```
   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
BUG
BYP
CAN
CAN
CAN
CHE
CHE


CLU
CLU
CLU
CLU
CLU
CLU
CLU

CLU
CLU
CLU
CLU
CLU
CLU
CLU


CLU
CLU
```

FILEID**OPCOMINI

F 6

```
  OOOOOO    PPPPPPPP      CCCCCCCC    OOOOOO    MM      MM    IIIIII    NN      NN    IIIIII
  OOOOOO    PPPPPPPP      CCCCCCCC    OOOOOO    MM      MM    IIIIII    NN      NN    IIIIII
OO      OO  PP      PP  CC          OO      OO  MMMM  MMMM      II      NN      NN      II
OO      OO  PP      PP  CC          OO      OO  MMMM  MMMM      II      NN      NN      II
OO      OO  PP      PP  CC          OO      OO  MM  MM  MM      II      NNNN    NN      II
OO      OO  PP      PP  CC          OO      OO  MM  MM  MM      II      NNNN    NN      II
OO      OO  PPPPPPPP    CC          OO      OO  MM      MM      II      NN  NN  NN      II
OO      OO  PPPPPPPP    CC          OO      OO  MM      MM      II      NN  NN  NN      II
OO      OO  PP          CC          OO      OO  MM      MM      II      NN    NNNN      II
OO      OO  PP          CC          OO      OO  MM      MM      II      NN    NNNN      II
OO      OO  PP          CC          OO      OO  MM      MM      II      NN      NN      II          ....
OO      OO  PP          CC          OO      OO  MM      MM      II      NN      NN      II          ....
  OOOOOO    PP            CCCCCCCC    OOOOOO    MM      MM    IIIIII    NN      NN    IIIIII        ....
  OOOOOO    PP            CCCCCCCC    OOOOOO    MM      MM    IIIIII    NN      NN    IIIIII        ....

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$OPCOMINIT   (
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 logic to start and initialize OPCOM.
0 !
0 ! Environment:
0 !
0 !       VAX/VMS operating system.
0 !
0 ! Author:
0 !
0 !       Steven T. Jeffreys
0 !
0 ! Creation date:
0 !
0 !       March 10, 1981
0 !
0 ! Revision history:
0 !
0 !       V03-009 CWH3169         CW Hobbs                        5-May-1984
0 !               Second pass for cluster-wide OPCOM:
0 !                - When we come up, ask for ACKs from any nodes that we can
0 !                  see.
0 !                - Use SYI$_NODENAME rathen than SCSNODE so that a useful
0 !                  length is returned
```

```
;    58        0058  0 !
;    59        0059  0 !        V03-003 CWH3003        CW Hobbs                        14-Apr-1984
;    60        0060  0 !                Change SCSNODE syi items to the single item SYI$_SCSNODE.
;    61        0061  0 !
;    62        0062  0 !        V03-002 CWH3002        CW Hobbs                        16-Sep-1983
;    63        0063  0 !                Use jacket routines for VM calls
;    64        0064  0 !
;    65        0065  0 !        V03-001 CWH3001        CW Hobbs                        30-Jul-1983
;    66        0066  0 !                Various and sundry things to make OPCOM distributed
;    67        0067  0 !                across the cluster.
;    68        0068  0 !
;    69        0069  0 ! V02-002        STJ0162        Steven T. Jeffreys,     08-Feb-1982
;    70        0070  0 !                Make references to library routines use general addressing mode.
;    71        0071  0 !
;    72        0072  0 !
;    73        0073  0 !--
;    74        0074  0
;    75        0075  1 BEGIN                                                   ! Start of OPCOMMAIN
;    76        0076  1
;    77        0077  1 LIBRARY 'SYS$LIBRARY:LIB.L32';
;    78        0078  1 LIBRARY 'LIB$:OPCOMLIB';
;    79        0079  1
;    80        0080  1 FORWARD ROUTINE
;    81        0081  1         OPCOM_INIT      : NOVALUE,              ! Initialization routine
;    82        0082  1         KERNEL_SETUP;                           ! Kernel mode setup
;    83        0083  1
;    84        0084  1 BUILTIN
;    85        0085  1         INSQUE,                                         ! Insert entry onto a queue
;    86        0086  1         REMQUE;                                         ! Remove entry from a queue
```

```
1 GLOBAL ROUTINE OPCOM_INIT : NOVALUE =
1
1 !++
1 ! Functional description:
1 !
1 !       This routine will start OPCOM and perform all of
1 !       the necesary initialization to ensure OPCOM's
1 !       correct functioning.
1 !
1 ! Input:
1 !
1 !       None.
1 !
1 ! Implicit Input:
1 !
1 !       The operator mailbox exists and is available.
1 !       The process has CHMKRNL privilege.
1 !
1 ! Output:
1 !
1 !       None.
1 !
1 ! Implict output:
1 !
