMMMM
000         000  PPP         PPP  CCC              000         000  MMM   MMM   MMM
000         000  PPP         PPP  CCC              000         000  MMM   MMM   MMM
000         000  PPP         PPP  CCC              000         000  MMM   MMM   MMM
000         000  PPPPPPPPPPPP     CCC              000         000  MMM         MMM
000         000  PPPPPPPPPPPP     CCC              000         000  MMM         MMM
000         000  PPPPPPPPPPPP     CCC              000         000  MMM         MMM
000         000  PPP              CCC              000         000  MMM         MMM
000         000  PPP              CCC              000         000  MMM         MMM
000         000  PPP              CCC              000         000  MMM         MMM
000         000  PPP              CCC              000         000  MMM         MMM
000         000  PPP              CCC              000         000  MMM         MMM
000         000  PPP              CCC              000         000  MMM         MMM
   000000000     PPP                 CCCCCCCCCCCC     000000000     MMM         MMM
   000000000     PPP                 CCCCCCCCCCCC     000000000     MMM         MMM
   000000000     PPP                 CCCCCCCCCCCC     000000000     MMM         MMM
```

```
_$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
```

```
**FILE**ID**CLUSMSG

  CCCCCCCC  LL          UU      UU    SSSSSSSS  MM      MM    SSSSSSSS    GGGGGGGG
  CCCCCCCC  LL          UU      UU    SSSSSSSS  MM      MM    SSSSSSSS    GGGGGGGG
CC          LL          UU      UU  SS          MMMM  MMMM  SS          GG
CC          LL          UU      UU  SS          MMMM  MMMM  SS          GG
CC          LL          UU      UU  SS          MM  MM  MM  SS          GG
CC          LL          UU      UU  SS          MM  MM  MM  SS          GG
CC          LL          UU      UU    SSSSSS    MM      MM    SSSSSS    GG
CC          LL          UU      UU    SSSSSS    MM      MM    SSSSSS    GG
CC          LL          UU      UU          SS  MM      MM          SS  GG  GGGGGG
CC          LL          UU      UU          SS  MM      MM          SS  GG  GGGGGG
CC          LL          UU      UU          SS  MM      MM          SS  GG      GG    ....
CC          LL          UU      UU          SS  MM      MM          SS  GG      GG    ....
  CCCCCCCC  LLLLLLLLLL  UUUUUUUUUU  SSSSSSSS    MM      MM  SSSSSSSS      GGGGGG      ....
  CCCCCCCC  LLLLLLLLLL  UUUUUUUUUU  SSSSSSSS    MM      MM  SSSSSSSS      GGGGGG      ....

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
```

```
0 MODULE  OPC$CLUSMSG       (
0                             LANGUAGE (BLISS32),
0                             IDENT = 'V04-000'
0                             ) =
0
0 !****************************************************************************
0 !*                                                                          *
0 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
0 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
0 !*  ALL RIGHTS RESERVED.                                                    *
0 !*                                                                          *
0 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
0 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
0 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
0 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
0 !*  TRANSFERRED.                                                            *
0 !*                                                                          *
0 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
0 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0 !*  CORPORATION.                                                            *
0 !*                                                                          *
0 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
0 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
0 !*                                                                          *
0 !*                                                                          *
0 !****************************************************************************
0
0 !++
0 ! FACILITY:
0 !
0 !       OPCOM
0 !
0 ! ABSTRACT:
0 !
0 !       This module contains the specialized logic to service
0 !       a particular type of request sent by a user to OPCOM.
0 !
0 ! Environment:
0 !
0 !       VAX/VMS operating system.
0 !
0 ! Author:
0 !
0 !       CW Hobbs
0 !
0 ! Creation date:
0 !
0 !       16-JUL-1983
0 !
0 ! Revision history:
0 !
0 !       V03-006 CWH3006         CW Hobbs                        24-May-1984
0 !               REPLY /USER etc. stopped working in a non-cluster system
0 !               because a check in CWH3169 was being applied to clm__rpybrd_local
0 !               messages.  Move the check inside the block which excludes
0 !               local node replies.
```

```
0 !
0 !     V03-005 CWH3005         CW Hobbs                        16-May-1984
0 !             Fix RSH0112 so that the receiving node will also see that
0 !             no unformatted text was sent.
0 !
0 !     V03-004 CWH3169         CW Hobbs                        5-May-1984
0 !             Second pass for cluster-wide OPCOM:
0 !               - Add CLM_L_CSID to clm message header, and make the embedded
0 !                 RQCB distinct, rather than overlaying on top of the header.
0 !               - If an input message has a standard header, then redo the
0 !                 header so that the local time is first, and put the remote
0 !                 time at the end.
0 !               - When a message is received, make sure that the CSID matches
0 !                 a node that we can see.  If not, discard the message.
0 !
0 !     V03-003 RSH0112         R. Scott Hanna          12-Mar-1984
0 !             CLUSMSG_RQCB_SEND / Increase the local buffer size
0 !             and prevent unformatted security auditing messages
0 !             from being sent to other cluster members.
0 !
0 !     V03-002 CWH3002         CW Hobbs                16-Sep-1983
0 !             Add CLUMBX message type, use VM jacket routines
0 !
0 !
0 !--
```

```
BEGIN                                                   ! Start of CLUSMSG

LIBRARY 'SYS$LIBRARY:LIB.L32';
LIBRARY 'LIB$:OPCOMLIB';

FORWARD ROUTINE
        CLUSMSG_ACK_PLEASE : NOVALUE,                   ! Request an acknowledgement
        CLUSMSG_CLM_ACK_HANDLER : NOVALUE,              ! Handle an acknowledgement
        CLUSMSG_CLM_ACK_PLEASE_HANDLER : NOVALUE,       ! Handle a request for an acknowledgement
        CLUSMSG_CLM_NOTIFY_HANDLER : NOVALUE,           ! Log message and notify operators
        CLUSMSG_CONV_CLM_RQCB,                          ! Convert a CLMRQCB structure to an RQCB
        CLUSMSG_HANDLER : NOVALUE,                      ! Main level, check message and dispatch
        CLUSMSG_RQCB_SEND,                              ! Convert RQCB to CLMRQCB and send to cluster
        CLUSMSG_STATE_SEND;                             ! Send current state to cluster node(s)

EXTERNAL ROUTINE
        !
        ! Miscellaneous routines
        !
        ALLOCATE_DS,
        CLUSCOMM_SEND,                                  ! Send message to the cluster
        CLUSUTIL_CONFIGURE,                             ! Configure the club membership
        CLUSUTIL_FIND_NOD_BY_CSID,                      ! Find a NOD block by its CSID
        CLUSUTIL_NODE_ACTIVATE,                         ! Make a node active
        CLUSUTIL_NODE_MESSAGE,                          ! Tell operators of a node activity
        DEALLOCATE_RQCB,                                ! Release an RQCB
        DUMP_LOG_FILE,                                  ! Write a string to the log file
        IMPLICITLY_CANCELED,                            ! Look for implicitly canceled requests
        IMPLIED_CANCEL,                                 ! Cancel queue of requests to be canceled
        IMPLIED_DISABLE,                                ! Disable stale operators
        LOG_MESSAGE,                                    ! Write a message to the logfile
        NOTIFY_LISTED_OPERATORS,                        ! Send messages to operators
        SHARE_FAO_BUFFER,                               ! Format an FAO string
        WRITE_LOG_FILE,                                 ! Write a string to the log file
        !
        ! Handlers for cluster messages (CLMs)
        !
        CANCEL_CLM_HANDLER              : NOVALUE,      ! Cancel request from remote
        CLUSREPLY_RPYBRD_HANDLER        : NOVALUE,      ! Broadcast from remote REPLY command
        CLUSREPLY_RPYBRD_LOCAL_HANDLER  : NOVALUE,      ! Broadcast from local REPLY command
        CLUSREPLY_RPYNOT_HANDLER        : NOVALUE,      ! Notification from remote REPLY command
        OPERUTIL_CLM_IMP_DISABLE        : NOVALUE,      ! Implicitly disable a remote operator
        OPRENABLE_CLM_HANDLER           : NOVALUE,      ! Enable/Disable remote operator
        REPLY_CLM_HANDLER               : NOVALUE,      ! Remote reply (/PEND, /TO) handler
        REQUEST_CLM_HANDLER             : NOVALUE,      ! Remote request handler
        REQUEST_CLM_CHECK_HANDLER       : NOVALUE,      ! Remote check request handler
        SHUTDOWN_CLM_HANDLER            : NOVALUE;      ! Shutdown ordered by remote handler

EXTERNAL
        LCL_NOD         : $ref_bblock,
        LCL_CSID        : LONG,
        NOD_HEAD        : VECTOR [2, LONG],
        OCD_VECTOR      : VECTOR,                       ! OCD list heads
        GLOBAL_STATUS   : BITVECTOR;

EXTERNAL LITERAL
        MCB_K_TYPE,
;  141          0140  1         RQCB_K_TYPE,
;  142          0141  1         MIN_SCOPE,                                ! Minimum scope value
;  143          0142  1         MAX_SCOPE;                                ! Maximum scope value
```

```
1 GLOBAL ROUTINE CLUSMSG_ACK_PLEASE (NOD : $ref_bblock) : NOVALUE =                %SBTTL 'clusmsg_ack_please'
1
1 !++
1 ! Functional description:
1 !
1 !       Request an acknowledgement from a remote node.
1 !
1 ! Input:
1 !
1 !       NOD - pointer to NOD structure of the remote node
1 !
1 ! Implicit Input:
1 !
1 !       LCL_NOD - pointer to NOD structure for local node
1 !
1 ! Output:
1 !
1 !       None.
1 !
1 ! Implict output:
1 !
1 !       None.
1 !
1 ! Side effects:
1 !
1 !       Message sent to remote.
1 !
1 ! Routine value:
1 !
1 !       None.
1 !--
1
2 BEGIN                                                     ! Start of CLUSMSG_ACK_PLEASE
2
2 LOCAL
2     MSG : $bblock [CLMACK_K_SIZE],
2     STATUS;
2 !
2 ! If we have an ack pending, just return to avoid flooding with ack messages.  To resend
2 ! an ack, you must clear this bit before calling this routine.
2 !
2 IF .NOD [NOD_V_ACK_PEND]
2 THEN
2     RETURN;
2 !
2 ! If we have already tried to talk to this guy, let them know
2 !
2 IF .NOD [NOD_V_ACK_ATTEMPTED]
2 THEN
2     CLUSUTIL_NODE_MESSAGE (.NOD, OPC$_NODE_RETRY, FALSE);
2 NOD [NOD_V_ACK_ATTEMPTED] = TRUE;
2 !
2 ! Fill in the ack message header
2 !
2 MSG [CLM_B_RQSTCODE]     = OPC$_X_CLUSMSG;
2 MSG [CLM_B_CLM_CODE]     = CLM__ACKNOWLEDGE_PLEASE;
2 MSG [CLM_B_DS_VERSION]   = CLMACK_K_DS_VERSION;
```

```
;   202       0200  2 MSG [CLM_B_SW_VERSION]  = OPC$K_SW_VERSION;
;   203       0201  2 MSG [CLM_W_LENGTH]      = CLMACK_K_SIZE;
;   204       0202  2 MSG [CLM_W_FILL_1]      = 0;
;   205       0203  2 MSG [CLM_L_CSID]        = .LCL_CSID;
;   206       0204  2 !
;   207       0205  2 ! Fill in the ack message from the local node info
;   208       0206  2 !
;   209       0207  2 MSG [CLMACK_L_CSID] = .LCL_NOD [NOD_L_NODE_CSID];
;   210       0208  2 MSG [CLMACK_L_SYSTEMIDL] = .LCL_NOD [NOD_L_NODE_SYSTEMIDL];
;   211       0209  2 MSG [CLMACK_W_SYSTEMIDH] = .LCL_NOD [NOD_W_NODE_SYSTEMIDH];
;   212       0210  2 !
;   213       0211  2 ! Send the message
;   214       0212  2 !
;   215       0213  2 STATUS = CLUSCOMM_SEND (.NOD [NOD_L_NODE_CSID], CLMACK_K_SIZE, MSG);
;   216       0214  2 !
;   217       0215  2 ! If we were able to send, mark it as pending
;   218       0216  2 !
;   219       0217  2 NOD [NOD_V_ACK_PEND] = .STATUS;
;   220       0218  2
;   221       0219  2 RETURN;
;   222       0220  1 END;


