```
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
```

FILEID**NMLFILEIO

```
NN      NN  MM      MM  LL          FFFFFFFFFF   IIIIII   LL          EEEEEEEEEE   IIIIII    000000
NN      NN  MM      MM  LL          FFFFFFFFFF   IIIIII   LL          EEEEEEEEEE   IIIIII    000000
NN      NN  MMMM  MMMM  LL          FF             II     LL          EE             II     00    00
NN      NN  MMMM  MMMM  LL          FF             II     LL          EE             II     00    00
NNNN    NN  MM  MM  MM  LL          FF             II     LL          EE             II     00    00
NNNN    NN  MM  MM  MM  LL          FF             II     LL          EE             II     00    00
NN NN   NN  MM      MM  LL          FFFFFFFF       II     LL          EEEEEEEE       II     00    00
NN NN   NN  MM      MM  LL          FFFFFFFF       II     LL          EEEEEEEE       II     00    00
NN   NNNN   MM      MM  LL          FF             II     LL          EE             II     00    00
NN   NNNN   MM      MM  LL          FF             II     LL          EE             II     00    00
NN    NN    MM      MM  LL          FF             II     LL          EE             II     00    00
NN    NN    MM      MM  LL          FF             II     LL          EE             II     00    00
NN    NN    MM      MM  LLLLLLLLLL  FF           IIIIII   LLLLLLLLLL  EEEEEEEEEE   IIIIII    000000
NN    NN    MM      MM  LLLLLLLLLL  FF           IIIIII   LLLLLLLLLL  EEEEEEEEEE   IIIIII    000000

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

```
;  1          0001  0 %TITLE 'NML File I/O modules'
;  2          0002  0 MODULE NML$FILEIO (
;  3          0003  0                 LANGUAGE (BLISS32),
;  4          0004  0                 ADDRESSING_MODE (NONEXTERNAL=GENERAL),
;  5          0005  0                 ADDRESSING_MODE (EXTERNAL=GENERAL),
;  6          0006  0                 IDENT = 'V04-000'
;  7          0007  C                 ) =
;  8          0008  1 BEGIN
;  9          0009  1 !
; 10          0010  1 !**************************************************************************
; 11          0011  1 !*                                                                        *
; 12          0012  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                               *
; 13          0013  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                 *
; 14          0014  1 !*  ALL RIGHTS RESERVED.                                                   *
; 15          0015  1 !*                                                                        *
; 16          0016  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
; 17          0017  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
; 18          0018  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
; 19          0019  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
; 20          0020  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
; 21          0021  1 !*  TRANSFERRED.                                                           *
; 22          0022  1 !*                                                                        *
; 23          0023  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE  *
; 24          0024  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT  *
; 25          0025  1 !*  CORPORATION.                                                           *
; 26          0026  1 !*                                                                        *
; 27          0027  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS  *
; 28          0028  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                *
; 29          0029  1 !*                                                                        *
; 30          0030  1 !*                                                                        *
; 31          0031  1 !**************************************************************************
; 32          0032  1 !
; 33          0033  1
; 34          0034  1 !++
; 35          0035  1 ! FACILITY:  DECnet-VAX V2.0 Network Management Listener
; 36          0036  1 !
; 37          0037  1 ! ABSTRACT:
; 38          0038  1 !
; 39          0039  1 !       This module contains routines to handle I/O for the permanent
; 40          0040  1 !       data base files.
; 41          0041  1 !
; 42          0042  1 ! ENVIRONMENT:  VAX/VMS Operating System
; 43          0043  1 !
; 44          0044  1 ! AUTHOR:  Distributed Systems Software Engineering
; 45          0045  1 !
; 46          0046  1 ! CREATION DATE:  30-DEC-1979
; 47          0047  1 !
; 48          0048  1 ! MODIFIED BY:
; 49          0049  1 !       V03-003 MKP0003         Kathy Perko     4-July-1983
; 50          0050  1 !               Convert node permanant database to four ISAM keys.
; 51          0051  1 !               This will make it much faster.
; 52          0052  1 !
; 53          0053  1 !       V03-002 MKP0002         Kathy Perko     29-June-1982
; 54          0054  1 !               Modify entity qualifier handling to use the qualifier's
; 55          0055  1 !               Parameter Semantic Table (PST) entry address instead of
; 56          0056  1 !               the Network Management parameter ID as input.
; 57          0057  1 !               Fix bug in NML$MATCHRECORD so it quits looking if there's
```

```
;    58          0058  1 !                    no qualifier.
;    59          0059  1 !
;    60          0060  1 !          V03-001 MKP0001          Kathy Perko     3-May-1982
;    61          0061  1 !                    Change NML$MATCHRECORD to handle entity qualifiers.
;    62          0062  1 !
;    63          0063  1 !--
;    64          0064  1
```

```
;    66          0065  1 %SBTTL 'Declarations'
;    67          0066  1
;    68          0067  1 !
;    69          0068  1 ! TABLE OF CONTENTS:
;    70          0069  1 !
;    71          0070  1
;    72          0071  1 FORWARD ROUTINE
;    73          0072  1     NML$OPENFILE,
;    74          0073  1     NML$CLOSEFILE,
;    75          0074  1     NML$READRECORD,
;    76          0075  1     NML$MATCHRECORD,
;    77          0076  1     NML$WRITERECORD,
;    78          0077  1     NML$DELETRECORD,
;    79          0078  1     NML$CHKFILEIO;
;    80          0079  1
;    81          0080  1 !
;    82          0081  1 ! INCLUDE FILES:
;    83          0082  1 !
;    84          0083  1
;    85          0084  1 LIBRARY 'LIB$:NMLLIB.L32';
;    86          0085  1 LIBRARY 'SHRLIB$:NMALIBRY.L32';
;    87          0086  1 LIBRARY 'SYS$LIBRARY:STARLET.L32';
;    88          0087  1
;    89          0088  1 !
;    90          0089  1 ! EXTERNAL REFERENCES:
;    91          0090  1 !
;    92          0091  1
;    93          0092  1 $NML_EXTDEF;
;    94          0093  1
;    95          0094  1 EXTERNAL LITERAL
;    96          0095  1     NML$_READERR,
;    97          0096  1     NML$_WRITERR,
;    98          0097  1     NML$_DELETERR,
;    99          0098  1     NML$_RECREPLC,
;   100          0099  1     NML$_RECADDED,
;   101          0100  1     NML$_RECDELET,
;   102          0101  1     NML$_NORECOWN;
;   103          0102  1
;   104          0103  1 EXTERNAL
;   105          0104  1     nml$gq_proprvmsk : BBLOCK [8];
;   106          0105  1
;   107          0106  1 EXTERNAL ROUTINE
;   108          0107  1     nma$closefile,
;   109          0108  1     nma$deleterec,
;   110          0109  1     nma$matchrec,
;   111          0110  1     nma$openfile,
;   112          0111  1     nma$readrec,
;   113          0112  1     nma$writerec,
;   114          0113  1     nma$searchfld,
;   115          0114  1     nml$bld_reply,
;   116          0115  1     nml$error_1,
;   117          0116  1     nml$close_node_file,
;   118          0117  1     nml$delete_node_rec,
;   119          0118  1     nml$open_node_file,
;   120          0119  1     nml$read_node_rec,
;   121          0120  1     nml$send,
;   122          0121  1     nml$write_node_rec;
```

; 123 0122 1

```
;  125     0123  1 %SBTTL 'NML$OPENFILE  Open permanent data base file'
;  126     0124  1 GLOBAL ROUTINE nml$openfile (fid, access) =
;  127     0125  1
;  128     0126  1 !++
;  129     0127  1 ! FUNCTIONAL DESCRIPTION:
;  130     0128  1 !
;  131     0129  1 !       This routine opens the permanent data base file(s) specified by
;  132     0130  1 !       the code in FID with the required access as specified by the
;  133     0131  1 !       code in ACCESS.
;  134     0132  1 !
;  135     0133  1 ! FORMAL PARAMETERS:
;  136     0134  1 !
;  137     0135  1 !       FID             Permanent data base file identification code
;  138     0136  1 !                       (NMA$C_OPN_xxxx).
;  139     0137  1 !       ACCESS          File access code (NMA$C_OPN_AC_RO=>read,
;  140     0138  1 !                       NMA$C_OPN_AC_RW=>read/write).
;  141     0139  1 !
;  142     0140  1 ! IMPLICIT INPUTS:
;  143     0141  1 !
;  144     0142  1 !       NONE
;  145     0143  1 !
;  146     0144  1 ! IMPLICIT OUTPUTS:
;  147     0145  1 !
;  148     0146  1 !       NONE
;  149     0147  1 !
;  150     0148  1 ! ROUTINE VALUE:
;  151     0149  1 ! COMPLETION CODES:
;  152     0150  1 !
;  153     0151  1 !       Returns a code indicating success.
;  154     0152  1 !
;  155     0153  1 ! SIDE EFFECTS:
;  156     0154  1 !
;  157     0155  1 !       Causes errors to be signaled.
;  158     0156  1 !
;  159     0157  1 !--
;  160     0158  1
;  161     0159  2 BEGIN
;  162     0160  2
;  163     0161  2 LOCAL
;  164     0162  2     status;
;  165     0163  2 !
;  166     0164  2 ! Require OPERATOR privilege to write to permanent database files.  Since
;  167     0165  2 ! the files get left open for the entire NCP session, if the caller has
;  168     0166  2 ! OPERATOR privilege, always open the files for read and write.  If the
;  169     0167  2 ! caller doesn't have OPER privilege, and is trying to open the file for
;  170     0168  2 ! write access, return a privilege violation.
;  171     0169  2 !
;  172     0170  2 IF .nml$gq_proprvmsk [prv$v_oper] THEN
;  173     0171  2     access = nma$c_opn_ac_rw
;  174     0172  2 ELSE
;  175     0173  2 IF .access EQLU nma$c_opn_ac_rw THEN
;  176     0174  2     nml$error_1 (nma$c_sts_pri);
;  177     0175  2 !
;  178     0176  2 ! Open the permanent data base file.  Since the node permanent database
;  179     0177  2 ! structure is quite different from the others, it is handled by separate
;  180     0178  2 ! routines.  It's different because it's so much larger, it must be faster.
;  181     0179  2 !
```

```
;  182     0180  2 IF .fid EQL nma$c_opn_node THEN
;  183     0181  2     status = nml$open_node_file ()
;  184     0182  2 ELSE
;  185     0183  2     status = nma$openfile (.fid,
;  186     0184  2                           .access);
;  187     0185  2 !
;  188     0186  2 ! Check the status and return it if it is success.  If an error
;  189     0187  2 ! has occurred then a file open error will be signalled.
;  190     0188  2 !
;  191     0189  2 RETURN nml$chkfileio (nma$c_sts_fop,
;  192     0190  2                       .status);
;  193     0191  2
;  194     0192  1 END;                            ! End of NML$OPENFILE
```

```
                                .TITLE  NML$FILEIO NML File I/O modules
                                .IDENT  \V04-000\

                                .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$GW_PRMDESCNT
                                .EXTRN  NML$_READERR, NML$_WRITERR
```

```
                                                        .EXTRN  NML$_DELETERR, NML$_RECREPLC
                                                        .EXTRN  NML$_RECADDED, NML$_RECDELET
                                                        .EXTRN  NML$_NORECOWN, NML$GQ_PROPRVMSK
                                                        .EXTRN  NMA$CLOSEFILE, NMA$DELETEREC
                                                        .EXTRN  NMA$MATCHREC, NMA$OPENFILE
                                                        .EXTRN  NMA$READREC, NMA$WRITEREC
                                                        .EXTRN  NMA$SEARCHFLD, NML$BLD_REPLY
                                                        .EXTRN  NML$ERROR_1, NML$CLOSE_NODE_FILE
                                                        .EXTRN  NML$DELETE_NODE_REC
                                                        .EXTRN  NML$OPEN_NODE_FILE
                                                        .EXTRN  NML$READ_NODE_REC
                                                        .EXTRN  NML$SEND, NML$WRITE_NODE_REC

                                                        .PSECT  $CODE$,NOWRT,2

                                    0000 00000          .ENTRY  NML$OPENFILE, Save nothing                  ; 0124
06 00000000G   00                02   E1 00002          BBC     #2, NML$GQ_PROPRVMSK+2, 1$                  ; 0170
         08    AC                01   D0 0000A          MOVL    #1, ACCESS                                  ; 0171
                                 10   11 0000E          BRB     2$                                          ;
               01          08    AC   D1 00010  1$:     CMPL    ACCESS, #1                                  ; 0173
                                 0A   12 00014          BNEQ    2$                                          ;
               7E                03   CE 00016          MNEGL   #3, -(SP)                                   ; 0174
   00000000G   00                01   FB 00019          CALLS   #1, NML$ERROR_1                             ;
                           04    AC   D5 00020  2$:     TSTL    FID                                         ; 0180
                                 09   12 00023          BNEQ    3$                                          ;
   00000000G   00                00   FB 00025          CALLS   #0, NML$OPEN_NODE_FILE                      ; 0181
                                 0B   11 0002C          BRB     4$                                          ;
               7E          04    AC   7D 0002E  3$:     MOVQ    FID, -(SP)                                  ; 0183
   00000000G   00                02   FB 00032          CALLS   #2, NMA$OPENFILE                            ;
                                 50   DD 00039  4$:     PUSHL   STATUS                                      ; 0190
               7E                0D   CE 0003B          MNEGL   #13, -(SP)                                  ; 0189
   00000000V   00                02   FB 0003E          CALLS   #2, NML$CHKFILEIO                           ;
                                      04 00045          RET                                                 ; 0192

; Routine Size:  70 bytes,    Routine Base:  $CODE$ + 0000
```

```
;   196          0193  1 %SBTTL 'NML$CLOSEFILE  Close permanent data base file'
;   197          0194  1 GLOBAL ROUTINE nml$closefile (fid) =
;   198          0195  1
;   199          0196  1 !++
;   200          0197  1 ! FUNCTIONAL DESCRIPTION:
;   201          0198  1 !
;   202          0199  1 !       This routine closes the permanent data base file(s) specified by
;   203          0200  1 !       the code in FID.
;   204          0201  1 !
;   205          0202  1 ! FORMAL PARAMETERS:
;   206          0203  1 !
;   207          0204  1 !       FID             Permanent data base file identification code
;   208          0205  1 !                       (NMA$C_OPN_xxxx).
;   209          0206  1 ! ROUTINE VALUE:
;   210          0207  1 ! COMPLETION CODES:
;   211          0208  1 !       Returns a code indicating success.
;   212          0209  1 !
;   213          0210  1 ! SIDE EFFECTS:
;   214          0211  1 !       Causes errors to be signaled.
;   215          0212  1 !
;   216          0213  1 !--
;   217          0214  1
;   218          0215  2 BEGIN
;   219          0216  2
;   220          0217  2 LOCAL
;   221          0218  2     status;
;   222          0219  2
;   223          0220  2 IF .fid EQL nma$c_opn_all THEN          ! If it failed because of ALL
;   224          0221  3     BEGIN
;   225          0222  3     INCRU idx FROM nma$c_opn_min                ! Close all the files by
;   226          0223  3                 TO nma$c_opn_max DO             ! Calling ourselves
;   227          0224  4         BEGIN
;   228          0225  4         IF .idx EQL nma$c_opn_node THEN
;   229          0226  4             status = nml$close_node_file (.idx)
;   230          0227  4         ELSE
;   231          0228  4             status = nma$closefile (.idx);
;   232          0229  3         END;
;   233          0230  3     END
;   234          0231  2 ELSE
;   235          0232  3     BEGIN
;   236          0233  3     IF .fid EQL nma$c_opn_node THEN
;   237          0234  3         status = nml$close_node_file (.fid)
;   238          0235  3     ELSE
;   239          0236  3         status = nma$closefile (.fid);
;   240          0237  2     END;
;   241          0238  2 RETURN .status;
;   242          0239  1 END;                ! OF  nml$closefile


                                        C01C 00000          .ENTRY  NML$CLOSEFILE, Save R2,R3,R4                    ; 0194
                              54 00000000G  00  9E 00002    MOVAB   NML$CLOSE_NODE_FILE, R4                         ;
                              53 00000000G  00  9E 00009    MOVAB   NMA$CLOSEFILE, R3                               ;
                    0000007F    8F    04    AC  D1 00010    CMPL    FID, #127                                       ; 0220
                                        1A  12 00018        BNEQ    4$                                              ;
```

```
                                52  D4 0001A          CLRL    IDX                                     ; 0222
                                52  D5 0001C 1$:      TSTL    IDX                                     ; 0225
                                07  12 0001E          BNEQ    2$                                      ;
                                52  DD 00020          PUSHL   IDX                                     ; 0226
                            64  01  FB 00022          CALLS   #1, NML$CLOSE_NODE_FILE                 ;
                                05  11 00025          BRB     3$                                      ;
                                52  DD 00027 2$:      PUSHL   IDX                                     ; 0228
                            63  01  FB 00029          CALLS   #1, NMA$CLOSEFILE                       ;
                                52  D6 0002C 3$:      INCL    IDX                                     ; 0222
                            07  52  D1 0002E          CMPL    IDX, #7                                 ;
                                E9  1B 00031          BLEQU   1$                                      ;
                                    04 00033          RET                                             ; 0220
                        04  AC  D5 00034 4$:          TSTL    FID                                     ; 0233
                                07  12 00037          BNEQ    5$                                      ;
                        04  AC  DD 00039          PUSHL   FID                                     ; 0234
                            64  01  FB 0003C          CALLS   #1, NML$CLOSE_NODE_FILE                 ;
                                    04 0003F          RET                                             ;
                        04  AC  DD 00040 5$:          PUSHL   FID                                     ; 0236
                            63  01  FB 00043          CALLS   #1, NMA$CLOSEFILE                       ;
                                    04 00046          RET                                             ; 0239

; Routine Size:  71 bytes,    Routine Base:  $CODE$ + 0046
```

```
;  244          0240  1 %SBTTL 'NML$READRECORD  Read record from permanent data base file'
;  245          0241  1 GLOBAL ROUTINE nml$readrecord (fid, key, key_value_dsc,
;  246          0242  1                                bufdsc, rtndsc, node_type) =
;  247          0243  1
;  248          0244  1 !++
;  249          0245  1 ! FUNCTIONAL DESCRIPTION:
;  250          0246  1 !
;  251          0247  1 !       This routine reads a record from a permanent data base file.
;  252          0248  1 !
;  253          0249  1 ! FORMAL PARAMETERS:
;  254          0250  1 !
;  255          0251  1 !       FID             Permanent data base file identification code.
;  256          0252  1 !       KEY             Address of buffer to hold the record key.
;  257          0253  1 !       KEY_VALUE_DSC   Node database only - address of descriptor of node
;  258          0254  1 !                       to read.
;  259          0255  1 !       BUFDSC          Descriptor of buffer to hold record.
;  260          0256  1 !       RTNDSC          Descriptor of data in record.
;  261          0257  1 !       NODE_TYPE       Node database only - address at which to return
;  262          0258  1 !                       node type (executor, remote, or loopnode).
;  263          0259  1 !
;  264          0260  1 ! IMPLICIT INPUTS:
;  265          0261  1 !
;  266          0262  1 !       NONE
;  267          0263  1 !
;  268          0264  1 ! IMPLICIT OUTPUTS:
;  269          0265  1 !
;  270          0266  1 !       NONE
;  271          0267  1 !
;  272          0268  1 ! ROUTINE VALUE:
;  273          0269  1 ! COMPLETION CODES:
;  274          0270  1 !
;  275          0271  1 !       Returns a code indicating success or end of file.
;  276          0272  1 !
;  277          0273  1 ! SIDE EFFECTS:
;  278          0274  1 !
;  279          0275  1 !       Signals error.
;  280          0276  1 !
;  281          0277  1 !--
;  282          0278  1
;  283          0279  2 BEGIN
;  284          0280  2
;  285          0281  2 LOCAL
;  286          0282  2     status;
;  287          0283  2 !
;  288          0284  2 ! Read record.
;  289          0285  2 !
;  290          0286  2 IF .fid NEQ nma$c_opn_node THEN
;  291          0287  2     status = nma$readrec (.fid, .key, .bufdsc, .rtndsc)
;  292          0288  2 ELSE
;  293          0289  2     status = nml$read_node_rec (..key, .key_value_dsc, .node_type,
;  294          0290  2                                 .bufdsc, .rtndsc);
;  295          0291  2 !
;  296          0292  2 ! If the operation was successful or the end of the file was reached (record
;  297          0293  2 ! not found) then return the success code.  Otherwise, cause a file I/O error
;  298          0294  2 ! message to be signalled.
;  299          0295  2 !
;  300          0296  2 IF .status OR
```

```
;  301          0297  2     (.status EQLU rms$_rnf) THEN
;  302          0298  2      RETURN .status
;  303          0299  2 ELSE
;  304          0300  2      RETURN nml$chkfileio (nma$c_sts_fio,
;  305          0301  2                            .status);
;  306          0302  1 END;                              ! End of NML$READRECORD


                                    0000 00000          .ENTRY  NML$READRECORD, Save nothing                    ; 0241
                                04  AC  D5 00002          TSTL    FID                                             ; 0286
                                    11  13 00005          BEQL    1$
                             7E     10  AC  7D 00007          MOVQ    BUFDSC, -(SP)                               ; 0287
                             7E     04  AC  7D 0000B          MOVQ    FID, -(SP)
                00000000G    00         04  FB 0000F          CALLS   #4, NMA$READREC
                                        14  11 00016          BRB     2$
                             7E     10  AC  7D 00018 1$:      MOVQ    BUFDSC, -(SP)                               ; 0290
                                    18  AC  DD 0001C          PUSHL   NODE_TYPE                                   ; 0289
                                    0C  AC  DD 0001F          PUSHL   KEY_VALUE_DSC
                                    08  BC  DD 00022          PUSHL   @KEY
                00000000G    00         05  FB 00025          CALLS   #5, NML$READ_NODE_REC
                             15         50  E8 0002C 2$:      BLBS    STATUS, 3$                                  ; 0296
                000182B2     8F         50  D1 0002F          CMPL    STATUS, #98994                              ; 0297
                                        0C  13 00036          BEQL    3$
                                        50  DD 00038          PUSHL   STATUS                                      ; 0301
                             7E         12  CE 0003A          MNEGL   #18, -(SP)                                  ; 0300
                00000000V    00         02  FB 0003D          CALLS   #2, NML$CHKFILEIO
                                        04 00044 3$:      RET                                                     ; 0302

; Routine Size:  69 bytes,    Routine Base:  $CODE$ + 008D
```

```
;  308     0303  1 %SBTTL 'NML$MATCHRECORD  Match record from permanent data base file'
;  309     0304  1 GLOBAL ROUTINE nml$matchrecord (fid, bufdsc, key_adr,
;  310     0305  1                                id, id_len, id_adr,
;  311     0306  1                                qual_pst, qual_len, qual_adr, rtndsc) =
;  312     0307  1
;  313     0308  1 !++
;  314     0309  1 ! FUNCTIONAL DESCRIPTION:
;  315     0310  1 !
;  316     0311  1 !       This routine matches a record from a permanent data base file.
;  317     0312  1 !
;  318     0313  1 ! FORMAL PARAMETERS:
;  319     0314  1 !
;  320     0315  1 !       FID             Permanent data base file identification code.
;  321     0316  1 !       BUFDSC          Descriptor of buffer to contain the record.
;  322     0317  1 !       KEY_ADR         Address of buffer for record key.
;  323     0318  1 !
;  324     0319  1 !       ID              Code of parameter to match.
;  325     0320  1 !       ID_LEN          Length of parameter value to match.
;  326     0321  1 !       ID_ADR          Address of parameter value string to match.
;  327     0322  1 !
;  328     0323  1 !       QUAL_PST        Parameter Semantic Table entry address of qualifier
;  329     0324  1 !                       parameter to match.
;  330     0325  1 !       QUAL_LEN        Length of qualifier parameter value to match.
;  331     0326  1 !       QUAL_ADR        Address of qualifier parameter value string to match.
;  332     0327  1 !
;  333     0328  1 !       RTNDSC          Descriptor of data in record.
;  334     0329  1 !
;  335     0330  1 ! ROUTINE VALUE:
;  336     0331  1 ! COMPLETION CODES:
;  337     0332  1 !
;  338     0333  1 !       A success code or an error indicating end of file will be returned.
;  339     0334  1 !
;  340     0335  1 ! SIDE EFFECTS:
;  341     0336  1 !
;  342     0337  1 !       Any errors will cause a status message to be signalled.
;  343     0338  1 !
;  344     0339  1 !--
;  345     0340  1
;  346     0341  2 BEGIN
;  347     0342  2
;  348     0343  2 LOCAL
;  349     0344  2     rec_qual_len,
;  350     0345  2     rec_qual_adr,
;  351     0346  2     field_len,
;  352     0347  2     status;
;  353     0348  2
;  354     0349  2 status = 1;
;  355     0350  2 !
;  356     0351  2 ! If looking for KNOWN entities, set up to do a wildcard match.
;  357     0352  2 !
;  358     0353  2 IF .id_len EQL nma$c_ent_kno THEN
;  359     0354  2     field_len = 0
;  360     0355  2 ELSE
;  361     0356  2     field_len = .id_len;
;  362     0357  2 !
;  363     0358  2 ! Read records in the permanent data base until one is found which has
;  364     0359  2 ! fields which match the ID and qualifier (if it's specified) parameters,
```

```
;   365     0360  2 ! or until end-of-file.
;   366     0361  2 !
;   367     0362  2 WHILE .status NEQU rms$_rnf DO
;   368     0363  3     BEGIN
;   369     0364  3     !
;   370     0365  3     ! Get a record with a field that matches the ID.
;   371     0366  3     !
;   372     0367  3     status = nma$matchrec (.fid, .bufdsc, .key_adr,
;   373     0368  3                            .id, .field_len, .id_adr, .rtndsc);
;   374     0369  3     IF .status THEN
;   375     0370  4         BEGIN
;   376     0371  4         MAP qual_pst: REF BBLOCK;
;   377     0372  4         !
;   378     0373  4         ! If there's no qualifier to match, or the record contains
;   379     0374  4         ! a field that matches the qualifier specified, return success.
;   380     0375  4         !
;   381     0376  4         IF .nml$gl_prs_flgs [nml$v_prs_qualifier] THEN
;   382     0377  5             BEGIN
;   383     0378  5             rec_qual_adr = 0;           ! Search from beginning of record.
;   384     0379  5             IF nma$searchfld (.rtndsc, .qual_pst [pst$w_dataid],
;   385     0380  5                               rec_qual_len, rec_qual_adr) THEN
;   386     0381  6                 BEGIN
;   387     0382  6                 IF CH$EQL (.rec_qual_len, .rec_qual_adr,
;   388     0383  6                            .qual_len, .qual_adr) THEN
;   389     0384  6                     RETURN .status;
;   390     0385  5                 END;
;   391     0386  5             END
;   392     0387  4         ELSE
;   393     0388  4             RETURN .status;
;   394     0389  4         END
;   395     0390  3     ELSE
;   396     0391  3         !
;   397     0392  3         ! If the error wasn't "record not found", cause a file I/O error
;   398     0393  3         ! message to be signalled.  (When DEFINEing an entity not already
;   399     0394  3         ! in the permanent database, RMS$_RNF will be returned).
;   400     0395  3         !
;   401     0396  3         IF .status NEQU rms$_rnf THEN
;   402     0397  3             RETURN nml$chkfileio (nma$c_sts_fio,
;   403     0398  3                                   .status);
;   404     0399  2         !
;   405     0400  2         ! The ID or qualifier did not match.  Continue searching the file for a
;   406     0401  2         ! record with both an ID and qualifier that match the ones specified.
;   407     0402  2         !
;   408     0403  2         (.key_adr) <0,16> = .(.key_adr)<0,16> + 1;
;   409     0404  2     END;
;   410     0405  2 RETURN .status;
;   411     0406  1 END;                                ! End of NML$MATCHRECORD


                                                003C 00000          .ENTRY  NML$MATCHRECORD, Save R2,R3,R4,R5          ; 0304
                                             5E     08  C2 00002    SUBL2   #8, SP                                     ;
                                             54     01  D0 00005    MOVL    #1, STATUS                                 ; 0349
                                 FFFFFFFF    8F     14  AC  D1 00008    CMPL    ID_LEN, #-1                            ; 0353
                                                        04  12 00010    BNEQ    1$                                     ;
```

```
                                                    55  D4  00012          CLRL    FIELD_LEN                                                   ; 0354
                                                    04  11  00014          BRB     2$                                                          ;
                                              55    14  AC  D0  00016  1$:  MOVL    ID_LEN, FIELD_LEN                                           ; 0356
                                 000182B2     8F      54  D1  0001A  2$:  CMPL    STATUS, #98994                                              ; 0362
                                                    66  13  00021          BEQL    5$                                                          ;
                                                    28  AC  DD  00023          PUSHL   RTNDSC                                                      ; 0368
                                                    18  AC  DD  00026          PUSHL   ID_ADR                                                      ;
                                                    55  DD  00029          PUSHL   FIELD_LEN                                                   ;
                                              7E    0C  AC  7D  0002B          MOVQ    KEY_ADR, -(SP)                                              ; 0367
                                              7E    04  AC  7D  0002F          MOVQ    FID, -(SP)                                                  ;
                                 00000000G    00      07  FB  00033          CALLS   #7, NMA$MATCHREC                                            ;
                                              54      50  D0  0003A          MOVL    R0, STATUS                                                  ;
                                              2E      54  E9  0003D          BLBC    STATUS, 3$                                                  ; 0369
                           41 00000000G       00      02  E1  00040          BBC     #2, NML$GL_PRS_FLGS, 5$                                     ; 0376
                                                    6E  D4  00048          CLRL    REC_QUAL_ADR                                                ; 0378
                                                    5E  DD  0004A          PUSHL   SP                                                          ; 0379
                                                    08  AE  9F  0004C          PUSHAB  REC_QUAL_LEN                                                ;
                                              7E    1C  BC  3C  0004F          MOVZWL  @QUAL_PST, -(SP)                                            ;
                                                    28  AC  DD  00053          PUSHL   RTNDSC                                                      ;
                                 00000000G    00      04  FB  00056          CALLS   #4, NMA$SEARCHFLD                                           ;
                                              24      50  E9  0005D          BLBC    R0, 4$                                                      ;
20  AC                  00          00      BE      04  AE  2D  00060          CMPC5   REC_QUAL_LEN, @REC_QUAL_ADR, #0, QUAL_LEN, -               ; 0382
                                                    24  BC      00068          @QUAL_ADR                                                           ;
                                                    18  12  0006A          BNEQ    4$                                                          ;
                                                    1B  11  0006C          BRB     5$                                                          ; 0388
                                 000182B2     8F      54  D1  0006E  3$:  CMPL    STATUS, #98994                                              ; 0396
                                                    0D  13  00075          BEQL    4$                                                          ;
                                                    54  DD  00077          PUSHL   STATUS                                                      ; 0398
                                              7E    12  CE  00079          MNEGL   #18, -(SP)                                                  ; 0397
                                 00000000V    00      02  FB  0007C          CALLS   #2, NML$CHKFILEIO                                           ;
                                                    04  00083          RET                                                                 ;
                                                    0C  BC  B6  00084  4$:  INCW    @KEY_ADR                                                    ; 0403
                                                    91  11  00087          BRB     2$                                                          ; 0362
                                        50          54  D0  00089  5$:  MOVL    STATUS, R0                                                  ; 0405
                                                    04  0008C          RET                                                                 ; 0406

; Routine Size:  141 bytes,    Routine Base:  $CODE$ + 00D2
```

```
;  413          0407  1 %SBTTL 'NML$WRITERECORD  Write record to permanent data base file'
;  414          0408  1 GLOBAL ROUTINE nml$writerecord (fid, entity, key, recdsc, write_type) =
;  415          0409  1
;  416          0410  1 !++
;  417          0411  1 ! FUNCTIONAL DESCRIPTION:
;  418          0412  1 !
;  419          0413  1 !       This routine writes the record with the specified key into a
;  420          0414  1 !       permanent data base file.
;  421          0415  1 !
;  422          0416  1 ! FORMAL PARAMETERS:
;  423          0417  1 !
;  424          0418  1 !       FID             Permanent data base file identification code.
;  425          0419  1 !       ENTITY          Entity type.
;  426          0420  1 !       KEY             Address of key of record to be written.
;  427          0421  1 !       RECDSC          Descriptor of record data to be written.
;  428          0422  1 !       WRITE_TYPE      Node database only - specifies whether write is
;  429          0423  1 !                       an update of an existing record, or addition of
;  430          0424  1 !                       a new one.
;  431          0425  1 !
;  432          0426  1 ! IMPLICIT INPUTS:
;  433          0427  1 !
;  434          0428  1 !       NONE
;  435          0429  1 !
;  436          0430  1 ! IMPLICIT OUTPUTS:
;  437          0431  1 !
;  438          0432  1 !       NONE
;  439          0433  1 !
;  440          0434  1 ! ROUTINE VALUE:
;  441          0435  1 ! COMPLETION CODES:
;  442          0436  1 !
;  443          0437  1 !       A code indicating success will be returned.
;  444          0438  1 !
;  445          0439  1 ! SIDE EFFECTS:
;  446          0440  1 !
;  447          0441  1 !       Any errors will cause a file I/O error to be signalled.
;  448          0442  1 !
;  449          0443  1 !--
;  450          0444  1
;  451          0445  2 BEGIN
;  452          0446  2
;  453          0447  2 LOCAL
;  454          0448  2     status;
;  455          0449  2 !
;  456          0450  2 ! Write record.
;  457          0451  2 !
;  458          0452  2 IF .fid NEQ nma$c_opn_node THEN
;  459          0453  2     status = nma$writerec (.fid, .key, .recdsc)
;  460          0454  2 ELSE
;  461          0455  3     BEGIN
;  462          0456  3     status = nml$write_node_rec (.write_type, .entity, .recdsc);
;  463          0457  3     !
;  464          0458  3     ! If a duplicate key was detected, it must be a duplicate node
;  465          0459  3     ! name (that's the only key that can't have a duplicate).  Return
;  466          0460  3     ! the error to the caller so it can be returned to NCP the same way
;  467          0461  3     ! duplicate addresses are.
;  468          0462  3     !
;  469          0463  3     IF .status EQL rms$_dup THEN
```

```
;  470          0464  3          RETURN .status;
;  471          0465  2      END;
;  472          0466  2  !
;  473          0467  2  ! Check the status and return if it is success.  Otherwise, cause a
;  474          0468  2  ! file I/O error message to be signalled.
;  475          0469  2  !
;  476          0470  2  RETURN nml$chkfileio (nma$c_sts_fio, .status);
;  477          0471  1  END;                                ! End of NML$WRITERECORD


                                        0000 00000          .ENTRY  NML$WRITERECORD, Save nothing                   ; 0408
                                    04  AC  D5 00002          TSTL    FID                                             ; 0452
                                        10  13 00005          BEQL    1$                                              ;
                                    7E  0C  AC  7D 00007          MOVQ    KEY, -(SP)                                  ; 0453
                                    04  AC  DD 0000B          PUSHL   FID                                             ;
                00000000G   00          03  FB 0000E          CALLS   #3, NMA$WRITEREC                                ;
                                        19  11 00015          BRB     2$                                              ;
                                    10  AC  DD 00017 1$:      PUSHL   RECDSC                                          ; 0456
                                    08  AC  DD 0001A          PUSHL   ENTITY                                          ;
                                    14  AC  DD 0001D          PUSHL   WRITE_TYPE                                      ;
                00000000G   00          03  FB 00020          CALLS   #3, NML$WRITE_NODE_REC                          ;
                000184EC    8F          50  D1 00027          CMPL    STATUS, #99564                                  ; 0463
                                        0C  13 0002E          BEQL    3$                                              ;
                                        50  DD 00030 2$:      PUSHL   STATUS                                          ; 0470
                                    7E      12  CE 00032          MNEGL   #18, -(SP)                                  ;
                00000000V   00          02  FB 00035          CALLS   #2, NML$CHKFILEIO                               ;
                                            04 0003C 3$:      RET                                                     ; 0471

; Routine Size:  61 bytes,    Routine Base:  $CODE$ + 015F
```

```
;  479          0472  1 %SBTTL 'NML$DELETRECORD  Delete record from permanent data base file'
;  480          0473  1 GLOBAL ROUTINE nml$deletrecord (fid, key, key_value_dsc) =
;  481          0474  1
;  482          0475  1 !++
;  483          0476  1 ! FUNCTIONAL DESCRIPTION:
;  484          0477  1 !
;  485          0478  1 !       This routine deletes the record with the specified key from
;  486          0479  1 !       the permanent data base file.
;  487          0480  1 !
;  488          0481  1 ! FORMAL PARAMETERS:
;  489          0482  1 !
;  490          0483  1 !       FID             Permanent data base file identification code.
;  491          0484  1 !       KEY             Address of key of record to be written.
;  492          0485  1 !       KEY_VALUE_DSC   Node database only - address of descriptor of node
;  493          0486  1 !                       ID.
;  494          0487  1 !
;  495          0488  1 ! IMPLICIT INPUTS:
;  496          0489  1 !
;  497          0490  1 !       NONE
;  498          0491  1 !
;  499          0492  1 ! IMPLICIT OUTPUTS:
;  500          0493  1 !
;  501          0494  1 !       NONE
;  502          0495  1 !
;  503          0496  1 ! ROUTINE VALUE:
;  504          0497  1 ! COMPLETION CODES:
;  505          0498  1 !
;  506          0499  1 !       A code indicating success will be returned.
;  507          0500  1 !
;  508          0501  1 ! SIDE EFFECTS:
;  509          0502  1 !
;  510          0503  1 !       Any errors will cause a file I/O error to be signalled.
;  511          0504  1 !
;  512          0505  1 !--
;  513          0506  1
;  514          0507  2 BEGIN
;  515          0508  2
;  516          0509  2 LOCAL
;  517          0510  2     status;
;  518          0511  2 !
;  519          0512  2 ! Delete record from the permanent data base file.
;  520          0513  2 !
;  521          0514  2 IF .fid NEQ nma$c_opn_node THEN
;  522          0515  2     status = nma$deleterec (.fid, .key)
;  523          0516  2 ELSE
;  524          0517  2     status = nml$delete_node_rec (..key, .key_value_dsc);
;  525          0518  2 !
;  526          0519  2 ! Check the status and return if it is success.  Otherwise, cause a
;  527          0520  2 ! file I/O error message to be signalled.
;  528          0521  2 !
;  529          0522  2 RETURN nml$chkfileio (nma$c_sts_fio, .status);
;  530          0523  1 END;                                     ! End of NML$DELETRECORD
```

```
                                    0000 00000          .ENTRY  NML$DELETERECORD, Save nothing                  ; 0473
                                04  AC   D5 00002       TSTL    FID                                             ; 0514
                                    0D   13 00005       BEQL    1$                                              ;
                             7E     04   AC 7D 00007     MOVQ    FID, -(SP)                                      ; 0515
                00000000G    00          02 FB 0000B     CALLS   #2, NMA$DELETEREC                               ;
                                         0D 11 00012     BRB     2$                                              ;
                                0C   AC   DD 00014 1$:   PUSHL   KEY_VALUE_DSC                                  ; 0517
                                08   BC   DD 00017       PUSHL   @KEY                                            ;
                00000000G    00          02 FB 0001A     CALLS   #2, NML$DELETE_NODE_REC                         ;
                                         50 DD 00021 2$:  PUSHL   STATUS                                         ; 0522
                             7E          12 CE 00023     MNEGL   #18, -(SP)                                      ;
                00000000V    00          02 FB 00026     CALLS   #2, NML$CHKFILEIO                               ;
                                         04 0002D        RET                                                     ; 0523

; Routine Size:  46 bytes,    Routine Base:  $CODE$ + 019C
```

```
;  532          0524  1 %SBTTL 'NML$CHKFILEIO  Return file I/O status'
;  533          0525  1 GLOBAL ROUTINE nml$chkfileio (opcode, status) =
;  534          0526  1
;  535          0527  1 !++
;  536          0528  1 ! FUNCTIONAL DESCRIPTION:
;  537          0529  1 !
;  538          0530  1 !       This routine checks the status of file I/O operations and
;  539          0531  1 !       signals a status message if an error occurs.
;  540          0532  1 !
;  541          0533  1 ! FORMAL PARAMETERS:
;  542          0534  1 !
;  543          0535  1 !       OPCODE          Operation error code to return in message.
;  544          0536  1 !       STATUS          Operation status to be examined.
;  545          0537  1 !
;  546          0538  1 ! IMPLICIT INPUTS:
;  547          0539  1 !
;  548          0540  1 !       NONE
;  549          0541  1 !
;  550          0542  1 ! IMPLICIT OUTPUTS:
;  551          0543  1 !
;  552          0544  1 !       NONE
;  553          0545  1 !
;  554          0546  1 ! ROUTINE VALUE:
;  555          0547  1 ! COMPLETION CODES:
;  556          0548  1 !
;  557          0549  1 !       Returns success (NML$_STS_SUC) if the STATUS indicates success.
;  558          0550  1 !
;  559          0551  1 ! SIDE EFFECTS:
;  560          0552  1 !
;  561          0553  1 !       Signals an error message if STATUS indicates error.
;  562          0554  1 !
;  563          0555  1 !--
;  564          0556  1
;  565          0557  2 BEGIN
;  566          0558  2
;  567          0559  2 LOCAL
;  568          0560  2     msgsize;                    ! Message size
;  569          0561  2 !
;  570          0562  2 ! If the status is not success then build a status message and signal it.
;  571          0563  2 !
;  572          0564  2 IF NOT .status THEN
;  573          0565  3     BEGIN
;  574          0566  3     !
;  575          0567  3     ! File operation failed, so signal error message.
;  576          0568  3     !
;  577          0569  3     nml$ab_msgblock [msb$l_flags] = msb$m_det_fld OR msb$m_msg_fld;
;  578          0570  3     !
;  579          0571  3     ! If the file was not opened for the specified access, the calling process
;  580          0572  3     ! doesn't have OPER privilege (or the file would have been opened for any
;  581          0573  3     ! access).
;  582          0574  3     !
;  583          0575  3     IF .status EQL rms$_fac THEN
;  584          0576  4         nml$ab_msgblock [msb$b_code] = nma$c_sts_pri    ! Privilege violation.
;  585          0577  3     ELSE
;  586          0578  3     !
;  587          0579  3     ! For any other file access error, return the error supplied by the calling
;  588          0580  3     ! routine.
```

```
;  589          0581  3     !
;  590          0582  4         BEGIN
;  591          0583  4         nml$ab_msgblock [msb$l_flags] = .nml$ab_msgblock [msb$l_flags] OR
;  592          0584  4                                         msb$m_sysm_fld;
;  593          0585  4         nml$ab_msgblock [msb$b_code] = .opcode; ! Add error code
;  594          0586  4         nml$ab_msgblock [msb$l_text] = .status;
;  595          0587  3         END;
;  596          0588  3     nml$ab_msgblock [msb$w_detail] = nma$c_fopdtl_pdb; ! Add file id code
;  597          0589  3     nml$bld_reply (nml$ab_msgblock, msgsize);
;  598          0590  3     $signal_msg (nml$ab_sndbuffer, .msgsize);
;  599          0591  2     END;
;  600          0592  2
;  601          0593  2 RETURN nml$_sts_suc
;  602          0594  1 END;                                ! End of NML$CHKFILEIO


                                                  0004 00000          .ENTRY  NML$CHKFILEIO, Save R2                          ; 0525
                                52 00000000G   00  9E 00002          MOVAB   NML$AB_MSGBLOCK, R2                             ;
                                5E             04  C2 00009          SUBL2   #4, SP                                          ;
                                44       08    AC  E8 0000C          BLBS    STATUS, 3$                                      ; 0564
                                62             06  D0 00010          MOVL    #6, NML$AB_MSGBLOCK                             ; 0569
                        00018514   8F    08    AC  D1 00013          CMPL    STATUS, #99604                                  ; 0575
                                               06  12 0001B          BNEQ    1$                                              ;
                              04   A2          03  8E 0001D          MNEGB   #3, NML$AB_MSGBLOCK+4                           ; 0576
                                               0E  11 00021          BRB     2$                                              ;
                                62       40    8F  88 00023  1$:     BISB2   #64, NML$AB_MSGBLOCK                            ; 0583
                              04   A2       04  AC  90 00027          MOVB    OPCODE, NML$AB_MSGBLOCK+4                       ; 0585
                              0C   A2       08  AC  D0 0002C          MOVL    STATUS, NML$AB_MSGBLOCK+12                      ; 0586
                                           08  A2  B4 00031  2$:     CLRW    NML$AB_MSGBLOCK+8                               ; 0588
                                         4004  8F  BB 00034          PUSHR   #^M<R2,SP>                                      ; 0589
                        00000000G      00        02  FB 00038          CALLS   #2, NML$BLD_REPLY                               ;
                                               6E  DD 0003F          PUSHL   MSGSIZE                                         ; 0590
                                   00000000G   00  9F 00041          PUSHAB  NML$AB_SNDBUFFER                                ;
                                   01F90000    8F  DD 00047          PUSHL   #33095680                                       ;
                        00000000G      00        03  FB 0004D          CALLS   #3, LIB$SIGNAL                                  ;
                                50             01  D0 00054  3$:     MOVL    #1, R0                                          ; 0593
                                               04  00057          RET                                                     ; 0594

; Routine Size:  88 bytes,    Routine Base:  $CODE$ + 01CA
```

```
;  604          0595  1 END                                   ! End of module
;  605          0596  1
;  606          0597  0 ELUDOM



                                                        .EXTRN  LIB$SIGNAL

;                               PSECT SUMMARY
;
;       Name                    Bytes                   Attributes
;
;  $CODE$                        546  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        28         8          27        00:00.1
;  _$255$DUA28:[SHRLIB]NMALIBRY.L32;1      887        10         1          47        00:00.2
;  _$255$DUA28:[SYSLIB]STARLET.L32;1      9776         4         0         581        00:02.1



;                                   COMMAND QUALIFIERS

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

; Size:         546 code + 0 data bytes
; Run Time:         00:13.0
; Elapsed Time:     00:35.4
; Lines/CPU Min:     2753
; Lexemes/CPU-Min:   7900
; Memory Used:  101 pages
; Compilation Complete
```

NMLFORWRD
LIS

NMLENTRY
LIS

NMLDEFINE
LIS

NMLFILEIO
LIS

NMLLIB
LIS

NMLINISTA
LIS

NMLDISC
LIS