1 !       None.
1 !
1 ! Side effects:
1 !
1 !       The operator logfile will be created on the system disk.
1 !
1 ! Routine value:
1 !
1 !       SS$_NOPRIV      - The process does not have CHKRNL privilege.
1 !       SS$_DEVALLOC    - The operator mailbox already has a listener.
1 !       SS$_NORMAL      - OPCOM successfully completed its shutdown sequence.
1 !--
1
2 BEGIN                                           ! Start of OPCOM_INIT
2
2 EXTERNAL ROUTINE
2       CLUSMSG_ACK_PLEASE : NOVALUE,            ! Request an ACK from another node
2       CLUSUTIL_CONFIGURE,                      ! Cluster configuration
2       CLUSUTIL_INIT    : NOVALUE,              ! Cluster initializations
2       CLUSUTIL_NODE_MESSAGE : NOVALUE,         ! Tell about cluster initializations
2       OPRENABLE_HANDLER: NOVALUE,              ! Enable operator terminal
2       LOGFILE_HANDLER  : NOVALUE,              ! Initialize system logfile
2       SHARE_INIT_OPER_NAME : NOVALUE,          ! Initialize the operator name vector
2       TIME_STAMP       : NOVALUE;              ! Issue periodic timestamp
2
2 EXTERNAL LITERAL
2       OCD_K_TYPE,                              ! OCD structure type
2       MIN_SCOPE,                               ! Min scope value
2       MAX_SCOPE,                               ! Max scope value
2       MIN_DS_TYPE,                             ! Min DS_TYPE value
2       MAX_DS_TYPE;                             ! Max DS_TYPE value
2
2 EXTERNAL
```

```
;   145        0144   2           SYI_SWPOUTPGCNT : LONG,
;   146        0145   2           GLOBAL_STATUS   : BITVECTOR,           ! Global status flags
;   147        0146   2           OCD_VECTOR      : VECTOR,              ! Pointers to OCD's
;   148        0147   2           SCB_TABLE       : VECTOR,              ! Start of SCB table
;   149        0148   2           WAIT_DELTA      : $bblock,             ! Quadword AST wait time
;   150        0149   2           OPA0_ENABLE_MSG : $bblock,             ! Descriptor of enable message
;   151        0150   2           LOGFILE_MSG     : $bblock,             ! Descriptor of INIT message
;   152        0151   2           OPER_MBX_CHAN   : WORD,                ! Channel to _MBA2:
;   153        0152   2           OPER_MBX_NAME   : $bblock,             ! Quadword descriptor of '_MBA2:'
;   154        0153   2           LCL_NODENAME    : VECTOR,              ! Name of local node (DECnet or VAXcluster)
;   155        0154   2           LCL_NOD         : $ref_bblock,         ! Local node block
;   156        0155   2           LCL_CSID        : LONG,                ! CSID for local node
;   157        0156   2           NOD_HEAD        : VECTOR,              ! Head of node queue
;   158        0157   2           LOGFILE_FAB     : $ref_bblock,         ! RMS control structure
;   159        0158   2           LOGFILE_RAB     : $ref_bblock,         ! RMS control structure
;   160        0159   2           LOGFILE_OPEN    : LONG;                ! Global boolean
;   161        0160   2
;   162        0161   2   LOCAL
;   163        0162   2           IOSB            : $bblock [8],         ! I/O status block
;   164        0163   2           SIZE            : LONG,                ! Size of data structure
;   165        0164   2           TYPE            : LONG,                ! Type of data structure
;   166        0165   2           BLOCK           : $ref_bblock,         ! VM block
;   167        0166   2           SCB             : $ref_bblock,         ! SCB block
;   168        0167   2           PTR             : LONG,
;   169        0168   2           STATUS          : LONG;
;   170        0169   2
;   171        0170   2   OWN
;   172        0171   2       NODENAME_BUF        : VECTOR [16, BYTE],
;   173        0172   2       NODENAME_DESC       : VECTOR [2, LONG] INITIAL (0, NODENAME_BUF),
;   174        0173   2       TRANLOG_DESC        : VECTOR [2, LONG] INITIAL (16, NODENAME_BUF),
;   175        0174   2       SYI_ITEMS           : VECTOR [7, LONG] PRESET (
;   176        0175   2                                 [0] = (SYI$_NODENAME^16 OR 16),
;   177        0176   2                                 [1] = NODENAME_BUF,
;   178        0177   2                                 [2] = NODENAME_DESC [0],
;   179        0178   2                                 [3] = (SYI$_SWPOUTPGCNT^16 OR 4),
;   180        0179   2                                 [4] = SYI_SWPOUTPGCNT,