                                                          .TITLE  OPC$CLUSMSG
                                                          .IDENT  \V04-000\

                                                          .EXTRN  ALLOCATE_DS, CLUSCOMM_SEND
                                                          .EXTRN  CLUSUTIL_CONFIGURE
                                                          .EXTRN  CLUSUTIL_FIND_NOD_BY_CSID
                                                          .EXTRN  CLUSUTIL_NODE_ACTIVATE
                                                          .EXTRN  CLUSUTIL_NODE_MESSAGE
                                                          .EXTRN  DEALLOCATE_RQCB
                                                          .EXTRN  DUMP_LOG_FILE, IMPLICITLY_CANCELED
                                                          .EXTRN  IMPLIED_CANCEL, IMPLIED_DISABLE
                                                          .EXTRN  LOG_MESSAGE, NOTIFY_LISTED_OPERATORS
                                                          .EXTRN  SHARE_FAO_BUFFER
                                                          .EXTRN  WRITE_LOG_FILE, CANCEL_CLM_HANDLER
                                                          .EXTRN  CLUSREPLY_RPYBRD_HANDLER
                                                          .EXTRN  CLUSREPLY_RPYBRD_LOCAL_HANDLER
                                                          .EXTRN  CLUSREPLY_RPYNOT_HANDLER
                                                          .EXTRN  OPERUTIL_CLM_IMP_DISABLE
                                                          .EXTRN  OPRENABLE_CLM_HANDLER
                                                          .EXTRN  REPLY_CLM_HANDLER
                                                          .EXTRN  REQUEST_CLM_HANDLER
                                                          .EXTRN  REQUEST_CLM_CHECK_HANDLER
                                                          .EXTRN  SHUTDOWN_CLM_HANDLER
                                                          .EXTRN  LCL_NOD, LCL_CSID
                                                          .EXTRN  NOD_HEAD, OCD_VECTOR
                                                          .EXTRN  GLOBAL_STATUS, MCB_K_TYPE
                                                          .EXTRN  RQCB_K_TYPE, MIN_SCOPE
                                                          .EXTRN  MAX_SCOPE

                                                          .PSECT  $CODE$,NOWRT,2

                                      0004 00000          .ENTRY  CLUSMSG_ACK_PLEASE, Save R2                      ; 0143
                    5E           18    C2 00002           SUBL2   #24, SP                                          ;
                    52      04   AC    D0 00005           MOVL    NOD, R2                                          ; 0184
```

```
                        54      2A   A2  E8 00009         BLBS    42(R2), 2$                                ;
           0F     2A    A2           01  E1 0000D         BBC     #1, 42(R2), 1$                            ; 0190
                                     7E  D4 00012         CLRL    -(SP)                                     ; 0192
                           0005823B  8F  DD 00014         PUSHL   #361019                                   ;
                                     52  DD 0001A         PUSHL   R2                                        ;
                  0000G   CF         03  FB 0001C         CALLS   #3, CLUSUTIL_NODE_MESSAGE                 ;
                    2A    A2         02  88 00021  1$:    BISB2   #2, 42(R2)                                ; 0193
                          6E   0213  8F  B0 00025         MOVW    #531, MSG                                 ; 0197
                    02    AE 00160902 8F D0 0002A         MOVL    #1444098, MSG+2                           ; 0199
                                 06  AE  B4 00032         CLRW    MSG+6                                     ; 0202
                    08    AE   0000G CF  D0 00035         MOVL    LCL_CSID, MSG+8                           ; 0203
                          50   0000G CF  D0 0003B         MOVL    LCL_NOD, R0                               ; 0207
                    0C    AE     2C  A0  D0 00040         MOVL    44(R0), MSG+12                            ;
                    10    AE     50  A0  D0 00045         MOVL    80(R0), MSG+16                            ; 0208
                    14    AE     54  A0  B0 0004A         MOVW    84(R0), MSG+20                            ; 0209
                                     5E  DD 0004F         PUSHL   SP                                        ; 0213
                                     16  DD 00051         PUSHL   #22                                       ;
                                 2C  A2  DD 00053         PUSHL   44(R2)                                    ;
                  0000G   CF         03  FB 00056         CALLS   #3, CLUSCOMM_SEND                         ;
    2A    A2      01      00         50  F0 0005B         INSV    STATUS, #0, #1, 42(R2)                    ; 0217
                                     04  00061     2$:    RET                                               ; 0220

; Routine Size:  98 bytes,    Routine Base:  $CODE$ + 0000
```

```
      0221   1 GLOBAL ROUTINE CLUSMSG_CLM_ACK_HANDLER (BUFFER_DESC : $ref_bblock, CLM : $ref_bblock, LEN) : NOVALUE =
      0222   1
      0223   1 !++
      0224   1 ! Functional description:
      0225   1 !
      0226   1 !       Handle an acknowledgement from a remote node.
      0227   1 !
      0228   1 ! Input:
      0229   1 !
      0230   1 !       BUFFER_DESC -   pointer to message from remote node, including $SNDOPR header
      0231   1 !       CLM -           pointer to CLMACK structure
      0232   1 !       LEN -           length of LEN
      0233   1 !
      0234   1 ! Implicit Input:
      0235   1 !
      0236   1 !       None.
      0237   1 !
      0238   1 ! Output:
      0239   1 !
      0240   1 !       None.
      0241   1 !
      0242   1 ! Implict output:
      0243   1 !
      0244   1 !       None.
      0245   1 !
      0246   1 ! Side effects:
      0247   1 !
      0248   1 !       Message sent to remote.
      0249   1 !
      0250   1 ! Routine value:
      0251   1 !
      0252   1 !       None.
      0253   1 !--
      0254   1
      0255   2 BEGIN                                         ! Start of CLUSMSG_CLM_ACK_HANDLER
      0256   2
      0257   2 LOCAL
      0258   2     NOD : $ref_bblock,
      0259   2     STATUS;
      0260   2 !
      0261   2 ! Check the version number of the message.  If the message is from any other version,
      0262   2 ! simply ignore it.
      0263   2 !
      0264   2 IF .CLM [CLM_B_DS_VERSION] NEQ CLMACK_K_DS_VERSION
      0265   2 THEN
      0266   2     RETURN DUMP_LOG_FILE (.BUFFER_DESC, %ASCID 'CLM__ACK mismatch');
      0267   2 !
      0268   2 ! Find the NOD structure
      0269   2 !
      0270   2 NOD = CLUSUTIL_FIND_NOD_BY_CSID (.CLM [CLMACK_L_CSID]);
      0271   2 IF .NOD EQL 0
      0272   2 THEN
      0273   2     RETURN DUMP_LOG_FILE (.BUFFER_DESC, %ASCID 'no NOD for ACK');
      0274   2 !
      0275   2 ! Mark the NOD as active
      0276   2 !
      0277   2 CLUSUTIL_NODE_ACTIVATE (.NOD);
```

```
;  281          0278  2
;  282          0279  2 RETURN;
;  283          0280  1 END;


                                                                    .PSECT  $PLIT$,NOWRT,NOEXE,2

74  61  6D  73  69  6D  20  4B  43  41   5F   5F   4D   4C   43   00000 P.AAB:  .ASCII  \CLM__ACK mismatch\<0><0><0>           ;
                                         00   00   00   68   63   0000F                                                       ;
                                                   010E0011   00014 P.AAA:  .LONG   17694737                                  ;
                                                  00000000'   00018         .ADDRESS P.AAB                                    ;
00  4B  43  41  20  72  6F  66  20  44   4F   4E   20   6F   6E   0001C P.AAD:  .ASCII  \no NOD for ACK\<0><0>                  ;
                                                        00   0002B                                                            ;
                                                   010E000E   0002C P.AAC:  .LONG   17694734                                  ;
                                                  00000000'   00030         .ADDRESS P.AAD                                    ;


                                                                    .PSECT  $CODE$,NOWRT,2

                                                    0004   00000         .ENTRY  CLUSMSG_CLM_ACK_HANDLER, Save R2         ; 0221
                                  52       08   AC   D0   00002         MOVL    CLM, R2                                  ; 0264
                                  02       02   A2   91   00006         CMPB    2(R2), #2                                 ;
                                                06   13   0000A         BEQL    1$                                       ;
                                       0000'    CF   9F   0000C         PUSHAB  P.AAA                                    ; 0266
                                                11   11   00010         BRB     2$                                       ;
                                           0C   A2   DD   00012 1$:     PUSHL   12(R2)                                   ; 0270
                          0000G   CF            01   FB   00015         CALLS   #1, CLUSUTIL_FIND_NOD_BY_CSID             ;
                                  52            50   D0   0001A         MOVL    R0, NOD                                  ;
                                                0D   12   0001D         BNEQ    3$                                       ; 0271
                                       0000'    CF   9F   0001F         PUSHAB  P.AAC                                    ; 0273
                                           04   AC   DD   00023 2$:     PUSHL   BUFFER_DESC                              ;
                          0000G   CF            02   FB   00026         CALLS   #2, DUMP_LOG_FILE                        ;
                                                     04   0002B         RET                                              ;
                                                52   DD   0002C 3$:     PUSHL   NOD                                      ; 0277
                          0000G   CF            01   FB   0002E         CALLS   #1, CLUSUTIL_NODE_ACTIVATE               ;
                                                     04   00033         RET                                              ; 0280

; Routine Size:  52 bytes,    Routine Base:  $CODE$ + 0062
```

```
1 GLOBAL ROUTINE CLUSMSG_CLM_ACK_PLEASE_HANDLER (BUFFER_DESC : $ref_bblock, CLM : $ref_bblock, LEN) : NOVALUE
1
1 !++
1 ! Functional description:
1 !
1 !       Request an acknowledgement from a remote node.
1 !
1 ! Input:
1 !
1 !       BUFFER_DESC -   pointer to message from remote node, including $SNDOPR header
1 !       CLM -           pointer to CLMRQCB structure
1 !       LEN -           length of LEN
1 !
1 ! Implicit Input:
1 !
1 !       None.
1 !
1 ! Output:
1 !
1 !       None.
1 !
1 ! Implict output:
1 !
1 !       None.
1 !
1 ! Side effects:
1 !
1 !       Message sent to remote.
1 !
1 ! Routine value:
1 !
1 !       None.
1 !--
1
2 BEGIN                                                   ! Start of CLUSMSG_ACK_PLEASE_HANDLER
2
2 LOCAL
2     MSG : $bblock [CLMACK_K_SIZE],
2     NOD : $ref_bblock,
2     STATUS;
2 !
2 ! Check the version number of the message.  If the message is from any other version,
2 ! simply ignore it.
2 !
2 IF .CLM [CLM_B_DS_VERSION] NEQ CLMACK_K_DS_VERSION
2 THEN
2     RETURN DUMP_LOG_FILE (.BUFFER_DESC, %ASCID 'CLM__ACK mismatch');
2 !
2 ! Tell the requestor everthing we know
2 !
2 CLUSMSG_STATE_SEND (.CLM [CLMACK_L_CSID]);
2 !
2 ! Fill in the ack message header
2 !
2 MSG [CLM_B_RQSTCODE]     = OPC$_X_CLUSMSG;
2 MSG [CLM_B_CLM_CODE]     = CLM__ACKNOWLEDGEMENT;
2 MSG [CLM_B_DS_VERSION]   = CLMACK_K_DS_VERSION;
```

