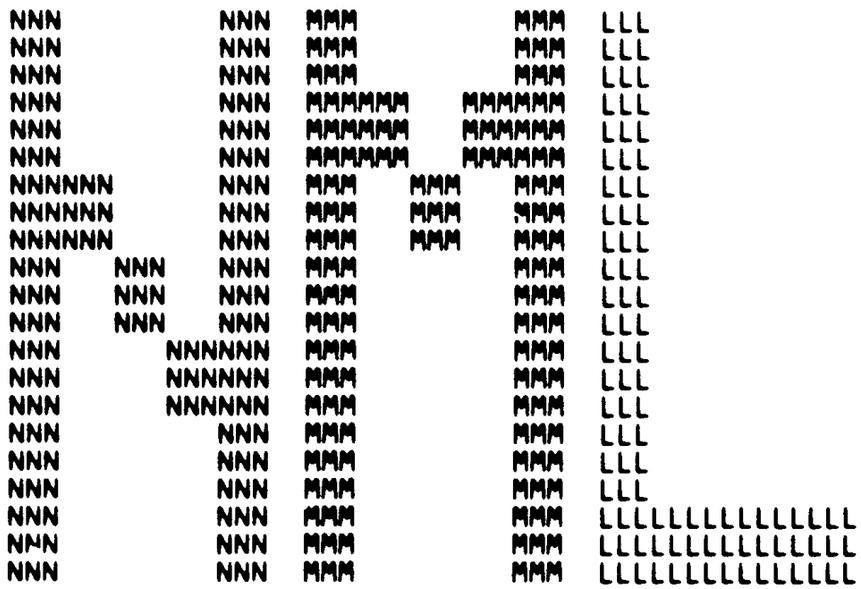
NNN NNN MMM MMM LLL
NNNNNN NNN MMM MMM LLL
NNN NNN NNN MMMMMMM MMMMMM LLL
NNN NNN MMM MMM LLL
NNN NNN MMM MMM LLLLLLLLLLLLLLL
NNN NNN MMM MMM LLLLLLLLLLLLLLL
NNN NNN MMM MMM LLLLLLLLLLLLLLL