;   181        0180   2                                 [5] = 0,
;   182        0181   2                                 [6] = 0);
;   183        0182   2
;   184        0183   2   !
;   185        0184   2   ! Assign a channel to the operator mailbox, _MBA2:.  The mailbox is
;   186        0185   2   ! defined by the EXEC module DEVICEDAT, and has an initial reference
;   187        0186   2   ! count of 1.  This means that no one can allocate the device, no one
;   188        0187   2   ! can delete it, and that once we assign a channel to it, the reference
;   189        0188   2   ! count had better be 2.  If the $ASSIGN fails, then give up.
;   190        0189   2   !
;   191        0190   3   IF NOT (STATUS = $ASSIGN (CHAN=OPER_MBX_CHAN, DEVNAM=OPER_MBX_NAME))
;   192        0191   2   THEN
;   193        0192   2       $signal_stop (.STATUS);
;   194        0193   2   !
;   195        0194   2   ! We now have a channel to the operator mailbox.  Now we must go into kernel
;   196        0195   2   ! mode to do some further checking.  We must guaranty that we have CHMKRNL
;   197        0196   2   ! privilege, and that the reference count on the operator mailbox is equal to
;   198        0197   2   ! 2.  While in kernel mode, also set the OPR bit in the default operator UCB,
;   199        0198   2   ! _OPA0:.
;   200        0199   2   !
;   201        0200   3   IF NOT (STATUS = $CMKRNL (ROUTIN=KERNEL_SETUP))
```

```
2 THEN
2     !
2     ! Something is wrong.  The only error that we can tolerate is the
2     ! nonexistence of _OPA0:, which is indicated by a status of SS$_NOSUCHDEV.
2     ! Any other error condition will force us to abort.
2     !
3     IF NOT (.STATUS EQL SS$_NOSUCHDEV)
2     THEN
2         $signal_stop (.STATUS);
2
2 !
2 ! Initialize the vector of operator names
2 !
2 SHARE_INIT_OPER_NAME ();
2
2 !
2 ! Build the look-aside lists for all data structures.
2 !
2 INCR J FROM MIN_DS_TYPE TO MAX_DS_TYPE DO
3     BEGIN
3     SCB  = .SCB_TABLE [.J-1];               ! Get the SCB address
3     SIZE = .SCB [SCB_W_SIZE];               ! Get data structure size
3     TYPE = .J;                              ! Get data structure type
3     INCR I FROM 1 TO .SCB [SCB_W_LAL_COUNT] DO
4         BEGIN
4         !
4         ! Allocate a block of memory, initialize it, and
4         ! put it on the appropriate look-aside list.
4         !
5         IF NOT (STATUS = OPC$GET_VM (SIZE, BLOCK))
4         THEN
4             $signal_stop (.STATUS);
4         CH$FILL (0, .SIZE, .BLOCK);         ! Zero the block
4         BLOCK [HDR_L_FLINK] = BLOCK [HDR_L_FLINK];
4         BLOCK [HDR_L_BLINK] = BLOCK [HDR_L_FLINK];
4         BLOCK [HDR_W_SIZE] = .SIZE;         ! Set block size
4         BLOCK [HDR_B_TYPE] = .TYPE;         ! Set block type
4         BLOCK [HDR_V_LAL] = TRUE;           ! Mark this as an LAL block
4         INSQUE (BLOCK [HDR_L_FLINK], SCB [SCB_L_FLINK]);
3         END;
2     END;
2
2 !
2 ! Do a $GETSYI to get information about the current system
2 !
3 IF NOT (STATUS = $GETSYI (ITMLST=SYI_ITEMS))
2 THEN
2     $signal_stop (.STATUS);
2 !
2 ! If the SCS nodename is null, try to translate SYS$NODE to find the DECnet name.  Remove the "_" and "::"
2 ! from the translated name.
2 !
2 IF .NODENAME_DESC [0] EQL 0
2 THEN
3     BEGIN
4     IF NOT (STATUS = $TRNLOG (LOGNAM=%ASCID 'SYS$NODE', RSLLEN=TRANLOG_DESC, RSLBUF=TRANLOG_DESC, DSBMSK=6))
3     THEN
```

```
;  259         0258  3            $signal_stop (.STATUS);
;  260         0259  3        IF .STATUS EQL SS$_NORMAL                     ! If we translated, remove the underscores and colons
;  261         0260  3        THEN
;  262         0261  4            BEGIN
;  263         0262  4            PTR = CH$FIND_NOT_CH (.TRANLOG_DESC [0], .TRANLOG_DESC [1], %C '_');
;  264         0263  4            IF .PTR NEQ 0
;  265         0264  4            THEN
;  266         0265  5                BEGIN
;  267         0266  5                TRANLOG_DESC [0] = .TRANLOG_DESC [0] - (.PTR - .TRANLOG_DESC [1]);
;  268         0267  5                TRANLOG_DESC [1] = .PTR;
;  269         0268  4                END;
;  270         0269  4            PTR = CH$FIND_CH (.TRANLOG_DESC [0], .TRANLOG_DESC [1], %C ':');
;  271         0270  4            IF .PTR NEQ 0
;  272         0271  4            THEN
;  273         0272  4                TRANLOG_DESC [0] = .PTR - .TRANLOG_DESC [1];
;  274         0273  4            NODENAME_DESC [0] = .TRANLOG_DESC [0];