```
;   342          0338  2 MSG [CLM_B_SW_VERSION]   = OPC$K_SW_VERSION;
;   343          0339  2 MSG [CLM_W_LENGTH]       = CLMACK_K_SIZE;
;   344          0340  2 MSG [CLM_W_FILL_1]       = 0;
;   345          0341  2 MSG [CLM_L_CSID]         = .LCL_CSID;
;   346          0342  2 !
;   347          0343  2 ! Fill in the ack message from the local node info
;   348          0344  2 !
;   349          0345  2 MSG [CLMACK_L_CSID] = .LCL_NOD [NOD_L_NODE_CSID];
;   350          0346  2 MSG [CLMACK_L_SYSTEMIDL] = .LCL_NOD [NOD_L_NODE_SYSTEMIDL];
;   351          0347  2 MSG [CLMACK_W_SYSTEMIDH] = .LCL_NOD [NOD_W_NODE_SYSTEMIDH];
;   352          0348  2 !
;   353          0349  2 ! Send the acknowledge message back to from where it came
;   354          0350  2 !
;   355          0351  2 CLUSCOMM_SEND (.CLM [CLMACK_L_CSID], CLMACK_K_SIZE, MSG);
;   356          0352  2 !
;   357          0353  2 ! If we haven't talked to this guy before, then request an acknowledgement from him
;   358          0354  2 !
;   359          0355  2 IF (NOD = CLUSUTIL_FIND_NOD_BY_CSID (.CLM [CLMACK_L_CSID])) NEQ 0
;   360          0356  2 THEN
;   361          0357  3     BEGIN
;   362          0358  3     IF .NOD [NOD_B_STATE] EQL NOD_K_STATE_START
;   363          0359  3     THEN
;   364          0360  4         BEGIN
;   365          0361  4         NOD [NOD_V_ACK_PEND] = FALSE;                ! Clear so that we can
;   366          0362  4         CLUSMSG_ACK_PLEASE (.NOD);                   !  request an acknowledgement
;   367          0363  3         END;
;   368          0364  2     END;
;   369          0365  2
;   370          0366  2 RETURN;
;   371          0367  1 END;


                                                                   .PSECT  $PLIT$,NOWRT,NOEXE,2

74  61  6D  73  69  6D  20  4B  43  41   5F  5F  4D  4C  43  00034 P.AAF:  .ASCII  \CLM__ACK mismatch\<0><0><0>                  ;
                                         00  00  00  68  63  00043                                                                 ;
                                                   010E0011  00048 P.AAE:  .LONG   17694737                                        ;
                                                   00000000' 0004C         .ADDRESS P.AAF                                          ;


                                                                   .PSECT  $CODE$,NOWRT,2

                                                      0004 00000           .ENTRY  CLUSMSG_CLM_ACK_PLEASE_HANDLER, Save R2         ; 0281
                           5E                   18  C2 00002           SUBL2   #24, SP                                         ;
                           52           08      AC  D0 00005           MOVL    CLM, R2                                         ; 0325
                           02           02      A2  91 00009           CMPB    2(R2), #2                                       ;
                                                0D  13 0000D           BEQL    1$                                              ;
                                     0000'      CF  9F 0000F           PUSHAB  P.AAE                                           ; 0327
                                        04      AC  DD 00013           PUSHL   BUFFER_DESC                                     ;
                    0000G  CF                   02  FB 00016           CALLS   #2, DUMP_LOG_FILE                               ;
                                                    04 0001B           RET                                                     ;
                                        0C      A2  DD 0001C 1$:       PUSHL   12(R2)                                          ; 0331
                    0000V  CF                   01  FB 0001F           CALLS   #1, CLUSMSG_STATE_SEND                          ;
                           6E         0113      8F  B0 00024           MOVW    #275, MSG                                       ; 0335
                      02   AE 00160902          8F  D0 00029           MOVL    #1444098, MSG+2                                 ; 0337
```

```
                                    06    AE  B4 00031        CLRW    MSG+6                                  ; 0340
                  08    AE       0000G    CF  D0 00034        MOVL    LCL_CSID, MSG+8                        ; 0341
                        50       0000G    CF  D0 0003A        MOVL    LCL_NOD, R0                            ; 0345
                  0C    AE          2C    A0  D0 0003F        MOVL    44(R0), MSG+12                         ;
                  10    AE          50    A0  D0 00044        MOVL    80(R0), MSG+16                         ; 0346
                  14    AE          54    A0  B0 00049        MOVW    84(R0), MSG+20                         ; 0347
                                          5E  DD 0004E        PUSHL   SP                                     ; 0351
                                          16  DD 00050        PUSHL   #22                                    ;
                                    0C    A2  DD 00052        PUSHL   12(R2)                                 ;
               0000G    CF                03  FB 00055        CALLS   #3, CLUSCOMM_SEND                      ;
                                    0C    A2  DD 0005A        PUSHL   12(R2)                                 ; 0355
               0000G    CF                01  FB 0005D        CALLS   #1, CLUSUTIL_FIND_NOD_BY_CSID          ;
                                          50  D5 00062        TSTL    NOD                                    ;
                                          11  13 00064        BEQL    2$                                     ;
                  02                22    A0  91 00066        CMPB    34(NOD), #2                            ; 0358
                                          0B  12 0006A        BNEQ    2$                                     ;
                  2A    A0                01  8A 0006C        BICB2   #1, 42(NOD)                            ; 0361
                                          50  DD 00070        PUSHL   NOD                                    ; 0362
               FEF3     CF                01  FB 00072        CALLS   #1, CLUSMSG_ACK_PLEASE                 ;
                                              04 00077 2$:    RET                                            ; 0367

; Routine Size:  120 bytes,    Routine Base:  $CODE$ + 0096
```

```
1 GLOBAL ROUTINE CLUSMSG_CLM_NOTIFY_HANDLER (BUFFER_DESC : $ref_bblock, CLM : $ref_bblock, LEN) : NOVALUE =
1
1 !++
1 ! Functional description:
1 !
1 !       This routine is the handler for all simple messages received from remote nodes.  Simple
1 !       messages are those which merely need to be logged and sent to interested operators.
1 !
1 ! Input:
1 !
1 !       BUFFER_DESC -   pointer to message from remote node, including $SNDOPR header
1 !       CLM -           pointer to CLMRQCB structure
1 !       LEN -           length of LEN
1 !
1 ! Implicit Input:
1 !
1 !       None.
1 !
1 ! Output:
1 !
1 !       None.
1 !
1 ! Implict output:
1 !
1 !       Some accounting data will be updated
1 !       to reflect the receipt of the message.
1 !
1 ! Side effects:
1 !
1 !       None.
1 !
1 ! Routine value:
1 !
1 !       None.
1 !--
1
2 BEGIN                                                 ! Start of CLUSMSG_CLM_NOTIFY_HANDLER
2
2 LOCAL
2       RQCB            : $ref_bblock,          ! RQCB data structure
2       OCD             : $ref_bblock,          ! OCD data structure
2       OCD_COUNT       : LONG,                 ! Count of OCDs in OCD list
2       OCD_INDEX       : LONG,                 ! Index into OCD_VECTOR
2       OPER_COUNT      : LONG,                 ! Count of operators in operator list
2       STATUS          : LONG;
2
2 !
2 ! Check the version number of the message.  If the message is from any other version,
2 ! simply ignore it.
2 !
2 IF .CLM [CLM_B_DS_VERSION] NEQ CLMRQCB_K_DS_VERSION
2 THEN
2     RETURN DUMP_LOG_FILE (.BUFFER_DESC, %ASCID 'clm notify mismatch');
2 !
2 ! Allocate an RQCB and convert the message RQCB into the new RQCB
2 !
2 IF NOT CLUSMSG_CONV_CLM_RQCB (.CLM, RQCB)
```

```
;  430          0425  2 THEN
;  431          0426  2      RETURN DUMP_LOG_FILE (.BUFFER_DESC, ascid_INVALIDRQCB);
;  432          0427  2 !
;  433          0428  2 ! Log it, and send it to all interested operators.
;  434          0429  2 ! Every operator in the data base is a candidate for the message.
;  435          0430  2 !
;  436          0431  2 OCD_INDEX = MAX_SCOPE;
;  437          0432  2 WHILE (.OCD_INDEX GEQ MIN_SCOPE) DO
;  438          0433  3      BEGIN
;  439          0434  3      !
;  440          0435  3      ! Scan the OCD list for each class of operator.
;  441          0436  3      !
;  442          0437  3      OCD_COUNT = .OCD_VECTOR [(.OCD_INDEX - 1) * 2 + 1];
;  443          0438  3      OCD = .OCD_VECTOR [(.OCD_INDEX - 1) * 2];
;  444          0439  3      WHILE (.OCD_COUNT GTR 0) DO
;  445          0440  4           BEGIN
;  446          0441  4           !
;  447          0442  4           ! Notify every operator in the OCD's operator list.
;  448          0443  4           ! Also log the message for each OCD.
;  449          0444  4           !
;  450          0445  4           RQCB [RQCB_L_OCD] = .OCD;                   ! Set OCD address
;  451          0446  4           LOG_MESSAGE (.RQCB);                        ! Log the message
;  452          0447  4           NOTIFY_LISTED_OPERATORS (.RQCB);            ! Inform the operators
;  453          0448  4           OCD_COUNT = .OCD_COUNT - 1;                 ! Decrement operator count
;  454          0449  4           OCD = .OCD [OCD_L_FLINK];                   ! Get next OCD address
;  455          0450  3           END;
;  456          0451  3      OCD_INDEX = .OCD_INDEX - 1;
;  457          0452  2      END;
;  458          0453  2 !
;  459          0454  2 ! Free the rqcb
;  460          0455  2 !
;  461          0456  2 DEALLOCATE_RQCB (.RQCB);
;  462          0457  2 RETURN;
;  463          0458  2
;  464          0459  1 END;                                                    ! End of CLUSMSG_CLM_NOTIFY_HANDLER


                                                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

6D  73  69  6D  20  79  66  69  74  6F  6E  20  6D  6C  63  00050 P.AAH: .ASCII  \clm notify mismatch\<0>                     ;
                                            00  68  63  74  61  0005F                                                              ;
                                                  010E0013  00064 P.AAG: .LONG   17694739                                          ;
                                                  00000000' 00068        .ADDRESS P.AAH                                            ;

                                                                        .EXTRN  ASCID_INVALIDRQCB

                                                                        .PSECT  $CODE$,NOWRT,2

                                                      003C 00000        .ENTRY  CLUSMSG_CLM_NOTIFY_HANDLER, Save R2,R3,R4,-  ; 0368
                                                                                R5                                           ;
                              5E             04  C2 00002               SUBL2   #4, SP                                       ;
                              52       08    AC  D0 00005               MOVL    CLM, R2                                      ; 0418
                              02       02    A2  91 00009               CMPB    2(R2), #2                                    ;
                                             06  13 0000D               BEQL    1$                                           ;
                                   0000'     CF  9F 0000F               PUSHAB  P.AAG                                        ; 0420
                                             10  11 00013               BRB     2$                                           ;
```

```
                                  4004    8F  BB 00015 1$:      PUSHR   #^M<R2,SP>                              ; 0424
                    0000V  CF             02  FB 00019          CALLS   #2, CLUSMSG_CONV_CLM_RQCB               ;
                           0D             50  E8 0001E          BLBS    R0, 3$                                  ;
                                  0000G   CF  9F 00021          PUSHAB  ASCID_INVALIDRQCB                       ; 0426
                                     04   AC  DD 00025 2$:      PUSHL   BUFFER_DESC                             ;
                    0000G  CF             02  FB 00028          CALLS   #2, DUMP_LOG_FILE                       ;
                                              04 0002D          RET                                             ;
                           52 00000000G   8F  D0 0002E 3$:      MOVL    #MAX_SCOPE, OCD_INDEX                   ; 0431
                                          53  6E  D0 00035      MOVL    RQCB, R3                                ; 0445
                00000000G  8F             52  D1 00038 4$:      CMPL    OCD_INDEX, #MIN_SCOPE                   ; 0432
                                          35  19 0003F          BLSS    7$                                      ;
             50            52             01  78 00041          ASHL    #1, OCD_INDEX, R0                       ; 0437
                           55    0000GCF40    D0 00045          MOVL    OCD_VECTOR-4[R0], OCD_COUNT             ;
             50            52             01  78 0004B          ASHL    #1, OCD_INDEX, R0                       ; 0438
                           54    0000GCF40    D0 0004F          MOVL    OCD_VECTOR-8[R0], OCD                   ;
                                          55  D5 00055 5$:      TSTL    OCD_COUNT                               ; 0439
                                          19  15 00057          BLEQ    6$                                      ;
                   24      A3             54  D0 00059          MOVL    OCD, 36(R3)                             ; 0445
                                          6E  DD 0005D          PUSHL   RQCB                                    ; 0446
                    0000G  CF             01  FB 0005F          CALLS   #1, LOG_MESSAGE                         ;
                                          6E  DD 00064          PUSHL   RQCB                                    ; 0447
                    0000G  CF             01  FB 00066          CALLS   #1, NOTIFY_LISTED_OPERATORS             ;
                                          55  D7 0006B          DECL    OCD_COUNT                               ; 0448
                           54             64  D0 0006D          MOVL    (OCD), OCD                              ; 0449
                                          E3  11 00070          BRB     5$                                      ; 0439
                                          52  D7 00072 6$:      DECL    OCD_INDEX                               ; 0451
                                          C2  11 00074          BRB     4$                                      ; 0432
                                          6E  DD 00076 7$:      PUSHL   RQCB                                    ; 0456
                    0000G  CF             01  FB 00078          CALLS   #1, DEALLOCATE_RQCB                     ;
                                              04 0007D          RET                                             ; 0459

