J 6

```
NNN         NNN  MMM         MMM  LLL
NNN         NNN  MMM         MMM  LLL
NNN         NNN  MMM         MMM  LLL
NNN         NNN  MMMMMM   MMMMMM  LLL
NNN         NNN  MMMMMM   MMMMMM  LLL
NNN         NNN  MMMMMM   MMMMMM  LLL
NNNNNN      NNN  MMM   MMM   MMM  LLL
NNNNNN      NNN  MMM   MMM   MMM  LLL
NNNNNN      NNN  MMM   MMM   MMM  LLL
NNN   NNN   NNN  MMM         MMM  LLL
NNN   NNN   NNN  MMM         MMM  LLL
NNN   NNN   NNN  MMM         MMM  LLL
NNN      NNNNNN  MMM         MMM  LLL
NNN      NNNNNN  MMM         MMM  LLL
NNN      NNNNNN  MMM         MMM  LLL
NNN         NNN  MMM         MMM  LLL
NNN         NNN  MMM         MMM  LLL
NNN         NNN  MMM         MMM  LLL
NNN         NNN  MMM         MMM  LLLLLLLLLLLLLLL
NNN         NNN  MMM         MMM  LLLLLLLLLLLLLLL
NNN         NNN  MMM         MMM  LLLLLLLLLLLLLLL
```

_$

Ps

NP

NP

$G

$O

NP

PA

_L

E 8

```
**FILE**ID**NMLLOGOPS
```

```
NN      NN MM      MM LL         LL           OOOOOO     GGGGGGGG   OOOOOO   PPPPPPPP     SSSSSSSS
NN      NN MM      MM LL         LL           OOOOOO     GGGGGGGG   OOOOOO   PPPPPPPP     SSSSSSSS
NN      NN MMMM  MMMM LL         LL         OO      OO GG         OO      OO PP      PP SS
NN      NN MMMM  MMMM LL         LL         OO      OO GG         OO      OO PP      PP SS
NNNN    NN MM  MM  MM LL         LL         OO      OO GG         OO      OO PP      PP SS
NNNN    NN MM  MM  MM LL         LL         OO      OO GG         OO      OO PP      PP SS
NN  NN  NN MM      MM LL         LL         OO      OO GG         OO      OO PPPPPPPP     SSSSSS
NN  NN  NN MM      MM LL         LL         OO      OO GG         OO      OO PPPPPPPP     SSSSSS
NN    NNNN MM      MM LL         LL         OO      OO GG  GGGGGG OO      OO PP                 SS
NN    NNNN MM      MM LL         LL         OO      OO GG  GGGGGG OO      OO PP                 SS
NN      NN MM      MM LL         LL         OO      OO GG      GG OO      OO PP                 SS
NN      NN MM      MM LL         LL         OO      OO GG      GG OO      OO PP                 SS ::::
NN      NN MM      MM LLLLLLLLLL LLLLLLLLLL   OOOOOO     GGGGGG     OOOOOO   PP         SSSSSSSS   ::::
NN      NN MM      MM LLLLLLLLLL LLLLLLLLLL   OOOOOO     GGGGGG     OOOOOO   PP         SSSSSSSS   ::::

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
```

```
0001   0 %TITLE 'NML Logging data base operations module'
0002   0 MODULE NML$LOGOPS (
0003   0                 LANGUAGE (BLISS32),
0004   0                 ADDRESSING_MODE (EXTERNAL=LONG_RELATIVE),
0005   0                 ADDRESSING_MODE (NONEXTERNAL=LONG_RELATIVE),
0006   0                 IDENT = 'V04-000'
0007   0                 ) =
0008   1 BEGIN
0009   1 !
0010   1 !****************************************************************************
0011   1 !*                                                                          *
0012   1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
0013   1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
0014   1 !*  ALL RIGHTS RESERVED.                                                    *
0015   1 !*                                                                          *
0016   1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
0017   1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0018   1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
0019   1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
0020   1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
0021   1 !*  TRANSFERRED.                                                            *
0022   1 !*                                                                          *
0023   1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
0024   1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0025   1 !*  CORPORATION.                                                            *
0026   1 !*                                                                          *
0027   1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
0028   1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
0029   1 !*                                                                          *
0030   1 !*                                                                          *
0031   1 !****************************************************************************
0032   1
0033   1
0034   1 !++
0035   1 ! FACILITY:  DECnet-VAX V2.0 Network Management Listener
0036   1 !
0037   1 ! ABSTRACT:
0038   1 !
0039   1 !       These routines handle all logging data base operations.
0040   1 !
0041   1 ! ENVIRONMENT:  VAX/VMS Operating System
0042   1 !
0043   1 ! AUTHOR:  Distributed Systems Software Engineering
0044   1 !
0045   1 ! CREATION DATE:  26-JUN-1980
0046   1 !
0047   1 ! MODIFIED BY:
0048   1 !       V03-002 MKP0002         Kathy Perko     23-Nov-1982
0049   1 !               Add module as a source for events.
0050   1 !
0051   1 !       V02-001 MKP0001         Kathy Perko     16-Nov-1981
0052   1 !               Add circuit entity as a logging source type.
0053   1 !
0054   1 !--
0055   1
```

```
%SBTTL 'Declarations'

!
! TABLE OF CONTENTS:
!

FORWARD ROUTINE
    NML$ADDFILTERS,
    NML_MODFIL,
    NML_MODCLS,
    NML_MODKNO,
    NML$GETSPCFILTERS,
    NML$GETCOMFILTERS,
    NML$GETGBLFILTERS,
    NML$CLEANEVT             : NOVALUE,
    NML$CLEANSRC             : NOVALUE,
    NML$MATCHSRC,
    NML$GETNXTSNK,
    NML$GETNXTSRC,
    NML$MATCHEVT,
    NML$GETNXTEVT,
    NML$BLDSRC               : NOVALUE,
    NML$BLDEVT               : NOVALUE,
    NML$ADDSRC,
    NML$REPSRC,
    NML$REMSRC               : NOVALUE,
    NML$ADDEVT,
    NML$MODEVT               : NOVALUE,
    NML$REMEVT               : NOVALUE;

!
! INCLUDE FILES:
!

LIBRARY 'LIB$:NMLLIB.L32';
LIBRARY 'SHRLIB$:NMALIBRY.L32';
LIBRARY 'SYS$LIBRARY:STARLET.L32';

!
! OWN STORAGE:
!

OWN
    NML$T_EVTBUFFER : BBLOCK [EVT$K_LENGTH],
    NML$T_SRCBUFFER : BBLOCK [NML$K_RECBFLEN];
BIND
    NML$Q_EVTBFDSC = UPLIT (EVT$K_LENGTH, NML$T_EVTBUFFER) : DESCRIPTOR,
    NML$Q_SRCBFDSC = UPLIT (NML$K_RECBFLEN, NML$T_SRCBUFFER) : DESCRIPTOR;

!
! EXTERNAL REFERENCES:
!

$NML_EXTDEF;

EXTERNAL LITERAL
    NML$GK_EVENTS;
```

```
;  114      0113  1
;  115      0114  1 EXTERNAL
;  116      0115  1      NML$AB_EVENTS : BBLOCKVECTOR [0, ETB$K_ENTRYLEN];
;  117      0116  1
;  118      0117  1 EXTERNAL ROUTINE
;  119      0118  1      NML$ERROR_2;
;  120      0119  1
```

```
1 %SBTTL 'NML$ADDFILTERS  Add event filters for sink node'
1 GLOBAL ROUTINE NML$ADDFILTERS
1       (FCT, BUFDSC, SNK, SRC, ENTDSC, CLASS, MSKLEN, MSKPTR, RESDSC) =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine adds event filters to the data base entry for a sink
1 !       node.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       FCT             Function code. (0=CLEAR/PURGE, 1=SET/DEFINE)
1 !       BUFDSC          Descriptor of buffer to contain modified data base
1 !                       entry.
1 !       SNK             Logging sink type code.
1 !       SRC             Event source type code.
1 !       ENTDSC          Event source id string descriptor.
1 !       CLASS           Event class code.
1 !       MSKLEN          Length of filter mask.
1 !       MSKPTR          Address of filter mask.
1 !       RESDSC          Descriptor of data in buffer.
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NML$GB_EVTMSKTYP
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       TRUE is returned if operation is successful.  Otherwise, FALSE
1 !       is returned.
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !
1 !--
1
2     BEGIN
2
2     MAP
2         BUFDSC : REF DESCRIPTOR,
2         ENTDSC : REF DESCRIPTOR,
2         RESDSC : REF DESCRIPTOR;
2
2     LOCAL
2         SRCPTR : REF BBLOCK,              ! Pointer to source block
2         STATUS;                           ! Routine status code
2
2     STATUS = TRUE;                        ! Initialize return status
2 !
2 ! Get the source block.
```

```
!
    IF NML$MATCHSRC (.RESDSC, .SNK, .SRC, .ENTDSC, SRCPTR)
    THEN
        BEGIN

        CH$MOVE (.SRCPTR [SRC$W_LENGTH],
                 .SRCPTR,
                 NML$T_SRCBUFFER);
        NML$REMSRC (.RESDSC, .SRCPTR);
        SRCPTR = NML$T_SRCBUFFER;

        END
    ELSE
        BEGIN

        NML$BLDSRC (NML$Q_SRCBFDSC, .SNK, .SRC, .ENTDSC);
        SRCPTR = .NML$Q_SRCBFDSC [DSC$A_POINTER];

        END;
!
! Add the events to the source block.
!
    SELECTONEU .NML$GB_EVTMSKTYP OF
        SET

        [2]:                                ! All events in class

            NML_MODCLS (.FCT, NML$Q_SRCBFDSC, .SRCPTR, .CLASS, .SRC);

        [3]:                                ! Known events

            NML_MODKNO (.FCT, NML$Q_SRCBFDSC, .SRCPTR, .SRC);

        [OTHERWISE]:                        ! Add specified events to class

            NML_MODFIL (.FCT,
                        FALSE,
                        NML$Q_SRCBFDSC,
                        .SRCPTR,
                        .CLASS,
                        .MSKLEN,
                        .MSKPTR);

        TES;
!
! Add the source block to the data base entry.
!
    IF NOT NML$ADDSRC (.BUFDSC, .RESDSC, .SRCPTR)
    THEN
        STATUS = FALSE;
!
! Clean up the sink node filters.
!
    NML$CLEANEVT (.SNK, .RESDSC);
    NML$CLEANSRC (.BUFDSC, .SNK, .RESDSC);

    RETURN .STATUS
```

```
:  236          0234  2
:  237          0235  1       END;                                    ! End of NML$ADDFILTERS


                                                       .TITLE  NML$LOGOPS NML Logging data base operations mod
:                                                                         ule
                                                       .IDENT  \V04-000\

                                                       .PSECT  $PLIT$,NOWRT,NOEXE,2

                              00000014  00000 P.AAA:   .LONG   20                                          ;
                              00000000' 00004          .ADDRESS NML$T_EVTBUFFER                            ;
                              00000400  00008 P.AAB:   .LONG   1024                                        ;
                              00000000' 0000C          .ADDRESS NML$T_SRCBUFFER                            ;

                                                       .PSECT  $OWN$,NOEXE,2

                                    00000 NML$T_EVTBUFFER:
                                                       .BLKB   20
                                    00014 NML$T_SRCBUFFER:
                                                       .BLKB   1024

                                          NML$Q_EVTBFDSC=      P.AAA
                                          NML$Q_SRCBFDSC=      P.AAB
                                                       .EXTRN  NML$GB_EVTSRCTYP
                                                       .EXTRN  NML$GQ_EVTSRCDSC
                                                       .EXTRN  NML$GW_EVTCLASS
                                                       .EXTRN  NML$GB_EVTMSKTYP
                                                       .EXTRN  NML$GQ_EVTMSKDSC
                                                       .EXTRN  NML$GW_EVTSNKADR
                                                       .EXTRN  NML$GW_ACP_CHAN
                                                       .EXTRN  NML$GL_LOGMASK, NML$GQ_ENTSTRDSC
                                                       .EXTRN  NML$AB_QIOBUFFER
                                                       .EXTRN  NML$GQ_QIOBFDSC
                                                       .EXTRN  NML$AB_EXEBUFFER
                                                       .EXTRN  NML$GL_EXEDATPTR
                                                       .EXTRN  NML$GQ_EXEDATDSC
                                                       .EXTRN  NML$GQ_EXEBFDSC
                                                       .EXTRN  NML$AB_RCVBUFFER
                                                       .EXTRN  NML$GQ_RCVBFDSC
                                                       .EXTRN  NML$AB_SNDBUFFER
                                                       .EXTRN  NML$GQ_SNDBFDSC
                                                       .EXTRN  NML$GL_RCVDATLEN
                                                       .EXTRN  NML$AB_CPTABLE, NML$AB_MSGBLOCK
                                                       .EXTRN  NML$AB_ENTITY_ID
                                                       .EXTRN  NML$AB_QUALIFIER_ID
                                                       .EXTRN  NML$AB_ENTITYDATA
                                                       .EXTRN  NML$AB_NML_NMV, NML$AB_PRMSEM
                                                       .EXTRN  NML$AB_RECBUF, NML$AL_ENTINFTAB
                                                       .EXTRN  NML$AL_PERMINFTAB
                                                       .EXTRN  NML$AW_PRM_DES, NML$GB_CMD_VER
                                                       .EXTRN  NML$GB_ENTITY_CODE
                                                       .EXTRN  NML$GB_ENTITY_FORMAT
                                                       .EXTRN  NML$GL_QUALIFIER_PST
                                                       .EXTRN  NML$GB_QUALIFIER_FORMAT
                                                       .EXTRN  NML$GB_FUNCTION
                                                       .EXTRN  NML$GB_INFO, NML$GB_OPTIONS
```

```
                                                          .EXTRN   NML$GL_PRMCODE, NML$GL_PRS_FLGS
                                                          .EXTRN   NML$GL_NML_ENTITY
                                                          .EXTRN   NML$GQ_NETNAMDSC
                                                          .EXTRN   NML$GQ_RECBFDSC
                                                          .EXTRN   NML$GW_PRMDESCNT
                                                          .EXTRN   NML$GK_EVENTS, NML$AB_EVENTS
                                                          .EXTRN   NML$ERROR_2

                                                          .PSECT   $CODE$,NOWRT,2

                                         03FC 00000       .ENTRY   NML$ADDFILTERS, Save R2,R3,R4,R5,R6,R7,R8,- ; 0121
                                                                   R9                                          ;
                     59 00000000'   EF    9E 00002        MOVAB    NML$T_SRCBUFFER, R9                         ;
                     58 00000000'   EF    9E 00009        MOVAB    NML$Q_SRCBFDSC, R8                          ;
                     5E             04    C2 00010        SUBL2    #4, SP                                      ;
                     57             01    D0 00013        MOVL     #1, STATUS                                  ; 0174
                                    5E    DD 00016        PUSHL    SP                                          ; 0178
                     7E       10    AC    7D 00018        MOVQ     SRC, -(SP)                                  ;
                              0C    AC    DD 0001C        PUSHL    SNK                                         ;
                     56       24    AC    D0 0001F        MOVL     RESDSC, R6                                  ;
                                    56    DD 00023        PUSHL    R6                                          ;
            00000000V EF            05    FB 00025        CALLS    #5, NML$MATCHSRC                            ;
                     16             50    E9 0002C        BLBC     R0, 1$                                      ;
        69        00 BE       00    BE    28 0002F        MOVC3    @SRCPTR, @SRCPTR, NML$T_SRCBUFFER           ; 0182
                                    6E    DD 00035        PUSHL    SRCPTR                                      ; 0185
                                    56    DD 00037        PUSHL    R6                                          ;
            00000000V EF            02    FB 00039        CALLS    #2, NML$REMSRC                              ;
                     6E             69    9E 00040        MOVAB    NML$T_SRCBUFFER, SRCPTR                     ; 0186
                                    14    11 00043        BRB      2$                                          ; 0178
                     7E       10    AC    7D 00045 1$:    MOVQ     SRC, -(SP)                                  ; 0192
                              0C    AC    DD 00049        PUSHL    SNK                                         ;
                                    58    DD 0004C        PUSHL    R8                                          ;
            00000000V EF            04    FB 0004E        CALLS    #4, NML$BLDSRC                              ;
                     6E       04    A8    D0 00055        MOVL     NML$Q_SRCBFDSC+4, SRCPTR                    ; 0193
                     50 00000000G   EF    9A 00059 2$:    MOVZBL   NML$GB_EVTMSKTYP, R0                        ; 0199
                     02             50    91 00060        CMPB     R0, #2                                      ; 0202
                                    17    12 00063        BNEQ     3$                                          ;
                              10    AC    DD 00065        PUSHL    SRC                                         ; 0204
                              18    AC    DD 00068        PUSHL    CLASS                                       ;
                              08    AE    DD 0006B        PUSHL    SRCPTR                                      ;
                                    58    DD 0006E        PUSHL    R8                                          ;
                              04    AC    DD 00070        PUSHL    FCT                                         ;
            00000000V EF            05    FB 00073        CALLS    #5, NML_MODCLS                              ;
                                    31    11 0007A        BRB      5$                                          ;
                     03             50    91 0007C 3$:    CMPB     R0, #3                                      ; 0206
                                    14    12 0007F        BNEQ     4$                                          ;
                              10    AC    DD 00081        PUSHL    SRC                                         ; 0208
                              04    AE    DD 00084        PUSHL    SRCPTR                                      ;
                                    58    DD 00087        PUSHL    R8                                          ;
                              04    AC    DD 00089        PUSHL    FCT                                         ;
            00000000V EF            04    FB 0008C        CALLS    #4, NML_MODKNO                              ;
                                    18    11 00093        BRB      5$                                          ;
                     7E       1C    AC    7D 00095 4$:    MOVQ     MSKLEN, -(SP)                               ; 0217
                              18    AC    DD 00099        PUSHL    CLASS                                       ; 0216
                              0C    AE    DD 0009C        PUSHL    SRCPTR                                      ; 0215
                                    58    DD 0009F        PUSHL    R8                                          ; 0212
                                    7E    D4 000A1        CLRL     -(SP)                                       ;
```

```
                                      04     AC   DD 000A3              PUSHL   FCT                                    :
                    00000000V   EF           07   FB 000A6              CALLS   #7, NML_MODFIL                         :
                                             6E   DD 000AD  5$:         PUSHL   SRCPTR                                 : 0224
                                             56   DD 000AF              PUSHL   R6                                     :
                                      08     AC   DD 000B1              PUSHL   BUFDSC                                 :
                    00000000V   EF           03   FB 000B4              CALLS   #3, NML$ADDSRC                         :
                                02           50   E8 000BB              BLBS    R0, 6$                                 :
                                             57   D4 000BE              CLRL    STATUS                                 : 0226
                                             56   DD 000C0  6$:         PUSHL   R6                                     : 0230
                                      0C     AC   DD 000C2              PUSHL   SNK                                    :
                    00000000V   EF           02   FB 000C5              CALLS   #2, NML$CLEANEVT                       :
                                             56   DD 000CC              PUSHL   R6                                     : 0231
                                7E    08     AC   7D 000CE              MOVQ    BUFDSC, -(SP)                          :
                    00000000V   EF           03   FB 000D2              CALLS   #3, NML$CLEANSRC                       :
                                50           57   D0 000D9              MOVL    STATUS, R0                             : 0233
                                                  04 000DC              RET                                            : 0235

; Routine Size:  221 bytes,    Routine Base:  $CODE$ + 0000
```

```
1 %SBTTL 'NML_MODFIL  Modify event filters'
1 ROUTINE NML_MODFIL (FCT, ZER, BUFDSC, SRCPTR, CLASS, MSKLEN, MSKPTR) =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine adds event filters to the data base entry for a sink
1 !       node.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       FCT             Function code. (0=CLEAR/PURGE, 1=SET/DEFINE).
1 !       ZER             Zero mask flag. (TRUE=yes, FALSE=no).
1 !       BUFDSC          Descriptor of buffer to contain modified data base
1 !                       entry.
1 !       SRCPTR          Pointer to source block in buffer.
1 !       CLASS           Event class code.
1 !       MSKLEN          Length of filter mask.
1 !       MSKPTR          Address of filter mask.
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       TRUE is returned if operation is successful.  Otherwise, FALSE
1 !       is returned.
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !
1 !--
1
2   BEGIN
2
2   MAP
2       BUFDSC : REF DESCRIPTOR,
2       SRCPTR : REF BBLOCK;
2
2   LOCAL
2       EVTPTR,                         ! Pointer to event block
2       STATUS;                         ! Routine status code
2
2   STATUS = TRUE;                      ! Initialize return status
2 !
2 ! Get the event block.
2 !
2   IF NML$MATCHEVT (.SRCPTR,
2                   .CLASS,
2                   EVTPTR)
```

```
;   296        0293  2         THEN
;   297        0294  3             BEGIN
;   298        0295  3
;   299        0296  3             NML$MODEVT (.FCT, .ZER, .EVTPTR, .MSKLEN, .MSKPTR);
;   300        0297  3
;   301        0298  3             END
;   302        0299  2         ELSE
;   303        0300  3             BEGIN
;   304        0301  3
;   305        0302  3             NML$BLDEVT (.FCT, .CLASS, .MSKLEN, .MSKPTR, NML$T_EVTBUFFER);
;   306        0303  3             EVTPTR = NML$T_EVTBUFFER;
;   307        0304  3 !
;   308        0305  3 ! Add the event block to the source block.
;   309        0306  3 !
;   310        0307  3             IF NOT NML$ADDEVT (.BUFDSC, .SRCPTR, .EVTPTR)
;   311        0308  3             THEN
;   312        0309  3                 STATUS = FALSE;
;   313        0310  3
;   314        0311  2             END;
;   315        0312  2
;   316        0313  2         RETURN .STATUS
;   317        0314  2
;   318        0315  1         END;                                    ! End of NML_MODFIL





                                          000C 00000 NML_MODFIL:
                                                                .WORD   Save R2,R3                                      ; 0237
                        53 00000000'   EF   9E 00002            MOVAB   NML$T_EVTBUFFER, R3                             ;
                        5E             04   C2 00009            SUBL2   #4, SP                                          ;
                        52             01   D0 0000C            MOVL    #1, STATUS                                      ; 0286
                                       5E   DD 0000F            PUSHL   SP                                              ; 0290
                        7E         10  AC   7D 00011            MOVQ    SRCPTR, -(SP)                                   ;
              00000000V EF             03   FB 00015            CALLS   #3, NML$MATCHEVT                                ;
                        14             50   E9 0001C            BLBC    R0, 1$                                          ;
                        7E         18  AC   7D 0001F            MOVQ    MSKLEN, -(SP)                                   ; 0296
                                   08  AE   DD 00023            PUSHL   EVTPTR                                          ;
                        7E         04  AC   7D 00026            MOVQ    FCT, -(SP)                                      ;
              00000000V EF             05   FB 0002A            CALLS   #5, NML$MODEVT                                  ;
                                       28   11 00031            BRB     2$                                              ; 0290
                                       53   DD 00033 1$:        PUSHL   R3                                              ; 0302
                        7E         18  AC   7D 00035            MOVQ    MSKLEN, -(SP)                                   ;
                                   14  AC   DD 00039            PUSHL   CLASS                                           ;
                                   04  AC   DD 0003C            PUSHL   FCT                                             ;
              00000000V EF             05   FB 0003F            CALLS   #5, NML$BLDEVT                                  ;
                        6E             63   9E 00046            MOVAB   NML$T_EVTBUFFER, EVTPTR                         ; 0303
                                       6E   DD 00049            PUSHL   EVTPTR                                          ; 0307
                        7E         0C  AC   7D 0004B            MOVQ    BUFDSC, -(SP)                                   ;
              00000000V EF             03   FB 0004F            CALLS   #3, NML$ADDEVT                                  ;
                        02             50   E8 00056            BLBS    R0, 2$                                          ;
                                       52   D4 00059            CLRL    STATUS                                          ; 0309
                        50             52   D0 0005B 2$:        MOVL    STATUS, R0                                      ; 0313
                                            04 0005E            RET                                                     ; 0315

; Routine Size:  95 bytes,    Routine Base:  $CODE$ + 00DD
```

```
1 %SBTTL 'NML_MODCLS  Modify class filters'
1 ROUTINE NML_MODCLS (FCT, BUFDSC, SRCPTR, CLASS, SRC) =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine adds event filters to the data base entry for a sink
1 !       node.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       FCT             Function code. (0=CLEAR/PURGE, 1=SET/DEFINE)
1 !       BUFDSC          Descriptor of buffer to contain modified data base
1 !                       entry.
1 !       SRCPTR          Pointer to source block in buffer.
1 !       CLASS           Event class code.
1 !       SRC             Source type code.
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       TRUE is returned if operation is successful.  Otherwise, FALSE
1 !       is returned.
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !
1 !--
1
2     BEGIN
2
2     MAP
2         BUFDSC : REF DESCRIPTOR,
2         SRCPTR : REF BBLOCK,
2         CLASS  : WORD;
2
2     LOCAL
2         MSK,                              ! Address of filter mask
2         STATUS;                           ! Routine status code
2
2     MSK = UPLIT (-1, 0);
2
2     IF .FCT
2     THEN
2
2     INCR I FROM 0 TO NML$GK_EVENTS - 1 DO
3         BEGIN
3
```

```
;   377         0373  3              IF .NML$AB_EVENTS [.I, ETB$W_CLASS] EQLU .CLASS
;   378         0374  3              THEN
;   379         0375  4                  BEGIN
;   380         0376  4
;   381         0377  4                  SELECTONEU .SRC OF
;   382         0378  4                      SET
;   383         0379  4
;   384         0380  4                  [NMA$C_ENT_NOD]:            ! Node
;   385         0381  4                      MSK = .NML$AB_EVENTS [.I, ETB$A_NODE];
;   386         0382  4
;   387         0383  4                  [NMA$C_ENT_CIR]:            ! Circuit
;   388         0384  4                      MSK = .NML$AB_EVENTS [.I, ETB$A_CIRCUIT];
;   389         0385  4
;   390         0386  4                  [NMA$C_ENT_LIN]:            ! Line
;   391         0387  4                      MSK = .NML$AB_EVENTS [.I, ETB$A_LINE];
;   392         0388  4
;   393         0389  4                  [NMA$C_ENT_MOD]:            ! Module
;   394         0390  4                      MSK = .NML$AB_EVENTS [.I, ETB$A_MODULE];
;   395         0391  4
;   396         0392  4                  [OTHERWISE]:                ! Must be global
;   397         0393  4                      MSK = .NML$AB_EVENTS [.I, ETB$A_GLOBAL];
;   398         0394  4
;   399         0395  4                  TES;
;   400         0396  4
;   401         0397  4                  EXITLOOP;
;   402         0398  4
;   403         0399  3                  END;
;   404         0400  2              END;
;   405         0401  2
;   406         0402  2          STATUS = NML_MODFIL (.FCT,
;   407         0403  2                               TRUE,
;   408         0404  2                               .BUFDSC,
;   409         0405  2                               .SRCPTR,
;   410         0406  2                               .CLASS,
;   411         0407  2                               EVT$S_LOGMSK,
;   412         0408  2                               .MSK);
;   413         0409  2
;   414         0410  2          RETURN .STATUS
;   415         0411  2
;   416         0412  1          END;                                ! End of NML_MODCLS


                                                            .PSECT  $PLIT$,NOWRT,NOEXE,2

                        00000000  FFFFFFFF  00010 P.AAC:    .LONG   -1, 0                                             ;


                                                            .PSECT  $CODE$,NOWRT,2

                                         001C 00000 NML_MODCLS:
                                                            .WORD   Save R2,R3,R4                                     ; 0317
                        54 00000000G  EF   9E 00002         MOVAB   NML$AB_EVENTS, R4                                 ;
                        53 00000000'  EF   9E 00009         MOVAB   P.AAC, MSK                                        ; 0365
                        50         04  AC   E9 00010         BLBC    FCT, 8$                                           ; 0367
                        50             01   CE 00014         MNEGL   #1, I                                             ; 0373
```

```
                                          43  11 00017          BRB      7$                                   ;
            51         50                 16  C5 00019 1$:      MULL3    #22, I, R1                           ;
                                        6441  9F 0001D          PUSHAB   NML$AB_EVENTS[R1]                    ;
                  10   AC                 9E  B1 00020          CMPW     @(SP)+, CLASS                        ;
                                          36  12 00024          BNEQ     7$                                   ;
                       52         14      AC  D0 00026          MOVL     SRC, R2                              ; 0377
                                          06  12 0002A          BNEQ     2$                                   ; 0380
                                  06 A441  9F 0002C          PUSHAB   NML$AB_EVENTS+6[R1]                  ; 0381
                                          25  11 00030          BRB      6$                                   ;
                       03                 52  D1 00032 2$:      CMPL     R2, #3                               ; 0383
                                          06  12 00035          BNEQ     3$                                   ;
                                  0A A441  9F 00037          PUSHAB   NML$AB_EVENTS+10[R1]                 ; 0384
                                          1A  11 0003B          BRB      6$                                   ;
                       01                 52  D1 0003D 3$:      CMPL     R2, #1                               ; 0386
                                          06  12 00040          BNEQ     4$                                   ;
                                  0E A441  9F 00042          PUSHAB   NML$AB_EVENTS+14[R1]                 ; 0387
                                          0F  11 00046          BRB      6$                                   ;
                       04                 52  D1 00048 4$:      CMPL     R2, #4                               ; 0389
                                          06  12 0004B          BNEQ     5$                                   ;
                                  12 A441  9F 0004D          PUSHAB   NML$AB_EVENTS+18[R1]                 ; 0390
                                          04  11 00051          BRB      6$                                   ;
                                  02 A441  9F 00053 5$:      PUSHAB   NML$AB_EVENTS+2[R1]                  ; 0393
                       53                 9E  D0 00057 6$:      MOVL     @(SP)+, MSK                          ;
                                          08  11 0005A          BRB      8$                                   ; 0375
            B5         50 00000000G       8F  F3 0005C 7$:      AOBLEQ   #NML$GK_EVENTS-1, I, 1$              ; 0370
                                          53  DD 00064 8$:      PUSHL    MSK                                  ; 0408
                                          08  DD 00066          PUSHL    #8                                   ; 0402
                       7E         10      AC  3C 00068          MOVZWL   CLASS, -(SP)                         ; 0406
                       7E         08      AC  7D 0006C          MOVQ     BUFDSC, -(SP)                        ; 0404
                                          01  DD 00070          PUSHL    #1                                   ; 0402
                                  04      AC  DD 00072          PUSHL    FCT                                  ;
                  FF27  CF                07  FB 00075          CALLS    #7, NML_MODFIL                       ;
                                          04  0007A          RET                                           ; 0412

; Routine Size:  123 bytes,    Routine Base:  $CODE$ + 013C
```

```
1 %SBTTL 'NML_MODKNO  Modify known filters'
1 ROUTINE NML_MODKNO (FCT, BUFDSC, SRCPTR, SRC) =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine adds event filters to the data base entry for a sink
1 !       node.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       FCT             Function code. (0=CLEAR/PURGE, 1=SET/DEFINE)
1 !       BUFDSC          Descriptor of buffer to contain modified data base
1 !                       entry.
1 !       SRCPTR          Pointer to source block in buffer.
1 !       SRC             Source type code.
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       TRUE is returned if operation is successful.  Otherwise, FALSE
1 !       is returned.
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !
1 !--
1
2     BEGIN
2
2     MAP
2         BUFDSC : REF DESCRIPTOR,
2         SRCPTR : REF BBLOCK;
2
2     LOCAL
2         CLASS  : WORD,
2         EVTPTR : REF BBLOCK,
2         MSK,
2         STATUS;                         ! Routine status code
2
2     STATUS = FALSE;
2
2     INCR I FROM 0 TO NML$GK_EVENTS - 1 DO
3         BEGIN
3
3         CLASS = .NML$AB_EVENTS [.I, ETB$W_CLASS];
3
3         SELECTONEU .SRC OF
```

```
3               SET
3
3               [NMA$C_ENT_NOD]:                  ! Node
3                   MSK = .NML$AB_EVENTS [.I, ETB$A_NODE];
3
3               [NMA$C_ENT_CIR]:                  ! Circuit
3                   MSK = .NML$AB_EVENTS [.I, ETB$A_CIRCUIT];
3
3               [NMA$C_ENT_LIN]:                  ! Line
3                   MSK = .NML$AB_EVENTS [.I, ETB$A_LINE];
3
3               [NMA$C_ENT_MOD]:                  ! Line
3                   MSK = .NML$AB_EVENTS [.I, ETB$A_MODULE];
3
3               [OTHERWISE]:                      ! Must be global
3                   MSK = .NML$AB_EVENTS [.I, ETB$A_GLOBAL];
3
3               TES;
3
3           STATUS = NML_MODFIL (.FCT,
3                                TRUE,
3                                .BUFDSC,
3                                .SRCPTR,
3                                .CLASS,
3                                EVT$S_LOGMSK,
3                                .MSK);
3           IF NOT .STATUS
3           THEN
3               EXITLOOP;
3
2           END;
2   !
2   ! If the function is clear and everything is alright up to this point then
2   ! go through all event classes that are present in the source block and clear
2   ! out all the filters.  This covers the case where filters are present for
2   ! an unknown class.
2   !
2       IF .STATUS
2       AND NOT .FCT
2       THEN
3           BEGIN
3
3           EVTPTR = 0;
3           WHILE NML$GETNXTEVT (.SRCPTR, EVTPTR) DO
4               BEGIN
4
4               CLASS = .EVTPTR [EVT$W_CLASS];
4               NML$MODEVT (.FCT, FALSE, .EVTPTR, EVT$S_LOGMSK, UPLIT (-1, -1));
4
3               END;
3
2           END;
2
2       RETURN .STATUS
2
1       END;                                      ! End of NML_MODKNO
```

```
                                                          .PSECT  $PLIT$,NOWRT,NOEXE,2

                    FFFFFFFF  FFFFFFFF  00018 P.AAD:      .LONG   -1, -1                                   ;


                                                          .PSECT  $CODE$,NOWRT,2

                                  007C 00000 NML_MODKNO:
                                                          .WORD   Save R2,R3,R4,R5,R6                      ; 0414
                    56 00000000G   EF   9E 00002          MOVAB   NML$AB_EVENTS, R6                        ;
                                   54   D4 00009          CLRL    STATUS                                   ; 0462
                    52             01   CE 0000B          MNEGL   #1, I                                    ; 0489
                                   59   10 0000E          BSBB    7$                                       ;
          50        52             16   C5 00010 1$:      MULL3   #22, I, R0                               ; 0467
                                 6640   9F 00014          PUSHAB  NML$AB_EVENTS[R0]                        ;
                    55             9E   B0 00017          MOVW    @(SP)+, CLASS                            ;
                    51        10   AC   D0 0001A          MOVL    SRC, R1                                  ; 0469
                                   06   12 0001E          BNEQ    2$                                       ; 0472
                              06 A640   9F 00020          PUSHAB  NML$AB_EVENTS+6[R0]                      ; 0473
                                   25   11 00024          BRB     6$                                       ;
                    03             51   D1 00026 2$:      CMPL    R1, #3                                   ; 0475
                                   06   12 00029          BNEQ    3$                                       ;
                              0A A640   9F 0002B          PUSHAB  NML$AB_EVENTS+10[R0]                     ; 0476
                                   1A   11 0002F          BRB     6$                                       ;
                    01             51   D1 00031 3$:      CMPL    R1, #1                                   ; 0478
                                   06   12 00034          BNEQ    4$                                       ;
                              0E A640   9F 00036          PUSHAB  NML$AB_EVENTS+14[R0]                     ; 0479
                                   0F   11 0003A          BRB     6$                                       ;
                    04             51   D1 0003C 4$:      CMPL    R1, #4                                   ; 0481
                                   06   12 0003F          BNEQ    5$                                       ;
                              12 A640   9F 00041          PUSHAB  NML$AB_EVENTS+18[R0]                     ; 0482
                                   04   11 00045          BRB     6$                                       ;
                              02 A640   9F 00047 5$:      PUSHAB  NML$AB_EVENTS+2[R0]                      ; 0485
                    53             9E   D0 0004B 6$:      MOVL    @(SP)+, MSK                              ;
                                   53   DD 0004E          PUSHL   MSK                                      ; 0495
                                   08   DD 00050          PUSHL   #8                                       ; 0489
                    7E             55   3C 00052          MOVZWL  CLASS, -(SP)                             ; 0493
                    7E        08   AC   7D 00055          MOVQ    BUFDSC, -(SP)                            ; 0491
                                   01   DD 00059          PUSHL   #1                                       ; 0489
                              04   AC   DD 0005B          PUSHL   FCT                                      ;
               FEC3 CF             07   FB 0005E          CALLS   #7, NML_MODFIL                           ;
                    54             50   D0 00063          MOVL    R0, STATUS                               ;
                    3D             54   E9 00066          BLBC    STATUS, 9$                               ; 0496
          9F        52 00000000G   8F   F3 00069 7$:      AOBLEQ  #NML$GK_EVENTS-1, I, 1$                  ; 0464
                    32             54   E9 00071          BLBC    STATUS, 9$                               ; 0507
                    2E        04   AC   E8 00074          BLBS    FCT, 9$                                  ; 0508
                                   6E   D4 00078          CLRL    EVTPTR                                   ; 0512
                                   5E   DD 0007A 8$:      PUSHL   SP                                       ; 0513
                              0C   AC   DD 0007C          PUSHL   SRCPTR                                   ;
            00000000V EF           02   FB 0007F          CALLS   #2, NML$GETNXTEVT                        ;
                    1D             50   E9 00086          BLBC    R0, 9$                                   ;
                    55        00   BE   B0 00089          MOVW    @EVTPTR, CLASS                           ; 0516
                       00000000'   EF   9F 0008D          PUSHAB  P.AAD                                    ; 0517
                                   08   DD 00093          PUSHL   #8                                       ;
```

```
                                        08      AE  DD 00095         PUSHL   EVTPTR                              ;
                                                7E  D4 00098         CLRL    -(SP)                               ;
                                        04      AC  DD 0009A         PUSHL   FCT                                 ;
                        00000000V  EF           05  FB 0009D         CALLS   #5, NML$MODEVT                      ;
                                                D4  11 000A4         BRB     8$                                  ; 0513
                                   50           54  D0 000A6 9$:     MOVL    STATUS, R0                          ; 0523
                                                    04 000A9         RET                                         ; 0525

; Routine Size:  170 bytes,     Routine Base:  $CODE$ + 01B7
```

```
%SBTTL 'NML$GETSPCFILTERS  Get event filters'
GLOBAL ROUTINE NML$GETSPCFILTERS
                (DATDSC, SNK, SRC, ENTDSC, CLASS, MSKPTR, RESLEN) =

!++
! FUNCTIONAL DESCRIPTION:
!
!       This routine gets event filters for the specified source and class.
!
! FORMAL PARAMETERS:
!
!       DATDSC          Descriptor of current data base entry.
!       SNK             Logging sink type code.
!       SRC             Event source type code.
!       ENTDSC          Event source id string descriptor.
!       CLASS           Event class code.
!       MSKPTR          Address of filter mask quadword.
!       RESLEN          Address of longword to contain byte count of
!                       resulting mask.
!
! IMPLICIT INPUTS:
!
!       NONE
!
! IMPLICIT OUTPUTS:
!
!       NONE
!
! ROUTINE VALUE:
! COMPLETION CODES:
!
!       TRUE is returned if operation is successful.  Otherwise, FALSE
!       is returned.
!
! SIDE EFFECTS:
!
!       NONE
!
!--

    BEGIN

    MAP
        DATDSC : REF DESCRIPTOR,
        ENTDSC : REF DESCRIPTOR,
        MSKPTR : REF BITVECTOR;

    LOCAL
        EVTPTR : REF BBLOCK,            ! Pointer to event block
        FILPTR : REF BITVECTOR,         ! Pointer to event filter mask
        LOGPTR : REF BITVECTOR,         ! Pointer to event log mask
        SRCPTR,                         ! Pointer to source block
        ZERCNT;                         ! Trailing zero byte count
!
! Get the source block.
!
```

```
;  589      0583  2         IF NOT NML$MATCHSRC (.DATDSC, .SNK, .SRC, .ENTDSC, SRCPTR)
;  590      0584  2         THEN
;  591      0585  2             RETURN FALSE;
;  592      0586  2   !
;  593      0587  2   ! Get the event block.
;  594      0588  2   !
;  595      0589  2         IF NOT NML$MATCHEVT (.SRCPTR, .CLASS, EVTPTR)
;  596      0590  2         THEN
;  597      0591  2             RETURN FALSE;
;  598      0592  2   !
;  599      0593  2   ! Get combined specific and global filters.
;  600      0594  2   !
;  601      0595  2         IF NOT NML$GETCOMFILTERS (.DATDSC, .SNK, .CLASS, .MSKPTR, .RESLEN)
;  602      0596  2         THEN
;  603      0597  2             RETURN FALSE;
;  604      0598  2
;  605      0599  2         RETURN TRUE
;  606      0600  2
;  607      0601  1         END;                                       ! End of NML$GETSPCFILTERS



                                          0000 00000          .ENTRY   NML$GETSPCFILTERS, Save nothing                   ; 0527
                          5E           08  C2 00002          SUBL2    #8, SP                                            ;
                                       5E  DD 00005          PUSHL    SP                                                ; 0583
                          7E      0C   AC  7D 00007          MOVQ     SRC, -(SP)                                        ;
                          7E      04   AC  7D 0000B          MOVQ     DATDSC, -(SP)                                     ;
                00000000V EF           05  FB 0000F          CALLS    #5, NML$MATCHSRC                                  ;
                          2C           50  E9 00016          BLBC     R0, 1$                                            ;
                                  04   AE  9F 00019          PUSHAB   EVTPTR                                            ; 0589
                                  14   AC  DD 0001C          PUSHL    CLASS                                             ;
                                  08   AE  DD 0001F          PUSHL    SRCPTR                                            ;
                00000000V EF           03  FB 00022          CALLS    #3, NML$MATCHEVT                                  ;
                          19           50  E9 00029          BLBC     R0, 1$                                            ;
                          7E      18   AC  7D 0002C          MOVQ     MSKPTR, -(SP)                                     ; 0595
                                  14   AC  DD 00030          PUSHL    CLASS                                             ;
                          7E      04   AC  7D 00033          MOVQ     DATDSC, -(SP)                                     ;
                00000000V EF           05  FB 00037          CALLS    #5, NML$GETCOMFILTERS                             ;
                          04           50  E9 0003E          BLBC     R0, 1$                                            ;
                          50           01  D0 00041          MOVL     #1, R0                                            ; 0599
                                           04 00044          RET                                                        ;
                                       50  D4 00045 1$:      CLRL     R0                                                ; 0601
                                           04 00047          RET                                                        ;

; Routine Size:  72 bytes,    Routine Base:  $CODE$ + 0261
```

```
1 %SBTTL 'NML$GETCOMFILTERS  Get event filters'
1 GLOBAL ROUTINE NML$GETCOMFILTERS (DATDSC, EVTPTR, SNK, MSKPTR, RESLEN) =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine gets event filters from the specified event block
1 !       and combines them with the global filters for the class.  The
1 !       resulting mask is the complete event mask for the class and source.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       DATDSC          Descriptor of current data base entry.
1 !       EVTPTR          Pointer to event block.
1 !       SNK             Event sink type code.
1 !       MSKPTR          Address of filter mask quadword.
1 !       RESLEN          Address of longword to contain byte count of
1 !                       resulting mask.
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       TRUE is returned if operation is successful.  Otherwise, FALSE
1 !       is returned.
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !
1 !--
1
2     BEGIN
2
2     MAP
2         DATDSC : REF DESCRIPTOR,
2         EVTPTR : REF BBLOCK,                ! Pointer to event block
2         MSKPTR : REF BITVECTOR;
2
2     LOCAL
2         CLASS,                              ! Event class
2         FILPTR : REF BITVECTOR,             ! Pointer to event filter mask
2         LOGPTR : REF BITVECTOR,             ! Pointer to event log mask
2         ZERCNT;                             ! Trailing zero byte count
2
2 !
2 ! Get global filter mask for this class.
2 !
2     CLASS = .EVTPTR [EVT$W_CLASS];
2     NML$GETGBLFILTERS (.DATDSC, .SNK, .CLASS, .MSKPTR);
```

```
;   666         0659 2 !
;   667         0660 2 ! Combine specific masks with global mask.
;   668         0661 2 !
;   669         0662 2   LOGPTR = EVTPTR [EVT$Q_LOGMSK];
;   670         0663 2   FILPTR = EVTPTR [EVT$Q_FILTERMSK];
;   671         0664 2
;   672         0665 2   INCR I FROM 0 TO (EVT$S_LOGMSK * 8) - 1 DO
;   673         0666 3       BEGIN
;   674         0667 3
;   675         0668 3       MSKPTR [.I] = .MSKPTR [.I] OR .LOGPTR [.I];
;   676         0669 3       MSKPTR [.I] = .MSKPTR [.I] AND NOT .FILPTR [.I];
;   677         0670 3
;   678         0671 2       END;
;   679         0672 2 !
;   680         0673 2 ! Adjust count to exclude zero bytes on the end of the quadword mask.
;   681         0674 2 !
;   682         0675 2   ZERCNT = 0;
;   683         0676 2
;   684         0677 2   DECR I FROM EVT$S_LOGMSK - 1 DO
;   685         0678 3       BEGIN
;   686         0679 3
;   687         0680 3       IF .(.MSKPTR + .I)<0,8> EQLU 0
;   688         0681 3       THEN
;   689         0682 3           ZERCNT = .ZERCNT + 1
;   690         0683 3       ELSE
;   691         0684 3           EXITLOOP;
;   692         0685 3
;   693         0686 2       END;
;   694         0687 2 !
;   695         0688 2 ! Set up mask length for return.
;   696         0689 2 !
;   697         0690 2   .RESLEN = EVT$S_LOGMSK - .ZERCNT;
;   698         0691 2
;   699         0692 2   RETURN TRUE
;   700         0693 2
;   701         0694 1   END;                                   ! End of NML$GETCOMFILTERS





                                          003C 00000           .ENTRY  NML$GETCOMFILTERS, Save R2,R3,R4,R5              ; 0603
                     50        08    BC    3C 00002           MOVZWL  @EVTPTR, CLASS                                   ; 0657
                     53        10    AC    D0 00006           MOVL    MSKPTR, R3                                       ; 0658
                                     09    BB 0000A           PUSHR   #^M<R0,R3>                                       ;
                               0C    AC    DD 0000C           PUSHL   SNK                                              ;
                               04    AC    DD 0000F           PUSHL   DATDSC                                           ;
           00000000V  EF             04    FB 00012           CALLS   #4, NML$GETGBLFILTERS                            ;
                54         08    AC    04    C1 00019         ADDL3   #4, EVTPTR, LOGPTR                               ; 0662
                55         08    AC    0C    C1 0001E         ADDL3   #12, EVTPTR, FILPTR                              ; 0663
                                       51    D4 00023         CLRL    I                                                ; 0665
     52         63         01          51    EF 00025  1$:    EXTZV   I, #1, (R3), R2                                  ; 0668
     50         64         01          51    EF 0002A         EXTZV   I, #1, (LOGPTR), R0                              ;
                           50          52    C8 0002F         BISL2   R2, R0                                           ;
     63         01         51          50    F0 00032         INSV    R0, I, #1, (R3)                                  ;
     52         63         01          51    EF 00037         EXTZV   I, #1, (R3), R2                                  ; 0669
     50         65         01          51    EF 0003C         EXTZV   I, #1, (FILPTR), R0                              ;
```

```
                                 52          50  CA 00041          BICL2   R0, R2                                  ;
        63            01         51          52  F0 00044          INSV    R2, I, #1, (R3)                         ;
                      D8         51          3F  F3 00049          AOBLEQ  #63, I, 1$                              ; 0665
                                 50          07  7D 0004D          MOVQ    #7, I                                   ; 0677
                                           6043  95 00050 2$:      TSTB    (I)[R3]                                 ; 0680
                                             05  12 00053          BNEQ    3$                                      ;
                                             51  D6 00055          INCL    ZERCNT                                  ; 0682
                                 F6          50  F4 00057          SOBGEQ  I, 2$                                   ; 0677
               14     BC         08          51  C3 0005A 3$:      SUBL3   ZERCNT, #8, @RESLEN                     ; 0690
                                 50          01  D0 0005F          MOVL    #1, R0                                  ; 0692
                                                 04 00062          RET                                             ; 0694

; Routine Size:  99 bytes,     Routine Base:  $CODE$ + 02A9
```

```
1 %SBTTL 'NML$GETGBLFILTERS  Get global filters for sink and class'
1 GLOBAL ROUTINE NML$GETGBLFILTERS (DATDSC, SNK, CLASS, MSKPTR) =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine returns the global filters for the specified
1 !       sink type and class.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       DATDSC          Descriptor of source block buffer.
1 !       SNK             Logging sink type code.
1 !       CLASS           Event class code.
1 !       MSKPTR          Pointer to quadword to contain global filter mask.
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       TRUE is returned if global filters are found, FALSE is returned
1 !       if no global filters are found.  If no global filters are found
1 !       the resulting filter mask will be zeroed.
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !
1 !--
1
2   BEGIN
2
2   LOCAL
2       EVTPTR : REF BBLOCK,            ! Event block pointer
2       SRCPTR : REF BBLOCK,            ! Source block pointer
2       STATUS;                         ! Routine status
2 !
2 ! Zero the filter mask.
2 !
2   CH$FILL (0, EVT$S_LOGMSK, .MSKPTR);
2 !
2 ! If global filters are found then just return.
2 !
2   IF NOT NML$MATCHSRC (.DATDSC,
2                        .SNK,
2                        NMA$C_ENT_KNO,
2                        UPLIT (0, 0),
2                        SRCPTR)
2   THEN
2       RETURN FALSE;
```

```
;  760          0752  2
;  761          0753  2 !
;  762          0754  2 ! If global filters are found for the specified class then move them
;  763          0755  2 ! into the result mask.
;  764          0756  2 !
;  765          0757  2     IF NML$MATCHEVT (.SRCPTR,
;  766          0758  2                      .CLASS,
;  767          0759  2                      EVTPTR)
;  768          0760  2     THEN
;  769          0761  3         BEGIN
;  770          0762  3
;  771          0763  3         CH$MOVE (EVT$S_LOGMSK,
;  772          0764  3                  EVTPTR [EVT$Q_LOGMSK],
;  773          0765  3                  .MSKPTR);
;  774          0766  3         STATUS = TRUE;
;  775          0767  3
;  776          0768  3         END
;  777          0769  2     ELSE
;  778          0770  2         STATUS = FALSE;
;  779          0771  2
;  780          0772  2     RETURN .STATUS
;  781          0773  2
;  782          0774  1     END;                                            ! End of NML$GETGBLFILTERS



                                                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

                                     00000000  00000000  00020 P.AAE:           .LONG   0, 0                                                   ;


                                                                                .PSECT  $CODE$,NOWRT,2

                                                     003C 00000                 .ENTRY  NML$GETGBLFILTERS, Save R2,R3,R4,R5                    ; 0696
                                   5E             08  C2 00002                  SUBL2   #8, SP                                                 ;
         08            00          6E             00  2C 00005                  MOVC5   #0, (SP), #0, #8, @MSKPTR                              ; 0741
                                           10     BC     0000A
                                                  5E  DD 0000C                  PUSHL   SP                                                     ; 0745
                                   00000000'      EF  9F 0000E                  PUSHAB  P.AAE                                                  ; 0748
                                   7E             01  CE 00014                  MNEGL   #1, -(SP)                                              ; 0745
                                   7E      04     AC  7D 00017                  MOVQ    DATDSC, -(SP)                                          ;
                         0000000V  EF             05  FB 0001B                  CALLS   #5, NML$MATCHSRC                                       ;
                                   21             50  E9 00022                  BLBC    R0, 1$                                                 ;
                                           04     AE  9F 00025                  PUSHAB  EVTPTR                                                 ; 0757
                                           0C     AC  DD 00028                  PUSHL   CLASS                                                  ; 0758
                                           08     AE  DD 0002B                  PUSHL   SRCPTR                                                 ; 0757
                         0000000V  EF             03  FB 0002E                  CALLS   #3, NML$MATCHEVT                                       ;
                                   0E             50  E9 00035                  BLBC    R0, 1$                                                 ;
                                   50      04     AE  D0 00038                  MOVL    EVTPTR, R0                                             ; 0764
               10    BC      04    A0             08  28 0003C                  MOVC3   #8, 4(R0), @MSKPTR                                     ; 0765
                                   50             01  D0 00042                  MOVL    #1, STATUS                                             ; 0766
                                                  04     00045                  RET                                                            ; 0757
                                           50     D4     00046 1$:              CLRL    R0                                                     ; 0774
                                                  04     00048                  RET                                                            ;

; Routine Size:  73 bytes,    Routine Base:  $CODE$ + 030C
```

```
1 %SBTTL 'NML$CLEANEVT  Clean event masks'
1 GLOBAL ROUTINE NML$CLEANEVT (SNK, BLKDSC) : NOVALUE =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine runs through all sources for the specified sink type
1 !       and deletes all event filters that match the global filters.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       SNK                 Logging sink type code.
1 !       BLKDSC              Descriptor of all source block data.
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !
1 !--
1
2     BEGIN
2
2     LOCAL
2         EVTPTR : REF BBLOCK,
2         FILMSK : REF BITVECTOR,
2         GBLEVT : REF BBLOCK,
2         GBLMSK : REF BITVECTOR,
2         LOGMSK : REF BITVECTOR,
2         GBLSRC : REF BBLOCK,
2         SRCPTR : REF BBLOCK,
2         STATUS;
2 !
2 ! If there are no global filters then just clean up the filter masks.
2 !
2     IF NML$MATCHSRC (.BLKDSC, .SNK, NMA$C_ENT_KNO, 0, GBLSRC)
2     THEN
3         BEGIN
3 !
3 ! Make sure the filter mask is zeroed for the global filters.
3 !
3         GBLEVT = 0;
3         WHILE NML$GETNXTEVT (.GBLSRC, GBLEVT) DO
4             BEGIN
4
```

```
;  841      0832 4                  GBLMSK = GBLEVT [EVT$Q_FILTERMSK];
;  842      0833 4
;  843      0834 4                  INCR I FROM 0 TO (EVT$S_FILTERMSK * 8) - 1 DO
;  844      0835 5                      BEGIN
;  845      0836 5
;  846      0837 5                      GBLMSK [.I] = 0;
;  847      0838 5
;  848      0839 4                      END;
;  849      0840 3                  END;
;  850      0841 3              END
;  851      0842 2          ELSE
;  852      0843 2              GBLSRC = 0;
;  853      0844 2  !
;  854      0845 2  ! For every source clean up all event masks.
;  855      0846 2  !
;  856      0847 2          SRCPTR = 0;
;  857      0848 2          WHILE NML$GETNXTSNK (.BLKDSC, .SNK, SRCPTR) DO
;  858      0849 3              BEGIN
;  859      0850 3              IF .(SRCPTR [SRC$B_SRCTYPE])<0,8,1> NEQ NMA$C_ENT_KNO
;  860      0851 3              THEN
;  861      0852 4                  BEGIN
;  862      0853 4  !
;  863      0854 4  ! For every event mask get rid of everything that matches the global
;  864      0855 4  ! filters.
;  865      0856 4  !
;  866      0857 4                  EVTPTR = 0;
;  867      0858 4                  WHILE NML$GETNXTEVT (.SRCPTR, EVTPTR) DO
;  868      0859 5                      BEGIN
;  869      0860 5
;  870      0861 5                      LOGMSK = EVTPTR [EVT$Q_LOGMSK];
;  871      0862 5                      FILMSK = EVTPTR [EVT$Q_FILTERMSK];
;  872      0863 5
;  873      0864 5                      IF .GBLSRC NEQA 0
;  874      0865 5                      THEN
;  875      0866 5                          STATUS = NML$MATCHEVT (.GBLSRC,
;  876      0867 5                                                 .EVTPTR [EVT$W_CLASS],
;  877      0868 5                                                 GBLEVT)
;  878      0869 5                      ELSE
;  879      0870 5                          STATUS = FALSE;
;  880      0871 5
;  881      0872 5                      IF .STATUS
;  882      0873 6                      AND (.GBLSRC NEQA 0)
;  883      0874 5                      THEN
;  884      0875 6                          BEGIN
;  885      0876 6
;  886      0877 6                          GBLMSK = GBLEVT [EVT$Q_LOGMSK];
;  887      0878 6
;  888      0879 6                          INCR I FROM 0 TO (EVT$S_LOGMSK * 8) - 1 DO
;  889      0880 7                              BEGIN
;  890      0881 7
;  891      0882 7                              LOGMSK [.I] = .LOGMSK [.I] AND NOT .GBLMSK [.I];
;  892      0883 7                              FILMSK [.I] = .FILMSK [.I] AND .GBLMSK [.I];
;  893      0884 7
;  894      0885 6                              END;
;  895      0886 6                          END
;  896      0887 5                      ELSE
;  897      0888 6                          BEGIN
;   898             0889   6                        INCR I FROM 0 TO (EVT$S_LOGMSK * 8) - 1 DO
;   899             0890   7                            BEGIN
;   900             0891   7
;   901             0892   7                            FILMSK [.I] = 0;
;   902             0893   7
;   903             0894   6                            END;
;   904             0895   5                        END;
;   905             0896   4                    END;
;   906             0897   3                END;
;   907             0898   2            END;
;   908             0899   2
;   909             0900   1        END;                                        ! End of NML$CLEANEVT





                                                  03FC 00000           .ENTRY   NML$CLEANEVT, Save R2,R3,R4,R5,R6,R7,R8,R9    ; 0776
                          59 00000000V  EF    9E 00002                 MOVAB    NML$GETNXTEVT, R9                             ;
                                   5E    10    C2 00009                SUBL2    #16, SP                                       ;
                                         5E    DD 0000C                PUSHL    SP                                            ; 0822
                                         7E    D4 0000E                CLRL     -(SP)                                         ;
                                   7E    01    CE 00010                MNEGL    #1, -(SP)                                     ;
                                  04    AC    DD 00013                 PUSHL    SNK                                           ;
                                  08    AC    DD 00016                 PUSHL    BLKDSC                                        ;
                       00000000V  EF    05    FB 00019                 CALLS    #5, NML$MATCHSRC                              ;
                                   20    50    E9 00020                BLBC     R0, 4$                                        ;
                                  0C    AE    D4 00023                 CLRL     GBLEVT                                        ; 0828
                                  0C    AE    9F 00026  1$:            PUSHAB   GBLEVT                                        ; 0829
                                  04    AE    DD 00029                 PUSHL    GBLSRC                                        ;
                                   69    02    FB 0002C                CALLS    #2, NML$GETNXTEVT                             ;
                                   13    50    E9 0002F                BLBC     R0, 5$                                        ;
                54        0C    AE    0C    C1 00032                   ADDL3    #12, GBLEVT, GBLMSK                           ; 0832
                                         50    D4 00037                CLRL     I                                             ; 0837
                00        64    50    E5 00039  2$:                    BBCC     I, (GBLMSK), 3$                               ;
                F8        50    3F    F3 0003D  3$:                    AOBLEQ   #63, I, 2$                                    ; 0834
                                         E3    11 00041                BRB      1$                                            ; 0829
                                         6E    D4 00043  4$:           CLRL     GBLSRC                                        ; 0843
                                  04    AE    D4 00045  5$:            CLRL     SRCPTR                                        ; 0847
                                  04    AE    9F 00048  6$:            PUSHAB   SRCPTR                                        ; 0848
                                  04    AC    DD 0004B                 PUSHL    SNK                                           ;
                                  08    AC    DD 0004E                 PUSHL    BLKDSC                                        ;
                       00000000V  EF    03    FB 00051                 CALLS    #3, NML$GETNXTSNK                             ;
                                   01    50    E8 00058                BLBS     R0, 7$                                        ;
                                               04 0005B                RET                                                    ;
                            53    04    AE    D0 0005C  7$:            MOVL     SRCPTR, R3                                    ; 0850
                      FF    8F    03    A3    91 00060                 CMPB     3(R3), #-1                                    ;
                                         E1    13 00065                BEQL     6$                                            ;
                                  08    AE    D4 00067                 CLRL     EVTPTR                                        ; 0857
                                  08    AE    9F 0006A  8$:            PUSHAB   EVTPTR                                        ; 0858
                                         53    DD 0006D                PUSHL    R3                                            ;
                                   69    02    FB 0006F                CALLS    #2, NML$GETNXTEVT                             ;
                                   D3    50    E9 00072                BLBC     R0, 6$                                        ;
                56        08    AE    04    C1 00075                   ADDL3    #4, EVTPTR, LOGMSK                            ; 0861
                55        08    AE    0C    C1 0007A                   ADDL3    #12, EVTPTR, FILMSK                           ; 0862
                                         52    D4 0007F                CLRL     R2                                            ; 0864
                                         6E    D5 00081                TSTL     GBLSRC                                        ;
                                         18  13 00083           BEQL    9$                                  ;
                                         52  D6 00085           INCL    R2                                  ;
                                    0C   AE  9F 00087           PUSHAB  GBLEVT                              ; 0866
                           7E       0C   BE  3C 0008A           MOVZWL  @EVTPTR, -(SP)                      ; 0867
                                    08   AE  DD 0008E           PUSHL   GBLSRC                              ; 0866
                 00000000V EF            03  FB 00091           CALLS   #3, NML$MATCHEVT                    ;
                           57            50  D0 00098           MOVL    R0, STATUS                          ;
                                         02  11 0009B           BRB     10$                                 ;
                                         57  D4 0009D 9$:       CLRL    STATUS                              ; 0870
                           37            57  E9 0009F 10$:      BLBC    STATUS, 12$                         ; 0872
                           34            52  E9 000A2           BLBC    R2, 12$                             ; 0873
             54         0C AE            04  C1 000A5           ADDL3   #4, GBLEVT, GBLMSK                  ; 0877
                                         50  D4 000AA           CLRL    I                                   ; 0882
     52      66            01            50  EF 000AC 11$:      EXTZV   I, #1, (LOGMSK), R2                 ;
     51      64            01            50  EF 000B1           EXTZV   I, #1, (GBLMSK), R1                 ;
                           52            51  CA 000B6           BICL2   R1, R2                              ;
     66      01            50            52  F0 000B9           INSV    R2, I, #1, (LOGMSK)                 ;
     52      65            01            50  EF 000BE           EXTZV   I, #1, (FILMSK), R2                 ; 0883
     51      64            01            50  EF 000C3           EXTZV   I, #1, (GBLMSK), R1                 ;
                           58            52  D2 000C8           MCOML   R2, R8                              ;
                           51            58  CA 000CB           BICL2   R8, R1                              ;
     65      01            50            51  F0 000CE           INSV    R1, I, #1, (FILMSK)                 ;
             D5            50            3F  F3 000D3           AOBLEQ  #63, I, 11$                         ; 0879
                                         91  11 000D7           BRB     8$                                  ; 0872
                                         50  D4 000D9 12$:      CLRL    I                                   ; 0889
             00            65            50  E5 000DB 13$:      BBCC    I, (FILMSK), 14$                    ; 0892
             F8            50            3F  F3 000DF 14$:      AOBLEQ  #63, I, 13$                         ; 0889
                                         85  11 000E3           BRB     8$                                  ; 0858
                                             04 000E5           RET                                         ; 0900

; Routine Size:  230 bytes,     Routine Base:  $CODE$ + 0355
```

```
1 %SBTTL 'NML$CLEANSRC  Clean sources'
1 GLOBAL ROUTINE NML$CLEANSRC (BUFDSC, SNK, BLKDSC) : NOVALUE =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine goes through all source blocks for the specified
1 !       sink type and removes all event blocks that have no filters set.
1 !       Source blocks with event blocks are also removed.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       BUFDSC          Descriptor of buffer containing source blocks.
1 !       SNK             Logging sink type code.
1 !       BLKDSC          Descriptor of all source block data in buffer.
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !
1 !--
1
2    BEGIN
2
2    LOCAL
2       EVTPTR : REF BBLOCK,            ! Pointer to event block
2       FILMSK : REF BITVECTOR,
2       LOGMSK : REF BITVECTOR,
2       OLDEVT : REF BBLOCK,            ! Pointer to previous event block
2       OLDSRC : REF BBLOCK,            ! Pointer to previous source block
2       SRCPTR : REF BBLOCK,            ! Pointer to current source block
2       STATUS;
2
2    OLDSRC = 0;
2    SRCPTR = 0;
2    WHILE NML$GETNXTSNK (.BLKDSC, .SNK, SRCPTR) DO
3       BEGIN
3
3       CH$MOVE (.SRCPTR [SRC$W_LENGTH], .SRCPTR, NML$T_SRCBUFFER);
3
3       OLDEVT = 0;
3       EVTPTR = 0;
3       WHILE NML$GETNXTEVT (NML$T_SRCBUFFER, EVTPTR) DO
4          BEGIN
```

```
;  968        0958 4
;  969        0959 4              LOGMSK = EVTPTR [EVT$Q_LOGMSK];
;  970        0960 4              FILMSK = EVTPTR [EVT$Q_FILTERMSK];
;  971        0961 4
;  972        0962 4              STATUS = FALSE;
;  973        0963 4              INCR I FROM 0 TO (EVT$S_LOGMSK * 8) - 1 DO
;  974        0964 5                  BEGIN
;  975        0965 5
;  976        0966 5                  IF .LOGMSK [.I] OR .FILMSK [.I]
;  977        0967 5                  THEN
;  978        0968 6                      BEGIN
;  979        0969 6                      STATUS = TRUE;
;  980        0970 6                      EXITLOOP;
;  981        0971 5                      END;
;  982        0972 4                  END;
;  983        0973 4
;  984        0974 4              IF NOT .STATUS
;  985        0975 4              THEN
;  986        0976 5                  BEGIN
;  987        0977 5                  NML$REMEVT (NML$T_SRCBUFFER, .EVTPTR);
;  988        0978 5                  EVTPTR = .OLDEVT;         ! Back up event pointer
;  989        0979 5                  END
;  990        0980 4              ELSE
;  991        0981 4                  OLDEVT = .EVTPTR;
;  992        0982 4
;  993        0983 3              END;
;  994        0984 3
;  995        0985 3          IF .NML$T_SRCBUFFER [SRC$W_MSKCOUNT] NEQU 0
;  996        0986 3          THEN
;  997        0987 4              BEGIN
;  998        0988 4              NML$REPSRC (.BUFDSC, .BLKDSC, .SRCPTR, NML$T_SRCBUFFER);
;  999        0989 4              OLDSRC = .SRCPTR;
; 1000        0990 4              END
; 1001        0991 3          ELSE
; 1002        0992 4              BEGIN
; 1003        0993 4              NML$REMSRC (.BLKDSC, .SRCPTR);
; 1004        0994 4              SRCPTR = .OLDSRC;             ! Back up the source pointer
; 1005        0995 3              END;
; 1006        0996 3
; 1007        0997 2          END;
; 1008        0998 2
; 1009        0999 1      END;                                  ! End of NML$CLEANSRC



                                   0FFC 00000          .ENTRY   NML$CLEANSRC, Save R2,R3,R4,R5,R6,R7,R8,R9,-; 0902
                                                                R10,R11                                    ;
                  5E           04  C2 00002            SUBL2    #4, SP                                     ;
                               58  D4 00005            CLRL     OLDSRC                                     ; 0947
                               7E  D4 00007            CLRL     SRCPTR                                     ; 0948
                               5E  DD 00009 1$:        PUSHL    SP                                         ; 0949
                       08      AC  DD 0000B            PUSHL    SNK                                        ;
                       0C      AC  DD 0000E            PUSHL    BLKDSC                                     ;
      00000000V   EF           03  FB 00011            CALLS    #3, NML$GETNXTSNK                          ;
                  01           50  E8 00018            BLBS     R0, 2$                                     ;
```

```
                                             04 0001B           RET                                                       ;
                        56             6E    D0 0001C 2$:       MOVL    SRCPTR, R6                                        ; 0952
   00000000'  EF        66             66    28 0001F           MOVC3   (R6), (R6), NML$T_SRCBUFFER                       ;
                                       59    D4 00027           CLRL    OLDEVT                                            ; 0954
                                 04    AE    D4 00029           CLRL    EVTPTR                                            ; 0955
                                 04    AE    9F 0002C 3$:       PUSHAB  EVTPTR                                            ; 0956
                          00000000'    EF    9F 0002F           PUSHAB  NML$T_SRCBUFFER                                   ;
              00000000V   EF           02    FB 00035           CALLS   #2, NML$GETNXTEVT                                 ;
                          3E           50    E9 0003C           BLBC    R0, 9$                                            ;
       5A          04     AE           04    C1 0003F           ADDL3   #4, EVTPTR, LOGMSK                                ; 0959
       5B          04     AE           0C    C1 00044           ADDL3   #12, EVTPTR, FILMSK                               ; 0960
                                       57    D4 00049           CLRL    STATUS                                            ; 0962
                                       50    D4 0004B           CLRL    I                                                 ; 0966
       04                 6A           50    E0 0004D 4$:       BBS     I, (LOGMSK), 5$                                   ;
       05                 6B           50    E1 00051           BBC     I, (FILMSK), 6$                                   ;
                          57           01    D0 00055 5$:       MOVL    #1, STATUS                                        ; 0969
                                       04    11 00058           BRB     7$                                                ; 0968
       EF                 50           3F    F3 0005A 6$:       AOBLEQ  #63, I, 4$                                        ; 0963
                          16           57    E8 0005E 7$:       BLBS    STATUS, 8$                                        ; 0974
                                 04    AE    DD 00061           PUSHL   EVTPTR                                            ; 0977
                          00000000'    EF    9F 00064           PUSHAB  NML$T_SRCBUFFER                                   ;
              00000000V   EF           02    FB 0006A           CALLS   #2, NML$REMEVT                                    ;
                   04     AE           59    D0 00071           MOVL    OLDEVT, EVTPTR                                    ; 0978
                                       B5    11 00075           BRB     3$                                                ; 0974
                          59     04    AE    D0 00077 8$:       MOVL    EVTPTR, OLDEVT                                    ; 0981
                                       AF    11 0007B           BRB     3$                                                ; 0956
                          00000000'    EF    B5 0007D 9$:       TSTW    NML$T_SRCBUFFER+22                                ; 0985
                                       1A    13 00083           BEQL    10$                                               ;
                          00000000'    EF    9F 00085           PUSHAB  NML$T_SRCBUFFER                                   ; 0988
                                       56    DD 0008B           PUSHL   R6                                                ;
                                 0C    AC    DD 0008D           PUSHL   BLKDSC                                            ;
                                 04    AC    DD 00090           PUSHL   BUFDSC                                            ;
              00000000V   EF           04    FB 00093           CALLS   #4, NML$REPSRC                                    ;
                          58           56    D0 0009A           MOVL    R6, OLDSRC                                        ; 0989
                                       0F    11 0009D           BRB     11$                                               ; 0985
                                       56    DD 0009F 10$:      PUSHL   R6                                                ; 0993
                                 0C    AC    DD 000A1           PUSHL   BLKDSC                                            ;
              00000000V   EF           02    FB 000A4           CALLS   #2, NML$REMSRC                                    ;
                          6E           58    D0 000AB           MOVL    OLDSRC, SRCPTR                                    ; 0994
                                   FF58    31 000AE 11$:      BRW     1$                                                ; 0949
                                             04 000B1           RET                                                       ; 0999

; Routine Size:  178 bytes,    Routine Base:  $CODE$ + 043B
```

```
1 %SBTTL 'NML$MATCHSRC  Match specific source'
1 GLOBAL ROUTINE NML$MATCHSRC (BLKDSC, SNK, SRC, ENTDSC, SRCPTR) =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine searches the sink node buffer for a source block
1 !       that matches the specified event source.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       BLKDSC          Descriptor of source block buffer.
1 !       SNK             Logging sink type code.
1 !       SRC             Event source type code.
1 !       ENTDSC          Event source id string descriptor.
1 !       SRCPTR          Pointer to longword in which to return address
1 !                       of source block.
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       TRUE is returned if a match is found, FALSE is returned if no match.
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !
1 !--
1
2    BEGIN
2
2    MAP
2       SRC     : BYTE,
2       ENTDSC  : REF DESCRIPTOR;
2
2    LOCAL
2       PTR     : REF BBLOCK,            ! Temporary source block pointer
2       STATUS,                          ! Routine status
2       TSTLEN,                          ! Length of source to compare
2       TSTPTR;                          ! Address of source to compare
2
2    PTR = 0;                            ! Initialize source pointer
2    STATUS = FALSE;                     ! Initialize routine status
2
2    WHILE NML$GETNXTSNK (.BLKDSC, .SNK, PTR) DO
3       BEGIN
3       IF .PTR [SRC$B_SRCTYPE] EQLU .SRC
3       THEN
4           BEGIN
```

```
4  !
4  ! Select the length and address of the source to compare.
4  !
4               SELECTONEU .SRC OF
4                   SET
4
4                   [NMA$C_ENT_NOD]:            ! Node
5                       BEGIN
5
5                       IF .(ENTDSC [DSC$A_POINTER])<0,16> EQLU
5                           .PTR [SRC$W_NODADR]
5                       THEN
5                           STATUS = TRUE;
5
4                       END;
4
4                   [NMA$C_ENT_CIR,
4                    NMA$C_ENT_LIN,
4                    NMA$C_ENT_MOD]:            ! Circuit or Line or Module
5                       BEGIN
5
5                       IF CH$EQL (.ENTDSC [DSC$W_LENGTH],
5                                  .ENTDSC [DSC$A_POINTER],
5                                  .PTR [SRC$B_IDLENGTH],
5                                  PTR [SRC$T_ID])
5                       THEN
5                           STATUS = TRUE;
5
4                       END;
4
4                   [OTHERWISE]:                ! Null
5                       BEGIN
5
5                       STATUS = TRUE;
5
4                       END;
4                   TES;
4
4               IF .STATUS
4               THEN
5                   BEGIN
5
5                   .SRCPTR = .PTR;
5                   EXITLOOP;
5
4                   END;
3               END;
2           END;
2
2       RETURN .STATUS
2
1       END;                                    ! End of NML$MATCHSRC
```

```
                                                                   B 11
NML$LOGOPS        NML Logging data base operations module        16-Sep-1984 00:19:25     VAX-11 Bliss-32 V4.0-742              Page  36
V04-000           NML$MATCHSRC  Match specific source            14-Sep-1984 12:50:11     [NML.SRC]NMLLOGOPS.B32;1                  (12)

                                        007C 00000          .ENTRY  NML$MATCHSRC, Save R2,R3,R4,R5,R6                          ; 1001
                                     7E   D4 00002          CLRL    PTR                                                        ; 1049
                                     56   D4 00004          CLRL    STATUS                                                     ; 1050
                    55         0C    AC   9A 00006          MOVZBL  SRC, R5                                                    ; 1054
                                     5E   DD 0000A 1$:      PUSHL   SP                                                         ; 1052
                    7E         04    AC   7D 0000C          MOVQ    BLKDSC, -(SP)                                              ;
           00000000V           EF    03   FB 00010          CALLS   #3, NML$GETNXTSNK                                          ;
                               45    50   E9 00017          BLBC    R0, 7$                                                     ;
                               54    6E   D0 0001A          MOVL    PTR, R4                                                    ; 1054
   55      03    A4            08    00   ED 0001D          CMPZV   #0, #8, 3(R4), R5                                          ;
                               E5    12   00023             BNEQ    1$                                                         ;
                               55    D5   00025             TSTL    R5                                                         ; 1063
                               0B    12   00027             BNEQ    2$                                                         ;
                    50         10    AC   D0 00029          MOVL    ENTDSC, R0                                                 ; 1066
              04    A4         04    A0   B1 0002D          CMPW    4(R0), 4(R4)                                               ; 1067
                               1F    11   00032             BRB     4$                                                         ;
                    01               55   91 00034 2$:      CMPB    R5, #1                                                     ; 1073
                               0A    13   00037             BEQL    3$                                                         ;
                    03               55   91 00039          CMPB    R5, #3                                                     ;
                               17    1F   0003C             BLSSU   5$                                                         ;
                    04               55   91 0003E          CMPB    R5, #4                                                     ;
                               12    1A   00041             BGTRU   5$                                                         ;
                    50         10    AC   D0 00043 3$:      MOVL    ENTDSC, R0                                                 ; 1078
                    51         04    A4   9A 00047          MOVZBL  4(R4), R1                                                  ; 1080
   51              00    04    B0    60   2D 0004B          CMPC5   (R0), @4(R0), #0, R1, 5(R4)                                ; 1081
                               05    A4      00051                                                                             ;
                               03    12   00053 4$:         BNEQ    6$                                                         ;
                    56               01   D0 00055 5$:      MOVL    #1, STATUS                                                 ; 1090
                    AF               56   E9 00058 6$:      BLBC    STATUS, 1$                                                 ; 1095
              14    BC               54   D0 0005B          MOVL    R4, @SRCPTR                                                ; 1099
                    50               56   D0 0005F 7$:      MOVL    STATUS, R0                                                 ; 1106
                                          04 00062          RET                                                                ; 1108

; Routine Size:  99 bytes,    Routine Base:  $CODE$ + 04ED
```

```
1 %SBTTL 'NML$GETNXTSNK  Get next source block for specified sink'
1 GLOBAL ROUTINE NML$GETNXTSNK (BLKDSC, SNK, SRCPTR) =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine searches the sink node buffer for the next source block
1 !       that matches the specified sink type.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       BLKDSC          Descriptor of event source block buffer.
1 !       SNK             Logging sink type code to match.
1 !       SRCPTR          Address of longword in which to return address
1 !                       of source block.  If within range of buffer
1 !                       it will be used as the starting point from which
1 !                       to get the next source block that matches the
1 !                       specified sink.
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       TRUE is returned if a match is found, FALSE is returned if no match.
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !
1 !--
1
2     BEGIN
2
2     LOCAL
2         PTR : REF BBLOCK,               ! Temporary source block pointer
2         STATUS;                         ! Routine status
2
2     STATUS = FALSE;                     ! Initialize routine status
2     PTR = ..SRCPTR;                     ! Initialize source pointer
2
2     WHILE NML$GETNXTSRC (.BLKDSC, PTR) DO
3         BEGIN
3         IF .PTR [SRC$B_SINKTYPE] EQLU .SNK
3         THEN
4             BEGIN
4             .SRCPTR = .PTR;             ! Set source pointer for return
4             STATUS = TRUE;
4             EXITLOOP
3             END;
2         END;
```

```
; 1178          1166  2
; 1179          1167  2          RETURN .STATUS
; 1180          1168  2
; 1181          1169  1          END;                                          ! End of NML$GETNXTSNK



                                                         0004 00000             .ENTRY  NML$GETNXTSNK, Save R2                          ; 1110
                                                       52  D4 00002             CLRL    STATUS                                          ; 1153
                                              0C       BC  DD 00004             PUSHL   @SRCPTR                                         ; 1154
                                                       5E  DD 00007  1$:        PUSHL   SP                                              ; 1156
                                              04       AC  DD 00009             PUSHL   BLKDSC                                          ;
                            00000000V     EF       02  FB 0000C             CALLS   #2, NML$GETNXTSRC                               ;
                                          13       50  E9 00013             BLBC    R0, 2$                                          ;
                                          50       6E  D0 00016             MOVL    PTR, R0                                         ; 1158
     08    AC        02    A0             08       00  ED 00019             CMPZV   #0, #8, 2(R0), SNK                              ;
                                                   E5  12 00020             BNEQ    1$                                              ;
                                 0C       BC       50  D0 00022             MOVL    R0, @SRCPTR                                     ; 1161
                                          52       01  D0 00026             MOVL    #1, STATUS                                      ; 1162
                                          50       52  D0 00029  2$:        MOVL    STATUS, R0                                      ; 1167
                                                       04 0002C             RET                                                     ; 1169

; Routine Size:  45 bytes,    Routine Base:  $CODE$ + 0550
```

```
1 %SBTTL 'NML$GETNXTSRC  Get next source block'
1 GLOBAL ROUTINE NML$GETNXTSRC (BLKDSC, SRCPTR) =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !         This routine searches the sink node buffer for the next source
1 !         block.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !         BLKDSC          Descriptor of source block buffer.
1 !         SRCPTR          Address of longword in which to return the address
1 !                         of the next source block.  If value is within buffer
1 !                         range on input then it is used as the address of the
1 !                         starting source block.
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !         NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !         NONE
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !         TRUE is returned if a match is found, FALSE is returned if no match.
1 !
1 ! SIDE EFFECTS:
1 !
1 !         NONE
1 !
1 !--
1
2     BEGIN
2
2     MAP
2         BLKDSC : REF DESCRIPTOR;
2
2     LOCAL
2         BUFEND,                           ! Pointer to end of buffer
2         PTR     : REF BBLOCK,             ! Temporary source block pointer
2         STATUS;                           ! Routine status
2 !
2 ! If descriptor indicates no source blocks (length=0) then
2 ! return failure.
2 !
2     IF .BLKDSC [DSC$W_LENGTH] EQLU 0
2     THEN
2         RETURN FALSE;
2
2     BUFEND = .BLKDSC [DSC$A_POINTER] + .BLKDSC [DSC$W_LENGTH];
2     PTR = ..SRCPTR;                       ! Initialize source pointer
2 !
```

```
; 1240          1227  2 ! If PTR contains a value on input that is within the buffer range then
; 1241          1228  2 ! use it as the starting point.  If the value is not valid then return
; 1242          1229  2 ! the address of the first source block in the buffer.
; 1243          1230  2 !
; 1244          1231  3     IF  (.PTR LSSA .BLKDSC [DSC$A_POINTER])
; 1245          1232  2         OR
; 1246          1233  3         (.PTR GEQA .BUFEND)
; 1247          1234  2     THEN
; 1248          1235  2         PTR = .BLKDSC [DSC$A_POINTER]
; 1249          1236  2     ELSE
; 1250          1237  2         PTR = .PTR + .PTR [SRC$W_LENGTH];
; 1251          1238  2 !
; 1252          1239  2 ! If pointer is still within range of buffer then return TRUE else
; 1253          1240  2 ! return FALSE to indicate no more source blocks.
; 1254          1241  2 !
; 1255          1242  2     IF .PTR GEQA .BUFEND
; 1256          1243  2     THEN
; 1257          1244  2         STATUS = FALSE
; 1258          1245  2     ELSE
; 1259          1246  3         BEGIN
; 1260          1247  3         .SRCPTR = .PTR;                              ! Set source pointer for return
; 1261          1248  3         STATUS = TRUE;
; 1262          1249  2         END;
; 1263          1250  2
; 1264          1251  2     RETURN .STATUS
; 1265          1252  2
; 1266          1253  1     END;                                             ! End of NML$GETNXTSRC



                                       0004 00000        .ENTRY  NML$GETNXTSRC, Save R2                             ; 1171
                    51        04   AC  D0 00002          MOVL    BLKDSC, R1                                         ; 1220
                                   61  B5 00006          TSTW    (R1)                                               ;
                                   2F  13 00008          BEQL    4$                                                 ;
                    52             61  3C 0000A          MOVZWL  (R1), BUFEND                                       ; 1224
                    52        04   A1  C0 0000D          ADDL2   4(R1), BUFEND                                      ;
                    50        08   BC  D0 00011          MOVL    @SRCPTR, PTR                                       ; 1225
             04     A1             50  D1 00015          CMPL    PTR, 4(R1)                                         ; 1231
                                   05  1F 00019          BLSSU   1$                                                 ;
                    52             50  D1 0001B          CMPL    PTR, BUFEND                                        ; 1233
                                   06  1F 0001E          BLSSU   2$                                                 ;
                    50        04   A1  D0 00020 1$:      MOVL    4(R1), PTR                                         ; 1235
                                   06  11 00024          BRB     3$                                                 ;
                    51             60  3C 00026 2$:      MOVZWL  (PTR), R1                                          ; 1237
                    50             51  C0 00029          ADDL2   R1, PTR                                            ;
                    52             50  D1 0002C 3$:      CMPL    PTR, BUFEND                                        ; 1242
                                   08  1E 0002F          BGEQU   4$                                                 ;
             08     BC             50  D0 00031          MOVL    PTR, @SRCPTR                                       ; 1247
                    50             01  D0 00035          MOVL    #1, STATUS                                         ; 1248
                                       04 00038          RET                                                        ; 1251
                                   50  D4 00039 4$:      CLRL    R0                                                 ; 1253
                                       04 0003B          RET                                                        ;

; Routine Size:  60 bytes,    Routine Base:  $CODE$ + 057D
```

```
%SBTTL 'NML$MATCHEVT  Get event block matching specified class'
GLOBAL ROUTINE NML$MATCHEVT (SRCPTR, CLASS, EVTPTR) =

!++
! FUNCTIONAL DESCRIPTION:
!
!       This routine searches the source block for an event block that
!       matches the specified class.
!
! FORMAL PARAMETERS:
!
!       SRCPTR          Pointer to source block.
!       CLASS           Class code to match.
!       EVTPTR          Address of longword in which the pointer to
!                       the matched event block will be returned.
!
! IMPLICIT INPUTS:
!
!       NONE
!
! IMPLICIT OUTPUTS:
!
!       NONE
!
! ROUTINE VALUE:
! COMPLETION CODES:
!
!       TRUE is returned if a match is found, FALSE is returned if no match.
!
! SIDE EFFECTS:
!
!       NONE
!
!--

    BEGIN

    MAP
        SRCPTR : REF BBLOCK;

    LOCAL
        PTR    : REF BBLOCK,                ! Temporary event block pointer
        STATUS;                             ! Routine status

    PTR = 0;                                ! Initialize source pointer
    STATUS = FALSE;                         ! Initialize routine status

    WHILE NML$GETNXTEVT (.SRCPTR, PTR) DO
        BEGIN
        IF .PTR [EVT$W_CLASS] EQLU .CLASS
        THEN
            BEGIN
            .EVTPTR = .PTR;                 ! Set event pointer for return
            STATUS = TRUE;
            EXITLOOP
            END;
        END;
```

```
; 1325          1311  2
; 1326          1312  2       RETURN .STATUS
; 1327          1313  2
; 1328          1314  1       END;                                          ! End of NML$MATCHEVT



                                                       0004 00000          .ENTRY  NML$MATCHEVT, Save R2                            ; 1255
                                                    7E  D4 00002           CLRL    PTR                                              ; 1298
                                                    52  D4 00004           CLRL    STATUS                                           ; 1299
                                                    5E  DD 00006 1$:       PUSHL   SP                                               ; 1301
                                               04   AC  DD 00008           PUSHL   SRCPTR                                           ;
                               00000000V  EF        02  FB 0000B           CALLS   #2, NML$GETNXTEVT                                ;
                                          10        50  E9 00012           BLBC    R0, 2$                                           ;
      08    AC        00    BE            10        00  ED 00015           CMPZV   #0, #16, @PTR, CLASS                             ; 1303
                                                    E8  12 0001C           BNEQ    1$                                               ;
                                     0C   BC        6E  D0 0001E           MOVL    PTR, @EVTPTR                                     ; 1306
                                          52        01  D0 00022           MOVL    #1, STATUS                                       ; 1307
                                          50        52  D0 00025 2$:       MOVL    STATUS, R0                                       ; 1312
                                                        04 00028           RET                                                      ; 1314

; Routine Size:  41 bytes,    Routine Base:  $CODE$ + 05B9
```

```
1 %SBTTL 'NML$GETNXTEVT  Get next event block'
1 GLOBAL ROUTINE NML$GETNXTEVT (SRCPTR, EVTPTR) =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine searches the source block for the next event block.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       SRCPTR          Pointer to source block.
1 !       EVTPTR          Address of longword to contain address of matched
1 !                       event block.  If the value is within the event block
1 !                       range then it is used as the starting event block
1 !                       address.
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       TRUE is returned if a match is found, FALSE is returned if no match.
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !
1 !--
1
2     BEGIN
2
2     MAP
2         SRCPTR : REF BBLOCK;
2
2     LOCAL
2         CLASSES,                              ! Number of event event blocks
2         MASKEND,                              ! Pointer to end of masks
2         MASKPTR,                              ! Pointer to masks
2         PTR     : REF BBLOCK,                 ! Temporary event block pointer
2         STATUS;                               ! Routine status
2
2     CLASSES = .SRCPTR [SRC$W_MSKCOUNT];
2 !
2 ! If no event masks are present (count=0) then
2 ! return failure.
2 !
2     IF .CLASSES EQLU 0
2     THEN
2         RETURN FALSE;
2
2     MASKPTR = .SRCPTR + SRC$K_LENGTH;
```

```
; 1387          1372  2       MASKEND = .MASKPTR + (.CLASSES * EVT$K_LENGTH);
; 1388          1373  2       PTR = ..EVTPTR;                               ! Initialize event pointer
; 1389          1374  2   !
; 1390          1375  2   ! If PTR contains a value on input that is within the buffer range then
; 1391          1376  2   ! use it as the starting point.  If the value is not valid then return
; 1392          1377  2   ! the address of the first event block in the buffer.
; 1393          1378  2   !
; 1394          1379  3       IF   (.PTR LSSA .MASKPTR)
; 1395          1380  2            OR
; 1396          1381  3            (.PTR GEQA .MASKEND)
; 1397          1382  2       THEN
; 1398          1383  2            PTR = .MASKPTR
; 1399          1384  2       ELSE
; 1400          1385  2            PTR = .PTR + EVT$K_LENGTH;
; 1401          1386  2   !
; 1402          1387  2   ! If pointer is still within range of buffer then return TRUE else
; 1403          1388  2   ! return FALSE to indicate no more event blocks.
; 1404          1389  2   !
; 1405          1390  2       IF .PTR GEQA .MASKEND
; 1406          1391  2       THEN
; 1407          1392  2            STATUS = FALSE
; 1408          1393  2       ELSE
; 1409          1394  3            BEGIN
; 1410          1395  3            .EVTPTR = .PTR;                          ! Set event pointer for return
; 1411          1396  3            STATUS = TRUE;
; 1412          1397  2            END;
; 1413          1398  2
; 1414          1399  2       RETURN .STATUS
; 1415          1400  2
; 1416          1401  1       END;                                          ! End of NML$GETNXTEVT



                                           0004 00000          .ENTRY  NML$GETNXTEVT, Save R2                          ; 1316
                        50        04  AC   D0 00002            MOVL    SRCPTR, R0                                      ; 1362
                        51        16  A0   3C 00006            MOVZWL  22(R0), CLASSES                                 ;
                                      2D   13 0000A            BEQL    4$                                              ; 1367
                        50            18   C0 0000C            ADDL2   #24, MASKPTR                                    ; 1371
                        51            14   C4 0000F            MULL2   #20, R1                                         ; 1372
                  52    51            50   C1 00012            ADDL3   MASKPTR, R1, MASKEND                            ;
                        51        08  BC   D0 00016            MOVL    @EVTPTR, PTR                                    ; 1373
                        50            51   D1 0001A            CMPL    PTR, MASKPTR                                    ; 1379
                                      05   1F 0001D            BLSSU   1$                                              ;
                        52            51   D1 0001F            CMPL    PTR, MASKEND                                    ; 1381
                                      05   1F 00022            BLSSU   2$                                              ;
                        51            50   D0 00024  1$:       MOVL    MASKPTR, PTR                                    ; 1383
                                      03   11 00027            BRB     3$                                              ;
                        51            14   C0 00029  2$:       ADDL2   #20, PTR                                        ; 1385
                        52            51   D1 0002C  3$:       CMPL    PTR, MASKEND                                    ; 1390
                                      08   1E 0002F            BGEQU   4$                                              ;
                   08   BC            51   D0 00031            MOVL    PTR, @EVTPTR                                    ; 1395
                        50            01   D0 00035            MOVL    #1, STATUS                                      ; 1396
                                           04 00038            RET                                                     ; 1399
                                      50   D4 00039  4$:       CLRL    R0                                              ; 1401
                                           04 0003B            RET                                                     ;
```

```
; Routine Size:  60 bytes,    Routine Base:  $CODE$ + 05E2
```

```
%SBTTL 'NML$BLDSRC  Build a source block'
GLOBAL ROUTINE NML$BLDSRC (BUFDSC, SNK, SRC, ENTDSC) : NOVALUE =

!++
! FUNCTIONAL DESCRIPTION:
!
!       This routine builds a source block.
!
! FORMAL PARAMETERS:
!
!       BUFDSC          Descriptor of buffer to hold new source block.
!                       (Assumed to be at least SRC$K_LENGTH bytes.)
!       SNK             Logging sink type code.
!       SRC             Event source type code.
!       ENTDSC          Event source id string descriptor.
!
! IMPLICIT INPUTS:
!
!       NONE
!
! IMPLICIT OUTPUTS:
!
!       NONE
!
! ROUTINE VALUE:
! COMPLETION CODES:
!
!       NONE
!
! SIDE EFFECTS:
!
!       NONE
!
!--

    BEGIN

    MAP
        BUFDSC : REF DESCRIPTOR,
        ENTDSC : REF DESCRIPTOR;

    LOCAL
        SRCPTR : REF BBLOCK;

    SRCPTR = .BUFDSC [DSC$A_POINTER];
    CH$FILL (0, SRC$K_LENGTH, .SRCPTR); ! Zero the event block

    SRCPTR [SRC$W_LENGTH] = SRC$K_LENGTH;
    SRCPTR [SRC$B_SINKTYPE] = .SNK;
    SRCPTR [SRC$B_SRCTYPE] = .SRC;

    SELECTONEU .SRC OF
        SET
        [NMA$C_ENT_NOD]:                ! Node

            CH$MOVE (2,
                     ENTDSC [DSC$A_POINTER],
; 1475          1459  2                         SRCPTR [SRC$W_NODADR]);
; 1476          1460  2
; 1477          1461  2             [NMA$C_ENT_CIR,
; 1478          1462  2              NMA$C_ENT_LIN,
; 1479          1463  2              NMA$C_ENT_MOD]:          ! Circuit or Line or Module
; 1480          1464  3                 BEGIN
; 1481          1465  3
; 1482          1466  3                 SRCPTR [SRC$B_IDLENGTH] = .ENTDSC [DSC$W_LENGTH];
; 1483          1467  3                 CH$MOVE (.ENTDSC [DSC$W_LENGTH],
; 1484          1468  3                          .ENTDSC [DSC$A_POINTER],
; 1485          1469  3                         SRCPTR [SRC$T_ID]);
; 1486          1470  3
; 1487          1471  2                 END;
; 1488          1472  2
; 1489          1473  2             TES;
; 1490          1474  2
; 1491          1475  1         END;                                        ! End of NML$BLDSRC



                                          007C 00000           .ENTRY  NML$BLDSRC, Save R2,R3,R4,R5,R6                    ; 1403
                        50        04   AC   D0 00002           MOVL    BUFDSC, R0                                         ; 1446
                        56        04   A0   D0 00006           MOVL    4(R0), SRCPTR
        18          00  6E             00   2C 0000A           MOVC5   #0, ($P), #0, #24, (SRCPTR)                        ; 1447
                                       66      0000F                                                                      ;
                        66             18   B0 00010           MOVW    #24, (SRCPTR)                                      ; 1449
                  02    A6        08   AC   90 00013           MOVB    SNK, 2(SRCPTR)                                     ; 1450
                        50        0C   AC   D0 00018           MOVL    SRC, R0                                            ; 1451
                  03    A6             50   90 0001C           MOVB    R0, 3(SRCPTR)
                                       50   D5 00020           TSTL    R0                                                 ; 1455
                                       0A   12 00022           BNEQ    1$
                        50        10   AC   D0 00024           MOVL    ENTDSC, R0                                         ; 1458
                  04    A6        04   A0   B0 00028           MOVW    4(R0), 4(SRCPTR)
                                            04 0002D           RET                                                        ;
                        01             50   D1 0002E  1$:      CMPL    R0, #1                                             ; 1461
                                       0A   13 00031           BEQL    2$
                        03             50   D1 00033           CMPL    R0, #3
                                       13   1F 00036           BLSSU   3$
                        04             50   D1 00038           CMPL    R0, #4
                                       0E   1A 0003B           BGTRU   3$
                        50        10   AC   D0 0003D  2$:      MOVL    ENTDSC, R0                                         ; 1466
                  04    A6             60   90 00041           MOVB    (R0), 4(SRCPTR)
            05    A6    04    B0       60   28 00045           MOVC3   (R0), @4(R0), 5(SRCPTR)                            ; 1469
                                            04 0004B  3$:      RET                                                        ; 1475

; Routine Size:  76 bytes,    Routine Base:  $CODE$ + 061E
```

```
1 %SBTTL 'NML$BLDEVT  Build an event class block'
1 GLOBAL ROUTINE NML$BLDEVT (FCT, CLASS, MSKLEN, MSKPTR, EVTPTR) : NOVALUE =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine builds an event class block.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       FCT             Mask operation code. (0=CLEAR, 1=SET)
1 !       CLASS           Event class code.
1 !       MSKLEN          Length in bytes of event mask.
1 !       MSKPTR          Address of event mask.
1 !       EVTPTR          Address of event block to be filled in.
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !
1 !--
1
2    BEGIN
2
2    MAP
2       EVTPTR : REF BBLOCK;
2
2    CH$FILL (0, EVT$K_LENGTH, .EVTPTR); ! Zero the event block
2
2    EVTPTR [EVT$W_CLASS] = .CLASS;        ! Fill in the class code
2 !
2 ! If function is SET (FCT=1) then move the mask into the log mask.
2 ! Otherwise (FCT=0), function is CLEAR so move the mask into the filter
2 ! mask.
2
2    IF .FCT
2    THEN
2       CH$MOVE (.MSKLEN, .MSKPTR, EVTPTR [EVT$Q_LOGMSK])
2    ELSE
2       CH$MOVE (.MSKLEN, .MSKPTR, EVTPTR [EVT$Q_FILTERMSK]);
2
1    END;                                   ! End of NML$BLDEVT
```

```
                                                  007C 00000          .ENTRY  NML$BLDEVT, Save R2,R3,R4,R5,R6                    ; 1477
                            56        14     AC   D0 00002            MOVL    EVTPTR, R6                                         ; 1516
      14           00       6E               00   2C 00006            MOVC5   #0, (SP), #0, #20, (R6)                            ;
                                             66      0000B                                                                       ;
                            66        08     AC   B0 0000C            MOVW    CLASS, (R6)                                        ; 1518
                            08        04     AC   E9 00010            BLBC    FCT, 1$                                            ; 1524
             04    A6       10        BC     0C   AC 28 00014         MOVC3   MSKLEN, @MSKPTR, 4(R6)                             ; 1526
                                                  04 0001B            RET                                                        ;
             0C    A6       10        BC     0C   AC 28 0001C 1$:     MOVC3   MSKLEN, @MSKPTR, 12(R6)                            ; 1528
                                                  04 00023            RET                                                        ; 1530

; Routine Size:  36 bytes,    Routine Base:  $CODE$ + 066A
```

```
1 %SBTTL 'NML$ADDSRC  Add a source block to buffer'
1 GLOBAL ROUTINE NML$ADDSRC (BUFDSC, SRCDSC, SRCPTR) =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine adds a source block to the specified buffer.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       BUFDSC          Descriptor of source block buffer.
1 !       SRCDSC          Descriptor of source block data in buffer.
1 !       SRCPTR          Pointer to source block to be added.
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       Returns TRUE if the source block was added.  Returns FALSE if
1 !       there was not enough room in the buffer.
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !
1 !--
1
2    BEGIN
2
2    MAP
2       BUFDSC : REF DESCRIPTOR,
2       SRCDSC : REF DESCRIPTOR,
2       SRCPTR : REF BBLOCK;
2
2 !
2 ! Make sure source block will fit in the buffer.
2 !
3    IF (.BUFDSC [DSC$W_LENGTH] - .SRCDSC [DSC$W_LENGTH])
2       LSS
2       .SRCPTR [SRC$W_LENGTH]
2    THEN
2       RETURN FALSE;
2 !
2 ! Block will fit so move it.
2 !
2    CH$MOVE (.SRCPTR [SRC$W_LENGTH],
2             .SRCPTR,
2             .SRCDSC [DSC$A_POINTER] + .SRCDSC [DSC$W_LENGTH]);
2 !
2 ! Calculate resulting buffer length.
```

```
; 1606          1588  2 !
; 1607          1589  2        SRCDSC [DSC$W_LENGTH] =
; 1608          1590  2            .SRCDSC [DSC$W_LENGTH] + .SRCPTR [SRC$W_LENGTH];
; 1609          1591  2
; 1610          1592  2        RETURN TRUE
; 1611          1593  2
; 1612          1594  1        END;                                  ! End of NML$ADDSRC



                                            007C 00000         .ENTRY  NML$ADDSRC, Save R2,R3,R4,R5,R6                    ; 1532
                        56         08   AC   D0 00002          MOVL    SRCDSC, R6                                           ; 1575
                        50         04   BC   3C 00006          MOVZWL  @BUFDSC, R0                                          ;
                        51              66   3C 0000A          MOVZWL  (R6), R1                                             ;
                        50              51   C2 0000D          SUBL2   R1, R0                                               ;
      50      0C   BC   10              00   ED 00010          CMPZV   #0, #16, @SRCPTR, R0                                 ; 1577
                                        15   14 00016          BGTR    1$                                                   ;
                        50              66   3C 00018          MOVZWL  (R6), R0                                             ; 1585
                        50         04   A6   C0 0001B          ADDL2   4(R6), R0                                            ;
           60      0C   BC         0C   BC   28 0001F          MOVC3   @SRCPTR, @SRCPTR, (R0)                               ;
                        66         0C   BC   A0 00025          ADDW2   @SRCPTR, (R6)                                        ; 1590
                        50              01   D0 00029          MOVL    #1, R0                                               ; 1592
                                             04 0002C          RET                                                          ;
                                        50   D4 0002D 1$:      CLRL    R0                                                   ; 1594
                                             04 0002F          RET                                                          ;

; Routine Size:  48 bytes,    Routine Base:  $CODE$ + 068E
```

```
1 %SBTTL 'NML$REPSRC  Replace a source block in buffer'
1 GLOBAL ROUTINE NML$REPSRC (BUFDSC, SRCDSC, OLDSRC, NEWSRC) =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !      This routine adds a source block to the specified buffer.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !      BUFDSC          Descriptor of source block buffer.
1 !      SRCDSC          Descriptor of source block data in buffer.
1 !      OLDSRC          Pointer to old source block in buffer.
1 !      NEWSRC          Pointer to source block to be added.
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !      NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !      NONE
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !      Returns TRUE if the source block was added.  Returns FALSE if
1 !      there was not enough room in the buffer.
1 !
1 ! SIDE EFFECTS:
1 !
1 !      NONE
1 !
1 !--
1
2     BEGIN
2
2     MAP
2         BUFDSC : REF DESCRIPTOR,
2         SRCDSC : REF DESCRIPTOR,
2         OLDSRC : REF BBLOCK,
2         NEWSRC : REF BBLOCK;
2
2     LOCAL
2         FREELEN,
2         MOVLEN;
2 !
2 ! Make sure source block will fit in the buffer.
2 !
2     FREELEN =   .BUFDSC [DSC$W_LENGTH] -
2                 .SRCDSC [DSC$W_LENGTH] +
2                 .OLDSRC [SRC$W_LENGTH];
2     IF .FREELEN LSS .NEWSRC [SRC$W_LENGTH]
2     THEN
2         RETURN FALSE;
2
2     FREELEN = .FREELEN - .NEWSRC [SRC$W_LENGTH];
```

```
; 1671          1652  2 !
; 1672          1653  2 ! Block will fit so move it.
; 1673          1654  2 !
; 1674          1655  2     MOVLEN = .SRCDSC [DSC$A_POINTER] + .SRCDSC [DSC$W_LENGTH];
; 1675          1656  2     MOVLEN = .MOVLEN - .OLDSRC;
; 1676          1657  2     MOVLEN = .MOVLEN - .OLDSRC [SRC$W_LENGTH];
; 1677          1658  2
; 1678          1659  2     CH$MOVE (.MOVLEN,
; 1679          1660  2              .OLDSRC + .OLDSRC [SRC$W_LENGTH],
; 1680          1661  2              .OLDSRC + .NEWSRC [SRC$W_LENGTH]);
; 1681          1662  2
; 1682          1663  2     CH$MOVE (.NEWSRC [SRC$W_LENGTH],
; 1683          1664  2              .NEWSRC,
; 1684          1665  2              .OLDSRC);
; 1685          1666  2 !
; 1686          1667  2 ! Calculate resulting buffer length.
; 1687          1668  2 !
; 1688          1669  2     SRCDSC [DSC$W_LENGTH] =
; 1689          1670  2         .BUFDSC [DSC$W_LENGTH] - .FREELEN;
; 1690          1671  2
; 1691          1672  2     RETURN TRUE
; 1692          1673  2
; 1693          1674  1     END;                                        ! End of NML$ADDSRC



                                            03FC 00000          .ENTRY  NML$REPSRC, Save R2,R3,R4,R5,R6,R7,R8,R9       ; 1596
                        58         08    AC  D0 00002          MOVL    SRCDSC, R8                                     ; 1645
                        50         04    BC  3C 00006          MOVZWL  @BUFDSC, R0                                    ;
                        51               68  3C 0000A          MOVZWL  (R8), R1                                       ;
                        50               51  C2 0000D          SUBL2   R1, R0                                         ;
                        56         0C    AC  D0 00010          MOVL    OLDSRC, R6                                     ; 1646
                        51               66  3C 00014          MOVZWL  (R6), R1                                       ;
           59           50               51  C1 00017          ADDL3   R1, R0, FREELEN                                ;
                        57         10    BC  3C 0001B          MOVZWL  @NEWSRC, R7                                    ; 1647
                        57               59  D1 0001F          CMPL    FREELEN, R7                                    ;
                                         24  19 00022          BLSS    1$                                             ;
                        59               57  C2 00024          SUBL2   R7, FREELEN                                    ; 1651
                        50               68  3C 00027          MOVZWL  (R8), MOVLEN                                   ; 1655
                        50         04    A8  C0 0002A          ADDL2   4(R8), MOVLEN                                  ;
                        50               56  C2 0002E          SUBL2   R6, MOVLEN                                     ; 1656
                        50               51  C2 00031          SUBL2   R1, MOVLEN                                     ; 1657
         6746         6146               50  28 00034          MOVC3   MOVLEN, (R1)[R6], (R7)[R6]                     ; 1661
           66       10  BC               57  28 0003A          MOVC3   R7, @NEWSRC, (R6)                              ; 1665
           68       04  BC               59  A3 0003F          SUBW3   FREELEN, @BUFDSC, (R8)                         ; 1670
                        50               01  D0 00044          MOVL    #1, R0                                         ; 1672
                                             04 00047          RET                                                    ;
                                         50  D4 00048 1$:      CLRL    R0                                             ; 1674
                                             04 0004A          RET                                                    ;

; Routine Size:  75 bytes,    Routine Base:  $CODE$ + 06BE
```

```
1 %SBTTL 'NML$REMSRC  Remove source block from buffer'
1 GLOBAL ROUTINE NML$REMSRC (BLKDSC, SRCPTR) : NOVALUE =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine the specified source block from the buffer.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       BLKDSC          Descriptor of source block buffer.
1 !       SRCPTR          Pointer to source block in buffer to be removed.
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !
1 !--
1
2     BEGIN
2
2     MAP
2         BLKDSC : REF DESCRIPTOR,
2         SRCPTR : REF BBLOCK;
2
2     LOCAL
2         BUFEND,
2         LEN,
2         PTR;
2 !
2 ! Set up length and pointers to remove source block.
2 !
2     LEN = .SRCPTR [SRC$W_LENGTH];
2     PTR = .SRCPTR + .LEN;
2     BUFEND = .BLKDSC [DSC$A_POINTER] + .BLKDSC [DSC$W_LENGTH];
2 !
2 ! Move the end of the buffer back over the source block to be removed.
2 !
2     CH$MOVE (.BUFEND - .PTR,
2              .PTR,
2              .SRCPTR);
2 !
2 ! Update the descriptor.
2 !
```

```
; 1752          1732  2       BLKDSC [DSC$W_LENGTH] =
; 1753          1733  2           .BLKDSC [DSC$W_LENGTH] - .LEN;
; 1754          1734  2
; 1755          1735  1       END;                                        ! End of NML$REMSRC



                                                        00FC 00000          .ENTRY  NML$REMSRC, Save R2,R3,R4,R5,R6,R7             ; 1676
                                   57        08   BC   3C 00002          MOVZWL  @SRCPTR, LEN                                    ; 1720
                   51              57        08   AC   C1 00006          ADDL3   SRCPTR, LEN, PTR                                ; 1721
                                   56        04   AC   D0 0000B          MOVL    BLKDSC, R6                                      ; 1722
                                   50             66   3C 0000F          MOVZWL  (R6), BUFEND                                    ;
                                   50        04   A6   C0 00012          ADDL2   4(R6), BUFEND                                   ;
                                   50             51   C2 00016          SUBL2   PTR, R0                                         ; 1726
              08   BC              61             50   28 00019          MOVC3   R0, (PTR), @SRCPTR                              ; 1728
                                   66             57   A2 0001E          SUBW2   LEN, (R6)                                       ; 1733
                                                       04 00021          RET                                                     ; 1735

; Routine Size:  34 bytes,    Routine Base:  $CODE$ + 0709
```

```
1 %SBTTL 'NML$ADDEVT  Add an event block to source buffer'
1 GLOBAL ROUTINE NML$ADDEVT (BUFDSC, SRCPTR, EVTPTR) =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine adds an event block to the specified source buffer.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       BUFDSC          Descriptor of buffer containing source block.
1 !       SRCPTR          Pointer to source block in buffer.
1 !       EVTPTR          Pointer to event block to be added.
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       Returns TRUE if the event block was added.  Returns FALSE if
1 !       there was not enough room in the buffer.
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !
1 !--
1
2     BEGIN
2
2     MAP
2       BUFDSC : REF DESCRIPTOR,
2       SRCPTR : REF BBLOCK,
2       EVTPTR : REF BBLOCK;
2 !
2 ! Make sure event block will fit in the buffer.
2 !
3     IF (.BUFDSC [DSC$W_LENGTH] - .SRCPTR [SRC$W_LENGTH])
2       LSS
2       EVT$K_LENGTH
2     THEN
2       RETURN FALSE;
2 !
2 ! Block will fit so move it.
2 !
2     CH$MOVE (EVT$K_LENGTH,
2             .EVTPTR,
2             .SRCPTR + .SRCPTR [SRC$W_LENGTH]);
2 !
2 ! Calculate resulting buffer length and store it in source block.
2 ! Also increment the mask count.
```

```
; 1814         1793  2 !
; 1815         1794  2      SRCPTR [SRC$W_LENGTH] =
; 1816         1795  2          .SRCPTR [SRC$W_LENGTH] + EVT$K_LENGTH;
; 1817         1796  2
; 1818         1797  2      SRCPTR [SRC$W_MSKCOUNT] =
; 1819         1798  2          .SRCPTR [SRC$W_MSKCOUNT] + 1;
; 1820         1799  2
; 1821         1800  2      RETURN TRUE
; 1822         1801  2
; 1823         1802  1      END;                                        ! End of NML$ADDEVT



                                                   007C 00000        .ENTRY  NML$ADDEVT, Save R2,R3,R4,R5,R6                    ; 1737
                              56        08    AC    D0 00002         MOVL    SRCPTR, R6                                         ; 1779
                              51              66    3C 00006         MOVZWL  (R6), R1                                           ;
                              50        14    A1    9E 00009         MOVAB   20(R1), R0                                         ; 1780
         50       04    BC    10              00    ED 0000D         CMPZV   #0, #16, @BUFDSC, R0                               ;
                                              10    19 00013         BLSS    1$                                                 ;
                 6146      0C    BC           14    28 00015         MOVC3   #20, @EVTPTR, (R1)[R6]                             ; 1789
                              66              14    A0 0001B         ADDW2   #20, (R6)                                          ; 1795
                                        16    A6    B6 0001E         INCW    22(R6)                                             ; 1798
                              50              01    D0 00021         MOVL    #1, R0                                             ; 1800
                                                    04 00024         RET                                                        ;
                                              50    D4 00025 1$:     CLRL    R0                                                 ; 1802
                                                    04 00027         RET                                                        ;

; Routine Size:  40 bytes,     Routine Base:  $CODE$ + 072B
```

```
1 %SBTTL 'NML$MODEVT  Modify event block'
1 GLOBAL ROUTINE NML$MODEVT (FCT, ZER, EVTPTR, MSKLEN, MSKPTR) : NOVALUE =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !        This routine the modifies the specified event block.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !        FCT              Mask operation code. (FALSE=CLEAR, TRUE=SET).
1 !        ZER              Zero flag.  (TRUE=yes, FALSE=no).
1 !        EVTPTR           Pointer to event block.
1 !        MSKLEN           Length of mask value to be added.
1 !        MSKPTR           Pointer to mask value to be added.
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !        NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !        NONE
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !        NONE
1 !
1 ! SIDE EFFECTS:
1 !
1 !        NONE
1 !
1 !--
1
2     BEGIN
2
2     MAP
2         EVTPTR : REF BBLOCK,
2         MSKPTR : REF BITVECTOR;
2
2     LOCAL
2         BITLEN,                          ! Length of mask in bits
2         OLDMSK : REF BITVECTOR,          ! Mask not changed
2         RESMSK : REF BITVECTOR;          ! Address of result mask
2 !
2 ! If the operation is SET (FCT=1) then modify log mask.
2 ! Otherwise, operation is CLEAR (FCT=0) so modify filter mask.
2 !
2     IF .FCT
2     THEN
3         BEGIN
3         RESMSK = EVTPTR [EVT$Q_LOGMSK];
3         OLDMSK = EVTPTR [EVT$Q_FILTERMSK];
3         END
2     ELSE
3         BEGIN
```

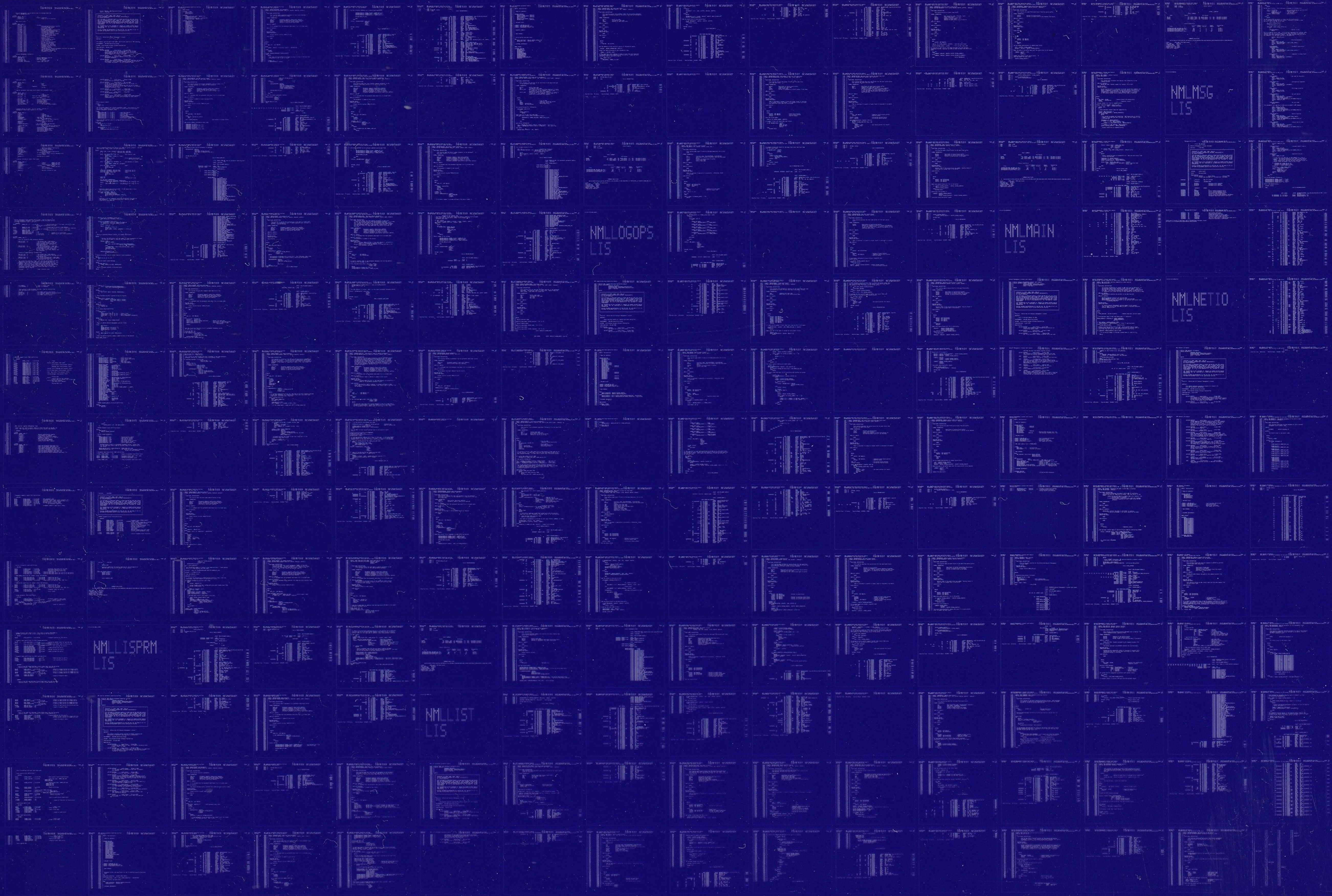
```
; 1882          1860  3              RESMSK = EVTPTR [EVT$Q_FILTERMSK];
; 1883          1861  3              OLDMSK = EVTPTR [EVT$Q_LOGMSK];
; 1884          1862  2              END;
; 1885          1863  2   !
; 1886          1864  2   ! Set the correct bits in the result mask.
; 1887          1865  2   !
; 1888          1866  2       BITLEN = .MSKLEN * 8;
; 1889          1867  2
; 1890          1868  2       INCR I FROM 0 TO .BITLEN - 1 DO
; 1891          1869  3              BEGIN
; 1892          1870  3
; 1893          1871  3              RESMSK [.I] = .RESMSK [.I] OR .MSKPTR [.I];
; 1894          1872  3              OLDMSK [.I] = .OLDMSK [.I] AND NOT .MSKPTR [.I];
; 1895          1873  3
; 1896          1874  2              END;
; 1897          1875  2   !
; 1898          1876  2   ! If the other mask should be zeroed (ZER=TRUE) then zero it.
; 1899          1877  2   !
; 1900          1878  2       IF .ZER
; 1901          1879  2       THEN
; 1902          1880  3              BEGIN
; 1903          1881  3
; 1904          1882  3              MAP OLDMSK : REF VECTOR [, BYTE];
; 1905          1883  3
; 1906          1884  3              INCR I FROM 0 TO EVT$S_LOGMSK - 1 DO
; 1907          1885  4                   BEGIN
; 1908          1886  4
; 1909          1887  4                   OLDMSK [.I] = 0;
; 1910          1888  4
; 1911          1889  3                   END;
; 1912          1890  2              END;
; 1913          1891  2
; 1914          1892  1       END;                                        ! End of NML$MODEVT



                                                   003C 00000          .ENTRY   NML$MODEVT, Save R2,R3,R4,R5                    ; 1804
                     51         0C    AC        04  C1 00002           ADDL3    #4, EVTPTR, R1                                  ; 1855
                     50         0C    AC        0C  C1 00007           ADDL3    #12, EVTPTR, R0                                 ; 1856
                                      05    04  AC  E9 0000C           BLBC     FCT, 1$                                         ; 1852
                                      53        50  7D 00010           MOVQ     R0, OLDMSK                                      ; 1856
                                                06  11 00013           BRB      2$                                              ; 1852
                                      54        50  D0 00015  1$:      MOVL     R0, RESMSK                                      ; 1860
                                      53        51  D0 00018           MOVL     R1, OLDMSK                                      ; 1861
                     55         10    AC        03  78 0001B  2$:      ASHL     #3, MSKLEN, BITLEN                              ; 1866
                                      51        01  CE 00020           MNEGL    #1, I                                           ; 1872
                                                26  11 00023           BRB      4$                                              ;
         52          64               01        51  EF 00025  3$:      EXTZV    I, #1, (RESMSK), R2                             ; 1871
         50     14   BC               01        51  EF 0002A           EXTZV    I, #1, @MSKPTR, R0                              ;
                                      50        52  C8 00030           BISL2    R2, R0                                          ;
         64          01               51        50  F0 00033           INSV     R0, I, #1, (RESMSK)                             ;
         52          63               01        51  EF 00038           EXTZV    I, #1, (OLDMSK), R2                             ; 1872
         50     14   BC               01        51  EF 0003D           EXTZV    I, #1, @MSKPTR, R0                              ;
                                      52        50  CA 00043           BICL2    R0, R2                                          ;
         63          01               51        52  F0 00046           INSV     R2, I, #1, (OLDMSK)                             ;
```

```
                    D6          51          55  F2 0004B 4$:    AOBLSS  BITLEN, I, 3$                           ; 1868
                                09    08    AC  E9 0004F        BLBC    ZER, 6$                                 ; 1878
                                            51  D4 00053        CLRL    I                                       ; 1884
                                          6143  94 00055 5$:    CLRB    (I)[OLDMSK]                             ; 1887
                    F9          51          07  F3 00058        AOBLEQ  #7, I, 5$                               ; 1884
                                                04 0005C 6$:    RET                                             ; 1892

; Routine Size:  93 bytes,    Routine Base:  $CODE$ + 0753
```

```
1 %SBTTL 'NML$REMEVT  Remove event block from source buffer'
1 GLOBAL ROUTINE NML$REMEVT (SRCPTR, EVTPTR) : NOVALUE =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine the specified event block from the source buffer.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       SRCPTR          Pointer to source block.
1 !       EVTPTR          Pointer to event block to be removed from source.
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !
1 !--
1
2    BEGIN
2
2    MAP
2       SRCPTR : REF BBLOCK,
2       EVTPTR : REF BBLOCK;
2
2    LOCAL
2       BUFEND,
2       PTR;
2 !
2 ! Set up length and pointers to remove event block.
2 !
2    PTR = .EVTPTR + EVT$K_LENGTH;
2    BUFEND = .SRCPTR + .SRCPTR [SRC$W_LENGTH];
2 !
2 ! Move the end of the buffer back over the event block to be removed.
2 !
2    CH$MOVE (.BUFEND - .PTR,
2             .PTR,
2             .EVTPTR);
2 !
2 ! Update the length of the source block.
2 ! Also decrement the mask count.
2 !
2    SRCPTR [SRC$W_LENGTH] =
; 1973          1950  2           .SRCPTR [SRC$W_LENGTH] - EVT$K_LENGTH;
; 1974          1951  2
; 1975          1952  2       SRCPTR [SRC$W_MSKCOUNT] =
; 1976          1953  2           .SRCPTR [SRC$W_MSKCOUNT] - 1;
; 1977          1954  2
; 1978          1955  1       END;                                  ! End of NML$REMEVT



                                          007C 00000          .ENTRY   NML$REMEVT, Save R2,R3,R4,R5,R6          ; 1894
              51       08   AC          14  C1 00002          ADDL3    #20, EVTPTR, PTR                         ; 1937
                            56      04  AC  D0 00007          MOVL     SRCPTR, R6                               ; 1938
                            50          66  3C 0000B          MOVZWL   (R6), BUFEND                             ;
                            50          56  C0 0000E          ADDL2    R6, BUFEND                               ;
                            50          51  C2 00011          SUBL2    PTR, R0                                  ; 1942
           08   BC          61          50  28 00014          MOVC3    R0, (PTR), @EVTPTR                       ; 1944
                            66          14  A2 00019          SUBW2    #20, (R6)                                ; 1950
                                16  A6  B7 0001C          DECW     22(R6)                                   ; 1953
                                          04 0001F          RET                                               ; 1955

; Routine Size:  32 bytes,    Routine Base:  $CODE$ + 07B0
; 1980          1956  1 END                                   ! End of module
; 1981          1957  1
; 1982          1958  0 ELUDOM



;                                       PSECT SUMMARY
;
;       Name                           Bytes                       Attributes
;
;   $OWN$                               1044  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;   $PLIT$                                40  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;   $CODE$                              2000  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)



;                               Library Statistics
;
;                                               -------- Symbols --------       Pages         Processing
;       File                                    Total    Loaded   Percent       Mapped        Time
;
;  _$255$DUA28:[NML.OBJ]NMLLIB.L32;1              341        40        11          27           00:00.1
;  _$255$DUA28:[SHRLIB]NMALIBRY.L32;1             887         5         0          47           00:00.2
;  _$255$DUA28:[SYSLIB]STARLET.L32;1             9776         2         0         581           00:02.1



;                                       COMMAND QUALIFIERS

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

; Size:           2000 code + 1084 data bytes
; Run Time:          00:40.1
; Elapsed Time:      01:39.0
; Lines/CPU Min:      2931
; Lexemes/CPU-Min: 12503
; Memory Used:  134 pages
; Compilation Complete
```

0284 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY


NMLMSG
LIS

NMLLOGOPS
LIS

NMLMAIN
LIS

NMLNETIO
LIS

NMLLISPRM
LIS

NMLLIST
LIS