;  275         0274  4            NODENAME_DESC [1] = .TRANLOG_DESC [1];
;  276         0275  3            END;
;  277         0276  2        END;
;  278         0277  2    !
;  279         0278  2    ! Set the global message
;  280         0279  2    !
;  281         0280  2    IF .NODENAME_DESC [0] NEQ 0
;  282         0281  2    THEN
;  283         0282  3        BEGIN
;  284         0283  3        LCL_NODENAME [0] = .NODENAME_DESC [0];
;  285         0284  3        LCL_NODENAME [1] = .NODENAME_DESC [1];
;  286         0285  2        END;
;  287         0286  2    !
;  288         0287  2    ! Perform cluster initialization functions necessary before can enable
;  289         0288  2    ! operators.
;  290         0289  2    !
;  291         0290  2    CLUSUTIL_INIT ();
;  292         0291  2
;  293         0292  2    !
;  294         0293  2    ! Set up _OPA0: as the default operator.  This is done by calling the enable handler with a the
;  295         0294  2    ! address of a buffer descriptor that points to a predefined enable message.
;  296         0295  2    !
;  297         0296  2    OPRENABLE_HANDLER (OPA0_ENABLE_MSG);
;  298         0297  2
;  299         0298  2    !
;  300         0299  2    ! Open the log file.  This is done by calling the INIT message handler with the address of a buffer
;  301         0300  2    ! descriptor that points to a predefined INIT message.  Set logging enabled.
;  302         0301  2    !
;  303         0302  2    LOGFILE_HANDLER (LOGFILE_MSG);
;  304         0303  2    GLOBAL_STATUS [GBLSTS_K_LOGGING_ENABLED] = TRUE;
;  305         0304  2
;  306         0305  2    !
;  307         0306  2    ! Configure the nodes into the cluster database, and notify any
;  308         0307  2    ! operators.  This should be done after OPA0 and the logfile are ready.
;  309         0308  2    !
;  310         0309  2    IF .GLOBAL_STATUS [GBLSTS_K_IN_VAXcluster]
;  311         0310  2    THEN
;  312         0311  3        BEGIN
;  313         0312  3        CLUSUTIL_NODE_MESSAGE (.LCL_NOD, OPC$_NODE_INIT, FALSE);    ! Tell them we have initialized the local no
;  314         0313  3        IF CLUSUTIL_CONFIGURE ()                                     ! Now bring in everybody else, true if anyon
;  315         0314  3        THEN                                                         !  has been added to the club
```

```
  4          BEGIN
  4          LOCAL
  4              NOD : $REF_BBLOCK;
  4          NOD = .NOD_HEAD [0];                                       ! Start at the head of the node queue
  4          WHILE .NOD NEQ NOD_HEAD [0]                                ! While not pointing at the head
  4          DO
  5              BEGIN
  5              IF .NOD [NOD_L_NODE_CSID] NEQ .LCL_CSID                ! If not the local node
  5              THEN
  5                  CLUSMSG_ACK_PLEASE (.NOD);                         ! Request an ack
  5              NOD = .NOD [NOD_L_FLINK];                              ! Move to the next node
  4              END;
  3          END;
  2      END;
  2  !
  2  ! Set the operator mailbox protection to
  2  ! allow OWNER reads and writes.
  2  !
P 3  IF NOT (STATUS = $QIO  (FUNC = (IO$_SETMODE OR IO$M_SETPROT),
P 3                          CHAN = .OPER_MBX_CHAN,
P 3                          IOSB = IOSB,
P 3                          P2 = READ_WRITE
  3                         ))
  2  THEN
  2      $signal_stop (.STATUS);
  2
  2  !
  2  ! Check the function status.
  2  !
  3  IF NOT (STATUS = .IOSB [0,0,16,0])
  2  THEN
  2      $signal_stop (.STATUS);
  2
  2  !
  2  ! Set the timer ast for the first time stamp message.
  2  ! Also set the associated GLOBAL_STATUS bits.
  2  !
  2  GLOBAL_STATUS [GBLSTS_K_BUSY] = TRUE;
  2  GLOBAL_STATUS [GBLSTS_K_TIMESTAMP_PENDING] = FALSE;
  3  IF NOT (STATUS = $SETIMR (EFN = EFN_K_TIME_STAMP, DAYTIM = WAIT_DELTA, ASTADR = TIME_STAMP))
  2  THEN
  2      $signal_stop (.STATUS);
  2
  1  END;                                                 ! End of OPCOM_INIT

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

                                                          .PSECT  $PLIT$,NOWRT,NOEXE,2

      45  44  4F  4E  24  53  59  53   00000 P.AAB:       .ASCII  \SYS$NODE\
                               010E0008  00008 P.AAA:     .LONG   17694728
                               00000000' 0000C            .ADDRESS P.AAB

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