; Routine Size:  126 bytes,    Routine Base:  $CODE$ + 010E
```

```
1 GLOBAL ROUTINE CLUSMSG_CONV_CLM_RQCB (CLM : $ref_bblock, RET_RQCB) =    %SBTTL 'CLUSMSG_CONV_CLM_RQCB (CLM,
1
1 !++
1 ! Functional description:
1 !
1 !       Convert a CLMRQCB to a local RQCB
1 !
1 ! Input:
1 !
1 !       CLM      - Pointer to CLMRQCB structure
1 !       RET_RQCB - Address of longword to receive address of allocated RQCB
1 !
1 ! Implicit Input:
1 !
1 !       None.
1 !
1 ! Output:
1 !
1 !       None.
1 !
1 ! Implict output:
1 !
1 !       None.
1 !
1 ! Side effects:
1 !
1 !       Data structure will be allocated
1 !
1 ! Routine value:
1 !
1 !       Success or failure
1 !--
1
2 BEGIN                                                   ! Start of CLUSMSG_CONV_CLM_RQCB
2
2 LOCAL
2       LEN                     : LONG,
2       EOB                     : LONG,
2       PTR                     : $ref_bblock,
2       RQCB                    : $ref_bblock,
2       RQCBUF                  : $ref_bblock,
2       STATUS                  : LONG;
2
2 !
2 ! Set the return RQCB to null
2 !
2 .RET_RQCB = 0;
2 !
2 ! Make sure that it is an RQCB in the message
2 !
2 RQCBUF = CLM [CLMRQCB_T_RQCB_OVERLAY];
2 IF .RQCBUF [RQCB_W_SIZE] NEQ RQCB_K_SIZE
2   OR
2     .RQCBUF [RQCB_B_TYPE] NEQ RQCB_K_TYPE
2 THEN
2     RETURN FALSE;
2 !
```

```
2 ! Next thing, allocate an RQCB and copy the most of the CLM RQCB to the new RQCB,
2 ! taking care not to overwrite the RQCB header data
2 !
2 ALLOCATE_DS (RQCB_K_TYPE, RQCB);
2 CH$MOVE (RQCB_K_OVERLAY_SIZE, RQCBUF [RQCB_T_OVERLAY], RQCB [RQCB_T_OVERLAY]);
2 !
2 ! Take all of the character strings appended to the CLMRQCB and hang them from the RQCB
2 !
2 PTR = CLM [CLMRQCB_T_TEXT];                          ! Pointer to next data in text area
2 EOB = .CLM + .CLM [CLM_W_LENGTH];                    ! Pointer to last byte +1 of text area
2 !
2 ! If the original had an MCB, make a new MCB
2 !
2 IF (LEN = .RQCBUF [RQCB_L_MCB]) NEQ 0
2 THEN
3     BEGIN
3     LOCAL
3         FAO_DESC : VECTOR [2, LONG],
3         FAO_BUFF : VECTOR [OPC$K_MAXMESSAGE, BYTE],
3         MCB : $ref_bblock,
3         NOD : $ref_bblock,
3         NEXT;
3     IF (NEXT = .LEN + .PTR) GTRU .EOB
3     THEN
4         BEGIN
4         DEALLOCATE_RQCB (.RQCB);
4         RETURN FALSE;
3         END;
3     ALLOCATE_DS (MCB_K_TYPE, MCB);
3     RQCB [RQCB_L_MCB] = .MCB;
3     MCB [MCB_L_RQCB] = .RQCB;
3     MCB [MCB_L_MSGID] = .CLM [CLMRQCB_L_MCB_MSGID];      ! Restore message id
3     MCB [MCB_L_STATUS] = .CLM [CLMRQCB_L_MCB_STATUS];    !  and status
3     !
3     ! If the message is a standard header message, then readjust it so that we store the local
3     ! time at the front and record the remote time later in the message.
3     ! We check to make sure it hasn't been adjusted already, as can happen if the request was
3     ! being passed around.
3     !
3     IF CH$EQL (20, UPLIT BYTE ('%%%%%%%%%%%  OPCOM  '), 20, .PTR+1)
3     THEN
4         BEGIN
4         LOCAL
4             PAR, CR;
4         PAR = CH$FIND_CH (.LEN, .PTR, %C'(');              ! Find first open paren
4         CR  = CH$FIND_CH (.LEN, .PTR, 13);                 ! Find first carriage return (gotta have one!)
4         IF .PAR EQL 0                                      ! If no paren
4           OR
4             .PAR GTR .CR                                   !  or if paren after first <CR>
4         THEN
5             BEGIN
5             FAO_DESC [0] = OPC$K_MAXMESSAGE;
5             FAO_DESC [1] = FAO_BUFF;
5             NOD = CLUSUTIL_FIND_NOD_BY_CSID (.RQCB [RQCB_L_CSID]);
5             IF .NOD EQL 0
5             THEN
6                 BEGIN
```

```
  6                             WRITE_LOG_FILE (SHARE_FAO_BUFFER (%ASCID 'Unable to find NOD for CSID !XL', .RQCB [RQCB_L_CS
  6                             DEALLOCATE_RQCB (.RQCB);
  6                             RETURN FALSE;
  5                             END;
P 5                     $FAO (%ASCID '!AD!%D!AD    (from node !6AS at !AD)!AD', FAO_DESC, FAO_DESC,
  5                                             21, .PTR, 0, 13, .PTR+44, NOD [NOD_Q_NAME_DESC], 23, .PTR+21, .LEN-57, .
  5                     LEN = .FAO_DESC [0];
  5                     PTR = FAO_BUFF;
  4                     END;
  3                 END;
  3         MCB [MCB_L_TEXTLEN] = .LEN;
  4         IF NOT (STATUS = OPC$GET_VM (MCB [MCB_L_TEXTLEN], MCB [MCB_L_TEXTPTR]))
  3         THEN
  3             $signal_stop (.STATUS);
  3         CH$MOVE (.LEN, .PTR, .MCB [MCB_L_TEXTPTR]);                 ! Copy the message
  3         PTR = .NEXT;                                                ! Update the output pointer
  2         END;
  2     !
  2     ! If the original had an operator name, make a new operator name
  2     !
  2     IF (LEN = .RQCBUF [RQCB_L_OPER_LEN]) NEQ 0
  2     THEN
  3         BEGIN
  3         LOCAL
  3             NEXT;
  3         IF (NEXT = .LEN + .PTR) GTRU .EOB
  3         THEN
  4             BEGIN
  4             DEALLOCATE_RQCB (.RQCB);
  4             RETURN FALSE;
  3             END;
  4         IF NOT (STATUS = OPC$GET_VM (RQCB [RQCB_L_OPER_LEN], RQCB [RQCB_L_OPER_PTR]))
  3         THEN
  3             $signal_stop (.STATUS);
  3         CH$MOVE (.LEN, .PTR, .RQCB [RQCB_L_OPER_PTR]);              ! Copy the message
  3         PTR = .NEXT;                                                ! Update the output pointer
  2         END;
  2     !
  2     ! If the original had text field, make a new one
  2     !
  2     IF (LEN = .RQCBUF [RQCB_L_TEXT_LEN]) NEQ 0
  2     THEN
  3         BEGIN
  3         LOCAL
  3             NEXT;
  3         IF (NEXT = .LEN + .PTR) GTRU .EOB
  3         THEN
  4             BEGIN
  4             DEALLOCATE_RQCB (.RQCB);
  4             RETURN FALSE;
  3             END;
  4         IF NOT (STATUS = OPC$GET_VM (RQCB [RQCB_L_TEXT_LEN], RQCB [RQCB_L_TEXT_PTR]))
  3         THEN
  3             $signal_stop (.STATUS);
  3         CH$MOVE (.LEN, .PTR, .RQCB [RQCB_L_TEXT_PTR]);              ! Copy the message
  3         PTR = .NEXT;                                                ! Update the output pointer
  2         END;
```

```
;   637            0631  2 !
;   638            0632  2 ! Set the return RQCB to the one we allocated
;   639            0633  2 !
;   640            0634  2 .RET_RQCB = .RQCB;
;   641            0635  2 !
;   642            0636  2 RETURN TRUE;
;   643            0637  1 END;                                                        ! End of CLUSMSG_CONV_CLM_RQCB


                                                                               .PSECT  $PLIT$,NOWRT,NOEXE,2

50 4F 20 20 25 25 25 25 25 25 25 25 25 25 25 0006C P.AAI:  .ASCII  \%%%%%%%%%%%  OPCOM  \                       ;
                                 20 20 4D 4F 43 0007B                                                            ;
20 64 6E 69 66 20 6F 74 20 65 6C 62 61 6E 55 00080 P.AAK:  .ASCII  \Unable to find NOD for CSID !XL\<0>         ;
58 21 20 44 49 53 43 20 72 6F 66 20 44 4F 4E 0008F                                                            ;
                                          00 4C 0009E                                                            ;
                                    010E001F  000A0 P.AAJ:  .LONG   17694751                                     ;
                                    00000000' 000A4         .ADDRESS P.AAK                                       ;
66 28 20 20 20 20 44 41 21 44 25 21 44 41 21 000A8 P.AAM:  .ASCII  \!AD!%D!AD     (from node !6AS at !AD)!AD-   ;
61 20 53 41 36 21 20 65 64 6F 6E 20 6D 6F 72 000B7          \<0>                                                 ;
         00 44 41 21 29 44 41 21 20 74 000C6                                                            ;
                                    010E0027  000D0 P.AAL:  .LONG   17694759                                     ;
                                    00000000' 000D4         .ADDRESS P.AAM                                       ;

                                                            .EXTRN  SYS$FAO, OPC$GET_VM
                                                            .EXTRN  LIB$STOP

                                                            .PSECT  $CODE$,NOWRT,2

                                          0FFC 00000         .ENTRY  CLUSMSG_CONV_CLM_RQCB, Save R2,R3,R4,R5,R6,-; 0460
                                                                     R7,R8,R9,R10,R11                            ;
                       5E     F7EC  CE  9E 00002         MOVAB   -2068(SP), SP                                ;
                                08  BC  D4 00007         CLRL    @RET_RQCB                                    ; 0506
                       5B       04  AC  D0 0000A         MOVL    CLM, R11                                     ; 0510
                       57       0C  AB  9E 0000E         MOVAB   12(R11), RQCBUF                              ;
                0094   8F       08  A7  B1 00012         CMPW    8(RQCBUF), #148                              ; 0511
                                0A  12 00018             BNEQ    1$                                           ;
00000000G 8F     0A  A7        08       00  ED 0001A     CMPZV   #0, #8, 10(RQCBUF), #RQCB_K_TYPE              ; 0513
                                03  13 00024 1$:         BEQL    2$                                           ;
                              0172  31 00026             BRW     17$                                          ;
                       04       AE  9F 00029 2$:         PUSHAB  RQCB                                         ; 0520
                   00000000G    8F  DD 0002C             PUSHL   #RQCB_K_TYPE                                 ;
                0000G  CF       02  FB 00032             CALLS   #2, ALLOCATE_DS                              ;
                       58       04  AE  D0 00037         MOVL    RQCB, R8                                     ; 0521
         10  A8    10  A7     0084  8F  28 0003B         MOVC3   #132, 16(RQCBUF), 16(R8)                      ;
                       56     00A8  CB  9E 00043         MOVAB   168(R11), PTR                                ; 0525
                       50       04  AB  3C 00048         MOVZWL  4(R11), R0                                   ; 0526
             6E        5B       50  C1 0004C             ADDL3   R0, R11, EOB                                 ;
             5A        6C       A7  D0 00050             MOVL    108(RQCBUF), LEN                             ; 0530
                                03  12 00054             BNEQ    3$                                           ;
                              00D1  31 00056             BRW     10$                                          ;
             59        5A       56  C1 00059 3$:         ADDL3   PTR, LEN, NEXT                               ; 0539
                       6E       59  D1 0005D             CMPL    NEXT, EOB                                    ;
                                76  1A 00060             BGTRU   7$                                           ;
                       08       AE  9F 00062             PUSHAB  MCB                                          ; 0545
                   00000000G    8F  DD 00065             PUSHL   #MCB_K_TYPE                                  ;
```

```
                  0000G   CF            02 FB 0006B          CALLS   #2, ALLOCATE_DS                    :
                          54      08    AE D0 00070          MOVL    MCB, R4                            : 0546
                    6C    A8            54 D0 00074          MOVL    R4, 108(R8)                        :
                    24    A4            58 D0 00078          MOVL    R8, 36(R4)                         : 0547
                    2C    A4    00A0    CB D0 0007C          MOVL    160(R11), 44(R4)                   : 0548
                    28    A4    00A4    CB D0 00082          MOVL    164(R11), 40(R4)                   : 0549
          01    A6 0000'  CF            14 29 00088          CMPC3   #20, P.AAI, 1(PTR)                 : 0556
                                        7C 12 0008F          BNEQ    9$                                 :
                66        5A            28 3A 00091          LOCC    #40, LEN, (PTR)                    : 0561
                                        02 12 00095          BNEQ    4$                                 :
                                        51 D4 00097          CLRL    R1                                 :
                          52            51 D0 00099 4$:      MOVL    R1, PAR                            :
                66        5A            0D 3A 0009C          LOCC    #13, LEN, (PTR)                    : 0562
                                        02 12 000A0          BNEQ    5$                                 :
                                        51 D4 000A2          CLRL    R1                                 :
                                        52 D5 000A4 5$:      TSTL    PAR                                : 0563
                                        05 13 000A6          BEQL    6$                                 :
                          51            52 D1 000A8          CMPL    PAR, CR                            : 0565
                                        60 15 000AB          BLEQ    9$                                 :
                    F8    AD    0800    8F 3C 000AD 6$:      MOVZWL  #2048, FAO_DESC                    : 0568
                    FC    AD      0C    AE 9E 000B3          MOVAB   FAO_BUFF, FAO_DESC+4               : 0569
                                  14    A8 DD 000B8          PUSHL   20(R8)                             : 0570
                  0000G   CF            01 FB 000BB          CALLS   #1, CLUSUTIL_FIND_NOD_BY_CSID      :
                          52            50 D0 000C0          MOVL    R0, NOD                            :
                                        16 12 000C3          BNEQ    8$                                 : 0571
                                  14    A8 DD 000C5          PUSHL   20(R8)                             : 0574
                               0000'    CF 9F 000C8          PUSHAB  P.AAJ                              :
                  0000G   CF            02 FB 000CC          CALLS   #2, SHARE_FAO_BUFFER               :
                                        50 DD 000D1          PUSHL   R0                                 :
                  0000G   CF            01 FB 000D3          CALLS   #1, WRITE_LOG_FILE                 :
                                      0089 31 000D8 7$:      BRW     12$                                : 0575
                                  39    A6 9F 000DB 8$:      PUSHAB  57(PTR)                            : 0579
                                  C7    AA 9F 000DE          PUSHAB  -57(LEN)                           :
                                  15    A6 9F 000E1          PUSHAB  21(PTR)                            :
                                        17 DD 000E4          PUSHL   #23                                :
                                  30    A2 9F 000E6          PUSHAB  48(NOD)                            :
                                  2C    A6 9F 000E9          PUSHAB  44(PTR)                            :
                                        0D DD 000EC          PUSHL   #13                                :
                                        7E D4 000EE          CLRL    -(SP)                              :
                                        56 DD 000F0          PUSHL   PTR                                :
                                        15 DD 000F2          PUSHL   #21                                :
                                  F8    AD 9F 000F4          PUSHAB  FAO_DESC                           :
                                  F8    AD 9F 000F7          PUSHAB  FAO_DESC                           :
                               0000'    CF 9F 000FA          PUSHAB  P.AAL                              :
              00000000G   00            0D FB 000FE          CALLS   #13, SYS$FAO                       :
                          5A      F8    AD D0 00105          MOVL    FAO_DESC, LEN                      : 0580
                          56      0C    AE 9E 00109          MOVAB   FAO_BUFF, PTR                      : 0581
                    30    A4            5A D0 0010D 9$:      MOVL    LEN, 48(R4)                        : 0584
                                  34    A4 9F 00111          PUSHAB  52(R4)                             : 0585
                                  30    A4 9F 00114          PUSHAB  48(R4)                             :
                  0000G   CF            02 FB 00117          CALLS   #2, OPC$GET_VM                     :
                          5B            50 D0 0011C          MOVL    R0, STATUS                         :
                          5E            5B E9 0011F          BLBC    STATUS, 14$                        :
          34    B4        66            5A 28 00122          MOVC3   LEN, (PTR), @52(R4)                : 0588
                          56            59 D0 00127          MOVL    NEXT, PTR                          : 0589
                          5A      7C    A7 D0 0012A 10$:     MOVL    124(RQCBUF), LEN                   : 0594
                                        24 13 0012E          BEQL    11$                                :
```

```
             59               5A           56  C1 00130          ADDL3   PTR, LEN, NEXT                 ; 0599
                              6E           59  D1 00134          CMPL    NEXT, EOB                      ;
                                           2B  1A 00137          BGTRU   12$                            ;
                                    0080   C8  9F 00139          PUSHAB  128(R8)                        ; 0605
                                      7C   A8  9F 0013D          PUSHAB  124(R8)                        ;
                      0000G   CF           02  FB 00140          CALLS   #2, OPC$GET_VM                 ;
                              5B           50  D0 00145          MOVL    R0, STATUS                     ;
                              35           5B  E9 00148          BLBC    STATUS, 14$                    ;
      0080   D8               66           5A  28 0014B          MOVC3   LEN, (PTR), @128(R8)           ; 0608
                              56           59  D0 00151          MOVL    NEXT, PTR                      ; 0609
                              5A    0084   C7  D0 00154 11$:     MOVL    132(RQCBUF), LEN               ; 0614
                                           38  13 00159          BEQL    16$                            ;
             57               5A           56  C1 0015B          ADDL3   PTR, LEN, NEXT                 ; 0619
                              6E           57  D1 0015F          CMPL    NEXT, EOB                      ;
                                           09  1B 00162          BLEQU   13$                            ;
                                           58  DD 00164 12$:     PUSHL   R8                             ; 0622
                      0000G   CF           01  FB 00166          CALLS   #1, DEALLOCATE_RQCB            ;
                                           2E  11 0016B          BRB     17$                            ; 0623
                                    0088   C8  9F 0016D 13$:     PUSHAB  136(R8)                        ; 0625
                                    0084   C8  9F 00171          PUSHAB  132(R8)                        ;
                      0000G   CF           02  FB 00175          CALLS   #2, OPC$GET_VM                 ;
                              5B           50  D0 0017A          MOVL    R0, STATUS                     ;
                              0A           5B  E8 0017D          BLBS    STATUS, 15$                    ;
                                           5B  DD 00180 14$:     PUSHL   STATUS                         ; 0627
                  00000000G   00           01  FB 00182          CALLS   #1, LIB$STOP                   ;
                                               04 00189          RET                                    ;
      0088   D8               66           5A  28 0018A 15$:     MOVC3   LEN, (PTR), @136(R8)           ; 0628
                              56           57  D0 00190          MOVL    NEXT, PTR                      ; 0629
                        08    BC           58  D0 00193 16$:     MOVL    R8, @RET_RQCB                  ; 0634
                              50           01  D0 00197          MOVL    #1, R0                         ; 0636
                                               04 0019A          RET                                    ;
                                           50  D4 0019B 17$:     CLRL    R0                             ; 0637
                                               04 0019D          RET                                    ;

; Routine Size:  414 bytes,    Routine Base:  $CODE$ + 018C
```

```
1 GLOBAL ROUTINE CLUSMSG_HANDLER (buffer_desc : $ref_bblock) : NOVALUE =
1
1 !++
1 ! Functional description:
1 !
1 !       This routine processes all messages alleged to have come from remote nodes (plus local broadcasts).
1 !
1 ! Input:
1 !
1 !       BUFFER_DESC : The address of a quadword buffer descriptor that
1 !                     describes the buffer containing the message.
1 !
1 ! Implicit Input:
1 !
1 !       None.
1 !
1 ! Output:
1 !
1 !       None.
1 !
1 ! Implict output:
1 !
1 !       None.
1 !
1 ! Side effects:
1 !
1 !       None.
1 !
1 ! Routine value:
1 !
1 !       None.
1 !--
1
2 BEGIN                                                 ! Start of CLUSMSG_HANDLER
2
2 LOCAL
2       len,                                            ! Length of message without the $SNDOPR header
2       msg             : $ref_bblock,                  ! Pointer to reply command message
2       status;
2 !
2 ! Get a pointer to the regular part of the message, and compute the length.
2 !
2 msg = .buffer_desc [dsc$a_pointer] + opc$k_comhdrsiz;   ! Init the message pointer
2 len = .buffer_desc [dsc$w_length] - opc$k_comhdrsiz;    ! Init the message pointer
2 !
2 ! Check the version number of the message.  If the message is from any other version,
2 ! simply ignore it.
2 !
2 IF .msg [clm_b_sw_version] NEQ opc$k_sw_version
2 THEN
2     RETURN DUMP_LOG_FILE (.BUFFER_DESC, %ASCID 'clm software mismatch');
2 !
2 ! Check the actual length of the message vs. the length stored in the
2 ! message.  If any difference, ignore the message
2 !
2 IF .msg [clm_w_length] NEQ .len
2 THEN
```

```
2       RETURN DUMP_LOG_FILE (.BUFFER_DESC, %ASCID 'clm length mismatch');
2  !
2  ! Perform some privilege and sanity checks on CLM messages
2  !
2  IF .msg [clm_b_clm_code] NEQ clm__rpybrd_local  ! Local replies are checked in CLUSREPLY module
2  THEN
3      BEGIN
3      BIND
3          hdr = .buffer_desc [dsc$a_pointer] : $bblock;   ! Start of $sndopr header
3      !
3      ! If not in a cluster, nothing to do but shout
3      !
3      IF NOT .GLOBAL_STATUS [GBLSTS_K_IN_VAXcluster]
3      THEN
3          RETURN DUMP_LOG_FILE (.BUFFER_DESC, %ASCID 'clm message in non-cluster');
3      !
3      ! Try to make sure that this is coming from the CLUSTER_SERVER process.  Since process name is
3      ! not (yet) part of the $SNDOPR header, we will check that the sender has both the UIC [1,4] and
3      ! has all privileges enabled.  This isn't completely solid, but someone with SETPRV would probably
3      ! be able to circumvent any check we could make.
3      !
3      IF   .hdr [4,0,32,0] NEQ -1                  ! First longword of priv mask in $sndopr header
3        OR
3          .hdr [8,0,32,0] NEQ -1                  ! Second longword of privs
3        OR
3          .hdr [12,0,32,0] NEQ %X'00010004'       ! UIC of [1,4]
3      THEN
3          RETURN DUMP_LOG_FILE (.BUFFER_DESC, %ASCID 'clm privilege violation');
3      !
3      ! Find the sending node in the database.  If we don't see it, then reconfigure.  If we
3      ! still do not see it after a reconfigure, then discard the message.  It is most likely
3      ! from a node which has crashed and rebooted.
3      !
3      IF CLUSUTIL_FIND_NOD_BY_CSID (.msg [clm_l_csid]) EQL 0
3      THEN
4          BEGIN
4          CLUSUTIL_CONFIGURE ();                                  ! Might find the node
4          IF CLUSUTIL_FIND_NOD_BY_CSID (.msg [clm_l_csid]) EQL 0
4          THEN
4              RETURN;
3          END;
2      END;
2  !
2  ! Dispatch the request to the proper handler.
2  !
2  CASE .msg [clm_b_clm_code]
2      FROM  0  TO  clm__request_end_mark-1  OF
2      SET
2      !
2      [clm__acknowledgement] :    CLUSMSG_CLM_ACK_HANDLER          (.buffer_desc, .msg, .len);
2      [clm__acknowledge_please] : CLUSMSG_CLM_ACK_PLEASE_HANDLER   (.buffer_desc, .msg, .len);
2      [clm__cancel] :             CANCEL_CLM_HANDLER               (.buffer_desc, .msg, .len);
2      [clm__check_operator] :     OPRENABLE_CLM_HANDLER            (.buffer_desc, .msg, .len);
2      [clm__check_request] :      REQUEST_CLM_CHECK_HANDLER        (.buffer_desc, .msg, .len);
2      [clm__clumbx] :             CLUSMSG_CLM_NOTIFY_HANDLER       (.buffer_desc, .msg, .len);
2      [clm__cluster] :            CLUSMSG_CLM_NOTIFY_HANDLER       (.buffer_desc, .msg, .len);
2      [clm__device] :             CLUSMSG_CLM_NOTIFY_HANDLER       (.buffer_desc, .msg, .len);
```

```
;   759      0752  2         [clm__imp_disable] :          OPERUTIL_CLM_IMP_DISABLE          (.buffer_desc, .msg, .len);
;   760      0753  2         [clm__oprenable] :            OPRENABLE_CLM_HANDLER             (.buffer_desc, .msg, .len);
;   761      0754  2         [clm__reply] :                REPLY_CLM_HANDLER                 (.buffer_desc, .msg, .len);
;   762      0755  2         [clm__reply_complete] :       CANCEL_CLM_HANDLER                (.buffer_desc, .msg, .len);
;   763      0756  2         [clm__request] :              REQUEST_CLM_HANDLER               (.buffer_desc, .msg, .len);
;   764      0757  2         [clm__rpybrd] :               CLUSREPLY_RPYBRD_HANDLER          (.buffer_desc, .msg, .len);
;   765      0758  2         [clm__rpybrd_local] :         CLUSREPLY_RPYBRD_LOCAL_HANDLER    (.buffer_desc, .msg, .len);
;   766      0759  2         [clm__rpynot] :               CLUSREPLY_RPYNOT_HANDLER          (.buffer_desc, .msg, .len);
;   767      0760  2         [clm__security] :             CLUSMSG_CLM_NOTIFY_HANDLER        (.buffer_desc, .msg, .len);
;   768      0761  2         [clm__shutdown] :             SHUTDOWN_CLM_HANDLER              (.buffer_desc, .msg, .len);
;   769      0762  2         !
;   770      0763  2         ! Let the unknown message handler figure out what to do with it.
;   771      0764  2         !
;   772      0765  2         [INRANGE,OUTRANGE] :          DUMP_LOG_FILE (.BUFFER_DESC, %ASCID 'unknown CLM_CODE in message');
;   773      0766  2         TES;
;   774      0767  2
;   775      0768  2  RETURN;
;   776      0769  1  END;                                                    ! End of CLUSMSG_HANDLER



                                                                              .PSECT  $PLIT$,NOWRT,NOEXE,2

69 6D 20 65 72 61 77 74 66 6F 73 20 6D 6C 63  000D8 P.AAO:   .ASCII  \clm software mismatch\<0><0><0>         ;
                  00 00 00 68 63 74 61 6D 73  000E7                                                           ;
                                    010E0015  000F0 P.AAN:   .LONG   17694741                                 ;
                                   00000000'  000F4          .ADDRESS P.AAO                                   ;
6D 73 69 6D 20 68 74 67 6E 65 6C 20 6D 6C 63  000F8 P.AAQ:   .ASCII  \clm length mismatch\<0>                 ;
                              00 68 63 74 61  00107                                                           ;
                                    010E0013  0010C P.AAP:   .LONG   17694739                                 ;
                                   00000000'  00110          .ADDRESS P.AAQ                                   ;
20 6E 69 20 65 67 61 73 73 65 6D 20 6D 6C 63  00114 P.AAS:   .ASCII  \clm message in non-cluster\<0><0>       ;
      00 00 72 65 74 73 75 6C 63 2D 6E 6F 6E  00123                                                           ;
                                    010E001A  00130 P.AAR:   .LONG   17694746                                 ;
                                   00000000'  00134          .ADDRESS P.AAS                                   ;
76 20 65 67 65 6C 69 76 69 72 70 20 6D 6C 63  00138 P.AAU:   .ASCII  \clm privilege violation\<0>             ;
                  00 6E 6F 69 74 61 6C 6F 69  00147                                                           ;
                                    010E0017  00150 P.AAT:   .LONG   17694743                                 ;
                                   00000000'  00154          .ADDRESS P.AAU                                   ;
44 4F 43 5F 4D 4C 43 20 6E 77 6F 6E 6B 6E 75  00158 P.AAW:   .ASCII  \unknown CLM_CODE in message\<0>         ;
      00 65 67 61 73 73 65 6D 20 6E 69 20 45  00167                                                           ;
                                    010E001B  00174 P.AAV:   .LONG   17694747                                 ;
                                   00000000'  00178          .ADDRESS P.AAW                                   ;



                                                                              .PSECT  $CODE$,NOWRT,2

                                        003C  00000          .ENTRY  CLUSMSG_HANDLER, Save R2,R3,R4,R5       ; 0638
                        54       04   AC  D0  00002          MOVL    BUFFER_DESC, R4                          ; 0680
            52       04 A4            26  C1  00006          ADDL3   #38, 4(R4), MSG                          ;
                        55            64  3C  0000B          MOVZWL  (R4), LEN                                ; 0681
                        55            26  C2  0000E          SUBL2   #38, LEN                                 ;
                        09       03   A2  91  00011          CMPB    3(MSG), #9                               ; 0686
                                      06  13  00015          BEQL    1$                                       ;
                              0000'   CF  9F  00017          PUSHAB  P.AAN                                    ; 0688
                                      45  11  0001B          BRB     5$                                       ;
```

```
                                                                    L  6
OPC$CLUSMSG                                                        16-Sep-1984 01:21:35    VAX-11 Bliss-32 V4.0-742                Page  25
V04-000             CLUSMSG_CONV_CLM_RQCB (CLM, RET_RQCB)          14-Sep-1984 12:50:37    [OPCOM.SRC]CLUSMSG.B32;1                      (8)

        55       04   A2              10           00  ED 0001D 1$:    CMPZV   #0, #16, 4(MSG), LEN                          ; 0693
                                                   06  13 00023        BEQL    2$                                            ;
                                         0000'     CF  9F 00025        PUSHAB  P.AAP                                         ; 0695
                                                   37  11 00029        BRB     5$                                            ;
                                      10      01   A2  91 0002B 2$:    CMPB    1(MSG), #16                                   ; 0699
                                                   51  13 0002F        BEQL    7$                                            ;
                                      53      04   A4  D0 00031        MOVL    4(R4), R3                                     ; 0703
                                      06   0000G   CF  E8 00035        BLBS    GLOBAL_STATUS+1, 3$                           ; 0707
                                         0000'     CF  9F 0003A        PUSHAB  P.AAR                                         ; 0709
                                                   73  11 0003E        BRB     10$                                           ;
                        FFFFFFFF      8F      04   A3  D1 00040 3$:    CMPL    4(R3), #-1                                    ; 0716
                                                   14  12 00048        BNEQ    4$                                            ;
                        FFFFFFFF      8F      08   A3  D1 0004A        CMPL    8(R3), #-1                                    ; 0718
                                                   0A  12 00052        BNEQ    4$                                            ;
                        00010004      8F      0C   A3  D1 00054        CMPL    12(R3), #65540                                ; 0720
                                                   06  13 0005C        BEQL    6$                                            ;
                                         0000'     CF  9F 0005E 4$:    PUSHAB  P.AAT                                         ; 0722
                                                   4F  11 00062 5$:    BRB     10$                                           ;
                                              08   A2  DD 00064 6$:    PUSHL   8(MSG)                                        ; 0728
                            0000G     CF           01  FB 00067        CALLS   #1, CLUSUTIL_FIND_NOD_BY_CSID                 ;
                                                   50  D5 0006C        TSTL    R0                                            ;
                                                   12  12 0006E        BNEQ    7$                                            ;
                            0000G     CF           00  FB 00070        CALLS   #0, CLUSUTIL_CONFIGURE                        ; 0731
                                              08   A2  DD 00075        PUSHL   8(MSG)                                        ; 0732
                            0000G     CF           01  FB 00078        CALLS   #1, CLUSUTIL_FIND_NOD_BY_CSID                 ;
                                                   50  D5 0007D        TSTL    R0                                            ;
                                                   01  12 0007F        BNEQ    7$                                            ;
                                                       04 00081        RET                                                   ;
                13                    00      01   A2  8F 00082 7$:    CASEB   1(MSG), #0, #19                               ; 0740
    0070        003E         0034            0028         00087 8$:    .WORD   9$-8$,-                                       ;
    00A2        00A2         0048            005C         0008F                11$-8$,-                                      ;
    005C        0052         0028            00A2         00097                12$-8$,-                                      ;
    0084        007A         0070            0066         0009F                17$-8$,-                                      ;
    00AC        00A2         0098            008E         000A7                15$-8$,-                                      ;
                                                                               13$-8$,-                                      ;
                                                                               22$-8$,-                                      ;
                                                                               22$-8$,-                                      ;
                                                                               22$-8$,-                                      ;
                                                                               9$-8$,-                                       ;
                                                                               14$-8$,-                                      ;
                                                                               15$-8$,-                                      ;
                                                                               16$-8$,-                                      ;
                                                                               17$-8$,-                                      ;
                                                                               18$-8$,-                                      ;
                                                                               19$-8$,-                                      ;
                                                                               20$-8$,-                                      ;
                                                                               21$-8$,-                                      ;
                                                                               22$-8$,-                                      ;
                                                                               23$-8$                                        ;
                                         0000'     CF  9F 000AF 9$:    PUSHAB  P.AAV                                         ; 0765
                                                   54  DD 000B3 10$:   PUSHL   R4                                            ;
                            0000G     CF           02  FB 000B5        CALLS   #2, DUMP_LOG_FILE                             ;
                                                       04 000BA        RET                                                   ;
                                                   24  BB 000BB 11$:   PUSHR   #^M<R2,R5>                                    ; 0744
                                                   54  DD 000BD        PUSHL   R4                                            ;
                             FC74     CF           03  FB 000BF        CALLS   #3, CLUSMSG_CLM_ACK_HANDLER                   ;
                                                       04 000C4        RET                                                   ;
```

```
                                   24  BB 000C5 12$:         PUSHR    #^M<R2,R5>                                 ; 0745
                                   54  DD 000C7              PUSHL    R4                                         ;
                    FC9E   CF      03  FB 000C9              CALLS    #3, CLUSMSG_CLM_ACK_PLEASE_HANDLER         ;
                                       04 000CE              RET                                                 ;
                                   24  BB 000CF 13$:         PUSHR    #^M<R2,R5>                                 ; 0748
                                   54  DD 000D1              PUSHL    R4                                         ;
                    0000G  CF      03  FB 000D3              CALLS    #3, REQUEST_CLM_CHECK_HANDLER              ;
                                       04 000D8              RET                                                 ;
                                   24  BB 000D9 14$:         PUSHR    #^M<R2,R5>                                 ; 0752
                                   54  DD 000DB              PUSHL    R4                                         ;
                    0000G  CF      03  FB 000DD              CALLS    #3, OPERUTIL_CLM_IMP_DISABLE               ;
                                       04 000E2              RET                                                 ;
                                   24  BB 000E3 15$:         PUSHR    #^M<R2,R5>                                 ; 0753
                                   54  DD 000E5              PUSHL    R4                                         ;
                    0000G  CF      03  FB 000E7              CALLS    #3, OPRENABLE_CLM_HANDLER                  ;
                                       04 000EC              RET                                                 ;
                                   24  BB 000ED 16$:         PUSHR    #^M<R2,R5>                                 ; 0754
                                   54  DD 000EF              PUSHL    R4                                         ;
                    0000G  CF      03  FB 000F1              CALLS    #3, REPLY_CLM_HANDLER                      ;
                                       04 000F6              RET                                                 ;
                                   24  BB 000F7 17$:         PUSHR    #^M<R2,R5>                                 ; 0755
                                   54  DD 000F9              PUSHL    R4                                         ;
                    0000G  CF      03  FB 000FB              CALLS    #3, CANCEL_CLM_HANDLER                     ;
                                       04 00100              RET                                                 ;
                                   24  BB 00101 18$:         PUSHR    #^M<R2,R5>                                 ; 0756
                                   54  DD 00103              PUSHL    R4                                         ;
                    0000G  CF      03  FB 00105              CALLS    #3, REQUEST_CLM_HANDLER                    ;
                                       04 0010A              RET                                                 ;
                                   24  BB 0010B 19$:         PUSHR    #^M<R2,R5>                                 ; 0757
                                   54  DD 0010D              PUSHL    R4                                         ;
                    0000G  CF      03  FB 0010F              CALLS    #3, CLUSREPLY_RPYBRD_HANDLER               ;
                                       04 00114              RET                                                 ;
                                   24  BB 00115 20$:         PUSHR    #^M<R2,R5>                                 ; 0758
                                   54  DD 00117              PUSHL    R4                                         ;
                    0000G  CF      03  FB 00119              CALLS    #3, CLUSREPLY_RPYBRD_LOCAL_HANDLER         ;
                                       04 0011E              RET                                                 ;
                                   24  BB 0011F 21$:         PUSHR    #^M<R2,R5>                                 ; 0759
                                   54  DD 00121              PUSHL    R4                                         ;
                    0000G  CF      03  FB 00123              CALLS    #3, CLUSREPLY_RPYNOT_HANDLER               ;
                                       04 00128              RET                                                 ;
                                   24  BB 00129 22$:         PUSHR    #^M<R2,R5>                                 ; 0760
                                   54  DD 0012B              PUSHL    R4                                         ;
                    FCB2   CF      03  FB 0012D              CALLS    #3, CLUSMSG_CLM_NOTIFY_HANDLER             ;
                                       04 00132              RET                                                 ;
                                   24  BB 00133 23$:         PUSHR    #^M<R2,R5>                                 ; 0761
                                   54  DD 00135              PUSHL    R4                                         ;
                    0000G  CF      03  FB 00137              CALLS    #3, SHUTDOWN_CLM_HANDLER                   ;
                                       04 0013C              RET                                                 ; 0769

; Routine Size:  317 bytes,    Routine Base:  $CODE$ + 032A
```

```
1 GLOBAL ROUTINE CLUSMSG_RQCB_SEND (CSID, CLM_CODE, RQCB : $ref_bblock) = %SBTTL 'CLUSMSG_RQCB_SEND (CSID, CLM
1
1 !++
1 ! Functional description:
1 !
1 !       Put an RQCB into a self-relative format, and send it to remote node(s)
1 !
1 ! Input:
1 !
1 !       CSID     - Id of target node, -1 for broadcast to all nodes except local
1 !       CLM_CODE - Secondary operation code
1 !       RQCB     - Address of block
1 !
1 ! Implicit Input:
1 !
1 !       None.
1 !
1 ! Output:
1 !
1 !       None.
1 !
1 ! Implict output:
1 !
1 !       None.
1 !
1 ! Side effects:
1 !
1 !       Messages will be sent to remote nodes.
1 !
1 ! Routine value:
1 !
1 !       Status from comm primitive.
1 !--
1
2 BEGIN                                                  ! Start of CLUSCOMM_SEND
2
2 LOCAL
2       BUFFER          : BLOCK [OPC$K_MAXMESSAGE+RQCB_K_SIZE+256, BYTE],
2       LEN             : LONG,
2       RQCBUF          : $ref_bblock,
2       PTR             : $ref_bblock,
2       FINAL_STAT      : LONG,
2       STATUS          : LONG;
2
2 !
2 ! If not in a cluster we are done, return with success
2 !
2 IF NOT .GLOBAL_STATUS [GBLSTS_K_IN_VAXcluster]
2 THEN
2     RETURN SS$_NORMAL;
2 !
2 ! First thing, make sure that it is an RQCB
2 !
2 IF .RQCB [RQCB_W_SIZE] NEQ RQCB_K_SIZE
2   OR
2    .RQCB [RQCB_B_TYPE] NEQ RQCB_K_TYPE
2 THEN
```

```
2         $signal_stop (OPC$_NOTRQCB);
2     !
2     ! Next thing, copy the entire RQCB to the buffer
2     !
2     RQCBUF = BUFFER [CLMRQCB_T_RQCB_OVERLAY];
2     CH$MOVE (RQCB_K_SIZE, .RQCB, .RQCBUF);
2     !
2     ! Take all of the character strings hanging off the RQCB and append them to
2     ! the end of the buffer.
2     !
2     PTR = BUFFER [CLMRQCB_T_TEXT];
2     IF .RQCBUF [RQCB_L_MCB] NEQ 0
2     THEN
3         BEGIN
3         LOCAL
3             MCB : $ref_bblock;
3         MCB = .RQCBUF [RQCB_L_MCB];
3         BUFFER [CLMRQCB_L_MCB_MSGID] = .MCB [MCB_L_MSGID];  ! Copy message id
3         BUFFER [CLMRQCB_L_MCB_STATUS] = .MCB [MCB_L_STATUS];!  and status
3         LEN = .MCB [MCB_L_TEXTLEN];
3         CH$MOVE (.LEN, .MCB [MCB_L_TEXTPTR], .PTR);         ! Copy the message
3         PTR = .PTR + .LEN;                                  ! Update the output pointer
3         RQCBUF [RQCB_L_MCB] = .LEN;                         ! Replace MCB address with text length
2         END;
2     IF (LEN = .RQCBUF [RQCB_L_OPER_LEN]) NEQ 0
2     THEN
3         BEGIN
3         CH$MOVE (.LEN, .RQCBUF [RQCB_L_OPER_PTR], .PTR);    ! Copy the message
3         PTR = .PTR + .LEN;                                  ! Update the output pointer
2         END;
2     IF (LEN = .RQCBUF [RQCB_L_TEXT_LEN]) NEQ 0
2     THEN
3         BEGIN
4         IF ((.RQCBUF [RQCB_W_MSGTYPE] EQLU MSG$_OPRQST) AND
4             (.RQCBUF [RQCB_B_RQSTCODE] EQLU OPC$_RQ_SECURITY))
3         THEN
3             RQCBUF [RQCB_L_TEXT_LEN] = 0                            ! Don't send raw messages for security alarm
3         ELSE
4             BEGIN
4             CH$MOVE (.LEN, .RQCBUF [RQCB_L_TEXT_PTR], .PTR);        ! Copy the message
4             PTR = .PTR + .LEN;                                      ! Update the output pointer
3             END;
2         END;
2     !
2     ! Zero any remaining address fields, to prevent embarrasing mixups on the remote node.
2     !
2     RQCBUF [RQCB_L_OCD] = 0;
2     RQCBUF [RQCB_L_OPER_PTR] = 0;
2     RQCBUF [RQCB_L_TEXT_PTR] = 0;
2     RQCBUF [RQCB_L_DSBLFLINK] = 0;
2     RQCBUF [RQCB_L_DSBLBLINK] = 0;
2     !
2     ! Put the cluster message header on top of the queue header of the RQCB
2     !
2     LEN = .PTR - BUFFER;                            ! Compute final length
2     BUFFER [CLM_B_RQSTCODE] = OPC$_X_CLUSMSG;
2     BUFFER [CLM_B_CLM_CODE] = .CLM_CODE;            ! Use the input argument
```

```
;   892        0884   2 BUFFER [CLM_B_DS_VERSION] = CLMRQCB_K_DS_VERSION;
;   893        0885   2 BUFFER [CLM_B_SW_VERSION] = OPC$K_SW_VERSION;
;   894        0886   2 BUFFER [CLM_W_LENGTH]    = .LEN;
;   895        0887   2 BUFFER [CLM_W_FILL_1]    = 0;
;   896        0888   2 BUFFER [CLM_L_CSID]      = .LCL_CSID;
;   897        0889   2 !
;   898        0890   2 ! Send it off to the designated target(s)
;   899        0891   2 !
;   900        0892   2 RETURN CLUSCOMM_SEND (.CSID, .LEN, BUFFER);
;   901        0893   1 END;                                                    ! End of CLUSMSG_RQCB_SEND





                                                       01FC 00000          .ENTRY  CLUSMSG_RQCB_SEND, Save R2,R3,R4,R5,R6,R7,- ; 0770
                                                                                    R8                                          ;
                                     5E       F66C   CE   9E 00002          MOVAB   -2452(SP), SP                               ;
                                     04      0000G   CF   E8 00007          BLBS    GLOBAL_STATUS+1, 1$                         ; 0817
                                     50              01   D0 0000C          MOVL    #1, R0                                      ; 0819
                                                          04 0000F          RET                                                 ;
                                     51        0C    AC   D0 00010 1$:      MOVL    RQCB, R1                                    ; 0823
                             0094    8F        08    A1   B1 00014          CMPW    8(R1), #148                                 ;
                                                     0C   12 0001A          BNEQ    2$                                          ;
00000000G  8F        0A    A1              08    00   ED 0001C          CMPZV   #0, #8, 10(R1), #RQCB_K_TYPE                ; 0825
                                                     0E   13 00026          BEQL    3$                                          ;
                                 00058264    8F   DD 00028 2$:      PUSHL   #361060                                     ; 0827
                          00000000G  00              01   FB 0002E          CALLS   #1, LIB$STOP                                ;
                                                          04 00035          RET                                                 ;
                                     56        0C    AE   9E 00036 3$:      MOVAB   BUFFER+12, RQCBUF                           ; 0831
                        66           61      0094    8F   28 0003A          MOVC3   #148, (R1), (RQCBUF)                        ; 0832
                                     58      00A8    CE   9E 00040          MOVAB   BUFFER+168, PTR                             ; 0837
                                               6C    A6   D5 00045          TSTL    108(RQCBUF)                                 ; 0838
                                                     20   13 00048          BEQL    4$                                          ;
                                     50        6C    A6   D0 0004A          MOVL    108(RQCBUF), MCB                            ; 0843
                             00A0    CE        2C    A0   D0 0004E          MOVL    44(MCB), BUFFER+160                         ; 0844
                             00A4    CE        28    A0   D0 00054          MOVL    40(MCB), BUFFER+164                         ; 0845
                                     57        30    A0   D0 0005A          MOVL    48(MCB), LEN                                ; 0846
                        68           34    B0        57   28 0005E          MOVC3   LEN, @52(MCB), (PTR)                        ; 0847
                                     58              57   C0 00063          ADDL2   LEN, PTR                                    ; 0848
                               6C    A6              57   D0 00066          MOVL    LEN, 108(RQCBUF)                            ; 0849
                                     57        7C    A6   D0 0006A 4$:      MOVL    124(RQCBUF), LEN                            ; 0851
                                                     09   13 0006E          BEQL    5$                                          ;
                        68         0080    D6        57   28 00070          MOVC3   LEN, @128(RQCBUF), (PTR)                    ; 0854
                                     58              57   C0 00076          ADDL2   LEN, PTR                                    ; 0855
                                     57      0084    C6   D0 00079 5$:      MOVL    132(RQCBUF), LEN                            ; 0857
                                                     1B   13 0007E          BEQL    7$                                          ;
                                     08        2C    A6   B1 00080          CMPW    44(RQCBUF), #8                              ; 0860
                                                     0C   12 00084          BNEQ    6$                                          ;
                                     07        52    A6   91 00086          CMPB    82(RQCBUF), #7                              ; 0861
                                                     06   12 0008A          BNEQ    6$                                          ;
                                             0084    C6   D4 0008C          CLRL    132(RQCBUF)                                 ; 0863
                                                     09   11 00090          BRB     7$                                          ;
                        68         0088    D6        57   28 00092 6$:      MOVC3   LEN, @136(RQCBUF), (PTR)                    ; 0866
                                     58              57   C0 00098          ADDL2   LEN, PTR                                    ; 0867
                                               24    A6   D4 0009B 7$:      CLRL    36(RQCBUF)                                  ; 0873
                                             0080    C6   D4 0009E          CLRL    128(RQCBUF)                                 ; 0874
```

```
                                  0088   C6  7C 000A2          CLRQ     136(RQCBUF)                          : 0875
                                  0090   C6  D4 000A6          CLRL     144(RQCBUF)                          : 0877
                          50             6E  9E 000AA          MOVAB    BUFFER, R0                           : 0881
              57          58             50  C3 000AD          SUBL3    R0, PTR, LEN                         :
                          6E             13  90 000B1          MOVB     #19, BUFFER                          : 0882
                     01   AE      08     AC  90 000B4          MOVB     CLM_CODE, BUFFER+1                   : 0883
                     02   AE    0902     8F  B0 000B9          MOVW     #2306, BUFFER+2                      : 0884
                     04   AE             57  3C 000BF          MOVZWL   LEN, BUFFER+4                        : 0886
                     08   AE   0000G     CF  D0 000C3          MOVL     LCL_CSID, BUFFER+8                   : 0888
                                4080     8F  BB 000C9          PUSHR    #^M<R7,SP>                           : 0892
                                  04     AC  DD 000CD          PUSHL    CSID                                 :
                  0000G   CF             03  FB 000D0          CALLS    #3, CLUSCOMM_SEND                    :
                                             04 000D5          RET                                           : 0893

; Routine Size:  214 bytes,    Routine Base:  $CODE$ + 0467
```

```
1 GLOBAL ROUTINE CLUSMSG_STATE_SEND (CSID) =
1
1 !++
1 ! Functional description:
1 !
1 !       CLUSMSG_STATE_SEND sends the state of the current OPCOM process to a remote process.
1 !       The state consists of the active operators and active requests.
1 !
1 ! Input:
1 !
1 !       None.
1 !
1 ! Implicit Input:
1 !
1 !       None.
1 !
1 ! Output:
1 !
1 !       None.
1 !
1 ! Implict output:
1 !
1 !       None.
1 !
1 ! Side effects:
1 !
1 !       None.
1 !
1 ! Routine value:
1 !
1 !       None.
1 !--
1
2 BEGIN                                                 ! Start of CLUSMSG_STATE_SEND
2
2 LOCAL
2         RQCB            : $ref_bblock,                ! RQCB data structure
2         OCD             : $ref_bblock,                ! OCD data structure
2         NEXT_OCD        : $ref_bblock,                ! ditto
2         OCD_COUNT       : LONG,                       ! Count of OCDs in list
2         EXIT_STATUS     : LONG,
2         STATUS          : LONG;
2
2 !
2 ! Loop through all requests, and send each of them off
2 !
2 EXIT_STATUS = TRUE;
2 INCR I FROM MIN_SCOPE TO MAX_SCOPE DO
3     BEGIN
3     !
3     ! For each each class of operator (SYSTEM, GROUP, USER) ...
3     !
3     NEXT_OCD = .OCD_VECTOR [(.I-1)*2];              ! Get first OCD in list
3     INCR J FROM 1 TO .OCD_VECTOR [(.I-1)*2+1] DO
4         BEGIN
4         !
4         ! For each OCD in the operator class list...
```

```
4               !
4               OCD = .NEXT_OCD;                         ! Get current OCD address
4               NEXT_OCD = .OCD [OCD_L_FLINK];           ! Get next OCD address
4               RQCB = .OCD [OCD_L_RQSTFLINK];           ! Get first request address
4               WHILE .RQCB NEQ OCD [OCD_L_RQSTFLINK] DO
5                   BEGIN
5                   !
5                   ! For each request in the OCD list...
5                   !
5                   IF NOT IMPLICITLY_CANCELED (.RQCB)
5                   THEN
5                       !
5                       ! The request is still good, send it off to the target(s)
5                       !
6                       IF NOT (STATUS = CLUSMSG_RQCB_SEND (.CSID, CLM__CHECK_REQUEST, .RQCB))
5                       THEN
5                           EXIT_STATUS = .STATUS;
5                   RQCB = .RQCB [RQCB_L_FLINK];                   ! Get next request address
4                   END;
3               END;
2           END;
2   !
2   ! After sweeping through the data base, we may have discovered some implicitly canceled requests and
2   ! implicitly disabled operators.  Process them now.  The requests should be done first, as yet more
2   ! implicitly disabled operators may turn up.
2   !
2   IMPLIED_CANCEL ();
2   IMPLIED_DISABLE ();
2   !
2   ! Send the list of operators off to the world.
2   !
2   INCR I FROM MIN_SCOPE TO MAX_SCOPE DO
3       BEGIN
3       !
3       ! For each each class of operator (SYSTEM, GROUP, USER) ...
3       !
3       NEXT_OCD = .OCD_VECTOR [(.I-1)*2];             ! Get first OCD in list
3       INCR J FROM 1 TO .OCD_VECTOR [(.I-1)*2+1] DO
4           BEGIN
4           !
4           ! For each OCD in the operator class list...
4           !
4           OCD = .NEXT_OCD;                        ! Get current OCD address
4           NEXT_OCD = .OCD [OCD_L_FLINK];          ! Get next OCD address
4           RQCB = .OCD [OCD_L_OPERFLINK];          ! Get first operator address
4           WHILE .RQCB NEQ OCD [OCD_L_OPERFLINK] DO
5               BEGIN
5               !
5               ! Tell the world about this operator
5               !
6               IF NOT (STATUS = CLUSMSG_RQCB_SEND (.CSID, CLM__CHECK_OPERATOR, .RQCB))
5               THEN
5                   EXIT_STATUS = .STATUS;
5               RQCB = .RQCB [RQCB_L_FLINK];        ! Get next operator address
4               END;
3           END;
2       END;
```

```
; 1017             1008  2
; 1018             1009  2 RETURN .EXIT_STATUS;
; 1019             1010  1 END;                                                    ! End of CLUSMSG_STATE_SEND



                                                 0FFC 00000            .ENTRY   CLUSMSG_STATE_SEND, Save R2,R3,R4,R5,R6,R7,-; 0894
                                                                                R8,R9,R10,R11                                :
                             5B 00000000G  8F   D0 00002               MOVL     #MAX_SCOPE, R11                              :
                             5A       0000G  CF   9E 00009               MOVAB    OCD_VECTOR-8, R10                            :
                             59                01   D0 0000E               MOVL     #1, EXIT_STATUS                              : 0940
               52 00000000G  8F                01   C3 00011               SUBL3    #1, #MIN_SCOPE, I                            : 0941
                                               4C   11 00019               BRB      6$                                           :
               50            52                01   78 0001B 1$:           ASHL     #1, I, R0                                    : 0946
                             57              6A40   D0 0001F               MOVL     OCD_VECTOR-8[R0], NEXT_OCD                   :
                             56           04 AA40   D0 00023               MOVL     OCD_VECTOR-4[R0], R6                         : 0947
                                               55   D4 00028               CLRL     J                                            :
                                               37   11 0002A               BRB      5$                                           :
                             53                57   D0 0002C 2$:           MOVL     NEXT_OCD, OCD                                : 0952
                             57                63   D0 0002F               MOVL     (OCD), NEXT_OCD                              : 0953
                             54           3C   A3   D0 00032               MOVL     60(OCD), RQCB                                : 0954
                             50           3C   A3   9E 00036 3$:           MOVAB    60(OCD), R0                                  : 0955
                             50                54   D1 0003A               CMPL     RQCB, R0                                     :
                                               24   13 0003D               BEQL     5$                                           :
                                               54   DD 0003F               PUSHL    RQCB                                         : 0960
                     0000G   CF                01   FB 00041               CALLS    #1, IMPLICITLY_CANCELED                      :
                             15                50   E8 00046               BLBS     R0, 4$                                       :
                                               54   DD 00049               PUSHL    RQCB                                         : 0965
                                               05   DD 0004B               PUSHL    #5                                           :
                                          04   AC   DD 0004D               PUSHL    CSID                                         :
                      FED5   CF                03   FB 00050               CALLS    #3, CLUSMSG_RQCB_SEND                        :
                             58                50   D0 00055               MOVL     R0, STATUS                                   :
                             03                58   E8 00058               BLBS     STATUS, 4$                                   :
                             59                58   D0 0005B               MOVL     STATUS, EXIT_STATUS                          : 0967
                             54                64   D0 0005E 4$:           MOVL     (RQCB), RQCB                                 : 0968
                                               D3   11 00061               BRB      3$                                           : 0955
               C5            55                56   F3 00063 5$:           AOBLEQ   R6, J, 2$                                    : 0947
               B0            52                5B   F3 00067 6$:           AOBLEQ   R11, I, 1$                                   : 0941
                     0000G   CF                00   FB 0006B               CALLS    #0, IMPLIED_CANCEL                           : 0977
                     0000G   CF                00   FB 00070               CALLS    #0, IMPLIED_DISABLE                          : 0978
               52 00000000G  8F                01   C3 00075               SUBL3    #1, #MIN_SCOPE, I                            : 1001
                                               42   11 0007D               BRB      12$                                          :
               50            52                01   78 0007F 7$:           ASHL     #1, I, R0                                    : 0987
                             57              6A40   D0 00083               MOVL     OCD_VECTOR-8[R0], NEXT_OCD                   :
                             56           04 AA40   D0 00087               MOVL     OCD_VECTOR-4[R0], R6                         : 0988
                                               55   D4 0008C               CLRL     J                                            :
                                               2D   11 0008E               BRB      11$                                          :
                             53                57   D0 00090 8$:           MOVL     NEXT_OCD, OCD                                : 0993
                             57                63   D0 00093               MOVL     (OCD), NEXT_OCD                              : 0994
                             54           50   A3   D0 00096               MOVL     80(OCD), RQCB                                : 0995
                             50           50   A3   9E 0009A 9$:           MOVAB    80(OCD), R0                                  : 0996
                             50                54   D1 0009E               CMPL     RQCB, R0                                     :
                                               1A   13 000A1               BEQL     11$                                          :
                                               54   DD 000A3               PUSHL    RQCB                                         : 1001
                                               04   DD 000A5               PUSHL    #4                                           :
```

```
                                04   AC  DD 000A7        PUSHL   CSID                                      ;
                  FE7B    CF         03  FB 000AA        CALLS   #3, CLUSMSG_RQCB_SEND                     ;
                          58         50  D0 000AF        MOVL    R0, STATUS                                ;
                          03         58  E8 000B2        BLBS    STATUS, 10$                               ;
                          59         58  D0 000B5        MOVL    STATUS, EXIT_STATUS                       ; 1003
                          54         64  D0 000B8 10$:   MOVL    (RQCB), RQCB                              ; 1004
                                     DD  11 000BB        BRB     9$                                        ; 0996
              CF          55         56  F3 000BD 11$:   AOBLEQ  R6, J, 8$                                 ; 0988
              BA          52         5B  F3 000C1 12$:   AOBLEQ  R11, I, 7$                                ; 0982
                          50         59  D0 000C5        MOVL    EXIT_STATUS, R0                           ; 1009
                                         04 000C8        RET                                               ; 1010

; Routine Size:  201 bytes,    Routine Base:  $CODE$ + 053D
; 1021           1011  1 END                                                 ! End of module
; 1022           1012  0 ELUDOM



;                               PSECT SUMMARY
;
;       Name                       Bytes                      Attributes
;
;  $CODE$                             1542  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $PLIT$                              380  NOVEC,NOWRT,  RD ,NOEXE,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        9         0          1000          00:01.8
;  _$255$DUA28:[OPCOM.OBJ]OPCOMLIB.L32;1      633       80        12            43          00:00.9



;                                COMMAND QUALIFIERS

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

; Size:          1542 code + 380 data bytes
; Run Time:         00:31.5
; Elapsed Time:     01:36.6
; Lines/CPU Min:     1928
; Lexemes/CPU-Min: 15412
; Memory Used:  195 pages
; Compilation Complete
```

0289 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

CLUMBX
LIS

DEVICE
LIS

LOGEVENT
LIS

CLUSREPLY
LIS

CLUSUTIL
LIS

DEBUG
LIS

CLUSCOMM
LIS

CLUSMSG
LIS