FILEID**NMLV2COMP

```
NN       NN  MM       MM  LL          VV       VV    222222      CCCCCCCC    000000    MM       MM  PPPPPPPP
NN       NN  MM       MM  LL          VV       VV    222222      CCCCCCCC    000000    MM       MM  PPPPPPPP
NN       NN  MMMM   MMMM  LL          VV       VV  22      22  CC          00      00  MMMM   MMMM  PP      PP
NN       NN  MMMM   MMMM  LL          VV       VV  22      22  CC          00      00  MMMM   MMMM  PP      PP
NNNN     NN  MM  MM  MM  LL          VV       VV          22  CC          00      00  MM  MM  MM  PP      PP
NNNN     NN  MM  MM  MM  LL          VV       VV          22  CC          00      00  MM  MM  MM  PP      PP
NN  NN   NN  MM      MM  LL          VV       VV        22    CC          00      00  MM      MM  PPPPPPPP
NN  NN   NN  MM      MM  LL          VV       VV        22    CC          00      00  MM      MM  PPPPPPPP
NN    NNNN  MM      MM  LL          VV       VV      22      CC          00      00  MM      MM  PP
NN    NNNN  MM      MM  LL          VV       VV      22      CC          00      00  MM      MM  PP
NN      NN  MM      MM  LL            VV   VV      22        CC          00      00  MM      MM  PP                ....
NN      NN  MM      MM  LL            VV   VV      22        CC          00      00  MM      MM  PP                ....
NN      NN  MM      MM  LLLLLLLLLL      VV      2222222222    CCCCCCCC    000000    MM      MM  PP                ....
NN      NN  MM      MM  LLLLLLLLLL      VV      2222222222    CCCCCCCC    000000    MM      MM  PP                ....

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
```

```
;  1          0001  0 %TITLE 'Process NICE V2.0 requests'
;  2          0002  0 MODULE NML$V2COMP (IDENT = 'V04-000',
;  3          0003  0                    ADDRESSING_MODE (NONEXTERNAL=GENERAL),
;  4          0004  0                    ADDRESSING_MODE (EXTERNAL=GENERAL)) =
;  5          0005  1 BEGIN
;  6          0006  1 !
;  7          0007  1 !*****************************************************************************
;  8          0008  1 !*                                                                           *
;  9          0009  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
; 10          0010  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
; 11          0011  1 !*  ALL RIGHTS RESERVED.                                                     *
; 12          0012  1 !*                                                                           *
; 13          0013  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
; 14          0014  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
; 15          0015  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
; 16          0016  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
; 17          0017  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
; 18          0018  1 !*  TRANSFERRED.                                                             *
; 19          0019  1 !*                                                                           *
; 20          0020  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
; 21          0021  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
; 22          0022  1 !*  CORPORATION.                                                             *
; 23          0023  1 !*                                                                           *
; 24          0024  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
; 25          0025  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
; 26          0026  1 !*                                                                           *
; 27          0027  1 !*                                                                           *
; 28          0028  1 !*****************************************************************************
; 29          0029  1 !
; 30          0030  1
; 31          0031  1 !++
; 32          0032  1 ! FACILITY:  DECnet-VAX V2.0 Network Management Listener
; 33          0033  1 !
; 34          0034  1 ! ABSTRACT:
; 35          0035  1 !
; 36          0036  1 !       This module contains the entry points for the
; 37          0037  1 !       the portions of NML dealing with NICE V2 messages.
; 38          0038  1 !
; 39          0039  1 ! ENVIRONMENT:  VAX/VMS Operating System
; 40          0040  1 !
; 41          0041  1 ! AUTHOR:  Tim Halvorsen & Kathy Perko, October 1981
; 42          0042  1 !
; 43          0043  1 ! MODIFIED BY:
; 44          0044  1 !
; 45          0045  1 !       V03-004 MKP0008         Kathy Perko     2-Jan-1984
; 46          0046  1 !               Get rid of definition for NML$K_ENTBUFLEN since it's in
; 47          0047  1 !               NMLLIB now.
; 48          0048  1 !
; 49          0049  1 !       V03-003 MKP0007         Kathy Perko     29-June-1982
; 50          0050  1 !               Redo SHOW LINKS to use qualifier logic for WITH NODE commands.
; 51          0051  1 !               Rename some EIT fields.
; 52          0052  1 !
; 53          0053  1 !       V03-002 MKP0006         Kathy Perko     28-April-1982
; 54          0054  1 !               Delete start key and add second search key to NETACP
; 55          0055  1 !               QIO interface.
; 56          0056  1 !
; 57          0057  1 !       V03-001 MKP0005         Kathy Perko     17-Mar-1982
```

```
;    58          0058  1 !               Fix V2-V3 SHOW LINE so that it handles multidrop
;    59          0059  1 !               circuits.  I.E. it returns info for DMP-0.1, DMP-0.2, etc.
;    60          0060  1 !
;    61          0061  1 !       V02-004 MKP0004         Kathy Perko     1-Mar-1982
;    62          0062  1 !               Fix ZERO NODE from a V2 node.
;    63          0063  1 !
;    64          0064  1 !       V02-003 MKP0003         Kathy Perko     31-Jan-1982
;    65          0065  1 !               Fix NICE message so the line parameter, Receive Buffers
;    66          0066  1 !               is returned as a word.
;    67          0067  1 !
;    68          0068  1 !       V02-002 MKP0002         Kathy Perko     4-Jan-1982
;    69          0069  1 !               Add SHOW LINKS to V2 compatibility.
;    70          0070  1 !
;    71          0071  1 !       V02-001 MKP0001         Kathy Perko     29-Nov-1981
;    72          0072  1 !               Add zero counters to V2 compatibility.  Also, fix
;    73          0073  1 !               SHOW LINE SUMMARY and STATUS to return "on-starting"
;    74          0074  1 !               instead of "synchronizing" for state.
;    75          0075  1 !
;    76          0076  1 !--
```

```
;    77          0077  1
;    78          0078  1 %SBTTL 'Declarations'
;    79          0079  1
;    80          0080  1 !
;    81          0081  1 ! TABLE OF CONTENTS:
;    82          0082  1 !
;    83          0083  1
;    84          0084  1 FORWARD ROUTINE
;    85          0085  1     nml$v2_compatibility,                   ! Process any V2 NICE messages
;    86          0086  1     nml$v2_show_line:   NOVALUE,            ! Dispatch a V2 SHOW LINE request
;    87          0087  1     nml_v2_dispatch:    NOVALUE,            ! Dispatch to show or set routine
;    88          0088  1     nml_v2_showknown:   NOVALUE,            ! Show known lines
;    89          0089  1     nml_v2_showactive:  NOVALUE,            ! Show active lines
;    90          0090  1     nml_v2_showline:    NOVALUE,            ! Show a single line
;    91          0091  1     nml$sho_v2line_substa,                  ! Put line substate into NICE message.
;    92          0092  1     nml$v2_show_links:  NOVALUE,            ! Dispatch to show known links.
;    93          0093  1     nml_v2_show_links:  NOVALUE,            ! Show known links [with node <id>]
;    94          0094  1     nml$sholinks,                           ! Format links for response message.
;    95          0095  1     nml$v2_chg_line:    NOVALUE,            ! Dispatch a V2 SET LINE request
;    96          0096  1     nml$chk_v2_circ,                        ! Check NICE command for circuit params.
;    97          0097  1     nml$chk_v2_line,                        ! Check NICE command for line params.
;    98          0098  1     nml$chk_v2_sta,                         ! Check NICE command state parameter.
;    99          0099  1     nml_v2_chg_line:    NOVALUE,            ! Process V2 SET LINE request.
;   100          0100  1     nml_v2_chg_entity,                      ! Update volatile entity
;   101          0101  1     nml_v2_chg_known:   NOVALUE;            ! Update known volatile lines.
;   102          0102  1
;   103          0103  1 !
;   104          0104  1 ! INCLUDE FILES:
;   105          0105  1 !
;   106          0106  1
;   107          0107  1 LIBRARY 'LIB$:NMLLIB';                      ! Facility-wide definitions
;   108          0108  1
;   109          0109  1 LIBRARY 'SHRLIB$:NMALIBRY';                 ! NICE definitions
;   110          0110  1
;   111          0111  1 LIBRARY 'SYS$LIBRARY:STARLET';              ! VMS common definitions
;   112          0112  1
;   113          0113  1 LIBRARY 'SHRLIB$:NET';                      ! NETACP NFB definitions
;   114          0114  1
;   115          0115  1 !
;   116          0116  1 ! EXTERNAL REFERENCES:
;   117          0117  1 !
;   118          0118  1
;   119          0119  1 EXTERNAL
;   120          0120  1     nml$gb_ncp_version: BYTE,               ! NCP NICE version number
;   121          0121  1     nml$ab_npa_blk: $NPA_BLKDEF,            ! NPARSE context block
;   122          0122  1     nml$npa_setv2line,                      ! NPARSE table for V2 SET LINE
;   123          0123  1                                             !       commands.
;   124          0124  1     nml$npa_clearv2line;                    ! NPARSE table for V2 CLEAR LINE
;   125          0125  1                                             !       commands.
;   126          0126  1
;   127          0127  1 $NML_EXTDEF;                                ! Define common external data
;   128          0128  1
;   129          0129  1 MAP
;   130          0130  1     nml$gb_options: BBLOCK;
;   131          0131  1
;   132          0132  1 EXTERNAL ROUTINE
;   133          0133  1     nml$bld_reply,
```

```
;  134     0134  1      nml$send,
;  135     0135  1      nml$mainhandler,
;  136     0136  1      nml$error_1,
;  137     0137  1      nml$error_2,
;  138     0138  1      nml$get_entity_ids,
;  139     0139  1      nml$showentity,
;  140     0140  1      nml$shoparam,
;  141     0141  1      nml$shonodeid,
;  142     0142  1      nml$shoexeparam,
;  143     0143  1      nml$bldp2,
;  144     0144  1      nml$getinftabs,
;  145     0145  1      nml$bldshowbufs,
;  146     0146  1      nml$getdata,
;  147     0147  1      nml$processdata,
;  148     0148  1      nml$addmsgprm,
;  149     0149  1      lib$establish,
;  150     0150  1      lib$revert,
;  151     0151  1      nma$nparse,
;  152     0152  1      nml$setknown,
;  153     0153  1      nml$setentity,
;  154     0154  1      nml$saveparam,
;  155     0155  1      nml$getexeadr,
;  156     0156  1      nml$getidstring,
;  157     0157  1      nml$showparlist,
;  158     0158  1      nml$bldsetqbf,
;  159     0159  1      nml$netqio;
;  160     0160  1
;  161     0161  1  EXTERNAL LITERAL
;  162     0162  1      cpt$gk_pcci_sta,
;  163     0163  1      cpt$gk_pcli_sta;
;  164     0164  1
;  165     0165  1  !
;  166     0166  1  ! The NICE parameter for receive buffers (NMA$C_PCLI_BFN) got changed
;  167     0167  1  ! from 2700 in V2 to 1105 in V3.  Because of this, declare a V2 parameter
;  168     0168  1  ! id here.
;  169     0169  1  !
;  170     0170  1  GLOBAL LITERAL
;  171     0171  1       nma$c_pcli_bf$ = 2700;
;  172     0172  1
;  173     0173  1  !
;  174     0174  1  ! Own storage
;  175     0175  1  !
;  176     0176  1  OWN
;  177     0177  1      nml$l_v2_entity:                        ! Current entity (line or circuit)
;  178     0178  1                      INITIAL (nml$c_line),
;  179     0179  1      nml$l_state,                            ! New state for a line
;  180     0180  1                                              !       and circuit.
;  181     0181  1  !
;  182     0182  1  ! Buffers and descriptors.
;  183     0183  1  !
;  184     0184  1      NML$T_NFBBUFFER  : VECTOR [100, BYTE], ! NFB QIO buffer
;  185     0185  1      NML$T_P2BUFFER   : VECTOR [NML$K_P2BUFLEN,  BYTE], ! P2 QIO buffer
;  186     0186  1      NML$T_ENTBUFFER  : VECTOR [NML$K_ENTBUFLEN, BYTE]; ! Entity buffer
;  187     0187  1
;  188     0188  1  BIND
;  189     0189  1      NML$Q_NFBBFDSC   = UPLIT (%ALLOCATION(NML$T_NFBBUFFER),  NML$T_NFBBUFFER)
;  190     0190  1                       : DESCRIPTOR,
```

```
;  191          0191  1     NML$Q_P2BFDSC   = UPLIT (%ALLOCATION(NML$T_P2BUFFER),  NML$T_P2BUFFER)
;  192          0192  1                       : DESCRIPTOR;
;  193          0193  1 OWN
;  194          0194  1     NML$Q_ENTBFDSC : DESCRIPTOR
;  195          0195  1                      INITIAL (0, NML$T_ENTBUFFER);
;  196          0196  1
;  197          0197  1 !
;  198          0198  1 ! The data which uses the following macros would normally be put into
;  199          0199  1 ! NMLDAT, but, since this module will eventually be thrown away, they
;  200          0200  1 ! are here to make it easier to throw it out.  The macros are patterned
;  201          0201  1 ! after the ones in NMLDAT.
;  202          0202  1 !
;  203          0203  1 MACRO
;  204        M 0204  1     PRM_LIST (TAB, TYP) [] =
;  205        M 0205  1         BIND
;  206        M 0206  1             %NAME ('NML$Q_',TAB, TYP, '_TABDSC') =
;  207        M 0207  1                 UPLIT ((%LENGTH - 2) / 3,
;  208        M 0208  1                         UPLIT BYTE ($DEXTN (%REMAINING)));
;  209          0209  1     %,
;  210          0210  1
;  211        M 0211  1     $DEXTN [A, B, C] =
;  212        M 0212  1         WORD (%NAME ('PST$K_', A, '_', B)),
;  213        M 0213  1         LONG (C)
;  214          0214  1     %,
;  215          0215  1
;  216        M 0216  1     EXT_LIST [TYP, ID, RTN] =
;  217        M 0217  1         EXTERNAL LITERAL
;  218        M 0218  1             %NAME ('PST$K_', TYP, '_', ID);
;  219          0219  1     %;
```

```
;  221          0220  1 %SBTTL 'NML$V2_COMPATIBILITY    Process V2.0 NICE messages'
;  222          0221  1 GLOBAL ROUTINE NML$V2_COMPATIBILITY =
;  223          0222  1
;  224          0223  1 !---
;  225          0224  1 !
;  226          0225  1 !       This routine is called to look at an incoming NICE message
;  227          0226  1 !       and if the message is coming from an NCP speaking V2.0 NICE,
;  228          0227  1 !       then the message will be appropriately parsed and mapped to
;  229          0228  1 !       the V3.0 network management parameters.
;  230          0229  1 !
;  231          0230  1 ! Inputs:
;  232          0231  1 !
;  233          0232  1 !       nml$ab_rcvbuffer = Buffer containing NICE message
;  234          0233  1 !       nml$gl_rcvdatlen = Length of NICE message
;  235          0234  1 !
;  236          0235  1 ! Outputs:
;  237          0236  1 !
;  238          0237  1 !       Routine = True if message handled here, else false.
;  239          0238  1 !---
;  240          0239  1
;  241          0240  2 BEGIN
;  242          0241  2
;  243          0242  2 BUILTIN FP;
;  244          0243  2
;  245          0244  2 .fp = nml$mainhandler;                  ! Indicate that all signals should
;  246          0245  2                                         ! return to this level (with R0=0)
;  247          0246  2
;  248          0247  2 !
;  249          0248  2 ! If the NCP on the other side of the link is not speaking V2.0, then
;  250          0249  2 ! exit immediately and let the rest of NML handle it.
;  251          0250  2 !
;  252          0251  2
;  253          0252  2 IF .nml$gb_ncp_version NEQ 2             ! If NCP not V2.0,
;  254          0253  2 THEN
;  255          0254  2     RETURN false;                       ! Have caller handle message
;  256          0255  2
;  257          0256  2 SELECTONEU .nml$gb_function             ! Dispatch on function code
;  258          0257  2 OF
;  259          0258  2     SET
;  260          0259  2
;  261          0260  2 !
;  262          0261  2 ! For SHOW LINE, depending on the information type requested, we must
;  263          0262  2 ! either convert the entity to a circuit, or issue QIOs to both the
;  264          0263  2 ! line and circuit database to collect the information and then collate
;  265          0264  2 ! the parameters back into a single response message.
;  266          0265  2 !
;  267          0266  2
;  268          0267  2     [nma$c_fnc_rea]:
;  269          0268  2
;  270          0269  3         BEGIN
;  271          0270  3         IF NOT .nml$gl_prs_flgs [nml$v_prs_vms] THEN
;  272          0271  4             BEGIN
;  273          0272  4             IF .nml$gb_entity_code EQL nma$c_ent_lin    ! If SHOW LINE
;  274          0273  4             THEN
;  275          0274  5                 BEGIN
;  276          0275  5                 nml$v2_show_line();     ! then call processing routine
;  277          0276  5                 RETURN true;            ! and indicate nothing left to do
```

```
4                       END;
4                   END
3               ELSE
4                   BEGIN
4                   IF .nml$gb_entity_code EQL nma$c_sent_lnk THEN      ! If SHOW LINKS
5                       BEGIN
5                       nml$v2_show_links ();   ! then call processing routine
5                       RETURN true;            ! and indicate nothing left to do.
4                       END;
3                   END;
2               END;
2
2   !
2   ! For SET LINE, we do not allow mixed parameters in the same message.  That
2   ! is, we do not allow V2 parameters which map to both V3 lines and circuits
2   ! in the same request.  This avoids having to issue QIOs to both databases
2   ! in some cases, and allows us to simply change the entity and use the normal
2   ! SET processing.
2   !
2
2       [nma$c_fnc_cha]:
2
2           IF NOT .nml$gl_prs_flgs [nml$v_prs_vms] AND
3               (.nml$gb_entity_code EQL nma$c_ent_lin)     ! If SET LINE
2           THEN
3               BEGIN
3               nml$v2_chg_line();              ! then call processing routine
3               RETURN true;                    ! and indicate nothing left to do
2               END;
2
2   !
2   ! For ZERO LINE counters, change the entity ID from LINE to CIRCUIT (V2 LINE
2   ! counters are now V3 CIRCUIT counters), and then return to the normal
2   ! path to perform the zero.
2   !
2       [nma$c_fnc_zer]:
2           IF .nml$gb_entity_code EQL nma$c_ent_lin THEN
2               nml$gb_entity_code = nma$c_ent_cir;
2
2   !
2   ! For LOAD/DUMP/TRIGGER/LOOP, NPARSE initialization has not yet processed
2   ! the entity ID - only the option byte.  So, if LINE is indicated by the
2   ! low bit of the option byte, then change the entity type field (low 3 bits)
2   ! to CIRCUIT.  Else, NODE is indicated, so leave the entity type field zero.
2   ! Either way, return to the normal path to actually perform the operation.
2   !
2
2       [nma$c_fnc_loa,                         ! For LOAD/DUMP/TRIGGER/LOOP
2        nma$c_fnc_dum,
2        nma$c_fnc_tri,
2        nma$c_fnc_tes]:
2
3           BEGIN
3           IF .nml$gb_options <0,1>        ! If low bit (line/node) set,
3           THEN
3               nml$gb_options [nma$v_opt_ent] = nma$c_ent_cir  ! Mark CIRCUIT
3           ELSE
```

```
;  335          0334  3              nml$gb_options [nma$v_opt_ent] = nma$c_ent_nod; ! Else, mark NODE
;  336          0335  3          CH$WCHAR(.nml$gb_options, .nml[$ab_npa_blk [npa$l_fldptr]);
;  337          0336  2          END;
;  338          0337  2
;  339          0338  2      TES;
;  340          0339  2
;  341          0340  2 RETURN false;                       ! Indicate that caller must handle it
;  342          0341  2
;  343          0342  1 END;


                                                        .TITLE  NML$V2COMP Process NICE V2.0 requests
                                                        .IDENT  \V04-000\

                                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

                                        00000064  00000 P.AAA:  .LONG   100                                             ;
                                        00000000' 00004         .ADDRESS NML$T_NFBBUFFER                                ;
                                        00000068  00008 P.AAB:  .LONG   104                                             ;
                                        00000000' 0000C         .ADDRESS NML$T_P2BUFFER                                 ;

                                                        .PSECT  $OWN$,NOEXE,2

                                        00000000  00000 NML$L_V2_ENTITY:
                                                        .LONG   0                                                       ;
                                                  00004 NML$L_STATE:
                                                        .BLKB   4
                                                  00008 NML$T_NFBBUFFER:
                                                        .BLKB   100
                                                  0006C NML$T_P2BUFFER:
                                                        .BLKB   104
                                                  000D4 NML$T_ENTBUFFER:
                                                        .BLKB   64
                                        00000000  00114 NML$Q_ENTBFDSC:
                                                        .LONG   0                                                       ;
                                        00000000' 00118         .ADDRESS NML$T_ENTBUFFER                                ;

                                                NMA$C_PCLI_BF$==    2700
                                                NML$Q_NFBBFDSC=     P.AAA
                                                NML$Q_P2BFDSC=      P.AAB
                                                        .EXTRN  NML$GB_NCP_VERSION
                                                        .EXTRN  NML$AB_NPA_BLK, NML$NPA_SETV2LINE
                                                        .EXTRN  NML$NPA_CLEARV2LINE
                                                        .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$BLD_REPLY, NML$SEND
                                .EXTRN  NML$MAINHANDLER
                                .EXTRN  NML$ERROR_1, NML$ERROR_2
                                .EXTRN  NML$GET_ENTITY_IDS
                                .EXTRN  NML$SHOWENTITY, NML$SHOPARAM
                                .EXTRN  NML$SHONODEID, NML$SHOEXEPARAM
                                .EXTRN  NML$BLDP2, NML$GETINFTABS
                                .EXTRN  NML$BLDSHOWBUFS
                                .EXTRN  NML$GETDATA, NML$PROCESSDATA
                                .EXTRN  NML$ADDMSGPRM, LIB$ESTABLISH
                                .EXTRN  LIB$REVERT, NMA$NPARSE
                                .EXTRN  NML$SETKNOWN, NML$SETENTITY
                                .EXTRN  NML$SAVEPARAM, NML$GETEXEADR
                                .EXTRN  NML$GETIDSTRING
                                .EXTRN  NML$SHOWPARLIST
                                .EXTRN  NML$BLDSETQBF, NML$NETQIO
                                .EXTRN  CPT$GK_PCCI_STA
                                .EXTRN  CPT$GK_PCLI_STA

                                .PSECT  $CODE$,NOWRT,2

                    001C 00000  .ENTRY  NML$V2_COMPATIBILITY, Save R2,R3,R4          ; 0221
54 00000000G    00   9E 00002  MOVAB    NML$GL_PRS_FLGS, R4                         ;
53 00000000G    00   9E 00009  MOVAB    NML$GB_OPTIONS, R3                          ;
52 00000000G    00   9E 00010  MOVAB    NML$GB_ENTITY_CODE, R2                      ;
6D 00000000G    00   9E 00017  MOVAB    NML$MAINHANDLER, (FP)                       ; 0244
02 00000000G    00   91 0001E  CMPB     NML$GB_NCP_VERSION, #2                      ; 0252
                76   12 00025  BNEQ     8$                                          ;
50 00000000G    00   9A 00027  MOVZBL   NML$GB_FUNCTION, R0                         ; 0256
14              50   91 0002E  CMPB     R0, #20                                     ; 0267
                22   12 00031  BNEQ     2$                                          ;
51              62   9A 00033  MOVZBL   NML$GB_ENTITY_CODE, R1                     ; 0272
0E              64   E8 00036  BLBS     NML$GL_PRS_FLGS, 1$                         ; 0270
```

```
                                   01      51 D1 00039       CMPL    R1, #1                              ; 0272
                                           5F 12 0003C       BNEQ    8$                                  ;
                      00000000V    00      00 FB 0003E       CALLS   #0, NML$V2_SHOW_LINE                ; 0275
                                           22 11 00045       BRB     3$                                  ; 0276
                                   07      51 91 00047 1$:   CMPB    R1, #7                              ; 0281
                                           51 12 0004A       BNEQ    8$                                  ;
                      00000000V    00      00 FB 0004C       CALLS   #0, NML$V2_SHOW_LINKS               ; 0283
                                           14 11 00053       BRB     3$                                  ; 0284
                                   13      50 91 00055 2$:   CMPB    R0, #19                             ; 0297
                                           13 12 00058       BNEQ    4$                                  ;
                                   40      64 E8 0005A       BLBS    NML$GL_PRS_FLGS, 8$                 ; 0299
                                   01      62 91 0005D       CMPB    NML$GB_ENTITY_CODE, #1              ; 0300
                                           3B 12 00060       BNEQ    8$                                  ;
                      00000000V    00      00 FB 00062       CALLS   #0, NML$V2_CHG_LINE                 ; 0303
                                   50      01 D0 00069 3$:   MOVL    #1, R0                              ; 0304
                                              04 0006C       RET                                         ;
                                   15      50 91 0006D 4$:   CMPB    R0, #21                             ; 0312
                                           0A 12 00070       BNEQ    5$                                  ;
                                   01      62 91 00072       CMPB    NML$GB_ENTITY_CODE, #1              ; 0313
                                           26 12 00075       BNEQ    8$                                  ;
                                   62      03 90 00077       MOVB    #3, NML$GB_ENTITY_CODE              ; 0314
                                           21 11 0007A       BRB     8$                                  ; 0313
                                   0F      50 91 0007C 5$:   CMPB    R0, #15                             ; 0324
                                           1C 1F 0007F       BLSSU   8$                                  ;
                                   12      50 91 00081       CMPB    R0, #18                             ;
                                           17 1A 00084       BGTRU   8$                                  ;
                                   07      63 E9 00086       BLBC    NML$GB_OPTIONS, 6$                  ; 0330
        63             03          00      03 F0 00089       INSV    #3, #0, #3, NML$GB_OPTIONS          ; 0332
                                           03 11 0008E       BRB     7$                                  ;
                                   63      07 8A 00090 6$:   BICB2   #7, NML$GB_OPTIONS                  ; 0334
                                   50 00000000G 00 D0 00093 7$: MOVL NML$AB_NPA_BLK+20, R0               ; 0335
                                   60      63 90 0009A       MOVB    NML$GB_OPTIONS, (R0)                ;
                                           50 D4 0009D 8$:   CLRL    R0                                  ; 0342
                                              04 0009F       RET                                         ;

; Routine Size:  160 bytes,    Routine Base:  $CODE$ + 0000
```

```
;  345          0343  1 %SBTTL 'NML$V2_SHOW_LINE  V2 compatibility read line routine'
;  346          0344  1 ROUTINE NML$V2_SHOW_LINE : NOVALUE =
;  347          0345  1
;  348          0346  1 !++
;  349          0347  1 ! FUNCTIONAL DESCRIPTION:
;  350          0348  1 !
;  351          0349  1 ! FORMAL PARAMETERS:
;  352          0350  1 !
;  353          0351  1 ! IMPLICIT INPUTS:
;  354          0352  1 !
;  355          0353  1 !       NML$GB_INFO contains the information type.
;  356          0354  1 !
;  357          0355  1 !--
;  358          0356  1
;  359          0357  2 BEGIN
;  360          0358  2
;  361          0359  2 LOCAL
;  362          0360  2     INDEX;                              ! Index into list descriptor table
;  363          0361  2
;  364          0362  2 MAP
;  365          0363  2     NML$GB_ENTITY_FORMAT : BYTE SIGNED;
;  366          0364  2
;  367          0365  2 !
;  368          0366  2 ! Information can be read only from volatile data bases.
;  369          0367  2 !
;  370          0368  2 IF NOT .NML$GB_OPTIONS [NMA$V_OPT_PER]  ! If volatile database requested,
;  371          0369  2 THEN
;  372          0370  3     BEGIN
;  373          0371  3     !
;  374          0372  3     ! Read volatile data base
;  375          0373  3     !
;  376          0374  3     INDEX =
;  377          0375  4         (SELECTONEU .NML$GB_INFO
;  378          0376  4         OF
;  379          0377  4             SET
;  380          0378  4             [NMA$C_OPINF_SUM]: NML$C_SUMMARY;
;  381          0379  4             [NMA$C_OPINF_STA]: NML$C_STATUS;
;  382          0380  4             [NMA$C_OPINF_CHA]: NML$C_CHARACTERISTICS;
;  383          0381  4             [NMA$C_OPINF_COU]: NML$C_COUNTERS;
;  384          0382  4             [OTHERWISE]: -1;    ! Option error
;  385          0383  3             TES);
;  386          0384  3
;  387          0385  3     IF .INDEX NEQU -1
;  388          0386  3     THEN
;  389          0387  4         BEGIN
;  390          0388  4         !
;  391          0389  4         ! Dispatch to the appropriate SHOW routine.  Note that V2 lines
;  392          0390  4         ! are considered circuits by V3.
;  393          0391  4         !
;  394          0392  4         SELECTONEU .NML$GB_ENTITY_FORMAT OF
;  395          0393  4             SET
;  396          0394  4
;  397          0395  4             [NMA$C_ENT_ACT]:            ! Active
;  398          0396  4                 NML_V2_DISPATCH (NML$C_CIRCUIT,
;  399          0397  4                                  NML_V2_SHOWACTIVE,     ! Routine
;  400          0398  4                                  .INDEX,                ! Info code
;  401          0399  4                                  0, 0);
```

```
0400  4
0401  4              [NMA$C_ENT_KNO]:                  ! Known
0402  4                  NML_V2_DISPATCH (NML$C_CIRCUIT,
0403  4                                   NML_V2_SHOWKNOWN,         ! Routine
0404  4                                   .INDEX,                   ! Info code
0405  4                                   0, 0);
0406  4
0407  4              [1 TO 16]:                        ! Line name
0408  4                  NML_V2_DISPATCH (NML$C_CIRCUIT,
0409  4                                   NML_V2_SHOWLINE,          ! Routine
0410  4                                   .INDEX,                   ! Info code
0411  4                                   .NML$GB_ENTITY_FORMAT,    ! Id string length
0412  4                                   NML$AB_ENTITY_ID);        ! Id string address
0413  4
0414  4              TES;
0415  4
0416  4          NML$ERROR_2 (NMA$C_STS_IDE,               ! Identification error
0417  4               NMA$C_ENT_LIN);
0418  3          END;
0419  2      END;
0420  2
0421  2 NML$ERROR_1 (NMA$C_STS_FUN);        ! Send option error message
0422  2
0423  1 END;                                ! End of NML$READ
```

```
                               000C 00000 NML$V2_SHOW_LINE:
                                                 .WORD     Save R2,R3                              ; 0344
             00000000G   00    95 00002          TSTB      NML$GB_OPTIONS                          ; 0368
                         03    18 00008          BGEQ      1$                                      ;
                       0090    31 0000A          BRW       11$                                     ;
          50 00000000G   00    9A 0000D 1$:      MOVZBL    NML$GB_INFO, R0                         ; 0375
                         04    12 00014          BNEQ      2$                                      ; 0378
                         53    D4 00016          CLRL      INDEX                                   ;
                         21    11 00018          BRB       6$                                      ;
          01             50    91 0001A 2$:      CMPB      R0, #1                                  ; 0379
                         05    12 0001D          BNEQ      3$                                      ;
          53             01    D0 0001F          MOVL      #1, INDEX                               ;
                         17    11 00022          BRB       6$                                      ;
          02             50    91 00024 3$:      CMPB      R0, #2                                  ; 0380
                         05    12 00027          BNEQ      4$                                      ;
          53             02    D0 00029          MOVL      #2, INDEX                               ;
                         0D    11 0002C          BRB       6$                                      ;
          03             50    91 0002E 4$:      CMPB      R0, #3                                  ; 0381
                         05    12 00031          BNEQ      5$                                      ;
          53             03    D0 00033          MOVL      #3, INDEX                               ;
                         03    11 00036          BRB       6$                                      ;
          53             01    CE 00038 5$:      MNEGL     #1, INDEX                               ; 0382
 FFFFFFFF 8F             53    D1 0003B 6$:      CMPL      INDEX, #-1                              ; 0385
                         59    13 00042          BEQL      11$                                     ;
          52 00000000G   00    98 00044          CVTBL     NML$GB_ENTITY_FORMAT, R2                ; 0392
       FE 8F             52    91 0004B          CMPB      R2, #-2                                 ; 0395
                         0C    12 0004F          BNEQ      7$                                      ;
                         7E    7C 00051          CLRQ      -(SP)                                   ; 0396
```

```
                                         53  DD 00053          PUSHL   INDEX                               ; 0398
                             00000000V   00  9F 00055          PUSHAB  NML_V2_SHOWACTIVE                   ; 0396
                                         2B  11 0005B          BRB     9$                                  ;
                     FF   8F             52  91 0005D 7$:      CMPB    R2, #-1                             ; 0401
                                         0C  12 00061          BNEQ    8$                                  ;
                                         7E  7C 00063          CLRQ    -(SP)                               ; 0402
                                         53  DD 00065          PUSHL   INDEX                               ; 0404
                             00000000V   00  9F 00067          PUSHAB  NML_V2_SHOWKNOWN                    ; 0402
                                         19  11 0006D          BRB     9$                                  ;
                                         52  D5 0006F 8$:      TSTL    R2                                  ; 0407
                                         1E  13 00071          BEQL    10$                                 ;
                          10             52  91 00073          CMPB    R2, #16                             ;
                                         19  1A 00076          BGTRU   10$                                 ;
                             00000000G   00  9F 00078          PUSHAB  NML$AB_ENTITY_ID                    ; 0408
                                         52  DD 0007E          PUSHL   R2                                  ; 0411
                                         53  DD 00080          PUSHL   INDEX                               ; 0410
                             00000000V   00  9F 00082          PUSHAB  NML_V2_SHOWLINE                     ; 0408
                                         09  DD 00088 9$:      PUSHL   #9                                  ;
                00000000V 00             05  FB 0008A          CALLS   #5, NML_V2_DISPATCH                 ;
                                         01  DD 00091 10$:     PUSHL   #1                                  ; 0416
                          7E             09  CE 00093          MNEGL   #9, -(SP)                           ;
                00000000G 00             02  FB 00096          CALLS   #2, NML$ERROR_2                     ;
                          7E             01  CE 0009D 11$:     MNEGL   #1, -(SP)                           ; 0421
                00000000G 00             01  FB 000A0          CALLS   #1, NML$ERROR_1                     ;
                                             04 000A7          RET                                         ; 0423

; Routine Size:  168 bytes,    Routine Base:  $CODE$ + 00A0
```

```
1 %SBTTL 'NML_V2_DISPATCH  Dispatch to V2 show or set routine'
1 ROUTINE NML_V2_DISPATCH (ENT, RTN, INF, PRM1, PRM2, PRM3) : NOVALUE =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine is called when processing a show or set command
1 !       from a V2 system.  It dispatches to the appropriate V2 show or
1 !       set routine.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       ENT             Entity type code.
1 !       RTN             Address of entity routine to be called.
1 !       INF             Information identity code (index).
1 !       PRM1            Routine parameter value.
1 !       PRM2            Routine parameter value.
1 !       PRM3            Routine parameter value.
1 !
1 !--
1
2 BEGIN
2
2 LOCAL
2     MSG_SIZE;
2
2 !
2 ! Send success with multiple responses message.
2 !
2 NML$BLD_REPLY (UPLIT(0, NMA$C_STS_MOR), MSG_SIZE);
2 NML$SEND (NML$AB_SNDBUFFER, .MSG_SIZE);
2 !
2 ! Enable condition handler to allow done message to be sent.
2 !
2 LIB$ESTABLISH (NML$MAINHANDLER);
2 !
2 ! Call entity-specific routine.
2 !
2 (.RTN) (.ENT, .INF, .PRM1, .PRM2, .PRM3);
2 !
2 ! Signal done message.
2 !
2 LIB$REVERT ();                        ! Disable condition handler
2 NML$ERROR_1 (NMA$C_STS_DON);          ! Signal no more responses
2
1 END;                                  ! End of NML_V2_DISPATCH


                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

                00000002  00000000  00010 P.AAC:  .LONG   0, 2                    ;


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

```
                                        0000 00000 NML_V2_DISPATCH:
                                                        .WORD   Save nothing                        ; 0425
                5E                 04   C2 00002        SUBL2   #4, SP
                                   5E   DD 00005        PUSHL   SP                                  ; 0453
                       00000000'   00   9F 00007        PUSHAB  P.AAC
     00000000G  00                 02   FB 0000D        CALLS   #2, NML$BLD_REPLY
                                   6E   DD 00014        PUSHL   MSG_SIZE                            ; 0454
                       00000000G   00   9F 00016        PUSHAB  NML$AB_SNDBUFFER
     00000000G  00                 02   FB 0001C        CALLS   #2, NML$SEND
                       00000000G   00   9F 00023        PUSHAB  NML$MAINHANDLER                     ; 0458
     00000000G  00                 01   FB 00029        CALLS   #1, LIB$ESTABLISH
                7E          14     AC   7D 00030        MOVQ    PRM2, -(SP)                         ; 0462
                7E          0C     AC   7D 00034        MOVQ    INF, -(SP)
                            04     AC   DD 00038        PUSHL   ENT
           08   BC                 05   FB 0003B        CALLS   #5, @RTN
     00000000G  00                 00   FB 0003F        CALLS   #0, LIB$REVERT                      ; 0466
                7E          80     8F   98 00046        CVTBL   #-128, -(SP)                        ; 0467
     00000000G  00                 01   FB 0004A        CALLS   #1, NML$ERROR_1
                                        04 00051        RET                                         ; 0469

; Routine Size:  82 bytes,    Routine Base:  $CODE$ + 0148
```

```
;  474     0470  1 %SBTTL 'NML_V2_SHOWKNOWN  Show known V2 line parameters'
;  475     0471  1 ROUTINE NML_V2_SHOWKNOWN (ENTITY, INF) : NOVALUE =
;  476     0472  1
;  477     0473  1 !++
;  478     0474  1 ! FUNCTIONAL DESCRIPTION:
;  479     0475  1 !       This routine reads the volatile data base entries for all
;  480     0476  1 !       lines.
;  481     0477  1 !
;  482     0478  1 ! FORMAL PARAMETERS:
;  483     0479  1 !       ENTITY          Entity type code.
;  484     0480  1 !       INF             Information type code.
;  485     0481  1 !
;  486     0482  1 !--
;  487     0483  1
;  488     0484  2 BEGIN
;  489     0485  2
;  490     0486  2 LOCAL
;  491     0487  2         BUFEND,
;  492     0488  2         LENGTH,
;  493     0489  2         LISDSC : DESCRIPTOR,
;  494     0490  2         PTR,
;  495     0491  2         STATUS,
;  496     0492  2         STRTFLG;
;  497     0493  2
;  498     0494  2 STRTFLG = FALSE;
;  499     0495  2
;  500     0496  2 WHILE NML$GET_ENTITY_IDS (.ENTITY, NMA$C_ENT_KNO, 0, .STRTFLG, LISDSC) DO
;  501     0497  3     BEGIN
;  502     0498  3
;  503     0499  3     STRTFLG = TRUE;
;  504     0500  3
;  505     0501  3     BUFEND = .LISDSC [DSC$A_POINTER] + .LISDSC [DSC$W_LENGTH];
;  506     0502  3     PTR = .LISDSC [DSC$A_POINTER];
;  507     0503  3
;  508     0504  3     WHILE .PTR LSSA .BUFEND DO
;  509     0505  4         BEGIN
;  510     0506  4
;  511     0507  4         LENGTH = .(.PTR)<0,16>;
;  512     0508  4         PTR = .PTR + 2;
;  513     0509  4
;  514     0510  4         NML_V2_SHOWLINE (.ENTITY, .INF, .LENGTH, .PTR);
;  515     0511  4
;  516     0512  4         PTR = .PTR + .LENGTH;   ! Advance pointer
;  517     0513  4
;  518     0514  3         END;
;  519     0515  2     END;
;  520     0516  2
;  521     0517  1 END;                            ! End of NML_V2_SHOWKNOWN


                                003C 00000 NML_V2_SHOWKNOWN:
                                           .WORD   Save R2,R3,R4,R5                ; 0471
                             5E      08  C2 00002          SUBL2   #8, SP          ;
                                     53  D4 00005          CLRL    STRTFLG         ; 0494
```

```
                                    4008    8F  BB 00007 1$:      PUSHR   #^M<R3,SP>                    ; 0496
                                            7E  D4 0000B         CLRL    -(SP)                          ;
                                        7E  01  CE 0000D         MNEGL   #1, -(SP)                      ;
                                      04    AC  DD 00010         PUSHL   ENTITY                         ;
                          00000000G   00    05  FB 00013         CALLS   #5, NML$GET_ENTITY_IDS         ;
                                      2A        50  E9 0001A     BLBC    R0, 3$                         ;
                                      53        01  D0 0001D     MOVL    #1, STRTFLG                    ; 0499
                                      55        6E  3C 00020     MOVZWL  LISDSC, BUFEND                 ; 0501
                                      55    04  AE  C0 00023     ADDL2   LISDSC+4, BUFEND               ;
                                      52    04  AE  D0 00027     MOVL    LISDSC+4, PTR                  ; 0502
                                      55        52  D1 0002B 2$:  CMPL    PTR, BUFEND                    ; 0504
                                                D7  1E 0002E     BGEQU   1$                             ;
                                      54        82  3C 00030     MOVZWL  (PTR)+, LENGTH                 ; 0507
                                                52  DD 00033     PUSHL   PTR                            ; 0510
                                                54  DD 00035     PUSHL   LENGTH                         ;
                                        7E  04  AC  7D 00037     MOVQ    ENTITY, -(SP)                  ;
                          00000000V   00        04  FB 0003B     CALLS   #4, NML_V2_SHOWLINE            ;
                                      52        54  C0 00042     ADDL2   LENGTH, PTR                    ; 0512
                                                E4  11 00045     BRB     2$                             ; 0504
                                                    04 00047 3$: RET                                    ; 0517

; Routine Size:  72 bytes,    Routine Base:  $CODE$ + 019A
```

```
;  523          0518  1 %SBTTL 'NML_V2_SHOWACTIVE  Show active line parameters'
;  524          0519  1 ROUTINE NML_V2_SHOWACTIVE (ENTITY, INF) : NOVALUE =
;  525          0520  1
;  526          0521  1 !++
;  527          0522  1 ! FUNCTIONAL DESCRIPTION:
;  528          0523  1 !
;  529          0524  1 !       This routine reads the volatile data base entries for all
;  530          0525  1 !       lines.
;  531          0526  1 !
;  532          0527  1 ! FORMAL PARAMETERS:
;  533          0528  1 !
;  534          0529  1 !       ENTITY          Entity type code.
;  535          0530  1 !       INF             Information type code.
;  536          0531  1 !
;  537          0532  1 !--
;  538          0533  1
;  539          0534  2 BEGIN
;  540          0535  2
;  541          0536  2 LOCAL
;  542          0537  2         BUFEND,
;  543          0538  2         LENGTH,
;  544          0539  2         LISDSC : DESCRIPTOR,
;  545          0540  2         PTR,
;  546          0541  2         STATE : BYTE,
;  547          0542  2         STATUS,
;  548          0543  2         STRTFLG;
;  549          0544  2
;  550          0545  2 STRTFLG = FALSE;
;  551          0546  2
;  552          0547  2 WHILE NML$GET_ENTITY_IDS (.ENTITY, NMA$C_ENT_ACT, 0, .STRTFLG, LISDSC) DO
;  553          0548  3     BEGIN
;  554          0549  3
;  555          0550  3     STRTFLG = TRUE;
;  556          0551  3
;  557          0552  3     BUFEND = .LISDSC [DSC$A_POINTER]  + .LISDSC [DSC$W_LENGTH];
;  558          0553  3     PTR = .LISDSC [DSC$A_POINTER];
;  559          0554  3
;  560          0555  3     WHILE .PTR LSSA .BUFEND DO
;  561          0556  4         BEGIN
;  562          0557  4         !
;  563          0558  4         ! Get line or circuit state.
;  564          0559  4         !
;  565          0560  4         STATE = .(.PTR)<0,8>;
;  566          0561  4         PTR = .PTR + 4;
;  567          0562  4
;  568          0563  4         LENGTH = .(.PTR)<0,16>;
;  569          0564  4         PTR = .PTR + 2;
;  570          0565  4         !
;  571          0566  4         ! Process line or circuit.
;  572          0567  4         !
;  573          0568  4         IF .STATE NEQ NMA$C_STATE_OFF
;  574          0569  4         THEN
;  575          0570  4            NML_V2_SHOWLINE (.ENTITY, .INF, .LENGTH, .PTR);
;  576          0571  4
;  577          0572  4         PTR = .PTR + .LENGTH;    ! Advance pointer
;  578          0573  4
;  579          0574  3         END;
```

```
;  580          0575  2      END;
;  581          0576  2
;  582          0577  1 END;                            ! End of NML_V2_SHOWACTIVE


                                              007C 00000 NML_V2_SHOWACTIVE:
                                                                 .WORD   Save R2,R3,R4,R5,R6                  ; 0519
                  5E               08  C2 00002                  SUBL2   #8, SP                               ;
                  53               D4  00005                     CLRL    STRTFLG                              ; 0545
                        4008       8F  BB 00007 1$:              PUSHR   #^M<R3,SP>                           ; 0547
                                   7E  D4 0000B                  CLRL    -(SP)                                ;
                  7E               02  CE 0000D                  MNEGL   #2, -(SP)                            ;
                           04      AC  DD 00010                  PUSHL   ENTITY                               ;
       00000000G  00               05  FB 00013                  CALLS   #5, NML$GET_ENTITY_IDS               ;
                  35               50  E9 0001A                  BLBC    R0, 4$                               ;
                  53               01  D0 0001D                  MOVL    #1, STRTFLG                          ; 0550
                  56               6E  3C 00020                  MOVZWL  LISDSC, BUFEND                       ; 0552
                  56       04      AE  C0 00023                  ADDL2   LISDSC+4, BUFEND                     ;
                  52       04      AE  D0 00027                  MOVL    LISDSC+4, PTR                        ; 0553
                  56               52  D1 0002B 2$:              CMPL    PTR, BUFEND                          ; 0555
                                   D7  1E 0002E                  BGEQU   1$                                   ;
                  55               82  90 00030                  MOVB    (PTR)+, STATE                        ; 0560
                  52               03  C0 00033                  ADDL2   #3, PTR                              ; 0561
                  54               82  3C 00036                  MOVZWL  (PTR)+, LENGTH                       ; 0563
                  01               55  91 00039                  CMPB    STATE, #1                            ; 0568
                                   0F  13 0003C                  BEQL    3$                                   ;
                                   52  DD 0003E                  PUSHL   PTR                                  ; 0570
                                   54  DD 00040                  PUSHL   LENGTH                               ;
                  7E       04      AC  7D 00042                  MOVQ    ENTITY, -(SP)                        ;
       00000000V  00               04  FB 00046                  CALLS   #4, NML_V2_SHOWLINE                  ;
                  52               54  C0 0004D 3$:              ADDL2   LENGTH, PTR                          ; 0572
                                   D9  11 00050                  BRB     2$                                   ; 0555
                                       04 00052 4$:              RET                                          ; 0577

; Routine Size:  83 bytes,    Routine Base:  $CODE$ + 01E2
```

```
;   584          0578  1 %SBTTL 'NML_V2_SHOWLINE  Show V2 line parameters'
;   585          0579  1 ROUTINE NML_V2_SHOWLINE (ENTITY, INF, LEN, ADR) : NOVALUE =
;   586          0580  1
;   587          0581  1 !++
;   588          0582  1 ! FUNCTIONAL DESCRIPTION:
;   589          0583  1 !
;   590          0584  1 !       This routine reads the volatile data base entries for all
;   591          0585  1 !       V2 lines - I.E. it gets the appropriate LINE and CIRCUIT
;   592          0586  1 !       parameters from the V3 NETACP to do a show for a V2 NCP.
;   593          0587  1 !       The reason the routine is as messy as it is, is so that
;   594          0588  1 !       the V2-V3 compatibility code can be easily thrown away for V4.
;   595          0589  1 !
;   596          0590  1 ! FORMAL PARAMETERS:
;   597          0591  1 !
;   598          0592  1 !       ENTITY          Entity ID
;   599          0593  1 !       INF             Information type code.
;   600          0594  1 !       LEN             Length of entity id string.
;   601          0595  1 !       ADR             Address of entity id string.
;   602          0596  1 !
;   603          0597  1 !--
;   604          0598  1
;   605          0599  2 BEGIN
;   606          0600  2 !
;   607          0601  2 ! Data for SHOW LINE CHARACTERISTICS.
;   608          0602  2 !
;   609          0603  2 BIND
;   610          0604  2     NML$GQ_LINBFDSC = NML$GQ_EXEBFDSC: DESCRIPTOR,
;   611          0605  2     NML$GQ_LINDATDSC = NML$GQ_EXEDATDSC: DESCRIPTOR,
;   612          0606  2     NML$GL_LINDATPTR = NML$GL_EXEDATPTR;
;   613          0607  2
;   614          0608  2 BIND ROUTINE
;   615          0609  2     NML$SHOLINBYTE = NML$SHOEXEPARAM,
;   616          0610  2     NML$SHOLINWORD = NML$SHOEXEPARAM;
;   617          0611  2
;   618          0612  2 MACRO
;   619        M 0613  2     CHAR_PARAMS =
;   620        M 0614  2          PCCI, SER, NML$SHOPARAM          ! Line service
;   621        M 0615  2         ,PCCI, LCT, NML$SHOPARAM          ! Line line counter
;   622        M 0616  2         ,PCCI, BLO, NML$SHOPARAM          ! Block size
;   623        M 0617  2         ,PCCI, COS, NML$SHOPARAM          ! Cost
;   624        M 0618  2         ,PCLI, CON, NML$SHOLINBYTE        ! Controller
;   625        M 0619  2         ,PCLI, DUP, NML$SHOLINBYTE        ! Duplex
;   626        M 0620  2         ,PCLI, PRO, NML$SHOLINBYTE        ! Protocol (V2 Type)
;   627        M 0621  2         ,PCLI, STI, NML$SHOLINWORD        ! Service Timer
;   628        M 0622  2         ,PCLI, RTT, NML$SHOLINWORD        ! Retransmit Timer (V2 normal timer)
;   629        M 0623  2         ,PCCI, TRI, NML$SHOPARAM          ! Tributary
;   630        M 0624  2         ,PCLI, BF$, NML$SHOLINWORD        ! Receive buffers
;   631          0625  2         %;
;   632          0626  2
;   633          0627  2 EXT_LIST (CHAR_PARAMS);
;   634          0628  2 PRM_LIST (LIN, V2CHA, CHAR_PARAMS);
;   635          0629  2
;   636          0630  2 !
;   637          0631  2 ! NFB to get the V2 line parameters that are circuit parameters in V3.
;   638          0632  2 !
;   639        P 0633  2 $NFBDSC (NML$Q_CIRC_NFBDSC, SHOW, , CRI
;   640        P 0634  2         ,NAM,             ! Search key one = circuit name, oper1 = eql
```

```
;  641          P 0635  2                ,,                  ! Wildcard search key two, oper2 = eql
;  642          P 0636  2                ,NAM                ! Name
;  643          P 0637  2                ,SER                ! Service
;  644          P 0638  2                ,LCT                ! Counter timer
;  645          P 0639  2                ,BLO                ! Block size
;  646          P 0640  2                ,COS                ! Cost
;  647          P 0641  2                ,TRI                ! Tributary
;  648            0642  2                );
;  649            0643  2
;  650            0644  2 !
;  651            0645  2 ! NFB to get the V2 line parameters that are line parameters in V3.
;  652            0646  2 !
;  653          P 0647  2 $NFBDSC (NML$Q_LINE_NFBDSC, SHOW, , PLI
;  654          P 0648  2                ,NAM,               ! Search key one = circuit name, oper1 = eql
;  655          P 0649  2                ,,                  ! Wildcard search key two, oper2 = eql
;  656          P 0650  2                ,CON                ! Controller
;  657          P 0651  2                ,DUP                ! Duplex
;  658          P 0652  2                 PRO                ! Protocol (V2 Line type)
;  659          P 0653  2                ,STI                ! Service timer
;  660          P 0654  2                ,RTT                ! Retransmit timer (V2 Normal timer)
;  661          P 0655  2                ,BFN                ! Receive buffers
;  662            0656  2                );
;  663            0657  2
;  664            0658  2 !
;  665            0659  2 !    Circuit summary
;  666            0660  2 !
;  667            0661  2 MACRO
;  668          M 0662  2     SUMMARY_PARAMS =
;  669          M 0663  2          PCCI, STA, NML$SHOPARAM                 ! State
;  670          M 0664  2         ,PCCI, SUB, NML$SHO_V2LINE_SUBSTA        ! Substate
;  671          M 0665  2         ,PCCI, LOO, NML$SHOPARAM                 ! Loopback name
;  672          M 0666  2         ,PCCI, ADJ, NML$SHONODEID                ! Adjacent node
;  673            0667  2          %;
;  674            0668  2
;  675            0669  2 EXT_LIST (SUMMARY_PARAMS);
;  676            0670  2 PRM_LIST (LIN, V2SUM, SUMMARY_PARAMS);
;  677            0671  2
;  678            0672  2 !
;  679            0673  2 ! Data for SHOW LINE SUMMARY and STATUS.
;  680            0674  2 !
;  681            0675  2 MACRO
;  682            0676  2 !    Circuit status
;  683          M 0677  2     STATUS_PARAMS =
;  684          M 0678  2          PCCI, STA, NML$SHOPARAM                 ! State
;  685          M 0679  2         ,PCCI, SUB, NML$SHO_V2LINE_SUBSTA        ! Substate
;  686          M 0680  2         ,PCCI, LOO, NML$SHOPARAM                 ! Loopback name
;  687          M 0681  2         ,PCCI, ADJ, NML$SHONODEID                ! Adjacent node
;  688          M 0682  2         ,PCCI, BLO, NML$SHOPARAM                 ! Block size
;  689            0683  2          %;
;  690            0684  2
;  691            0685  2 PRM_LIST (LIN, V2STA, STATUS_PARAMS);
;  692            0686  2
```

```
;  694          0687  2 LOCAL
;  695          0688  2     DATDSC : DESCRIPTOR,                ! QIO data descriptor
;  696          0689  2     DATPTR,                             ! Pointer into P4 buffer
;  697          0690  2     TABDSC : REF DESCRIPTOR,            ! NICE parameter formatting descriptor
;  698          0691  2     DUMDSC : REF DESCRIPTOR,            ! Dummy descriptor
;  699          0692  2     MSGDSC : DESCRIPTOR,                ! Output message descriptor
;  700          0693  2     NFBDSC : REF DESCRIPTOR,            ! NFB descriptor
;  701          0694  2     P2DSC  : DESCRIPTOR,                ! P2 parameter descriptor
;  702          0695  2     PERIOD_PTR,
;  703          0696  2     LINE_LEN;                           ! Length of circuit's corresponding
;  704          0697  2                                         !       line ID.
;  705          0698  2
;  706          0699  2
;  707          0700  2 SELECTU .INF OF
;  708          0701  2     SET
;  709          0702  2     [NML$C_STATUS, NML$C_SUMMARY, NML$C_COUNTERS]:
;  710          0703  2         !
;  711          0704  2         ! For status, summary, and counters the show parameters for V3
;  712          0705  2         ! circuits are the ones required for show parameters for V2 lines.
;  713          0706  2         ! Formatting the SUBSTATE parameter, however, is different.
;  714          0707  2         !
;  715          0708  3         BEGIN
;  716          0709  3         !
;  717          0710  3         ! Get canned NFB to get parameters from NETACP and build P2 buffer
;  718          0711  3         ! to get parameters from specified circuit.
;  719          0712  3         !
;  720          0713  3         NML$GETINFTABS (NML$C_CIRCUIT, .INF, NFBDSC, TABDSC, 0);
;  721          0714  3         NML$BLDP2 (.LEN, .ADR, -1, 0, NML$Q_P2BFDSC, P2DSC);
;  722          0715  2         END;
;  723          0716  2
;  724          0717  2     [NML$C_CHARACTERISTICS]:
;  725          0718  2         !
;  726          0719  2         ! Some V2 line characteristics are V3 line parameters and some
;  727          0720  2         ! are V3 circuit parameters.  Issue QIOs to both volatile data
;  728          0721  2         ! databases to get them.
;  729          0722  2         !
;  730          0723  3         BEGIN
;  731          0724  3         !
;  732          0725  3         ! If the circuit is multipoint, convert the circuit ID to a line ID.
;  733          0726  3         ! (Circuit ID DMP-0.2 = line ID DMP-0).
;  734          0727  3         !
;  735          0728  3         PERIOD_PTR = CH$FIND_CH (.LEN, .ADR, %C'.');
;  736          0729  3         IF .PERIOD_PTR NEQ 0 THEN
;  737          0730  3             LINE_LEN = .PERIOD_PTR - .ADR
;  738          0731  3         ELSE
;  739          0732  3             LINE_LEN = .LEN;
;  740          0733  3         NML$BLDP2 (.LINE_LEN, .ADR, -1, 0, NML$Q_P2BFDSC, P2DSC);
;  741          0734  3         !
;  742          0735  3         ! Use canned NFB to get line parameters from NETACP.
;  743          0736  3         !
;  744          0737  3         IF NOT NML$GETDATA (NML$Q_LINE_NFBDSC, P2DSC,
;  745          0738  3                             NML$GQ_LINBFDSC, NML$GQ_LINDATDSC)
;  746          0739  3         THEN
;  747          0740  4             BEGIN
;  748          0741  4             NML$BLD_REPLY (NML$AB_MSGBLOCK, MSGDSC [DSC$W_LENGTH]);
;  749          0742  4             NML$SEND (NML$AB_SNDBUFFER, .MSGDSC [DSC$W_LENGTH]);
;  750          0743  4             RETURN
```

```
;   751     0744  3             END;
;   752     0745  3         !
;   753     0746  3         ! Set up pointer to buffer with line characteristics.  The buffer
;   754     0747  3         ! with the circuit characteristics is handled by DATPTR.
;   755     0748  3         !
;   756     0749  3         NML$GL_LINDATPTR = .NML$G_LINDATDSC [DSC$A_POINTER];
;   757     0750  3         NFBDSC = NML$Q_CIRC_NFBDSC;
;   758     0751  3         TABDSC = NML$Q_LINV2CHA_TABDSC;
;   759     0752  3         NML$BLDP2 (.LEN, .ADR, -1, 0, NML$Q_P2BFDSC, P2DSC);
;   760     0753  2         END;
;   761     0754  2     TES;
;   762     0755  2 !
;   763     0756  2 ! Use canned NFB to get circuit parameters from NETACP.
;   764     0757  2 !
;   765     0758  2 IF NML$GETDATA (.NFBDSC, P2DSC, NML$GQ_QIOBFDSC, DATDSC)
;   766     0759  2 THEN
;   767     0760  3     BEGIN
;   768     0761  4     TABDSC = (SELECTONEU .INF OF
;   769     0762  4         SET
;   770     0763  4         [NML$C_STATUS]:  NML$Q_LINV2STA_TABDSC;
;   771     0764  4         [NML$C_SUMMARY]: NML$Q_LINV2SUM_TABDSC;
;   772     0765  4         [OTHERWISE]: .TABDSC;
;   773     0766  3         TES);
;   774     0767  3     DATPTR = .DATDSC [DSC$A_POINTER];
;   775     0768  3     !
;   776     0769  3     ! Format the line and circuit parameters into a single NICE
;   777     0770  3     ! response message.  NML$Q_LINV2CHA_TABDSC causes the formatting
;   778     0771  3     ! routine to switch between the line and circuit buffer when
;   779     0772  3     ! necessary.
;   780     0773  3     !
;   781     0774  3     NML$PROCESSDATA (.ENTITY, .TABDSC, DATDSC, DATPTR, MSGDSC);
;   782     0775  3     END
;   783     0776  2 ELSE
;   784     0777  3     BEGIN
;   785     0778  3     NML$BLD_REPLY (NML$AB_MSGBLOCK, MSGDSC [DSC$W_LENGTH]);
;   786     0779  3     MSGDSC [DSC$A_POINTER] = NML$AB_SNDBUFFER;
;   787     0780  2     END;
;   788     0781  2 !
;   789     0782  2 ! Send NICE response message to NCP.
;   790     0783  2 !
;   791     0784  2 NML$SEND (.MSGDSC [DSC$A_POINTER], .MSGDSC [DSC$W_LENGTH]);
;   792     0785  1 END;    ! of   NML_V2_SHOWLINE


                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

                                    0000G 00018 P.AAE:  .WORD   PST$K_PCCI_SER                  ;
                                000000000G 0001A        .ADDRESS NML$SHOPARAM                   ;
                                    0000G 0001E         .WORD   PST$K_PCCI_LCT                  ;
                                000000000G 00020        .ADDRESS NML$SHOPARAM                   ;
                                    0000G 00024         .WORD   PST$K_PCCI_BLO                  ;
                                000000000G 00026        .ADDRESS NML$SHOPARAM                   ;
                                    0000G 0002A         .WORD   PST$K_PCCI_COS                  ;
                                000000000G 0002C        .ADDRESS NML$SHOPARAM                   ;
                                    0000G 00030         .WORD   PST$K_PCLI_CON                  ;
                                000000000G 00032        .ADDRESS NML$SHOLINBYTE                 ;
```

```
         0000G  00036            .WORD    PST$K_PCLI_DUP                          ;
     00000000G  00038            .ADDRESS NML$SHOLINBYTE                          ;
         0000G  0003C            .WORD    PST$K_PCLI_PRO                          ;
     00000000G  0003E            .ADDRESS NML$SHOLINBYTE                          ;
         0000G  00042            .WORD    PST$K_PCLI_STI                          ;
     00000000G  00044            .ADDRESS NML$SHOLINWORD                          ;
         0000G  00048            .WORD    PST$K_PCLI_RTT                          ;
     00000000G  0004A            .ADDRESS NML$SHOLINWORD                          ;
         0000G  0004E            .WORD    PST$K_PCCI_TRI                          ;
     00000000G  00050            .ADDRESS NML$SHOPARAM                            ;
         0000G  00054            .WORD    PST$K_PCLI_BFS                          ;
     00000000G  00056            .ADDRESS NML$SHOLINWORD                          ;
                0005A            .BLKB    2
     0000000B   0005C  P.AAD:    .LONG    11                                      ;
     00000000'  00060            .ADDRESS P.AAE                                   ;
     00000030   00064  P.AAF:    .LONG    48                                      ;
     00000000'  00068            .ADDRESS U.1                                     ;
     00000030   0006C  P.AAG:    .LONG    48                                      ;
     00000000'  00070            .ADDRESS U.3                                     ;
         0000G  00074  P.AAI:    .WORD    PST$K_PCCI_STA                          ;
     00000000G  00076            .ADDRESS NML$SHOPARAM                            ;
         0000G  0007A            .WORD    PST$K_PCCI_SUB                          ;
     00000000V  0007C            .ADDRESS NML$SHO_V2LINE_SUBSTA                   ;
         0000G  00080            .WORD    PST$K_PCCI_LOO                          ;
     00000000G  00082            .ADDRESS NML$SHOPARAM                            ;
         0000G  00086            .WORD    PST$K_PCCI_ADJ                          ;
     00000000G  00088            .ADDRESS NML$SHONODEID                           ;
     00000004   0008C  P.AAH:    .LONG    4                                       ;
     00000000'  00090            .ADDRESS P.AAI                                   ;
         0000G  00094  P.AAK:    .WORD    PST$K_PCCI_STA                          ;
     00000000G  00096            .ADDRESS NML$SHOPARAM                            ;
         0000G  0009A            .WORD    PST$K_PCCI_SUB                          ;
     00000000V  0009C            .ADDRESS NML$SHO_V2LINE_SUBSTA                   ;
         0000G  000A0            .WORD    PST$K_PCCI_LOO                          ;
     00000000G  000A2            .ADDRESS NML$SHOPARAM                            ;
         0000G  000A6            .WORD    PST$K_PCCI_ADJ                          ;
     00000000G  000A8            .ADDRESS NML$SHONODEID                           ;
         0000G  000AC            .WORD    PST$K_PCCI_BLO                          ;
     00000000G  000AE            .ADDRESS NML$SHOPARAM                            ;
                000B2            .BLKB    2
     00000005   000B4  P.AAJ:    .LONG    5                                       ;
     00000000'  000B8            .ADDRESS P.AAK                                   ;

                                 .PSECT   $OWN$,NOEXE,2

           22   :  1C  ;_NFB
                       U.1:      .BYTE    34                                      ;
           00   0011D            .BYTE    0                                       ;
           04   0011E            .BYTE    4                                       ;
           00   0011F            .BYTE    0                                       ;
     04020041   00120            .LONG    67240001                                ;
     00000001   00124            .LONG    1                                       ;
           00   00128            .BYTE    0                                       ;
           00   00129            .BYTE    0                                       ;
         0000   0012A            .WORD    0                                       ;
     04020041   0012C            .LONG    67240001                                ;
     04000002   00130            .LONG    67108866                                ;
```

```
                04010015  00134          .LONG   67174421
                04010017  00138          .LONG   67174423
                04010018  0013C          .LONG   67174424
                04010024  00140          .LONG   67174436
                00000000  00144          .LONG   0
                          00148          .BLKB   4
                      22  0014C  :_NFB
                                 U.3:     .BYTE   34
                      00  0014D          .BYTE   0
                      05  0014E          .BYTE   5
                      00  0014F          .BYTE   0
                05020041  00150          .LONG   84017217
                00000001  00154          .LONG   1
                      00  00158          .BYTE   0
                      00  00159          .BYTE   0
                    0000  0015A          .WORD   0
                05000004  0015'          .LONG   83886084
                05000003  00160          .LONG   83886083
                05010014  00164          .LONG   83951636
                05010015  00168          .LONG   83951637
                05010021  0016C          .LONG   83951649
                0501001E  00170          .LONG   83951646
                00000000  00174          .LONG   0
                          00178          .BLKB   4

                                 NML$Q_LINV2CHA_TABDSC=
                                                 P.AAD
                                 U.2=            P.AAF
                                 U.4=            P.AAG
                                 NML$Q_LINV2SUM_TABDSC=
                                                 P.AAH
                                 NML$Q_LINV2STA_TABDSC=
                                                 P.AAJ
                                         .EXTRN  PST$K_PCCI_SER, PST$K_PCCI_LCT
                                         .EXTRN  PST$K_PCCI_BLO, PST$K_PCCI_COS
                                         .EXTRN  PST$K_PCLI_CON, PST$K_PCLI_DUP
                                         .EXTRN  PST$K_PCLI_PRO, PST$K_PCLI_STI
                                         .EXTRN  PST$K_PCLI_RTT, PST$K_PCCI_TRI
                                         .EXTRN  PST$K_PCLI_BFS, PST$K_PCCI_STA
                                         .EXTRN  PST$K_PCCI_SUB, PST$K_PCCI_LOO
                                         .EXTRN  PST$K_PCCI_ADJ

                                         .PSECT  $CODE$,NOWRT,2

                          01FC 00000 NML_V2_SHOWLINE:
                                         .WORD   Save R2,R3,R4,R5,R6,R7,R8        ; 0579
58 00000000G   00   9E 00002             MOVAB   NML$AB_SNDBUFFER, R8             ;
57 00000000G   00   9E 00009             MOVAB   NML$BLD_REPLY, R7                ;
56 00000000G   00   9E 00010             MOVAB   NML$AB_MSGBLOCK, R6              ;
55 00000000G   00   9E 00017             MOVAB   NML$GETDATA, R5                  ;
54 00000000G   00   9E 0001E             MOVAB   NML$BLDP2, R4                    ;
53 00000000'   00   9E 00025             MOVAB   NML$Q_P2BFDSC, R3                ;
5E             24   C2 0002C             SUBL2   #36, SP                          ;
52       08    AC   D0 0002F             MOVL    INF, R2                          ; 0700
01             52   D1 00033             CMPL    R2, #1                           ; 0702
               05   1B 00036             BLEQU   1$                               ;
03             52   D1 00038             CMPL    R2, #3                           ;
```

```
                              24  12 0003B         BNEQ    2$                                ;
                              7E  D4 0003D 1$:     CLRL    -(SP)                             ; 0713
                     04       AE  9F 0003F         PUSHAB  TABDSC                            ;
                     0C       AE  9F 00042         PUSHAB  NFBDSC                            ;
                              52  DD 00045         PUSHL   R2                                ;
                              09  DD 00047         PUSHL   #9                                ;
        00000000G  00         05  FB 00049         CALLS   #5, NMLSGETINFTABS                ;
                     0C       AE  9F 00050         PUSHAB  P2DSC                             ; 0714
                              53  DD 00053         PUSHL   R3                                ;
                              7E  D4 00055         CLRL    -(SP)                             ;
                7E            01  CE 00057         MNEGL   #1, -(SP)                         ;
                7E   0C       AC  7D 0005A         MOVQ    LEN, -(SP)                        ;
                64            06  FB 0005E         CALLS   #6, NMLSBLDP2                     ;
                02            52  D1 00061 2$:     CMPL    R2, #2                            ; 0717
                              79  12 00064         BNEQ    7$                                ;
   10   BC     0C  AC         2E  3A 00066         LOCC    #46, LEN, @ADR                    ; 0728
                              02  12 0006C         BNEQ    3$                                ;
                              51  D4 0006E         CLRL    R1                                ;
                              51  D5 00070 3$:     TSTL    PERIOD_PTR                        ; 0729
                              07  13 00072         BEQL    4$                                ;
        50     51    10       AC  C3 00074         SUBL3   ADR, PERIOD_PTR, LINE_LEN         ; 0730
                              04  11 00079         BRB     5$                                ;
                50   0C       AC  D0 0007B 4$:     MOVL    LEN, LINE_LEN                     ; 0732
                     0C       AE  9F 0007F 5$:     PUSHAB  P2DSC                             ; 0733
                              53  DD 00082         PUSHL   R3                                ;
                              7E  D4 00084         CLRL    -(SP)                             ;
                7E            01  CE 00086         MNEGL   #1, -(SP)                         ;
                     10       AC  DD 00089         PUSHL   ADR                               ;
                              50  DD 0008C         PUSHL   LINE_LEN                          ;
                64            06  FB 0008E         CALLS   #6, NMLSBLDP2                     ;
                  00000000G   00  9F 00091         PUSHAB  NMLSGQ_LINDATDSC                  ; 0737
                  00000000G   00  9F 00097         PUSHAB  NMLSGQ_LINBFDSC                   ;
                     14       AE  9F 0009D         PUSHAB  P2DSC                             ;
                     64       A3  9F 000A0         PUSHAB  NMLSQ_LINE_NFBDSC                 ;
                65            04  FB 000A3         CALLS   #4, NMLSGETDATA                   ;
                11            50  E8 000A6         BLBS    R0, 6$                            ;
                     14       AE  9F 000A9         PUSHAB  MSGDSC                            ; 0741
                              56  DD 000AC         PUSHL   R6                                ;
                67            02  FB 000AE         CALLS   #2, NMLSBLD_REPLY                 ;
                7E   14       AE  3C 000B1         MOVZWL  MSGDSC, -(SP)                     ; 0742
                              58  DD 000B5         PUSHL   R8                                ;
                            0087  31 000B7         BRW     13$                               ;
        00000000G  00 00000000G   00  D0 000BA 6$:     MOVL    NMLSGQ_LINDATDSC+4, NMLSGL_LINDATPTR  ; 0749
                04  AE   5C       A3  9E 000C5         MOVAB   NMLSQ_CIRC_NFBDSC, NFBDSC         ; 0750
                    6E   54       A3  9E 000CA         MOVAB   NMLSQ_LINV2CHA_TABDSC, TABDSC     ; 0751
                     0C       AE  9F 000CE         PUSHAB  P2DSC                             ; 0752
                              53  DD 000D1         PUSHL   R3                                ;
                              7E  D4 000D3         CLRL    -(SP)                             ;
                7E            01  CE 000D5         MNEGL   #1, -(SP)                         ;
                7E   0C       AC  7D 000D8         MOVQ    LEN, -(SP)                        ;
                64            06  FB 000DC         CALLS   #6, NMLSBLDP2                     ;
                     1C       AE  9F 000DF 7$:     PUSHAB  DATDSC                            ; 0758
                  00000000G   00  9F 000E2         PUSHAB  NMLSGQ_QIOBFDSC                   ;
                     14       AE  9F 000E8         PUSHAB  P2DSC                             ;
                     10       AE  DD 000EB         PUSHL   NFBDSC                            ;
                65            04  FB 000EE         CALLS   #4, NMLSGETDATA                   ;
                3A            50  E9 000F1         BLBC    R0, 11$                           ;
```

```
                                01              52  D1 000F4          CMPL    R2, #1                                          ; 0763
                                                07  12 000F7          BNEQ    8$                                              ;
                                50      00AC    C3  9E 000F9          MOVAB   NML$Q_LINV2STA_TABDSC, R0                       ;
                                                0E  11 000FE          BRB     10$                                             ;
                                                52  D5 00100  8$:     TSTL    R2                                              ; 0764
                                                07  12 00102          BNEQ    9$                                              ;
                                50      0084    C3  9E 00104          MOVAB   NML$Q_LINV2SUM_TABDSC, R0                       ;
                                                03  11 00109          BRB     10$                                             ;
                                50              6E  D0 0010B  9$:     MOVL    TABDSC, R0                                      ; 0765
                                6E              50  D0 0010E  10$:    MOVL    R0, TABDSC                                      ; 0761
                        08      AE      20      AE  D0 00111          MOVL    DATDSC+4, DATPTR                                ; 0767
                                        14      AE  9F 00116          PUSHAB  MSGDSC                                          ; 0774
                                        0C      AE  9F 00119          PUSHAB  DATPTR                                          ;
                                        24      AE  9F 0011C          PUSHAB  DATDSC                                          ;
                                        0C      AE  DD 0011F          PUSHL   TABDSC                                          ;
                                        04      AC  DD 00122          PUSHL   ENTITY                                          ;
                    00000000G   00              05  FB 00125          CALLS   #5, NML$PROCESSDATA                             ;
                                                0C  11 0012C          BRB     12$                                             ; 0758
                                        14      AE  9F 0012E  11$:    PUSHAB  MSGDSC                                          ; 0778
                                                56  DD 00131          PUSHL   R6                                              ;
                                67              02  FB 00133          CALLS   #2, NML$BLD_REPLY                               ;
                                18      AE      68  9E 00136          MOVAB   NML$AB_SNDBUFFER, MSGDSC+4                      ; 0779
                                7E              14  AE 3C 0013A  12$:    MOVZWL  MSGDSC, -(SP)                           ; 0784
                                        1C      AE  DD 0013E          PUSHL   MSGDSC+4                                        ;
                    00000000G   00              02  FB 00141  13$:    CALLS   #2, NML$SEND                                    ;
                                                    04 00148          RET                                                     ; 0785

; Routine Size:  329 bytes,    Routine Base:  $CODE$ + 0235
```

```
;  794          0786  1 %SBTTL 'NML$SHO_V2LINE_SUBSTA  Show V2 Line substate'
;  795          0787  1 GLOBAL ROUTINE NML$SHO_V2LINE_SUBSTA (SEM_LIST, BUFDSC, MSGSIZE,
;  796          0788  1                                       DATDSC, DATPTR)=
;  797          0789  1
;  798          0790  1 !++
;  799          0791  1 ! FUNCTIONAL DESCRIPTION:
;  800          0792  1 !       This routine is called when processing a SHOW LINE command from
;  801          0793  1 !       a remote NCP which is running Network Management V2.0.  It gets
;  802          0794  1 !       the circuit substate from the QIO buffer, and puts it into the NICE
;  803          0795  1 !       response message.
;  804          0796  1 !
;  805          0797  1 ! FORMAL PARAMETERS:
;  806          0798  1 !       SEM_LIST        Parameter semantic table entry address.
;  807          0799  1 !       BUFDSC          Output message buffer descriptor address.
;  808          0800  1 !       MSGSIZE         Address of current output message size.
;  809          0801  1 !       DATDSC          QIO buffer descriptor address.
;  810          0802  1 !       DATPTR          Current pointer into QIO data buffer.
;  811          0803  1 !
;  812          0804  1 ! ROUTINE VALUE:
;  813          0805  1 ! COMPLETION CODES:
;  814          0806  1 !
;  815          0807  1 !       Always returns success (NML$_STS_SUC).
;  816          0808  1 !
;  817          0809  1 !--
;  818          0810  1
;  819          0811  2 BEGIN
;  820          0812  2
;  821          0813  2 MAP
;  822          0814  2     SEM_LIST : REF BBLOCK;
;  823          0815  2
;  824          0816  2 IF .(..DATPTR)<0,32> NEQU -1
;  825          0817  2 THEN
;  826          0818  3     BEGIN
;  827          0819  3     !
;  828          0820  3     ! Change the "synchronizing" substate to "on-starting" so the V2
;  829          0821  3     ! NCP will print out something intelligible.
;  830          0822  3     !
;  831          0823  3     IF .(..DATPTR)<0,32> EQL NMA$C_LINSS_SYN THEN
;  832          0824  3         ..DATPTR = NMA$C_LINSS_STA;
;  833          0825  3     !
;  834          0826  3     ! Add the line substate to the NICE message.
;  835          0827  3     !
;  836          0828  3     NML$ADDMSGPRM (     .BUFDSC,
;  837          0829  3                         .MSGSIZE,
;  838          0830  3                         .SEM_LIST [PST$W_DATAID],
;  839          0831  3                         .SEM_LIST [PST$B_DATATYPE],
;  840          0832  3                         1,
;  841          0833  3                         ..DATPTR);
;  842          0834  3
;  843          0835  2     END;
;  844          0836  2 .DATPTR = ..DATPTR + 4;
;  845          0837  2 RETURN NML$_STS_SUC
;  846          0838  1 END;                            ! End of NML$SHO_V2LINE_SUBSTA
```

```
                                  0004 00000       .ENTRY  NML$SHO_V2LINE_SUBSTA, Save R2                    : 0787
               52        14   AC   D0 00002       MOVL    DATPTR, R2                                        : 0816
     FFFFFFFF  8F        00   B2   D1 00006       CMPL    @0(R2), #-1                                       :
                              23   13 0000E       BEQL    2$                                                :
               0A        00   B2   D1 00010       CMPL    @0(R2), #10                                       : 0823
                              03   12 00014       BNEQ    1$                                                :
                         00   B2   D4 00016       CLRL    @0(R2)                                            : 0824
                              62   DD 00019 1$:   PUSHL   (R2)                                              : 0833
                              01   DD 0001B       PUSHL   #1                                                : 0828
               50        04   AC   D0 0001D       MOVL    SEM_LIST, R0                                      : 0831
               7E        03   A0   9A 00021       MOVZBL  3(R0), -(SP)                                      :
               7E             60   3C 00025       MOVZWL  (R0), -(SP)                                       : 0830
               7E        08   AC   7D 00028       MOVQ    BUFDSC, -(SP)                                     : 0828
     00000000G 00             06   FB 0002C       CALLS   #6, NML$ADDMSGPRM                                 :
               62             04   C0 00033 2$:   ADDL2   #4, (R2)                                          : 0836
               50             01   D0 00036       MOVL    #1, R0                                            : 0837
                                   04 00039       RET                                                       : 0838

; Routine Size:  58 bytes,    Routine Base:  $CODE$ + 037E
```

```
1 %SBTTL 'NML$V2_SHOW_LINKS  Dispatch to show volatile LINK parameters'
1 ROUTINE NML$V2_SHOW_LINKS (INDEX) : NOVALUE =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine shows a summary of V2 LINK parameters from the volatile
1 !       data base.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       INDEX           Entity information table index code.
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NML$GB_ENTITY_FORMAT contains the entity format code.
1 !
1 !       If the NICE command is a request to SHOW KNOWN LINKS WITH NODE x:
1 !       NML$GW_QUALIFIER_CPT contains the address of the Change Parameter
1 !               Table entry for the node name or address.
1 !       NML$GB_QUALIFIER_FORMAT contains the node id length.
1 !       NML$AB_QUALIFIER_ID contains the node id.
1 !
1 !--
1
2 BEGIN
2
2 MAP
2     NML$GB_ENTITY_FORMAT : BYTE SIGNED;
2
2 !
2 ! All functions specifying the LINK entity must be system-specific.
2 !
2 SELECTONEU .NML$GB_ENTITY_FORMAT OF
2     SET
2     [NMA$C_ENT_KNO]:            ! Known, or known with node.
2         NML_V2_DISPATCH (NML$C_LINKS,
2                          NML_V2_SHOW_LINKS,      ! Routine address
2                          .NML$GL_QUALIFIER_PST,
2                          .NML$GB_QUALIFIER_FORMAT,
2                          NML$AB_QUALIFIER_ID);
2
2     TES;
2
2 NML$ERROR_2 (NMA$C_STS_IDE,              ! Identification error
2              NMA$C_SENT_LNK);
2
1 END;                            ! End of NML$V2_SHOW_LINKS


                                 0000 00000 NML$V2_SHOW_LINKS:
                                                  .WORD   Save nothing                          : 0840
              50 00000000G  00  98 00002          CVTBL   NML$GB_ENTITY_FORMAT, R0              : 0872
     FF    8F               50  91 00009          CMPB    R0, #-1                               : 0874
```

```
                                   20  12 0000D          BNEQ    1$                                     ;
                      00000000G   00  9F 0000F          PUSHAB  NML$AB_QUALIFIER_ID                    ; 0875
                   7E 00000000G   00  9A 00015          MOVZBL  NML$GB_QUALIFIER_FORMAT, -(SP)         ; 0878
                      00000000G   00  DD 0001C          PUSHL   NML$GL_QUALIFIER_PST                   ; 0877
                      00000000V   00  9F 00022          PUSHAB  NML_V2_SHOW_LINKS                      ; 0875
                                  18  DD 00028          PUSHL   #24                                    ;
             FD61   CF            05  FB 0002A          CALLS   #5, NML_V2_DISPATCH                    ;
                                  07  DD 0002F 1$:      PUSHL   #7                                     ; 0883
                    7E            09  CE 00031          MNEGL   #9, -(SP)                              ;
        00000000G   00            02  FB 00034          CALLS   #2, NML$ERROR_2                        ;
                                      04 0003B          RET                                            ; 0886

; Routine Size:  60 bytes,    Routine Base:  $CODE$ + 03B8


;  896         0887  1
```

```
;  898          0886  1 %SBTTL 'NML_V2_SHOW_LINKS  Show V2 volatile links parameters'
;  899          0889  1 ROUTINE NML_V2_SHOW_LINKS (ENTITY, QUAL_PST, QUAL_LEN, QUAL_ADR) : NOVALUE =
;  900          0890  1
;  901          0891  1 !++
;  902          0892  1 ! FUNCTIONAL DESCRIPTION:
;  903          0893  1 !
;  904          0894  1 !       This routine is called to perform SHOW LINK commands from nodes
;  905          0895  1 !       running V2 Network Management.  The parameters returned are
;  906          0896  1 !       different from those returned to a V2 node.
;  907          0897  1 !
;  908          0898  1 !       V2 nodes only accept the SHOW KNOWN LINKS and the SHOW KNOWN
;  909          0899  1 !       LINKS WITH NODE <node-id> commands.  The links are returned by
;  910          0900  1 !       node.  I.E.  One response message is sent to NCP for each remote
;  911          0901  1 !       node which there are current logical links to.  Each response
;  912          0902  1 !       message contains the node ID, followed by a list of link
;  913          0903  1 !       numbers and their PIDs.  For a V3 node, NML returns one link per
;  914          0904  1 !       response message along with its associated parameters.
;  915          0905  1 !
;  916          0906  1 !       For SHOW KNOWN LINKS command, build QIO buffers to get NETACP
;  917          0907  1 !       to return information about all known links on this node.
;  918          0908  1 !       For SHOW KNOWN LINKS WITH NODE <nodeid> command, build QIO
;  919          0909  1 !       buffers to return information about all links to the specified
;  920          0910  1 !       node from this node.
;  921          0911  1 !
;  922          0912  1 !       The QIO is repeated until all links of the specified type have
;  923          0913  1 !       been returned by the ACP.  As each link's information is received,
;  924          0914  1 !       it is formatted into a NICE message and returned to NCP.
;  925          0915  1 !
;  926          0916  1 ! FORMAL PARAMETERS:
;  927          0917  1 !
;  928          0918  1 !       ENTITY          Entity type code (always NML$C_LINKS)
;  929          0919  1 !       QUAL_PST        Address of node qualifier's entry in the Parameter
;  930          0920  1 !                       Semantic Table (PST).
;  931          0921  1 !       QUAL_LEN        Length of node qualifier ID string.
;  932          0922  1 !       QUAL_ADR        Address of node qualifier ID string.
;  933          0923  1 !
;  934          0924  1 !--
;  935          0925  1
;  936          0926  2 BEGIN
;  937          0927  2
;  938          0928  2 LOCAL
;  939          0929  2         P2DSC    : DESCRIPTOR,          ! P2 buffer descriptor.
;  940          0930  2         LAST_PNA,                       ! Last link's partner node address.
;  941          0931  2         LINK_CNT,                       ! Count of link entities returned by
;  942          0932  2                                         !     NETACP.
;  943          0933  2         STRDSC   : DESCRIPTOR,          ! Descriptor for node id for NICE
;  944          0934  2                                         !    response message.
;  945          0935  2         DATDSC   : DESCRIPTOR,          ! Return P4  buffer descriptor.
;  946          0936  2         DATPTR,                         ! P4 buffer pointer.
;  947          0937  2         LAD_BUF  : BBLOCK               ! Buffer for accumulating LADs in NICE
;  948          0938  2                    [NML$K_SNDBFLEN],    !     message format.
;  949          0939  2         LAD_BUF_DSC : DESCRIPTOR,       ! Descriptor for full size of LAD_BUF.
;  950          0940  2         LAD_DATA_DSC : DESCRIPTOR,      ! Descriptor for data in LAD_BUF.
;  951          0941  2         LAD_LEN,                        ! Longword for length of data in
;  952          0942  2                                         !      LAD_BUF (NML$SHOWPRMLIST needs a
;  953          0943  2                                         !      longword - I'm going to murder
;  954          0944  2                                         !      Davidson.)
```

```
;  955          0945  2          MSGSIZE,
;  956          0946  2          STATUS;
;  957          0947  2
;  958          0948  2
;  959          0949  2 !
;  960          0950  2 ! For formatting the link and its rID into the NICE response message
;  961          0951  2 !
;  962          0952  2 MACRO
;  963          0953  2     LINK_PARAMS = PCLK, LAD,    NML$SHOLINKS %;
;  964          0954  2
;  965          0955  2 EXT_LIST (LINK_PARAMS);
;  966          0956  2 PRM_LIST (LNK, V2SHO, LINK_PARAMS);
;  967          0957  2
;  968          0958  2 !
;  969          0959  2 ! This NFB is used get the link information for all the links to
;  970          0960  2 ! a given node.
;  971          0961  2 !
;  972        P 0962  2 $NFBDSC (NML_Q_V2_SHOLNK, SHOW, NFB$M_MULT OR NFB$M_ERRUPD, LLI
;  973        P 0963  2          ,NFB$C_WILDCARD,                 ! Search key one = wildcard, oper1 = eql
;  974        P 0964  2          ,NFB$C_WILDCARD,                 ! Search key two = wildcard, oper2 = eql
;  975        P 0965  2      !
;  976        P 0966  2      ! Link parameters for NETACP to return in P4 buffer.
;  977        P 0967  2      !
;  978        P 0968  2          ,PNA                             ! Partner node address
;  979        P 0969  2          ,PNN                             ! Partner node name
;  980        P 0970  2          ,LLN                             ! Logical link number
;  981        P 0971  2          ,PID                             ! Process ID
;  982          0972  2          );
;  983          0973  2
;  984          0974  2 MAP
;  985          0975  2     NML_Q_V2_SHOLNK : DESCRIPTOR;
;  986          0976  2 !
;  987          0977  2 ! Modify canned NFB descriptor to do the show links requested by the NICE
;  988          0978  2 ! command.  Use special NFBs that only get the information required for
;  989          0979  2 ! a V2 SHOW LINK: node name and address, link number, and PID.
;  990          0980  2 !
;  991          0981  2 NML$BLDSHOWBUFS (.ENTITY, NMA$C_ENT_KNO, 0,
;  992          0982  2                  .NML_Q_V2_SHOLNK [DSC$A_POINTER],      ! Address of NFB to fill in.
;  993          0983  2                  NML$Q_P2BFDSC,                         ! Buffer for P2.
;  994          0984  2                  P2DSC,                                 ! Return P2 descriptor.
;  995          0985  2                  .QUAL_PST,                             ! Node PST (if present)
;  996          0986  2                  .QUAL_LEN,                             ! Node ID length.
;  997          0987  2                  .QUAL_ADR);                            ! Node ID address.
;  998          0988  2 !
;  999          0989  2 ! Set up for loop to get link info from NETACP.
; 1000          0990  2 !
; 1001          0991  2 LAD_BUF_DSC [DSC$W_LENGTH] = NML$K_SNDBFLEN;
; 1002          0992  2 LAD_BUF_DSC [DSC$A_POINTER] = LAD_BUF;
; 1003          0993  2 LAD_DATA_DSC [DSC$A_POINTER] = LAD_BUF;
; 1004          0994  2 LAST_PNA = -1;
; 1005          0995  2 STATUS = 1;
; 1006          0996  2 LAD_LEN = 0;
; 1007          0997  2 !
; 1008          0998  2 ! NETACP will return all links to a given node consecutively.
; 1009          0999  2 ! This routine takes advantage of this fact.
; 1010          1000  2 !
; 1011          1001  2 WHILE .STATUS DO
```

```
;1012     1002  3     BEGIN
;1013     1003  3     STATUS = NML$GETDATA (NML_Q_V2_SHOLNK, P2DSC, NML$GQ_QIOBFDSC, DATDSC);
;1014     1004  3     IF .STATUS THEN
;1015     1005  4         BEGIN
;1016     1006  4         DATPTR = .DATDSC [DSC$A_POINTER];
;1017     1007  4         LINK_CNT = .(.P2DSC [DSC$A_POINTER]);
;1018     1008  4         WHILE (LINK_CNT = .LINK_CNT - 1) GEQ 0 DO
;1019     1009  5             BEGIN
;1020     1010  5             !
;1021     1011  5             ! If different node, and not first time build message and send it.
;1022     1012  5             !
;1023     1013  5             IF .LAST_PNA NEQ ..DATPTR THEN
;1024     1014  6                 BEGIN
;1025     1015  6                 IF .LAD_LEN NEQ 0 THEN
;1026     1016  7                     BEGIN
;1027     1017  7                     NML$AB_MSGBLOCK [MSB$L_FLAGS] = MSB$M_ENTD_FLD OR
;1028     1018  7                                                     MSB$M_DATA_FLD;
;1029     1019  7                     NML$AB_MSGBLOCK [MSB$B_CODE] = NMA$C_STS_SUC;
;1030     1020  7                     NML$AB_MSGBLOCK [MSB$A_ENTITY] = STRDSC;
;1031     1021  7                     LAD_DATA_DSC [DSC$W_LENGTH] = .LAD_LEN;
;1032     1022  7                     NML$AB_MSGBLOCK [MSB$A_DATA] = LAD_DATA_DSC;
;1033     1023  7                     NML$BLD_REPLY (NML$AB_MSGBLOCK, MSGSIZE);
;1034     1024  7                     NML$SEND (NML$AB_SNDBUFFER, .MSGSIZE);
;1035     1025  7                     !
;1036     1026  7                     ! Set up to build NICE message for next node in NETACPs
;1037     1027  7                     ! logical link database.
;1038     1028  7                     !
;1039     1029  7                     LAD_LEN = 0;
;1040     1030  7                     MSGSIZE = 0;
;1041     1031  6                     END;
;1042     1032  6                 !
;1043     1033  6                 ! Build string descriptor for node in STRDSC, and build
;1044     1034  6                 ! the node ID for the NICE response message.  This node ID
;1045     1035  6                 ! is in the standard Network Management format of node
;1046     1036  6                 ! address, node name length, node name.
;1047     1037  6                 !
;1048     1038  6                 LAST_PNA = ..DATPTR;
;1049     1039  6                 NML$GETIDSTRING (NML$C_NODE, DATPTR, STRDSC);
;1050     1040  6                 END
;1051     1041  5             ELSE
;1052     1042  5                 !
;1053     1043  5                 ! Skip over node address and name here.
;1054     1044  5                 !
;1055     1045  5                 DATPTR = .DATPTR + 6 + .(.DATPTR+4)<0,16>;
;1056     1046  5             !
;1057     1047  5             ! Format link # and PID into a buffer in NICE format.
;1058     1048  5             !
;1059     1049  5             NML$SHOWPARLIST (LAD_BUF_DSC,
;1060     1050  5                              LAD_LEN,
;1061     1051  5                              NML$Q_LNKV2SHO_TABDSC,
;1062     1052  5                              DATDSC,
;1063     1053  5                              DATPTR);
;1064     1054  4             END;
;1065     1055  3         END;
;1066     1056  2     END;
;1067     1057  2 !
;1068     1058  2 ! Build the last NICE response message.  If there was an error, but there is
```

```
;  1069          1059  2 ! a node id to add, do so.  If the last completion status was end-of-file
;  1070          1060  2 ! (NML$_STS_CMP) then the end of the link data base was successfully reached,
;  1071          1061  2 ! so add whatever links are left in the LAD buffer.
;  1072          1062  2 !
;  1073          1063  2 IF .LAD_LEN GTR 0 THEN
;  1074          1064  3     BEGIN
;  1075          1065  3     NML$AB_MSGBLOCK [MSB$L_FLAGS] = .NML$AB_MSGBLOCK [MSB$L_FLAGS] OR
;  1076          1066  3                 MSB$M_ENTD_FLD;
;  1077          1067  3     NML$AB_MSGBLOCK [MSB$A_ENTITY] = STRDSC;
;  1078          1068  3     IF .STATUS EQL NML$_STS_CMP THEN
;  1079          1069  4         BEGIN
;  1080          1070  4         NML$AB_MSGBLOCK [MSB$L_FLAGS] = MSB$M_ENTD_FLD OR MSB$M_DATA_FLD;
;  1081          1071  4         NML$AB_MSGBLOCK [MSB$B_CODE] = NMA$C_STS_SOC;
;  1082          1072  4         LAD_DATA_DSC [DSC$W_LENGTH] = .LAD_LEN;
;  1083          1073  4         NML$AB_MSGBLOCK [MSB$A_DATA] = LAD_DATA_DSC;
;  1084          1074  3         END;
;  1085          1075  2     END;
;  1086          1076  2 !
;  1087          1077  2 ! Put the pieces of the NICE response message together and send it
;  1088          1078  2 ! to NCP.
;  1089          1079  2 !
;  1090          1080  2 NML$BLD_REPLY (NML$AB_MSGBLOCK, MSGSIZE);
;  1091          1081  2 NML$SEND (NML$AB_SNDBUFFER,
;  1092          1082  2              .MSGSIZE);
;  1093          1083  1 END;                 ! of    NML_V2_SHOW_LINKS


                                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

    0000G 000BC P.AAM:  .WORD   PST$K_PCLK_LAD                                          ;
00000000V 000BE         .ADDRESS NML$SHOLINKS                                           ;
          000C2         .BLKB   2                                                       ;
00000001  000C4 P.AAL:  .LONG   1                                                       ;
00000000' 000C8         .ADDRESS P.AAM                                                  ;
00000028  000CC P.AAN:  .LONG   40                                                      ;
00000000' 000D0         .ADDRESS U.5                                                    ;

                                                        .PSECT  $OWN$,NOEXE,2

      22  0017C ;_NFB
                U.5:    .BYTE   34                                                      ;
      03  0017D         .BYTE   3                                                       ;
      08  0017E         .BYTE   8                                                       ;
      00  0017F         .BYTE   0                                                       ;
00000001  00180         .LONG   1                                                       ;
00000001  00184         .LONG   1                                                       ;
      00  00188         .BYTE   0                                                       ;
      00  00189         .BYTE   0                                                       ;
    0000  0018A         .WORD   0                                                       ;
08010014  0018C         .LONG   134283284                                               ;
08020043  00190         .LONG   134348867                                               ;
08010012  00194         .LONG   134283282                                               ;
08010015  00198         .LONG   134283285                                               ;
00000000  0019C         .LONG   0                                                       ;
          001A0         .BLKB   4                                                       ;
```

```
                                                NML$Q_LNKV2SHO_TABDSC=
                                                                     P.AAL
                                                U.6=                 P.AAN
                                                        .EXTRN  PST$K_PCLK_LAD

                                                        .PSECT  $CODE$,NOWRT,2

                                        07FC 00000 NML_V2_SHOW_LINKS:
                                                        .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10                ; 0889
                    5A 00000000G   00   9E 00002        MOVAB   NML$SEND, R10                                   ;
                    59 00000000G   00   9E 00009        MOVAB   NML$AB_SNDBUFFER, R9                            ;
                    58 00000000G   00   9E 00010        MOVAB   NML$BLD_REPLY, R8                               ;
                    57 00000000'   00   9E 00017        MOVAB   NML$Q_P2BFDSC, R7                               ;
                    56 00000000G   00   9E 0001E        MOVAB   NML$AB_MSGBLOCK, R6                             ;
                    5E      FDCC   CE   9E 00025        MOVAB   -564(SP), SP                                    ;
                    7E        0C   AC   7D 0002A        MOVQ    QUAL_LEN, -(SP)                                 ; 0986
                              08   AC   DD 0002E        PUSHL   QUAL_PST                                        ; 0985
                              F8   AD   9F 00031        PUSHAB  P2DSC                                           ; 0981
                                   57   DD 00034        PUSHL   R7                                              ;
                            00C8   C7   DD 00036        PUSHL   NML_Q_V2_SHOLNK+4                               ; 0982
                                   7E   D4 0003A        CLRL    -(SP)                                           ; 0981
                    7E             01   CE 0003C        MNEGL   #1, -(SP)                                       ;
                              04   AC   DD 0003F        PUSHL   ENTITY                                          ;
          00000000G   00           09   FB 00042        CALLS   #9, NML$BLDSHOWBUFS                             ;
                14    AE      0200   8F   B0 00049        MOVW    #512, LAD_BUF_DSC                               ; 0991
                18    AE        1C   AE   9E 0004F        MOVAB   LAD_BUF, LAD_BUF_DSC+4                          ; 0992
                10    AE        1C   AE   9E 00054        MOVAB   LAD_BUF, LAD_DATA_DSC+4                         ; 0993
                    55             01   CE 00059        MNEGL   #1, LAST_PNA                                    ; 0994
                    53             01   D0 0005C        MOVL    #1, STATUS                                      ; 0995
                              04   AE   D4 0005F        CLRL    LAD_LEN                                         ; 0996
                    1A             53   E9 00062 1$:     BLBC    STATUS, 2$                                      ; 1001
                              E8   AD   9F 00065        PUSHAB  DATDSC                                          ; 1003
                 00000000G    00   9F 00068        PUSHAB  NML$GQ_QIOBFDSC                                 ;
                              F8   AD   9F 0006E        PUSHAB  P2DSC                                           ;
                            00C4   C7   9F 00071        PUSHAB  NML_Q_V2_SHOLNK                                 ;
          00000000G   00           04   FB 00075        CALLS   #4, NML$GETDATA                                 ;
                    53             50   D0 0007C        MOVL    R0, STATUS                                      ;
                    77             53   E9 0007F 2$:     BLBC    STATUS, 7$                                      ; 1004
                    6E        EC   AD   D0 00082        MOVL    DATDSC+4, DATPTR                                ; 1006
                    54        FC   BD   D0 00086        MOVL    @P2DSC+4, LINK_CNT                              ; 1007
                                   54   D7 0008A 3$:     DECL    LINK_CNT                                        ; 1008
                                   D4   19 0008C        BLSS    1$                                              ;
                    52             6E   D0 0008E        MOVL    DATPTR, R2                                      ; 1013
                    62             55   D1 00091        CMPL    LAST_PNA, (R2)                                  ;
                                   42   13 00094        BEQL    5$                                              ;
                              04   AE   D5 00096        TSTL    LAD_LEN                                         ; 1015
                                   29   13 00099        BEQL    4$                                              ;
                    66             30   D0 0009B        MOVL    #48, NML$AB_MSGBLOCK                            ; 1017
                04  A6             01   90 0009E        MOVB    #1, NML$AB_MSGBLOCK+4                           ; 1019
                14  A6        F0   AD   9E 000A2        MOVAB   STRDSC, NML$AB_MSGBLOCK+20                      ; 1020
                0C  AE        04   AE   B0 000A7        MOVW    LAD_LEN, LAD_DATA_DSC                           ; 1021
                18  A6        0C   AE   9E 000AC        MOVAB   LAD_DATA_DSC, NML$AB_MSGBLOCK+24                ; 1022
                              08   AE   9F 000B1        PUSHAB  MSGSIZE                                         ; 1023
                                   56   DD 000B4        PUSHL   R6                                              ;
                    68             02   FB 000B6        CALLS   #2, NML$BLD_REPLY                               ;
                              08   AE   DD 000B9        PUSHL   MSGSIZE                                         ; 1024
                                   59   DD 000BC        PUSHL   R9                                              ;
```

```
                    6A           02  FB 000BE        CALLS   #2, NML$SEND                         ;
                         04      AE  7C 000C1        CLRQ    LAD_LEN                              ; 1029
                    55           62  D0 000C4 4$:    MOVL    (R2), LAST_PNA                       ; 1038
                         F0      AD  9F 000C7        PUSHAB  STRDSC                               ; 1039
                         04      AE  9F 000CA        PUSHAB  DATPTR                               ;
                                 03  DD 000CD        PUSHL   #3                                   ;
         00000000G  00           03  FB 000CF        CALLS   #3, NML$GETIDSTRING                  ;
                                 09  11 000D6        BRB     6$                                   ; 1013
                    50   04      A2  3C 000D8 5$:    MOVZWL  4(R2), R0                            ; 1045
                    6E   06 A240     9E 000DC        MOVAB   6(R2)[R0], DATPTR                    ;
                                 5E  DD 000E1 6$:    PUSHL   SP                                   ; 1049
                         E8      AD  9F 000E3        PUSHAB  DATDSC                               ;
                       00BC      C7  9F 000E6        PUSHAB  NML$Q_LNKV2SHO_TABDSC                ;
                         10      AE  9F 000EA        PUSHAB  LAD_LEN                              ;
                         24      AE  9F 000ED        PUSHAB  LAD_BUF_DSC                          ;
         00000000G  00           05  FB 000F0        CALLS   #5, NML$SHOWPARLIST                  ;
                                 91  11 000F7        BRB     3$                                   ; 1008
                         04      AE  D5 000F9 7$:    TSTL    LAD_LEN                              ; 1063
                                 22  15 000FC        BLEQ    8$                                   ;
                    66           10  88 000FE        BISB2   #16, NML$AB_MSGBLOCK                 ; 1065
              14    A6   F0      AD  9E 00101        MOVAB   STRDSC, NML$AB_MSGBLOCK+20           ; 1067
        FFFFFFF0    8F           53  D1 00106        CMPL    STATUS, #-16                         ; 1068
                                 11  12 0010D        BNEQ    8$                                   ;
                    66           30  D0 0010F        MOVL    #48, NML$AB_MSGBLOCK                 ; 1070
              04    A6           01  90 00112        MOVB    #1, NML$AB_MSGBLOCK+4                ; 1071
              0C    AE   04      AE  B0 00116        MOVW    LAD_LEN, LAD_DATA_DSC                ; 1072
              18    A6   0C      AE  9E 0011B        MOVAB   LAD_DATA_DSC, NML$AB_MSGBLOCK+24     ; 1073
                         08      AE  9F 00120 8$:    PUSHAB  MSGSIZE                              ; 1080
                                 56  DD 00123        PUSHL   R6                                   ;
                    68           02  FB 00125        CALLS   #2, NML$BLD_REPLY                    ;
                         08      AE  DD 00128        PUSHL   MSGSIZE                              ; 1082
                                 59  DD 0012B        PUSHL   R9                                   ; 1081
                    6A           02  FB 0012D        CALLS   #2, NML$SEND                         ;
                                     04 00130        RET                                          ; 1083

; Routine Size:  305 bytes,    Routine Base:  $CODE$ + 03F4

; 1094          1084  1
```

```
;  1096          1085  1 %SBTTL 'NML$SHOLINKS  Get logical link parameters'
;  1097          1086  1 GLOBAL ROUTINE NML$SHOLINKS (SEM_LIST, BUFDSC, MSGSIZE, DATDSC, DATPTR) =
;  1098          1087  1
;  1099          1088  1 !++
;  1100          1089  1 ! FUNCTIONAL DESCRIPTION:
;  1101          1090  1 !
;  1102          1091  1 !       This routine adds a logical link id to the NICE response message.
;  1103          1092  1 !
;  1104          1093  1 ! FORMAL PARAMETERS:
;  1105          1094  1 !
;  1106          1095  1 !       SEM_LIST        Parameter semantic table entry address.
;  1107          1096  1 !       BUFDSC          Output message buffer descriptor address.
;  1108          1097  1 !       MSGSIZE         Address of current output message size.
;  1109          1098  1 !       DATDSC          QIO buffer descriptor address.
;  1110          1099  1 !       DATPTR          Current pointer into QIO data buffer.
;  1111          1100  1 !
;  1112          1101  1 ! IMPLICIT INPUTS:
;  1113          1102  1 !
;  1114          1103  1 !       Coded multiple link address and process id fields are added to output
;  1115          1104  1 !       message.
;  1116          1105  1 !
;  1117          1106  1 ! ROUTINE VALUE:
;  1118          1107  1 ! COMPLETION CODES:
;  1119          1108  1 !
;  1120          1109  1 !       NML$_STS_SIZ if the response message buffer overflows.
;  1121          1110  1 !       NML$_STS_SUC
;  1122          1111  1 !
;  1123          1112  1 !--
;  1124          1113  1
;  1125          1114  2 BEGIN
;  1126          1115  2
;  1127          1116  2 MAP
;  1128          1117  2     DATDSC   : REF DESCRIPTOR,
;  1129          1118  2     SEM_LIST : REF BLOCK [, BYTE];
;  1130          1119  2
;  1131          1120  2 LOCAL
;  1132          1121  2     PRM_BUFFER  : BBLOCK [30],
;  1133          1122  2     PRMSIZE,
;  1134          1123  2     STRPTR,
;  1135          1124  2     STATUS;
;  1136          1125  2
;  1137          1126  2 !
;  1138          1127  2 ! Now, get the link address and PID and format them for the
;  1139          1128  2 ! NICE response message.
;  1140          1129  2 !
;  1141          1130  2 STRPTR = PRM_BUFFER;                                                    ; R
;  1142          1131  2 CH$WCHAR_A (2, STRPTR); ! Move link address
;  1143          1132  2 STRPTR = CH$MOVE (2, ..DATPTR, .STRPTR);
;  1144          1133  2 .DATPTR = ..DATPTR + 4;                                               ; 1
;  1145          1134  2
;  1146          1135  2 CH$WCHAR_A (%X'20' OR 4, STRPTR); ! Move process id
;  1147          1136  2 STRPTR = CH$MOVE (4, ..DATPTR, .STRPTR);
;  1148          1137  2 .DATPTR = ..DATPTR + 4;
;  1149          1138  2
;  1150          1139  2 PRMSIZE = .STRPTR - PRM_BUFFER;
;  1151          1140  2
;  1152          1141  2 STATUS = NML$ADDMSGPRM (.BUFDSC,
```

```
; 1153       1142  2                    .MSGSIZE,
; 1154       1143  2                    NMA$C_PCLK_LAD,
; 1155       1144  2                    NMA$M_PTY_CMU OR 2,
; 1156       1145  2                    .PRMSIZE,
; 1157       1146  2                    PRM_BUFFER);
; 1158       1147  2
; 1159       1148  2 RETURN .STATUS
; 1160       1149  2
; 1161       1150  1 END;                            ! End of NML$SHOLINKS


                                        0000 00000    .ENTRY  NML$SHOLINKS, Save nothing       ; 1086
                       5E             20  C2 00002    SUBL2   #32, SP                          ;
                       51             6E  9E 00005    MOVAB   PRM_BUFFER, STRPTR               ; 1130
                       81             02  90 00008    MOVB    #2, (STRPTR)+                    ; 1131
                       50       14    BC  D0 0000B    MOVL    @DATPTR, R0                      ; 1132
                       81             60  B0 0000F    MOVW    (R0), (STRPTR)+                  ;
                 14    BC             04  C0 00012    ADDL2   #4, @DATPTR                      ; 1133
                       81             24  90 00016    MOVB    #36, (STRPTR)+                   ; 1135
                       50       14    BC  D0 00019    MOVL    @DATPTR, R0                      ; 1136
                       81             60  D0 0001D    MOVL    (R0), (STRPTR)+                  ;
                 14    BC             04  C0 00020    ADDL2   #4, @DATPTR                      ; 1137
                       50             6E  9E 00024    MOVAB   PRM_BUFFER, R0                   ; 1139
          50           51             50  C3 00027    SUBL3   R0, STRPTR, PRMSIZE              ;
                              4001    8F  BB 0002B    PUSHR   #^M<R0,SP>                       ; 1145
                       7E       C2    8F  9A 0002F    MOVZBL  #194, -(SP)                      ; 1144
                       7E       69    8F  9A 00033    MOVZBL  #105, -(SP)                      ; 1141
                       7E       08    AC  7D 00037    MOVQ    BUFDSC, -(SP)                    ;
             00000000G 00             06  FB 0003B    CALLS   #6, NML$ADDMSGPRM                ;
                                          04 00042    RET                                      ; 1150

; Routine Size:  67 bytes,    Routine Base:  $CODE$ + 0525
```

```
;  1163          1151  1 %SBTTL 'NML$V2_CHG_LINE  Set V2 line parameters'
;  1164          1152  1 ROUTINE NML$V2_CHG_LINE : NOVALUE =
;  1165          1153  1
;  1166          1154  1 !++
;  1167          1155  1 ! FUNCTIONAL DESCRIPTION:
;  1168          1156  1 !
;  1169          1157  1 !       This routine is called when NML receives a SET or CLEAR LINE command
;  1170          1158  1 !       from a V2 NCP.  It transforms the V2 SET or CLEAR LINE command into
;  1171          1159  1 !       the appropriate V3 QIO.  Note that some V2 line parameters are
;  1172          1160  1 !       V3 circuit parameters.  Line and circuit parameters may not be
;  1173          1161  1 !       mixed in a single V2 command.
;  1174          1162  1 !
;  1175          1163  1 !--
;  1176          1164  1
;  1177          1165  2 BEGIN
;  1178          1166  2
;  1179          1167  2 MAP
;  1180          1168  2         NML$GB_ENTITY_FORMAT  :  BYTE SIGNED;
;  1181          1169  2
;  1182          1170  2 LOCAL
;  1183          1171  2     FUNCTION,
;  1184          1172  2     NPARSE_TAB;
;  1185          1173  2 !
;  1186          1174  2 ! Information can be read only from volatile data bases.
;  1187          1175  2 !
;  1188          1176  2 IF NOT .NML$GB_OPTIONS [NMA$V_OPT_PER]  ! If volatile database requested,
;  1189          1177  2 THEN
;  1190          1178  3     BEGIN
;  1191          1179  3     IF .NML$GB_OPTIONS [NMA$V_OPT_CLE]
;  1192          1180  3     THEN
;  1193          1181  4         BEGIN
;  1194          1182  4         NPARSE_TAB = NML$NPA_CLEARV2LINE;
;  1195          1183  4         FUNCTION = NFB$C_FC_CLEAR;
;  1196          1184  4         END
;  1197          1185  3     ELSE
;  1198          1186  4         BEGIN
;  1199          1187  4         NPARSE_TAB = NML$NPA_SETV2LINE;
;  1200          1188  4         FUNCTION = NFB$C_FC_SET;
;  1201          1189  3         END;
;  1202          1190  3     IF NMA$NPARSE (NML$AB_NPA_BLK,
;  1203          1191  3                    .NPARSE_TAB)
;  1204          1192  3     THEN
;  1205          1193  3         SELECTONEU .NML$GB_ENTITY_FORMAT OF
;  1206          1194  3             SET
;  1207          1195  3             [NMA$C_ENT_KNO]:            ! Known
;  1208          1196  3                 NML_V2_DISPATCH (.NML$L_V2_ENTITY,
;  1209          1197  3                                  NML_V2_CHG_KNOWN,
;  1210          1198  3                                  .FUNCTION, 0);
;  1211          1199  3
;  1212          1200  3             [1 TO 16]:
;  1213          1201  3                 NML_V2_DISPATCH (.NML$L_V2_ENTITY,
;  1214          1202  3                                  NML_V2_CHG_LINE,
;  1215          1203  3                                  .NML$GB_ENTITY_FORMAT,
;  1216          1204  3                                  NML$AB_ENTITY_ID,
;  1217          1205  3                                  .FUNCTION);
;  1218          1206  3             TES;
;  1219          1207  3     NML$ERROR_2 (NMA$C_STS_IDE, NMA$C_ENT_LIN);
```

```
; 1220          1208  2       END;
; 1221          1209  2 NML$ERROR_1 (NMA$C_STS_FUN, NMA$C_ENT_LIN);
; 1222          1210  1 END;      !  of    NML$V2_CHG_LINE


                                        003C 00000 NML$V2_CHG_LINE:
                                                        .WORD   Save R2,R3,R4,R5                                ; 1152
                    55 00000000G  00   9E 00002         MOVAB   NML$GB_OPTIONS, R5                              ;
                    54      FBD3  CF   9E 00009         MOVAB   NML_V2_DISPATCH, R4                             ;
                    53 00000000'  00   9E 0000E         MOVAB   NML$L_V2_ENTITY, R3                             ;
                                  65   95 00015         TSTB    NML$GB_OPTIONS                                  ; 1176
                                  74   19 00017         BLSS    5$                                              ;
        0C          65            06   E1 00019         BBC     #6, NML$GB_OPTIONS, 1$                          ; 1179
                    50 00000000G  00   9E 0001D         MOVAB   NML$NPA_CLEARV2LINE, NPARSE_TAB                 ; 1182
                    52            24   D0 00024         MOVL    #36, FUNCTION                                   ; 1183
                                  0A   11 00027         BRB     2$                                              ; 1179
                    50 00000000G  00   9E 00029 1$:     MOVAB   NML$NPA_SETV2LINE, NPARSE_TAB                   ; 1187
                    52            23   D0 00030         MOVL    #35, FUNCTION                                   ; 1188
                                  50   DD 00033 2$:     PUSHL   NPARSE_TAB                                      ; 1191
                       00000000G  00   9F 00035         PUSHAB  NML$AB_NPA_BLK                                  ; 1190
           00000000G  00          02   FB 0003B         CALLS   #2, NMA$NPARSE                                  ;
                      3C          50   E9 00042         BLBC    R0, 4$                                          ;
                    50 00000000G  00   98 00045         CVTBL   NML$GB_ENTITY_FORMAT, R0                        ; 1193
               FF   8F            50   91 0004C         CMPB    R0, #-1                                         ; 1195
                                  11   12 00050         BNEQ    3$                                              ;
                                  7E   D4 00052         CLRL    -(SP)                                           ; 1196
                                  52   DD 00054         PUSHL   FUNCTION                                        ; 1198
                       00000000V  00   9F 00056         PUSHAB  NML_V2_CHG_KNOWN                                ; 1196
                                  63   DD 0005C         PUSHL   NML$L_V2_ENTITY                                 ;
                    64            04   FB 0005E         CALLS   #4, NML_V2_DISPATCH                             ;
                                  1E   11 00061         BRB     4$                                              ;
                                  50   D5 00063 3$:     TSTL    R0                                              ; 1200
                                  1A   13 00065         BEQL    4$                                              ;
                    10            50   91 00067         CMPB    R0, #16                                         ;
                                  15   1A 0006A         BGTRU   4$                                              ;
                                  52   DD 0006C         PUSHL   FUNCTION                                        ; 1205
                       00000000G  00   9F 0006E         PUSHAB  NML$AB_ENTITY_ID                                ; 1201
                                  50   DD 00074         PUSHL   R0                                              ; 1203
                       00000000V  00   9F 00076         PUSHAB  NML_V2_CHG_LINE                                 ; 1201
                                  63   DD 0007C         PUSHL   NML$L_V2_ENTITY                                 ;
                    64            05   FB 0007E         CALLS   #5, NML_V2_DISPATCH                             ;
                                  01   DD 00081 4$:     PUSHL   #1                                              ; 1207
                    7E            09   CE 00083         MNEGL   #9, -(SP)                                       ;
           00000000G  00          02   FB 00086         CALLS   #2, NML$ERROR_2                                 ;
                                  01   DD 0008D 5$:     PUSHL   #1                                              ; 1209
                    7E            01   CE 0008F         MNEGL   #1, -(SP)                                       ;
           00000000G  00          02   FB 00092         CALLS   #2, NML$ERROR_1                                 ;
                                       04 00099         RET                                                     ; 1210

; Routine Size:  154 bytes,    Routine Base:  $CODE$ + 0568
```

```
; 1224      1211  1 %SBTTL  'NML$CHK_V2_CIRC        Check Set V2 Circuit parameter group'
; 1225      1212  1 GLOBAL ROUTINE NML$CHK_V2_CIRC =
; 1226      1213  1
; 1227      1214  1 !++
; 1228      1215  1 ! FUNCTIONAL DESCRIPTION:
; 1229      1216  1 !
; 1230      1217  1 !       This is an NPARSE action routine that is called when parsing a
; 1231      1218  1 !       SET LINE command from a V2 NCP.  These commands could have both
; 1232      1219  1 !       line and circuit parameters in the same command.  To adhere with
; 1233      1220  1 !       Network Management architecture, we do not allow a mix in a single
; 1234      1221  1 !       SET command.  Check the parameter code to make sure it is a circuit
; 1235      1222  1 !       parameter.
; 1236      1223  1 !
; 1237      1224  1 ! IMPLICIT INPUTS:
; 1238      1225  1 !       NPARSE_BLOCK (pointed to by AP) contains the parsed parameter data.
; 1239      1226  1 !       NPA$L_FLDPTR is a pointer to the parameter code in the received
; 1240      1227  1 !       message buffer.
; 1241      1228  1 !
; 1242      1229  1 !       If the parameter is not a circuit parameter, then an invalid parameter
; 1243      1230  1 !       grouping error (NMA$C_STS_PGP) is signalled.
; 1244      1231  1 !--
; 1245      1232  1
; 1246      1233  2 BEGIN
; 1247      1234  2
; 1248      1235  2 $NPA_ARGDEF;                ! Define NPARSE block reference.
; 1249      1236  2
; 1250      1237  2 !
; 1251      1238  2 ! If this is not a circuit parameter, return error.
; 1252      1239  2 !
; 1253      1240  2 IF .NML$GL_PRS_FLGS [NML$V_PRS_V2_LINE]
; 1254      1241  2 THEN
; 1255      1242  2     NML$ERROR_2 (NMA$C_STS_PGP,
; 1256      1243  2                 .(.NPARSE_BLOCK [NPA$L_FLDPTR])<0,16>);
; 1257      1244  2 NML$GL_PRS_FLGS [NML$V_PRS_V2_CIRCUIT] = 1;      ! Set grouping flag.
; 1258      1245  2 NML$L_V2_ENTITY = NML$C_CIRCUIT;
; 1259      1246  2 RETURN NML$_STS_SUC
; 1260      1247  1 END;                         ! End of NML$CHK_V2_CIRC


                                         0000 00000        .ENTRY  NML$CHK_V2_CIRC, Save nothing          ; 1212
    0E 00000000G  00               06  E1 00002        BBC     #6, NML$GL_PRS_FLGS+1, 1$              ; 1240
              7E        14         BC  3C 0000A        MOVZWL  @20(NPARSE_BLOCK), -(SP)               ; 1243
              7E                   1B  CE 0000E        MNEGL   #27, -(SP)                             ; 1242
       00000000G  00               02  FB 00011        CALLS   #2, NML$ERROR_2                        ;
       00000000G  00        80     8F  88 00018 1$:    BISB2   #128, NML$GL_PRS_FLGS+1                ; 1244
       00000000'  00               09  D0 00020        MOVL    #9, NML$L_V2_ENTITY                    ; 1245
              50                   01  D0 00027        MOVL    #1, R0                                 ; 1246
                                       04 0002A        RET                                            ; 1247

; Routine Size:  43 bytes,    Routine Base:  $CODE$ + 0602
```

```
; 1262        1248  1 %SBTTL  'NML$CHK_V2_LINE        Check Set V2 Line parameter group'
; 1263        1249  1 GLOBAL ROUTINE NML$CHK_V2_LINE =
; 1264        1250  1
; 1265        1251  1 !++
; 1266        1252  1 ! FUNCTIONAL DESCRIPTION:
; 1267        1253  1 !
; 1268        1254  1 !       This is an NPARSE action routine that is called when parsing a
; 1269        1255  1 !       SET LINE command from a V2 NCP.  These commands could have both
; 1270        1256  1 !       line and circuit parameters in the same command.  To adhere with
; 1271        1257  1 !       Network Management architecture, we do not allow a mix in a single
; 1272        1258  1 !       SET command.  Check the parameter code to make sure it is a line
; 1273        1259  1 !       parameter.
; 1274        1260  1 !
; 1275        1261  1 ! IMPLICIT INPUTS:
; 1276        1262  1 !       NPARSE_BLOCK (pointed to by AP) contains the parsed parameter data.
; 1277        1263  1 !       NPA$L_FLDPTR is a pointer to the parameter code in the received
; 1278        1264  1 !       message buffer.
; 1279        1265  1 !
; 1280        1266  1 !       If the parameter is not a line parameter, then an invalid parameter
; 1281        1267  1 !       grouping error (NMA$C_STS_PGP) is signalled.
; 1282        1268  1 !--
; 1283        1269  1
; 1284        1270  2 BEGIN
; 1285        1271  2
; 1286        1272  2 $NPA_ARGDEF;              ! Define NPARSE block reference.
; 1287        1273  2
; 1288        1274  2 !
; 1289        1275  2 ! If this is not a line parameter, return error.
; 1290        1276  2 !
; 1291        1277  2 IF .NML$GL_PRS_FLGS [NML$V_PRS_V2_CIRCUIT]
; 1292        1278  2 THEN
; 1293        1279  2     NML$ERROR_2 (NMA$C_STS_PGP,
; 1294        1280  2                 .(.NPARSE_BLOCK [NPA$L_FLDPTR])<0,16>);
; 1295        1281  2 NML$GL_PRS_FLGS [NML$V_PRS_V2_LINE] = 1;        ! Set grouping flag.
; 1296        1282  2 NML$L_V2_ENTITY = NML$C_LINE;
; 1297        1283  2 RETURN NML$_STS_SUC
; 1298        1284  1 END;                      ! End of NML$CHK_V2_LINE


                                   0000 00000        .ENTRY  NML$CHK_V2_LINE, Save nothing        ; 1249
                     00000000G  00  95 00002         TSTB    NML$GL_PRS_FLGS+1                     ; 1277
                                0E  18 00008         BGEQ    1$                                    ;
               7E          14   BC  3C 0000A         MOVZWL  @20(NPARSE_BLOCK), -(SP)              ; 1280
               7E               1B  CE 0000E         MNEGL   #27, -(SP)                            ; 1279
      00000000G  00             02  FB 00011         CALLS   #2, NML$ERROR_2                       ;
      00000000G  00          40 8F  88 00018 1$:     BISB2   #64, NML$GL_PRS_FLGS+1                ; 1281
                     00000000'  00  D4 00020         CLRL    NML$L_V2_ENTITY                       ; 1282
               50               01  D0 00026         MOVL    #1, R0                                ; 1283
                                    04 00029         RET                                           ; 1284

; Routine Size:  42 bytes,    Routine Base:  $CODE$ + 062D
```

```
1 %SBTTL  'NML$CHK_V2_STA Check Set V2 Line parameter group'
1 GLOBAL ROUTINE NML$CHK_V2_STA=
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This is an NPARSE action routine that is called when parsing a
1 !       SET LINE command from a V2 NCP, and a state change is found.
1 !       Set up the proper fields so the state change is made to both
1 !       the line and the circuit.  State is the only V2 parameter for
1 !       which this is done.
1 !
1 ! IMPLICIT INPUTS:
1 !       NPARSE_BLOCK (pointed to by AP) contains the parsed parameter data.
1 !       NPA$L_FLDPTR is a pointer to the parameter code in the received
1 !       message buffer.
1 !
1 !--
1
2 BEGIN
2
2 $NPA_ARGDEF;              ! Define NPARSE block reference.
2
2
2 !
2 ! Save the new state.
2 !
2 NML$L_STATE = .(.NPARSE_BLOCK [NPA$L_FLDPTR])<0,8>;
2 NML$GL_PRS_FLGS [NML$V_PRS_V2_STA] = 1; ! Set state change flag.
2 RETURN NML$_STS_SUC
1 END;                      ! End of NML$CHK_V2_LINE
```

```
                       0000 00000      .ENTRY  NML$CHK_V2_STA, Save nothing          ; 1286
00000000'  00    14  BC  9A 00002      MOVZBL  @20(NPARSE_BLOCK), NML$L_STATE        ; 1312
00000000G  00        01  88 0000A      BISB2   #1, NML$GL_PRS_FLGS+2                 ; 1313
       50            01  D0 00011      MOVL    #1, R0                                ; 1314
                         04 00014      RET                                           ; 1315

; Routine Size:  21 bytes,    Routine Base:  $CODE$ + 0657
```

```
;  1332          1316  1 %SBTTL 'NML_V2_CHG_LINE  Set volatile database line parameters'
;  1333          1317  1 ROUTINE NML_V2_CHG_LINE (ENT, LEN, ADR, FCN) : NOVALUE =
;  1334          1318  1
;  1335          1319  1 !++
;  1336          1320  1 ! FUNCTIONAL DESCRIPTION:
;  1337          1321  1 !       This routine adds and clears parameters in the volatile data
;  1338          1322  1 !       base for V2 line entities.  Since the line entity was broken
;  1339          1323  1 !       into the line and circuit entities for V3, this can require a
;  1340          1324  1 !       QIO to either data base.  Only the state parameter is updated
;  1341          1325  1 !       in both data bases.
;  1342          1326  1 !
;  1343          1327  1 ! FORMAL PARAMETERS:
;  1344          1328  1 !       ENT             Entity type code.
;  1345          1329  1 !       LEN             Byte count of entity id string.
;  1346          1330  1 !       ADR             Address of entity id string.
;  1347          1331  1 !       FCN             Function (set or clear)
;  1348          1332  1 !
;  1349          1333  1 !--
;  1350          1334  2 BEGIN
;  1351          1335  2
;  1352          1336  2 MAP
;  1353          1337  2         NML$GB_ENTITY_FORMAT  :  BYTE SIGNED;
;  1354          1338  2
;  1355          1339  2 LOCAL
;  1356          1340  2     STATE_LGTH,
;  1357          1341  2     MSGSIZE,
;  1358          1342  2     STATUS;
;  1359          1343  2
;  1360          1344  2 !
;  1361          1345  2 ! If there is a state parameter in the NICE command, add it to the
;  1362          1346  2 ! parameter list using the field ID for the appropriate data base.
;  1363          1347  2 !
;  1364          1348  2 IF .NML$GL_PRS_FLGS [NML$V_PRS_V2_STA]
;  1365          1349  2 THEN
;  1366          1350  3     BEGIN
;  1367          1351  3     IF .FCN EQL NFB$C_FC_CLEAR THEN
;  1368          1352  3         STATE_LGTH = 0
;  1369          1353  3     ELSE
;  1370          1354  3         STATE_LGTH = 1;
;  1371          1355  3     IF .ENT EQL NML$C_LINE
;  1372          1356  3     THEN
;  1373          1357  3         NML$SAVEPARAM ( CPT$GK_PCLI_STA, .STATE_LGTH, NML$L_STATE)
;  1374          1358  3     ELSE
;  1375          1359  3         NML$SAVEPARAM ( CPT$GK_PCCI_STA, .STATE_LGTH, NML$L_STATE);
;  1376          1360  2     END;
;  1377          1361  2 STATUS = NML_V2_CHG_ENTITY (.ENT, .LEN, .ADR, .FCN);
;  1378          1362  2 IF .STATUS
;  1379          1363  2    AND .NML$GL_PRS_FLGS [NML$V_PRS_V2_STA]
;  1380          1364  2 THEN
;  1381          1365  2     !
;  1382          1366  2     ! If there is a state change in the NICE command, it must be made
;  1383          1367  2     ! to both the circuit and line data bases.  Update the data base
;  1384          1368  2     ! not already done here.
;  1385          1369  2     !
;  1386          1370  3     BEGIN
;  1387          1371  3     NML$GW_PRMDESCNT = 0;       ! Only update the state this time.
;  1388          1372  3     IF .ENT EQL NML$C_LINE
```

```
;  1389          1373  3     THEN
;  1390          1374  4         BEGIN
;  1391          1375  4         ENT = NML$C_CIRCUIT;
;  1392          1376  4         NML$SAVEPARAM ( CPT$GK_PCCI_STA, .STATE_LGTH, NML$L_STATE);
;  1393          1377  4         END
;  1394          1378  3     ELSE
;  1395          1379  4         BEGIN
;  1396          1380  4         ENT = NML$C_LINE;
;  1397          1381  4         NML$SAVEPARAM ( CPT$GK_PCLI_STA, .STATE_LGTH, NML$L_STATE);
;  1398          1382  3         END;
;  1399          1383  3     STATUS = NML_V2_CHG_ENTITY (.ENT, .LEN, .ADR, .FCN);
;  1400          1384  2     END;
;  1401          1385  2 IF .NML$GB_ENTITY_FORMAT EQL NMA$C_ENT_KNO THEN
;  1402          1386  3     BEGIN
;  1403          1387  3     !
;  1404          1388  3     ! If updating KNOWN lines, add the entity identification to the
;  1405          1389  3     ! NICE response message.
;  1406          1390  3     !
;  1407          1391  3     NML$AB_MSGBLOCK [MSB$V_ENTD_FLD] = 1;
;  1408          1392  3     NML$AB_MSGBLOCK [MSB$A_ENTITY] = NML$Q_ENTBFDSC;
;  1409          1393  2     END;
;  1410          1394  2 !
;  1411          1395  2 ! Build and send the response message.
;  1412          1396  2 !
;  1413          1397  2 NML$BLD_REPLY (NML$AB_MSGBLOCK, MSGSIZE);
;  1414          1398  2 NML$SEND (NML$AB_SNDBUFFER, .MSGSIZE);
;  1415          1399  1 END;                          ! End of NML_V2_CHG_LINE

                                03FC 00000 NML_V2_CHG_LINE:
                                           .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9          ; 1317
59 00000000V    00   9E 00002              MOVAB   NML_V2_CHG_ENTITY, R9                 ;
58 00000000G    00   9E 00009              MOVAB   NML$SAVEPARAM, R8                     ;
57 00000000G    8F   D0 0C010              MOVL    #CPT$GK_PCCI_STA, R7                  ;
56 00000000G    8F   D0 00017              MOVL    #CPT$GK_PCLI_STA, R6                  ;
55 00000000G    00   9E 0001E              MOVAB   NML$AB_MSGBLOCK, R5                   ;
54 00000000'    00   9E 00025              MOVAB   NML$L_STATE, R4                       ;
5E              04   C2 0002C              SUBL2   #4, SP                                ;
1F 00000000G    00   E9 0002F              BLBC    NML$GL_PRS_FLGS+2, 5$                 ; 1348
24        10    AC   D1 00036              CMPL    FCN, #36                              ; 1351
                04   12 0003A              BNEQ    1$                                    ;
                52   D4 0003C              CLRL    STATE_LGTH                            ; 1352
                03   11 0003E              BRB     2$                                    ;
52              01   D0 00040 1$:          MOVL    #1, STATE_LGTH                        ; 1354
          04    AC   D5 00043 2$:          TSTL    ENT                                   ; 1355
                06   12 00045              BNEQ    3$                                    ;
                14   BB 00048              PUSHR   #^M<R2,R4>                            ; 1357
                56   DD 0004A              PUSHL   R6                                    ;
                04   11 0004C              BRB     4$                                    ;
                14   BB 0004E 3$:          PUSHR   #^M<R2,R4>                            ; 1359
                57   DD 00050              PUSHL   R7                                    ;
68              03   FB 00052 4$:          CALLS   #3, NML$SAVEPARAM                     ;
7E        0C    AC   7D 00055 5$:          MOVQ    ADR, -(SP)                            ; 1361
7E        04    AC   7D 00059              MOVQ    ENT, -(SP)                            ;
```

```
                                69              04  FB 0005D          CALLS   #4, NML_V2_CHG_ENTITY                    ;
                                53              50  D0 00060          MOVL    R0, STATUS                              ;
                                34              53  E9 00063          BLBC    STATUS, 8$                              ; 1362
                                2D 00000000G    00  E9 00066          BLBC    NML$GL_PRS_FLGS+2, 8$                   ; 1363
                                   00000000G    00  B4 0006D          CLRW    NML$GW_PRMDESCNT                        ; 1371
                                           04   AC  D5 00073          TSTL    ENT                                     ; 1372
                                               0A  12 00076          BNEQ    6$                                      ;
                             04    AC                09  D0 00078          MOVL    #9, ENT                             ; 1375
                                               14  BB 0007C          PUSHR   #^M<R2,R4>                              ; 1376
                                               57  DD 0007E          PUSHL   R7                                      ;
                                               07  11 00080          BRB     7$                                      ;
                                           04   AC  D4 00082  6$:     CLRL    ENT                                     ; 1380
                                               14  BB 00085          PUSHR   #^M<R2,R4>                              ; 1381
                                               56  DD 00087          PUSHL   R6                                      ;
                                68              03  FB 00089  7$:     CALLS   #3, NML$SAVEPARAM                       ;
                                7E          0C  AC  7D 0008C          MOVQ    ADR, -(SP)                              ; 1383
                                7E          04  AC  7D 00090          MOVQ    ENT, -(SP)                              ;
                                69              04  FB 00094          CALLS   #4, NML_V2_CHG_ENTITY                   ;
                                53              50  D0 00097          MOVL    R0, STATUS                              ;
                             FF 8F 00000000G    00  91 0009A  8$:     CMPB    NML$GB_ENTITY_FORMAT, #-1               ; 1385
                                               09  12 000A2          BNEQ    9$                                      ;
                                65              10  88 000A4          BISB2   #16, NML$AB_MSGBLOCK                    ; 1391
                             14 A5      0110    C4  9E 000A7          MOVAB   NML$Q_ENTBFDSC, NML$AB_MSGBLOCK+20      ; 1392
                                        4020    8F  BB 000AD  9$:     PUSHR   #^M<R5,SP>                              ; 1397
                   00000000G    00              02  FB 000B1          CALLS   #2, NML$BLD_REPLY                       ;
                                               6E  DD 000B8          PUSHL   MSGSIZE                                 ; 1398
                                   00000000G    00  9F 000BA          PUSHAB  NML$AB_SNDBUFFER                        ;
                   00000000G    00              02  FB 000C0          CALLS   #2, NML$SEND                            ;
                                               04  00C7          RET                                             ; 1399

; Routine Size:  200 bytes,    Routine Base:  $CODE$ + 066C
```

```
;  1417          1400  1 %SBTTL 'NML_V2_CHG_ENTITY  Set volatile database line parameters'
;  1418          1401  1 ROUTINE NML_V2_CHG_ENTITY (ENT, LEN, ADR, FCN) =
;  1419          1402  1
;  1420          1403  1 !++
;  1421          1404  1 ! FUNCTIONAL DESCRIPTION:
;  1422          1405  1 !
;  1423          1406  1 !       This routine adds or clears the specified V2 parameters in
;  1424          1407  1 !       the volatile data base entry for the specified component.
;  1425          1408  1 !
;  1426          1409  1 ! FORMAL PARAMETERS:
;  1427          1410  1 !
;  1428          1411  1 !       ENT             Entity type code.
;  1429          1412  1 !       LEN             Byte count of entity id string.
;  1430          1413  1 !       ADR             Address of entity id string.
;  1431          1414  1 !       FCN             Function (set or clear)
;  1432          1415  1 !
;  1433          1416  1 ! ROUTINE VALUE:
;  1434          1417  1 ! COMPLETION CODES:
;  1435          1418  1 !
;  1436          1419  1 !       The translated status of the SET QIO is returned.
;  1437          1420  1 !--
;  1438          1421  1
;  1439          1422  2 BEGIN
;  1440          1423  2
;  1441          1424  2 LOCAL
;  1442          1425  2         DB,                             ! Database ID
;  1443          1426  2         SRCHKEY1,                       ! Search key one ID
;  1444          1427  2         SRCHKEY2,                       ! Search key two ID
;  1445          1428  2         NFBDSC : DESCRIPTOR,            ! NFB buffer descriptor
;  1446          1429  2         P2DSC  : DESCRIPTOR,            ! QIO P2 buffer descriptor
;  1447          1430  2         QBFDSC : DESCRIPTOR,            ! QIO P4 buffer descriptor
;  1448          1431  2         STATUS;
;  1449          1432  2
;  1450          1433  2 STATUS = NML$_STS_SUC;
;  1451          1434  2 !
;  1452          1435  2 ! Get entity information.
;  1453          1436  2 !
;  1454          1437  2 DB      = .NML$AB_ENTITYDATA [.ENT, EIT$B_DATABASE];! Database ID
;  1455          1438  2 SRCHKEY1 = .NML$AB_ENTITYDATA [.ENT, EIT$L_SRCH_ID1]; ! Search key one ID
;  1456          1439  2 SRCHKEY2 = .NML$AB_ENTITYDATA [.ENT, EIT$L_SRCH_ID2]; ! Search key two ID
;  1457          1440  2 !
;  1458          1441  2 ! Build the NFB and P2 buffers for the QIO to NETACP.
;  1459          1442  2 !
;  1460          1443  2 NML$BLDSETQBF (.FCN, .DB,
;  1461          1444  2                 .SRCHKEY1, .LEN, .ADR,
;  1462          1445  2                 .SRCHKEY2, -1, 0,
;  1463          1446  2                 NML$Q_NFBBFDSC, NFBDSC,
;  1464          1447  2                 NML$Q_P2BFDSC, P2DSC,
;  1465          1448  2                 NML$GQ_QIOBFDSC, QBFDSC);
;  1466          1449  2 !
;  1467          1450  2 ! Add the parameters to volatile data base entry.
;  1468          1451  2 !
;  1469          1452  2 STATUS = NML$NETQIO (NFBDSC, P2DSC, 0, QBFDSC);
;  1470          1453  2 IF .STATUS THEN
;  1471          1454  3     BEGIN
;  1472          1455  3     NML$AB_MSGBLOCK [MSB$L_FLAGS] = 0;
;  1473          1456  3     NML$AB_MSGBLOCK [MSB$B_CODE] = NMA$C_STS_SUC;
```

```
; 1474        1457  2      END;
; 1475        1458  2 RETURN .STATUS
; 1476        1459  1 END;                          ! End of NML_V2_CHG_ENTITY


                                     001C 00000 NML_V2_CHG_ENTITY:
                                                           .WORD    Save R2,R3,R4                          ; 1401
                 54 00000000G  00     9E 00002             MOVAB    NML$AB_ENTITYDATA+5, R4
                 5E              18   C2 00009             SUBL2    #24, SP
                 53              01   D0 0000C             MOVL     #1, STATUS                             ; 1433
  50       04    AC              2C   C5 0000F             MULL3    #44, ENT, R0                           ; 1437
                 52            6440   9A 00014             MOVZBL   NML$AB_ENTITYDATA+5[R0], DB
                          01 A440     9F 00018             PUSHAB   NML$AB_ENTITYDATA+6[R0]                ; 1438
                 51              9E   D0 0001C             MOVL     @(SP)+, SRCHKEY1
                          05 A440     9F 0001F             PUSHAB   NML$AB_ENTITYDATA+10[R0]               ; 1439
                 50              9E   D0 00023             MOVL     @(SP)+, SRCHKEY2
                                 5E   DD 00026             PUSHL    SP                                     ; 1443
                    00000000G    00   9F 00028             PUSHAB   NML$GQ_QIOBFDSC
                           10    AE   9F 0002E             PUSHAB   P2DSC
                    00000000'    00   9F 00031             PUSHAB   NML$Q_P2BFDSC
                           20    AE   9F 00037             PUSHAB   NFBDSC
                    00000000'    00   9F 0003A             PUSHAB   NML$Q_NFBBFDSC
                                 7E   D4 00040             CLRL     -(SP)
                 7E              01   CE 00042             MNEGL    #1, -(SP)                              ; 1445
                                 50   DD 00045             PUSHL    SRCHKEY2
                 7E        08    AC   7D 00047             MOVQ     LEN, -(SP)                             ; 1444
                                 51   DD 0004B             PUSHL    SRCHKEY1
                                 52   DD 0004D             PUSHL    DB                                     ; 1443
                           10    AC   DD 0004F             PUSHL    FCN
      00000000G  00              0E   FB 00052             CALLS    #14, NML$BLDSETQBF
                                 5E   DD 00059             PUSHL    SP                                     ; 1452
                                 7E   D4 0005B             CLRL     -(SP)
                           10    AE   9F 0005D             PUSHAB   P2DSC
                           1C    AE   9F 00060             PUSHAB   NFBDSC
      00000000G  00              04   FB 00063             CALLS    #4, NML$NETQIO
                 53              50   D0 0006A             MOVL     R0, STATUS
                 0D              53   E9 0006D             BLBC     STATUS, 1$                             ; 1453
                    00000000G    00   D4 00070             CLRL     NML$AB_MSGBLOCK                        ; 1455
      00000000G  00              01   90 00076             MOVB     #1, NML$AB_MSGBLOCK+4                  ; 1456
                 50              53   D0 0007D 1$:         MOVL     STATUS, R0                             ; 1458
                                      04 00080             RET                                             ; 1459

; Routine Size:  129 bytes,    Routine Base:  $CODE$ + 0734
```

```
;  1478          1460  1 %SBTTL 'NML_V2_CHG_KNOWN  Set volatile entity parameters'
;  1479          1461  1 ROUTINE NML_V2_CHG_KNOWN (ENT, FCN) : NOVALUE =
;  1480          1462  1
;  1481          1463  1 !++
;  1482          1464  1 ! FUNCTIONAL DESCRIPTION:
;  1483          1465  1 !
;  1484          1466  1 !       This routine sets or clears the specified parameters for each
;  1485          1467  1 !       of the components of the given entity type.
;  1486          1468  1 !
;  1487          1469  1 ! INPUTS:
;  1488          1470  1 !
;  1489          1471  1 !       ENT     Entity type code.
;  1490          1472  1 !       FCN     Function (set or clear).
;  1491          1473  1 !
;  1492          1474  1 !--
;  1493          1475  1
;  1494          1476  2 BEGIN
;  1495          1477  2
;  1496          1478  2 LOCAL
;  1497          1479  2         BUFEND,
;  1498          1480  2         ENTADD,
;  1499          1481  2         ENTLEN,
;  1500          1482  2         LISDSC    : DESCRIPTOR,
;  1501          1483  2         ENTIDPTR,
;  1502          1484  2         PTR,
;  1503          1485  2         STATUS,
;  1504          1486  2         STRTFLG;
;  1505          1487  2 !
;  1506          1488  2 ! Process every entry in the data base.
;  1507          1489  2 !
;  1508          1490  2 STRTFLG = FALSE;
;  1509          1491  2 WHILE NML$GET_ENTITY_IDS (.ENT, NMA$C_ENT_KNO, 0, .STRTFLG, LISDSC) DO
;  1510          1492  3     BEGIN
;  1511          1493  3
;  1512          1494  3     STRTFLG = TRUE;
;  1513          1495  3
;  1514          1496  3     BUFEND = .LISDSC [DSC$A_POINTER] + .LISDSC [DSC$W_LENGTH];
;  1515          1497  3     PTR = .LISDSC [DSC$A_POINTER];
;  1516          1498  3
;  1517          1499  3     WHILE .PTR LSSA .BUFEND DO
;  1518          1500  4         BEGIN
;  1519          1501  4
;  1520          1502  4         ENTIDPTR = NML$T_ENTBUFFER;
;  1521          1503  4         NML$Q_ENTBFDSC [DSC$W_LENGTH] = NML$K_ENTBUFLEN;
;  1522          1504  4 !
;  1523          1505  4 ! Get entity id for SET QIO and id string for response message.
;  1524          1506  4 !
;  1525          1507  4         ENTLEN = .(.PTR)<0,16>;
;  1526          1508  4         PTR = .PTR + 2;
;  1527          1509  4         ENTADD = .PTR;
;  1528          1510  4         CH$WCHAR_A (.ENTLEN, ENTIDPTR);
;  1529          1511  4         ENTIDPTR = CH$MOVE (.ENTLEN,
;  1530          1512  4                             .ENTADD,
;  1531          1513  4                             .ENTIDPTR);
;  1532          1514  4         PTR = .PTR + .ENTLEN;
;  1533          1515  4
;  1534          1516  4         NML$Q_ENTBFDSC [DSC$W_LENGTH] = .ENTIDPTR - NML$T_ENTBUFFER;
```

```
; 1535          1517  4          !
; 1536          1518  4          ! Add the parameters to volatile data base entry.
; 1537          1519  4          !
; 1538          1520  4          NML_V2_CHG_LINE ( .ENT, .ENTLEN, .ENTADD, .FCN);
; 1539          1521  3          END;
; 1540          1522  2       END;
; 1541          1523  1 END;                            ! End of NML_V2_CHG_KNOWN


                                       0FFC 00000 NML_V2_CHG_KNOWN:
                                                  .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11              ; 1461
              5B 00000000'     00  9E 00002       MOVAB   NML$T_ENTBUFFER, R11                             ;
              5E               08  C2 00009       SUBL2   #8, SP                                            ;
                               58  D4 0000C       CLRL    STRTFLG                                           ; 1490
                        4100   8F  BB 0000E 1$:   PUSHR   #^M<R8,SP>                                        ; 1491
                               7E  D4 00012       CLRL    -(SP)                                             ;
              7E               01  CE 00014       MNEGL   #1, -(SP)                                         ;
                          04   AC  DD 00017       PUSHL   ENT                                               ;
       00000000G 00            05  FB 0001A       CALLS   #5, NML$GET_ENTITY_IDS                            ;
              44               50  E9 00021       BLBC    R0, 3$                                            ;
              58               01  D0 00024       MOVL    #1, STRTFLG                                       ; 1494
              5A               6E  3C 00027       MOVZWL  LISDSC, BUFEND                                    ; 1496
              5A          04   AE  C0 0002A       ADDL2   LISDSC+4, BUFEND                                  ;
              56          04   AE  D0 0002E       MOVL    LISDSC+4, PTR                                     ; 1497
              5A               56  D1 00032 2$:   CMPL    PTR, BUFEND                                       ; 1499
              D7               1E  00035          BGEQU   1$                                                ;
              53               6B  9E 00037       MOVAB   NML$T_ENTBUFFER, ENTIDPTR                         ; 1502
         40   AB          40   8F  9B 0003A       MOVZBW  #64, NML$Q_ENTBFDSC                               ; 1503
              57               86  3C 0003F       MOVZWL  (PTR)+, ENTLEN                                    ; 1507
              59               56  D0 00042       MOVL    PTR, ENTADD                                       ; 1509
              83               57  90 00045       MOVB    ENTLEN, (ENTIDPTR)+                               ; 1510
      63      69               57  28 00048       MOVC3   ENTLEN, (ENTADD), (ENTIDPTR)                      ; 1513
              56               57  C0 0004C       ADDL2   ENTLEN, PTR                                       ; 1514
              50               6B  9E 0004F       MOVAB   NML$T_ENTBUFFER, R0                               ; 1516
   40   AB    53               50  A3 00052       SUBW3   R0, ENTIDPTR, NML$Q_ENTBFDSC                      ;
                          08   AC  DD 00057       PUSHL   FCN                                               ; 1520
                        0280   8F  BB 0005A       PUSHR   #^M<R7,R9>                                        ;
                          04   AC  DD 0005E       PUSHL   ENT                                               ;
           FE51   CF           04  FB 00061       CALLS   #4, NML_V2_CHG_LINE                               ;
                               CA  11 00066       BRB     2$                                                ; 1499
                                   04 00068 3$:   RET                                                       ; 1523

; Routine Size:  105 bytes,    Routine Base:  $CODE$ + 07B5


; 1542          1524  1
```

```
; 1544          1525  1 END                                   . End of module
; 1545          1526  0 ELUDOM
```

```
;                                   PSECT SUMMARY

;       Name                    Bytes                           Attributes

;   $OWN$                         420  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;   $PLIT$                        212  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;   $CODE$                       2078  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;   .  ABS  .                       0  NOVEC,NOWRT,NORD ,NOEXE,NOSHR,  LCL,  ABS,  CON,NOPIC,ALIGN(0)


;                               Library Statistics

;                                        -------- Symbols --------      Pages       Processing
;       File                             Total    Loaded   Percent      Mapped      Time

;   _$255$DUA28:[NML.OBJ]NMLLIB.L32;1      341        56        16          27        00:00.1
;   _$255$DUA28:[SHRLIB]NMALIBRY.L32;1     887        31         3          47        00:00.2
;   _$255$DUA28:[SYSLIB]STARLET.L32;1     9776         2         0         581        00:02.2
;   _$255$DUA28:[SHRLIB]NET.L32;1         1279        27         2          63        00:01.0
```

```
;                                   COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:NMLV2COMP/OBJ=OBJ$:NMLV2COMP MSRC$:NMLV2COMP/UPDATE=(ENH$:NMLV2COMP)

; Size:          2078 code + 632 data bytes
; Run Time:         00:40.9
; Elapsed Time:     01:20.4
; Lines/CPU Min:     2238
; Lexemes/CPU-Min: 15895
; Memory Used:  174 pages
; Compilation Complete
```

NML_SHOW
LIS

NML_UPGRAD
LIS

NML_V2COMP
LIS

NML_UTIL
LIS

NPARSE
LIS

REQUEST
MAP

OPCOM

REPLY
MAP

NMLXFER
LIS

OPCCRASH
MAP

NMLV2STA
LIS

OPCOMLIB
B32

ALLOCATE
LIS

NMLZERO
LIS

OPCOMDEF
SDL

OPCOM
MAP

OPCDEFTMP
SDL

CANCEL
LIS