```
                                00000 NODENAME_BUF:
                                              .BLKB   16
                      00000000  00010 NODENAME_DESC:
                                              .LONG   0
                      00000000' 00014         .ADDRESS NODENAME_BUF                 :
                      00000010  00018 TRANLOG_DESC:
                                              .LONG   16
                      00000000' 0001C         .ADDRESS NODENAME_BUF                 :
                      10D90010  00020 SYI_ITEMS:
                                              .LONG   282656784
            00000000' 00000000' 00024         .ADDRESS NODENAME_BUF, NODENAME_DESC  :
                      103B0004  0002C         .LONG   272302084
                      00000000G 00030         .ADDRESS SYI_SWPOUTPGCNT              :
            00000000  00000000  00034         .LONG   0, 0                          :

                                              .EXTRN  CLUSMSG_ACK_PLEASE
                                              .EXTRN  CLUSUTIL_CONFIGURE
                                              .EXTRN  CLUSUTIL_INIT, CLUSUTIL_NODE_MESSAGE
                                              .EXTRN  OPRENABLE_HANDLER
                                              .EXTRN  LOGFILE_HANDLER
                                              .EXTRN  SHARE_INIT_OPER_NAME
                                              .EXTRN  TIME_STAMP, OCD_K_TYPE
                                              .EXTRN  MIN_SCOPE, MAX_SCOPE
                                              .EXTRN  MIN_DS_TYPE, MAX_DS_TYPE
                                              .EXTRN  SYI_SWPOUTPGCNT
                                              .EXTRN  GLOBAL_STATUS, OCD_VECTOR
                                              .EXTRN  SCB_TABLE, WAIT_DELTA
                                              .EXTRN  OPA0_ENABLE_MSG
                                              .EXTRN  LOGFILE_MSG, OPER_MBX_CHAN
                                              .EXTRN  OPER_MBX_NAME, LCL_NODENAME
                                              .EXTRN  LCL_NOD, LCL_CSID
                                              .EXTRN  NOD_HEAD, LOGFILE_FAB
                                              .EXTRN  LOGFILE_RAB, LOGFILE_OPEN
                                              .EXTRN  SYS$ASSIGN, LIB$STOP
                                              .EXTRN  SYS$CMKRNL, OPC$GET_VM
                                              .EXTRN  SYS$GETSYI, SYS$TRNLOG
                                              .EXTRN  SYS$QIO, SYS$SETIMR

                                              .PSECT  $CODE$,NOWRT,2

                           0FFC 00000         .ENTRY  OPCOM_INIT, Save R2,R3,R4,R5,R6,R7,R8,R9,-   : 0087
                                                      R10,R11                                       :
           5E             14  C2 00002        SUBL2   #20, SP                                       :
                          7E  7C 00005        CLRQ    -(SP)                                         : 0190
                   0000G  CF  9F 00007        PUSHAB  OPER_MBX_CHAN                                 :
                   0000G  CF  9F 0000B        PUSHAB  OPER_MBX_NAME                                 :
00000000G  00             04  FB 0000F        CALLS   #4, SYS$ASSIGN                                :
           59             50  D0 00016        MOVL    R0, STATUS                                    :
           1C             59  E9 00019        BLBC    STATUS, 1$                                    :
                          7E  D4 0001C        CLRL    -(SP)                                         : 0200
                   0000V  CF  9F 0001E        PUSHAB  KERNEL_SETUP                                  :
00000000G  00             02  FB 00022        CALLS   #2, SYS$CMKRNL                                :
           59             50  D0 00029        MOVL    R0, STATUS                                    :
           0C             59  E8 0002C        BLBS    STATUS, 2$                                    :
00000908   8F             59  D1 0002F        CMPL    STATUS, #2312                                 : 0207
                          03  13 00036        BEQL    2$                                            :
                        01AF  31 00038 1$:    BRW     18$                                           :
                           0000G CF           00 FB 0003B 2$:     CALLS    #0, SHARE_INIT_OPER_NAME                ; 0214      ; M
                    58 00000000G 8F           01 C3 00040         SUBL3    #1, #MIN_DS_TYPE, J                     ; 0219      ; M
                                              50 11 00048         BRB      6$                                      ;           ;
                                 56    0000GCF48 D0 0004A 3$:     MOVL     SCB_TABLE-4[J], SCB                     ; 0221      ;
                              08 AE           66 3C 00050         MOVZWL   (SCB), SIZE                             ; 0222      ;
                                 6E           58 D0 00054         MOVL     J, TYPE                                 ; 0223      ;
                                 5B        02 A6 3C 00057         MOVZWL   2(SCB), R11                             ; 0224      ;
                                              5A D4 0005B         CLRL     I                                       ;           ; M
                                              37 11 0005D         BRB      5$                                      ;           ; M
                                           04 AE 9F 0005F 4$:     PUSHAB   BLOCK                                   ; 0230      ; M
                                           0C AE 9F 00062         PUSHAB   SIZE                                    ;           ; M
                           0000G CF           02 FB 00065         CALLS    #2, OPC$GET_VM                          ;           ; M
                                 59           50 D0 0006A         MOVL     R0, STATUS                              ;           ; M
                                 C8           59 E9 0006D         BLBC     STATUS, 1$                              ;           ; M
                                 57        04 AE D0 00070         MOVL     BLOCK, R7                               ; 0233      ; M
        08  AE         00        6E           00 2C 00074         MOVC5    #0, (SP), #0, SIZE, (R7)                ;           ; M
                                              67    0007A                                                          ;           ; M
                                 67           57 D0 0007B         MOVL     R7, (R7)                                ; 0234      ; M
                              04 A7           57 D0 0007E         MOVL     R7, 4(R7)                               ; 0235      ;
                              08 A7        08 AE B0 00082         MOVW     SIZE, 8(R7)                             ; 0236      ;
                              0A A7           6E 90 00087         MOVB     TYPE, 10(R7)                            ; 0237      ;
                              28 A7           01 88 0008B         BISB2    #1, 40(R7)                              ; 0238      ;
                                 50        08 A6 9E 0008F         MOVAB    8(R6), R0                               ; 0239      ;
                                 60           67 0E 00093         INSQUE   (R7), (R0)                              ;           ; M
                       C5        5A           5B F3 00096 5$:     AOBLEQ   R11, I, 4$                              ; 0224      ; M
                       A8        58 00000000G 8F F3 0009A 6$:     AOBLEQ   #MAX_DS_TYPE, J, 3$                     ; 0219      ; M
                                              7E 7C 000A2         CLRQ     -(SP)                                   ; 0246      ; M
                                              7E D4 000A4         CLRL     -(SP)                                   ;           ; M
                                        0000' CF 9F 000A6         PUSHAB   SYI_ITEMS                               ;           ; M
                                              7E 7C 000AA         CLRQ     -(SP)                                   ;           ; M
                                              7E D4 000AC         CLRL     -(SP)                                   ;           ; M
                       00000000G 00           07 FB 000AE         CALLS    #7, SYS$GETSYI                          ;           ; M
                                 59           50 D0 000B5         MOVL     R0, STATUS                              ;           ; M
                                 20           59 E9 000B8         BLBC     STATUS, 7$                              ;           ; M
                                        0000' CF D5 000BB         TSTL     NODENAME_DESC                           ; 0253      ;
                                              65 12 000BF         BNEQ     13$                                     ;           ;
                                              06 DD 000C1         PUSHL    #6                                      ; 0256      ;
                                              7E 7C 000C3         CLRQ     -(SP)                                   ;           ;
                                        0000' CF 9F 000C5         PUSHAB   TRANLOG_DESC                            ;           ;
                                        0000' CF 9F 000C9         PUSHAB   TRANLOG_DESC                            ;           ;
                                        0000' CF 9F 000CD         PUSHAB   P.AAA                                   ;           ;
                       C0000000G 00           06 FB 000D1         CALLS    #6, SYS$TRNLOG                          ;           ;
                                 59           50 D0 000D8         MOVL     R0, STATUS                              ;           ;
                                 03           59 E8 000DB 7$:     BLBS     STATUS, 8$                              ;           ;
                                            0109 31 000DE         BRW      18$                                     ;           ;
                                 01           59 D1 000E1 8$:     CMPL     STATUS, #1                              ; 0259      ;
                                              40 12 000E4         BNEQ     13$                                     ;           ;
                 0000' DF  0000' CF        5F 8F 3B 000E6         SKPC     #95, TRANLOG_DESC, @TRANLOG_DESC+4      ; 0262      ; M
                                              02 12 000EF         BNEQ     9$                                      ;           ; M
                                              51 D4 000F1         CLRL     R1                                      ;           ; M
                                              51 D5 000F3 9$:     TSTL     PTR                                     ; 0263      ; M
                                              10 13 000F5         BEQL     10$                                     ;           ; M
                       50  0000' CF           51 C3 000F7         SUBL3    PTR, TRANLOG_DESC+4, R0                 ; 0266      ; M
                           0000' CF           50 C0 000FD         ADDL2    R0, TRANLOG_DESC                        ;           ; M
                           0000' CF           51 D0 00102         MOVL     PTR, TRANLOG_DESC+4                     ; 0267      ; M
                 0000' DF  0000' CF           3A 3A 00107 10$:    LOCC     #58, TRANLOG_DESC, @TRANLOG_DESC+4      ; 0269      ; M
                                              02 12 0010F         BNEQ     11$                                     ;           ; M
                                              51  D4 00111          CLRL    R1                                       ;        : M
                                              51  D5 00113  11$:    TSTL    PTR                                      ; 0270   : M
                                              08  13 00115          BEQL    12$                                      ;        : M
0000'  CF               51         0000'     CF  C3 00117          SUBL3   TRANLOG_DESC+4, PTR, TRANLOG_DESC        ; 0272   : M
               0000'    CF         0000'     CF  7D 0011F  12$:    MOVQ    TRANLOG_DESC, NODENAME_DESC              ; 0273   : M
                        50         0000'     CF  D0 00126  13$:    MOVL    NODENAME_DESC, R0                        ; 0280   : M
                                              0C  13 0012B          BEQL    14$                                      ;        : M
               0000G    CF                   50  D0 0012D          MOVL    R0, LCL_NODENAME                         ; 0283   : M
               0000G    CF         0000'     CF  D0 00132          MOVL    NODENAME_DESC+4, LCL_NODENAME+4          ; 0284   : M
               0000G    CF                   00  FB 00139  14$:    CALLS   #0, CLUSUTIL_INIT                        ; 0290   : M
                                   0000G     CF  9F 0013E          PUSHAB  OPA0_ENABLE_MSG                          ; 0296   : M
               0000G    CF                   01  FB 00142          CALLS   #1, OPRENABLE_HANDLER                    ;        : M
                                   0000G     CF  9F 00147          PUSHAB  LOGFILE_MSG                              ; 0302   : M
               0000G    CF                   01  FB 0014B          CALLS   #1, LOGFILE_HANDLER                      ;        : M
               0000G    CF                   02  88 00150          BISB2   #2, GLOBAL_STATUS                        ; 0303   : M
                        3C         0000G     CF  E9 00155          BLBC    GLOBAL_STATUS+1, 17$                     ; 0309   : M
                                              7E  D4 0015A          CLRL    -(SP)                                    ; 0312   : M
                              00058233       8F  DD 0015C          PUSHL   #361011                                  ;        : M
                                   0000G     CF  DD 00162          PUSHL   LCL_NOD                                  ;        : M
               0000G    CF                   03  FB 00166          CALLS   #3, CLUSUTIL_NODE_MESSAGE                ;        : M
               0000G    CF                   00  FB 0016B          CALLS   #0, CLUSUTIL_CONFIGURE                   ; 0313   : M
                        23                   50  E9 00170          BLBC    R0, 17$                                  ;        : M
                        52         0000G     CF  D0 00173          MOVL    NOD_HEAD, NOD                            ; 0318   : M
                        50         0000G     CF  9E 00178  15$:    MOVAB   NOD_HEAD, R0                             ; 0319   : M
                        50                   52  D1 0017D          CMPL    NOD, R0                                  ;        : M
                                              14  13 00180          BEQL    17$                                      ;        : M
               0000G    CF            2C     A2  D1 00182          CMPL    44(NOD), LCL_CSID                        ; 0322   : M
                                              07  13 00188          BEQL    16$                                      ;        : M
                                              52  DD 0018A          PUSHL   NOD                                      ; 0324   : M
               0000G    CF                   01  FB 0018C          CALLS   #1, CLUSMSG_ACK_PLEASE                   ;        :
                        52                   62  D0 00191  16$:    MOVL    (NOD), NOD                               ; 0325   :
                                              E2  11 00194          BRB     15$                                      ; 0319   :
                                              7E  7C 00196  17$:    CLRQ    -(SP)                                    ; 0337   :
                                              7E  7C 00198          CLRQ    -(SP)                                    ;        :
                        7E         FF0F      8F  3C 0019A          MOVZWL  #65295, -(SP)                            ;        :
                                              7E  7C 0019F          CLRQ    -(SP)                                    ;        :
                                              7E  D4 001A1          CLRL    -(SP)                                    ;        :
                                      2C     AE  9F 001A3          PUSHAB  IOSB                                     ;        :
                        7E         0223      8F  3C 001A6          MOVZWL  #547, -(SP)                              ;        :
                        7E         0000G     CF  3C 001AB          MOVZWL  OPER_MBX_CHAN, -(SP)                     ;        :
                                              7E  D4 001B0          CLRL    -(SP)                                    ;        :
           00000000G    00                   0C  FB 001B2          CALLS   #12, SYS$QIO                             ;        :
                        59                   50  D0 001B9          MOVL    R0, STATUS                               ;        : M
                        2B                   59  E9 001BC          BLBC    STATUS, 18$                              ;        : M
                        59            0C     AE  3C 001BF          MOVZWL  IOSB, STATUS                             ; 0344   : M
                        24                   59  E9 001C3          BLBC    STATUS, 18$                              ;        : M
               0000G    CF            40     8F  88 001C6          BISB2   #64, GLOBAL_STATUS                       ; 0352   : M
               0000G    CF                   20  8A 001CC          BICB2   #32, GLOBAL_STATUS                       ; 0353   : M
                                              7E  D4 001D1          CLRL    -(SP)                                    ; 0354   : M
                                   0000G     CF  9F 001D3          PUSHAB  TIME_STAMP                               ;        : M
                                   0000G     CF  9F 001D7          PUSHAB  WAIT_DELTA                               ;        : M
                                              04  DD 001DB          PUSHL   #4                                       ;        : M
           00000000G    00                   04  FB 001DD          CALLS   #4, SYS$SETIMR                           ;        : M
                        59                   50  D0 001E4          MOVL    R0, STATUS                               ;        : M
                        09                   59  E8 001E7          BLBS    STATUS, 19$                              ;        : M
                                              59  DD 001EA  18$:    PUSHL   STATUS                                   ; 0356   : M
           00000000G    00                   01  FB 001EC          CALLS   #1, LIB$STOP                             ;        :
```

