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
```

K 15

```
**FILE**ID**NMAFILES
```

```
NN      NN  MM      MM    AAAAAA    FFFFFFFFFF    IIIIII    LL          EEEEEEEEEE    SSSSSSSS
NN      NN  MM      MM    AAAAAA    FFFFFFFFFF    IIIIII    LL          EEEEEEEEEE    SSSSSSSS
NN      NN  MMMM  MMMM  AA      AA  FF              II      LL          EE          SS
NN      NN  MMMM  MMMM  AA      AA  FF              II      LL          EE          SS
NNNN    NN  MM  MM  MM  AA      AA  FF              II      LL          EE          SS
NNNN    NN  MM  MM  MM  AA      AA  FF              II      LL          EE          SS
NN  NN  NN  MM      MM  AA      AA  FFFFFFFF        II      LL          EEEEEEEE      SSSSSS
NN  NN  NN  MM      MM  AA      AA  FFFFFFFF        II      LL          EEEEEEEE      SSSSSS
NN    NNNN  MM      MM  AAAAAAAAAA  FF              II      LL          EE                SS
NN    NNNN  MM      MM  AAAAAAAAAA  FF              II      LL          EE                SS
NN      NN  MM      MM  AA      AA  FF              II      LL          EE                SS
NN      NN  MM      MM  AA      AA  FF              II      LL          EE                SS        ....
NN      NN  MM      MM  AA      AA  FF            IIIIII    LLLLLLLLLL  EEEEEEEEEE  SSSSSSSS          ....
NN      NN  MM      MM  AA      AA  FF            IIIIII    LLLLLLLLLL  EEEEEEEEEE  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
```

```
0 %TITLE  'File Routines for Network Management'
0 MODULE NMAFILES (
0                 LANGUAGE (BLISS32),
0                 ADDRESSING_MODE (NONEXTERNAL=GENERAL),
0                 ADDRESSING_MODE (EXTERNAL=GENERAL),
0                 IDENT = 'V04-000'
0                 ) =
1 BEGIN
1
1 !
1 !****************************************************************************
1 !*                                                                          *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
1 !*  ALL RIGHTS RESERVED.                                                    *
1 !*                                                                          *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
1 !*  TRANSFERRED.                                                            *
1 !*                                                                          *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
1 !*  CORPORATION.                                                            *
1 !*                                                                          *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
1 !*                                                                          *
1 !*                                                                          *
1 !****************************************************************************
1 !
1
1 !++
1 ! FACILITY:     DECnet Network Management Layer (NMA)
1 !
1 ! ABSTRACT:
1 !
1 !      This module contains routines which manage the files used by
1 !      network management.  These files contain permanent data about the
1 !      configuration of the network.
1 !
1 ! ENVIRONMENT:  VAX/VMS Operating System
1 !
1 ! AUTHOR:       Darrell Duffy   , CREATION DATE:  18-December-1979
1 !
1 ! MODIFIED BY:
1 !
1 !      V03-007 MKP0007         Kathy Perko             2-April-1984
1 !              If call is made to open a file and it is already open,
1 !              do a $REWIND to get back to the beginning of the file.
1 !
1 !      V03-006 MKP0006         Kathy Perko             5-Feb-1984
1 !              Fix NMA$READREC so that the correct key is returned to
1 !              the caller.
1 !
```

```
1 !       V03-005 MKP0005         Kathy Perko             6-Aug-1983
1 !               Enhance node permanent database to use multiple ISAM keys
1 !               so it's faster to access.  When returning permanent database
1 !               records, don't include key in the data returned.
1 !
1 !       V03-004 MKP0004         Kathy Perko             25-April-1983
1 !               Allow multiple NMLs to read and update the permanent database
1 !               files at once.
1 !
1 !       V03-004 MKP0004         Kathy Perko             25-April-1983
1 !               Add NI configurator permanent database.
1 !
1 !       V03-003 MKP0003         Kathy Perko             12-Nov-1982
1 !               Allow multiple NMLs to update the permanent database
1 !               files at once.
1 !
1 !       V03-002 MKP0002         Kathy Perko             18-Oct-1982
1 !               Change the way NML opens and closes files so that it checks
1 !               to see if the operation has already been done. This will
1 !               improve the performance of operations which now open and close
1 !               various files more than once.
1 !
1 !       V03-001 MKP0001         Kathy Perko             3-Aug-1982
1 !               Split module permanent data base into two: one for X25 and
1 !               one for X29.
1 !
1 !       V02-001 LMK0001         Len Kawell              27-Jul-1981
1 !               Add CIRCUIT and MODULE files.
1 !--
```

```
;    88          0087  1 %SBTTL  'Definitions'
;    89          0088  1
;    90          0089  1 !
;    91          0090  1 ! TABLE OF CONTENTS:
;    92          0091  1 !
;    93          0092  1
;    94          0093  1 FORWARD ROUTINE
;    95          0094  1     NMA$OPENFILE,                          ! Open file by id
;    96          0095  1     NMA$SELECTFILE,                        ! Find filedescriptor by fileid
;    97          0096  1     NMA$OPENFAB,                           ! Open a file by descriptor
;    98          0097  1     NMA$CLOSEFILE,                         ! Close a file by id
;    99          0098  1     NMA$MATCHREC,                          ! Find record with specified field
;   100          0099  1     NMA$READREC,                           ! Get a record from a file
;   101          0100  1     NMA$WRITEREC,                          ! Put a record to a file
;   102          0101  1     NMA$DELETEREC;                         ! Delete a record from a file
;   103          0102  1
;   104          0103  1 !
;   105          0104  1 ! INCLUDE FILES:
;   106          0105  1 !
;   107          0106  1
;   108          0107  1 LIBRARY 'LIB$:NMLLIB.L32';
;   109          0108  1 LIBRARY 'SHRLIB$:NMALIBRY.L32';
;   110          0109  1 LIBRARY 'SYS$LIBRARY:STARLET.L32';
;   111          0110  1
;   112          0111  1 !
;   113          0112  1 ! MACROS:
;   114          0113  1 !
;   115          0114  1
;   116          0115  1 !
;   117          0116  1 ! Define fields in a file descriptor.
;   118          0117  1 !
;   119          0118  1
;   120          0119  1 FIELD
;   121          0120  1     FDSCFLDS =
;   122          0121  1     SET
;   123          0122  1         FDSCFNS = [0,  0, 32, 0],
;   124          0123  1         FDSCFNA = [4,  0, 32, 0],
;   125          0124  1         FDSCFAB = [8,  0, 32, 0],
;   126          0125  1         FDSCRAB = [12, 0, 32, 0]
;   127          0126  1     TES;
;   128          0127  1
;   129          0128  1 !
;   130          0129  1 ! Macro to build file descriptors.
;   131          0130  1 !
;   132          0131  1 !       FILE            Designator of the file
;   133          0132  1 !       FILENAME        Filename string for file
;   134          0133  1 !
;   135          0134  1
;   136          0135  1 MACRO
;   137        M 0136  1     $NMA_BLDFILEDSC [FILE, FILENAME] =  ! Build as many as you like
;   138        M 0137  1
;   139        M 0138  1     OWN                                 ! Declare the fab and rab
;   140        M 0139  1         %NAME ('NMA$A_', FILE, '_FAB') : $FAB_DECL,
;   141        M 0140  1         %NAME ('NMA$A_', FILE, '_RAB') : $RAB_DECL;
;   142        M 0141  1
;   143        M 0142  1     BIND                                        ! The descriptor
;   144        M 0143  1         %NAME ('NMA$A_', FILE, '_DSC') =
:  145        M 0144  1         UPLIT
:  146        M 0145  1             (
:  147        M 0146  1             %CHARCOUNT (FILENAME),              ! Descriptor of filename str
:  148        M 0147  1             UPLIT BYTE (FILENAME),              ! Addr
:  149        M 0148  1             %NAME ('NMA$A_', FILE, '_FAB'),     ! Fab address
:  150        M 0149  1             %NAME ('NMA$A_', FILE, '_RAB')      ! Rab address
:  151        M 0150  1             );
:  152          0151  1 %;
:  153          0152  1
:  154          0153  1 !
:  155          0154  1 ! EQUATED SYMBOLS:
:  156          0155  1 !
:  157          0156  1
:  158          0157  1 !
:  159          0158  1 ! OWN STORAGE:
:  160          0159  1 !
:  161          0160  1
:  162          0161  1 OWN
:  163          0162  1     NMA$W_KEYBUF : WORD;                ! Key buffer
:  164          0163  1
:  165        P 0164  1 $NMA_BLDFILEDSC
:  166        P 0165  1     (
:  167        P 0166  1     NODE,       'NETNODE',              ! Remote node database
:  168        P 0167  1     LINE,       'NETLINE',              ! Line database
:  169        P 0168  1     LOG,        'NETLOGING',            ! Logging database
:  170        P 0169  1     OBJ,        'NETOBJECT',            ! Object database
:  171        P 0170  1     CIR,        'NETCIRC',              ! Circuit database
:  172        P 0171  1     X25,        'NETX25',               ! X25 Module database
:  173        P 0172  1     X29,        'NETX29',               ! X29 Module database
:  174        P 0173  1     CNF,        'NETCONF'               ! Ni Configurator Module database
:  175          0174  1     );
:  176          0175  1
:  177          0176  1 !
:  178          0177  1 ! EXTERNAL REFERENCES:
:  179          0178  1 !
:  180          0179  1
:  181          0180  1 EXTERNAL ROUTINE
:  182          0181  1     NML$DEBUG_MSG,
:  183          0182  1     NML$DEBUG_TXT,
:  184          0183  1     NML$LOGFILEOP,
:  185          0184  1     NML$LOGRECORDOP;
:  186          0185  1
```

```
1 %SBTTL 'NMA$OPENFILE  Open a specified file'
1 GLOBAL ROUTINE NMA$OPENFILE (FILEID, ACCESS) =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine opens a specified file for specified access.
1 !       The fileid specifies the file, or all files and the access
1 !       specifies read only or read write.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       FILEID          Value of the fileid parameter (NMA$C_OPN_xxxxx)
1 !       ACCESS          Value of the access parameter (NMA$C_OPN_AC_Rx)
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       Failure or RMS error
1 !
1 !--
1
2 BEGIN
2
2 LOCAL
2     FAB : REF BLOCK [1,BYTE],                   ! The fab for the file
2     FILEDSC : REF BLOCK [1, BYTE]               ! File descriptor
2               FIELD (FDSCFLDS),
2     RAB,                                        ! The rab for the file
2     STATUS;                                     ! Status return
2
2 IF .FILEID EQL NMA$C_OPN_ALL THEN              ! If ALL
3     BEGIN
3
3     INCRU IDX FROM NMA$C_OPN_MIN                ! Open all the files by
3               TO NMA$C_OPN_MAX DO               ! Calling ourselves
4         BEGIN
4         STATUS = NMA$OPENFILE (.IDX, .ACCESS);  ! Call ourself to open it
4         IF NOT .STATUS THEN
4             EXITLOOP;
4         END
3     END
2 ELSE
3     BEGIN
3     STATUS = NMA$_SUCCESS;
3     IF NMA$SELECTFILE (.FILEID, FILEDSC) THEN   ! Obtain descriptor address
4         BEGIN
4         FAB = .FILEDSC [FDSCFAB];               ! Get address of FAB
4         IF .FAB [FAB$W_IFI] EQL 0 THEN          ! If file isn't open, do it.
5             BEGIN
5             STATUS = NMA$OPENFAB (.FILEDSC, .ACCESS); ! Open file by descriptor
5             IF .STATUS THEN
5                 NML$LOGFILEOP (DBG$C_FILEIO,
5                                .FILEID,
5                                $ASCID ('file opened.'));
5             END
4         ELSE
```

```
;  245      0243 4              !
;  246      0244 4              ! The file is already open  so don't reopen it.  However,
;  247      0245 4              ! set RMS's "next record" back to the beginning of the file.
;  248      0246 4              !
;  249      0247 5              BEGIN
;  250      0248 5              RAB = .FILEDSC [FDSCRAB];           ! Point to the rab
;  251      0249 5              $REWIND (RAB = .RAB);
;  252      0250 4              END;
;  253      0251 4          END
;  254      0252 3      ELSE
;  255      0253 3          RETURN NMA$_BADFID;                     ! If not all, return failure
;  256      0254 2      END;
;  257      0255 2
;  258      0256 2 RETURN .STATUS
;  259      0257 1 END;


                                                                     .TITLE  NMAFILES File Routines for Network Management
                                                                     .IDENT  \V04-000\

                                                                     .PSECT  $PLIT$,NOWRT,NOEXE,2

                      45  44  4F  4E  54  45  4E  00000 P.AAB:  .ASCII  \NETNODE\                                          ;
                                                  00007         .BLKB   1
                                        00000007  00008 P.AAA:  .LONG   7                                                  ;
                    00000000' 00000000' 00000000' 0000C         .ADDRESS P.AAB, NMA$A_NODE_FAB, NMA$A_NODE_RAB              ;
                      45  4E  49  4C  54  45  4E  00018 P.AAD:  .ASCII  \NETLINE\                                          ;
                                                  0001F         .BLKB   1
                                        00000007  00020 P.AAC:  .LONG   7                                                  ;
                    00000000' 00000000' 00000000' 00024         .ADDRESS P.AAD, NMA$A_LINE_FAB, NMA$A_LINE_RAB              ;
              47  4E  49  47  4F  4C  54  45  4E  00030 P.AAF:  .ASCII  \NETLOGING\                                        ;
                                                  00039         .BLKB   3
                                        00000009  0003C P.AAE:  .LONG   9                                                  ;
                    00000000' 00000000' 00000000' 00040         .ADDRESS P.AAF, NMA$A_LOG_FAB, NMA$A_LOG_RAB                ;
              54  43  45  4A  42  4F  54  45  4E  0004C P.AAH:  .ASCII  \NETOBJECT\                                        ;
                                                  00055         .BLKB   3
                                        00000009  00058 P.AAG:  .LONG   9                                                  ;
                    00000000' 00000000' 00000000' 0005C         .ADDRESS P.AAH, NMA$A_OBJ_FAB, NMA$A_OBJ_RAB                ;
                      43  52  49  43  54  45  4E  00068 P.AAJ:  .ASCII  \NETCIRC\                                          ;
                                                  0006F         .BLKB   1
                                        00000007  00070 P.AAI:  .LONG   7                                                  ;
                    00000000' 00000000' 00000000' 00074         .ADDRESS P.AAJ, NMA$A_CIR_FAB, NMA$A_CIR_RAB                ;
                          35  32  58  54  45  4E  00080 P.AAL:  .ASCII  \NETX25\                                           ;
                                                  00086         .BLKB   2
                                        00000006  00088 P.AAK:  .LONG   6                                                  ;
                    00000000' 00000000' 00000000' 0008C         .ADDRESS P.AAL, NMA$A_X25_FAB, NMA$A_X25_RAB                ;
                          39  32  58  54  45  4E  00098 P.AAN:  .ASCII  \NETX29\                                           ;
                                                  0009E         .BLKB   2
                                        00000006  000A0 P.AAM:  .LONG   6                                                  ;
                    00000000' 00000000' 00000000' 000A4         .ADDRESS P.AAN, NMA$A_X29_FAB, NMA$A_X29_RAB                ;
                      46  4E  4F  43  54  45  4E  000B0 P.AAP:  .ASCII  \NETCONF\                                          ;
                                                  000B7         .BLKB   1
                                        00000007  000B8 P.AAO:  .LONG   7                                                  ;
                    00000000' 00000000' 00000000' 000BC         .ADDRESS P.AAP, NMA$A_CNF_FAB, NMA$A_CNF_RAB                ;
  2E  64  65  6E  65  70  6F  20  65  6C  69  66  000C8 P.AAR:  .ASCII  \file opened.\                                     ;
                                        0000000C  000D4 P.AAQ:  .LONG   12                                                 ;
                                        00000000' 000D8         .ADDRESS P.AAR                                             ;
```

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

                                                               00000 NMA$W_KEYBUF:
                                                                              .BLKB   2
                                                               00002          .BLKB   2
                                                               00004 NMA$A_NODE_FAB:
                                                                              .BLKB   80
                                                               00054 NMA$A_NODE_RAB:
                                                                              .BLKB   68
                                                               00098 NMA$A_LINE_FAB:
                                                                              .BLKB   80
                                                               000E8 NMA$A_LINE_RAB:
                                                                              .BLKB   68
                                                               0012C NMA$A_LOG_FAB:
                                                                              .BLKB   80
                                                               0017C NMA$A_LOG_RAB:
                                                                              .BLKB   68
                                                               001C0 NMA$A_OBJ_FAB:
                                                                              .BLKB   80
                                                               00210 NMA$A_OBJ_RAB:
                                                                              .BLKB   68
                                                               00254 NMA$A_CIR_FAB:
                                                                              .BLKB   80
                                                               002A4 NMA$A_CIR_RAB:
                                                                              .BLKB   68
                                                               002E8 NMA$A_X25_FAB:
                                                                              .BLKB   80
                                                               00338 NMA$A_X25_RAB:
                                                                              .BLKB   68
                                                               0037C NMA$A_X29_FAB:
                                                                              .BLKB   80
                                                               003CC NMA$A_X29_RAB:
                                                                              .BLKB   68
                                                               00410 NMA$A_CNF_FAB:
                                                                              .BLKB   80
                                                               00460 NMA$A_CNF_RAB:
                                                                              .BLKB   68

                                                                     NMA$A_NODE_DSC=      P.AAA
                                                                     NMA$A_LINE_DSC=      P.AAC
                                                                     NMA$A_LOG_DSC=       P.AAE
                                                                     NMA$A_OBJ_DSC=       P.AAG
                                                                     NMA$A_CIR_DSC=       P.AAI
                                                                     NMA$A_X25_DSC=       P.AAK
                                                                     NMA$A_X29_DSC=       P.AAM
                                                                     NMA$A_CNF_DSC=       P.AAO
                                                                              .EXTRN  NML$DEBUG_MSG, NML$DEBUG_TXT
                                                                              .EXTRN  NML$LOGFILEOP, NML$LOGRECORDOP
                                                                              .EXTRN  SYS$REWIND

                                                                              .PSECT  $CODE$,NOWRT,2

                                                         000C  00000          .ENTRY  NMA$OPENFILE, Save R2,R3                          ; 0187
                              5E                     04    C2  00002          SUBL2   #4, SP                                            ;
                 0000007F     8F          04         AC    D1  00005          CMPL    FILEID, #127                                      ; 0217
                                                     1A    12  0000D          BNEQ    2$                                                ;
```

```
                                          52  D4 0000F         CLRL    IDX                                           ; 0223
                                    08    AC  DD 00011 1$:     PUSHL   ACCESS                                        ;
                                          52  DD 00014         PUSHL   IDX                                           ;
                        E6    AF          02  FB 00016         CALLS   #2, NMA$OPENFILE                              ;
                              53          50  D0 0001A         MOVL    R0, STATUS                                    ;
                              5A          53  E9 0001D         BLBC    STATUS, 4$                                    ; 0224
                                          52  D6 00020         INCL    IDX                                           ; 0220
                              07          52  D1 00022         CMPL    IDX, #7                                       ;
                                          EA  1B 00025         BLEQU   1$                                            ;
                                          51  11 00027         BRB     4$                                            ; 0218
                              53          01  D0 00029 2$:     MOVL    #1, STATUS                                    ; 0230
                                          5E  DD 0002C         PUSHL   SP                                            ; 0231
                                    04    AC  DD 0002E         PUSHL   FILEID                                        ;
                    00000000V   00        02  FB 00031         CALLS   #2, NMA$SELECTFILE                            ;
                              43          50  E9 00038         BLBC    R0, 5$                                        ;
                              50          6E  D0 0003B         MOVL    FILEDSC, R0                                   ; 0233
                              51    08    A0  D0 0003E         MOVL    8(R0), FAB                                    ;
                                    02    A1  B5 00042         TSTW    2(FAB)                                        ; 0234
                                          26  12 00045         BNEQ    3$                                            ;
                                    08    AC  DD 00047         PUSHL   ACCESS                                        ; 0236
                                          50  DD 0004A         PUSHL   R0                                            ;
                    00000000V   00        02  FB 0004C         CALLS   #2, NMA$OPENFAB                               ;
                              53          50  D0 00053         MOVL    R0, STATUS                                    ;
                              21          53  E9 00056         BLBC    STATUS, 4$                                    ; 0237
                                00000000' 00  9F 00059         PUSHAB  P.AAQ                                         ; 0240
                                    04    AC  DD 0005F         PUSHL   FILEID                                        ; 0239
                                          01  DD 00062         PUSHL   #1                                            ; 0238
                    00000000G   00        03  FB 00064         CALLS   #3, NML$LOGFILEOP                             ;
                                          0D  11 0006B         BRB     4$                                            ; 0234
                              50    0C    A0  D0 0006D 3$:     MOVL    12(R0), RAB                                   ; 0248
                                          50  DD 00071         PUSHL   RAB                                           ; 0249
                    00000000G   00        01  FB 00073         CALLS   #1, SYS$REWIND                                ;
                              50          53  D0 0007A 4$:     MOVL    STATUS, R0                                    ; 0256
                                              04 0007D         RET                                                   ;
                                          50  D4 0007E 5$:     CLRL    R0                                            ; 0257
                                              04 00080         RET                                                   ;

; Routine Size:  129 bytes,    Routine Base:  $CODE$ + 0000
```

```
1 %SBTTL 'NMA$SELECTFILE  Return a file descriptor'
1 GLOBAL ROUTINE NMA$SELECTFILE (FILEID, FILEDSC) =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine returns the address of the file descriptor for a
1 !       specified file.  Failure is returned if the fileid is not
1 !       valid.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       FILEID          Value of the fileid (NMA$C_OPN_xxxxx)
1 !       FILEDSC         Address to return address of file descriptor
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       Success or failure
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !
1 !--
1
2    BEGIN
2
2    LOCAL
2        STATUS;
2
2    STATUS = NMA$_SUCCESS;
2
2    .FILEDSC =                                      ! Obtain the file descriptor
3        BEGIN                                       ! Address
3
3        CASE .FILEID FROM NMA$C_OPN_MIN TO NMA$C_OPN_MAX OF
3            SET
3
3            [NMA$C_OPN_NODE]: NMA$A_NODE_DSC;
3            [NMA$C_OPN_LINE]: NMA$A_LINE_DSC;
3            [NMA$C_OPN_LOG]:  NMA$A_LOG_DSC;
3            [NMA$C_OPN_OBJ]:  NMA$A_OBJ_DSC;
3            [NMA$C_OPN_CIR]:  NMA$A_CIR_DSC;
3            [NMA$C_OPN_X25]:  NMA$A_X25_DSC;
3            [NMA$C_OPN_X29]:  NMA$A_X29_DSC;
3            [NMA$C_OPN_CNF]:  NMA$A_CNF_DSC;
3            [INRANGE,
3             OUTRANGE]:                             ! Code not known, fail
```

```
;  318         0315  4                 BEGIN
;  319         0316  4
;  320         0317  4                 STATUS = NMA$_BADFID;
;  321         0318  4                 0                                   ! Return invalid descriptor
;  322         0319  4
;  323         0320  3                 END;
;  324         0321  3
;  325         0322  3              TES
;  326         0323  2          END;
;  327         0324  2
;  328         0325  2      RETURN .STATUS
;  329         0326  2
;  330         0327  1      END;



                                          0004 00000          .ENTRY  NMA$SELECTFILE, Save R2                             ; 0259
                         52 00000000'  00   9E 00002          MOVAB   NMA$A_NODE_DSC, R2                                  ;
                         51            01   D0 00009          MOVL    #1, STATUS                                          ; 0297
             07          00       04   AC   CF 0000C          CASEL   FILEID, #0, #7                                      ; 0302
   0025      001F        0019          0014    00011 1$:      .WORD   2$-1$,-                                             ;
   003F      0038        0031          002B    00019                  3$-1$,-                                             ;
                                                                      4$-1$,-                                             ;
                                                                      5$-1$,-                                             ;
                                                                      6$-1$,-                                             ;
                                                                      7$-1$,-                                             ;
                                                                      8$-1$,-                                             ;
                                                                      9$-1$                                               ;
                                       50   7C 00021          CLRQ    R0                                                  ; 0315
                                       30   11 00023          BRB     10$                                                 ;
                         50            62   9E 00025 2$:      MOVAB   NMA$A_NODE_DSC, R0                                  ; 0302
                                       2B   11 00028          BRB     10$                                                 ;
                         50      18    A2   9E 0002A 3$:      MOVAB   NMA$A_LINE_DSC, R0                                  ;
                                       25   11 0002E          BRB     10$                                                 ;
                         50      34    A2   9E 00030 4$:      MOVAB   NMA$A_LOG_DSC, R0                                   ;
                                       1F   11 00034          BRB     10$                                                 ;
                         50      50    A2   9E 00036 5$:      MOVAB   NMA$A_OBJ_DSC, R0                                   ;
                                       19   11 0003A          BRB     10$                                                 ;
                         50      68    A2   9E 0003C 6$:      MOVAB   NMA$A_CIR_DSC, R0                                   ;
                                       13   11 00040          BRB     10$                                                 ;
                         50    0080    C2   9E 00042 7$:      MOVAB   NMA$A_X25_DSC, R0                                   ;
                                       0C   11 00047          BRB     10$                                                 ;
                         50    0098    C2   9E 00049 8$:      MOVAB   NMA$A_X29_DSC, R0                                   ;
                                       05   11 0004E          BRB     10$                                                 ;
                         50    00B0    C2   9E 00050 9$:      MOVAB   NMA$A_CNF_DSC, R0                                   ;
                      08 BC            50   D0 00055 10$:     MOVL    R0, @FILEDSC                                        ; 0300
                         50            51   D0 00059          MOVL    STATUS, R0                                          ; 0325
                                            04 0005C          RET                                                         ; 0327

; Routine Size:  93 bytes,    Routine Base:  $CODE$ + 0081
```

```
1 %SBTTL 'NMA$OPENFAB  Open or Create a File'
1 ROUTINE NMA$OPENFAB (FILEDSC, ACCESS) =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine does the actual open or create of a file.
1 !       First the fab is loaded wi.ı the correct attributes and then
1 !       a create or open service is done.  Create is used if the file
1 !       is to be opened with read-write access and the FOP CIF bit is
1 !       specified so that the file is created if it does not exist.
1 !       The created file will be indexed with a two byte binary key.
1 !       A rather large bucket size is used to allow for long records.
1 !       The protection is set to be read for world and group and the
1 !       UIC is set to the system.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       FILEDSC         Address of the filedescriptor for the fiιe
1 !       ACCESS          Value of the access parameter
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       Success or an RMS error
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !
1 !--
1
2    BEGIN
2
2    MAP                                      ! File descriptcr format
2       FILEDSC : REF BLOCK [1, BYTE] FIELD (FDSCFLDS);
2
2    LOCAL
2       STATUS,                               ! Return status
2       FAB,                                  ! Fab address
2       RAB,                                  ! Rab address
2       FNS,                                  ! Filename size
2       FNA;                                  ! Filename address
2
2    OWN
2       KEYXAB : $XABKEY_DECL,                ! Key xab for create
2       PROXAB : $XABPRO_DECL;                ! Protection xab for create
2
2    FNA = .FILEDSC [FDSCFNA];                ! Obtain descriptor fields
```

```
;   389          0385  2        FNS = .FILEDSC [FDSCFNS];
;   390          0386  2        FAB = .FILEDSC [FDSCFAB];
;   391          0387  2        RAB = .FILEDSC [FDSCRAB];
;   392          0388  2
;   393          0389  2        IF .ACCESS EQL NMA$C_OPN_AC_RW      ! Check access for read write
;   394          0390  2        THEN
;   395          0391  3            BEGIN
;   396          0392  3
;   397        P 0393  3            $FAB_INIT                       ! Initialize fab for create
;   398        P 0394  3                (
;   399        P 0395  3                FAB = .FAB,                 ! Fab address
;   400        P 0396  3                BKS = 9,                    ! Bucket size
;   401        P 0397  3                DNM = 'SYS$SYSTEM:.DAT',    ! Default filename string
;   402        P 0398  3                FAC = (UPD, PUT, GET, DEL), ! File access
;   403        P 0399  3                FNA = .FNA,                 ! Filename string address
;   404        P 0400  3                FNS = .FNS,                 ! Filename string size
;   405        P 0401  3                FOP = (CIF, MXV),           ! File open codes (create if, max ver)
;   406        P 0402  3                ORG = IDX,                  ! Organsization
;   407        P 0403  3                RFM = VAR,                  ! Record format
;   408        P 0404  3                SHR = (UPD, PUT, GET, DEL), ! Share
;   409        P 0405  3                XAB = PROXAB                ! Xab chain
;   410          0406  3                );
;   411          0407  3
;   412        P 0408  3            $XABKEY_INIT                    ! Initialize key xab
;   413        P 0409  3                (
;   414        P 0410  3                XAB = KEYXAB,               ! Xab address
;   415        P 0411  3                DTP = BN2,                  ! 2 byte binary
;   416        P 0412  3                POS0 = 0,                   ! Position
;   417        P 0413  3                SIZ0 = 2,                   ! Size
;   418        P 0414  3                KREF = 0                    ! Key reference (primary)
;   419          0415  3                );
;   420          0416  3
;   421        P 0417  3            $XABPRO_INIT                    ! Initialize protection xab
;   422        P 0418  3                (
;   423        P 0419  3                XAB = PROXAB,               ! Xab address
;   424        P 0420  3                UIC = (1, 4),               ! Uic of owner (system)
;   425        P 0421  3                PRO = (RWED, RWED, , ),     ! Protection (group and world no access)
;   426        P 0422  3                NXT = KEYXAB                ! Chain
;   427          0423  3                );
;   428          0424  3
;   429          0425  3            STATUS = $CREATE (FAB = .FAB);  ! Create the file if not found
;   430          0426  3
;   431          0427  3            END
;   432          0428  3
;   433          0429  2        ELSE
;   434          0430  2
;   435          0431  3            BEGIN
;   436        P 0432  3            $FAB_INIT                       ! Initialize the fab
;   437        P 0433  3                (
;   438        P 0434  3                FAB = .FAB,                 ! Fab address
;   439        P 0435  3                FAC = (GET),                ! File access
;   440        P 0436  3                FNA = .FNA,                 ! Filename string address
;   441        P 0437  3                FNS = .FNS,                 ! Filename string size
;   442        P 0438  3                DNM = 'SYS$SYSTEM:.DAT',    ! Default filename string
;   443        P 0439  3                SHR = (UPD, PUT, GET, DEL)  ! Share
;   444          0440  3                );
;   445          0441  3
;   446          0442  3            STATUS = $OPEN (FAB = .FAB);     ! Open the file
;   447          0443  3
;   448          0444  2            END;
;   449          0445  2
;   450          0446  2        IF NOT .STATUS                       ! Return failure status
;   451          0447  2        THEN
;   452          0448  2            RETURN .STATUS;
;   453          0449  2
;   454        P 0450  2        $RAB_INIT                            ! Initialize the rab
;   455        P 0451  2            (
;   456        P 0452  2            RAB = .RAB,                      ! Rab address
;   457        P 0453  2            FAB = .FAB,                      ! Fab address
;   458        P 0454  2            KBF = NMA$W_KEYBUF,              ! Key buffer address
;   459        P 0455  2            KRF = 0,                         ! Key of reference
;   460        P 0456  2            KSZ = 2,                         ! Key size
;   461        P 0457  2            RAC = KEY,                       ! Record access mode
;   462        P 0458  2            ROP = (UIF,KGE)                  ! Record options (put is update)
;   463          0459  2            );
;   464          0460  2
;   465          0461  2        RETURN $CONNECT (RAB = .RAB);        ! Connect record stream and return
;   466          0462  2
;   467          0463  1        END;



                                                                         .PSECT  $PLIT$,NOWRT,NOEXE,2

54  41  44  2E  3A  4D  45  54  53  59  53  24  53  59  53  000DC P.AAS:  .ASCII  \SYS$SYSTEM:.DAT\                               ;
54  41  44  2E  3A  4D  45  54  53  59  53  24  53  59  53  000EB P.AAT:  .ASCII  \SYS$SYSTEM:.DAT\                               ;

                                                                         .PSECT  $OWN$,NOEXE,2

                                                          004A4 KEYXAB:  .BLKB   76
                                                          004F0 PROXAB:  .BLKB   88

                                                                $RMS_PTR=            KEYXAB
                                                                $RMS_PTR=            PROXAB
                                                                         .EXTRN  SYS$CREATE, SYS$OPEN
                                                                         .EXTRN  SYS$CONNECT

                                                                         .PSECT  $CODE$,NOWRT,2

                                                     07FC 00000 NMA$OPENFAB:
                                                                         .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10              ; 0329
                                   5A 00000000'   00   9E 00002          MOVAB   PROXAB, R10
                                   50       04    AC   D0 00009          MOVL    FILEDSC, R0                                   ; 0384
                                   58       04    A0   D0 0000D          MOVL    4(R0), FNA
                                   59             60   D0 00011          MOVL    (R0), FNS                                     ; 0385
                                   56       08    A0   7D 00014          MOVQ    8(R0), FAB                                    ; 0386
                                   01       08    AC   D1 00018          CMPL    ACCESS, #1                                    ; 0389
                                                  03   13 0001C          BEQL    1$                                            ;
                                                0081   31 0001E          BRW     2$                                            ;
   0050    8F           00         6E             00   2C 00021 1$:      MOVC5   #0, (SP), #0, #80, (FAB)                      ; 0406
                                                  66      00028                                                                ;
                                   66     5003    8F   B0 00029          MOVW    #20483, (FAB)                                 ;
                               04  A6 02000002    8F   D0 0002E          MOVL    #33554434, 4(FAB)                             ;
                               16  A6     0F0F    8F   B0 00036          MOVW    #3855, 22(FAB)                                ;
```

```
                         1D    A6              20  90 0003C          MOVB    #32, 29(FAB)                         ;
                         1F    A6              02  90 00040          MOVB    #2, 31(FAB)                          ;
                         24    A6              6A  9E 00044          MOVAB   PROXAB, 36(FAB)                      ;
                         2C    A6              58  D0 00048          MOVL    FNA, 44(FAB)                         ;
                         30    A6 00000000'    00  9E 0004C          MOVAB   P.AAS, 48(FAB)                       ;
                         34    A6              59  90 00054          MOVB    FNS, 52(FAB)                         ;
                         35    A6              0F  90 00058          MOVB    #15, 53(FAB)                         ;
                         3E    A6              09  90 0005C          MOVB    #9, 62(FAB)                          ;
   004C   8F        00         6E              00  2C 00060          MOVC5   #0, (SP), #0, #76, $RMS_PTR          ; 0415
                                        B4     AA     00067                                                       ;
                         B4    AA     4C15     8F  B0 00069          MOVW    #19477, $RMS_PTR                     ;
                         C7    AA              02  90 0006F          MOVB    #2, $RMS_PTR+19                      ;
                         E2    AA              02  90 00073          MOVB    #2, $RMS_PTR+46                      ;
   0058   8F        00         6E              00  2C 00077          MOVC5   #0, (SP), #0, #88, $RMS_PTR          ; 0423
                                               6A     0007E                                                       ;
                               6A     5813     8F  B0 0007F          MOVW    #22547, $RMS_PTR                     ;
                         04    AA       B4     AA  9E 00084          MOVAB   KEYXAB, $RMS_PTR+4                   ;
                         08    AA     FF00     8F  B0 00089          MOVW    #-256, $RMS_PTR+8                    ;
                         0C    AA 00010004     8F  D0 0008F          MOVL    #6554 , $RMS_PTR+12                  ;
                                               56  DD 00097          PUSHL   FAB                                  ; 0425
                  00000000G    00              01  FB 00099          CALLS   #1, SYS$CREATE                       ;
                                               34  11 000A0          BRB     3$                                   ; 0389
   0050   8F        00         6E              00  2C 000A2 2$:      MOVC5   #0, (SP), #0, #80, (FAB)             ; 0440
                                               66     000A9                                                       ;
                               66     5003     8F  B0 000AA          MOVW    #20483, (FAB)                        ;
                         16    A6     0F02     8F  B0 000AF          MOVW    #3842, 22(FAB)                       ;
                         1F    A6              02  90 000B5          MOVB    #2, 31(FAB)                          ;
                         2C    A6              58  D0 000B9          MOVL    FNA, 44(FAB)                         ;
                         30    A6 00000000'    00  9E 000BD          MOVAB   P.AAT, 48(FAB)                       ;
                         34    A6              59  90 000C5          MOVB    FNS, 52(FAB)                         ;
                         35    A6              0F  90 000C9          MOVB    #15, 53(FAB)                         ;
                                               56  DD 000CD          PUSHL   FAB                                  ; 0442
                  00000000G    00              01  FB 000CF          CALLS   #1, SYS$OPEN                         ;
                               30              50  E9 000D6 3$:      BLBC    STATUS, 4$                           ; 0446
   0044   8F        00         6E              00  2C 000D9          MOVC5   #0, (SP), #0, #68, (RAB)             ; 0459
                                               67     000E0                                                       ;
                               67     4401     8F  B0 000E1          MOVW    #17409, (RAB)                        ;
                         04    A7 00200010     8F  D0 000E6          MOVL    #2097168, 4(RAB)                     ;
                         1E    A7              01  90 000EE          MOVB    #1, 30(RAB)                          ;
                         30    A7     FB10     CA  9E 000F2          MOVAB   NMA$W_KEYBUF, 48(RAB)                ;
                         34    A7              02  90 000F8          MOVB    #2, 52(RAB)                          ;
                         3C    A7              56  D0 000FC          MOVL    FAB, 60(RAB)                         ;
                                               57  DD 00100          PUSHL   RAB                                  ; 0461
                  00000000G    00              01  FB 00102          CALLS   #1, SYS$CONNECT                      ;
                                                   04 00109 4$:      RET                                          ; 0463

; Routine Size:  266 bytes,    Routine Base:  $CODE$ + 00DE
;  469          0464  1 %SBTTL 'NMA$CLOSEFILE  Close a specified file'
;  470          0465  1 GLOBAL ROUTINE NMA$CLOSEFILE (FILEID) =
;  471          0466  1
;  472          0467  1 !++
;  473          0468  1 ! FUNCTIONAL DESCRIPTION:
;  474          0469  1 !
;  475          0470  1 !       This routine closes a specified file or all the files.
;  476          0471  1 !
;  477          0472  1 ! FORMAL PARAMETERS:
;  478          0473  1 !
;  479          0474  1 !       FILEID          Value of the fileid parameter (NMA$C_OPN_xxxxx)
;  480          0475  1 !
;  481          0476  1 ! ROUTINE VALUE:
;  482          0477  1 ! COMPLETION CODES:
;  483          0478  1 !
;  484          0479  1 !       Status of last close operation.
;  485          0480  1 !
;  486          0481  1 !--
;  487          0482  1
;  488          0483  2 BEGIN
;  489          0484  2
;  490          0485  2 LOCAL
;  491          0486  2     FAB : REF BLOCK [1,BYTE],                       ! The fab for the file
;  492          0487  2     FILEDSC : REF BLOCK [1, BYTE]                   ! File descriptor
;  493          0488  2                 FIELD (FDSCFLDS),
;  494          0489  2     STATUS;                                         ! Status return
;  495          0490  2
;  496          0491  2 STATUS = NMA$_SUCCESS;
;  497          0492  2 IF NMA$SELECTFILE (.FILEID, FILEDSC) THEN       ! Obtain descriptor address
;  498          0493  3     BEGIN
;  499          0494  3     FAB = .FILEDSC [FDSCFAB];               ! Get address of FAB
;  500          0495  3     IF .FAB [FAB$W_IFI] NEQ 0 THEN                  ! If file isn't closed, do it.
;  501          0496  4         BEGIN
;  502          0497  4         STATUS =
;  503          0498  4             $CLOSE (FAB = .FILEDSC [FDSCFAB]); ! Call RMS to close the file
;  504          0499  4         IF .STATUS THEN
;  505          0500  4             NML$LOGFILEOP (DBG$C_FILEIO,
;  506          0501  4                             .FILEID,
;  507          0502  4                             $ASCID ('file closed.'));
;  508          0503  3         END;
;  509          0504  3     END
;  510          0505  2 ELSE
;  511          0506  2     STATUS = NMA$_BADFID;
;  512          0507  2 RETURN .STATUS
;  513          0508  2
;  514          0509  1 END;


                                                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

         2E  64  65  73  6F  6C  63  20  65  6C  69  66  000FA P.AAV:   .ASCII  \file closed.\                           ;
                                                         00106          .BLKB   2
                                              0000000C  00108 P.AAU:    .LONG   12                                      ;
                                              00000000' 0010C           .ADDRESS P.AAV                                  ;

                                                                        .EXTRN  SYS$CLOSE
```

```
                                                                     .PSECT  $CODE$,NOWRT,2

                                              0004 00000             .ENTRY  NMA$CLOSEFILE, Save R2                        ; 0465
                          5E               04   C2 00002             SUBL2   #4, SP                                        ;
                          52               01   D0 00005             MOVL    #1, STATUS                                    ; 0491
                                           5E   DD 00008             PUSHL   SP                                            ; 0492
                                    04     AC   DD 0000A             PUSHL   FILEID                                        ;
                    FE87  CF               02   FB 0000D             CALLS   #2, NMA$SELECTFILE                            ;
                          30               50   E9 00012             BLBC    R0, 1$                                        ;
                          50               6E   D0 00015             MOVL    FILEDSC, R0                                   ; 0494
                          51        08     A0   D0 00018             MOVL    8(R0), FAB                                    ;
                                    02     A1   B5 0001C             TSTW    2(FAB)                                        ; 0495
                                           26   13 0001F             BEQL    2$                                            ;
                                    08     A0   DD 00021             PUSHL   8(R0)                                         ; 0498
               00000000G  00               01   FB 00024             CALLS   #1, SYS$CLOSE                                 ;
                          52               50   D0 0002B             MOVL    R0, STATUS                                    ;
                          16               52   E9 0002E             BLBC    STATUS, 2$                                    ; 0499
                             00000000'     00   9F 00031             PUSHAB  P.AAU                                         ; 0502
                                    04     AC   DD 00037             PUSHL   FILEID                                        ; 0501
                                           01   DD 0003A             PUSHL   #1                                            ; 0500
               00000000G  00               03   FB 0003C             CALLS   #3, NML$LOGFILEOP                             ;
                                           02   11 00043             BRB     2$                                            ; 0492
                                           52   D4 00045 1$:         CLRL    STATUS                                        ; 0506
                          50               52   D0 00047 2$:         MOVL    STATUS, R0                                    ; 0507
                                                04 0004A             RET                                                   ; 0509

; Routine Size:  75 bytes,    Routine Base:  $CODE$ + 01E8
```

```
1 %SBTTL 'NMA$MATCHREC  Find a Record in a File'
1 GLOBAL ROUTINE NMA$MATCHREC (FILEID, BUFDSC, KEYADR, FIELDCODE,
1                                  FIELDSIZE, FIELDADR, RTNDSC) =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine searches a database for a record containing a given
1 !       field containing given data.  Degenerate cases are provided for
1 !       returning all records, or all records containing a specific field.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       FILEID          Value of the fileid code (NMA$C_OPN_xxxxx)
1 !       BUFDSC          Address of a descriptor of a buffer to use
1 !       KEYADR          Address of a word containing the key to start reading
1 !                       Key value is returned in this word.
1 !       FIELDCODE       Value of the field code (zero for wildcard)*****
1 !       FIELDSIZE       Value of the field size (zero for wildcard)
1 !       FIELDADR        Address of the field data
1 !       RTNDSC          Address of a descriptor to return descriptor of data
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES.
1 !
1 !       NMA or RMS error status
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !
1 !--
1
2     BEGIN
2
2     MAP
2         BUFDSC : REF VECTOR,              ! Buffer to use for record
2         RTNDSC : REF VECTOR;              ! Return data descriptor
2
2     LOCAL
2         FILEDSC : REF BLOCK [1, BYTE]    ! File descriptor
2                   FIELD (FDSCFLDS),
2         RAB     : REF BLOCK [1, BYTE],   ! The rab for the file
2         LCLDSC  : VECTOR [2],             ! A local data descriptor
2         FAB     : REF BLOCK [, BYTE],    ! The fab for the file
2         FLDADR,                           ! Field address
2         FLDSIZ,                           ! Field size
2         STATUS;                           ! Status return
2
```

```
2       EXTERNAL ROUTINE
2           NMA$SEARCHFLD;                      ! Search for a field value
2
2       STATUS = NMA$SELECTFILE (.FILEID,
2                                FILEDSC);      ! Obtain the file descriptor
2
2       IF NOT .STATUS
2       THEN
2           RETURN .STATUS;                     ! Bogus fileid
2
2       RAB = .FILEDSC [FDSCRAB];               ! Point to the rab
2       FAB = .FILEDSC [FDSCFAB];               ! Get address of FAB
2
2       IF .FAB [FAB$W_IFI] EQL 0               ! If file not open,
2       THEN
2           RETURN .FAB [FAB$L_STS];            ! return open failure status
2
2       RAB [RAB$W_USZ] = .BUFDSC [0];          ! Set the buffer to use
2       RAB [RAB$L_UBF] = .BUFDSC [1];
2
2       NMA$W_KEYBUF = ..KEYADR;                ! And the key value to use
2
2       WHILE 1                                 ! Try this forever
2       DO
3           BEGIN
3
3           STATUS = $GET (RAB = .RAB);         ! Read a record
3
3           LCLDSC [0] = .RAB [RAB$W_RSZ];  ! Pickup the real record descriptor
3           LCLDSC [1] = .RAB [RAB$L_RBF];
3           RTNDSC [0] = .RAB [RAB$W_RSZ] - NML$K_PERM_KEYS_LEN;
3           RTNDSC [1] = .RAB [RAB$L_RBF] + NML$K_PERM_KEYS_LEN;
3
3           IF NOT .STATUS                      ! If no good, return
3           THEN
3               RETURN .STATUS;
3
3           NMA$W_KEYBUF =                      ! Set the keyvalue returned
3                   .(.LCLDSC [1]) <0, 16, 0>;
3
3           (.KEYADR) <0, 16, 0> = .NMA$W_KEYBUF; ! Return for user to remember
3
3           FLDADR = 0;                         ! Start search from beginning
3           IF NMA$SEARCHFLD                    ! Look for the field
3                   (
3                   .RTNDSC,                    ! Here is the data
3                   .FIELDCODE,                 ! Value of the code to look for
3                   FLDSIZ,                     ! Return the size here
3                   FLDADR                      ! Return the address here
3                   )
3           THEN
4               BEGIN
4
4               IF .FIELDSIZE EQL 0             ! Wildcard
4               THEN
5                   BEGIN
5
```

```
:   630          0624  5                 STATUS = NMA$_SUCCESS;  ! It always succeeds
:   631          0625  5                 EXITLOOP;
:   632          0626  5
:   633          0627  4                 END;
:   634          0628  4
:   635          0629  4             IF CH$EQL                   ! Look at the data
:   636          0630  4                 (
:   637          0631  4                 .FLDSIZ,                ! Data in record
:   638          0632  4                 .FLDADR,
:   639          0633  4                 .FIELDSIZE,             ! User data
:   640          0634  4                 .FIELDADR,
:   641          0635  4                 0
:   642          0636  4                 )
:   643          0637  4             THEN
:   644          0638  5                 BEGIN
:   645          0639  5
:   646          0640  5                 STATUS = NMA$_SUCCESS;  ! We found such a record
:   647          0641  5                 EXITLOOP;
:   648          0642  5
:   649          0643  4                 END;
:   650          0644  3             END;
:   651          0645  3
:   652          0646  3         NMA$W_KEYBUF = .NMA$W_KEYBUF + 1;      ! Increment key ****
:   653          0647  3         (.KEYADR) <0, 16, 0> = .NMA$W_KEYBUF; ! Return for user to remember
:   654          0648  3
:   655          0649  2         END;
:   656          0650  2
:   657          0651  2     IF .STATUS
:   658          0652  2     THEN
:   659          0653  2         NML$LOGRECORDOP (DBG$C_FILEIO,
:   660          0654  2                          .FILEID,
:   661          0655  2                         $ASCID ('record matched'),
:   662          0656  2                         LCLDSC);
:   663          0657  2
:   664          0658  2     RETURN .STATUS
:   665          0659  2
:   666          0660  1     END;


                                                                          .PSECT  $PLIT$,NOWRT,NOEXE,2

  64  65  68  63  74  61  6D  20  64  72  6F  63  65  72  00110 P.AAX:   .ASCII  \record matched\                                 ;
                                                          0011E          .BLKB   2
                                              0000000E  00120 P.AAW:   .LONG   14                                               ;
                                              00000000' 00124          .ADDRESS P.AAX                                           ;

                                                                          .EXTRN  NMA$SEARCHFLD, SYS$GET

                                                                          .PSECT  $CODE$,NOWRT,2

                                                    00FC  00000          .ENTRY  NMA$MATCHREC, Save R2,R3,R4,R5,R6,R7             ; 0511
                                57 00000000'    00   9E  00002          MOVAB   NMA$W_KEYBUF, R7                                 ;
                                5E              14   C2  00009          SUBL2   #20, SP                                          ;
                                                5E   DD  0000C          PUSHL   SP                                               ; 0570
                                           04   AC   DD  0000E          PUSHL   FILEID                                           ;
                         FE38   CF              02   FB  00011          CALLS   #2, NMA$SELECTFILE                               ;
```

```
                                    56         50  D0 00016          MOVL    R0, STATUS                                    ;
                                    4F         56  E9 00019          BLBC    STATUS, 3$                                    ; 0573
                                    50         6E  D0 0001C          MOVL    FILEDSC, R0                                   ; 0577
                                    54     0C  A0  D0 0001F          MOVL    12(R0), RAB                                   ;
                                    50     08  A0  D0 00023          MOVL    8(R0), FAB                                    ; 0578
                                           02  A0  B5 00027          TSTW    2(FAB)                                        ; 0580
                                               05  12 0002A          BNEQ    1$                                            ;
                                    50     08  A0  D0 0002C          MOVL    8(FAB), R0                                    ; 0582
                                                   04 00030          RET                                                   ;
                                    50     08  AC  D0 00031  1$:     MOVL    BUFDSC, R0                                    ; 0584
                              20    A4         60  B0 00035          MOVW    (R0), 32(RAB)                                 ;
                              24    A4     04  A0  D0 00039          MOVL    4(R0), 36(RAB)                                ; 0585
                                    67     0C  BC  B0 0003E          MOVW    @KEYADR, NMA$W_KEYBUF                         ; 0587
                                    55     1C  AC  D0 00042          MOVL    RTNDSC, R5                                    ; 0598
                                               54  DD 00046  2$:     PUSHL   RAB                                           ; 0593
                         00000000G  00         01  FB 00048          CALLS   #1, SYS$GET                                   ;
                                    56         50  D0 0004F          MOVL    R0, STATUS                                    ;
                              0C    AE     22  A4  3C 00052          MOVZWL  34(RAB), LCLDSC                               ; 0595
                              10    AE     28  A4  D0 00057          MOVL    40(RAB), LCLDSC+4                             ; 0596
                              1C    BC     22  A4  3C 0005C          MOVZWL  34(RAB), @RTNDSC                              ; 0597
                              1C    BC         02  C2 00061          SUBL2   #2, @RTNDSC                                   ;
                    04    A5  28    A4         02  C1 00065          ADDL3   #2, 40(RAB), 4(R5)                            ; 0598
                                    57         56  E9 0006B  3$:     BLBC    STATUS, 7$                                    ; 0600
                                    67     10  BE  B0 0006E          MOVW    @LCLDSC+4, NMA$W_KEYBUF                       ; 0605
                              0C    BC         67  B0 00072          MOVW    NMA$W_KEYBUF, @KEYADR                         ; 0607
                                           04  AE  D4 00076          CLRL    FLDADR                                        ; 0609
                                           04  AE  9F 00079          PUSHAB  FLDADR                                        ; 0611
                                           0C  AE  9F 0007C          PUSHAB  FLDSIZ                                        ;
                                           10  AC  DD 0007F          PUSHL   FIELDCODE                                     ; 0613
                                           1C  AC  DD 00082          PUSHL   RTNDSC                                        ; 0612
                         00000000G  00         04  FB 00085          CALLS   #4, NMA$SEARCHFLD                             ;
                                    16         50  E9 0008C          BLBC    R0, 5$                                        ;
                                           14  AC  D5 0008F          TSTL    FIELDSIZE                                     ; 0620
                                               0C  13 00092          BEQL    4$                                            ;
    14    AC          00      04    BE     08  AE  2D 00094          CMPC5   FLDSIZ, @FLDADR, #0, FIELDSIZE, @FIELDADR     ; 0630
                                           18  BC     0009C                                                                ;
                                               05  12 0009E          BNEQ    5$                                            ;
                                    56         01  D0 000A0  4$:     MOVL    #1, STATUS                                    ; 0640
                                               08  11 000A3          BRB     6$                                            ; 0638
                                               67  B6 000A5  5$:     INCW    NMA$W_KEYBUF                                  ; 0646
                              0C    BC         67  B0 000A7          MOVW    NMA$W_KEYBUF, @KEYADR                         ; 0647
                                               99  11 000AB          BRB     2$                                            ; 0589
                                    15         56  E9 000AD  6$:     BLBC    STATUS, 7$                                    ; 0651
                                           0C  AE  9F 000B0          PUSHAB  LCLDSC                                        ; 0653
                                    00000000'  00  9F 000B3          PUSHAB  P.AAW                                         ; 0655
                                           04  AC  DD 000B9          PUSHL   FILEID                                        ; 0654
                                               01  DD 000BC          PUSHL   #1                                            ; 0653
                         00000000G  00         04  FB 000BE          CALLS   #4, NML$LOGRECORDOP                           ;
                                    50         56  D0 000C5  7$:     MOVL    STATUS, R0                                    ; 0658
                                               04  000C8             RET                                                   ; 0660

; Routine Size:  201 bytes,    Routine Base:  $CODE$ + 0233
```

```
1 %SBTTL 'NMA$READREC  Get a record from a File'
1 GLOBAL ROUTINE NMA$READREC (FILEID, KEYADR, BUFDSC, RTNDSC) =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine reads the next database record starting at the specified
1 !       key.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       FILEID          Value of the fileid code (NMA$C_OPN_xxxxx)
1 !       KEYADR          Address of a word containing the key to start reading
1 !                       Key value is returned in this word.
1 !       BUFDSC          Address of a descriptor of a buffer to use
1 !       RTNDSC          Address of a descriptor to return descriptor of data
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       NMA or RMS error status
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !
1 !--
1
2     BEGIN
2
2     MAP
2       BUFDSC : REF VECTOR,              ! Buffer to use for record
2       RTNDSC : REF VECTOR;              ! Return data descriptor
2
2     LOCAL
2       FILEDSC : REF BLOCK [1, BYTE]     ! File descriptor
2                 FIELD (FDSCFLDS),
2       FAB     : REF BLOCK [, BYTE],     ! The fab for the file
2       RAB     : REF BLOCK [1, BYTE],    ! The rab for the file
2       LCLDSC  : VECTOR [2],
2       STATUS;                           ! Status return
2
2     STATUS = NMA$SELECTFILE (.FILEID,
2                              FILEDSC);  ! Obtain the file descriptor
2
2     IF NOT .STATUS
2     THEN
2       RETURN .STATUS;                   ! Bogus fileid
2
```

```
0718  2
0719  2      RAB = .FILEDSC [FDSCRAB];               ! Point to the rab
0720  2      FAB = .FILEDSC [FDSCFAB];               ! Get address of FAB
0721  2
0722  2      IF .FAB [FAB$W_IFI] EQL 0               ! If file not open,
0723  2      THEN
0724  2          RETURN .FAB [FAB$L_STS];            ! Return open failure status
0725  2
0726  2      RAB [RAB$W_USZ] = .BUFDSC [0];          ! Set the buffer to use
0727  2      RAB [RAB$L_UBF] = .BUFDSC [1];
0728  2
0729  2      NMA$W_KEYBUF = ..KEYADR;                ! And the key value to use
0730  2
0731  2      STATUS = $GET (RAB = .RAB);             ! Read a record
0732  2
0733  2      RTNDSC [0] = .RAB [RAB$W_RSZ] - NML$K_PERM_KEYS_LEN;
0734  2      RTNDSC [1] = .RAB [RAB$L_RBF] + NML$K_PERM_KEYS_LEN;
0735  2
0736  2      IF NOT .STATUS                          ! If no good, return
0737  2      THEN
0738  2          RETURN .STATUS;
0739  2
0740  2      LCLDSC [0] = .RAB [RAB$W_RSZ];
0741  2      LCLDSC [1] = .RAB [RAB$L_RBF];
0742  2
0743  2      (.KEYADR)<0,16,0> = .(.LCLDSC [1])<0,16>; ! Return for user to remember
0744  2
0745  2      NML$LOGRECORDOP (DBG$C_FILEIO,
0746  2                       .FILEID,
0747  2                       $ASCID ('record read'),
0748  2                       LCLDSC);
0749  2
0750  2      RETURN NMA$_SUCCESS
0751  2
0752  1      END;



                                                                   .PSECT  $PLIT$,NOWRT,NOEXE,2

64  61  65  72  20  64  72  6F  63  65  72  00128 P.AAZ:   .ASCII  \record read\                                   ;
                                            00133          .BLKB   1
                                  0000000B  00134 P.AAY:   .LONG   11                                              ;
                                  00000000' 00138          .ADDRESS P.AAZ                                          ;



                                                                   .PSECT  $CODE$,NOWRT,2

                                      0004  00000          .ENTRY  NMA$READREC, Save R2                            ; 0662
                5E                  0C  C2  00002          SUBL2   #12, SP                                         ;
                                    5E  DD  00005          PUSHL   SP                                              ; 0711
                           04       AC  DD  00007          PUSHL   FILEID                                          ;
          FD76  CF                  02  FB  0000A          CALLS   #2, NMA$SELECTFILE                              ;
                6A                  50  E9  0000F          BLBC    STATUS, 2$                                      ; 0714
                51                  6E  D0  00012          MOVL    FILEDSC, R1                                     ; 0719
                51         08       A1  7D  00015          MOVQ    8(R1), FAB                                      ; 0720
```

```
                                02    A1   B5 00019         TSTW    2(FAB)                              ; 0722
                                      05   12 0001C         BNEQ    1$                                  ;
                      50        08    A1   D0 0001E         MOVL    8(FAB), R0                          ; 0724
                                           04 00022         RET                                         ;
                      51        0C    AC   D0 00023 1$:     MOVL    BUFDSC, R1                          ; 0726
                 20   A2              61   B0 00027         MOVW    (R1), 32(RAB)                       ;
                 24   A2        04    A1   D0 0002B         MOVL    4(R1), 36(RAB)                      ; 0727
           00000000'  00        08    BC   B0 00030         MOVW    @KEYADR, NMA$W_KEYBUF               ; 0729
                                      52   DD 00038         PUSHL   RAB                                 ; 0731
           00000000G  00              01   FB 0003A         CALLS   #1, SYS$GET                         ;
                      51        10    AC   D0 00041         MOVL    RTNDSC, R1                          ; 0733
                      61        22    A2   3C 00045         MOVZWL  34(RAB), (R1)                       ;
                      61              02   C2 00049         SUBL2   #2, (R1)                            ;
     04    A1         28   A2         02   C1 0004C         ADDL3   #2, 40(RAB), 4(R1)                  ; 0734
                      27              50   E9 00052         BLBC    STATUS, 2$                          ; 0736
                 04   AE        22    A2   3C 00055         MOVZWL  34(RAB), LCLDSC                     ; 0740
                 08   AE        28    A2   D0 0005A         MOVL    40(RAB), LCLDSC+4                   ; 0741
                 08   BC        08    BE   B0 0005F         MOVW    @LCLDSC+4, @KEYADR                  ; 0743
                                04    AE   9F 00064         PUSHAB  LCLDSC                              ; 0745
                         00000000'    00   9F 00067         PUSHAB  P.AAY                               ; 0747
                                04    AC   DD 0006D         PUSHL   FILEID                              ; 0746
                                      01   DD 00070         PUSHL   #1                                  ; 0745
           00000000G  00              04   FB 00072         CALLS   #4, NML$LOGRECORDOP                 ;
                      50              01   D0 00079         MOVL    #1, R0                              ; 0750
                                           04 0007C 2$:     RET                                         ; 0752

; Routine Size:  125 bytes,    Routine Base:  $CODE$ + 02FC
```

```
1 %SBTTL 'NMA$WRITEREC  Write a Record to a File'
1 GLOBAL ROUTINE NMA$WRITEREC (FILEID, KEYADR, BUFDSC) =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine puts a record to the specified file.  The key is
1 !       specified by keyadr.  The file was opened so that puts to existing
1 !       records act as updates.  The keyvalue is moved to the first two bytes
1 !       of the record before the write.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       FILEID          Value if the fileid
1 !       KEYADR          Address of a word of keyvalue
1 !       BUFDSC          Address of descriptor of data to write
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       RMS error code
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !
1 !--
1
2     BEGIN
2
2     MAP
2         BUFDSC : REF VECTOR;            ! User supplied data
2
2     LOCAL
2         RAB     : REF BLOCK [1, BYTE],  ! Address of rab
2         STATUS,                         ! Return status
2         FILEDSC : REF BLOCK [1, BYTE]   ! File descriptor address
2                   FIELD (FDSCFLDS),
2         LCLDSC  : VECTOR [2];
2
2     STATUS = NMA$SELECTFILE (.FILEID,
2                              FILEDSC);  ! Obtain file descriptor
2     IF NOT .STATUS
2     THEN
2         RETURN .STATUS;                 ! Return the status
2
2     RAB = .FILEDSC [FDSCRAB];           ! Obtain the rab address
2     LCLDSC [0] = .BUFDSC [0] + NML$K_PERM_KEYS_LEN;
2     LCLDSC [1] = .BUFDSC [1] - NML$K_PERM_KEYS_LEN;
```

```
           0810  2      RAB [RAB$W_RSZ] = .LCLDSC [0];        ! User buffer to write
           0811  2      RAB [RAB$L_RBF] = .LCLDSC [1];
           0812  2
           0813  2      NMA$W_KEYBUF = ..KEYADR;              ! Key value from user
           0814  2      (.LCLDSC [1])<0,16,0> = .NMA$W_KEYBUF; ! Move key to buffer for write
           0815  2
           0816  2      STATUS = $PUT (RAB = .RAB);           ! Put or update the record
           0817  2
           0818  2      IF .STATUS
           0819  2      THEN
           0820  2          NML$LOGRECORDOP (DBG$C_FILEIO,
           0821  2                           .FILEID,
           0822  2                           $ASCID ('record written'),
           0823  2                           LCLDSC);
           0824  2
           0825  2      RETURN .STATUS
           0826  2
           0827  1      END;



                                                              .PSECT  $PLIT$,NOWRT,NOEXE,2

  6E  65  74  74  69  72  77  20  64  72  6F  63  65  72  0013C P.ABB:  .ASCII  \record written\                         ;
                                                          0014A         .BLKB   2
                                            0000000E  0014C P.ABA:  .LONG   14                                       ;
                                            00000000' 00150         .ADDRESS P.ABB                                   ;

                                                                    .EXTRN  SYS$PUT

                                                                    .PSECT  $CODE$,NOWRT,2

                                               000C 00000           .ENTRY  NMA$WRITEREC, Save R2,R3                 ; 0754
                            53 00000000'   00   9E 00002            MOVAB   NMA$W_KEYBUF, R3                         ;
                            5E             0C   C2 00009            SUBL2   #12, SP                                  ;
                                           5E   DD 0000C            PUSHL   SP                                       ; 0801
                                     04    AC   DD 0000E            PUSHL   FILEID                                   ;
                     FCF2   CF             02   FB 00011            CALLS   #2, NMA$SELECTFILE                       ;
                            52             50   D0 00016            MOVL    R0, STATUS                               ;
                            4C             52   E9 00019            BLBC    STATUS, 1$                               ; 0803
                            50             6E   D0 0001C            MOVL    FILEDSC, R0                              ; 0807
                            51       0C    A0   D0 0001F            MOVL    12(R0), RAB                              ;
                            50       0C    AC   D0 00023            MOVL    BUFDSC, R0                               ; 0808
          04    AE          60             02   C1 00027            ADDL3   #2, (R0), LCLDSC                         ;
          08    AE     04   A0             02   C3 0002C            SUBL3   #2, 4(R0), LCLDSC+4                      ; 0809
                       22   A1       04    AE   B0 00032            MOVW    LCLDSC, 34(RAB)                          ; 0810
                       28   A1       08    AE   D0 00037            MOVL    LCLDSC+4, 40(RAB)                        ; 0811
                            63       08    BC   B0 0003C            MOVW    @KEYADR, NMA$W_KEYBUF                    ; 0813
                       08   BE             63   B0 00040            MOVW    NMA$W_KEYBUF, @LCLDSC+4                  ; 0814
                                           51   DD 00044            PUSHL   RAB                                      ; 0816
               00000000G    00             01   FB 00046            CALLS   #1, SYS$PUT                              ;
                            52             50   D0 0004D            MOVL    R0, STATUS                               ;
                            15             52   E9 00050            BLBC    STATUS, 1$                               ; 0818
                                     04    AE   9F 00053            PUSHAB  LCLDSC                                   ; 0820
                               00000000'   00   9F 00056            PUSHAB  P.ABA                                    ; 0822
                                     04    AC   DD 0005C            PUSHL   FILEID                                   ; 0821
                                           01   DD 0005F            PUSHL   #1                                       ; 0820
```

NML
V04

```
                    00000000G  00            04  FB 00061        CALLS    #4, NML$LOGRECORDOP                    ;
                               50            52  D0 00068 1$:    MOVL     STATUS, R0                             ; 0825
                                                 04 0006B        RET                                             ; 0827

; Routine Size:  108 bytes,    Routine Base:  $CODE$ + 0379
```

NML
V04

; R

```
%SBTTL 'NMA$DELETEREC  Delete a Record from the File'
GLOBAL ROUTINE NMA$DELETEREC (FILEID, KEYADR) =

!++
! FUNCTIONAL DESCRIPTION:
!
!       This routine deletes a record from the file by specified key
!       number.
!
! FORMAL PARAMETERS:
!
!       FILEID          Value if the fileid
!       KEYADR          Address of a word of keyvalue
!
! IMPLICIT INPUTS:
!
!       NONE
!
! IMPLICIT OUTPUTS:
!
!       NONE
!
! ROUTINE VALUE:
! COMPLETION CODES:
!
!       RMS error code
!
! SIDE EFFECTS:
!
!       NONE
!
!--

    BEGIN

    LOCAL
        RAB     : REF BLOCK [1, BYTE],  ! Address of rab
        STATUS,                         ! Return status
        FILEDSC : REF BLOCK [1, BYTE]   ! File descriptor address
                  FIELD (FDSCFLDS);

    STATUS = NMA$SELECTFILE (.FILEID,
                             FILEDSC);  ! Obtain file descriptor

    IF .STATUS
    THEN
        BEGIN

        RAB = .FILEDSC [FDSCRAB];       ! Obtain the rab address

        NMA$W_KEYBUF = ..KEYADR;        ! Key value from user

        STATUS = $DELETE (RAB = .RAB);  ! Delete the record

        IF .STATUS
        THEN
            NML$LOGRECORDOP (DBG$C_FILEIO,
;  894          0885  3                                        .FILEID,
;  895          0886  3                                        $ASCID ('record deleted'),
;  896          0887  3                                        UPLIT (2, NMA$W_KEYBUF));
;  897          0888  3
;  898          0889  2              END;
;  899          0890  2
;  900          0891  2          RETURN .STATUS
;  901          0892  2
;  902          0893  1          END;



                                                                                 .PSECT  $PLIT$,NOWRT,NOEXE,2

     64  65  74  65  6C  65  64  20  64  72  6F  63  65  72  00154 P.ABD:   .ASCII  \record deleted\                                      ;
                                                             00162          .BLKB   2
                                                   0000000E  00164 P.ABC:   .LONG   14                                                    ;
                                                   00000000' 00168          .ADDRESS P.ABD                                                ;
                                                   00000002  0016C P.ABE:   .LONG   2                                                     ;
                                                   00000000' 00170          .ADDRESS NMA$W_KEYBUF                                         ;

                                                                            .EXTRN  SYS$DELETE

                                                                            .PSECT  $CODE$,NOWRT,2

                                                        0004 00000          .ENTRY  NMA$DELETEREC, Save R2                                ; 0829
                                      5E               04  C2 00002          SUBL2   #4, SP                                                ;
                                                       5E  DD 00005          PUSHL   SP                                                    ; 0869
                                                04     AC  DD 00007          PUSHL   FILEID                                                ;
                               FC8D   CF               02  FB 0000A          CALLS   #2, NMA$SELECTFILE                                    ;
                                      52               50  D0 0000F          MOVL    R0, STATUS                                            ;
                                      36               52  E9 00012          BLBC    STATUS, 1$                                            ; 0872
                                      50               6E  D0 00015          MOVL    FILEDSC, R0                                           ; 0876
                                      50        0C     A0  D0 00018          MOVL    12(R0), RAB                                           ;
                          00000000'   00        08     BC  B0 0001C          MOVW    @KEYADR, NMA$W_KEYBUF                                 ; 0878
                                                       50  DD 00024          PUSHL   RAB                                                   ; 0880
                          00000000G   00               01  FB 00026          CALLS   #1, SYS$DELETE                                        ;
                                      52               50  D0 0002D          MOVL    R0, STATUS                                            ;
                                      18               52  E9 00030          BLBC    STATUS, 1$                                            ; 0882
                                          00000000'    00  9F 00033          PUSHAB  P.ABE                                                 ; 0887
                                          00000000'    00  9F 00039          PUSHAB  P.ABC                                                 ; 0886
                                                04     AC  DD 0003F          PUSHL   FILEID                                                ; 0885
                                                       01  DD 00042          PUSHL   #1                                                    ; 0884
                          00000000G   00               04  FB 00044          CALLS   #4, NML$LOGRECORDOP                                   ;
                                      50               52  D0 0004B 1$:      MOVL    STATUS, R0                                            ; 0891
                                                           04 0004E          RET                                                           ; 0893

; Routine Size:  79 bytes,    Routine Base:  $CODE$ + 03E5
;  904          0894  1 END                                   . End of module
;  905          0895  1
;  906          0896  0 ELUDOM



;                                   PSECT SUMMARY
;
;       Name                     Bytes                        Attributes
;
;  $OWN$                          1352  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $PLIT$                          372  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $CODE$                         1076  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         3         0         27         00:00.1
;  _$255$DUA28:[SHRLIB]NMALIBRY.L32;1          887        14         1         47         00:00.2
;  _$255$DUA28:[SYSLIB]STARLET.L32;1          9776       141         1        581         00:02.2




;                                 COMMAND QUALIFIERS

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

; Size:          1076 code + 1724 data bytes
; Run Time:         00:30.1
; Elapsed Time:     01:12.0
; Lines/CPU Min:     1784
; Lexemes/CPU-Min: 31149
; Memory Used:  196 pages
; Compilation Complete
```

NML
V04

; R

0280 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

CNFSEND
LIS

NPADEF
MDL

CNFSTORE
LIS

NMLSHR
MAP

NMAHEAD
B32

NPAMAC
MAR

CNFSHOW
LIS

CNFWQDEF
LIS

NMATAIL
B32

NML

NMLDDL
B32

NML
MAP

NMLDEF
MDL

NMAFILES
LIS

NMAFIELDS
LIS

CNFWORKQ
LIS

0281 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY