```
   OOOOOOOOO     PPPPPPPPPPPP        CCCCCCCCCCCC     OOOOOOOOO     MMM         MMM
   OOOOOOOOO     PPPPPPPPPPPP        CCCCCCCCCCCC     OOOOOOOOO     MMM         MMM
   OOOOOOOOO     PPPPPPPPPPPP        CCCCCCCCCCCC     OOOOOOOOO     MMM         MMM
OOO         OOO  PPP         PPP  CCC               OOO         OOO  MMMMMM   MMMMMM
OOO         OOO  PPP         PPP  CCC               OOO         OOO  MMMMMM   MMMMMM
OOO         OOO  PPP         PPP  CCC               OOO         OOO  MMMMMM   MMMMMM
OOO         OOO  PPP         PPP  CCC               OOO         OOO  MMM   MMM   MMM
OOO         OOO  PPP         PPP  CCC               OOO         OOO  MMM   MMM   MMM
OOO         OOO  PPP         PPP  CCC               OOO         OOO  MMM   MMM   MMM
OOO         OOO  PPPPPPPPPPPP     CCC               OOO         OOO  MMM         MMM
OOO         OOO  PPPPPPPPPPPP     CCC               OOO         OOO  MMM         MMM
OOO         OOO  PPPPPPPPPPPP     CCC               OOO         OOO  MMM         MMM
OOO         OOO  PPP              CCC               OOO         OOO  MMM         MMM
OOO         OOO  PPP              CCC               OOO         OOO  MMM         MMM
OOO         OOO  PPP              CCC               OOO         OOO  MMM         MMM
OOO         OOO  PPP              CCC               OOO         OOO  MMM         MMM
OOO         OOO  PPP              CCC               OOO         OOO  MMM         MMM
OOO         OOO  PPP              CCC               OOO         OOO  MMM         MMM
   OOOOOOOOO     PPP                 CCCCCCCCCCCC     OOOOOOOOO     MMM         MMM
   OOOOOOOOO     PPP                 CCCCCCCCCCCC     OOOOOOOOO     MMM         MMM
   OOOOOOOOO     PPP                 CCCCCCCCCCCC     OOOOOOOOO     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**DEVICE

```
DDDDDDDD    EEEEEEEEEE  VV      VV    IIIIII      CCCCCCCC  EEEEEEEEEE
DDDDDDDD    EEEEEEEEEE  VV      VV    IIIIII      CCCCCCCC  EEEEEEEEEE
DD      DD  EE          VV      VV      II      CC          EE
DD      DD  EE          VV      VV      II      CC          EE
DD      DD  EE          VV      VV      II      CC          EE
DD      DD  EEEEEEEE    VV      VV      II      CC          EEEEEEEE
DD      DD  EEEEEEEE    VV      VV      II      CC          EEEEEEEE
DD      DD  EE          VV      VV      II      CC          EE
DD      DD  EE          VV      VV      II      CC          EE
DD      DD  EE            VV  VV        II      CC          EE              ....
DD      DD  EE            VV  VV        II      CC          EE              ....
DDDDDDDD    EEEEEEEEEE      VV        IIIIII      CCCCCCCC  EEEEEEEEEE      ....
DDDDDDDD    EEEEEEEEEE      VV        IIIIII      CCCCCCCC  EEEEEEEEEE      ....

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$DEVICE       (
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 !       Steven T. Jeffreys
0 !
0 ! Creation date:
0 !
0 !       March 10, 1981
0 !
0 ! Revision history:
0 !
0 !       V03-005 CWH3005         CW Hobbs                        15-May-1984
0 !               If a device name already contains a "$", then use the name
0 !               as it was received.  If no "$", then expand to fulldevnam.
0 !               This fixes two problems:
0 !                - When a dual-pathed device goes offline, we have been
```

```
0 !                      reporting the good path as going offline, since by the
0 !                      time we performed the $GETDVI the device had failed over
0 !                      to the good path.
0 !                    - The previous change (CWH3169) made line printers, etc
0 !                      show up as _LPA0:, with no node information.  We can now
0 !                      show the correct node info and avoid confusion.
0 !
0 !         V03-004 CWH3169         CW Hobbs                        5-May-1984
0 !                 Second pass for cluster-wide OPCOM:
0 !                  - Change interface to SHARE_FULL_DEVNAME so that DVI$_ item
0 !                    code is specified, lets use show pathname in messages
0 !                  - Add OPC$_CTLRUCODE for generic controller microcode not
0 !                    up to rev level, convert TM78MVER and UDA50MVER to use it,
0 !                    add RC25MVER, RDRXMVER, TU81MVER, MAYAMVER messages.
0 !
0 !         V03-003 CWH3001         CW Hobbs                        30-Jul-1983
0 !                 Various and sundry things to make OPCOM distributed
0 !                 across the cluster.
0 !
0 !         V03-002 STJ50798        Steven T. Jeffreys,     1-Dec-1982
0 !                 Check secondary device characteristics to detect and
0 !                 handle spooled devices.
0 !
0 !         V03-001 RLRV3A1         Robert L. Rappaport     5-Apr-1982
0 !                 Added MSG$_UDA50MVER and MSG$_DUPUNITNO.
0 !
0 !         V02-003 STJ0155         Steven T. Jeffreys,     05-Feb-1982
0 !                 More mount verification message support.
0 !
0 !         V02-002 STJ0065         Steven T. Jeffreys,     14-Jul-1981
0 !                 Added support for mount verification messages.
0 !
0 !--
0
1 BEGIN                                         ! Start of DEVICE
1
1 LIBRARY 'SYS$LIBRARY:LIB.L32';
1 LIBRARY 'LIB$:OPCOMLIB';
1
1 FORWARD ROUTINE
1         DEVICE_HANDLER          : NOVALUE;
1
1 BUILTIN
1
1         INSQUE,                               ! Insert entry onto a queue
1         REMQUE;                               ! Remove entry from a queue
1
1 EXTERNAL ROUTINE
1         ALLOCATE_DS,                          ! Allocate a data structure
1         CLUSMSG_RQCB_SEND,                    ! Send RQCB to remote nodes
1         SHARE_FULL_DEVNAME,                   ! Get full device name, including SCS nodename
1         DEALLOCATE_RQCB : NOVALUE,            ! Dispose of an RQCB
1         DUMP_LOG_FILE,                        ! Put a random string in the log file
1         FORMAT_MESSAGE,                       ! Format a message
1         LOG_MESSAGE,                          ! Log an event
1         NOTIFY_LISTED_OPERATORS;              ! Notify interested operators
1
```

```
;  115        0115   1 EXTERNAL
;  116        0116   1         OCD_VECTOR      : VECTOR;                ! OCD list heads
;  117        0117   1
;  118        0118   1 EXTERNAL LITERAL
;  119        0119   1         RQCB_K_TYPE,                             ! RQCB structure type
;  120        0120   1         MIN_SCOPE,                               ! Minimum scope value
;  121        0121   1         MAX_SCOPE;                               ! Maximum scope value
```

```
1 GLOBAL ROUTINE DEVICE_HANDLER (BUFFER_DESC) : NOVALUE =
1
1 !++
1 ! Functional description:
1 !
1 !       This routine is the handler for all device ONLINE or OFFLINE messages.
1 !       Note that this message has a special format, different from all other
1 !       messages.  This is because the message is sent via  EXE$SNDEVMSG
1 !       instead of $SNDOPR.
1 !
1 !
1 !
1 ! Input:
1 !
1 !       BUFFER_DESC : The address of a quadword buffer descriptor that
1 !                     describes the buffer containing the message.
1 !
1 ! Implicit Input:
1 !
1 !       The format of the message is as follows:
1 !               WORD  <MSG$_DEVONLIN or MSG$_DEVOFFLIN>
1 !               WORD  <device unit number>
1 !               ASCIC <device name (DDC)>
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 DEVICE_HANDLER
2
2 MAP
2
2       BUFFER_DESC     : $ref_bblock;
2
2 LOCAL
2       MESSAGE_VECTOR  : VECTOR [5,LONG],      ! Message info
2       DEV_CHAR1       : $bblock [DIB$K_LENGTH],! Device characteristics buffer
2       CHAR_DESC1      : $desc_block,          ! Descriptor for above buffer
2       DEV_CHAR2       : $bblock [DIB$K_LENGTH],! Secondary characteristics
2       CHAR_DESC2      : $desc_block,          ! Sec. char. buffer descriptor
2       DEV_CLASS       : LONG,                 ! Device class
2       NAM_BUF         : $bblock [MAX_DEV_NAM],! Buffer for device name
2       NAM_DESC        : $desc_block,          ! Descriptor for above buffer
```

```
;  180        0179  2           FULL_DESC        : $ref_bblock,          ! Pointer to descriptor returned
;  181        0180  2           MESSAGE          : LONG,                 ! Message code
;  182        0181  2           MSG              : $ref_bblock,          ! Pointer to message text
;  183        0182  2           RQCB             : $ref_bblock,          ! RQCB data structure
;  184        0183  2           OCD              : $ref_bblock,          ! OCD data structure
;  185        0184  2           OCD_COUNT        : LONG,                 ! Count of OCDs in OCD list
;  186        0185  2           OCD_INDEX        : LONG,                 ! Index into OCD_VECTOR
;  187        0186  2           OPER_COUNT       : LONG,                 ! Count of operators in operator list
;  188        0187  2           DVI_ITEM         : LONG,                 ! Name item code
;  189        0188  2           STATUS           : LONG;
;  190        0189  2
;  191        0190  2 !
;  192        0191  2 ! Check the message for a minimum length.  Assume minimum
;  193        0192  2 ! length for a device name (minus the unit #) is 2 characters.
;  194        0193  2 !
;  195        0194  3 IF .BUFFER_DESC [DSC$W_LENGTH] LSS (2+2+3)
;  196        0195  2 THEN
;  197        0196  2     RETURN;
;  198        0197  2 !
;  199        0198  2 ! The message must be one of the known device messages.  All others
;  200        0199  2 ! are ignored.  Set the message code in the message vector to the
;  201        0200  2 ! proper value.
;  202        0201  2 !
;  203        0202  2 MSG = .BUFFER_DESC [DSC$A_POINTER];
;  204        0203  2 SELECTONE .MSG [0,0,16,0] OF
;  205        0204  2     SET
;  206        0205  2     [ MSG$_DEVONLIN      ] : MESSAGE_VECTOR [0] = OPC$_DEVONLINE;
;  207        0206  2     [ MSG$_DEVOFFLIN     ] : MESSAGE_VECTOR [0] = OPC$_DEVOFFLINE;
;  208        0207  2     [ MSG$_DEVOFFLINX    ] : MESSAGE_VECTOR [0] = OPC$_DEVOFFLINX;
;  209        0208  2     [ MSG$_WRONGVOL      ] : MESSAGE_VECTOR [0] = OPC$_WRONGVOL;
;  210        0209  2     [ MSG$_DEVWRTLCK     ] : MESSAGE_VECTOR [0] = OPC$_DEVWRTLCK;
;  211        0210  2     [ MSG$_MVCOMPLETE    ] : MESSAGE_VECTOR [0] = OPC$_MVCOMPLETE;
;  212        0211  2     [ MSG$_MVABORTED     ] : MESSAGE_VECTOR [0] = OPC$_MVABORTED;
;  213        0212  2     [ MSG$_DUPUNITNO     ] : MESSAGE_VECTOR [0] = OPC$_DUPUNITNO;
;  214        0213  3     [ MSG$_UDA50MVER     ] : BEGIN MESSAGE_VECTOR [0] = OPC$_CTLRUCODE;
;  215        0214  2                                   MESSAGE_VECTOR [3] = UPLIT BYTE (%ASCIC 'UDA50') END;
;  216        0215  3     [ MSG$_TM78MVER      ] : BEGIN MESSAGE_VECTOR [0] = OPC$_CTLRUCODE;
;  217        0216  2                                   MESSAGE_VECTOR [3] = UPLIT BYTE (%ASCIC 'TM78') END;
;  218        0217  3     [ MSG$_RC25MVER      ] : BEGIN MESSAGE_VECTOR [0] = OPC$_CTLRUCODE;
;  219        0218  2                                   MESSAGE_VECTOR [3] = UPLIT BYTE (%ASCIC 'RC25') END;
;  220        0219  3     [ MSG$_RDRXMVER      ] : BEGIN MESSAGE_VECTOR [0] = OPC$_CTLRUCODE;
;  221        0220  2                                   MESSAGE_VECTOR [3] = UPLIT BYTE (%ASCIC 'RD/RX') END;
;  222        0221  3     [ MSG$_TU81MVER      ] : BEGIN MESSAGE_VECTOR [0] = OPC$_CTLRUCODE;
;  223        0222  2                                   MESSAGE_VECTOR [3] = UPLIT BYTE (%ASCIC 'TU81') END;
;  224        0223  3     [ MSG$_MAYAMVER      ] : BEGIN MESSAGE_VECTOR [0] = OPC$_CTLRUCODE;
;  225        0224  2                                   MESSAGE_VECTOR [3] = UPLIT BYTE (%ASCIC 'MAYA') END;
;  226        0225  2     [ OTHERWISE          ] : RETURN;
;  227        0226  2     TES;
;  228        0227  2 !
;  229        0228  2 ! Format the device name into the local format.
;  230        0229  2 !
;  231        0230  2 NAM_DESC [0,0,32,0] = MAX_DEV_NAM;
;  232        0231  2 NAM_DESC [DSC$A_POINTER] = NAM_BUF;
;  233        0232  3 IF NOT $FAO (%ASCID '_!AC!UW:', NAM_DESC, NAM_DESC, MSG [4,0,0,0], .MSG [2,0,16,0])
;  234        0233  2 THEN
;  235        0234  2     RETURN;
;  236        0235  2 !
;  237          0236  2 ! Expand the device name to the clusterwide format, and stick it in the local name buffer.
;  238          0237  2 ! If the de ice name already contains a dollar sign, then use it as is.
;  239          0238  2 !
;  240          0239  3 FULL_DESC =     (IF CH$FIND_CH (.NAM_DESC [DSC$W_LENGTH], .NAM_DESC [DSC$A_POINTER], '$') NEQ 0
;  241          0240  3                 THEN
;  242          0241  3                     NAM_DESC
;  243          0242  3                 ELSE
;  244          0243  2                     SHARE_FULL_DEVNAME (NAM_DESC, DVI$_FULLDEVNAM));
;  245          0244  2 CH$MOVE (.FULL_DESC [DSC$W_LENGTH], .FULL_DESC [DSC$A_POINTER], NAM_BUF);
;  246          0245  2 NAM_DESC [0,0,32,0] = .FULL_DESC [DSC$W_LENGTH];
;  247          0246  2 !
;  248          0247  2 ! Get the device characteristics and put the device class code into a
;  249          0248  2 ! local variable.  Note that if a device is spooled, the primary device
;  250          0249  2 ! characteristics are those of the intermediate device.
;  251          0250  2 !
;  252          0251  2 CHAR_DESC1 [0,0,32,0] = DIB$K_LENGTH;
;  253          0252  2 CHAR_DESC1 [DSC$A_POINTER] = DEV_CHAR1;
;  254          0253  2 CHAR_DESC2 [0,0,32,0] = DIB$K_LENGTH;
;  255          0254  2 CHAR_DESC2 [DSC$A_POINTER] = DEV_CHAR2;
;  256        P 0255  2 IF NOT $GETDEV  (DEVNAM = NAM_DESC,
;  257        P 0256  2                  PRILEN = CHAR_DESC1 [DSC$W_LENGTH],
;  258        P 0257  2                  PRIBUF = CHAR_DESC1,
;  259        P 0258  2                  SCDLEN = CHAR_DESC2 [DSC$W_LENGTH],
;  260        P 0259  2                  SCDBUF = CHAR_DESC2
;  261          0260  3                 )
;  262          0261  2 THEN
;  263          0262  2     RETURN;
;  264          0263  2 DEV_CLASS = .DEV_CHAR1 [DIB$B_DEVCLASS];                ! Assume not spooled
;  265          0264  2 IF  .$bblock [DEV_CHAR2 [DIB$L_DEVCHAR], DEV$V_SPL]
;  266          0265  2 AND NOT CH$EQL (.CHAR_DESC1 [DSC$W_LENGTH],
;  267          0266  2                 .CHAR_DESC1 [DSC$A_POINTER],
;  268          0267  2                 .CHAR_DESC2 [DSC$W_LENGTH],
;  269          0268  2                 .CHAR_DESC2 [DSC$A_POINTER],
;  270          0269  2                 %C' '
;  271          0270  2                 )
;  272          0271  2 THEN
;  273          0272  2     DEV_CLASS = .DEV_CHAR2 [DIB$B_DEVCLASS];            ! Device is spooled
;  274          0273  2 !
;  275          0274  2 ! Allocate an RQCB.  This is necessary
;  276          0275  2 ! to format and later issue the message.
;  277          0276  2 !
;  278          0277  2 IF NOT ALLOCATE_DS (RQCB_K_TYPE, RQCB)
;  279          0278  2 THEN
;  280          0279  2     RETURN;
;  281          0280  2 !
;  282          0281  2 ! Set the operator interest mask depending on the device class.
;  283          0282  2 ! Also target the message to DEVICE class operators.
;  284          0283  2 !
;  285          0284  3 RQCB [RQCB_L_ATTNMASK1] = (SELECTONEU .DEV_CLASS OF
;  286          0285  3                             SET
;  287          0286  3                             [ DC$_DISK ]        : OPC$M_NM_DISKS;
;  288          0287  3                             [ DC$_TAPE ]        : OPC$M_NM_TAPES;
;  289          0288  3                             [ DC$_CARD ]        : OPC$M_NM_CARDS;
;  290          0289  3                             [ DC$_LP ]          : OPC$M_NM_PRINT;
;  291          0290  3                             [ OTHERWISE ]       : OPC$M_NM_DEVICE;
;  292          0291  2                             TES);
;  293          0292  2 RQCB [RQCB_L_ATTNMASK1] = (.RQCB [RQCB_L_ATTNMASK1] OR OPC$M_NM_DEVICE);
```

```
2 !
2 ! Format the message, then send it to everyone else in the cluster
2 !
2 MESSAGE_VECTOR [1] = 0;                          ! Use current system time
2 MESSAGE_VECTOR [2] = NAM_DESC;                   ! Set addr of dev name descriptor
2 FORMAT_MESSAGE (.RQCB, MESSAGE_VECTOR);
2 CLUSMSG_RQCB_SEND (-1, CLM__DEVICE, .RQCB);      ! Send it everywhere
2 !
2 ! Log the message, and send it to all interested operators.
2 ! Every operator in the data base is a candidate for the message.
2 !
2 OCD_INDEX = MAX_SCOPE;
2 WHILE (.OCD_INDEX GEQ MIN_SCOPE) DO
3     BEGIN
3     !
3     ! Scan the OCD list for each class of operator.
3     !
3     OCD_COUNT = .OCD_VECTOR [(.OCD_INDEX - 1) * 2 + 1];
3     OCD = .OCD_VECTOR [(.OCD_INDEX - 1) * 2];
3     WHILE (.OCD_COUNT GTR 0) DO
4         BEGIN
4         !
4         ! Notify every operator in the OCD's operator list.
4         ! Also log the message for each OCD.
4         !
4         RQCB [RQCB_L_OCD] = .OCD;                ! Set OCD address
4         LOG_MESSAGE (.RQCB);                     ! Log the message
4         NOTIFY_LISTED_OPERATORS (.RQCB);         ! Inform the operators
4         OCD_COUNT = .OCD_COUNT - 1;              ! Decrement operator count
4         OCD = .OCD [OCD_L_FLINK];                ! Get next OCD address
4         END;
3     OCD_INDEX = .OCD_INDEX - 1;
2     END;
2 !
2 ! Free the rqcb
2 !
2 DEALLOCATE_RQCB (.RQCB);
2 RETURN;
2
1 END;                                             ! End of DEVICE_HANDLER

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

                                                   .PSECT  $PLIT$,NOWRT,NOEXE,2

                 30  35  41  44  55  05  00000 P.AAA:  .ASCII  <5>\UDA50\                              ;
                     38  37  4D  54  04  00006 P.AAB:  .ASCII  <4>\TM78\                               ;
                     35  32  43  52  04  0000B P.AAC:  .ASCII  <4>\RC25\                               ;
                 58  52  2F  44  52  05  00010 P.AAD:  .ASCII  <5>\RD/RX\                              ;
                     31  38  55  54  04  00016 P.AAE:  .ASCII  <4>\TU81\                               ;
                     41  59  41  4D  04  0001B P.AAF:  .ASCII  <4>\MAYA\                               ;
         3A  57  55  21  43  41  21  5F  00020 P.AAH:  .ASCII  \_!AC!UW:\                              ;
                           010E0008  00028 P.AAG:  .LONG   17694728                                ;
                           00000000' 0002C         .ADDRESS P.AAH                                  ;
```

```
                                                  .EXTRN  ALLOCATE_DS, CLUSMSG_RQCB_SEND
                                                  .EXTRN  SHARE_FULL_DEVNAME
                                                  .EXTRN  DEALLOCATE_RQCB
                                                  .EXTRN  DUMP_LOG_FILE, FORMAT_MESSAGE
                                                  .EXTRN  LOG_MESSAGE, NOTIFY_LISTED_OPERATORS
                                                  .EXTRN  OCD_VECTOR, RQCB_K_TYPE
                                                  .EXTRN  MIN_SCOPE, MAX_SCOPE
                                                  .EXTRN  SYS$FAO, SYS$GETDEV

                                                  .PSECT  $CODE$,NOWRT,2

                              00FC 00000          .ENTRY  DEVICE_HANDLER, Save R2,R3,R4,R5,R6,R7       ; 0122
           57     0000'  CF   9E 00002            MOVAB   P.AAA, R7                                    ;
           5E     FEA8   CE   9E 00007            MOVAB   -344(SP), SP                                 ;
           50       04   AC   D0 0000C            MOVL    BUFFER_DESC, R0                              ; 0194
           07            60   B1 00010            CMPW    (R0), #7                                     ;
                         01   1E 00013            BGEQU   1$                                           ;
                              04 00015            RET                                                  ;
           50       04   A0   D0 00016 1$:        MOVL    4(R0), MSG                                   ; 0202
           07            60   B1 0001A            CMPW    (MSG), #7                                    ; 0205
                         0A   12 0001D            BNEQ    2$                                           ;
      EC   AD 00058054   8F   D0 0001F            MOVL    #360532, MESSAGE_VECTOR                      ;
                         73   11 00027            BRB     9$                                           ;
           05            60   B1 00029 2$:        CMPW    (MSG), #5                                    ; 0206
                         0A   12 0002C            BNEQ    3$                                           ;
      EC   AD 0005804C   8F   D0 0002E            MOVL    #360524, MESSAGE_VECTOR                      ;
                         79   11 00036            BRB     11$                                          ;
    0050   8F            60   B1 00038 3$:        CMPW    (MSG), #80                                   ; 0207
                         0A   12 0003D            BNEQ    4$                                           ;
      EC   AD 000581BB   8F   D0 0003F            MOVL    #360891, MESSAGE_VECTOR                      ;
                         7E   11 00047            BRB     13$                                          ;
    0051   8F            60   B1 00049 4$:        CMPW    (MSG), #81                                   ; 0208
                         0A   12 0004E            BNEQ    5$                                           ;
      EC   AD 000581C3   8F   D0 00050            MOVL    #360899, MESSAGE_VECTOR                      ;
                         6D   11 00058            BRB     13$                                          ;
    0052   8F            60   B1 0005A 5$:        CMPW    (MSG), #82                                   ; 0209
                         0A   12 0005F            BNEQ    6$                                           ;
      EC   AD 000581CB   8F   D0 00061            MOVL    #360907, MESSAGE_VECTOR                      ;
                         72   11 00069            BRB     15$                                          ;
    0054   8F            60   B1 0006B 6$:        CMPW    (MSG), #84                                   ; 0210
                         0A   12 00070            BNEQ    7$                                           ;
      EC   AD 000581EB   8F   D0 00072            MOVL    #360939, MESSAGE_VECTOR                      ;
                         77   11 0007A            BRB     17$                                          ;
    0055   8F            60   B1 0007C 7$:        CMPW    (MSG), #85                                   ; 0211
                         0A   12 00081            BNEQ    8$                                           ;
      EC   AD 000581F3   8F   D0 00083            MOVL    #360947, MESSAGE_VECTOR                      ;
                         7C   11 0008B            BRB     19$                                          ;
    0058   8F            60   B1 0008D 8$:        CMPW    (MSG), #88                                   ; 0212
                         0A   12 00092            BNEQ    10$                                          ;
      EC   AD 00058203   8F   D0 00094            MOVL    #360963, MESSAGE_VECTOR                      ;
                         6B   11 0009C 9$:        BRB     19$                                          ;
    0057   8F            60   B1 0009E 10$:       CMPW    (MSG), #87                                   ; 0213
                         0E   12 000A3            BNEQ    12$                                          ;
      EC   AD 000581FB   8F   D0 000A5            MOVL    #360955, MESSAGE_VECTOR                      ;
      F8   AD            67   9E 000AD            MOVAB   P.AAA, MESSAGE_VECTOR+12                     ; 0214
                         6D   11 000B1 11$:       BRB     22$                                          ;
    005A   8F            60   B1 000B3 12$:       CMPW    (MSG), #90                                   ; 0215
```

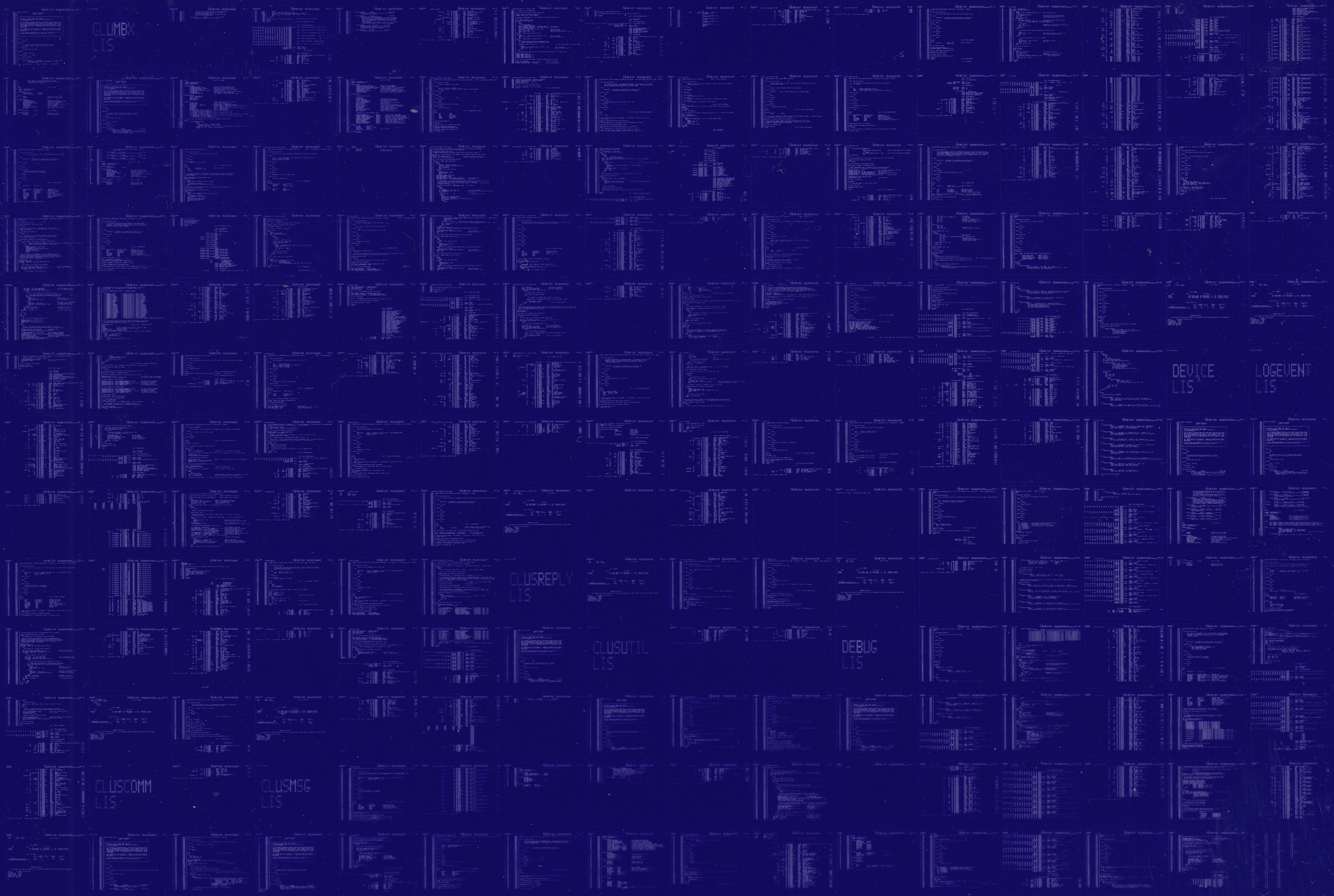
```
                                0F 12 000B8         BNEQ     14$                                     ;
                EC AD 000581FB  8F D0 000BA         MOVL     #360955, MESSAGE_VECTOR                 ;
                F8 AD       06  A7 9E 000C2         MOVAB    P.AAB, MESSAGE_VECTOR+12                ; 0216
                                57 11 000C7 13$:    BRB      22$                                     ;
              005D 8F           60 B1 000C9 14$:    CMPW     (MSG), #93                              ; 0217
                                0F 12 000CE         BNEQ     16$                                     ;
                EC AD 000581FB  8F D0 000D0         MOVL     #360955, MESSAGE_VECTOR                 ;
                F8 AD       0B  A7 9E 000D8         MOVAB    P.AAC, MESSAGE_VECTOR+12                ; 0218
                                41 11 000DD 15$:    BRB      22$                                     ;
              005E 8F           60 B1 000DF 16$:    CMPW     (MSG), #94                              ; 0219
                                0F 12 000E4         BNEQ     18$                                     ;
                EC AD 000581FB  8F D0 000E6         MOVL     #360955, MESSAGE_VECTOR                 ;
                F8 AD       10  A7 9E 000EE         MOVAB    P.AAD, MESSAGE_VECTOR+12                ; 0220
                                2B 11 000F3 17$:    BRB      22$                                     ;
              005F 8F           60 B1 000F5 18$:    CMPW     (MSG), #95                              ; 0221
                                0F 12 000FA         BNEQ     20$                                     ;
                EC AD 000581FB  8F D0 000FC         MOVL     #360955, MESSAGE_VECTOR                 ;
                F8 AD       16  A7 9E 00104         MOVAB    P.AAE, MESSAGE_VECTOR+12                ; 0222
                                15 11 00109 19$:    BRB      22$                                     ;
              0060 8F           60 B1 0010B 20$:    CMPW     (MSG), #96                              ; 0223
                                01 13 00110         BEQL     21$                                     ;
                                   04 00112         RET                                              ;
                EC AD 000581FB  8F D0 00113 21$:    MOVL     #360955, MESSAGE_VECTOR                 ;
                F8 AD       1B  A7 9E 0011B         MOVAB    P.AAF, MESSAGE_VECTOR+12                ; 0224
                04 AE       40  8F 9A 00120 22$:    MOVZBL   #64, NAM_DESC                           ; 0230
                08 AE       0C  AE 9E 00125         MOVAB    NAM_BUF, NAM_DESC+4                     ; 0231
                   7E       02  A0 3C 0012A         MOVZWL   2(MSG), -(SP)                           ; 0232
                            04  A0 9F 0012E         PUSHAB   4(MSG)                                  ;
                            0C  AE 9F 00131         PUSHAB   NAM_DESC                                ;
                            10  AE 9F 00134         PUSHAB   NAM_DESC                                ;
                            28  A7 9F 00137         PUSHAB   P.AAG                                   ;
         00000000G 00           05 FB 0013A         CALLS    #5, SYS$FAO                             ;
                   5C           50 E9 00141         BLBC     R0, 26$                                 ;
08   BE         04 AE           24 3A 00144         LOCC     #36, NAM_DESC, @NAM_DESC+4              ; 0239
                                02 12 0014A         BNEQ     23$                                     ;
                                51 D4 0014C         CLRL     R1                                      ;
                                51 D5 0014E 23$:    TSTL     R1                                      ;
                                06 13 00150         BEQL     24$                                     ;
                   56       04  AE 9E 00152         MOVAB    NAM_DESC, FULL_DESC                     ;
                                0F 11 00156         BRB      25$                                     ;
                   7E       E8  8F 9A 00158 24$:    MOVZBL   #232, -(SP)                             ; 0243
                            08  AE 9F 0015C         PUSHAB   NAM_DESC                                ;
             0000G CF           02 FB 0015F         CALLS    #2, SHARE_FULL_DEVNAME                  ;
                   56           50 D0 00164         MOVL     R0, FULL_DESC                           ;
0C   AE         04 B6           66 28 00167 25$:    MOVC3    (FULL_DESC), @4(FULL_DESC), NAM_BUF     ; 0244
                04 AE           66 3C 0016D         MOVZWL   (FULL_DESC), NAM_DESC                   ; 0245
              FF70 CD       74  8F 9A 00171         MOVZBL   #116, CHAR_DESC1                        ; 0251
              FF74 CD     FF78  CD 9E 00177         MOVAB    DEV_CHAR1, CHAR_DESC1+4                 ; 0252
                4C AE       74  8F 9A 0017E         MOVZBL   #116, CHAR_DESC2                        ; 0253
                50 AE       54  AE 9E 00183         MOVAB    DEV_CHAR2, CHAR_DESC2+4                 ; 0254
                            4C  AE 9F 00188         PUSHAB   CHAR_DESC2                              ; 0260
                            50  AE 9F 0018B         PUSHAB   CHAR_DESC2                              ;
                          FF70  CD 9F 0018E         PUSHAB   CHAR_DESC1                              ;
                          FF70  CD 9F 00192         PUSHAB   CHAR_DESC1                              ;
                            14  AE 9F 00196         PUSHAB   NAM_DESC                                ;
         00000000G 00           05 FB 00199         CALLS    #5, SYS$GETDEV                          ;
                   29           50 E9 001A0 26$:    BLBC     R0, 28$                                 ;
```

```
                                       54           FF7C  CD  9A 001A3          MOVZBL  DEV_CHAR1+4, DEV_CLASS                      ; 0263
                              12       54   AE            06  E1 001A8          BBC     #6, DEV_CHAR2, 27$                          ; 0264
          4C    AE            20     FF74   DD      FF70  CD  2D 001AD          CMPC5   CHAR_DESC1, @CHAR_DESC1+4, #32, -           ; 0265
                                                      50  BE     001B7                  CHAR_DESC2, @CHAR_DESC2+4                   ;
                                                          04  13 001B9          BEQL    27$                                         ;
                                       54             58  AE  9A 001BB          MOVZBL  DEV_CHAR2+4, DEV_CLASS                      ; 0272
                                                          5E  DD 001BF 27$:     PUSHL   SP                                          ; 0277
                                               00000000G  8F  DD 001C1          PUSHL   #RQCB_K_TYPE                                ;
                                    0000G   CF            02  FB 001C7          CALLS   #2, ALLOCATE_DS                             ;
                                            01            50  E8 001CC 28$:     BLBS    R0, 29$                                     ;
                                                              04 001CF          RET                                                 ;
                                            53            6E  D0 001D0 29$:     MOVL    RQCB, R3                                    ; 0284
                                            01            54  D1 001D3          CMPL    DEV_CLASS, #1                               ; 0286
                                                          05  12 001D6          BNEQ    30$                                         ;
                                            50            08  D0 001D8          MOVL    #8, R0                                      ;
                                                          29  11 001DB          BRB     34$                                         ;
                                            02            54  D1 001DD 30$:     CMPL    DEV_CLASS, #2                               ; 0287
                                                          05  12 001E0          BNEQ    31$                                         ;
                                            50            04  D0 001E2          MOVL    #4, R0                                      ;
                                                          1F  11 001E5          BRB     34$                                         ;
                                 00000041   8F            54  D1 001E7 31$:     CMPL    DEV_CLASS, #65                              ; 0288
                                                          05  12 001EE          BNEQ    32$                                         ;
                                            50            20  D0 001F0          MOVL    #32, R0                                     ;
                                                          11  11 001F3          BRB     34$                                         ;
                                 00000043   8F            54  D1 001F5 32$:     CMPL    DEV_CLASS, #67                              ; 0289
                                                          05  12 001FC          BNEQ    33$                                         ;
                                            50            02  D0 001FE          MOVL    #2, R0                                      ;
                                                          03  11 00201          BRB     34$                                         ;
                                            50            10  D0 00203 33$:     MOVL    #16, R0                                     ; 0290
                                       5C   A3            50  D0 00206 34$:     MOVL    R0, 92(R3)                                  ; 0284
                                       5C   A3            10  88 0020A          BISB2   #16, 92(R3)                                 ; 0292
                                                      F0  AD  D4 0020E          CLRL    MESSAGE_VECTOR+4                            ; 0296
                                       F4   AD        04  AE  9E 00211          MOVAB   NAM_DESC, MESSAGE_VECTOR+8                  ; 0297
                                                      EC  AD  9F 00216          PUSHAB  MESSAGE_VECTOR                              ; 0298
                                                          53  DD 00219          PUSHL   R3                                          ;
                                    0000G   CF            02  FB 0021B          CALLS   #2, FORMAT_MESSAGE                          ;
                                                          53  DD 00220          PUSHL   R3                                          ; 0299
                                                          08  DD 00222          PUSHL   #8                                          ;
                                            7E            01  CE 00224          MNEGL   #1, -(SP)                                   ;
                                    0000G   CF            03  FB 00227          CALLS   #3, CLUSMSG_RQCB_SEND                       ;
                                            52 00000000G  8F  D0 0022C          MOVL    #MAX_SCOPE, OCD_INDEX                       ; 0304
                                00000000G   8F            52  D1 00233 35$:     CMPL    OCD_INDEX, #MIN_SCOPE                       ; 0305
                                                          35  19 0023A          BLSS    38$                                         ;
                              50            52            01  78 0023C          ASHL    #1, OCD_INDEX, R0                           ; 0310
                                            55     0000GCF40  D0 00240          MOVL    OCD_VECTOR-4[R0], OCD_COUNT                 ;
                              50            52            01  78 00246          ASHL    #1, OCD_INDEX, R0                           ; 0311
                                            54     0000GCF40  D0 0024A          MOVL    OCD_VECTOR-8[R0], OCD                       ;
                                                          55  D5 00250 36$:     TSTL    OCD_COUNT                                   ; 0312
                                                          19  15 00252          BLEQ    37$                                         ;
                                       24   A3            54  D0 00254          MOVL    OCD, 36(R3)                                 ; 0318
                                                          53  DD 00258          PUSHL   R3                                          ; 0319
                                    0000G   CF            01  FB 0025A          CALLS   #1, LOG_MESSAGE                             ;
                                                          53  DD 0025F          PUSHL   R3                                          ; 0320
                                    0000G   CF            01  FB 00261          CALLS   #1, NOTIFY_LISTED_OPERATORS                 ;
                                                          55  D7 00266          DECL    OCD_COUNT                                   ; 0321
                                            54            64  D0 00268          MOVL    (OCD), OCD                                  ; 0322
                                                          E3  11 0026B          BRB     36$                                         ; 0312
```

```
                                                52  D7 0026D 37$:      DECL    OCD_INDEX                                              ; 0324
                                                C2  11 0026F           BRB     35$                                                    ; 0305
                                                53  DD 00271 38$:      PUSHL   R3                                                     ; 0329
                              0000G  CF         01  FB 00273           CALLS   #1, DEALLOCATE_RQCB                                    ;
                                                    04 00278           RET                                                            ; 0332

; Routine Size:  633 bytes,    Routine Base:  $CODE$ + 0000
;  335          0333  1 END                                                 ! End of DEVICE
;  336          0334  0 ELUDOM




;                                          PSECT SUMMARY
;
;       Name                          Bytes                         Attributes
;
;   $PLIT$                                 48  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;   $CODE$                                633  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         45         0       1000          00:01.9
;   _$255$DUA28:[OPCOM.OBJ]OPCOMLIB.L32;1            633          7         1         43          00:00.9




;                                          COMMAND QUALIFIERS

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

; Size:          633 code + 48 data bytes
; Run Time:         00:12.4
; Elapsed Time:     00:38.7
; Lines/CPU Min:     1612
; Lexemes/CPU-Min:  15103
; Memory Used:  163 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