```
                                           04 001F3 19$:    RET                                      ; 0358

; Routine Size:  500 bytes,    Routine Base:  $CODE$ + 0000

;  360          0359  1
```

```
1 ROUTINE KERNEL_SETUP =
1
1 !++
1 ! Functional description:
1 !
1 !       This routine executes in kernel mode, and performs some
1 !       additional setup necessary for the correct operation of OPCOM.
1 !
1 ! Input:
1 !
1 !       None.
1 !
1 ! Implicit Input:
1 !
1 !       A channel has been assigned to _MBA2:.
1 !
1 ! Output:
1 !
1 !       None.
1 !
1 ! Implict output:
1 !
1 !       If the _OPA0: exists, then the OPR bit in its UCB is set.
1 !
1 ! Side effects:
1 !
1 !       None.
1 !
1 ! Routine value:
1 !
1 !       SS$_NORMAL      if everything worked.
1 !       SS$_DEVALLOC if _MBA2: has a reference count greater than 2.
1 !       SS$_NOSUCHDEV   if _OPA0: does not exist.
1 !--
1
2 BEGIN                                           ! Start of KERNEL_SETUP
2
2 EXTERNAL
2       SYS$GL_OPRMBX   : $bblock ADDRESSING_MODE (GENERAL);    ! Address of MBA2 UCB
2
2 OWN
2       DEF_OPER_DEV    : $string_desc ('_OPA0:');! Default operator device name
2
2 LOCAL
2       STATUS          : LONG;
2
2 !
2 ! Check the reference count in the _MBA2: UCB.  If it is greater than 2,
2 ! it means that there is already a listener at the mailbox.
2 !
2 IF .SYS$GL_OPRMBX [UCB$W_REFC] GTR 2
2 THEN
2    RETURN (SS$_DEVALLOC);
2
2 !
2 ! Set the OPR bit in the default operator UCB.
2 !
```

```
;  419          0417  2 RETURN (EXE$SETOPR (DEF_OPER_DEV,ON));
;  420          0418  1 END;                                                 ! End of KERNEL_SETUP


                                                                    .PSECT  $PLIT$,NOWRT,NOEXE,2

                         3A  30  41  50  4F  5F  00010 P.AAC:      .ASCII  \_OPA0:\                                ;

                                                                    .PSECT  $OWN$,NOEXE,2

                                         0006  0003C DEF_OPER_DEV:
                                                                    .WORD   6                                       ;
                                      01  0E  0003E                 .BYTE   14, 1                                   ;
                                  00000000'  00040                  .ADDRESS P.AAC                                  ;

                                                                    .EXTRN  SYS$GL_OPRMBX, EXE$SETOPR

                                                                    .PSECT  $CODE$,NOWRT,2

                                         0000  00000 KERNEL_SETUP:
                                                                    .WORD   Save nothing                          ; 0360
                      02 00000000G  00  B1  00002                   CMPW    SYS$GL_OPRMBX+92, #2                  ; 0410
                                    06  1B  00009                   BLEQU   1$                                    ;
                      50      0840  8F  3C  0000B                   MOVZWL  #2112, R0                             ; 0412
                                        04  00010                   RET                                           ;
                                    01  DD  00011 1$:               PUSHL   #1                                    ; 0417
                              0000'  CF  9F  00013                  PUSHAB  DEF_OPER_DEV                          ;
              00000000G  00          02  FB  00017                  CALLS   #2, EXE$SETOPR                        ;
                                        04  0001E                   RET                                           ; 0418

; Routine Size:  31 bytes,    Routine Base:  $CODE$ + 01F4


;  421          0419  1
;  422          0420  1 END
;  423          0421  0 ELUDOM                                               ! End of OPCOMINIT




;                                  PSECT SUMMARY
;
;       Name                       Bytes                          Attributes
;
;   $OWN$                              68  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;   $PLIT$                             22  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;   $CODE$                            531  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)



;                             Library Statistics
;
;                                      -------- Symbols --------      Pages       Processing
;
```

```
;       File                                  Total    Loaded   Percent      Mapped     Time
;
;  _$255$DUA28:[SYSLIB]LIB.L32;1              18619       25         0        1000       00:01.9
;  _$255$DUA28:[OPCOM.OBJ]OPCOMLIB.L32;1        633       24         3          43       00:00.9




;                                    COMMAND QUALIFIERS

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

; Size:          531 code + 90 data bytes
; Run Time:         00:14.5
; Elapsed Time:     00:48.4
; Lines/CPU Min:     1744
; Lexemes/CPU-Min: 19379
; Memory Used:  183 pages
; Compilation Complete
```

0290 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY