J 6

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

_$

Ps

NP

NP

$G

$O

NP

PA

_L

```
                                                                        K 16
**FILE**ID**NMLSHOPRM




NN      NN  MM      MM  LL              SSSSSSSS  HH      HH    000000    PPPPPPPP    RRRRRRRR    MM      MM
NN      NN  MM      MM  LL              SSSSSSSS  HH      HH    000000    PPPPPPPP    RRRRRRRR    MM      MM
NN      NN  MMMM  MMMM  LL          SS          HH      HH  00      00  PP      PP  RR      RR  MMMM  MMMM
NN      NN  MMMM  MMMM  LL          SS          HH      HH  00      00  PP      PP  RR      RR  MMMM  MMMM
NNNN    NN  MM  MM  MM  LL          SS          HH      HH  00      00  PP      PP  RR      RR  MM  MM  MM
NNNN    NN  MM  MM  MM  LL          SS          HH      HH  00      00  PP      PP  RR      RR  MM  MM  MM
NN  NN  NN  MM      MM  LL              SSSSSS    HHHHHHHHHH  00      00  PPPPPPPP    RRRRRRRR    MM      MM
NN  NN  NN  MM      MM  LL              SSSSSS    HHHHHHHHHH  00      00  PPPPPPPP    RRRRRRRR    MM      MM
NN    NNNN  MM      MM  LL                  SS  HH      HH  00      00  PP          RR  RR      MM      MM
NN    NNNN  MM      MM  LL                  SS  HH      HH  00      00  PP          RR  RR      MM      MM
NN      NN  MM      MM  LL                  SS  HH      HH  00      00  PP          RR    RR    MM      MM      ....
NN      NN  MM      MM  LL                  SS  HH      HH  00      00  PP          RR    RR    MM      MM      ....
NN      NN  MM      MM  LLLLLLLLLL  SSSSSSSS    HH      HH    000000    PP          RR      RR  MM      MM      ....
NN      NN  MM      MM  LLLLLLLLLL  SSSSSSSS    HH      HH    000000    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
```

```
0001  0 %TITLE 'NML special volatile parameter handling routines'
0002  0 MODULE NML$SHOPRM (
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 !++
0C35  1 ! FACILITY:  DECnet-VAX V2.0 Network Management Listener
0036  1 !
0037  1 ! ABSTRACT:
0038  1 !
0039  1 !       This module contains routines to process volatile data base
0040  1 !       information from the NETACP QIO buffer.
0041  1 !
0042  1 ! ENVIRONMENT:  VAX/VMS Operating System
0043  1 !
C044  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 !       V03-009 MKP0011         Kathy Perko     9-April-1984
0051  1 !               If returning a node address to a Phase III NCP, clear the
0052  1 !               area number if it's in the exeuctor's area.  If it's not in
0053  1 !               the executor's area, return it as is - that's the best I
0054  1 !               can do.
0055  1 !
0056  1 !       V03-008 MKP0010         Kathy Perko     18-Oct-1983
0057  1 !               Fix previous bug correctly.
```

```
;    58          0058  1 !
;    59          0059  1 !        V03-007 MKP0009          Kathy Perko     27-Sept-1983
;    60          0060  1 !                Fix NMLSSHONODEID so it skips over the node name if no
;    61          0061  1 !                address is returned.
;    62          0062  1 !
;    63          0063  1 !        V03-006 MKP0008          Kathy Perko     17-Aug-1983
;    64          0064  1 !                Fix NMLSSHOEXEPARAM to call NMLSSHONODEID for EXECUTOR node
;    65          0065  1 !                ALIAS parameter.
;    66          0066  1 !
;    67          0067  1 !        V03-005 MKP0007          Kathy Perko     29-July-1983
;    68          0068  1 !                Add EXECUTOR node parameter, ALIAS, and clean up routines
;    69          0069  1 !                that SHOW node ids.
;    70          0070  1 !
;    71          0071  1 !        V03-004 MKP0006          Kathy Perko     29-Nov-1982
;    72          0072  1 !                If NCP is using NICE V3.0.0, clear the area number out of
;    73          0073  1 !                any node numbers returned.
;    74          0074  1 !
;    75          0075  1 !        V03-003 MKP0005          Kathy Perko     24-Nov-1982
;    76          0076  1 !                If NETACP doesn't return a state for a node, don't
;    77          0077  1 !                return one to NCP.
;    78          0078  1 !
;    79          0079  1 !        V03-002 MKP0004          Kathy Perko     25-June-1982
;    80          0080  1 !                Executor and X2n Server Destination subaddresses are now
;    81          0081  1 !                both returned by the ACP as longwords.  Fix up the show
;    82          0082  1 !                routines accordingly.
;    83          0083  1 !
;    84          0084  1 !        V03-001 MKP0003          Kathy Perko     1-April-1982
;    85          0085  1 !                Make changes for X-25 Protocol and Server Modules.
;    86          0086  1 !                Also combine some routines to make NMLSHR smaller.
;    87          0087  1 !
;    88          0088  1 !        V02-002 MKP0002          Kathy Perko     3-Jan-1982
;    89          0089  1 !                Delete routine NMLSSHOLINKS.  It has been moved to the
;    90          0090  1 !                NMLV2COMP module because it's only used for formatting
;    91          0091  1 !                SHOW LINKS commanos for V2 nodes.
;    92          0092  1 !
;    93          0093  1 !        V02-001 MKP0001          Kath) Perko     24-July-1981
;    94          0094  1 !                Delete NML call tc map VMS line to DNA line name.
;    95          0095  1 !--
;    96          0096  1
```

```
:   98      0097  1 %SBTTL 'Declarations'
:   99      0098  1
:  100      0099  1 !
:  101      0100  1 ! TABLE OF CONTENTS:
:  102      0101  1 !
:  103      0102  1
:  104      0103  1 FORWARD ROUTINE
:  105      0104  1     NML$SHOPARAM,
:  106      0105  1     NML$SHONMLVER,
:  107      0106  1     NML$SHOREMSTA,
:  108      0107  1     NML$SHOVERSION,
:  109      0108  1     NML$SHONODEID,
:  110      0109  1     NML$SHOSERVPASS,
:  111      0110  1     NML$SKIPLONG,
:  112      0111  1     NML$SKIPSTRING,
:  113      0112  1     NML$SHOEXEPARAM,
:  114      0113  1     NML$SHORANGE,
:  115      0114  1     NML$SHOCHANNELS,
:  116      0115  1     NML$SHOPWSET,
:  117      0116  1     NML$SHOCOUNTERS,
:  118      0117  1     NML$SHOOWNER;
:  119      0118  1
:  120      0119  1 !
:  121      0120  1 ! INCLUDE FILES:
:  122      0121  1 !
:  123      0122  1
:  124      0123  1 LIBRARY 'LIB$:NMLLIB.L32';
:  125      0124  1 LIBRARY 'SHRLIB$:NMALIBRY.L32';
:  126      0125  1 LIBRARY 'SHRLIB$:NET.L32';
:  127      0126  1 LIBRARY 'SYS$LIBRARY:STARLET.L32';
:  128      0127  1
:  129      0128  1 !
:  130      0129  1 ! OWN STORAGE:
:  131      0130  1 !
:  132      0131  1
:  133      0132  1 !
:  134      0133  1 ! Parameter buffer and descriptor for use in handling volatile data base
:  135      0134  1 ! data.
:  136      0135  1 !
:  137      0136  1
:  138      0137  1 OWN
:  139      0138  1     NML$T_PRMBUFFER : VECTOR [256, BYTE];
:  140      0139  1 BIND
:  141      0140  1     NML$Q_PRMDSC = UPLIT (256, NML$T_PRMBUFFER) : DESCRIPTOR;
:  142      0141  1
:  143      0142  1 !
:  144      0143  1 ! EXTERNAL REFERENCES:
:  145      0144  1 !
:  146      0145  1
:  147      0146  1 $NML_EXTDEF;
:  148      0147  1
:  149      0148  1 EXTERNAL
:  150      0149  1     nml$gb_ncp_version,
:  151      0150  1     nml$gw_vol_exec_addr: BBLOCK [2];
:  152      0151  1
:  153      0152  1 EXTERNAL ROUTINE
:  154      0153  1     NML$ADDMSGCOU,
;  155          0154  1      NML$ADDMSGPRM,
;  156          0155  1      NML$LISNMLVER,
;  157          0156  1      NML$GETNODNAM,
;  158          0157  1      NML$NETQIO,
;  159          0158  1      NML$ERROR_1;
;   161       0159  1 GLOBAL ROUTINE NML$SHOPARAM (SEM_LIST, BUFDSC, MSGSIZE, DATDSC, DATPTR)=
;   162       0160  1
;   163       0161  1 !++
;   164       0162  1 ! FUNCTIONAL DESCRIPTION:
;   165       0163  1 !
;   166       0164  1 !       This routine is used to format byte, word, longword, and string NICE
;   167       0165  1 !       parameters for SHOW commands.  It gets a longword or string parameter
;   168       0166  1 !       from the QIO buffer, and adds it to the NICE response message.
;   169       0167  1 !
;   170       0168  1 ! FORMAL PARAMETERS:
;   171       0169  1 !
;   172       0170  1 !       SEM_LIST        Parameter semantic table entry address.
;   173       0171  1 !       BUFDSC          Output message buffer descriptor address.
;   174       0172  1 !       MSGSIZE         Address of current output message size.
;   175       0173  1 !       DATDSC          QIO buffer descriptor address.
;   176       0174  1 !       DATPTR          Current pointer into QIO data buffer.
;   177       0175  1 !
;   178       0176  1 ! ROUTINE VALUE:
;   179       0177  1 ! COMPLETION CODES:
;   180       0178  1 !
;   181       0179  1 !       Always returns success (NML$_STS_SUC).
;   182       0180  1 !
;   183       0181  1 !--
;   184       0182  1
;   185       0183  2 BEGIN
;   186       0184  2
;   187       0185  2 MAP
;   188       0186  2     SEM_LIST : REF BBLOCK;
;   189       0187  2
;   190       0188  2 LOCAL
;   191       0189  2     DATA_TYPE: BBLOCK [1], ! NICE parameter data type
;   192       0190  2     NICE_LEN,              ! Length of parameter in NICE response message.
;   193       0191  2     CHECK_STRING;
;   194       0192  2
;   195       0193  2 !
;   196       0194  2 ! Using the NICE data type field in the Parameter Semantic Table (PST),
;   197       0195  2 ! determine how long the parameter will be in the NICE response message.
;   198       0196  2 !
;   199       0197  2 CHECK_STRING = 0;
;   200       0198  2 DATA_TYPE = .SEM_LIST [PST$B_DATATYPE];
;   201       0199  2 !
;   202       0200  2 ! Check to see if the parameter is coded.
;   203       0201  2 !
;   204       0202  2 IF .DATA_TYPE [NMA$V_PTY_COD] THEN
;   205       0203  3     BEGIN
;   206       0204  3     IF .DATA_TYPE [NMA$V_PTY_CMU] THEN
;   207       0205  3         NML$ERROR_1 (NMA$C_STS_MPR)     ! Signal NML error.
;   208       0206  3     ELSE
;   209       0207  3         !
;   210       0208  3         ! The parameter is a coded single field.  Get the parameter's length
;   211       0209  3         ! from the low order 6 bits.
;   212       0210  3         !
;   213       0211  3         NICE_LEN = .DATA_TYPE [NMA$V_PTY_CLE];
;   214       0212  3     END
;   215       0213  2 ELSE
;   216       0214  2 !
;   217       0215  2 ! The parameter is not coded.
```

```
2 !
3     BEGIN
3     IF .DATA_TYPE [NMA$V_PTY_ASC] OR     ! NICE parameter type = string
3        .DATA_TYPE [NMA$V_PTY_NLE] EQL 0 ! NICE parameter type = binary image
3     THEN
4        BEGIN
4        NICE_LEN = .(..DATPTR) <0,16>;
4        .DATPTR = ..DATPTR + 2;
4        CHECK_STRING = 1;
4        END
3     ELSE
3        NICE_LEN = .DATA_TYPE [NMA$V_PTY_NLE];
2     END;
2 !
2 ! If the ACP has a value for the parameter, add it to the NICE response
2 ! message.  The ACP does not have a value for the parameter if:
2 !        - It's a string, and the length is zero.
2 !        - It's a longword, and the value is -1.
2 ! The ACP returns only longwords or strings.
2 !
2 IF (.CHECK_STRING AND .NICE_LEN NEQ 0) OR
2    ((NOT .CHECK_STRING) AND (...DATPTR NEQ -1)) THEN
2     NML$ADDMSGPRM (      .BUFDSC,
2                          .MSGSIZE,
2                          .SEM_LIST [PST$W_DATAID],
2                          .SEM_LIST [PST$B_DATATYPE],
2                          .NICE_LEN,
2                          ..DATPTR);
2
2 !
2 ! Increment the pointer to the QIO P4 buffer to the next parameter
2 ! returned by the ACP.
2 !
2 IF .CHECK_STRING THEN
2     .DATPTR = ..DATPTR + .NICE_LEN
2 ELSE
2     .DATPTR = ..DATPTR + 4;
2
2 RETURN NML$_STS_SUC
2
1 END;                              ! End of NML$SHOPARAM
```

```
                                             .TITLE  NML$SHOPRM NML special volatile parameter handl
                                                                ing routine
                                             .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
```

```
                                                        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   NML$AB_RCVBUFFER
                                                                .EXTRN   NML$GQ_RCVBFDSC
                                                                .EXTRN   NML$AB_SNDBUFFER
                                                                .EXTRN   NML$GQ_SNDBFDSC
                                                                .EXTRN   NML$GL_RCVDATLEN
                                                                .EXTRN   NML$AB_CPTABLE, NML$AB_MSGBLOCK
                                                                .EXTRN   NML$AB_ENTITY_ID
                                                                .EXTRN   NML$AB_QUALIFIER_ID
                                                                .EXTRN   NML$AB_ENTITYDATA
                                                                .EXTRN   NML$AB_NML_NMV, NML$AB_PRMSEM
                                                                .EXTRN   NML$AB_RECBUF, NML$AL_ENTINFTAB
                                                                .EXTRN   NML$AL_PERMINFTAB
                                                                .EXTRN   NML$AW_PRM_DES, NML$GB_CMD_VER
                                                                .EXTRN   NML$GB_ENTITY_CODE
                                                                .EXTRN   NML$GB_ENTITY_FORMAT
                                                                .EXTRN   NML$GL_QUALIFIER_PST
                                                                .EXTRN   NML$GB_QUALIFIER_FORMAT
                                                                .EXTRN   NML$GB_FUNCTION
                                                                .EXTRN   NML$GB_INFO, NML$GB_OPTIONS
                                                                .EXTRN   NML$GL_PRMCODE, NML$GL_PRS_FLGS
                                                                .EXTRN   NML$GL_NML_ENTITY
                                                                .EXTRN   NML$GQ_NETNAMDSC
                                                                .EXTRN   NML$GQ_RECBFDSC
                                                                .EXTRN   NML$GW_PRMDESCNT
                                                                .EXTRN   NML$GB_NCP_VERSION
                                                                .EXTRN   NML$GW_VOL_EXEC_ADDR
                                                                .EXTRN   NML$ADDMSGCOU, NML$ADDMSGPRM
                                                                .EXTRN   NML$LISNMLVER, NML$GETNODNAM
                                                                .EXTRN   NML$NETQIO, NML$ERROR_1

                                                                .PSECT   $CODE$,NOWRT,2

                                                  003C 00000    .ENTRY   NML$SHOPARAM, Save R2,R3,R4,R5                         ; 0159
                                                55  D4 00002    CLRL     CHECK_STRING                                           ; 0197
                              53        04      AC  D0 00004    MOVL     SEM_LIST, R3                                           ; 0198
                              52        03      A3  90 00008    MOVB     3(R3), DATA_TYPE                                       ;
                                                17  18 0000C    BGEQ     2$                                                     ; 0202
                  0C          52                06  E1 0000E    BBC      #6, DATA_TYPE, 1$                                      ; 0204
                              7E                05  CE 00012    MNEGL    #5, -(SP)                                              ; 0205
                    00000000G 00                01  FB 00015    CALLS    #1, NML$ERROR_1                                        ;
                                                25  11 0001C    BRB      5$                                                     ;
       54             52      06                00  EF 0001E 1$: EXTZV   #0, #6, DATA_TYPE, NICE_LEN                            ; 0211
```

```
                                                            1E 11 00023          BRB      5$                                        ; 0202
                               05             52            06 E0 00025 2$:      BBS      #6, DATA_TYPE, 3$                         ; 0218
                                              0F            52 93 00029          BITB     DATA_TYPE, #15                            ; 0219
                                                            10 12 0002C          BNEQ     4$                                        ;
                                              50       14   AC D0 0002E 3$:      MOVL     DATPTR, R0                                ; 0222
                                              54       00   B0 3C 00032          MOVZWL   @0(R0), NICE_LEN                          ;
                                              60            02 C0 00036          ADDL2    #2, (R0)                                  ; 0223
                                              55            01 D0 00039          MOVL     #1, CHECK_STRING                          ; 0224
                                                            05 11 0003C          BRB      5$                                        ; 0218
                54             52             04            00 EF 0003E 4$:      EXTZV    #0, #4, DATA_TYPE, NICE_LEN               ; 0227
                                              07            55 E9 00043 5$:      BLBC     CHECK_STRING, 6$                          ; 0236
                                                            54 D5 00046          TSTL     NICE_LEN                                  ;
                                                            10 12 00048          BNEQ     7$                                        ;
                                              27            55 E8 0004A          BLBS     CHECK_STRING, 9$                          ; 0237
                                              50       14   BC D0 0004D 6$:      MOVL     @DATPTR, R0                               ;
                                  FFFFFFFF    8F            60 D1 00051          CMPL     (R0), #-1                                 ;
                                                            17 13 00058          BEQL     8$                                        ;
                                                       14   BC DD 0005A 7$:      PUSHL    @DATPTR                                   ; 0243
                                                            54 DD 0005D          PUSHL    NICE_LEN                                  ; 0242
                                              7E       03   A3 9A 0005F          MOVZBL   3(R3), -(SP)                              ; 0241
                                              7E            63 3C 00063          MOVZWL   (R3), -(SP)                               ; 0240
                                              7E       08   AC 7D 00066          MOVQ     BUFDSC, -(SP)                             ; 0238
                                 00000000G    00            06 FB 0006A          CALLS    #6, NML$ADDMSGPRM                         ;
                                              06            55 E9 00071 8$:      BLBC     CHECK_STRING, 10$                         ; 0250
                                        14    BC            54 C0 00074 9$:      ADDL2    NICE_LEN, @DATPTR                         ;
                                                            04 11 00078          BRB      11$                                       ;
                                        14    BC            04 C0 0007A 10$:     ADDL2    #4, @DATPTR                               ; 0252
                                              50            01 D0 0007E 11$:     MOVL     #1, R0                                    ; 0254
                                                               04 00081          RET                                                ; 0256

; Routine Size:  130 bytes,    Routine Base:  $CODE$ + 0000
;   260        0257  1 %SBTTL 'NML$SHONMLVER  Get NML version number'
;   261        0258  1 GLOBAL ROUTINE NML$SHONMLVER (SEM_TABLE, BUFDSC, MSGSIZE, DUMDSC, DATPTR) =
;   262        0259  1
;   263        0260  1 !++
;   264        0261  1 ! FUNCTIONAL DESCRIPTION:
;   265        0262  1 !
;   266        0263  1 !       This routine moves the network management version number into
;   267        0264  1 !       the output message as a coded multiple parameter.
;   268        0265  1 !
;   269        0266  1 ! FORMAL PARAMETERS:
;   270        0267  1 !
;   271        0268  1 !       SEM_TABLE       Parameter semantic table entry address.
;   272        0269  1 !       BUFDSC          Output message buffer descriptor.
;   273        0270  1 !       MSGSIZE         Address of current output message size.
;   274        0271  1 !       DUMDSC          Not used.
;   275        0272  1 !       DATPTR          Current pointer into QIO data buffer.
;   276        0273  1 !
;   277        0274  1 ! IMPLICIT INPUTS:
;   278        0275  1 !
;   279        0276  1 !       NONE
;   280        0277  1 !
;   281        0278  1 ! IMPLICIT OUTPUTS:
,   282        0279  1 !
;   283        0280  1 !       Parameter is added to output message buffer.
;   284        0281  1 !
;   285        0282  1 ! ROUTINE VALUE:
;   286        0283  1 ! COMPLETION CODES:
;   287        0284  1 !
;   288        0285  1 !       Always returns success (NML$_STS_SUC).
;   289        0286  1 !
;   290        0287  1 ! SIDE EFFECTS:
;   291        0288  1 !
;   292        0289  1 !       NONE
;   293        0290  1 !
;   294        0291  1 !--
;   295        0292  1
;   296        0293  2     BEGIN
;   297        0294  2
;   298        0295  2     NML$LISNMLVER (.SEM_TABLE, .BUFDSC, .MSGSIZE, .DUMDSC);
;   299        0296  2
;   300        0297  2     RETURN NML$_STS_SUC
;   301        0298  2
;   302        0299  1     END;                            ! End of NML$SHONMLVER



                                   0000 00000          .ENTRY  NML$SHONMLVER, Save nothing                       ; 0258
                 7E       0C    AC  7D 00002           MOVQ    MSGSIZE, -(SP)                                    ; 0295
                 7E       04    AC  7D 00006           MOVQ    SEM_TABLE, -(SP)                                  ;
       00000000G 00             04  FB 0000A           CALLS   #4, NML$LISNMLVER                                 ;
                 50             01  D0 00011           MOVL    #1, R0                                            ; 0297
                                04  00014              RET                                                       ; 0299

; Routine Size:  21 bytes,    Routine Base:  $CODE$ + 0082
```

```
;   304         0300  1 %SBTTL 'NML$SHOVERSION  Get coded multiple version number'
;   305         0301  1 GLOBAL ROUTINE NML$SHOVERSION (SEM_LIST, BUFDSC, MSGSIZE, DUMDSC, DATPTR)=
;   306         0302  1
;   307         0303  1 !++
;   308         0304  1 ! FUNCTIONAL DESCRIPTION:
;   309         0305  1 !
;   310         0306  1 !       This parameter moves network facility version numbers into the
;   311         0307  1 !       output message buffer as a coded multiple field.  Version numbers
;   312         0308  1 !       are a string of three bytes.
;   313         0309  1 !
;   314         0310  1 ! FORMAL PARAMETERS:
;   315         0311  1 !
;   316         0312  1 !       SEM_LIST        Parameter semantic table entry address.
;   317         0313  1 !       BUFDSC          Output message buffer descriptor address.
;   318         0314  1 !       MSGSIZE         Address of current output message size.
;   319         0315  1 !       DUMDSC          Not used.
;   320         0316  1 !
;   321         0317  1 ! IMPLICIT INPUTS:
;   322         0318  1 !
;   323         0319  1 !       NONE
;   324         0320  1 !
;   325         0321  1 ! IMPLICIT OUTPUTS:
;   326         0322  1 !
;   327         0323  1 !       The output message buffer contains the coded multiple version number.
;   328         0324  1 !
;   329         0325  1 ! ROUTINE VALUE:
;   330         0326  1 ! COMPLETION CODES:
;   331         0327  1 !
;   332         0328  1 !       Always returns success (NML$_STS_SUC).
;   333         0329  1 !
;   334         0330  1 ! SIDE EFFECTS:
;   335         0331  1 !
;   336         0332  1 !       NONE
;   337         0333  1 !
;   338         0334  1 !--
;   339         0335  1
;   340         0336  2     BEGIN
;   341         0337  2
;   342         0338  2     MAP
;   343         0339  2         SEM_LIST : REF BLOCK [, BYTE];
;   344         0340  2
;   345         0341  2     LOCAL
;   346         0342  2         BUFFER : VECTOR [6, BYTE],
;   347         0343  2         LEN,
;   348         0344  2         PTR;
;   349         0345  2 !
;   350         0346  2 ! Read version parameter.
;   351         0347  2 !
;   352         0348  2     LEN = .(.NML$GL_EXEDATPTR)<0,16>;
;   353         0349  2
;   354         0350  2     IF .LEN NEQU 3                          ! Length must be 3 bytes
;   355         0351  2     THEN
;   356         0352  2         RETURN NML$_STS_MPR;
;   357         0353  2
;   358         0354  2     NML$GL_EXEDATPTR = .NML$GL_EXEDATPTR + 2;
;   359         0355  2 !
;   360         0356  2 ! Add version parameter to message.
;  361            0357  2 !
;  362            0358  2              PTR = CH$PTR (BUFFER);        ! Point to output buffer
;  363            0359  2
;  364            0360  2              INCR I FROM 0 TO 2 DO
;  365            0361  3                  BEGIN
;  366            0362  3                  CH$WCHAR_A (1, PTR);
;  367            0363  3                  CH$WCHAR_A (CH$RCHAR_A (NML$GL_EXEDATPTR) - '0' , PTR);
;  368            0364  2                  END;
;  369            0365  2
;  370            0366  2              NML$ADDMSGPRM (       .BUFDSC,
;  371            0367  2                                    .MSGSIZE,
;  372            0368  2                                    .SEM_LIST [PST$W_DATAID],
;  373            0369  2                                    .SEM_LIST [PST$B_DATATYPE] OR 3,
;  374            0370  2                                    6,
;  375            0371  2                                    BUFFER);
;  376            0372  2
;  377            0373  2         RETURN NML$_STS_SUC
;  378            0374  2
;  379            0375  1         END;                                   ! End of NML$SHOVERSION





                                            001C 00000           .ENTRY   NML$SHOVERSION, Save R2,R3,R4                  ; 0301
                      54 00000000G    00   9E 00002           MOVAB    NML$GL_EXEDATPTR, R4                            ;
                      5E              08   C2 00009           SUBL2    #8, SP                                          ;
                      50              64   D0 0000C           MOVL     NML$GL_EXEDATPTR, R0                            ; 0348
                      50              60   3C 0000F           MOVZWL   (R0), LEN                                       ;
                      03              50   D1 00012           CMPL     LEN, #3                                         ; 0350
                                      04   13 00015           BEQL     1$                                              ;
                      50              0A   CE 00017           MNEGL    #10, R0                                         ; 0352
                                           04 0001A           RET                                                      ;
                      64              02   C0 0001B 1$:       ADDL2    #2, NML$GL_EXEDATPTR                            ; 0354
                      50              6E   9E 0001E           MOVAB    BUFFER, PTR                                     ; 0358
                                      52   D4 00021           CLRL     I                                               ; 0360
                      80              01   90 00023 2$:       MOVB     #1, (PTR)+                                      ; 0362
                      53              64   D0 00026           MOVL     NML$GL_EXEDATPTR, R3                            ; 0363
                      51              63   9A 00029           MOVZBL   (R3), R1                                        ;
                                      64   D6 0002C           INCL     NML$GL_EXEDATPTR                                ;
             80       51              30   83 0002E           SUBB3    #48, R1, (PTR)+                                 ;
             ED       52              02   F3 00032           AOBLEQ   #2, I, 2$                                       ; 0360
                                      5E   DD 00036           PUSHL    SP                                              ; 0366
                                      06   DD 00038           PUSHL    #6                                              ;
                      50        04    AC   D0 0003A           MOVL     SEM_LIST, R0                                    ; 0369
                      51        03    A0   9A 0003E           MOVZBL   3(R0), R1                                       ;
             7E       51              03   C9 00042           BISL3    #3, R1, -(SP)                                   ;
                      7E              60   3C 00046           MOVZWL   (R0), -(SP)                                     ; 0368
                      7E        08    AC   7D 00049           MOVQ     BUFDSC, -(SP)                                   ; 0366
               00000000G  00          06   FB 0004D           CALLS    #6, NML$ADDMSGPRM                               ;
                      50              01   D0 00054           MOVL     #1, R0                                          ; 0373
                                           04 00057           RET                                                      ; 0375

; Routine Size:  88 bytes,    Routine Base:  $CODE$ + 0097
```

```
1 %SBTTL 'NML$SHOREMSTA  Get remote node state'
1 GLOBAL ROUTINE NML$SHOREMSTA (SEM_LIST, BUFDSC, MSGSIZE, DATDSC, DATPTR)=
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine maps remote node status from the internal NETACP
1 !       bit value to the network management state value.  The following
1 !       states are possible:
1 !
1 !           reachable   (NETACP value = 1, NML value = NMA$C_STATE_REA)
1 !           unreachable (NETACP value = 0, NML value = NMA$C_STATE_UNR)
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 !       DATPTR          Current pointer into QIO data buffer.
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       If NETACP did not know the state of the remote node, returns
1 !       NML$_STS_PTY.
1 !
1 !
1 !--
1
2 BEGIN
2
2 MAP
2     SEM_LIST : REF BLOCK [, BYTE];
2
2 LOCAL
2     STATE : BYTE;
2
2 IF .(..DATPTR) EQLU -1 THEN
3     BEGIN
3     .DATPTR = ..DATPTR + 4;
3     RETURN NML$_STS_PTY;
2     END;
2 !
2 ! Map bit setting to correct network management value.
2 !
3 STATE = ( IF .(..DATPTR)<0,8> THEN
3               NMA$C_STATE_REA ! Reachable
3           ELSE
2               NMA$C_STATE_UNR);       ! Unreachable
2 !
2 ! Add state parameter to message.
2 !
2 NML$ADDMSGPRM (.BUFDSC,
2               .MSGSIZE,
2               .SEM_LIST [PST$W_DATAID],
2               .SEM_LIST [PST$B_DATATYPE],
;  438          0433  2                   1,
;  439          0434  2                   STATE);
;  440          0435  2
;  441          0436  2  .DATPTR = ..DATPTR + 4;
;  442          0437  2
;  443          0438  2  RETURN NML$_STS_SUC
;  444          0439  2
;  445          0440  1  END;                                ! End of NML$SHOREMSTA



                                           0000 00000          .ENTRY  NML$SHOREMSTA, Save nothing                          ; 0377
                       5E             04    C2 00002          SUBL2   #4, SP                                               ;
                       50      14     BC    D0 00005          MOVL    @DATPTR, R0                                          ; 0414
            FFFFFFFF   8F             60    D1 00009          CMPL    (R0), #-1                                            ;
                                      08    12 00010          BNEQ    1$                                                   ;
                  14   BC             04    C0 00012          ADDL2   #4, @DATPTR                                          ; 0416
                       50             0C    CE 00016          MNEGL   #12, R0                                              ; 0417
                                            04 00019          RET                                                          ;
                       50      14     BC    D0 0001A 1$:      MOVL    @DATPTR, R0                                          ; 0422
                       05             60    E9 0001E          BLBC    (R0), 2$                                             ;
                       50             04    D0 00021          MOVL    #4, R0                                               ;
                                      03    11 00024          BRB     3$                                                   ;
                       50             05    D0 00026 2$:      MOVL    #5, R0                                               ;
                       6E             50    90 00029 3$:      MOVB    R0, STAT                                             ;
                                      5E    DD 0002C          PUSHL   SP                                                   ; 0429
                                      01    DD 0002E          PUSHL   #1                                                   ;
                       50      04     AC    D0 00030          MOVL    SEM_LIST, R0                                         ; 0432
                       7E      03     A0    9A 00034          MOVZBL  3(R0), -(SP)                                         ;
                       7E             60    3C 00038          MOVZWL  (R0), -(SP)                                          ; 0431
                       7E      08     AC    7D 0003B          MOVQ    BUFDSC, -(SP)                                        ; 0429
            00000000G  00             06    FB 0003F          CALLS   #6, NML$ADDMSGPRM                                    ;
                  14   BC             04    C0 00046          ADDL2   #4, @DATPTR                                          ; 0436
                       50             01    D0 0004A          MOVL    #1, R0                                               ; 0438
                                            04 0004D          RET                                                          ; 0440

; Routine Size:  78 bytes,    Routine Base:  $CODE$ + 00EF
```

```
1 %SBTTL 'NML$SHONODEID  Get adjacent node id'
1 GLOBAL ROUTINE NML$SHONODEID (SEM_LIST, BUFDSC, MSGSIZE, DATDSC, DATPTR)=
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine adds the id of the remote node (NMA$C_PCLI_ADJ) to
1 !       which a line is connected to the output message buffer as a coded
1 !       multiple field.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       SEM_LIST        Parameter semantic table entry address.
1 !       BUFDSC          Output message buffer descriptor address.
1 !       MSGSIZE         Address of current output message size.
1 !       DATDSC          QIO buffer descriptor address.
1 !       DATPTR          Current pointer into QIO data buffer.
1 !
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       Always returns success (NML$_STS_SUC).
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !
1 !--
1
2     BEGIN
2
2     MAP
2         sem_list : REF BBLOCK [, BYTE];
2
2     LOCAL
2         cm_count,                         ! Coded multiple field count
2         name_dsc: VECTOR [2],             ! Descriptor of node name.
2         name_buf: BBLOCK [6],             ! Temporary buffer for node name.
2         totlen,                           ! Total length of field
2         nodadr,
2         ptr;
2
2 !
2 ! Get node address from P4 buffer returned by NETACP and increment pointer
2 ! to the next parameter in the buffer.
2 !
2 nodadr = .(..datptr)<0,32>;
2 .datptr = ..datptr + 4;
2 !
2 ! If address is zero then don't return this parameter.  If there is one,
2 ! skip over the node name parameter before returning.
2 !
2 IF .nodadr EQLU -1 THEN
3     BEGIN
3     IF .sem_list [pst$l_nfbid] EQL nfb$c_aji_add OR
3         .sem_list [pst$l_nfbid] EQL nfb$c_ndi_nnd OR
```

```
;  504       0498  3          .sem_list [pst$l_nfbid] EQL nfb$c_lli_pnn THEN
;  505       0499  3          .datptr = ..datptr + .(..datptr)<0,16> + 2;
;  506       0500  3      RETURN nml$_sts_pty;
;  507       0501  2      END;
;  508       0502  2  !
;  509       0503  2  ! If the NCP I'm talking to is speaking NICE V3.0.0 or less, and the node
;  510       0504  2  ! is in the executor's area, clear the area number from the node number.
;  511       0505  2  ! The theory is that the Phase III system should see node's in the executor's
;  512       0506  2  ! area normally (for a Phase III system), but node's outside the exeuctor's
;  513       0507  2  ! area shouldn't be represented as nodes in the executor's area.  So those
;  514       0508  2  ! will just have funny addresses because the area number will not be properly
;  515       0509  2  ! formatted by the Phase III system.
;  516       0510  2  !
;  517       0511  2  IF CH$RCHAR (nml$gb_ncp_version) LEQ 3 THEN
;  518       0512  3      BEGIN
;  519       0513  3      MAP
;  520       0514  3          nodadr: BBLOCK;
;  521       0515  3
;  522       0516  3      IF .nml$gw_vol_exec_addr [nma$v_area] EQL .nodadr [nma$v_area] THEN
;  523       0517  3          nodadr [nma$v_area] = 0;
;  524       0518  2      END;
;  525       0519  2
;  526       0520  2  ptr = nml$t_prmbuffer;
;  527       0521  2  !
;  528       0522  2  ! Add node address field.
;  529       0523  2  !
;  530       0524  2  CH$WCHAR_A (2, ptr);
;  531       0525  2  ptr = CH$MOVE (2, nodadr, .ptr);
;  532       0526  2  !
;  533       0527  2  ! Get the maximum number of fields in the coded multiple (some parameters
;  534       0528  2  ! are returned as a node number and name, and some are returned as simply
;  535       0529  2  ! a node number.
;  536       0530  2  !
;  537       0531  2  cm_count = .sem_list [pst$b_datatype] AND NOT nma$m_pty_cmu;
;  538       0532  2  !
;  539       0533  2  ! If a node name is ever part of this parameter, add the node name field
;  540       0534  2  ! (provided NETACP returned one) to the NICE message.
;  541       0535  2  !
;  542       0536  2  IF .cm_count EQL 2 THEN
;  543       0537  3      BEGIN
;  544       0538  3      SELECTONEU .sem_list [pst$l_nfbid] OF
;  545       0539  3          SET
;  546       0540  3          [nfb$c_aji_add,          ! Circuit adjacent node address
;  547       0541  3           nfb$c_ndi_nnd,          ! Node next node to destination
;  548       0542  3           nfb$c_lli_pnn]:         ! Logical link partner node
;  549       0543  4              BEGIN
;  550       0544  4              name_dsc [0] = .(..datptr)<0,16>;
;  551       0545  4              .datptr = ..datptr + 2;
;  552       0546  4              name_dsc [1] = ..datptr;
;  553       0547  4              .datptr = ..datptr + .name_dsc [0];
;  554       0548  3              END;
;  555       0549  3
;  556       0550  3          [OTHERWISE]:
;  557       0551  4              BEGIN
;  558       0552  4              name_dsc [0] = 6;
;  559       0553  4              name_dsc [1] = name_buf;
;  560       0554  4              nml$getnodnam (.nodadr, name_dsc, name_dsc [0]);
;  561          0555  3                 END;
;  562          0556  3             TES;
;  563          0557  3         !
;  564          0558  3         ! If a node name was returned by NETACP, add it to the message
;  565          0559  3         ! parameter.
;  566          0560  3         !
;  567          0561  3         IF .name_dsc [0] NEQU 0 THEN
;  568          0562  4             BEGIN
;  569          0563  4
;  570          0564  4             CH$WCHAR_A (nma$m_pty_asc, ptr);
;  571          0565  4             CH$WCHAR_A (.name_dsc [0], ptr);
;  572          0566  4             ptr = CH$MOVE (.name_dsc [0], .name_dsc [1], .ptr);
;  573          0567  4
;  574          0568  4             END
;  575          0569  3         ELSE
;  576          0570  3             cm_count = 1;
;  577          0571  2         END;
;  578          0572  2
;  579          0573  2 totlen = .ptr - nml$t_prmbuffer;
;  580          0574  2 !
;  581          0575  2 ! Add node id to output message as a coded multiple field.
;  582          0576  2 !
;  583          0577  2 nml$addmsgprm (.bufdsc,
;  584          0578  2                        .msgsize,
;  585          0579  2                        .sem_list [pst$w_dataid],
;  586          0580  2                        nma$m_pty_cmu OR .cm_count,
;  587          0581  2                        .totlen,
;  588          0582  2                        nml$t_prmbuffer);
;  589          0583  2
;  590          0584  2 RETURN nml$_sts_suc
;  591          0585  2
;  592          0586  1 END;                                    ! End of nml$shonodeid



                                          01FC 00000          .ENTRY   NML$SHONODEID, Save R2,R3,R4,R5,R6,R7,R8      ; 0442
                      58 00000000'   00   9E 00002          MOVAB    NML$T_PRMBUFFER, R8                           ;
                      5E             10   C2 00009          SUBL2    #16, SP                                       ;
                      51        14   AC   D0 0000C          MOVL     DATPTR, R1                                    ; 0488
                      52        00   B1   D0 00010          MOVL     @0(R1), NODADR                                ;
                      61             04   C0 00014          ADDL2    #4, (R1)                                      ; 0489
            FFFFFFFF  8F             52   D1 00017          CMPL     NODADR, #-1                                   ; 0494
                                     31   12 0001E          BNEQ     3$                                            ;
                      50        04   AC   D0 00020          MOVL     SEM_LIST, R0                                  ; 0496
            13010010  8F        0C   A0   D1 00024          CMPL     12(R0), #318832656                            ;
                                     14   13 0002C          BEQL     1$                                            ;
            02010022  8F        0C   A0   D1 0002E          CMPL     12(R0), #33620002                             ; 0497
                                     0A   13 00036          BEQL     1$                                            ;
            08020043  8F        0C   A0   D1 00038          CMPL     12(R0), #134348867                            ; 0498
                                     0B   12 00040          BNEQ     2$                                            ;
                      50        00   B1   3C 00042 1$:      MOVZWL   @0(R1), R0                                    ; 0499
                      50             61   C0 00046          ADDL2    (R1), R0                                      ;
                      61        02   A0   9E 00049          MOVAB    2(R0), (R1)                                   ;
                      50             0C   CE 0004D 2$:      MNEGL    #12, R0                                       ; 0500
                                          04 00050          RET                                                    ;
```

```
                                        03 00000000G  00  91 00051 3$:     CMPB    NML$GB_NCP_VERSION, #3                       ; 0511
                                                      15  1A 00058         BGTRU   4$                                           ;
           50             52            06            0A  EF 0005A         EXTZV   #10, #6, NODADR, R0                          ; 0516
           50 00000000G   00            06            02  ED 0005F         CMPZV   #2, #6, NML$GW_VOL_EXEC_ADDR+1, R0           ;
                                                      05  12 00068         BNEQ    4$                                           ;
                                        52     FC00   8F  AA 0006A         BICW2   #64512, NODADR                               ; 0517
                                        53            68  9E 0006F 4$:     MOVAB   NML$T_PRMBUFFER, PTR                         ; 0520
                                        83            02  90 00072         MOVB    #2, (PTR)+                                   ; 0524
                                        83            52  B0 00075         MOVW    NODADR, (PTR)+                               ; 0525
                                        56       04   AC  D0 00078         MOVL    SEM_LIST, R6                                 ; 0531
           57      03     A6            06            00  EF 0007C         EXTZV   #0, #6, 3(R6), CM_COUNT                      ;
                                        02            57  D1 00082         CMPL    CM_COUNT, #2                                 ; 0536
                                                      5F  12 00085         BNEQ    9$                                           ;
                                        50       0C   A6  D0 00087         MOVL    12(R6), R0                                   ; 0538
                             02010022   8F            50  D1 0008B         CMPL    R0, #33620002                                ; 0540
                                                      12  13 00092         BEQL    5$                                           ;
                             08020043   8F            50  D1 00094         CMPL    R0, #134348867                               ;
                                                      09  13 0009B         BEQL    5$                                           ;
                             13010010   8F            50  D1 0009D         CMPL    R0, #318832656                               ;
                                                      12  12 000A4         BNEQ    6$                                           ;
                                   08   AE       00   B1  3C 000A6 5$:     MOVZWL  @0(R1), NAME_DSC                             ; 0544
                                        61            02  C0 000AB         ADDL2   #2, (R1)                                     ; 0545
                                   0C   AE            61  D0 000AE         MOVL    (R1), NAME_DSC+4                             ; 0546
                                        61       08   AE  C0 000B2         ADDL2   NAME_DSC, (R1)                               ; 0547
                                                      17  11 000B6         BRB     7$                                           ; 0538
                                   08   AE            06  D0 000B8 6$:     MOVL    #6, NAME_DSC                                 ; 0552
                                   0C   AE            6E  9E 000BC         MOVAB   NAME_BUF, NAME_DSC+4                         ; 0553
                                                 08   AE  9F 000C0         PUSHAB  NAME_DSC                                     ; 0554
                                                 0C   AE  9F 000C3         PUSHAB  NAME_DSC                                     ;
                                                      52  DD 000C6         PUSHL   NODADR                                       ;
                            00000000G   00            03  FB 000C8         CALLS   #3, NML$GETNODNAM                            ;
                                        50       08   AE  D0 000CF 7$:     MOVL    NAME_DSC, R0                                 ; 0561
                                                      0E  13 000D3         BEQL    8$                                           ;
                                        83       40   8F  90 000D5         MOVB    #64, (PTR)+                                  ; 0564
                                        83            50  90 000D9         MOVB    R0, (PTR)+                                   ; 0565
                          63       0C   BE            50  28 000DC         MOVC3   R0, @NAME_DSC+4, (PTR)                       ; 0566
                                                      03  11 000E1         BRB     9$                                           ; 0561
                                        57            01  D0 000E3 8$:     MOVL    #1, CM_COUNT                                 ; 0570
                                        50            68  9E 000E6 9$:     MOVAB   NML$T_PRMBUFFER, R0                          ; 0573
                          50            53            50  C3 000E9         SUBL3   R0, PTR, TOTLEN                              ;
                                               0101   8F  BB 000ED         PUSHR   #^M<R0,R8>                                   ; 0581
                          7E            57 000000C0   8F  C9 000F1         BISL3   #192, CM_COUNT, -(SP)                        ; 0580
                                        7E            66  3C 000F9         MOVZWL  (R6), -(SP)                                  ; 0579
                                        7E       08   AC  7D 000FC         MOVQ    BUFDSC, -(SP)                                ; 0577
                            00000000G   00            06  FB 00100         CALLS   #6, NML$ADDMSGPRM                            ;
                                        50            01  D0 00107         MOVL    #1, R0                                       ; 0584
                                                          04 0010A         RET                                                  ; 0586

; Routine Size:  267 bytes,    Routine Base:  $CODE$ + 013D
```

```
1 %SBTTL 'NML$SHOOBJPRV  Get object privilege mask'
1 GLOBAL ROUTINE NML$SHOOBJPRV (SEM_LIST, BUFDSC, MSGSIZE, DATDSC, DATPTR)=
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine gets the privilege list (NMA$C_PCOB_PRV) for a network
1 !       object and adds it to the output message buffer.
1 !
1 !       Currently, only the first longword of the privilege mask can be
1 !       set so that is all that is returned.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       SEM_LIST        Parameter semantic table entry address.
1 !       BUFDSC          Output message buffer descriptor address.
1 !       MSGSIZE         Address of current output message size.
1 !       DATDSC          QIO buffer descriptor address.
1 !       DATPTR          Current pointer into QIO data buffer.
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       The output message buffer contains the object privilege mask.
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       Always returns success (NML$_STS_SUC).
1 !
1 ! SIDE EFFECTS:
1 !
1 !       Destroys the contents of NML$T_PRMBUFFER.
1 !
1 !--
1
2     BEGIN
2
2     MAP
2         SEM_LIST : REF BLOCK [, BYTE];
2
2     IF .(..DATPTR)<0,32> NEQU -1
2     THEN
2         NML$ADDMSGPRM ( .BUFDSC,
2                         .MSGSIZE,
2                         .SEM_LIST [PST$W_DATAID],
2                         .SEM_LIST [PST$B_DATATYPE] OR 4,
2                         4,
2                         ..DATPTR);
2
2     .DATPTR = ..DATPTR + 4;
2
2     RETURN NML$_STS_SUC
2
```

```
;  651            0644  1      END;                                     ! End of NML$SHOOBJPRV



                                                 0004 00000          .ENTRY  NML$SHOOBJPRV, Save R2                                ; 0588
                                    52      14   AC  D0 00002          MOVL    DATPTR, R2                                            ; 0631
                          FFFFFFFF  8F      00   B2  D1 00006          CMPL    @0(R2), #-1                                           ;
                                                 1E  13 0000E          BEQL    1$                                                    ;
                                                 62  DD 00010          PUSHL   (R2)                                                  ; 0638
                                                 04  DD 00012          PUSHL   #4                                                    ; 0633
                                    50      04   AC  D0 00014          MOVL    SEM_LIST, R0                                          ; 0636
                                    51      03   A0  9A 00018          MOVZBL  3(R0), R1                                             ;
                       7E           51           04  C9 0001C          BISL3   #4, R1, -(SP)                                         ;
                                    7E           60  3C 00020          MOVZWL  (R0), -(SP)                                           ; 0635
                                    7E      08   AC  7D 00023          MOVQ    BUFDSC, -(SP)                                         ; 0633
                          00000000G 00           06  FB 00027          CALLS   #6, NML$ADDMSGPRM                                     ;
                                    62           04  C0 0002E 1$:      ADDL2   #4, (R2)                                              ; 0640
                                    50           01  D0 00031          MOVL    #1, R0                                                ; 0642
                                                 04  00034             RET                                                           ; 0644

; Routine Size:  53 bytes,    Routine Base:  $CODE$ + 0248
```

```
1 %SBTTL 'NML$SHOSERVPASS  Get service password'
1 GLOBAL ROUTINE NML$SHOSERVPASS (SEM_LIST, BUFDSC, MSGSIZE, DATDSC, DATPTR)=
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine gets the service password (NMA$C_PCNO_SPA) for the
1 !       remote node and adds it to the output message as a hexadecimal
1 !       number.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       SEM_LIST        Parameter semantic table entry address.
1 !       BUFDSC          Output message buffer descriptor address.
1 !       MSGSIZE         Address of current output message size.
1 !       DATDSC          QIO buffer descriptor address.
1 !       DATPTR          Current pointer into QIO data buffer.
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       The output message buffer contains the hex service password.
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       Always returns success (NML$_STS_SUC).
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !
1 !--
1
2     BEGIN
2
2     MAP
2         SEM_LIST : REF BLOCK [, BYTE];
2
2     LOCAL
2         PRMSIZE;
2
2     PRMSIZE = .(..DATPTR)<0,16>;
2     .DATPTR = ..DATPTR + 2;
2 !
2 ! If the length is zero then the parameter is not set.
2 !
2     IF .PRMSIZE EQLU 0
2     THEN
2         RETURN NML$_STS_PTY;
2 !
2 ! Add the parameter to the message.
2 !
2     NML$ADDMSGPRM (.BUFDSC,
;  710        0702  2                          .MSGSIZE,
;  711        0703  2                          .SEM_LIST [PST$W_DATAID],
;  712        0704  2                          .SEM_LIST [PST$B_DATATYPE] OR .PRMSIZE,
;  713        0705  2                          .PRMSIZE,
;  714        0706  2                          ..DATPTR);
;  715        0707  2
;  716        0708  2         .DATPTR = ..DATPTR + .PRMSIZE;
;  717        0709  2
;  718        0710  2         RETURN NML$_STS_SUC
;  719        0711  2
;  720        0712  1         END;                                    ! End of NML$SHOSERVPASS



                                                  0004 00000          .ENTRY   NML$SHOSERVPASS, Save R2                      ; 0646
                            50        14     BC   D0 00002          MOVL     @DATPTR, R0                                   ; 0690
                            52               60   3C 00006          MOVZWL   (R0), PRMSIZE                                 ;
                     14     BC               02   C0 00009          ADDL2    #2, @DATPTR                                   ; 0691
                                             52   D5 0000D          TSTL     PRMSIZE                                       ; 0695
                                             04   12 0000F          BNEQ     1$                                            ;
                            50               0C   CE 00011          MNEGL    #12, R0                                       ; 0697
                                                  04 00014          RET                                                    ;
                                      14     BC   DD 00015 1$:      PUSHL    @DATPTR                                       ; 0706
                                             52   DD 00018          PUSHL    PRMSIZE                                       ; 0705
                            50        04     AC   D0 0001A          MOVL     SEM_LIST, R0                                  ; 0704
                            51        03     A0   9A 0001E          MOVZBL   3(R0), R1                                     ;
              7E            51               52   C9 00022          BISL3    PRMSIZE, R1, -(SP)                            ;
                            7E               60   3C 00026          MOVZWL   (R0), -(SP)                                   ; 0703
                            7E        08     AC   7D 00029          MOVQ     BUFDSC, -(SP)                                 ; 0701
                 00000000G  00               06   FB 0002D          CALLS    #6, NML$ADDMSGPRM                             ;
                     14     BC               52   C0 00034          ADDL2    PRMSIZE, @DATPTR                              ; 0708
                            50               01   D0 00038          MOVL     #1, R0                                        ; 0710
                                                  04 0003B          RET                                                    ; 0712

; Routine Size:  60 bytes,    Routine Base:  $CODE$ + 027D
```

```
1 %SBTTL 'NML$SHOLINEID  Get line id'
1 GLOBAL ROUTINE NML$SHOLINEID (SEM_LIST, BUFDSC, MSGSIZE, DATDSC, DATPTR)=
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine reads the line id string and converts it from
1 !       VMS format to DNA format and then adds it to the output 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          QIO buffer descriptor address.
1 !       DATPTR          Current pointer into QIO data buffer.
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       The output message contains the DNA line id.
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       Always returns success (NML$_STS_SUC).
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !
1 !--
1
2     BEGIN
2
2     MAP
2         SEM_LIST : REF BLOCK [, BYTE];
2
2     LOCAL
2         PRMSIZE;
2
2     PRMSIZE = .(..DATPTR)<0,16>;
2     .DATPTR = ..DATPTR + 2;
2 !
2 ! If the length is zero then the parameter is not set.
2 !
2     IF .PRMSIZE EQLU 0
2     THEN
2         RETURN NML$_STS_PTY;
2 !
2 ! Add the parameter to the message.
2 !
2     NML$ADDMSGPRM (     .BUFDSC,
2                         .MSGSIZE,
;   779         0770  2                        .SEM_LIST [PST$W_DATAID],
;   780         0771  2                        .SEM_LIST [PST$B_DATATYPE],
;   781         0772  2                        .PRMSIZE,
;   782         0773  2                        ..DATPTR);
;   783         0774  2
;   784         0775  2    .DATPTR = ..DATPTR + .PRMSIZE;
;   785         0776  2
;   786         0777  2    RETURN NML$_STS_SUC
;   787         0778  2
;   788         0779  1    END;                                     ! End of NML$SHOLINEID



                                              0004 00000        .ENTRY  NML$SHOLINEID, Save R2                                    ; 0714
                    50        14        BC    D0 00002          MOVL    @DATPTR, R0                                               ; 0757
                    52                  60    3C 00006          MOVZWL  (R0), PRMSIZE                                             ;
              14    BC                  02    C0 00009          ADDL2   #2, @DATPTR                                               ; 0758
                                        52    D5 0000D          TSTL    PRMSIZE                                                   ; 0762
                                        04    12 0000F          BNEQ    1$                                                        ;
                    50                  0C    CE 00011          MNEGL   #12, R0                                                   ; 0764
                                              04 00014          RET                                                               ;
                              14        BC    DD 00015 1$:      PUSHL   @DATPTR                                                   ; 0773
                                        52    DD 00018          PUSHL   PRMSIZE                                                   ; 0772
                    50        04        AC    D0 0001A          MOVL    SEM_LIST, R0                                              ; 0771
                    7E        03        A0    9A 0001E          MOVZBL  3(R0), -(SP)                                              ;
                    7E                  60    3C 00022          MOVZWL  (R0), -(SP)                                               ; 0770
                    7E        08        AC    7D 00025          MOVQ    BUFDSC, -(SP)                                             ; 0768
       00000000G    00                  06    FB 00029          CALLS   #6, NML$ADDMSGPRM                                         ;
              14    BC                  52    C0 00030          ADDL2   PRMSIZE, @DATPTR                                          ; 0775
                    50                  01    D0 00034          MOVL    #1, R0                                                    ; 0777
                                              04 00037          RET                                                               ; 0779

; Routine Size:  56 bytes,    Routine Base:  $CODE$ + 02B9
;  790      0780   1 %SBTTL 'NML$SKIPLONG   Skip longword in QIO P4 buffer'
;  791      0781   1 GLOBAL ROUTINE NML$SKIPLONG (SEM_LIST, BUFDSC, MSGSIZE, DATDSC, DATPTR)=
;  792      0782   1
;  793      0783   1 !++
;  794      0784   1 ! FUNCTIONAL DESCRIPTION:
;  795      0785   1 !
;  796      0786   1 !       This routine skips (advances the pointer past) a byte, word, or
;  797      0787   1 !       longword parameter in the QIO P4 buffer.  Note that the ACP always
;  798      0788   1 !       returns these parameters in a longword.
;  799      0789   1 !
;  800      0790   1 ! FORMAL PARAMETERS:
;  801      0791   1 !
;  802      0792   1 !       SEM_LIST        Parameter semantic table entry address.
;  803      0793   1 !       BUFDSC          Output message buffer descriptor address.
;  804      0794   1 !       MSGSIZE         Address of current output message size.
;  805      0795   1 !       DATDSC          QIO buffer descriptor address.
;  806      0796   1 !       DATPTR          Current pointer into QIO data buffer.
;  807      0797   1 !
;  808      0798   1 ! IMPLICIT INPUTS:
;  809      0799   1 !
;  810      0800   1 !       NONE
;  811      0801   1 !
;  812      0802   1 ! IMPLICIT OUTPUTS:
;  813      0803   1 !
;  814      0804   1 !       NONE
;  815      0805   1 !
;  816      0806   1 ! ROUTINE VALUE:
;  817      0807   1 ! COMPLETION CODES:
;  818      0808   1 !
;  819      0809   1 !       Always returns success (NML$_STS_SUC).
;  820      0810   1 !
;  821      0811   1 ! SIDE EFFECTS:
;  822      0812   1 !
;  823      0813   1 !       NONE
;  824      0814   1 !
;  825      0815   1 !--
;  826      0816   1
;  827      0817   2   BEGIN
;  828      0818   2
;  829      0819   2   .DATPTR = ..DATPTR + 4;
;  830      0820   2
;  831      0821   2   RETURN NML$_STS_SUC
;  832      0822   2
;  833      0823   1   END;                                  ! End of NML$SKIPLONG



                                     0000 00000        .ENTRY  NML$SKIPLONG, Save nothing                    : 0781
                  14   BC          04  C0 00002        ADDL2   #4, @DATPTR                                   : 0819
                       50          01  D0 00006        MOVL    #1, R0                                        : 0821
                                       04 00009        RET                                                   : 0823

; Routine Size:  10 bytes,    Routine Base:  $CODE$ + 02F1
;  835         0824   1 %SBTTL 'NML$SKIPSTRING   Skip string parameter'
;  836         0825   1 GLOBAL ROUTINE NML$SKIPSTRING (SEM_LIST, BUFDSC, MSGSIZE, DATDSC, DATPTR)=
;  837         0826   1
;  838         0827   1 !++
;  839         0828   1 ! FUNCTIONAL DESCRIPTION:
;  840         0829   1 !
;  841         0830   1 !       This routine skips (advances the pointer past) a string parameter
;  842         0831   1 !       in the QIO buffer.
;  843         0832   1 !
;  844         0833   1 ! FORMAL PARAMETERS:
;  845         0834   1 !
;  846         0835   1 !       SEM_LIST        Parameter semantic table entry address.
;  847         0836   1 !       BUFDSC          Output message buffer descriptor address.
;  848         0837   1 !       MSGSIZE         Address of current output message size.
;  849         0838   1 !       DATDSC          QIO buffer descriptor address.
;  850         0839   1 !       DATPTR          Current pointer into QIO data buffer.
;  851         0840   1 !
;  852         0841   1 ! IMPLICIT INPUTS:
;  853         0842   1 !
;  854         0843   1 !       NONE
;  855         0844   1 !
;  856         0845   1 ! IMPLICIT OUTPUTS:
;  857         0846   1 !
;  858         0847   1 !       NONE
;  859         0848   1 !
;  860         0849   1 ! ROUTINE VALUE:
;  861         0850   1 ! COMPLETION CODES:
;  862         0851   1 !
;  863         0852   1 !       Always returns success (NML$_STS_SUC).
;  864         0853   1 !
;  865         0854   1 ! SIDE EFFECTS:
;  866         0855   1 !
;  867         0856   1 !       NONE
;  868         0857   1 !
;  869         0858   1 !--
;  870         0859   1
;  871         0860   2     BEGIN
;  872         0861   2
;  873         0862   2     LOCAL
;  874         0863   2         LEN;
;  875         0864   2
;  876         0865   2     LEN = .(..DATPTR)<0,16>;
;  877         0866   2     .DATPTR = ..DATPTR + 2;
;  878         0867   2     .DATPTR = ..DATPTR + .LEN;
;  879         0868   2
;  880         0869   2     RETURN NML$_STS_SUC
;  881         0870   2
;  882         0871   1     END;                                 ! End of NML$SKIPSTRING



                                       0000 00000         .ENTRY  NML$SKIPSTRING, Save nothing                         ; 0825
                 50       14     BC  D0 00002         MOVL    @DATPTR, R0                                           ; 0865
                 50              60  3C 00006         MOVZWL  (R0), LEN                                             ;
            14   BC              02  C0 00009         ADDL2   #2, @DATPTR                                           ; 0866
```

```
                              14    BC              50  C0 0000D            ADDL2   LEN, @DATPTR                               ; 0867
                                    50              01  D0 00011            MOVL    #1, R0                                     ; 0869
                                                        04 00014            RET                                                ; 0871

; Routine Size:  21 bytes,    Routine Base:  $CODE$ + 02FB
;   884         0872  1 %SBTTL 'NML$SHOEXEPARAM   Show executor parameter'
;   885         0873  1 GLOBAL ROUTINE NML$SHOEXEPARAM (SEM_LIST, BUFDSC, MSGSIZE, DATDSC, DATPTR)=
;   886         0874  1
;   887         0875  1 !++
;   888         0876  1 ! FUNCTIONAL DESCRIPTION:
;   889         0877  1 !
;   890         0878  1 !       This routine adds a parameter from the executor data buffer to the
;   891         0879  1 !       output message.
;   892         0880  1 !
;   893         0881  1 ! FORMAL PARAMETERS:
;   894         0882  1 !
;   895         0883  1 !       SEM_LIST        Parameter semantic table entry address.
;   896         0884  1 !       BUFD<C          Output message buffer descriptor address.
;   897         0885  1 !       MSGSIZE         Address of current output message size.
;   898         0886  1 !       DATDSC          QIO buffer descriptor address.
;   899         0887  1 !       DATPTR          Current pointer into QIO data buffer.
;   900         0888  1 !
;   901         0889  1 ! ROUTINE VALUE:
;   902         0890  1 ! COMPLETION CODES:
;   903         0891  1 !
;   904         0892  1 !       Always returns success (NML$_STS_SUC).
;   905         0893  1 !
;   906         0894  1 !--
;   907         0895  1
;   908         0896  2 BEGIN
;   909         0897  2
;   910         0898  2 MAP
;   911         0899  2     SEM_LIST: REF BBLOCK;
;   912         0900  2
;   913         0901  2 LOCAL
;   914         0902  2     SUBRTN;
;   915         09)3  2
;   916         0904  2 SELECTONEU .SEM_LIST [PST$W_DATAID] OF
;   917         0905  2
;   918         0906  2     SET
;   919         0907  2     [NMA$C_PCNO_SAD]:   SUBRTN = NML$SHORANGE;
;   920         0908  2     [NMA$C_PCNO_ALI]:   SUBRTN = NML$SHONODEID;
;   921         0909  2     [OTHERWISE]:        SUBRTN = NML$SHOPARAM;
;   922         0910  2     TES;
;   923         0911  2 !
;   924         0912  2 ! Call the show parameter routine using the executor data descriptor.
;   925         0913  2 !
;   926         0914  2 (.SUBRTN) (.SEM_LIST,
;   927         0915  2            .BU:DSC,
;   928         0916  2            .MSGSIZE,
;   929         0917  2            NML$GQ_EXEDATDSC,
;   930         0918  2            NML$GL_EXEDATPTR);
;   931         0919  2
;   932         0920  2 RETURN NML$_STS_SUC
;   933         0921  2
;   934         0922  1 END;                            ! End of NML$SHOEXEPARAM


                                      0004 00000         .ENTRY  NML$SHOEXEPARAM, Save R2                 ; 0873
```

```
                        52        04    AC  D0 00002           MOVL    SEM_LIST, R2                               ; 0904
                        50              62  3C 00006           MOVZWL  (R2), R0                                   ;
                 038F   8F              50  B1 00009           CMPW    R0, #911                                   ; 0907
                                        09  12 0000E           BNEQ    1$                                         ;
                        51 00000000V    00  9E 00010           MOVAB   NML$SHORANGE, SUBRTN                       ;
                                        13  11 00017           BRB     3$                                         ;
                 0AB5   8F              50  B1 00019 1$:       CMPW    R0, #2741                                  ; 0908
                                        07  12 0001E           BNEQ    2$                                         ;
                        51      FE09    CF  9E 00020           MOVAB   NML$SHONODEID, SUBRTN                      ;
                                        05  11 00025           BRB     3$                                         ;
                        51      FCC5    CF  9E 00027 2$:       MOVAB   NML$SHOPARAM, SUBRTN                       ; 0909
                           00000000G    00  9F 0002C 3$:       PUSHAB  NML$GL_EXEDATPTR                           ; 0914
                           00000000G    00  9F 00032           PUSHAB  NML$GQ_EXEDATDSC                           ;
                        7E        08    AC  7D 00038           MOVQ    BUFDSC, -(SP)                              ; 0915
                                        52  DD 0003C           PUSHL   R2                                         ; 0914
                        61              05  FB 0003E           CALLS   #5, (SUBRTN)                               ;
                        50              01  D0 00041           MOVL    #1, R0                                     ; 0920
                                            04 00044           RET                                                ; 0922

; Routine Size:  69 bytes,    Routine Base:  $CODE$ + 0310
```

```
1 %SBTTL 'NML$SHORANGE    Show range parameter'
1 GLOBAL ROUTINE NML$SHORANGE (SEM_LIST, BUFDSC, MSGSIZE, DATDSC, DATPTR) =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
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 !       DATPTR          Current pointer into QIO data buffer.
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       Always returns success (NML$_STS_SUC).
1 !
1 !--
1
2 BEGIN
2
2 MAP
2     SEM_LIST : REF BBLOCK;
2
2 LOCAL
2     CM_COUNT,
2     RANGE_BEGIN: WORD,
2     RANGE_END: WORD,
2     LENGTH,
2     PTR;
2
2 !
2 ! If the address value is -1 then the parameter is not set.
2 !
2 IF .(..DATPTR)<0,32> EQLU -1 THEN
3     BEGIN
3     .DATPTR = ..DATPTR + 4;
3     RETURN NML$_STS_PTY;
2     END;
2
2 RANGE_BEGIN = .(..DATPTR)<0,16>;
2 RANGE_END = .(..DATPTR)<16,32>;
2 PTR = NML$T_PRMBUFFER;
2 CM_COUNT = 1;
2
2 CH$WCHAR_A (2, PTR);
2 PTR = CH$MOVE (2, RANGE_BEGIN, .PTR);
2 !
2 ! If the range beginning = range end, don't include range end.
2 !
2 IF .RANGE_BEGIN NEQ .RANGE_END THEN
3     BEGIN
3     CM_COUNT = .CM_COUNT +1;
```

```
;  993          0980  3        CH$WCHAR_A (2, PTR);
;  994          0981  3        PTR = CH$MOVE (2, RANGE_END, .PTR);
;  995          0982  2        END;
;  996          0983  2
;  997          0984  2  LENGTH = .PTR - NML$T_PRMBUFFER;
;  998          0985  2  !
;  999          0986  2  ! Add coded multiple subaddresses field to output message.
; 1000          0987  2  !
; 1001          0988  2  NML$ADDMSGPRM (.BUFDSC,
; 1002          0989  2                 .MSGSIZE,
; 1003          0990  2                 .SEM_LIST [PST$W_DATAID],
; 1004          0991  2                 .SEM_LIST [PST$B_DATATYPE] OR .CM_COUNT,
; 1005          0992  2                 .LENGTH,
; 1006          0993  2                 NML$T_PRMBUFFER);
; 1007          0994  2
; 1008          0995  2  !
; 1009          0996  2  ! Increment past range value in P4 buffer.
; 1010          0997  2  !
; 1011          0998  2  .DATPTR = ..DATPTR + 4;
; 1012          0999  2
; 1013          1000  2  RETURN NML$_STS_SUC;
; 1014          1001  1  END;                             ! end of NML$SHORANGE



                                           001C 00000          .ENTRY  NML$SHORANGE, Save R2,R3,R4              ; 0924
                        54 00000000'    00  9E 00002          MOVAB   NML$T_PRMBUFFER, R4                       ;
                        50          14  BC  D0 00009          MOVL    @DATPTR, R0                               ; 0961
              FFFFFFFF  8F              60  D1 0000D          CMPL    (R0), #-1                                 ;
                                        08  12 00014          BNEQ    1$                                        ;
                   14   BC              04  C0 00016          ADDL2   #4, @DATPTR                               ; 0963
                        50              0C  CE 0001A          MNEGL   #12, R0                                   ; 0964
                                            04 0001D          RET                                               ;
                        51          14  BC  D0 0001E 1$:      MOVL    @DATPTR, R1                               ; 0967
                        53              61  B0 00022          MOVW    (R1), RANGE_BEGIN                         ;
      50          61    20              10  EF 00025          EXTZV   #16, #32, (R1), R0                        ; 0968
                        52              50  B0 0002A          MOVW    R0, RANGE_END                             ;
                        51              64  9E 0002D          MOVAB   NML$T_PRMBUFFER, PTR                      ; 0969
                        50              01  D0 00030          MOVL    #1, CM_COUNT                              ; 0970
                        81              02  90 00033          MOVB    #2, (PTR)+                                ; 0972
                        81              53  B0 00036          MOVW    RANGE_BEGIN, (PTR)+                       ; 0973
                        52              53  B1 00039          CMPW    RANGE_BEGIN, RANGE_END                    ; 0977
                                        08  13 0003C          BEQL    2$                                        ;
                                        50  D6 0003E          INCL    CM_COUNT                                  ; 0979
                        81              02  90 00040          MOVB    #2, (PTR)+                                ; 0980
                        81              52  B0 00043          MOVW    RANGE_END, (PTR)+                         ; 0981
                        52              64  9E 00046 2$:      MOVAB   NML$T_PRMBUFFER, R2                       ; 0984
                        51              52  C2 00049          SUBL2   R2, LENGTH                                ;
                                        12  BB 0004C          PUSHR   #^M<R1,R4>                                ; 0992
                        51          04  AC  D0 0004E          MOVL    SEM_LIST, R1                              ; 0991
                        52          03  A1  9A 00052          MOVZBL  3(R1), R2                                 ;
                  7E    52              50  C9 00056          BISL3   CM_COUNT, R2, -(SP)                       ;
                        7E              61  3C 0005A          MOVZWL  (R1), -(SP)                               ; 0990
                        7E          08  AC  7D 0005D          MOVQ    BUFDSC, -(SP)                             ; 0988
             00000000G  00              06  FB 00061          CALLS   #6, NML$ADDMSGPRM                         ;
```

```
                                 14  BC          04  C0 00068        ADDL2    #4, @DATPTR                                 ; 0998
                                     50          01  D0 0006C        MOVL     #1, R0                                      ; 1000
                                                     04 0006F        RET                                                  ; 1001

; Routine Size:  112 bytes,    Routine Base:  $CODE$ + 0355
```

```
1 %SBTTL 'NML$SHOCHANNELS    Show channels parameter'
1 GLOBAL ROUTINE NML$SHOCHANNELS (SEM_LIST, BUFDSC, MSGSIZE, DATDSC, DATPTR) =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !       This routine is called to format the parameter for X25 Protocol DTE
1 !       channels in the SHOW NICE response message.  It takes the string
1 !       returned by the ACP in the P4 buffer and reformats it into NICE in as
1 !       many channel pairs as were returned in the string.
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 !       DATPTR          Current pointer into QIO data buffer.
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       Always returns success (NML$_STS_SUC).
1 !
1 !--
1
2 BEGIN
2
2 MAP
2     SEM_LIST : REF BBLOCK;
2
2 LOCAL
2     QIO_CHAN_LEN,       ! Length of channels string in QIO P4 buffer.
2     PTR;
2
2 !
2 ! If the string length is 0 then the parameter is not set.
2 !
2 IF .(..DATPTR)<0,16> EQL 0 THEN
3     BEGIN
3     .DATPTR = ..DATPTR + 2;
3     RETURN NML$_STS_PTY;
2     END;
2
2 QIO_CHAN_LEN = .(..DATPTR)<0,16>;
2 .DATPTR = ..DATPTR + 2;
2 WHILE .QIO_CHAN_LEN GTR 0 DO
3     BEGIN
3     PTR = NML$T_PRMBUFFER;
3     !
3     ! Build a temporary buffer containing a channel pair.  Each element
3     ! in the channel pair consist of a parameter type field (2) and
3     ! a word of parameter value.
3     !
3     CH$WCHAR_A (2, PTR);
3     PTR = CH$MOVE (2, ..DATPTR, .PTR);
```

```
: 1073          1059  3       .DATPTR = ..DATPTR + 2;
: 1074          1060  3       CH$WCHAR_A (2, PTR);
: 1075          1061  3       PTR = CH$MOVE (2, ..DATPTR, .PTR);
: 1076          1062  3       .DATPTR = ..DATPTR + 2;
: 1077          1063  3       !
: 1078          1064  3       ! Add coded multiple subaddresses field to output message.
: 1079          1065  3       !
: 1080          1066  3       NML$ADDMSGPRM (.BUFDSC,
: 1081          1067  3                      .MSGSIZE,
: 1082          1068  3                      .SEM_LIST [PST$W_DATAID],
: 1083          1069  3                      .SEM_LIST [PST$B_DATATYPE] OR 2,
: 1084          1070  3                      6,
: 1085          1071  3                      NML$T_PRMBUFFER);
: 1086          1072  3
: 1087          1073  3       !
: 1088          1074  3       ! Decrement count of channel pairs left in QIO buffer.
: 1089          1075  3       !
: 1090          1076  3       QIO_CHAN_LEN = .QIO_CHAN_LEN - 4;
: 1091          1077  2       END;
: 1092          1078  2
: 1093          1079  2 RETURN NML$_STS_SUC;
: 1094          1080  1 END;                              ! end of NML$SHOCHANNELS



                                          007C 00000          .ENTRY  NML$SHOCHANNELS, Save R2,R3,R4,R5,R6          ; 1003
                   56 00000000'    00  9E 00002          MOVAB   NML$T_PRMBUFFER, R6                           ;
                   52        14    AC  D0 00009          MOVL    DATPTR, R2                                    ; 1041
                             00    B2  B5 0000D          TSTW    @0(R2)                                        ;
                                   07  12 00010          BNEQ    1$                                            ;
                   62              02  C0 00012          ADDL2   #2, (R2)                                      ; 1043
                   50              0C  CE 00015          MNEGL   #12, R0                                       ; 1044
                                       04 00018          RET                                                   ;
                   55        00    B2  3C 00019 1$:      MOVZWL  @0(R2), QIO_CHAN_LEN                          ; 1047
                   62              02  C0 0001D          ADDL2   #2, (R2)                                      ; 1048
                   54        04    AC  D0 00020          MOVL    SEM_LIST, R4                                  ; 1069
                                   55  D5 00024 2$:      TSTL    QIO_CHAN_LEN                                  ; 1049
                                   37  15 00026          BLEQ    3$                                            ;
                   53              66  9E 00028          MOVAB   NML$T_PRMBUFFER, PTR                          ; 1051
                   83              02  90 0002B          MOVB    #2, (PTR)+                                    ; 1057
                   83        00    B2  B0 0002E          MOVW    @0(R2), (PTR)+                                ; 1058
                   62              C2  C0 00032          ADDL2   #2, (R2)                                      ; 1059
                   83              02  90 00035          MOVB    #2, (PTR)+                                    ; 1060
                   83        00    B2  B0 00038          MOVW    @0(R2), (PTR)+                                ; 1061
                   62              02  C0 0003C          ADDL2   #2, (R2)                                      ; 1062
                                   56  DD 0003F          PUSHL   R6                                            ; 1066
                                   06  DD 00041          PUSHL   #6                                            ;
                   50        03    A4  9A 00043          MOVZBL  3(R4), R0                                     ; 1069
        7E         50              02  C9 00047          BISL3   #2, R0, -(SP)                                 ;
                   7E        04    BC  3C 0004B          MOVZWL  @SEM_LIST, -(SP)                              ; 1068
                   7E        08    AC  7D 0004F          MOVQ    BUFDSC, -(SP)                                 ; 1066
         00000000G 00              06  FB 00053          CALLS   #6, NML$ADDMSGPRM                             ;
                   55              04  C2 0005A          SUBL2   #4, QIO_CHAN_LEN                              ; 1076
                                   C5  11 0005D          BRB     2$                                            ; 1049
                   50              01  D0 0005F 3$:      MOVL    #1, R0                                        ; 1079
```

```
                                                    04 00062         RET                                      ; 1080

; Routine Size:  99 bytes,    Routine Base:  $CODE$ + 03C5
```

```
1 %SBTTL 'NML$SHOPWSET   Show password set indication'
1 GLOBAL ROUTINE NML$SHOPWSET (SEM_LIST, BUFDSC, MSGSIZE, DATDSC, DATPTR) =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !       This routine is called while processing a SHOW X25-SERVER DESTINATION
1 !       command - after the PSI ACP returns the destination's parameters.
1 !       If the password is set, it puts a password value of zero in the
1 !       NICE response message.  If the password is not set, it does not
1 !       add anything to the 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          QIO buffer descriptor address.
1 !       DATPTR          Current pointer into QIO data buffer.
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       Always returns success (NML$_STS_SUC).
1 !
1 !--
1
2 BEGIN
2
2 MAP
2     SEM_LIST : REF BBLOCK;
2
2 LOCAL
2     PASSWORD_LEN;
2
2 PASSWORD_LEN = .(..DATPTR)<0,16>;
2 IF .PASSWORD_LEN GTR 0 THEN
3     BEGIN
3     !
3     ! Add password to message with a value of 0.  This indicates simply that
3     ! the password is set, without actually returning the password.
3     !
3     NML$ADDMSGPRM (.BUFDSC,
3                    .MSGSIZE,
3                    .SEM_LIST [PST$W_DATAID],
3                    .SEM_LIST [PST$B_DATATYPE],
3                    1,
3                    UPLIT (0));
2     END;
2 !
2 ! Increment past the password in the buffer.
2 !
2 .DATPTR = ..DATPTR + .PASSWORD_LEN + 2;
2
2 RETURN NML$_STS_SUC;
1 END;                          ! end of NML$SHOPWSET
```

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

                                   00000000  00008 P.AAB:   .LONG   0                                              ;


                                                              .PSECT  $CODE$,NOWRT,2

                                        001C 00000            .ENTRY  NML$SHOPWSET, Save R2,R3,R4                 ; 1082
                     54       14     AC  D0 00002            MOVL    DATPTR, R4                                   ; 1117
                     52              64  D0 00006            MOVL    (R4), R2                                     ;
                     53              62  3C 00009            MOVZWL  (R2), PASSWORD_LEN                           ;
                                     1E  15 0000C            BLEQ    1$                                           ; 1118
                        00000000'    00  9F 0000E            PUSHAB  P.AAB                                        ; 1129
                                     01  DD 00014            PUSHL   #1                                           ; 1124
                     50       04     AC  D0 00016            MOVL    SEM_LIST, R0                                 ; 1127
                     7E       03     A0  9A 0001A            MOVZBL  3(R0), -(SP)                                 ;
                     7E              60  3C 0001E            MOVZWL  (R0), -(SP)                                  ; 1126
                     7E       08     AC  7D 00021            MOVQ    BUFDSC, -(SP)                                ; 1124
           00000000G 00              06  FB 00025            CALLS   #6, NML$ADDMSGPRM                            ;
                     64       02   A342  9E 0002C 1$:        MOVAB   2(PASSWORD_LEN)[R2], (R4)                    ; 1134
                     50              01  D0 00031            MOVL    #1, R0                                       ; 1136
                                         04 00034            RET                                                  ; 1137

; Routine Size:  53 bytes,    Routine Base:  $CODE$ + 0428
```

```
1 %SBTTL 'NML$SHOCOUNTERS   Show entity counters'
1 GLOBAL ROUTINE NML$SHOCOUNTERS (SEM_LIST, BUFDSC, MSGSIZE, DATDSC, DATPTR) =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine puts counter parameters into the response message.
1 !       Since NETACP formats the counters in NICE format, and returns them
1 !       as a string, this simply involves moving the string into the
1 !       response message with no parameter type or string length.
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 !       DATPTR          Current pointer into QIO data buffer.
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       Message buffer contains counter parameters.
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       Always returns success (NML$_STS_SUC).
1 !
1 !--
1
2 BEGIN
2
2 MAP
2       SEM_LIST : REF BLOCK [, BYTE];
2
2 LOCAL
2       LEN;
2
2 LEN = .(..DATPTR)<0,16>;
2 .DATPTR = ..DATPTR + 2;
2 !
2 ! If the length is zero then no counters were returned.
2 !
2 IF .LEN EQL 0
2 THEN
2       RETURN NML$_STS_SUC;
2
2 NML$ADDMSGCOU ( .BUFDSC,
2               .MSGSIZE,
2               .LEN,
2               ..DATPTR);
2
2 .DATPTR = ..DATPTR + .LEN;
2
2 RETURN NML$_STS_SUC
2
1 END;                              ! End of NML$SHOCOUNiERS
```

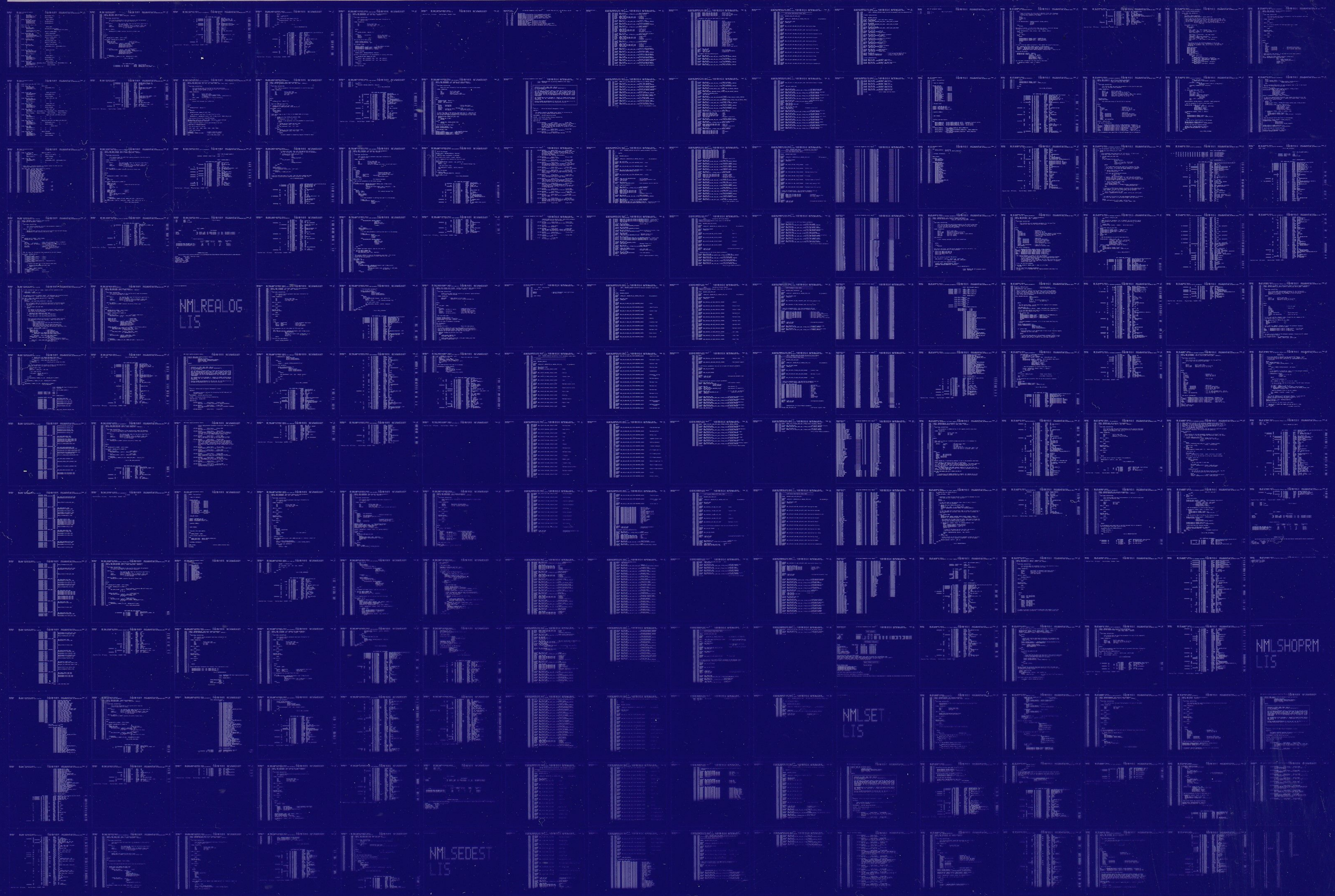
```
                                0004 00000          .ENTRY  NML$SHOCOUNTERS, Save R2                              ; 1139
                  50        14  BC   D0 00002          MOVL    @DATPTR, R0                                           ; 1176
                  52            60   3C 00006          MOVZWL  (R0), LEN                                             ;
              14  BC            02   C0 00009          ADDL2   #2, @DATPTR                                           ; 1177
                                52   D5 0000D          TSTL    LEN                                                   ; 1181
                                14   13 0000F          BEQL    1$                                                    ;
                            14  BC   DD 00011          PUSHL   @DATPTR                                               ; 1188
                                52   DD 00014          PUSHL   LEN                                                   ; 1187
                  7E        08  AC   7D 00016          MOVQ    BUFDSC, -(SP)                                         ; 1185
       00000000G  00            04   FB 0001A          CALLS   #4, NML$ADDMSGCOU                                     ;
              14  BC            52   C0 00021          ADDL2   LEN, @DATPTR                                          ; 1190
                  50            01   D0 00025 1$:      MOVL    #1, R0                                                ; 1192
                                     04 00028          RET                                                           ; 1194

; Routine Size:  41 bytes,    Routine Base:  $CODE$ + 045D
```

```
1 %SBTTL 'NML$SHOOWNER  Translate Data Link Mapping bit to Owner'
1 GLOBAL ROUTINE NML$SHOOWNER (SEM_LIST, BUFDSC, MSGSIZE, DATDSC, DATPTR)=
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine is called when doing a SHOW CIRC CHAR.  It looks
1 !       at the bit value returned by the ACP for DLM (Data Link Mapping),
1 !       and, if it's set, returns an OWNER parameter value for the
1 !       executor node to NCP.  The executor node is the only value
1 !       currently allowed for OWNER.
1 !
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 !       DATPTR          Current pointer into QIO data buffer.
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       Always returns success (NML$_STS_SUC).
1 !
1 !--
1
2 BEGIN
2
2 MAP
2     SEM_LIST : REF BLOCK [, BYTE];
2
2 BIND EXECUTOR = UPLIT BYTE
2       (NMA$M_PTY_COD+1, NMA$C_ENT_NOD,        ! Entity type = node
2       2, WORD (0));                           ! Node address = 0 (executor)
2
2 .DATPTR = ..DATPTR + 4;
2 !
2 ! If th= address value is -1 then the owner is not set.
2 ! If the bit value is clear, then there is no owner specified.
2 !
2 IF .(..DATPTR - 4)<0,32> EQLU -1 OR
3    NOT (.(..DATPTR - 4)<0,32>)
2 THEN
2     RETURN NML$_STS_PTY;
2
2 !
2 ! Add coded multiple executor node id field to output message.
2 !
2 NML$ADDMSGPRM (.BUFDSC,
2               .MSGSIZE,
2               .SEM_LIST [PST$W_DATAID],
2               .SEM_LIST [PST$B_DATATYPE] OR 2,
2               5,
2               EXECUTOR);
2
```

```
; 1269          1252  2 RETURN NML$_STS_SUC
; 1270          1253  1 END;                            ! End of NML$SHOOWNER


                                                                          .PSECT  $PLIT$,NOWRT,NOEXE,2

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

                                                                  EXECUTOR=           P.AAC


                                                                          .PSECT  $CODE$,NOWRT,2

                                                       0000 00000         .ENTRY  NML$SHOOWNER, Save nothing                  ; 1196
                                  14   BC            04  C0 00002         ADDL2   #4, @DATPTR                                 ; 1232
                                       50      14    BC  D0 00006         MOVL    @DATPTR, R0                                 ; 1237
                            FFFFFFFF   8F      FC    A0  D1 0000A         CMPL    -4(R0), #-1                                 ;
                                                     04  13 00012         BEQL    1$                                          ;
                                       04      FC    A0  E8 00014         BLBS    -4(R0), 2$                                  ; 1238
                                       50            0C  CE 00018 1$:     MNEGL   #12, R0                                     ; 1240
                                                         04 0001B         RET                                                 ;
                                          0^000000'  00  9F 0001C 2$:     PUSHAB  EXECUTOR                                    ; 1245
                                                     05  DD 00022         PUSHL   #5                                          ;
                                       50      04    AC  D0 00024         MOVL    SEM_LIST, R0                                ; 1248
                                       51      03    A0  9A 00028         MOVZBL  3(R0), R1                                   ;
                         7E            51            02  C9 0002C         BISL3   #2, R1, -(SP)                               ;
                                       7E            60  3C 00030         MOVZWL  (R0), -(SP)                                 ; 1247
                                       7E      08    AC  7D 00033         MOVQ    BUFDSC, -(SP)                               ; 1245
                            00000000G  00            06  FB 00037         CALLS   #6, NML$ADDMSGPRM                           ;
                                       50            01  D0 0003E         MOVL    #1, R0                                      ; 1252
                                                         04 00041         RET                                                 ; 1253

; Routine Size:  66 bytes,    Routine Base:  $CODE$ + 0486
; 1272          1254  1 END
; 1273          1255  1
; 1274          1256  0 ELUDOM



;                                         PSECT SUMMARY
;
;       Name                        Bytes                       Attributes
;
;   $OWN$                             256  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;   $PLIT$                             17  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;   $CODE$                           1224  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        22         6         27          00:00.1
;   _$255$DUA28:[SHRLIB]NMALIBRY.L32;1           887        15         1         47          00:00.2
;   _$255$DUA28:[SHRLIB]NET.L32;1               1279         3         0         63          00:00.3
;   _$255$DUA28:[SYSLIB]STARLET.L32;1           9776         0         0        581          00:03.3



;                                          COMMAND QUALIFIERS

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

; Size:          1224 code + 273 data bytes
; Run Time:         00:27.7
; Elapsed Time:     01:04.7
; Lines/CPU Min:      2721
; Lexemes/CPU-Min: 10955
; Memory Used:  128 pages
; Compilation Complete
```

0286 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



NMLREALOG
LIS

NMLSHOPRM
LIS

NMLSET
LIS

NMLSEDEST
LIS

0287 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY