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
```

```
**FILE**ID**NMLLISPRM                                K 2
```

```
NN      NN  MM      MM  LL          LL            IIIIII      SSSSSSSS  PPPPPPPP    RRRRRRRR    MM      MM
NN      NN  MM      MM  LL          LL            IIIIII      SSSSSSSS  PPPPPPPP    RRRRRRRR    MM      MM
NN      NN  MMMM  MMMM  LL          LL              II      SS          PP      PP  RR      RR  MMMM  MMMM
NN      NN  MMMM  MMMM  LL          LL              II      SS          PP      PP  RR      RR  MMMM  MMMM
NNNN    NN  MM  MM  MM  LL          LL              II      SS          PP      PP  RR      RR  MM  MM  MM
NNNN    NN  MM  MM  MM  LL          LL              II      SS          PP      PP  RR      RR  MM  MM  MM
NN  NN  NN  MM      MM  LL          LL              II        SSSSSS    PPPPPPPP    RRRRRRRR    MM      MM
NN  NN  NN  MM      MM  LL          LL              II        SSSSSS    PPPPPPPP    RRRRRRRR    MM      MM
NN    NNNN  MM      MM  LL          LL              II              SS  PP          RR  RR      MM      MM
NN    NNNN  MM      MM  LL          LL              II              SS  PP          RR  RR      MM      MM
NN      NN  MM      MM  LL          LL              II              SS  PP          RR    RR    MM      MM      ....
NN      NN  MM      MM  LL          LL              II              SS  PP          RR    RR    MM      MM      ....
NN      NN  MM      MM  LLLLLLLLLL  LLLLLLLLLL    IIIIII    SSSSSSSS    PP          RR      RR  MM      MM      ....
NN      NN  MM      MM  LLLLLLLLLL  LLLLLLLLLL    IIIIII    SSSSSSSS    PP          RR      RR  MM      MM      ....


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
```

NM
VO

```
0001  0 %TITLE 'NML special parameter handling routines'
0002  0 MODULE NML$LISPRM (
0003  0                 LANGUAGE (BLISS32),
0004  0                 ADDRESSING_MODE (NONEXTERNAL=GENERAL),
0005  0                 ADDRESSING_MODE (EXTERNAL=GENERAL),
0006  0                 IDENT = 'V04-000'
0007  0                 ) =
0008  1 BEGIN
0009  1 !
0010  1 !****************************************************************************
0011  1 !*                                                                          *
0012  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
0013  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
0014  1 !*  ALL RIGHTS RESERVED.                                                    *
0015  1 !*                                                                          *
0016  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  *
0017  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
0018  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER  *
0019  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  *
0020  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY  *
0021  1 !*  TRANSFERRED.                                                            *
0022  1 !*                                                                          *
0023  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE  *
0024  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT  *
0025  1 !*  CORPORATION.                                                            *
0026  1 !*                                                                          *
0027  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS  *
0028  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
0029  1 !*                                                                          *
0030  1 !*                                                                          *
0031  1 !****************************************************************************
0032  1 !
0033  1
0034  1 !++
0035  1 ! FACILITY:  DECnet-VAX V2.0 Network Management Listener
0036  1 !
0037  1 ! ABSTRACT:
0038  1 !
0039  1 !       This module contains action routines to handle changing and
0040  1 !       displaying of permanent data base entity parameters.
0041  1 !
0042  1 ! ENVIRONMENT:  VAX/VMS Operating System
0043  1 !
0044  1 ! AUTHOR:  Distributed Systems Software Engineering
0045  1 !
0046  1 ! CREATION DATE:  23-JAN-1980
0047  1 !
0048  1 ! MODIFIED BY:
0049  1 !
0050  1 !       V03-008 MKP0009         Kathy Perko     2-Aug-1984
0051  1 !               Fix DEFINE EXEC ADDR n so that, if n doesn't include an area
0052  1 !               number, area 1 is used.
0053  1 !
0054  1 !       V03-007 MKP0008         Kathy Perko     20-April-1984
0055  1 !               Fix DEF NODE nnn ADDR yyy so that, if the address is a duplicate
0056  1 !               of the executor's, the error message indicates "executor"
0057  1 !               instead of "remote node".
```

```
1 !
1 !        V03-006 MKP0007         Kathy Perko     18-April-1984
1 !                Fix DEF EXEC NAME or ADDRESS so that exec id globals
1 !                are updated.
1 !
1 !        V03-005 MKP0006         Kathy Perko     29-Jan-1984
1 !                If NCP is a V3.0.0, mask area in node numbers.
1 !
1 !        V03-004 MKP0005         Kathy Perko     4-Aug-1983
1 !                Change routines to manipulate permanent database record
1 !                fields to be transparent to ISAM keys at the beginning of
1 !                the records.  Also, redo checking on node ids for the new
1 !                node database format.
1 !
1 !        V03-003 MKP0004         Kathy Perko     29-July-1983
1 !                Redo NML$LISNODEID routine to return only the node id if
1 !                the PSTs datatype is NMA$M_PTY_CM1.
1 !
1 !        V03-002 MKP0003         Kathy Perko     13-July-1982
1 !                Fix NML$LISPARAM to add parameter lengths correctly.
1 !                Fix list routines for channels and set passwords.
1 !
1 !        V03-001 MKP0002         Kathy Perko     16-June-1982
1 !                Add new list routines for range and circuit owner paramters.
1 !
1 !        V02-001 MKP0001         Kathy Perko     2-April-1982
1 !                Add changes for X-25 Protocol Networks and DTE, and
1 !                for X-25 Server Modules.
1 !
1 !        V02-001 MKP001          Kathy Perko     24-July-1981
1 !                Delete NML call to map VMS line to DNA line name and
1 !                vice versa.
1 !--
1
```

```
;    93         0092  1 %SBTTL 'Declarations'
;    94         0093  1
;    95         0094  1 !
;    96         0095  1 ! TABLE OF CONTENTS:
;    97         0096  1 !
;    98         0097  1
;    99         0098  1 FORWARD ROUTINE
;   100         0099  1     NML$LISNMLVER,
;   101         0100  1     NML$LISLOONAM,
;   102         0101  1     NML$LISNODEID,
;   103         0102  1     NML$LISPARAM,
;   104         0103  1     NML$LISPASSWORD,
;   105         0104  1     NML$LISPWSET,
;   106         010:  1     NML$LISRANGE,
;   107         0106  1     NML$LISOWNER,
;   108         0107  1     NML$DEFPARAM,
;   109         0108  1     NML$DEFLINLTY,
;   110         0109  1     NML$DEFLINTRI,
;   111         0110  1     NML$DEF_NODE_ADDR,
;   112         0111  1     NML$DEF_EXEC_ID,
;   113         0112  1     NML_FIND_DUPLICATE_NODE,
;   114         0113  1     NML$DEFNODNLI,
;   115         0114  1     NML$DEFOBJNUM,
;   116         0115  1     NML$PURPARAM,
;   117         0116  1     NML$PURNODNNA;
;   118         0117  1
;   119         0118  1 !
;   120         0119  1 ! INCLUDE FILES:
;   121         0120  1 !
;   122         0121  1
;   123         0122  1 LIBRARY 'LIB$:NMLLIB.L32';
;   124         0123  1 LIBRARY 'SHRLIB$:NMALIBRY.L32';
;   125         0124  1 LIBRARY 'SYS$LIBRARY:STARLET.L32';
;   126         0125  1
;   127         0126  1 !
;   128         0127  1 ! OWN STORAGE:
;   129         0128  1 !
;   130         0129  1
;   131         0130  1 !
;   132         0131  1 ! Parameter buffer and descriptor for use in handling volatile data base
;   133         0132  1 ! data.
;   134         0133  1 !
;   135         0134  1 OWN
;   136         0135  1     nml$t_prmbuffer : VECTOR [256, BYTE];
;   137         0136  1 BIND
;   138         0137  1     nml$q_prmdsc = UPLIT (256, nml$t_prmbuffer) : DESCRIPTOR;
;   139         0138  1 !
;   140         0139  1 ! Entity buffer and descriptor.
;   141         0140  1 !
;   142         0141  1 OWN
;   143         0142  1     nml$t_entbuffer : BBLOCK [nml$k_entbuflen],
;   144         0143  1     nml$q_entbfdsc  : VECTOR [2];
;   145         0144  1
;   146         0145  1 !
;   147         0146  1 ! EXTERNAL REFERENCES:
;   148         0147  1 !
;   149         0148  1
;  150          0149  1 $NML_EXTDEF;
;  151          0150  1
;  152          0151  1 EXTERNAL LITERAL
;  153          0152  1     nml$_recbfovf,
;  154          0153  1     nml$_recdelet;
;  155          0154  1
;  156          0155  1 EXTERNAL
;  157          0156  1     nml$gw_perm_exec_addr : BBLOCK [2],
;  158          0157  1     nml$gb_ncp_version,
;  159          0158  1     nml$gq_perm_exec_name_dsc : VECTOR [2],
;  160          0159  1     nml$gq_proprvmsk : BBLOCK [8];
;  161          0160  1
;  162          0161  1 EXTERNAL ROUTINE
;  163          0162  1     nma$deletefld,
;  164          0163  1     nma$insertfld,
;  165          0164  1     nma$matchrec,
;  166          0165  1     nma$searchfld,
;  167          0166  1     nml$addmsgprm,
;  168          0167  1     nml$bld_reply,
;  169          0168  1     nml$delete_node_rec,
;  170          0169  1     nml$getexeadr,
;  171          0170  1     nml$getnodnam,
;  172          0171  1     nml$getrecowner,
;  173          0172  1     nml$read_loopnode,
;  174          0173  1     nml$readrecord,
;  175          0174  1     nml$send;
;  176          0175  1
```

```
1 %SBTTL 'NML$LISNMLVER  Get NML version number'
1 GLOBAL ROUTINE NML$LISNMLVER (SEM_TABLE, BUFDSC, MSGSIZE, DUMDSC) =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine moves the network management version number into
1 !       the output message as a coded multiple parameter.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       SEM_TABLE       Parameter semantic table entry address.
1 !       BUFDSC          Output message buffer descriptor.
1 !       MSGSIZE         Address of current output message size.
1 !       DUMDSC          Not used.
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       It is assumed that the permanent data base file is already open.
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       Parameter is added to output message buffer.
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       Always returns success (NML$_STS_SUC).
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !
1 !--
1
2     BEGIN
2
2     MAP
2       SEM_TABLE : REF BBLOCK;
2
2     LOCAL
2       BUFFER : VECTOR [6, BYTE],
2       PTR;
2
2     PTR = CH$PTR (BUFFER);                          ! Get pointer to output buffer
2
2 !
2 ! Add version numbers preceded by data type.
2 !
2     CH$WCHAR_A (1, PTR);
2     CH$WCHAR_A (NML$K_VERSION, PTR);
2     CH$WCHAR_A (1, PTR);
2     CH$WCHAR_A (NML$K_DEC_ECO, PTR);
2     CH$WCHAR_A (1, PTR);
2     CH$WCHAR_A (NML$K_USER_ECO, PTR);
2
2 !
```

```
;  235       0233  2 ! Add coded multiple version parameter to message.
;  236       0234  2 !
;  237       0235  2     NML$ADDMSGPRM (.BUFDSC,
;  238       0236  2                   .MSGSIZE,
;  239       0237  2                   .SEM_TABLE [PST$W_DATAID],
;  240       0238  2                   .SEM_TABLE [PST$B_DATATYPE] OR 3,
;  241       0239  2                   6,
;  242       0240  2                   BUFFER);
;  243       0241  2
;  244       0242  2     RETURN NML$_STS_SUC                                                                                    ; F
;  245       0243  2
;  246       0244  1     END;                                        ! End of NML$LISNMLVER


                                                                 .TITLE  NML$LISPRM NML special parameter handling routi
;                                                                                   nes
                                                                 .IDENT  \V04-000\

                                                                 .PSECT  $PLIT$,NOWRT,NOEXE,2

                               00000100  00000 P.AAA:    .LONG   256                                         ;
                               00000000' 00004           .ADDRESS NML$T_PRMBUFFER                            ;

                                                                 .PSECT  $OWN$,NOEXE,2

                                         00000 NML$T_PRMBUFFER:
                                                                 .BLKB   256
                                         00100 NML$T_ENTBUFFER:
                                                                 .BLKB   64
                                         00140 NML$Q_ENTBFDSC:
                                                                 .BLKB   8

                                               NML$Q_PRMDSC=          P.AAA
                                                                 .EXTRN  NML$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  N"L$AB_RCVBUFFER
                                                                 .EXTRN  NML$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$_RECBFOVF, NML$_RECDELET                       ;
                                                                   .EXTRN  NML$GW_PERM_EXEC_ADDR                              ;
                                                                   .EXTRN  NML$GB_NCP_VERSION                                 ;
                                                                   .EXTRN  NML$GQ_PERM_EXEC_NAME_DSC                          ;
                                                                   .EXTRN  NML$GQ_PROPRVMSK                                   ;
                                                                   .EXTRN  NMA$DELETEFLD, NMA$INSERTFLD                       ;
                                                                   .EXTRN  NMA$MATCHREC, NMA$SEARCHFLD                        ;
                                                                   .EXTRN  NML$ADDMSGPRM, NML$BLD_REPLY                       ;
                                                                   .EXTRN  NML$DELETE_NODE_REC                                ;
                                                                   .EXTRN  NML$GETEXEADR, NML$GETNODNAM                       ;
                                                                   .EXTRN  NML$GETRECOWNER                                    ;
                                                                   .EXTRN  NML$READ_LOOPNODE                                  ;
                                                                   .EXTRN  NML$READRECORD, NML$SEND                           ;

                                                                   .PSECT  $CODE$,NOWRT,2                                     ;

                                                  0000 00000       .ENTRY  NML$LISNMLVER, Save nothing                        ; 0177
                        5E                     08   C2 00002       SUBL2   #8, SP                                             ;
                        50                     6E   9E 00005       MOVAB   BUFFER, PTR                                        ; 0220
                        80 00010401            8F   D0 00008       MOVL    #66561, (PTR)+                                     ; 0225
                        80                     01   B0 0000F       MOVW    #1, (PTR)+                                         ; 0229
                                               5E   DD 00012       PUSHL   SP                                                 ; 0235
                                               06   DD 00014       PUSHL   #6                                                 ;
                        50          04         AC   D0 00016       MOVL    SEM_TABLE, R0                                      ; 0238
                        51          03         A0   9A 0001A       MOVZBL  3(R0), R1                                          ;
           7E           51                     03   C9 0001E       BISL3   #3, R1, -(SP)                                      ;
                        7E                     60   3C 00022       MOVZWL  (R0), -(SP)                                        ; 0237
                        7E          08         AC   7D 00025       MOVQ    BUFDSC, - SP)                                      ; 0235
                00000000G 00                   06   FB 00029       CALLS   #6, NML$ADDMSGPRM                                  ;
                        50                     01   D0 00030       MOVL    #1, R0                                             ; 0242
                                                    04 00033       RET                                                        ; 0244

; Routine Size:  52 bytes,    Routine Base:  $CODE$ + 0000
```

```
1 %SBTTL 'NML$LISLOONAM  Get loop node name'
1 GLOBAL ROUTINE NML$LISLOONAM (SEM_LIST, BUFDSC, MSGSIZE, DATDSC)=
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine returns the loopback node name for a line.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       SEM_LIST        Parameter semantic table entry address.
1 !       BUFDSC          Output message buffer descriptor address.
1 !       MSGSIZE         Address of current output message size.
1 !       DATDSC          Data buffer descriptor address.
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       It is assumed that the permanent data base file is already open.
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       Always returns success (NML$_STS_SUC).
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !
1 !--
1
2 BEGIN
2
2 MAP
2     sem_list : REF BBLOCK;
2
2 LOCAL
2     circuit_dsc : VECTOR [2],
2     node_dsc :    VECTOR [2],
2     node_rec_buf: BBLOCK [nml$k_recbflen],      ! Buffer for node data
2     node_rec_dsc: VECTOR [2],                   ! Descriptor of node data buffer
2     node_rec_data:VECTOR [2],                   ! Descriptor of data in node
2                                                 !       data buffer.
2     status;
2
2
2 !
2 ! Get the circuit ID from the circuit's permanent database record.
2 ! If this fails, it's a bug.
2 !
2 circuit_dsc [0] = 0;
2 circuit_dsc [1] = 0;
2 IF NOT nma$searchfld (.datdsc,
2                     nml$c_key_cir,
2                     circuit_dsc [0],
2                     circuit_dsc [1]) THEN
2     RETURN nml$_sts_mpr;
2 node_rec_dsc [0] = nml$k_recbflen;
```

```
;  305       0302  2 node_rec_dsc [1] = node_rec_buf;
;  306       0303  2 node_rec_data [1] = node_rec_buf;
;  307       0304  2 !
;  308       0305  2 ! Call routine to read through the known loopnodes in the node permanent
;  309       0306  2 ! database, looking for loopnode on the circuit being listed.
;  310       0307  2 !
;  311       0308  2 IF nml$read_loopnode (circuit_dsc,
;  312       0309  2                        node_rec_dsc,
;  313       0310  2                        node_rec_data) THEN
;  314       0311  3     BEGIN
;  315       0312  3     node_dsc [0]= 0;
;  316       0313  3     node_dsc [1] = 0;
;  317       0314  3     IF nma$searchfld (node_rec_data,
;  318       0315  3                       nma$c_pcno_nna,
;  319       0316  3                       node_dsc [0],
;  320       0317  3                       node_dsc [1]) THEN
;  321       0318  3         nml$addmsgprm (.bufdsc,
;  322       0319  3                        .msgsize,
;  323       0320  3                        .sem_list [pst$w_dataid],
;  324       0321  3                        .sem_list [pst$b_datatype],
;  325       0322  3                        .node_dsc [0],
;  326       0323  3                        .node_dsc [1]);
;  327       0324  2     END;
;  328       0325  2 RETURN nml$_sts_suc
;  329       0326  1 END;                                  ! End of NML$LISLOONAM
```

```
                               0004 00000           .ENTRY  NML$LISLOONAM, Save R2                  ; 0246
          52 00000000G  00   9E 00002           MOVAB   NMA$SEARCHFLD, R2                       ;
          5E      FBE0  CE   9E 00009           MOVAB   -1056(SP), SP                           ;
                    F8  AD   7C 0000E           CLRQ    CIRCUIT_DSC                             ; 0294
                    FC  AD   9F 00011           PUSHAB  CIRCUIT_DSC+4                           ; 0299
                    F8  AD   9F 00014           PUSHAB  CIRCUIT_DSC                             ; 0298
          7E            04   CE 00017           MNEGL   #4, -(SP)                               ; 0296
                    10  AC   DD 0001A           PUSHL   DATDSC                                  ;
          62            04   FB 0001D           CALLS   #4, NMA$SEARCHFLD                       ;
          04            50   E8 00020           BLBS    R0, 1$                                  ;
          50            0A   CE 00023           MNEGL   #10, R0                                 ; 0300
                             04 00026           RET                                             ;
     08   AE      0400  8F   3C 00027  1$:      MOVZWL  #1024, NODE_REC_DSC                     ; 0301
     0C   AE        10  AE   9E 0002D           MOVAB   NODE_REC_BUF, NODE_REC_DSC+4            ; 0302
     04   AE        10  AE   9E 00032           MOVAB   NODE_REC_BUF, NODE_REC_DATA+4           ; 0303
                        5E   DD 00037           PUSHL   SP                                      ; 0308
                    0C  AE   9F 00039           PUSHAB  NODE_REC_DSC                            ;
                    F8  AD   9F 0003C           PUSHAB  CIRCUIT_DSC                             ;
   00000000G  00             03   FB 0003F      CALLS   #3, NML$READ_LOOPNODE                   ;
          31            50   E9 00046           BLBC    R0, 2$                                  ;
                    F0  AD   7C 00049           CLRQ    NODE_DSC                                ; 0312
                    F4  AD   9F 0004C           PUSHAB  NODE_DSC+4                              ; 0317
                    F0  AD   9F 0004F           PUSHAB  NODE_DSC                                ; 0316
          7E      01F4  8F   3C 00052           MOVZWL  #500, -(SP)                             ; 0314
                    0C  AE   9F 00057           PUSHAB  NODE_REC_DATA                           ;
          62            04   FB 0005A           CALLS   #4, NMA$SEARCHFLD                       ;
          1A            50   E9 0005D           BLBC    R0, 2$                                  ;
```

```
                              7E        F0   AD  7D 00060          MOVQ    NODE_DSC, -(SP)                        ; 0322
                              50        04   AC  D0 00064          MOVL    SEM_LIST, R0                           ; 0321
                              7E        03   A0  9A 00068          MOVZBL  3(R0), -(SP)                           ;
                              7E             60  3C 0006C          MOVZWL  (R0), -(SP)                            ; 0320
                              7E        08   AC  7D 0006F          MOVQ    BUFDSC, -(SP)                          ; 0318
                    00000000G 00             06  FB 00073          CALLS   #6, NML$ADDMSGPRM                      ;
                              50             01  D0 0007A 2$:      MOVL    #1, R0                                 ; 0325
                                                 04 0007D          RET                                            ; 0326

; Routine Size:  126 bytes,    Routine Base:  $CODE$ + 0034
;    331          0327  1 %SBTTL 'NML$LISNODEID  Get host node id'
;    332          0328  1 GLOBAL ROUTINE NML$LISNODEID (SEM_LIST, BUFDSC, MSGSIZE, DATDSC)=
;    333          0329  1
;    334          0330  1 !++
;    335          0331  1 ! FUNCTIONAL DESCRIPTION:
;    336          0332  1 !
;    337          0333  1 !       This routine gets the host node identification string.
;    338          0334  1 !
;    339          0335  1 ! FORMAL PARAMETERS:
;    340          0336  1 !
;    341          0337  1 !       SEM_LIST        Parameter semantic table entry address.
;    342          0338  1 !       BUFDSC          Output message buffer descriptor address.
;    343          0339  1 !       MSGSIZE         Address of current output message size.
;    344          0340  1 !       DATDSC          Data buffer descriptor address.
;    345          0341  1 !
;    346          0342  1 ! IMPLICIT INPUTS:
;    347          0343  1 !
;    348          0344  1 !       It is assumed that the permanent data base file is already open.
;    349          0345  1 !
;    350          0346  1 ! IMPLICIT OUTPUTS:
;    351          0347  1 !
;    352          0348  1 !       NONE
;    353          0349  1 !
;    354          0350  1 ! ROUTINE VALUE:
;    355          0351  1 ! COMPLETION CODES:
;    356          0352  1 !
;    357          0353  1 !       Always returns success (NML$_STS_SUC).
;    358          0354  1 !
;    359          0355  1 ! SIDE EFFECTS:
;    360          0356  1 !
;    361          0357  1 !       NONE
;    362          0358  1 !
;    363          0359  1 !--
;    364          0360  1
;    365          0361  2 BEGIN
;    366          0362  2
;    367          0363  2 MAP
;    368          0364  2     sem_list : REF BBLOCK;
;    369          0365  2
;    370          0366  2 OWN
;    371          0367  2     tmpbuffer : BBLOCK [6];
;    372          0368  2 BIND
;    373          0369  2     tmpdsc = UPLIT (6, tmpbuffer) : DESCRIPTOR;
;    374          0370  2
;    375          0371  2 LOCAL
;    376          0372  2     cm_count,
;    377          0373  2     fldadr,
;    378          0374  2     fldsize,
;    379          0375  2     length,
;    380          0376  2     namdsc   : DESCRIPTOR,
;    381          0377  2     hostadr : WORD,
;    382          0378  2     ptr,
;    383          0379  2     reslen;
;    384          0380  2
;    385          0381  2 fldadr = 0;
;    386          0382  2
;    387          0383  2 IF NOT nma$searchfld (.datdsc,
2                           .sem_list [pst$w_dataid],
2                           fldsize,
2                           fldadr) THEN
2       RETURN nml$_sts_pty;
2
2   ptr = nml$t_prmbuffer;
2   !
2   ! Get the maximum number of fields in the coded multiple: 1 (node address
2   ! only) or 2 (node address and node name).
2   !
2   cm_count = .sem_list [pst$b_datatype] AND NOT nma$m_pty_cmu;
2
2   hostadr = .(.fldadr)<0,16>;
2   !
2   ! Add node address field.
2   !
2   CH$WCHAR_A (2, ptr);
2   !
2   ! If the NCP I'm talking to is speaking NICE V3.0.0 or less, clear the
2   ! area number from node numbers in the executor's area.
2   !
2   IF CH$RCHAR (nml$gb_ncp_version) LEQ 3 THEN
3       BEGIN
3       MAP
3           hostadr : BBLOCK [2];
3
3       IF .hostadr [nma$v_area] EQL .nml$gw_perm_exec_addr [nma$v_area] THEN
3           hostadr [nma$v_area] = 0;
2       END;
2
2   ptr = CH$MOVE (2, hostadr, .ptr);
2   IF .cm_count EQL 2 THEN
3       BEGIN
3       nml$getnodnam (.hostadr, tmpdsc, reslen);
3       namdsc [dsc$w_length] = .reslen;
3       namdsc [dsc$a_pointer] = tmpbuffer;
3       !
3       ! Add node name field if the length is not zero.
3       !
3       IF .namdsc [dsc$w_length] NEQU 0 THEN
4           BEGIN
4           CH$WCHAR_A (nma$m_pty_asc, ptr);
4           CH$WCHAR_A (.namdsc [dsc$w_length], ptr);
4           ptr = CH$MOVE (.namdsc [dsc$w_length],
4                           .namdsc [dsc$a_pointer],
4                           .ptr);
4           END
3       ELSE
3           cm_count = 1;
2       END;
2
2   length = .ptr - nml$t_prmbuffer;
2   nml$addmsgprm (.bufdsc,
2                   .msgsize,
2                   .sem_list [pst$w_dataid],
2                   nma$m_pty_cmu OR .cm_count,
2                   .length,
;  445          0441  2                nml$t_prmbuffer);
;  446          0442  2
;  447          0443  2 RETURN nml$_sts_suc
;  448          0444  1 END;                                 ! End of NML$LISNODEID


                                                                        .PSECT   $PLIT$,NOWRT,NOEXE,2

                                                    00000006  00008 P.AAB:  .LONG    6                                           ;
                                                    00000000' 0000C         .ADDRESS TMPBUFFER                                   ;

                                                                        .PSECT   $OWN$,NOEXE,2

                                                              00148 TMPBUFFER:
                                                                        .BLKB    6

                                                                   TMPDSC=          P.AAB


                                                                        .PSECT   $CODE$,NOWRT,2

                                                        01FC 00000      .ENTRY   NML$LISNODEID, Save R2,R3,R4,R5,R6,R7,R8    ; 0328
                                        58 00000000'  00  9E 00002      MOVAB    NML$T_PRMBUFFER, R8                         ;
                                        5E            10  C2 00009      SUBL2    #16, SP                                     ;
                                                      7E  D4 0000C      CLRL     FLDADR                                      ; 0381
                                                      5E  DD 0000E      PUSHL    SP                                          ; 0383
                                                  08  AE  9F 00010      PUSHAB   FLDSIZE                                     ;
                                        56        04  AC  D0 00013      MOVL     SEM_LIST, R6                                ; 0384
                                        7E            66  3C 00017      MOVZWL   (R6), -(SP)                                 ;
                                                  10  AC  DD 0001A      PUSHL    DATDSC                                      ; 0383
                            00000000G   00            04  FB 0001D      CALLS    #4, NMA$SEARCHFLD                           ;
                                        04            50  E8 00024      BLBᶜ     R0, 1$                                      ;
                                        50            0C  CE 00027      MNEGL    #12, R0                                     ; 0387
                                                          04 0002A      RET                                                  ;
                                        53            68  9E 0002B 1$:  MOVAB    NML$T_PRMBUFFER, PTR                        ; 0389
      57          03    A6              06            00  EF 0002E      EXTZV    #0, #8, 3(R6), CM_COUNT                     ; 0394
                                        50        00  BE  B0 00034      MOVW     @FLDADR, HOSTADR                            ; 0396
                                        83            02  90 00038      MOVB     #2, (PTR)+                                  ; 0400
                                        03 00000000G  00  91 0003B      CMPB     NML$GB_NCP_VERSION, #3                      ; 0405
                                                      15  1A 00042      BGTRU    2$                                          ;
      51 00000000G    00                06            02  EF 00044      EXTZV    #2, #6, NML$GW_PERM_EXEC_ADDR+1, R1         ; 0410
      51              50                06            0A  ED 0004D      CMPZV    #10, #6, HOSTADR, R1                        ;
                                                      05  12 00052      BNEQ     2$                                          ;
                                        50      FC00  8F  AA 00054      BICW2    #64512, HOSTADR                             ; 0411
                                        83            50  B0 00059 2$:  MOVW     HOSTADR, (PTR)+                             ; 0414
                                        02            57  D1 0005C      CMPL     CM_COUNT, #2                                ; 0415
                                                      35  12 0005F      BNEQ     4$                                          ;
                                                  08  AE  9F 00061      PUSHAB   RESLEN                                      ; 0417
                                           00000000'  00  9F 00064      PUSHAB   TMPDSC                                      ;
                                        7E            50  3C 0006A      MOVZWL   HOSTADR, -(SP)                              ;
                            00000000G   00            03  FB 0006D      CALLS    #3, NML$GETNODNAM                           ;
                                   0C   AE        08  AE  B0 00074      MOVW     RESLEN, NAMDSC                              ; 0418
                                   10   AE      0148  C8  9E 00079      MOVAB    TMPBUFFER, NAMDSC+4                         ; 0419
                                        50        0C  AE  3C 0007F      MOVZWL   NAMDSC, R0                                  ; 0423
                                                      0E  13 00083      BEQL     3$                                          ;
                                        83        40  8F  90 00085      MOVB     #64, (PTR)+                                 ; 0425
```

```
                                       83            50 90 00089          MOVB    R0, (PTR)+                                  ; 0426
                         63       10   BE            50 28 0008C          MOVC3   R0, @NAMDSC+4, (PTR)                        ; 0429
                                                     03 11 00091          BRB     4$                                          ; 0423
                                       57            01 D0 00093 3$:      MOVL    #1, CM_COUNT                                ; 0432
                                       50            68 9E 00096 4$:      MOVAB   NML$T_PRMBUFFER, R0                         ; 0435
                         50            53            50 C3 00099          SUBL3   R0, PTR, LENGTH                             ;
                                              0101   8F BB 0009D          PUSHR   #^M<R0,R8>                                  ; 0440
                         7E            57 000000C0   8F C9 000A1          BISL3   #192, CM_COUNT, -(SP)                       ; 0439
                                       7E            66 3C 000A9          MOVZWL  (R6), -(SP)                                 ; 0438
                                       7E       08   AC 7D 000AC          MOVQ    BUFDSC, -(SP)                               ; 0436
                            00000000G  00            06 FB 000B0          CALLS   #6, NML$ADDMSGPRM                           ;
                                       50            01 D0 000B7          MOVL    #1, R0                                      ; 0443
                                                        04 000BA          RET                                                 ; 0444

; Routine Size:  187 bytes,    Routine Base:  $CODE$ + 00B2
```

```
1 %SBTTL 'NML$LISPARAM  Get parameter'
1 GLOBAL ROUTINE NML$LISPARAM (SEM_LIST, BUFDSC, MSGSIZE, DATDSC)=
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine returns a parameter.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       SEM_LIST        Parameter semantic table entry address.
1 !       BUFDSC          Output message buffer descriptor address.
1 !       MSGSIZE         Address of current output message size.
1 !       DATDSC          QIO buffer descriptor address.
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       It is assumed that the permanent data base file is already open.
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       The output message buffer contains the coded multiple version number.
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       Always returns success (NML$_STS_SUC).
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !
1 !--
1
2 BEGIN
2
2 MAP
2     SEM_LIST : REF BBLOCK;
2
2 LOCAL
2     DATATYPE : BBLOCK [1],      ! NICE parameter data type.
2     FLDADR,
2     FLDSIZE;
2
2     FLDADR = 0;
2
2     IF NMA$SEARCHFLD (.DATDSC,
2                       .SEM_LIST [PST$W_DATAID],
2                       FLDSIZE,
2                       FLDADR)
2     THEN
3         BEGIN
3         DATATYPE = .SEM_LIST [PST$B_DATATYPE];
3         !
3         ! If the parameter is not an ASCII or hex image field, the length
3         ! goes in the datatype byte.  Add it here.
3         !
```

```
;   507          0502  3          IF (NOT .DATATYPE [NMA$V_PTY_ASC]) AND
;   508          0503  3             (.DATATYPE [NMA$V_PTY_TYP] NEQ NMA$C_PTY_HI) THEN
;   509          0504  3              DATATYPE = .DATATYPE OR .FLDSIZE;
;   510          0505  3          NML$ADDMSGPRM (.BUFDSC,
;   511          0506  3                         .MSGSIZE,
;   512          0507  3                         .SEM_LIST [PST$W_DATAID],
;   513          0508  3                         .DATATYPE,
;   514          0509  3                         .FLDSIZE,
;   515          0510  3                         .FLDADR);
;   516          0511  2          END;
;   517          0512  2
;   518          0513  2 RETURN NML$_STS_SUC
;   519          0514  1 END;                                  ! End of NML$LISPARAM



                                              0004 00000          .ENTRY  NML$LISPARAM, Save R2                                      ; 0446
                                5E              04  C2 00002          SUBL2   #4, SP                                                     ;
                                                7E  D4 00005          CLRL    FLDADR                                                     ; 0489
                                                5E  DD 00007          PUSHL   SP                                                         ; 0491
                                        08      AE  9F 00009          PUSHAB  FLDSIZE                                                    ;
                                52      04      AC  D0 0000C          MOVL    SEM_LIST, R2                                               ; 0492
                                7E              62  3C 00010          MOVZWL  (R2), -(SP)                                                ;
                                        10      AC  DD 00013          PUSHL   DATDSC                                                     ; 0491
                      00000000G 00              04  FB 00016          CALLS   #4, NMA$SEARCHFLD                                          ;
                                29              50  E9 0001D          BLBC    R0, 2$                                                     ;
                                50      03      A2  90 00020          MOVB    3(R2), DATATYPE                                            ; 0497
                      0B        50              06  E0 00024          BBS     #6, DATATYPE, 1$                                           ; 0502
       20             50        0F              00  ED 00028          CMPZV   #0, #15, DATATYPE, #32                                     ; 0503
                                                04  13 0002D          BEQL    1$                                                         ;
                                50      04      AE  88 0002F          BISB2   FLDSIZE, DATATYPE                                          ; 0504
                                                6E  DD 00033 1$:      PUSHL   FLDADR                                                     ; 0510
                                        08      AE  DD 00035          PUSHL   FLDSIZE                                                    ; 0509
                                7E              50  9A 00038          MOVZBL  DATATYPE, -(SP)                                            ; 0508
                                7E              62  3C 0003B          MOVZWL  (R2), -(SP)                                                ; 0507
                                7E      08      AC  7D 0003E          MOVQ    BUFDSC, -(SP)                                              ; 0505
                      00000000G 00              06  FB 00042          CALLS   #6, NML$ADDMSGPRM                                          ;
                                50              01  D0 00049 2$:      MOVL    #1, R0                                                     ; 0513
                                                    04 0004C          RET                                                                ; 0514

; Routine Size:  77 bytes,    Routine Base:  $CODE$ + 016D
```

```
1 %SBTTL 'NML$LISPASSWORD  Get parameter'
1 GLOBAL ROUTINE NML$LISPASSWORD (SEM_LIST, BUFDSC, MSGSIZE, DATDSC)=
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine adds a password parameter to the output message if
1 !       the user has the BYPASS privilege.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       SEM_LIST        Parameter semantic table entry address.
1 !       BUFDSC          Output message buffer descriptor address.
1 !       MSGSIZE         Address of current output message size.
1 !       DATDSC          Address of data buffer descriptor.
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       It is assumed that the permanent data base file is already open.
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       Always returns success (NML$_STS_SUC).
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !
1 !--
1
2 BEGIN
2
2 MAP
2     SEM_LIST : REF BBLOCK;
2
2 BIND
2     STRDSC = $ASCID ('no access rights') : DESCRIPTOR;
2
2 LOCAL
2     FLDADR,
2     FLDSIZE;
2
2 IF NOT .NML$GQ_PROPRVMSK [PRV$V_BYPASS]
2 THEN
3     BEGIN
3     !
3     ! User does not have BYPASS privilege so return string to indicate that
3     ! a password is set if one is found.
3     !
3
3     FLDADR = 0;
3     IF NMA$SEARCHFLD (.DATDSC,
;  578          0572  3                      .SEM_LIST [PST$W_DATAID],
;  579          0573  3                      FLDSIZE,
;  580          0574  3                      FLDADR)
;  581          0575  3     THEN
;  582          0576  4         BEGIN
;  583          0577  4
;  584          0578  4         NML$ADDMSGPRM (.BUFDSC,
;  585          0579  4                        .MSGSIZE,
;  586          0580  4                        .SEM_LIST [PST$W_DATAID],
;  587          0581  4                        .SEM_LIST [PST$B_DATATYPE],
;  588          0582  4                        .STRDSC [DSC$W_LENGTH],
;  589          0583  4                        .STRDSC [DSC$A_POINTER]);
;  590          0584  4
;  591          0585  4         RETURN NML$_STS_SUC
;  592          0586  4
;  593          0587  3         END;
;  594          0588  2     END;
;  595          0589  2 !
;  596          0590  2 ! Call the normal parameter routine.
;  597          0591  2 !
;  598          0592  2 NML$LISPARAM (.SEM_LIST,
;  599          0593  2               .BUFDSC,
;  600          0594  2               .MSGSIZE,
;  601          0595  2               .DATDSC);
;  602          0596  2
;  603          0597  2 RETURN NML$_STS_SUC
;  604          0598  1 END;                                    ! End of NML$LISPASSWORD



                                                                             .PSECT  $PLIT$,NOWRT,NOEXE,2

74  68  67  69  72  20  73  73  65  63  63  61  20  6F  6E  0001C P.AAD:   .ASCII  \no access rights\                  :
                                                        73  0001F                                                     :
                                                  00000010  00020 P.AAC:   .LONG   16                                  :
                                                  00000000' 00024          .ADDRESS P.AAD                              :

                                                                   STRDSC=           P.AAC


                                                                             .PSECT  $CODE$,NOWRT,2

                                                      0004 00000           .ENTRY  NML$LISPASSWORD, Save R2             ; 0516
                                 5E              08  C2 00002           SUBL2   #8, SP                               :
                     3C 000000C0G  00              05  E0 00005           BBS     #5, NML$GQ_PROPRVMSK+3, 1$           ; 0562
                                                6E  D4 0000D           CLRL    FLDADR                               ; 0570
                                                5E  DD 0000F           PUSHL   SP                                   ; 0571
                                       08       AE  9F 00011           PUSHAB  FLDSIZE                              :
                                 52    04       AC  D0 00014           MOVL    SEM_LIST, R2                         ; 0572
                                 7E             62  3C 00018           MOVZWL  (R2), -(SP)                          :
                                       10       AC  DD 0001B           PUSHL   DATDSC                               ; 0571
                        00000000G  00              04  FB 0001E           CALLS   #4, NMA$SEARCHFLD                    :
                                 21             50  E9 00025           BLBC    R0, 1$                               :
                                 00000000'      00  DD 00028           PUSHL   STRDSC+4                             ; 0583
                              7E 00000000'      00  3C 0002E           MOVZWL  STRDSC, -(SP)                        ; 0582
                              7E       03       A2  9A 00035           MOVZBL  3(R2), -(SP)                         ; 0581
                              7E                62  3C 00039           MOVZWL  (R2), -(SP)                          ; 0580
```

```
                          7E        08   AC  7D 0003C        MOVQ   BUFDSC, -(SP)                        ; 0578
              00000000G   00             06  FB 00040        CALLS  #6, NML$ADDMSGPRM                    ;
                                         0D  11 00047        BRB    2$                                   ; 0585
                          7E        0C   AC  7D 00049 1$:    MOVQ   MSGSIZE, -(SP)                       ; 0594
                          7E        04   AC  7D 0004D        MOVQ   SEM_LIST, -(SP)                      ; 0592
                   FF5D   CF             04  FB 00051        CALLS  #4, NML$LISPARAM                     ;
                          50             01  D0 00056 2$:    MOVL   #1, R0                               ; 0597
                                         04  00059           RET                                         ; 0598

; Routire Size:  90 bytes,    Routine Base:  $CODE$ + 01BA
```

```
1 %SBTTL 'NML$LISPWSET  List password set'
1 GLOBAL ROUTINE NML$LISPWSET (SEM_LIST, BUFDSC, MSGSIZE, DATDSC)=
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine is called while processing a LIST X25-S or X29-S DEST
1 !       command.  If a password is set, it adds a password set indicator to
1 !       the NICE response message.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       SEM_LIST        Parameter semantic table entry address.
1 !       BUFDSC          Output message buffer descriptor address.
1 !       MSGSIZE         Address of current output message size.
1 !       DATDSC          Address of data buffer descriptor.
1 !
1 ! IMPLICIT INPUTS:
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 ! SIDE EFFECTS:
1 !
1 !--
1
2 BEGIN
2
2 MAP
2     SEM_LIST : REF BBLOCK;
2
2 LOCAL
2     FLDSIZE,
2     FLDADR;
2
2 IF NMA$SEARCHFLD (.DATDSC,
2                 .SEM_LIST [PST$W_DATAID],
2                 FLDSIZE,
2                 FLDADR) THEN
3     BEGIN
3     !
3     ! Add password to message with a value of 0.  This indicates simply that
3     ! the password is defined, without actually returning the password.
3     !
3     NML$ADDMSGPRM (.BUFDSC,
3                    .MSGSIZE,
3                    .SEM_LIST [PST$W_DATAID],
3                    1,
3                    1,
3                    UPLIT (0));
2     END;
2 RETURN NML$_STS_SUC
1 END;                                  ! end of NML$LISPWSET
```

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

                            00000000  00028 P.AAE:    .LONG   0                                             ;

                                                          .PSECT  $CODE$,NOWRT,2

                                        0000 00000        .ENTRY  NML$LISPWSET, Save nothing               ; 0600
                   5E                08   C2 00002        SUBL2   #8, SP                                    ;
                                     5E   DD 00005        PUSHL   SP                                        ; 0636
                            08       AE   9F 00007        PUSHAB  FLDSIZE                                   ;
                   7E       04       BC   3C 0000A        MOVZWL  @SEM_LIST, -(SP)                          ; 0637
                            10       AC   DD 0000E        PUSHL   DATDSC                                    ; 0636
         00000000G 00                04   FB 00011        CALLS   #4, NMA$SEARCHFLD                         ;
                   19                50   E9 00018        BLBC    R0, 1$                                    ;
                      00000000'      00   9F 0001B        PUSHAB  P.AAE                                     ; 0650
                                     01   DD 00021        PUSHL   #1                                        ; 0645
                                     01   DD 00023        PUSHL   #1                                        ;
                   7E       04       BC   3C 00025        MOVZWL  @SEM_LIST, -(SP)                          ; 0647
                   7E       08       AC   7D 00029        MOVQ    BUFDSC, -(SP)                             ; 0645
         00000000G 00                06   FB 0002D        CALLS   #6, NML$ADDMSGPRM                         ;
                   50                01   D0 00034 1$:    MOVL    #1, R0                                    ; 0652
                                          04 00037        RET                                               ; 0653

; Routine Size:  56 bytes,    Routine Base:  $CODE$ + 0214
```

```
1 %SBTTL 'NML$LISRANGE  List range parameter'
1 GLOBAL ROUTINE NML$LISRANGE (SEM_LIST, BUFDSC, MSGSIZE, DATDSC)=
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine is called to list X25 and X29 Destination subaddresses
1 !       and X25 DTE channels.  The destination's subaddresses can be more
1 !       than one range pair, in which case the field length in the permanent
1 !       database is the number of range pairs times 4 (i.e. then length in
1 !       bytes).
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       SEM_LIST        Parameter semantic table entry address.
1 !       BUFDSC          Output message buffer descriptor address.
1 !       MSGSIZE         Address of current output message size.
1 !       DATDSC          Address of data buffer descriptor.
1 !
1 !--
1
2 BEGIN
2
2 MAP
2     SEM_LIST : REF BBLOCK;
2
2 LOCAL
2     FLDADR,
2     FLDSIZE,
2     CM_COUNT,
2     LENGTH,
2     PTR,
2     RANGE_BEGIN,
2     RANGE_END;
2
2 FLDADR = 0;
2
2 IF NMA$SEARCHFLD (.DATDSC,
2                   .SEM_LIST [PST$W_DATAID],
2                   FLDSIZE,
2                   FLDADR) THEN
3     BEGIN
3     !
3     ! For as many range pairs as are set, add them to the NICE response message
3     ! in the form: Parameter ID, Coded multiple data type, word data type,
3     ! range begin, word data type, range end.
3     !
3     WHILE .FLDSIZE GTR 0 DO
4         BEGIN
4         PTR = NML$T_PRMBUFFER;
4         CM_COUNT = 1;
4
4         CH$WCHAR_A (2, PTR);
4         PTR = CH$MOVE (2, (.FLDADR) <0,16>, .PTR);
4         !
4         ! If the range begin = range end, don't include range end.
4         !
```

```
;   719          0711  4          IF (.FLDADR) <0,16> NEQ (.FLDADR) <16,32> THEN
;   720          0712  5              BEGIN
;   721          0713  5              CM_COUNT = .CM_COUNT + 1;
;   722          0714  5              CH$WCHAR_A (2, PTR);
;   723          0715  5              PTR = CH$MOVE (2, (.FLDADR) <16,32>, .PTR);
;   724          0716  4              END;
;   725          0717  4
;   726          0718  4          LENGTH = .PTR - NML$T_PRMBUFFER;
;   727          0719  4          NML$ADDMSGPRM (.BUFDSC,
;   728          0720  4                          .MSGSIZE,
;   729          0721  4                          .SEM_LIST [PST$W_DATAID],
;   730          0722  4                          .SEM_LIST [PST$B_DATATYPE] OR .CM_COUNT,
;   731          0723  4                          .LENGTH,
;   732          0724  4                          NML$T_PRMBUFFER);
;   733          0725  4          !
;   734          0726  4          ! Increment pointer and length to get next range pair in the
;   735          0727  4          ! permanent data base record.
;   736          0728  4          !
;   737          0729  4          FLDADR = .FLDADR + 4;
;   738          0730  4          FLDSIZE = .FLDSIZE - 4;
;   739          0731  3          END;
;   740          0732  2      END;
;   741          0733  2
;   742          0734  2 RETURN NML$_STS_SUC
;   743          0735  1 END;                                   ! end of NML$LISRANGE




                                        007C 00000          .ENTRY  NML$LISRANGE, Save R2,R3,R4,R5,R6                    ; 0655
                      56 00000000'   00   9E 00002          MOVAB   NML$T_PRMBUFFER, R6                                  ;
                      5E             04   C2 00009          SUBL2   #4, SP                                               ;
                                     7E   D4 0000C          CLRL    F'DADR                                               ; 0689
                                     5E   DD 0000E          PUSHL   SF                                                   ; 0691
                               08    AE   9F 00010          PUSHAB  FL'SIZE                                              ;
                      7E       04    BC   3C 00013          MOVZWL  @' M_LIST, -(SP)                                     ; 0692
                               10    AC   DD 00017          PUSHL   DATDSC                                               ; 0691
           00000000G  00             04   FB 0001A          CALLS   #4, NMA$SEARCHFLD                                    ;
                      56             50   E9 00021          BLBC    R0, 3$                                               ;
                      53       04    AC   D0 00024          MOVL    SEM_LIST, R3                                         ; 0722
                               04    AE   D5 00028 1$:      TSTL    FLDSIZE                                              ; 0701
                                     4D   15 0002B          BLEQ    3$                                                   ;
                      52             66   9E 0002D          MOVAB   NML$T_PRMBUFFER, PTR                                 ; 0703
                      54             01   D0 00030          MOVL    #1, CM_COUNT                                         ; 0704
                      82             02   90 00033          MOVB    #2, (PTR)+                                           ; 0706
                      82       00    BE   B0 00036          MOVW    @FLDADR, (PTR)+                                      ; 0707
                      50             6E   D0 0003A          MOVL    FLDADR, R0                                           ; 0711
                      51       02    A0   9E 0003D          MOVAB   2(R0), R1                                            ;
                      51             6E   D1 00041          CMPL    FLDADR, R1                                           ;
                                     09   13 00044          BEQL    2$                                                   ;
                                     54   D6 00046          INCL    CM_COUNT                                             ; 0713
                      82             02   90 00048          MOVB    #2, (PTR)+                                           ; 0714
                      82       02    A0   B0 0004B          MOVW    2(R0), (PTR)+                                        ; 0715
                      50             66   9E 0004F 2$:      MOVAB   NML$T_PRMBUFFER, R0                                  ; 0718
     55               52             50   C3 00052          SUBL3   R0, PTR, LENGTH                                      ;
                             0060    8F   BB 00056          PUSHR   #^M<R5,R6>                                           ; 0723
                             50        03   A3  9A 0005A          MOVZBL  3(R3), R0                          ; 0722
               7E            50             54  C9 0005E          BISL3   CM_COUNT, R0, -(SP)                ;
                             7E        04   BC  3C 00062          MOVZWL  @SEM_LIST, -(SP)                   ; 0721
                             7E        08   AC  7D 00066          MOVQ    BUFDSC, -(SP)                      ; 0719
                  00000000G  00             06  FB 0006A          CALLS   #6, NML$ADDMSGPRM                  ;
                             6E             04  C0 00071          ADDL2   #4, FLDADR                         ; 0729
                       04    AE             04  C2 00074          SUBL2   #4, FLDSIZE                        ; 0730
                                            AE  11 00078          BRB     1$                                 ; 0701
                             50             01  D0 0007A 3$:      MOVL    #1, R0                             ; 0734
                                                04 0007D          RET                                        ; 0735

; Routine Size:  126 bytes,    Routine Base:  $CODE$ + 024C
```

```
1 %SBTTL 'NML$LISOWNER  Get OWNER parameter'
1 GLOBAL ROUTINE NML$LISOWNER (SEM_LIST, BUFDSC, MSGSIZE, DATDSC)=
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !       This routine adds the circuit parameter, OWNER, to the NICE
1 !       response message.  The owner parameter is saved as a bit value.
1 !       If it's set, the executor owns the circuit.  Check to see if
1 !       it's set, and, if so, return the executor node ID.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       SEM_LIST        Parameter semantic table entry address.
1 !       BUFDSC          Output message buffer descriptor address.
1 !       MSGSIZE         Address of current output message size.
1 !       DATDSC          QIO buffer descriptor address.
1 !
1 ! IMPLICI7 INPUTS:
1 !       It is assumed that the permanent data base file is already open.
1 !
1 ! IMPLICIT OUTPUTS:
1 !       The output message buffer contains the coded multiple executor node
1 !       address.
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !       Always returns success (NML$_STS_SUC).
1 !
1 !--
1
2 BEGIN
2
2 MAP
2     SEM_LIST : REF BBLOCK;
2
2 BIND EXECUTOR = UPLIT BYTE
2       (NMA$M_PTY_COD+1, NMA$C_ENT_NOD,        ! Entity type = node
2       2, WORD (0));   ! Node address = executor
2
2 LOCAL
2     FLDADR,
2     FLDSIZE;
2
2 FLDADR = 0;
2 IF NMA$SEARCHFLD (.DATDSC,
2                   .SEM_LIST [PST$W_DATAID],
2                   FLDSIZE,
2                   FLDADR) THEN
3     BEGIN
3     IF ..FLDADR THEN
3       NML$ADDMSGPRM (.BUFDSC,
3                       .MSGSIZE,
3                       .SEM_LIST [PST$W_DATAID],
3                       .SEM_LIST [PST$B_DATATYPE] OR 2,
3                       5,
3                       EXECUTOR);
2     END;
```

```
;  802          0793  2 RETURN NML$_STS_SUC
;  803          0794  1 END;                                   ! End of NML$LISOWNER


                                                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

                                          02   00  81  0002C P.AAF:     .BYTE   -127, 0, 2                                  ;
                                              0000   0002F              .WORD   0                                           ;

                                                               EXECUTOR=           P.AAF


                                                                        .PSECT  $CODE$,NOWRT,2

                                                 0004 00000             .ENTRY  NML$LISOWNER, Save R2                       ; 0737
                               5E          04   C2 00002                SUBL2   #4, SP                                      ;
                                           7E   D4 00005                CLRL    FLDADR                                      ; 0779
                                           5E   DD 00007                PUSHL   SP                                          ; 0780
                                    08     AE   9F 00009                PUSHAB  FLDSIZE                                     ;
                               52   04     AC   D0 0000C                MOVL    SEM_LIST, R2                                ; 0781
                               7E          62   3C 00010                MOVZWL  (R2), -(SP)                                 ;
                                    10     AC   DD 00013                PUSHL   DATDSC                                      ; 0780
                   00000000G   00          04   FB 00016                CALLS   #4, NMA$SEARCHFLD                           ;
                               22          50   E9 0001D                BLBC    R0, 1$                                      ;
                               1E   00     BE   E9 00020                BLBC    @FLDADR, 1$                                 ; 0785
                                  00000000'  00   9F 00024              PUSHAB  EXECUTOR                                    ; 0786
                                           05   DD 0002A                PUSHL   #5                                          ;
                               50   03     A2   9A 0002C                MOVZBL  3(R2), R0                                   ; 0789
                   7E          50          02   C9 00030                BISL3   #2, R0, -(SP)                               ;
                               7E          62   3C 00034                MOVZWL  (R2), -(SP)                                 ; 0788
                               7E   08     AC   7D 00037                MOVQ    BUFDSC, -(SP)                               ; 0786
                   00000000G   00          06   FB 0003B                CALLS   #6, NML$ADDMSGPRM                           ;
                               50          01   D0 00042 1$:            MOVL    #1, R0                                      ; 0793
                                                04 00045                RET                                                 ; 0794

; Routine Size:  70 bytes,    Routine Base:  $CODE$ + 02CA
```

```
1 %SBTTL 'NML$DEFPARAM  Add parameter'
1 GLOBAL ROUTINE NML$DEFPARAM (SEM_LIST, BUFSIZE, LENGTH, ADDR, RTNDSC)=
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine adds a parameter to a permanent data base record.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       SEM_LIST        Parameter semantic table entry address.
1 !       BUFSIZE         Permanent database record maximum size.
1 !       LENGTH          Length of parameter to insert in record.
1 !       ADDR            Address of parameter to insert in record.
1 !       RTNDSC          Permanent database record buffer descriptor address.
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       It is assumed that the permanent data base file is already open.
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       The parameter is added to the record.
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       Always returns success (NML$_STS_SUC).
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !
1 !--
1
2     BEGIN
2
2     MAP
2         SEM_LIST : REF BBLOCK;
2
2         IF NOT NMA$INSERTFLD (.BUFSIZE,
2                               .SEM_LIST [PST$W_DATAID],
2                               .LENGTH,
2                               .ADDR,
2                               .RTNDSC)
2         THEN
3             BEGIN
3 !
3 ! Insert failed.
3 !
3             NML$AB_MSGBLOCK [MSB$L_FLAGS] = MSB$M_MSG_FLD;      ! Set message text flag
3             NML$AB_MSGBLOCK [MSB$B_CODE] = NMA$C_STS_MPR;       ! Add error code
3             NML$AB_MSGBLOCK [MSB$L_TEXT] = NML$_RECBFOVF;
3
3             RETURN NML$_STS_MPR
3
2             END;
```

```
;  862          0852  2
;  863          0853  2      RETURN NML$_STS_SUC
;  864          0854  2
;  865          0855  1      END;                                               ! End of NML$DEFPARAM



                                           0004 00000          .ENTRY  NML$DEFPARAM, Save R2                            ; 0796
                          52 00000000G  00  9E 00002          MOVAB   NML$AB_MSGBLOCK, R2                              ;
                          7E        10  AC  7D 00009          MOVQ    ADDR, -(SP)                                      ; 0838
                                    0C  AC  DD 0000D          PUSHL   LENGTH                                           ; 0837
                          7E        04  BC  3C 00010          MOVZWL  @SEM_LIST, -(SP)                                 ; 0836
                                    08  AC  DD 00014          PUSHL   BUFSIZE                                          ; 0835
                 00000000G  00          05  FB 00017          CALLS   #5, NMA$INSERTFLD                                ;
                            13          50  E8 0001E          BLBS    R0, 1$                                           ;
                            62          04  D0 00021          MOVL    #4, NML$AB_MSGBLOCK                              ; 0845
                      04    A2          05  8E 00024          MNEGB   #5, NML$AB_MSGBLOCK+4                            ; 0846
                      0C    A2 00000000G  8F  D0 00028        MOVL    #NML$_RECBFOVF, NML$AB_MSGBLOCK+12                 ; 0847
                            50          0A  CE 00030          MNEGL   #10, R0                                          ; 0849
                                            04 00033          RET                                                      ;
                            50          01  D0 00034 1$:      MOVL    #1, R0                                           ; 0853
                                            04 00037          RET                                                      ; 0855

; Routine Size:  56 bytes,    Routine Base:  $CODE$ + 0310
;  867         0856    1 %SBTTL 'NML$DEFLINLTY  Add line type parameter'
;  868         0857    1 GLOBAL ROUTINE NML$DEFLINLTY (SEM_LIST, BUFDSC, LENGTH, ADDR, RTNDSC)=
;  869         0858    1
;  870         0859    1 !++
;  871         0860    1 ! FUNCTIONAL DESCRIPTION:
;  872         0861    1 !
;  873         0862    1 !       This routine adds the line type parameter to the permanent data
;  874         0863    1 !       base record if the value is valid.
;  875         0864    1 !
;  876         0865    1 ! FORMAL PARAMETERS:
;  877         0866    1 !
;  878         0867    1 !       SEM_LIST        Parameter semantic table entry address.
;  879         0868    1 !       BUFSIZE         Permanent database record maximum size.
;  880         0869    1 !       LENGTH          Length of parameter to insert in record.
;  881         0870    1 !       ADDR            Address of parameter to insert in record.
;  882         0871    1 !       RTNDSC          Permanent database record buffer descriptor address.
;  883         0872    1 !
;  884         0873    1 ! IMPLICIT INPUTS:
;  885         0874    1 !
;  886         0875    1 !       It is assumed that the permanent data base file is already open.
;  887         0876    1 !
;  888         0877    1 ! IMPLICIT OUTPUTS:
;  889         0878    1 !
;  890         0879    1 !       The parameter is added to the record.
;  891         0880    1 !
;  892         0881    1 ! ROUTINE VALUE:
;  893         0882    1 ! COMPLETION CODES:
;  894         0883    1 !
;  895         0884    1 !       Always returns success (NML$_STS_SUC).
;  896         0885    1 !
;  897         0886    1 ! SIDE EFFECTS:
;  898         0887    1 !
;  899         0888    1 !       NONE
;  900         0889    1 !
;  901         0890    1 !--
;  902         0891    1
;  903         0892    2     BEGIN
;  904         0893    2
;  905         0894    2     MAP
;  906         0895    2         SEM_LIST : REF BBLOCK;
;  907         0896    2
;  908         0897    2     LOCAL
;  909         0898    2         FLDADR,
;  910         0899    2         FLDSIZE,
;  911         0900    2         STATUS;
;  912         0901    2
;  913         0902    2     IF .(.ADDR)<0,8> EQL NMA$C_LINTY_POI
;  914         0903    2     THEN
;  915         0904    3         BEGIN
;  916         0905    3
;  917         0906    3         FLDSIZE = 0;
;  918         0907    3         IF NMA$SEARCHFLD (.RTNDSC,
;  919         0908    3                           NMA$C_PCLI_TRI,
;  920         0909    3                           FLDSIZE,
;  921         0910    3                           FLDADR)
;  922         0911    3         THEN
;  923         0912    4             BEGIN
;   924          0913  4  !
;   925          0914  4  ! Line has tributary address so it cannot have type=POINT.
;   926          0915  4  !
;   927          0916  4
;   928          0917  4             NML$AB_MSGBLOCK [MSB$L_FLAGS] = MSB$M_DET_FLD;
;   929          0918  4             NML$AB_MSGBLOCK [MSB$B_CODE] = NMA$C_STS_PVA;
;   930          0919  4             NML$AB_MSGBLOCK [MSB$W_DETAIL] = NMA$C_PCLI_LTY;
;   931          0920  4
;   932          0921  4             RETURN NML$_STS_PVA
;   933          0922  4
;   934          0923  3             END;
;   935          0924  2         END;
;   936          0925  2
;   937          0926  2     STATUS = NML$DEFPARAM (.SEM_LIST,
;   938          0927  2                            .BUFDSC,
;   939          0928  2                            .LENGTH,
;   940          0929  2                            .ADDR,
;   941          0930  2                            .RTNDSC);
;   942          0931  2
;   943          0932  2     RETURN .STATUS
;   944          0933  2
;   945          0934  1     END;                                         ! End of NML$DEFLINLTY



                                          0004 00000          .ENTRY   NML$DEFLINLTY, Save R2                                ; 0857
                   52 00000000G    00   9E 00002          MOVAB    NML$AB_MSGBLOCK, R2                                   ;
                   5E              08   C2 00009          SUBL2    #8, SP                                                ;
                           10      BC   95 0000C          TSTB     @ADDR                                                 ; 0902
                                   2B   12 0000F          BNEQ     1$                                                    ;
                           04      AE   D4 00011          CLRL     FLDSIZE                                               ; 0906
                                   5E   DD 00014          PUSHL    SP                                                    ; 0907
                           08      AE   9F 00016          PUSHAB   FLDSIZE                                               ;
                   7E    0474      8F   3C 00019          MOVZWL   #1140, -(SP)                                          ;
                           14      AC   DD 0001E          PUSHL    RTNDSC                                                ;
         00000000G 00              04   FB 00021          CALLS    #4, NMA$SEARCHFLD                                     ;
                   11              50   E9 00028          BLBC     R0, 1$                                                ;
                   62              02   D0 0002B          MOVL     #2, NML$AB_MSGBLOCK                                   ; 0917
              04   A2              10   8E 0002E          MNEGB    #16, NML$AB_MSGBLOCK+4                                ; 0918
              08   A2    0458      8F   B0 00032          MOVW     #1112, NML$AB_MSGBLOCK+8                              ; 0919
                   50              20   CE 00038          MNEGL    #32, R0                                               ; 0921
                                   04 0003B               RET                                                            ;
                   7E      10      AC   7D 0003C  1$:     MOVQ     ADDR, -(SP)                                           ; 0929
                   7E      08      AC   7D 00040          MOVQ     BUFDSC, -(SP)                                         ; 0927
                           04      AC   DD 00044          PUSHL    SEM_LIST                                              ; 0926
              FF7C CF              05   FB 00047          CALLS    #5, NML$DEFPARAM                                      ;
                                   04 0004C               RET                                                            ; 0934

; Routine Size:  77 bytes,    Routine Base:  $CODE$ + 0348
```

```
1 %SBTTL 'NML$DEFLINTRI  Add line tributary address parameter'
1 GLOBAL ROUTINE NML$DEFLINTRI (SEM_LIST, BUFDSC, LENGTH, ADDR, RTNDSC)=
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine adds the line tributary address parameter to the
1 !       permanent data base record if it is valid for this line.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       SEM_LIST        Parameter semantic table entry address.
1 !       BUFSIZE         Permanent database record maximum size.
1 !       LENGTH          Length of parameter to insert in record.
1 !       ADDR            Address of parameter to insert in record.
1 !       RTNDSC          Permanent database record buffer descriptor address.
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       It is assumed that the permanent data base file is already open.
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       The parameter is added to the record.
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       Always returns success (NML$_STS_SUC).
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !
1 !--
1
2     BEGIN
2
2     MAP
2         SEM_LIST : REF BBLOCK;
2
2     LOCAL
2         FLDADR,
2         FLDSIZE,
2         STATUS;
2
2     FLDSIZE = 0;
2     IF NMA$SEARCHFLD (.RTNDSC,
2                       NMA$C_PCLI_LTY,
2                       FLDSIZE,
2                       FLDADR)
2     THEN
3         BEGIN
3
3         IF .(.FLDADR)<0,8> EQL NMA$C_LINTY_POI
3         THEN
4             BEGIN
```

```
; 1004       0992  4  !
; 1005       0993  4  ! Line has type=POINT so no tributary address can be specified.
; 1006       0994  4  !
; 1007       0995  4
; 1008       0996  4              NML$AB_MSGBLOCK [MSB$L_FLAGS] = MSB$M_DET_FLD;
; 1009       0997  4              NML$AB_MSGBLOCK [MSB$B_CODE] = NMA$C_STS_PNA;
; 1010       0998  4              NML$AB_MSGBLOCK [MSB$W_DETAIL] = NMA$C_PCLI_TRI;
; 1011       0999  4
; 1012       1000  4              RETURN NML$_STS_PNA
; 1013       1001  4
; 1014       1002  3              END;
; 1015       1003  2          END;
; 1016       1004  2
; 1017       1005  2      STATUS = NML$DEFPARAM (.SEM_LIST,
; 1018       1006  2                             .BUFDSC,
; 1019       1007  2                             .LENGTH,
; 1020       1008  2                             .ADDR,
; 1021       1009  2                             .RTNDSC);
; 1022       1010  2
; 1023       1011  2      RETURN .STATUS
; 1024       1012  2
; 1025       1013  1      END;                                           ! End of NML$DEFLINTRI




                                        0004 00000          .ENTRY  NML$DEFLINTRI, Save R2                          ; 0936
                      52 00000000G   00   9E 00002          MOVAB   NML$AB_MSGBLOCK, R2                              ;
                      5E             08   C2 00009          SUBL2   #8, SP                                           ;
                              04     AE   D4 0000C          CLRL    FLDSIZE                                          ; 0981
                                     5E   DD 0000F          PUSHL   SP                                               ; 0982
                              08     AE   9F 00011          PUSHAB  FLDSIZE                                          ;
                      7E    0458     8F   3C 00014          MOVZWL  #1112, -(SP)                                     ;
                              14     AC   DD 00019          PUSHL   RTNDSC                                           ;
              00000000G   00         04   FB 0001C          CALLS   #4, NMA$SEARCHFLD                                ;
                          16         50   E9 00023          BLBC    R0, 1$                                           ;
                              00     BE   95 00026          TSTB    @FLDADR                                          ; 0989
                                     11   12 00029          BNEQ    1$                                               ;
                          62         02   D0 0002B          MOVL    #2, NML$AB_MSGBLOCK                              ; 0996
                   04     A2         16   8E 0002E          MNEGB   #22, NML$AB_MSGBLOCK+4                           ; 0997
                   08     A2  0474   8F   B0 00032          MOVW    #1140, NML$AB_MSGBLOCK+8                         ; 0998
                          50         2C   CE 00038          MNEGL   #44, R0                                          ; 1000
                                        04 0003B          RET
                          7E  10     AC   7D 0003C 1$:      MOVQ    ADDR, -(SP)                                      ; 1008
                          7E  08     AC   7D 00040          MOVQ    BUFDSC, -(SP)                                    ; 1006
                              04     AC   DD 00044          PUSHL   SEM_LIST                                         ; 1005
                 FF2F     CF         05   FB 00047          CALLS   #5, NML$DEFPARAM                                 ;
                                        04 0004C          RET                                                      ; 1013

; Routine Size:  77 bytes,    Routine Base:  $CODE$ + 0395
```

```
1 %SBTTL 'NML$DEF_NODE_ADDR   Add node address parameter'
1 GLOBAL ROUTINE NML$DEF_NODE_ADDR (SEM_LIST, BUFDSC, LENGTH, ADDR, RTNDSC)=
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !       This routine checks the node address parameter to make sure
1 !       it does not already exits in the node permanent database.  If it does
1 !       not, it adds the node address to the permanent data base record.
1 !       This routine is not used to check for duplicate node names because
1 !       the node database name key is declared as "noduplicates", so RMS
1 !       will do this check for node names when the record is written to
1 !       the file.
1 !
1 ! FORMAL PARAMETERS:
1 !       SEM_LIST        Parameter semantic table entry address.
1 !       BUFSIZE         Permanent database record maximum size.
1 !       LENGTH          Length of parameter to insert in record.
1 !       ADDR            Address of parameter to insert in record.
1 !       RTNDSC          Permanent database record buffer descriptor address.
1 !
1 ! IMPLICIT INPUTS:
1 !       It is assumed that the permanent data base file is already open.
1 !
1 ! IMPLICIT OUTPUTS:
1 !       The parameter is added to the record.
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !       Returns success (NML$_STS_SUC) if the node address is successfully
1 !               added to the permanent database record.
1 !       Returns nml$_sts_pva if the new address is already defined in the
1 !               node permanent database.
1 !
1 ! SIDE EFFECTS:
1 !       NONE
1 !
1 !--
1
2 BEGIN
2
2 MAP
2     sem_list    : REF BBLOCK,
2     rtndsc      : REF DESCRIPTOR;
2
2 LOCAL
2     status;
2
2 !
2 ! If there's another node in the permanent database with the new address,
2 ! return an error message to NCP.
2 !
2 IF nml_find_duplicate_node (.sem_list, .bufdsc,
2                             .length, .addr,
2                             .rtndsc) THEN
3     BEGIN
3     nml$ab_msgblock [msb$v_det_fld] = 1;
3     nml$ab_msgblock [msb$b_code] = nma$c_sts_pva;
```

```
; 1084          1071  3      nml$ab_msgblock [msb$w_detail] = .sem_list [pst$w_dataid];
; 1085          1072  3      RETURN nml$_sts_pva
; 1086          1073  2      END;
; 1087          1074  2
; 1088          1075  2 !
; 1089          1076  2 ! The node address is unique.  Add it to the node's permanent database record.
; 1090          1077  2 !
; 1091          1078  2 status = nml$defparam (.sem_list,
; 1092          1079  2                          .bufdsc,
; 1093          1080  2                          .length,
; 1094          1081  2                          .addr,
; 1095          1082  2                          .rtndsc);
; 1096          1083  2
; 1097          1084  2 RETURN .status
; 1098          1085  2
; 1099          1086  1 END;                                     ! End of NML$DEF_NODE_ADDR




                                              0004 00000          .ENTRY   NML$DEF_NODE_ADDR, Save R2                          ; 1015
                      52 00000000G   00    9E 00002          MOVAB    NML$AB_MSGBLOCK, R2                                 ;
                      7E         10  AC    7D 00009          MOVQ     ADDR, -(SP)                                         ; 1066
                      7E         08  AC    7D 0000D          MOVQ     BUFDSC, -(SP)                                       ; 1065
                                 04  AC    DD 00011          PUSHL    SEM_LIST                                            ;
            00000000V 00             05    FB 00014          CALLS    #5, NML_FIND_DUPLICATE_NODE                         ;
                      10             50    E9 0001B          BLBC     R0, 1$                                              ;
                      62             02    88 0001E          BISB2    #2, NML$AB_MSGBLOCK                                 ; 1069
                 04   A2             10    8E 00021          MNEGB    #16, NML$AB_MSGBLOCK+4                              ; 1070
                 08   A2         04  BC    B0 00025          MOVW     @SEM_LIST, NML$AB_MSGBLOCK+8                        ; 1071
                      50             20    CE 0002A          MNEGL    #32, R0                                             ; 1072
                                           04 0002D          RET                                                          ;
                      7E         10  AC    7D 0002E 1$:      MOVQ     ADDR, -(SP)                                         ; 1081
                      7E         08  AC    7D 00032          MOVQ     BUFDSC, -(SP)                                       ; 1079
                                 04  AC    DD 00036          PUSHL    SEM_LIST                                            ; 1078
               FEF0   CF             05    FB 00039          CALLS    #5, NML$DEFPARAM                                    ;
                                           04 0003E          RET                                                          ; 1086

; Routine Size:  63 bytes,    Routine Base:  $CODE$ + 03E2
```

```
1 %SBTTL 'NML$DEF_EXEC_ID   Add executor name or address parameter'
1 GLOBAL ROUTINE NML$DEF_EXEC_ID (SEM_LIST, BUFDSC, LENGTH, ADDR, RTNDSC)=
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !       This routine is called when processing a DEFINE EXECUTOR command
1 !       to change the name or address of the executor node.  It checks
1 !       the new name or address parameter to determine if it's already
1 !       assigned to some other node.  If it is, this means the identity
1 !       of the executor is being changed.  Delete the remote entry with
1 !       that name or address.  The new name or address is added to the
1 !       executor node permanent database record.  It is written back
1 !       to the file later.
1 !
1 ! FORMAL PARAMETERS:
1 !       SEM_LIST        Parameter semantic table entry address.
1 !       BUFSIZE         Permanent database record maximum size.
1 !       LENGTH          Length of parameter to insert in record.
1 !       ADDR            Address of parameter to insert in record.
1 !       RTNDSC          Permanent database record buffer descriptor address.
1 !
1 ! IMPLICIT INPUTS:
1 !       It is assumed that the permanent data base file is already open.
1 !
1 ! IMPLICIT OUTPUTS:
1 !       The new executor name or address is added to the record.
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !       Returns success (NML$_STS_SUC) if the node address is successfully
1 !               added to the permanent database record.
1 !
1 ! SIDE EFFECTS:
1 !       If the new executor name or address is already assigned to some
1 !       other node in the permanent database, that remote node is deleted from
1 !       the database.
1 !
1 !--
1
2 BEGIN
2
2 MAP
2     addr        : REF BBLOCK [2],
2     sem_list    : REF BBLOCK;
2
2 LOCAL
2     status,
2     temp;
2
2 IF nml_find_duplicate_node (.sem_list, .bufdsc,
2                             .length, .addr,
2                             .rtndsc) THEN
3     BEGIN
3     !
3     ! The executor node identity is being changed to that of a node that's
3     ! already in the database.  Delete the remote entry for that node (there
3     ! are no parameters that it makes sense to carry over in this case)
```

```
; 1158          1144  3       ! so the executor can become that node.
; 1159          1145  3       !
; 1160          1146  3       nml$delete_node_rec (.sem_list [pst$w_dataid],      ! Database key
; 1161          1147  3                           length);                       ! Name or address dsc.
; 1162          1148  3       nml$ab_msgblock [msb$v_msg_fld] = 1;
; 1163          1149  3       nml$ab_msgblock [msb$l_text] = nml$_recdelet;
; 1164          1150  2       END;
; 1165          1151  2   !
; 1166          1152  2   ! Put the RMS "current record" pointer back to the executor node's
; 1167          1153  2   ! entry.
; 1168          1154  2   !
; 1169          1155  2   !******************************TEMPORARY
; 1170          1156  2   nml$gw_perm_exec_addr = 0;
; 1171          1157  2   !*************************************
; 1172          1158  2   nml$getexeadr (temp);
; 1173          1159  2   !
; 1174          1160  2   ! If the new executor address is 0, leave it that way.  If the area number
; 1175          1161  2   ! of the address is 0, then default it to area 1 (this is for DEFINE EXEC
; 1176          1162  2   ! ADDRESS only) so the exec will have a valid area number in the database.
; 1177          1163  2   !
; 1178          1164  2   IF .sem_list [pst$w_dataid] EQL nma$c_pcno_add THEN
; 1179          1165  3       BEGIN
; 1180          1166  3       IF .addr [nma$v_addr] NEQ 0 AND
; 1181          1167  3           .addr [nma$v_area] EQL 0 THEN
; 1182          1168  3           addr [nma$v_area] = 1;
; 1183          1169  2       END;
; 1184          1170  2   status = nml$defparam (.sem_list,
; 1185          1171  2                          .bufdsc,
; 1186          1172  2                          .length,
; 1187          1173  2                          .addr,
; 1188          1174  2                          .rtndsc);
; 1189          1175  2
; 1190          1176  2   IF .sem_list [pst$w_dataid] EQL nma$c_pcno_add THEN
; 1191          1177  2       nml$gw_perm_exec_addr = .(.addr)<0,16>
; 1192          1178  2   ELSE
; 1193          1179  3       BEGIN
; 1194          1180  3       CH$MOVE (.length, .addr, .nml$gq_perm_exec_name_dsc [1]);
; 1195          1181  3       nml$gq_perm_exec_name_dsc [0] = .length;
; 1196          1182  2       END;
; 1197          1183  2   RETURN .status
; 1198          1184  2
; 1199          1185  1   END;                                        ! End of NML$DEF_EXEC_ID



                                       00FC 00000         .ENTRY   NML$DEF_EXEC_ID, Save R2,R3,R4,R5,R6,R7       ; 1088
                 57 00000000G     00     9E 00002         MOVAB    NML$GW_PERM_EXEC_ADDR, R7                     ;
                 5E               04     C2 00009         SUBL2    #4, SP                                        ;
                          14      AC     DD 0000C         PUSHL    RTNDSC                                        ; 1138
                 52       10      AC     D0 0000F         MOVL     ADDR, R2                                      ; 1137
                                  52     DD 00013         PUSHL    R2                                            ;
                 7E       08      AC     7D 00015         MOVQ     BUFDSC, -(SP)                                 ; 1136
                 53       04      AC     D0 00019         MOVL     SEM_LIST, R3                                  ;
                                  53     DD 0001D         PUSHL    R3                                            ;
      00000000V  00               05     FB 0001F         CALLS    #5, NML_FIND_DUPLICATE_NODE                   ;
```

```
                              1F            50 E9 00026         BLBC    R0, 1$                                  ;
                                      0C    AC 9F 00029         PUSHAB  LENGTH                                  ; 1146
                              7E            63 3C 0002C         MOVZWL  (R3), -(SP)                             ;
                   00000000G  00            02 FB 0002F         CALLS   #2, NML$DELETE_NODE_REC                 ;
                   00000000G  00            04 88 00036         BISB2   #4, NML$AB_MSGBLOCK                     ; 1148
                   00000000G  00 00000000G  8F D0 0003D         MOVL    #NML$_RECDELET, NML$AB_MSGBLOCK+12      ; 1149
                                            67 D4 00048 1$:     CLRL    NML$GQ_PERM_EXEC_ADDR                   ; 1156
                                            5E DD 0004A         PUSHL   SP                                      ; 1158
                   00000000G  00            01 FB 0004C         CALLS   #1, NML$GETEXEADR                       ;
                        01F6  8F            63 B1 00053         CMPW    (R3), #502                              ; 1164
                                            13 12 00058         BNEQ    2$                                      ;
                        03FF  8F            62 B3 0005A         BITW    (R2), #1023                             ; 1166
                                            0C 13 0005F         BEQL    2$                                      ;
                          FC  8F      01    A2 93 00061         BITB    1(R2), #252                             ; 1167
                                            05 12 00066         BNEQ    2$                                      ;
        62           06       0A            01 F0 00068         INSV    #1, #10, #6, (R2)                       ; 1168
                                      14    AC DD 0006D 2$:     PUSHL   RTNDSC                                  ; 1174
                                            52 DD 00070         PUSHL   R2                                      ; 1173
                              7E      08    AC 7D 00072         MOVQ    BUFDSC, -(SP)                           ; 1171
                                            53 DD 00076         PUSHL   R3                                      ; 1170
                        FE72  CF            05 FB 00078         CALLS   #5, NML$DEFPARAM                        ;
                              56            50 D0 0007D         MOVL    R0, STATUS                              ;
                        01F6  8F            63 B1 00080         CMPW    (R3), #502                              ; 1176
                                            05 12 00085         BNEQ    3$                                      ;
                              67            62 3C 00087         MOVZWL  (R2), NML$GW_PERM_EXEC_ADDR             ; 1177
                                            14 11 0008A         BRB     4$                                      ;
                              50 00000000G  00 D0 0008C 3$:     MOVL    NML$GQ_PERM_EXEC_NAME_DSC+4, R0         ; 1180
                     60       62      0C    AC 28 00093         MOVC3   LENGTH, (R2), (R0)                      ;
                   00000000G  00      0C    AC D0 00098         MOVL    LENGTH, NML$GQ_PFRM_EXEC_NAME_DSC       ; 1181
                              50            56 D0 000A0 4$:     MOVL    STATUS, R0                              ; 1183
                                               04 000A3         RET                                             ; 1185

; Routine Size:  164 bytes,    Routine Base:  $CODE$ + 0421
; 1201          1186  1 %SBTTL 'NML_FIND_DUPLICATE_NODE  Check perm db for node id'
; 1202          1187  1 ROUTINE NML_FIND_DUPLICATE_NODE (SEM_LIST, BUFDSC,
; 1203          1188  1                                 LENGTH, ADDR,
; 1204          1189  1                                 RTNDSC)=
; 1205          1190  1
; 1206          1191  1 !++
; 1207          1192  1 ! FUNCTIONAL DESCRIPTION:
; 1208          1193  1 !       This routine checks the node name or address parameter to see
; 1209          1194  1 !       if it already exists in the node permanent database.
; 1210          1195  1 !
; 1211          1196  1 ! FORMAL PARAMETERS:
; 1212          1197  1 !
; 1213          1198  1 !       SEM_LIST        Parameter semantic table entry address.
; 1214          1199  1 !       BUFSIZE         Permanent database record maximum size.
; 1215          1200  1 !       LENGTH          Length of parameter to insert in record.
; 1216          1201  1 !       ADDR            Address of parameter to insert in record.
; 1217          1202  1 !       RTNDSC          Permanent database record buffer descriptor address.
; 1218          1203  1 !
; 1219          1204  1 ! IMPLICIT INPUTS:
; 1220          1205  1 !       It is assumed that the permanent data base file is already open.
; 1221          1206  1 !
; 1222          1207  1 ! IMPLICIT OUTPUTS:
; 1223          1208  1 !       NML$Q_PRMDSC is the descriptor of the duplicate node's record
; 1224          1209  1 !       (if there is one) which is used to return the ID of that node
; 1225          1210  1 !       in the NICE error message.
; 1226          1211  1 !
; 1227          1212  1 ! ROUTINE VALUE:
; 1228          1213  1 ! COMPLETION CODES:
; 1229          1214  1 !       Returns status of node lookup.
; 1230          1215  1 !
; 1231          1216  1 ! SIDE EFFECTS:
; 1232          1217  1 !       None
; 1233          1218  1 !
; 1234          1219  1 !--
; 1235          1220  1
; 1236          1221  2 BEGIN
; 1237          1222  2
; 1238          1223  2 MAP
; 1239          1224  2     sem_list : REF BBLOCK;
; 1240          1225  2
; 1241          1226  2 LOCAL
; 1242          1227  2     key,
; 1243          1228  2     node_id_dsc: VECTOR [2],
; 1244          1229  2     dup_dsc:     VECTOR [2],
; 1245          1230  2     node_type,
; 1246          1231  2     status;
; 1247          1232  2
; 1248          1233  2 !
; 1249          1234  2 ! Look for a node name (or address) that was previously DEFINEd in the node's
; 1250          1235  2 ! permanent database record.
; 1251          1236  2 !
; 1252          1237  2 node_id_dsc [1] = 0;
; 1253          1238  2 node_id_dsc [0] = 0;
; 1254          1239  2 status = nma$searchfld (.rtndsc,
; 1255          1240  2                         .sem_list [pst$w_dataid],
; 1256          1241  2                         node_id_dsc [0],
; 1257          1242  2                         node_id_dsc [1]);
; 1258          1243  2
; 1259          1244  2 !
; 1260          1245  2 ! If there is no previously defined node ID, or the previous ID is different
; 1261          1246  2 ! from the new ID in the NICE DEFINE command, then check to see if there's
; 1262          1247  2 ! another node with the same name or address in the node permanent database.
; 1263          1248  2 !
; 1264          1249  2 IF NOT .status
; 1265          1250  2        OR
; 1266          1251  3    (.status AND
; 1267          1252  2    CH$NEQ (.node_id_dsc [0], .node_id_dsc [1], .length, .addr)) THEN
; 1268          1253  3     BEGIN
; 1269          1254  3     key = .sem_list [pst$w_dataid];               ! Make key a longword.
; 1270          1255  3     status = nml$readrecord (nma$c_opn_node,      ! Node database file ID
; 1271          1256  3                              key,                 ! Node database key
; 1272          1257  3                              length,              ! Address of key value descriptor
; 1273          1258  3                              nml$q_prmdsc,        ! Buffer for node record
; 1274          1259  3                              dup_dsc,             ! Duplicate node data descriptor
; 1275          1260  3                              node_type);          ! Node entity type.
; 1276          1261  3     IF .status THEN
; 1277          1262  4         BEGIN
; 1278          1263  4         !
; 1279          1264  4         ! There is another node with the new name or address DEFINEd.
; 1280          1265  4         ! Add duplicate node id to NICE response message parameters.  The node
; 1281          1266  4         ! ID will be returned in the NICE response to NCP.
; 1282          1267  4         !
; 1283          1268  4         nml$q_entbfdsc [0] = nml$k_entbuflen;
; 1284          1269  4         nml$q_entbfdsc [1] = nml$t_entbuffer;
; 1285          1270  4         nml$getrecowner (dup_dsc,
; 1286          1271  4                          .node_type,
; 1287          1272  4                          nml$q_entbfdsc,
; 1288          1273  4                          nml$q_entbfdsc [0]);
; 1289          1274  4         nml$ab_msgblock [msb$l_flags] = msb$m_entd_fld;  ! Set entit, descriptor flag
; 1290          1275  4         nml$ab_msgblock [msb$a_entity] = nml$q_entbfdsc; ! Add entity descriptor pointer
; 1291          1276  3         END;
; 1292          1277  3     END
; 1293          1278  2 ELSE
; 1294          1279  2     status = nml$_sts_cmp;
; 1295          1280  2 RETURN .status
; 1296          1281  1 END;                                     ! End of NML_FIND_DUPLICATE_NODE



                                                 003C 00000 NML_FIND_DUPLICATE_NODE:
                                                                  .WORD   Save R2,R3,R4,R5                          ; 1187
                                 55 00000000'  00  9E 00002       MOVAB   NML$Q_ENTBFDSC, R5                        ;
                                 5E            18  C2 00009       SUBL2   #24, SP                                   ;
                                         10    AE  7C 0000C       CLRQ    NODE_ID_DSC                               ; 1238
                                         14    AE  9F 0000F       PUSHAB  NODE_ID_DSC+4                             ; 1242
                                         14    AE  9F 00012       PUSHAB  NODE_ID_DSC                               ; 1241
                                 7E      04    BC  3C 00015       MOVZWL  @SEM_LIST, -(SP)                          ; 1240
                                         14    AC  DD 00019       PUSHL   RTNDSC                                    ; 1239
                        00000000G  00          04  FB 0001C       CALLS   #4, NMA$SEARCHFLD                         ;
                                 54            50  DC 00023       MOVL    R0, STATUS                                ;
                                 0C            54  E9 00026       BLBC    STATUS, 1$                                ; 1249
     0C    AC          00      14  BE      10    AE  2D 00029       CMPC5   NODE_ID_DSC, @NODE_ID_DSC+4, #0, LENGTH, -  ; 1252
```

```
                                                      10  BC     00031                 @ADDR                               ;
                                                          4F  13 00033         BEQL    2$                                  ;
                                       04  AE         04  BC  3C 00035 1$:     MOVZWL  @SEM_LIST, KEY                      ; 1254
                                                          5E  DD 0003A         PUSHL   SP                                  ; 1255
                                                      0C  AE  9F 0003C         PUSHAB  DUP_DSC                             ;
                                               00000000'  00  9F 0003F         PUSHAB  NML$Q_PRMDSC                        ;
                                                      0C  AC  9F 00045         PUSHAB  LENGTH                              ;
                                                      14  AE  9F 00048         PUSHAB  KEY                                 ;
                                                          7E  D4 0004B         CLRL    -(SP)                               ;
                                00000000G  00             06  FB 0004D         CALLS   #6, NML$READRECORD                  ;
                                           54             50  D0 00054         MOVL    R0, STATUS                          ;
                                           2D             54  E9 00057         BLBC    STATUS, 3$                          ; 1261
                                           65         40  8F  9A 0005A         MOVZBL  #64, NML$Q_ENTBFDSC                 ; 1268
                                       04  A5         C0  A5  9E 0005E         MOVAB   NML$T_ENTBUFFER, NML$Q_ENTBFDSC+4   ; 1269
                                                          55  DD 00063         PUSHL   R5                                  ; 1273
                                                          55  DD 00065         PUSHL   R5                                  ; 1270
                                                      08  AE  DD 00067         PUSHL   NODE_TYPE                           ; 1271
                                                      14  AE  9F 0006A         PUSHAB  DUP_DSC                             ; 1270
                                00000000G  00             04  FB 0006D         CALLS   #4, NML$GETRECOWNER                 ;
                                00000000G  00             10  D0 00074         MOVL    #16, NML$AB_MSGBLOCK                ; 1274
                                00000000G  00             65  9E 0007B         MOVAB   NML$Q_ENTBFDSC, NML$AB_MSGBLOCK+20  ; 1275
                                                          03  11 00082         BRB     3$                                  ; 1249
                                           54             10  CE 00084 2$:     MNEGL   #16, STATUS                         ; 1279
                                           50             54  D0 00087 3$:     MOVL    STATUS, R0                          ; 1280
                                                              04 0008A         RET                                         ; 1281

; Routine Size:  139 bytes,    Routine Base:  $CODE$ + 04C5
```

```
1 %SBTTL 'NML$DE;NODNLI  Add loop node l'ne parameter'
1 GLOBAL ROUTINE NML$DEFNODNLI (SEM_LIST, BUFDSC, LENGTH, ADDR, RTNDSC)=
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine adds the loop node line parameter to the permanent
1 !       data base record if this is a loop node and the circuit id is
1 !       unique (i.e. there is no other loop node set up on the circuit).
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       SEM_LIST        Parameter semantic table entry address.
1 !       BUFSIZE         Permanent database record maximum size.
1 !       LENGTH          Length of parameter to insert in record.
1 !       ADDR            Address of parameter to insert in record.
1 !       RTNDSC          Permanent database record buffer descriptor address.
1 !
1 ! IMPLICIT INPUTS:
1 !       It is assumed that the permanent data base file is already open.
1 !
1 ! IMPLICIT OUTPUTS:
1 !       The parameter is added to the record.
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !       Always returns success (NML$_STS_SUC).
1 !
1 ! SIDE EFFECTS:
1 !       NONE
1 !
1 !--
1
2 BEGIN
2
2 MAP
2     sem_list : REF BBLOCK;
2
2 LOCAL
2     fldadr,
2     fldsize,
2     circuit_dsc:        VECTOR [2],     ! Circuit already in node record (if any)
2     node_rec_buf:       BBLOCK [nml$k_recbflen], ! Buffer for node data
2     node_rec_dsc:       VECTOR [2],     ! Descriptor of node record buffer.
2     node_rec_data:      VECTOR [2],     ! Descriptor of data in node record buffer.
2     status;
2
2 fldadr = 0;
2 IF nma$searchfld (.rtndsc,
2                         nma$c_pcno_add,
2                         fldsize,
2                         fldadr) THEN
3     BEGIN
3     !
3     ! Node has address so circuit is not allowed.  Loopnodes have only one
3     ! parameter - a circuit ID.
3     !
```

```
    nml$ab_msgblock [msb$l_flags] = msb$m_det_fld;
    nml$ab_msgblock [msb$b_code] = nma$c_sts_pna;
    nml$ab_msgblock [msb$w_detail] = nma$c_pcno_nli;
    RETURN nml$_sts_pna
    END;

circuit_dsc [0] = 0;
circuit_dsc [1] = 0;
status = nma$searchfld (.rtndsc,
                        nma$c_pcno_nli,
                        circuit_dsc [0],
                        circuit_dsc [1]);
!
! If the loop node is already set up on the circuit specified in the NICE
! DEFINE command, I'm done.  Otherwise, make sure the circuit isn't already
! defined for some other loopnode.
!
IF NOT .status
OR (.status AND CH$NEQ (.circuit_dsc [0], .circuit_dsc [1],
                        .length, .addr)) THEN
    BEGIN
    !
    ! Check to make sure there aren't any other loopnodes on the specified
    ! circuit in the node database.
    !
    node_rec_dsc [0] = nml$k_recbflen;
    node_rec_dsc [1] = node_rec_buf;
    node_rec_data [1] = node_rec_buf;
    status = nml$read_loopnode (length,         ! Address of circuit descriptor
                          node_rec_dsc,         ! I/O buffer descriptor
                          node_rec_data);       ! Return node data descriptor
    IF .status NEQ rms$_eof THEN
        BEGIN
        !
        ! Circuit name must be unique for loop node.
        !
        nml$q_entbfdsc [0] = nml$k_entbuflen;
        nml$q_entbfdsc [1] = nml$t_entbuffer;
        nml$getrecowner (node_rec_data,
                        nml$c_loopnode,
                        nml$q_entbfdsc,
                        nml$q_entbfdsc [0]);
        nml$ab_msgblock [msb$a_entity] = nml$q_entbfdsc; ! Add entity descriptor pointer
        nml$ab_msgblock [msb$l_flags] = msb$m_det_fld OR msb$m_entd_fld;
        nml$ab_msgblock [msb$b_code] = nma$c_sts_pva;
        nml$ab_msgblock [msb$w_detail] = nma$c_pcno_nli;
        RETURN nml$_sts_pva
        END;
    END;
!
! The circuit is not already DEFINEd for some other loopnode.  Add it to
! the node's permanent database record.
!
status = nml$defparam (.sem_list,
                           .bufdsc,
                           .length,
                           .addr,
; 1412          1396  2                              .rtndsc);
; 1413          1397  2 RETURN .status
; 1414          1398  1 END;                                     ! End of NML$DEFNODNLI



                                                      00FC  00000          .ENTRY   NML$DEFNODNLI, Save R2,R3,R4,R5,R6,R7           ; 1283
                                    57  00000000G   00  9E  00002          MOV'AB   NMA$SEARCHFLD, R7                               ;
                                    56  00000000'   00  9E  00009          MOVAB    NML$Q_ENTBFDSC, R6                              ;
                                    55  00000000G   00  9E  00010          MOVAB    NML$AB_MSGBLOCK, R5                             ;
                                    5E       FBE4   CE  9E  00017          MOVAB    -1052(SP), SP                                   ;
                                                    7E  D4  0001C          CLRL     FLDADR                                          ; 1329
                                                    5E  DD  0001E          PUSHL    SP                                              ; 1330
                                               08   AE  9F  00020          PUSHAB   FLDSIZE                                         ;
                                    7E       01F6   8F  3C  00023          MOVZWL   #502, -(SP)                                     ;
                                               14   AC  DD  00028          PUSHL    RTNDSC                                          ;
                                    67              04  FB  0002B          CALLS    #4, NMA$SEARCHFLD                               ;
                                    11              50  E9  0002E          BLBC     R0, 1$                                          ;
                                    65              02  D0  00031          MOVL     #2, NML$AB_MSGBLOCK                             ; 1339
                               04   A5              16  8E  00034          MNEGB    #22, NML$AB_MSGBLOCK+4                          ; 1340
                               08   A5       01F5   8F  B0  00038          MOVW     #501, NML$AB_MSGBLOCK+8                         ; 1341
                                    50              2C  CE  0003E          MNEGL    #44, R0                                         ; 1342
                                                        04  00041          RET                                                      ;
                                               F8   AD  7C  00042 1$:      CLRQ     CIRCUIT_DSC                                     ; 1345
                                               FC   AD  9F  00045          PUSHAB   CIRCUIT_DSC+4                                   ; 1350
                                               F8   AD  9F  00048          PUSHAB   CIRCUIT_DSC                                     ; 1349
                                    7E       01F5   8F  3C  0004B          MOVZWL   #501, -(SP)                                     ; 1347
                                               14   AC  DD  00050          PUSHL    RTNDSC                                          ;
                                    67              04  FB  00053          CALLS    #4, NMA$SEARCHFLD                               ;
                                    54              50  D0  00056          MOVL     R0, STATUS                                      ;
                                    0C              54  E9  00059          BLBC     STATUS, 2$                                      ; 1356
  0C     AC          00        FC   BD         F8   AD  2D  0005C          CMPC5    CIRCUIT_DSC, @CIRCUIT_DSC+4, #0, LENGTH, -      ; 1357
                                               10   BC      00064                   @ADDR                                           ;
                                                    5A  13  00066          BEQL     3$                                              ;
                               10   AE       0400   8F  3C  00068 2$:      MOVZWL   #1024, NODE_REC_DSC                             ; 1364
                               14   AE         18   AE  9E  0006E          MOVAB    NODE_REC_BUF, NODE_REC_DSC+4                    ; 1365
                               0C   AE         18   AE  9E  00073          MOVAB    NODE_REC_BUF, NODE_REC_DATA+4                   ; 1366
                                               08   AE  9F  00078          PUSHAB   NODE_REC_DATA                                   ; 1367
                                               14   AE  9F  0007B          PUSHAB   NODE_REC_DSC                                    ;
                                               0C   AC  9F  0007E          PUSHAB   LENGTH                                          ;
                        00000000G   00              03  FB  00081          CALLS    #3, NML$READ_LOOPNODE                           ;
                                    54              50  D0  00088          MOVL     R0, STATUS                                      ;
                         0001827A   8F              54  D1  0008B          CMPL     STATUS, #98938                                  ; 1370
                                                    2E  13  00092          BEQL     3$                                              ;
                                    66         40   8F  9A  00094          MOVZBL   #64, NML$Q_ENTBFDSC                             ; 1375
                               04   A6         C0   A6  9E  00098          MOVAB    NML$T_ENTBUFFER, NML$Q_ENTBFDSC+4               ; 1376
                                                    56  DD  0009D          PUSHL    R6                                              ; 1380
                                                    56  DD  0009F          PUSHL    R6                                              ; 1377
                                                    05  DD  000A1          PUSHL    #5                                              ;
                                               14   AE  9F  000A3          PUSHAB   NODE_REC_DATA                                   ;
                        00000000G   00              04  FB  000A6          CALLS    #4, NML$GETRECOWNER                             ;
                               14   A5              66  9E  000AD          MOVAB    NML$Q_ENTBFDSC, NML$AB_MSGBLOCK+20              ; 1381
                                    65              12  D0  000B1          MOVL     #18, NML$AB_MSGBLOCK                            ; 1382
                               04   A5              10  8E  000B4          MNEGB    #16, NML$AB_MSGBLOCK+4                          ; 1383
                               08   A5       01F5   8F  B0  000B8          MOVW     #501, NML$AB_MSGBLOCK+8                         ; 1384
                        50          20  CE 000BE          MNEGL   #32, R0                                          ; 1385
                                        04 000C1          RET                                                      ;
                        7E      10  AC  7D 000C2 3$:      MOVQ    ADDR, -(SP)                                      ; 1395
                        7E      08  AC  7D 000C6          MOVQ    BUFDSC, -(SP)                                    ; 1393
                                04  AC  DD 000CA          PUSHL   SEM_LIST                                         ; 1392
                   FCEE CF          05  FB 000CD          CALLS   #5, NML$DEFPARAM                                 ;
                        54          50  D0 000D2          MOVL    R0, STATUS                                       ;
                                        04 000D5          RET                                                      ; 1398

; Routine Size:  214 bytes,    Routine Base:  $CODE$ + 0550
```

```
1 %SBTTL 'NML$DEFOBJNUM  Add object number parameter'
1 GLOBAL ROUTINE NML$DEFOBJNUM (SEM_LIST, BUFDSC, LENGTH, ADDR, RTNDSC)=
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine adds the object number parameter to the permanent
1 !       data base record if it is unique.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       SEM_LIST        Parameter semantic table entry address.
1 !       BUFSIZE         Permanent database record maximum size.
1 !       LENGTH          Length of parameter to insert in record.
1 !       ADDR            Address of parameter to insert in record.
1 !       RTNDSC          Permanent database record buffer descriptor address.
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       It is assumed that the permanent data base file is already open.
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       The parameter is added to the record.
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       Always returns success (NML$_STS_SUC).
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !
1 !--
1
2     BEGIN
2
2     MAP
2         SEM_LIST : REF BBLOCK;
2
2     LOCAL
2         DUMDSC : DESCRIPTOR,
2         FLDADR,
2         FLDSIZE,
2         KEY : WORD,
2         STATUS;
2
2     FLDADR = 0;
2     FLDSIZE = 0;
2     STATUS = NMA$SEARCHFLD (.RTNDSC,
2                             NMA$C_PCOB_NUM,
2                             FLDSIZE,
2                             FLDADR);
2
2     !
2     ! If no object number is already defined or the object number is
```

```
; 1473      1456  2      !             changed by the command, and
; 1474      1457  2      ! the object number is not zero (duplicate objects numbered 0 are allowed),
; 1475      1458  2      ! make sure that the new object number is not already in the
; 1476      1459  2      ! permanent data base.
; 1477      1460  2      !
; 1478      1461  3      IF (NOT .STATUS
; 1479      1462  3          OR (.STATUS AND CH$NEQ (.FLDSIZE, .FLDADR, .LENGTH, .ADDR)))
; 1480      1463  2      AND CH$NEQ (.LENGTH, UPLIT(0), .LENGTH, .ADDR)
; 1481      1464  2      THEN
; 1482      1465  3          BEGIN
; 1483      1466  3
; 1484      1467  3          KEY = 0;
; 1485      1468  3          IF NMA$MATCHREC (NMA$C_OPN_OBJ,
; 1486      1469  3                           NML$Q_PRMDSC,
; 1487      1470  3                           KEY,
; 1488      1471  3                           NMA$C_PCOB_NUM,
; 1489      1472  3                           .LENGTH,
; 1490      1473  3                           .ADDR,
; 1491      1474  3                           DUMDSC)
; 1492      1475  3          THEN
; 1493      1476  4              BEGIN
; 1494      1477  4  !
; 1495      1478  4  ! Object number is not unique.
; 1496      1479  4  !
; 1497      1480  4              NML$AB_MSGBLOCK [MSB$L_FLAGS] = MSB$M_DET_FLD;
; 1498      1481  4              NML$AB_MSGBLOCK [MSB$B_CODE] = NMA$C_STS_PVA;
; 1499      1482  4              NML$AB_MSGBLOCK [MSB$W_DETAIL] = NMA$C_PCOB_NUM;
; 1500      1483  4
; 1501      1484  4              RETURN NML$_STS_PVA
; 1502      1485  4
; 1503      1486  3              END;
; 1504      1487  2          END;
; 1505      1488  2
; 1506      1489  2      STATUS = NML$DEFPARAM (.SEM_LIST,
; 1507      1490  2                             .BUFDSC,
; 1508      1491  2                             .LENGTH,
; 1509      1492  2                             .ADDR,
; 1510      1493  2                             .RTNDSC);
; 1511      1494  2
; 1512      1495  2      RETURN .STATUS
; 1513      1496  2
; 1514      1497  1      END;                                          ! End of NML$DEFOBJNUM


                                                                      .PSECT  $PLIT$,NOWRT,NOEXE,2

                                                      00031           .BLKB   3
                                            00000000  00034 P.AAG:    .LONG   0                              ;


                                                                      .PSECT  $CODE$,NOWRT,2

                                                  003C 00000          .ENTRY  NML$DEFOBJNUM, Save R2,R3,R4,R5    ; 1400
                                 55 00000000G  00   9E 00002          MOVAB   NML$AB_MSGBLOCK, R5                ;
                                 5E            10   C2 00009          SUBL2   #16, SP                            ;
```

```
                                                     7E  D4 0000C         CLRL    FLDADR                                      ; 1447
                                                04   AE  D4 0000E         CLRL    FLDSIZE                                     ; 1448
                                                     5E  DD 00011         PUSHL   SP                                          ; 1449
                                                08   AE  9F 00013         PUSHAB  FLDSIZE                                     ;
                                       7E     0201   8F  3C 00016         MOVZWL  #513, -(SP)                                 ;
                                                14   AC  DD 0001B         PUSHL   RTNDSC                                      ;
                            00000000G  00            04  FB 0001E         CALLS   #4, NMA$SEARCHFLD                           ;
                                       54            50  D0 00025         MOVL    R0, STATUS                                  ;
                                       0C            54  E9 00028         BLBC    STATUS, 1$                                  ; 1461
      0C   AC            00       00   BE       04   AE  2D 0002B         CMPC5   FLDSIZE, @FLDADR, #0, LENGTH, @ADDR         ; 1462
                                                10   BC     00033                                                             ;
                                                     41  13 00035         BEQL    2$                                          ;
                    10   BC 00000000'  00       0C   AC  29 00037 1$:     CMPC3   LENGTH, P.AAG, @ADDR                        ; 1463
                                                     35  13 00041         BEQL    2$                                          ;
                                                08   AE  B4 00043         CLRW    KEY                                         ; 1467
                                                0C   AE  9F 00046         PUSHAB  DUMDSC                                      ; 1468
                                       7E       0C   AC  7D 00049         MOVQ    LENGTH, -(SP)                               ; 1472
                                       7E     0201   8F  3C 0004D         MOVZWL  #513, -(SP)                                 ; 1468
                                                18   AE  9F 00052         PUSHAB  KEY                                         ;
                                          00000000'  00  9F 00055         PUSHAB  NML$Q_PRMDSC                                ;
                                                     03  DD 0005B         PUSHL   #3                                          ;
                            00000000G  00            07  FB 0005D         CALLS   #7, NMA$MATCHREC                            ;
                                       11            50  E9 00064         BLBC    R0, 2$                                      ;
                                       65            02  D0 00067         MOVL    #2, NML$AB_MSGBLOCK                         ; 1480
                                  04   A5            10  8E 0006A         MNEGB   #16, NML$AB_MSGBLOCK+4                      ; 1481
                                  08   A5     0201   8F  B0 0006E         MOVW    #513, NML$AB_MSGBLOCK+8                     ; 1482
                                       50            20  CE 00074         MNEGL   #32, R0                                     ; 1484
                                                         04 00077         RET                                                 ;
                                       7E       10   AC  7D 00078 2$:     MOVQ    ADDR, -(SP)                                 ; 1492
                                       7E       08   AC  7D 0007C         MOVQ    BUFDSC, -(SP)                               ; 1490
                                                04   AC  DD 00080         PUSHL   SEM_LIST                                    ; 1489
                                FC62   CF            05  FB 00083         CALLS   #5, NML$DEFPARAM                            ;
                                       54            50  D0 00088         MOVL    R0, STATUS                                  ;
                                                         04 0008B         RET                                                 ; 1497

; Routine Size:  140 bytes,    Routine Base:  $CODE$ + 0626
```

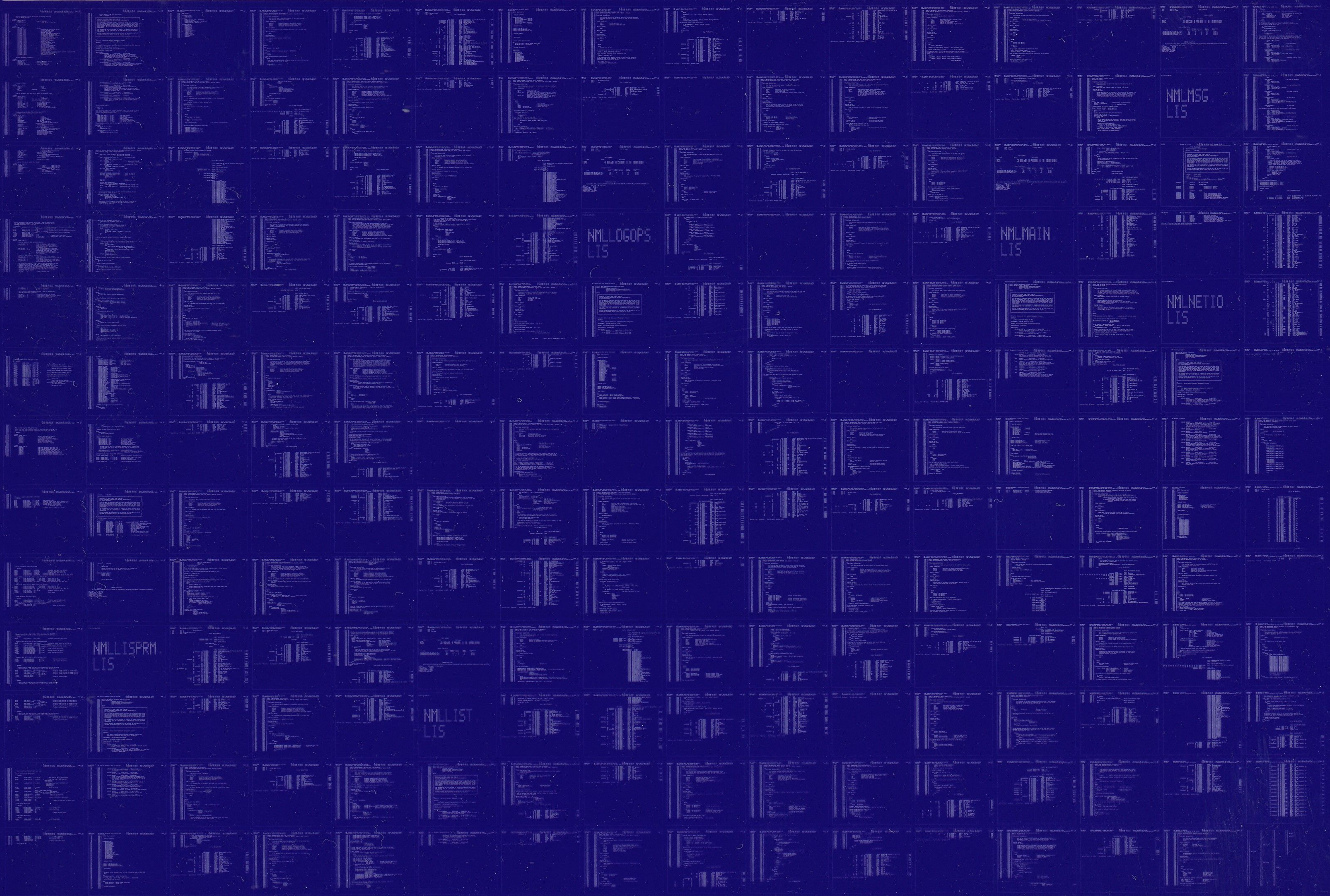
```
1 %SBTTL 'NML$PURPARAM  Delete parameter'
1 GLOBAL ROUTINE NML$PURPARAM (RTNDSC, SEM_LIST)=
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine removes a parameter from the permanent data base record.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       SEM_LIST        Parameter semantic table entry address.
1 !       RTNDSC          Record buffer descriptor address.
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       It is assumed that the permanent data base file is already open.
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       The parameter has been removed from the record.
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       Always returns success (NML$_STS_SUC).
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !
1 !--
1
2    BEGIN
2
2    MAP
2       SEM_LIST : REF BBLOCK;
2
2    NMA$DELETEFLD (.RTNDSC,
2                   .SEM_LIST [PST$W_DATAID]);
2
2    RETURN NML$_STS_SUC
2
1    END;                                          ! End of NML$PURPARAM




                                   0000 00000      .ENTRY  NML$PURPARAM, Save nothing                    ; 1499
              7E        08    BC   3C 00002      MOVZWL  @SEM_LIST, -(SP)                              ; 1536
                        04    AC   DD 00006      PUSHL   RTNDSC                                        ; 1535
     00000000G  00            02   FB 00009      CALLS   #2, NMA$DELETEFLD                             ;
              50              01   D0 00010      MOVL    #1, R0                                        ; 1538
                                   04 00013      RET                                                   ; 1540

; Routine Size:  20 bytes,    Routine Base:  $CODE$ + 06B2
```

```
1 %SBTTL 'NML$PURNODNNA  Delete node name parameter'
1 GLOBAL ROUTINE NML$PURNODNNA (RTNDSC, SEM_LIST)=
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !       This routine removes the node name parameter from the permanent
1 !       data base record if it is not required. It is required in the case
1 !       of a loop node.
1 !
1 ! FORMAL PARAMETERS:
1 !       RTNDSC          Data buffer descriptor address.
1 !       SEM_LIST        Parameter semantic table entry address.
1 !
1 ! IMPLICIT INPUTS:
1 !       It is assumed that the permanent data base file is already open.
1 !
1 ! IMPLICIT OUTPUTS:
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !       Error is returned if the parameter cannot be removed.
1 !
1 ! SIDE EFFECTS:
1 !       NONE
1 !
1 !--
1
2    BEGIN
2
2    MAP
2       SEM_LIST : REF BBLOCK;
2
2    LOCAL
2       FLDADR,
2       FLDSIZE;
2
2    FLDADR = 0;
2    FLDSIZE = 0;
2    IF NMA$SEARCHFLD (.RTNDSC,
2                      NMA$C_PCNO_NLI,
2                      FLDSIZE,
2                      FLDADR)
2    THEN
3       BEGIN
3 !
3 ! Node has circuit (is a loopnode) so name cannot be deleted.
3 !
3       NML$AB_MSGBLOCK [MSB$L_FLAGS] = MSB$M_DET_FLD;
3       NML$AB_MSGBLOCK [MSB$B_CODE] = NMA$C_STS_PNA;
3       NML$AB_MSGBLOCK [MSB$W_DETAIL] = NMA$C_PCNO_NNA;
3
3       RETURN NML$_STS_PNA
3
3       END
2    ELSE
2       NMA$DELETEFLD (.RTNDSC, .SEM_LIST [PST$W_DATAID]);
```

```
; 1617          1598  2
; 1618          1599  2       RETURN NML$_STS_SUC
; 1619          1600  2
; 1620          1601  1       END;                                              . End of NML$PURNODNNA



                                                     0004 00000           .ENTRY  NML$PURNODNNA, Save R2                                 ; 1542
                           52 00000000G        00    9E 00002           MOVAB   NML$AB_MSGBLOCK, R2                                    ;
                           5E                  04    C2 00009           SUBL2   #4, SP                                                 ;
                                               7E    D4 0000C           CLRL    FLDADR                                                 ; 1578
                                        04     AE    D4 0000E           CLRL    FLDSIZE                                                ; 1579
                                               5E    DD 00011           PUSHL   SP                                                     ; 1580
                                        08     AE    9F 00013           PUSHAB  FLDSIZE                                                ;
                           7E         01F5     8F    3C 00016           MOVZWL  #501, -(SP)                                            ;
                                        04     AC    DD 0001B           PUSHL   RTNDSC                                                 ;
                 00000000G 00                  04    FB 0001E           CALLS   #4, NMA$SEARCHFLD                                      ;
                           11                  50    E9 00025           BLBC    R0, 1$                                                 ;
                           62                  02    D0 00028           MOVL    #2, NML$AB_MSGBLOCK                                    ; 1589
                     04    A2                  16    8E 0002B           MNEGB   #22, NML$AB_MSGBLOCK+4                                 ; 1590
                     08    A2         01F4     8F    B0 0002F           MOVW    #500, NML$AB_MSGBLOCK+8                                ; 1591
                           50                  2C    CE 00035           MNEGL   #44, R0                                                ; 1593
                                               04 00038                 RET                                                            ;
                           7E           08     BC    3C 00039 1$:       MOVZWL  @SEM_LIST, -(SP)                                       ; 1597
                                        04     AC    DD 0003D           PUSHL   RTNDSC                                                 ;
                 00000000G 00                  02    FB 00040           CALLS   #2, NMA$DELETEFLD                                      ;
                           50                  01    D0 00047           MOVL    #1, R0                                                 ; 1599
                                               04 0004A                 RET                                                            ; 1601

; Routine Size:  75 bytes,    Routine Base:  $CODE$ + 06C6
; 1622          1602  1 END
; 1623          1603  1
; 1624          1604  0 ELUDOM



;                                       PSECT SUMMARY
;
;       Name                       Bytes                    Attributes
;
;   $OWN$                             334  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;   $PLIT$                             56  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;   $CODE$                           1809  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        42        12         27        00:00.1
;   _$255$DUA28:[SHRLIB]NMALIBRY.L32;1             887        21         2         47        00:00.2
;   _$255$DUA28:[SYSLIB]STARLET.L32;1             9776         4         0        581        00:02.2




;                                       COMMAND QUALIFIERS

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

; Size:          1809 code + 390 data bytes
; Run Time:         00:34.6
; Elapsed Time:     01:30.8
; Lines/CPU Min:     2781
; Lexemes/CPU-Min: 13283
; Memory Used:  131 pages
; Compilation Complete
```

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

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



NMLMSG
LIS

NMLLOGOPS
LIS

NMLMAIN
LIS

NMLNETIO
LIS

NMLLISPRM
LIS

NMLLIST
LIS