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

FILEID**NMLLIB

```
NN       NN  MM       MM  LL           LL           IIIIII    BBBBBBBB
NN       NN  MM       MM  LL           LL           IIIIII    BBBBBBBB
NN       NN  MMMM   MMMM  LL           LL             II      BB      BB
NN       NN  MMMM   MMMM  LL           LL             II      BB      BB
NNNN     NN  MM  MM  MM  LL           LL             II      BB      BB
NNNN     NN  MM  MM  MM  LL           LL             II      BB      BB
NN  NN   NN  MM      MM  LL           LL             II      BBBBBBBB
NN  NN   NN  MM      MM  LL           LL             II      BBBBBBBB
NN    NNNN  MM      MM  LL           LL             II      BB      BB
NN    NNNN  MM      MM  LL           LL             II      BB      BB
NN      NN  MM      MM  LL           LL             II      BB      BB      ....
NN      NN  MM      MM  LL           LL             II      BB      BB      ....
NN      NN  MM      MM  LLLLLLLLLL   LLLLLLLLLL   IIIIII    BBBBBBBB        ....
NN      NN  MM      MM  LLLLLLLLLL   LLLLLLLLLL   IIIIII    BBBBBBBB        ....

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

```
;  0001  0     !
;  0002  0     ! Version:      'V04-000'
;  0003  0     !
;  0004  0     !****************************************************************************
;  0005  0     !*                                                                          *
;  0006  0     !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
;  0007  0     !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
;  0008  0     !*  ALL RIGHTS RESERVED.                                                    *
;  0009  0     !*                                                                          *
;  0010  0     !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
;  0011  0     !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
;  0012  0     !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
;  0013  0     !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
;  0014  0     !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
;  0015  0     !*  TRANSFERRED.                                                            *
;  0016  0     !*                                                                          *
;  0017  0     !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
;  0018  0     !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
;  0019  0     !*  CORPORATION.                                                            *
;  0020  0     !*                                                                          *
;  0021  0     !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
;  0022  0     !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
;  0023  0     !*                                                                          *
;  0024  0     !*                                                                          *
;  0025  0     !****************************************************************************
;  0026  0     !
;  0027  0     !++
;  0028  0     !
;  0029  0     ! NMAHEAD.B32
;  0030  0     !
;  0031  0     !       Define $EQULST macro to make library from the NMALIBRY.B32 file
;  0032  0     !
;  0033  0     !       This source is taken from the following source:
;  0034  0     !--
;  0035  0     !++
;  0036  0     !
;  0037  0     ! UTLDEF.B32 - UTILITY DEFINITION MACROS FOR BLISS PROCESSING
;  0038  0     ! OF STARLET DEFINITION MACROS.
;  0039  0     !
;  0040  0     !--
;  0041  0     !
;  0042  0     !
;  0043  0     !
;  0044  0     ! MACRO TO GENERATE EQULST CONSTRUCTS.
;  0045  0     !
;  0046  0     MACRO
;M 0047  0             $EQULST(P,G,I,S)[A]=
;M 0048  0                 %NAME(P,GET1ST_ A) =
;M 0049  0                     %IF NUL2ND_ A
;M 0050  0                     %THEN (I) : %COUNT*(S)   ! ASSUMES I, S ALWAYS GENERATED BY CONVERSION PROGRAM
;M 0051  0                     %ELSE GET2ND_ A
;  0052  0                     %FI %,
;  0053  0
;M 0054  0             GET1ST_(A,B)=
;  0055  0                  A %,
;M 0056  0             GET2ND_(A,B)=
;  0057  0                  B %,       ! KNOWN NON-NULL
```

```
;  M 0058  0               NUL2ND (A,B)=
;    0059  0                     %NULL(B) %;
;    0060  0
;    0061  0     !
;    0062  0     !           End of NMAHEAD
;    0063  0     !
```

```
;    0064  0       ! NMLDEF.MDL - internal definitions for NML
;    0065  0       !
;    0066  0       !       Version 'V04-000'
;    0067  0       !
;    0068  0       !****************************************************************************
;    0069  0       !*                                                                          *
;    0070  0       !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
;    0071  0       !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
;    0072  0       !*  ALL RIGHTS RESERVED.                                                    *
;    0073  0       !*                                                                          *
;    0074  0       !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
;    0075  0       !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
;    0076  0       !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
;    0077  0       !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
;    0078  0       !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
;    0079  0       !*  TRANSFERRED.                                                            *
;    0080  0       !*                                                                          *
;    0081  0       !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
;    0082  0       !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
;    0083  0       !*  CORPORATION.                                                            *
;    0084  0       !*                                                                          *
;    0085  0       !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
;    0086  0       !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
;    0087  0       !*                                                                          *
;    0088  0       !*                                                                          *
;    0089  0       !****************************************************************************
;    0090  0
;    0091  0       !++
;    0092  0       !
;    0093  0       ! FACILITY:  VAX/VMS DECnet Network Management Listener
;    0094  0       !
;    0095  0       ! ABSTRACT:
;    0096  0       !
;    0097  0       !       This file contains varions MDL definitions for NML.
;    0098  0       !
;    0099  0       ! ENVIRONMENT:
;    0100  0       !
;    0101  0       !       n/a
;    0102  0       !
;    0103  0       ! AUTHOR:  Scott Davidson       CREATION DATE:  1-Aug-1979
;    0104  0       !
;    0105  0       ! MODIFICATION HISTORY:
;    0106  0       !
;    0107  0       !       V03-011 MKP0016         Kathy Perko     25 Mar-1984
;    0108  0       !               Add node parameter parsing constants.
;    0109  0       !
;    0110  0       !       V03-010 MKP0015         Kathy Perko     6-Jan-1984
;    0111  0       !               Add X25 Access Module entity.
;    0112  0       !
;    0113  0       !       V03-009 MKP0014         Kathy Perko     7-July-1983
;    0114  0       !               Add definitions for ISAM node permanent database with
;    0115  0       !               four ISAM keys.
;    0116  0       !
;    0117  0       !       V03-008 MKP0013         Kathy Perko     8-May-1983
;    0118  0       !               Coordinate NML$LOG bits in NML and MOM.
;    0119  0       !
;    0120  0       !       V03-007 MKP0012         Kathy Perko     24-April-1983
```

```
0121  0          !               Delete service functions from NML.  Also, add
0122  0          !               support for circuit service substate and service
0123  0          !               physical address.
0124  0          !
0125  0          !       V03-006 MKP0011         Kathy Perko     17-Jan-1983
0126  0          !               Support for NI Configurator Module.
0127  0          !
0128  0          !       V03-005 MKP0010         Kathy Perko     18-Nov-1982
0129  0          !               Add module as a source for logging events.
0130  0          !               Add Phase IV.
0131  0          !
0132  0          !       V03-004 MKP0009         Kathy Perko     27-Sept-1982
0133  0          !               Add Area and Adjacency entities.
0134  0          !
0135  0          !       V03-003 MKP0008         Kathy Perko     21-Sept-1982
0136  0          !               Add dispatch table definitions for ZERO.
0137  0          !
0138  0          !       V03-002 MKP0007         Kathy Perko     22-June-1982
0139  0          !               Set up "active network" for X-25 Protocol networks and
0140  0          !               enhance Entity Table (in NMLDAT) to include ACTIVE and
0141  0          !               KNOWN search key IDs, lengths, values, and operators.
0142  0          !               Add X29-Server entity.
0143  0          !
0144  0          !       V03-001 MKP0006         Kathy Perko     17-Mar-1982
0145  0          !               Add NML entity codes and permanent database keys for
0146  0          !               X-25 Access and Server Modules.
0147  0          !
0148  0          !       V02-006 MKP0005         Kathy Perko     20-Nov-1981
0149  0          !               Add parameter grouping flags for X.25 Protocol DTEs,
0150  0          !               Groups, and Networks.
0151  0          !
0152  0          !       V02-005 MKP0004         Kathy Perko     17-Nov-1981
0153  0          !               Add circuits to event source block definitions.
0154  0          !
0155  0          !       V02-004 MKP0003         Kathy Perko     3-Nov-1981
0156  0          !               Add field to MSB for second line of message text
0157  0          !               for RMS signalled messages.
0158  0          !
0159  0          !       V02-003 MKP0002         Kathy Perko     14-Oct-1981
0160  0          !               Add flags for parsing V2 to V3 SET LINE conversions.
0161  0          !
0162  0          !       V02-002 MKP0001         Kathy Perko     18-Sept-1981
0163  0          !               Change network management version number to 3.0.0
0164  0          !
0165  0          !--
```

```
;    0166  0
;    0167  0
;    0168  0       !
;    0169  0       ! Symbol definitions for the Network Management Listener
;    0170  0       !
;    0171  0
;    0172  0       !...$NMLDEF
;    0173  0            !
;    0174  0            ! Message parsing flag bit definitions (set in NML$GL_PRS_FLGS)
;    0175  0            !
;    0176  0
;    0177  0            !
;    0178  0            ! Common parsing flags.
;    0179  0            !
;    0180  0
;    0181  0       MACRO           NML$V_PRS_VMS    = 0,0,1,0%;                    ! VMS specific function
;    0182  0       LITERAL         NML$M_PRS_VMS    = 1^1 - 1^0;
;    0183  0       MACRO           NML$V_PRS_ALL    = 0,1,1,0%;                    ! Set/clear all parameters
;    0184  0       LITERAL         NML$M_PRS_ALL    = 1^2 - 1^1;
;    0185  0       MACRO           NML$V_PRS_QUALIFIER     = 0,2,1,0%;           ! NICE command includes a qualifier
;    0186  0       LITERAL         NML$M_PRS_QUALIFIER     = 1^3 - 1^2;
;    0187  0       MACRO           NML$V_PRS_ENTITY_FOUND  = 0,3,1,0%;    ! On multiple entity operations, at least one
;    0188  0       LITERAL         NML$M_PRS_ENTITY_FOUND  = 1^4 - 1^3;
;    0189  0                                               !       entity has been set, shown, etc.
;    0190  0
;    0191  0            !
;    0192  0            ! Specific entity parsing flags.
;    0193  0            !
;    0194  0            !
;    0195  0            ! Node parsing flags.
;    0196  0            !
;    0197  0
;    0198  0
;    0199  0       MACRO           NML$V_PRS_EXEPG = 0,8,1,0%;             ! Executor-only parameter group
;    0200  0       LITERAL         NML$M_PRS_EXEPG = 1^9 - 1^8;
;    0201  0       MACRO           NML$V_PRS_NODPG = 0,9,1,0%;             ! Executor/node parameter group
;    0202  0       LITERAL         NML$M_PRS_NODPG = 1^10 - 1^9;
;    0203  0       MACRO           NML$V_PRS_REMPG = 0,10,1,0%;            ! Remote node parameter group
;    0204  0       LITERAL         NML$M_PRS_REMPG = 1^11 - 1^10;
;    0205  0       MACRO           NML$V_PRS_LOOPG = 0,11,1,0%;            ! Loop node parameter group
;    0206  0       LITERAL         NML$M_PRS_LOOPG = 1^12 - 1^11;
;    0207  0
;    0208  0            !
;    0209  0            ! Logging parsing flags.
;    0210  0            !
;    0211  0
;    0212  0
;    0213  0       MACRO           NML$V_PRS_EXESNK        = 0,8,1,0%;             ! Sink node is executor
;    0214  0       LITERAL         NML$M_PRS_EXESNK        = 1^9 - 1^8;
;    0215  0       MACRO           NML$V_PRS_SNKNOD        = 0,9,1,0%;             ! Sink node specified
;    0216  0       LITERAL         NML$M_PRS_SNKNOD        = 1^10 - 1^9;
;    0217  0       MACRO           NML$V_PRS_KNOSNK        = 0,10,1,0%;            ! Known sink nodes
;    0218  0       LITERAL         NML$M_PRS_KNOSNK        = 1^11 - 1^10;
;    0219  0       MACRO           NML$V_PRS_EFIPG = 0,11,1,0%;            ! Event filter parameter group
;    0220  0       LITERAL         NML$M_PRS_EFIPG = 1^12 - 1^11;
;    0221  0       MACRO           NML$V_PRS_ESIPG = 0,12,1,0%;            ! Event sink parameter group
;    0222  0       LITERAL         NML$M_PRS_ESIPG = 1^13 - 1^12;
```

```
;    0223  0      MACRO           NML$V_PRS_EVE   = 0,13,1,0%;                    ! Event parameter processed
;    0224  0      LITERAL         NML$M_PRS_EVE   = 1^14 - 1^13;
;    0225  0
;    0226  0          !
;    0227  0          ! Line parsing flags.
;    0228  0          !
;    0229  0
;    0230  0
;    0231  0      MACRO           NML$V_PRS_LINE  = 0,8,6,0%;             ! Line flags (LIN$ definitions)
;    0232  0      LITERAL         NML$M_PRS_LINE  = 1^14 - 1^8;
;    0233  0      MACRO           NML$V_PRS_V2_LINE       = 0,14,1,0%;        ! Command parameters are for a line.
;    0234  0      LITERAL         NML$M_PRS_V2_LINE       = 1^15 - 1^14;
;    0235  0      MACRO           NML$V_PRS_V2_CIRCUIT    = 0,15,1,0%;        ! Command parameters are for a circuit.
;    0236  0      LITERAL         NML$M_PRS_V2_CIRCUIT    = 1^16 - 1^15;
;    0237  0      MACRO           NML$V_PRS_V2_STA        = 0,16,1,0%;        ! Command contains a state change.
;    0238  0      LITERAL         NML$M_PRS_V2_STA        = 1^17 - 1^16;
;    0239  0
;    0240  0          !
;    0241  0          ! NICE message parsing constants
;    0242  0          !
;    0243  0      LITERAL
;  P 0244  0      $EQULST (NML$C_,GBL,0,1
;  P 0245  0              ,(NODE_NUM_PARAM, 0)                ! Parameter is always a node number
;  P 0246  0              ,(NODE_ID_PARAM,  1)                ! Parameter can be a node number or name
;    0247  0              );
;    0248  0          !
;    0249  0          ! Protocol DTE parsing flags.
;    0250  0          !
;    0251  0
;    0252  0
;    0253  0      MACRO           NML$V_PRS_CHANNELS      = 0,8,1,0%;         ! Processing SET channels - first channel
;    0254  0      LITERAL         NML$M_PRS_CHANNELS      = 1^9 - 1^8;
;    0255  0                                  !           pair already encountered.
;    0256  0
```

```
;  0257  0
;  0258  0
;  0259  0          !
;  0260  0          ! Network Management version definitions for message handling
;  0261  0          ! (Set in NML$GB_CMD_VER)
;  0262  0          !
;  0263  0      LITERAL
;P 0264  0      $EQULST (NML$C_,GBL,0,1
;P 0265  0              ,(PHASE2, 1)                    ! Phase II function
;P 0266  0              ,(PHASE3_OR_4, 2)                       ! Phase III or IV function
;  0267  0              );
;  0268  0          !
;  0269  0          ! NML return codes
;  0270  0          !
;  0271  0      LITERAL
;P 0272  0      $EQULST (NML$_,GBL,0,1
;P 0273  0              ,(STS_SUC, 1)                   ! Success
;P 0274  0              ,(STS_FUN, -1*2)                        ! Unrecognized function or option
;P 0275  0              ,(STS_INV, -2*2)                        ! Invalid message format
;P 0276  0              ,(STS_PRI, -3*2)                        ! Privilege violation
;P 0277  0              ,(STS_SIZ, -4*2)                        ! Message too long
;P 0278  0              ,(STS_MPR, -5*2)                        ! Network management program error
;P 0279  0              ,(STS_PTY, -6*2)                        ! Unrecognized parameter type
;P 0280  0              ,(STS_MVE, -7*2)                        ! Incompatible management version
;P 0281  0              ,(STS_CMP, -8*2)                        ! Unrecognized component
;P 0282  0              ,(STS_IDE, -9*2)                        ! Invalid identification format
;P 0283  0              ,(STS_LCO, -10*2)                       ! Line communication error
;P 0284  0              ,(STS_STA, -11*2)                       ! Component in wrong state
;P 0285  0              ,(STS_FOP, -13*2)                       ! File open error
;P 0286  0              ,(STS_FCO, -14*2)                       ! Invalid file contents
;P 0287  0              ,(STS_RES, -15*2)                       ! Resource error
;P 0288  0              ,(STS_PVA, -16*2)                       ! Invalid parameter value
;P 0289  0              ,(STS_LPR, -17*2)                       ! Line protocol error
;P 0290  0              ,(STS_FIO, -18*2)                       ! File i/o error
;P 0291  0              ,(STS_MLD, -19*2)                       ! Mirror link disconnected
;P 0292  0              ,(STS_ROO, -20*2)                       ! No room for new entry
;P 0293  0              ,(STS_MCF, -21*2)                       ! Mirror connect failed
;P 0294  0              ,(STS_PNA, -22*2)                       ! Parameter not applicable
;P 0295  0              ,(STS_PLO, -23*2)                       ! Parameter value too long
;P 0296  0              ,(STS_HAR, -24*2)                       ! Hardware failure
;P 0297  0              ,(STS_OPE, -25*2)                       ! Operation failure
;P 0298  0              ,(STS_SYS, -26*2)                       ! System-specific network management function not supported
;P 0299  0              ,(STS_PGP, -27*2)                       ! Invalid parameter grouping
;P 0300  0              ,(STS_BLR, -28*2)                       ! Bad loopback response
;P 0301  0              ,(STS_PMS, -29*2)                       ! Parameter missing
;  0302  0              );
;  0303  0          !
;  0304  0          ! Network Management parameters
;  0305  0          !
;  0306  0      LITERAL
;P 0307  0      $EQULST (NML$K_,GBL,0,1
;P 0308  0              ,(VERSION, 4)                   ! Network Management version
;P 0309  0              ,(DEC_ECO, 0)                   ! DIGITAL ECO number
;P 0310  0              ,(USER_ECO, 0)                  ! User ECO number
;P 0311  0              ,(FAC_CODE, 505)                        ! Facility code
;P 0312  0              ,(SIG_CODE, 505*65536)          ! Signal code (505^16)
;  0313  0              );
```

```
;  0314  0          !
;  0315  0          ! Data definitions
;  0316  0          !
;  0317  0      LITERAL
;P 0318  0      $EQULST (NML$K_,GBL,0,1
;P 0319  0              ,(BYTE,)                                . Byte
;P 0320  0              ,(WORD,)                                . Word
;P 0321  0              ,(LONG,)                                . Longword
;P 0322  0      !*      QUAD                    ! Quadword
;P 0323  0              ,(STRING,)                              ! String
;  0324  0              );
;  0325  0          !
;  0326  0          ! Special permanent data base record search key parameter codes.
;  0327  0          !
;  0328  0      LITERAL
;P 0329  0      $EQULST (NML$C_,GBL,0,1
;P 0330  0              ,(KEY_LINE, -1)                         ! Line
;P 0331  0              ,(KEY_SINK, -2)                         ! Logging sink
;P 0332  0                                      ! Node uses name or address
;P 0333  0              ,(KEY_EXE,  -3)                         ! Executor node
;P 0334  0                                      ! Object uses name
;P 0335  0              ,(KEY_CIR,  -4)                         ! Circuit
;P 0336  0              ,(KEY_NET,      -5)             ! Protocol Module Network
;P 0337  0              ,(KEY_X25_SERV, -6)             ! X25-Server Module
;P 0338  0              ,(KEY_X29_SERV, -7)             ! X29-Server Module
;P 0339  0              ,(KEY_TRACE,    -8)             ! X25-Trace Module
;P 0340  0              ,(KEY_NI_CONFG, -9)             ! NI Configurator Module
;P 0341  0              ,(KEY_X25_ACCESS,       -10)            ! X25 Access Module Network
;  0342  0              );
;  0343  0
;  0344  0
;  0345  0          !
;  0346  0          ! Internal entity id codes.  Do not reorder.  Entries in
;  0347  0          ! NML$AB_ENTITYDATA table depend on this order.
;  0348  0          !
;  0349  0      LITERAL
;P 0350  0      $EQULST (NML$C_,GBL,0,1
;P 0351  0              ,(LINE,)                                ! Line
;P 0352  0              ,(LOGGING,)                             ! Logging
;P 0353  0              ,(SINK,)                                ! Logging sink
;P 0354  0              ,(NODE,)                                ! Node (by address)
;P 0355  0              ,(NODEBYNAME,)                  ! Node by name
;P 0356  0              ,(LOOPNODE,)                    ! Loop node (by name only)
;P 0357  0              ,(ADJACENT_NODE,)                       ! Nodes one hop away, volatile
;P 0358  0                                      !       database only
;P 0359  0              ,(EXECUTOR,)                    ! Executor node (by address=0)
;P 0360  0              ,(OBJECT,)                              ! Object
;P 0361  0              ,(CIRCUIT,)                             ! Circuit
;P 0362  0              ,(CIRCUIT_ADJACENT,)            ! By circuit, Nodes one hop away,
;P 0363  0                                      !       volatile database only.
;P 0364  0              ,(CIRCUIT_ADJ_SRV,)                     ! By circuit, service adjacencies
;P 0365  0              ,(AREA,)                                ! Area, volatile database only.
;P 0366  0              ,(X25_ACCESS,)                  ! X25-Access Module
;P 0367  0              ,(PROT_NET,)                    ! X25-Protocol Module Networks
;P 0368  0              ,(PROT_DTE,)                    ! X25-Protocol Module DTEs
;P 0369  0              ,(PROT_GRP,)                    ! X25-Protocol Module Groups
;P 0370  0              ,(X25_SERV,)                    ! X25-Server Module
```

```
;P 0371  0               ,(X25_SERV_DEST,)                       ! X25-Server Module Destination
;P 0372  0               ,(TRACE,)                               ! X25-Trace Module
;P 0373  0               ,(TRACEPNT,)                    ! X25-Trace Module Trecepoint
;P 0374  0               ,(X29_SERV,)                    ! X29-Server Module
;P 0375  0               ,(X29_SERV_DEST,)                       ! X29-Server Module Destination
;P 0376  0               ,(NI_CONFIG,)                   ! NI Configurator Module
;P 0377  0               ,(LINKS,)                               ! Logical links, volatile database only.
;P 0378  0               ,(PROTOCOL,)                    ! Protocol Module
;P 0379  0
;P 0380  0               ,(MAXENTITY,)                   ! Maximum entity number
;   0381  0               );
;   0382  0
;   0383  0           ! Internal information table index codes.
;   0384  0           !
;   0385  0       LITERAL
;P 0386  0       $EQULST (NML$C_,GBL,0,1
;P 0387  0               ,(SUMMARY,)                             ! Summary          (NMA$C_INF_SUM)
;P 0388  0               ,(STATUS,)                              ! Status           (NMA$C_INF_STA)
;P 0389  0               ,(CHARACTERISTICS,)                     ! Characteristics (NMA$C_INF_CHA)
;P 0390  0               ,(COUNTERS,)                    ! Counters         (NMA$C_INF_COU)
;P 0391  0               ,(EVENTS,)                              ! Events           (NMA$C_INF_EVE)
;P 0392  0               ,(ZERO,)                                ! Zero counters
;P 0393  0               ,(SERVICE,)                             ! Service parameters
;P 0394  0
;P 0395  0               ,(MAXINFO,)                             ! Maximum information type
;   0396  0               );
;   0397  0
;   0398  0
```

```
;  0399  0
;  0400  0
;  0401  0     !
;  0402  0     ! Network Management Node database definitions.  Used for manipulating
;  0403  0     ! the node permanent database.  This database (unlike the other entity
;  0404  0     ! permanent databases) uses 4 ISAM keys.
;  0405  0     !
;  0406  0     !...$NMNDEF
;  0407  0     MACRO           NMN$L_KEY_LIS   = 0,0,32,0%;            ! 3rd alternate key = list node (consists of node
;  0408  0                                     !       address key concatenated with node
;  0409  0                                     !       type key)
;  0410  0     MACRO           NMN$W_KEY_ADD   = 0,0,16,0%;            ! Primary key in record = node address
;  0411  0     MACRO           NMN$W_KEY_TYP   = 2,0,16,0%;            ! 1st alternate key in record = node type (executor,
;  0412  0                                     !       remote, or loopnode).
;  0413  0     MACRO           NMN$T_KEY_NAM   = 4,0,0,0%;             ! 2nd alternate key in record = node name
;  0414  0     LITERAL         NMN$S_KEY_NAM   = 6;
;  0415  0     LITERAL         NMN$C_NODE_KEYS_LEN     = 10;
;  0416  0     LITERAL         NMN$K_NODE_KEYS_LEN     = 10;                   ! Length of all three keys.
;  0417  0     MACRO           NMN$A_NOD_PARAMS        = 10,0,32,0%;           ! Beginning of node's NICE parameters
;  0418  0                                     !       in the record.
;  0419  0
;  0420  0     LITERAL
;P 0421  0     $EQULST (NMN$C_,GBL,0,1
;P 0422  0             !
;P 0423  0             ! Keys for accessing node permanent database.
;P  424  0             !
;P 0425  0             ,(ADD_KEY_REF,  0)              ! The primary key = node address
;P 0426  0             ,(TYP_KEY_REF,  1)              ! The 1st alternate key = node type
;P 0427  0                                     !       (nml$c_typ_exec, nml$c_typ_remote,
;P 0428  0                                     !       nml$c_typ_loopnode).  Overlaps with
;P 0429  0                                     !       node address key.
;P 0430  0             ,(NAM_KEY_REF,  2)              ! The 2nd alternate key = node name
;P 0431  0             ,(LIS_KEY_REF,  3)              ! The 3rd alternate key = node address
;P 0432  0                                     !       concatenated with node type.  Used
;P 0433  0                                     !       to LIST nodes in order by address,
;P 0434  0                                     !       but with exec first, then remotes, and
;P 0435  0                                     !       last loopnodes.
;P 0436  0             !
;P 0437  0             ! Lengths of node permanent database keys
;P 0438  0             !
;P 0439  0             ,(ADD_KEY_LEN,  2)              ! The primary key = node address
;P 0440  0             ,(TYP_KEY_LEN,  2)              ! The first alternate key overlaps with the
;P 0441  0                                     !       node address key.
;P 0442  0             ,(NAM_KEY_LEN,  6)              ! The second alternate key = node name
;P 0443  0             ,(LIS_KEY_LEN,  4)              ! The third alternate key = node address
;P 0444  0                                     !       concatenated with node type.
;P 0445  0             !
;P 0446  0             ! Key values for node key = type.  The LIST key concatenates the node
;P 0447  0             ! address key with the node type key.  This allows the the LIST
;P 0448  0             ! command to get nodes by type and, within type, sequentially by node
;P 0449  0             ! address.  The key value is constructed with a zero for the node
;P 0450  0             ! address! hence when you do a $GET of (type OR 0) with a match type
;P 0451  0             ! of GTR, it will get the first node of that type in the file.
;P 0452  0             ! Subsequent sequential reads will return the nodes of that type in
;P 0453  0             ! ascending order by address.
;P 0454  0             !
;P 0455  0             ,(TYP_EXEC,     0)              ! type = executor node
```

```
;P 0456  0              ,(TYP_REMOTE,   1)              ! type = a remote node
;P 0457  0              ,(TYP_LOOPNODE, 2)              ! type = a loopnode
;P 0458  0
;P 0459  0              !
;P 0460  0              ! Input values to node database routines.  Used for determining
;P 0461  0              ! what RMS operations to perform.
;P 0462  0              !
;P 0463  0              ,(PUT_REC,      1)              ! Do a $PUT (write a new record)
;P 0464  0              ,(UPDATE_REC,   2)              ! Do a $UPDATE (update an existing record)
;P 0465  0              ,(DELETE_REC,   3)              ! Do a $DELETE (delete a record)
;P 0466  0              ,(GET_REC,      4)              ! Do a $GET (read a record)
;   0467  0              );
;   0468  0
```

```
;  0469  0
;  0470  0
;  0471  0       !
;  0472  0       ! Message segment block (MSB) definitions
;  0473  0       !
;  0474  0       !...$MSBDEF
;  0475  0
;  0476  0       MACRO           MSB$L_FLAGS     = 0,0,32,0%;                    ! Flags
;  0477  0
;  0478  0       MACRO           MSB$V_CODE_FLD  = 0,0,1,0%;             ! Status code present (not used)
;  0479  0       LITERAL         MSB$M_CODE_FLD  = 1^1 - 1^0;
;  0480  0       MACRO           MSB$V_DET_FLD   = 0,1,1,0%;                     ! Error detail field present (DETAIL)
;  0481  0       LITERAL         MSB$M_DET_FLD   = 1^2 - 1^1;
;  0482  0       MACRO           MSB$V_MSG_FLD   = 0,2,1,0%;                     ! Message text field present (TEXT)
;  0483  0       LITERAL         MSB$M_MSG_FLD   = 1^3 - 1^2;
;  0484  0       MACRO           MSB$V_MSG2_FLD  = 0,3,1,0%;             ! Second line of message text present (TEXT2)
;  0485  0       LITERAL         MSB$M_MSG2_FLD  = 1^4 - 1^3;
;  0486  0       MACRO           MSB$V_ENTD_FLD  = 0,4,1,0%;             ! Enitity descriptor field present (ENTITY)
;  0487  0       LITERAL         MSB$M_ENTD_FLD  = 1^5 - 1^4;
;  0488  0       MACRO           MSB$V_DATA_FLD  = 0,5,1,0%;             ! Data  descriptor field present (DATA)
;  0489  0       LITERAL         MSB$M_DATA_FLD  = 1^6 - 1^5;
;  0490  0       MACRO           MSB$V_SYSM_FLD  = 0,6,1,0%;             ! System message field present (TEXT)
;  0491  0       LITERAL         MSB$M_SYSM_FLD  = 1^7 - 1^6;
;  0492  0
;  0493  0       MACRO           MSB$B_CODE      = 4,0,8,0%;                     ! Status code
;  0494  0       MACRO           MSB$W_DETAIL    = 8,0,16,0%;            ! Detail
;  0495  0       MACRO           MSB$L_TEXT      = 12,0,32,0%;                   ! Status code for text message.
;  0496  0       MACRO           MSB$L_TEXT2     = 16,0,32,0%;                   ! Status code for second line of text msg.
;  0497  0       MACRO           MSB$A_ENTITY    = 20,0,32,0%;           ! Entity descriptor address
;  0498  0       MACRO           MSB$A_DATA      = 24,0,32,0%;                   ! Data descriptor address
;  0499  0       LITERAL         MSB$C_LENGTH    = 28;
;  0500  0       LITERAL         MSB$K_LENGTH    = 28;                           ! Maximum MSB size
;  0501  0
```

```
;   0502  0
;   0503  0
;   0504  0        !
;   0505  0        ! NML internal logging (debugging) flags
;   0506  0        !
;   0507  0        ! These flags are used to enable logging of specified data to the NML log
;   0508  0        ! file.  The flags are defined by translating the logical name NML$LOG.
;   0509  0        !
;   0510  0        !...$DBGDEF
;   0511  0
;   0512  0        LITERAL
;P  0513  0        $EQULST (DBG$C_,GBL,0,1
;P  0514  0                ,(NETIO,)                       ! Network send/receive logging
;P  0515  0                ,(FILEIO,)                      ! File read/write logging
;P  0516  0                ,(NPARSE,)                      ! NPARSE state transition logging
;P  0517  0                ,(LOOPIO,)                      ! Loopback transmit/receive logging
;P  0518  0                ,(ACPQIO,)                      ! NETACP QIO logging
;P  0519  0                ,(MOPIO,)                       ! MOP send/receive logging
;P  0520  0                ,(SRVTRC,)                      ! Trace service operations
;P  0521  0                ,(EVENTS,)                      ! Network event (EVL) logging
;   0522  0                );
;   0523  0
;   0524  0        LITERAL
;P  0525  0        $EQULST (DBG$C_,GBL,16,1
;P  0526  0                ,(DMPNOD,)                      ! Dump node permanent data base file
;P  0527  0                ,(DMPLIN,)                      ! Dump line permanent data base file
;P  0528  0                ,(DMPLOG,)                      ! Dump logging permanent data base file
;P  0529  0                ,(DMPOBJ,)                      ! Dump object permanent data base file
;P  0530  0                ,(DMPCIR,)                      ! Dump circuit permanent data base file
;P  0531  0                ,(DMPX25,)                      ! Dump X25 module permanent data base file
;P  0532  0                ,(DMPX29,)                      ! Dump X29 module permanent data base file
;P  0533  0                ,(DMPCNF,)                      ! Dump Configurator Module permanent data base file
;   0534  0                );
;   0535  0
```

```
;    0536  0
;    0537  0
;    0538  0        !
;    0539  0        ! Parameter semantic table (PST) definitions
;    0540  0        !
;    0541  0        !...$PSTDEF
;    0542  0
;    0543  0        MACRO           PST$W_DATAID    = 0,0,16,0%;    ! DNA parameter code
;    0544  0        MACRO           PST$B_FORMAT    = 2,0,8,0%;     ! Parameter format (byte, word, longword, etc.)
;    0545  0        MACRO           PST$B_DATATYPE  = 3,0,8,0%;     ! Data type code (coded, coded multiple, etc.)
;    0546  0        MACRO           PST$L_MINVALUE  = 4,0,32,0%;    ! Minimum value or string length
;    0547  0        MACRO           PST$L_MAXVALUE  = 8,0,32,0%;    ! Maximum value or string length
;    0548  0        MACRO           PST$L_NFBID     = 12,0,32,0%;           ! ACP parameter identifier
;    0549  0        LITERAL         PST$C_ENTRYLEN  = 16;
;    0550  0        LITERAL         PST$K_ENTRYLEN  = 16;           ! Parameter semantic table entry length
;    0551  0
```

```
;    0552  0
;    0553  0
;    0554  0
;    0555  0       !
;    0556  0       ! Entity information table definitions.
;    0557  0       !
;    0558  0       !...$EITDEF
;    0559  0
;    0560  0       MACRO           EIT$B_FILEID    = 0,0,8,0%;             ! Permanent data base file id code
;    0561  0       MACRO           EIT$W_DETAIL    = 1,0,16,0%;            ! NICE error detail entity code
;    0562  0       MACRO           EIT$W_KEY       = 3,0,16,0%;            ! Permanent data base search key
;    0563  0       MACRO           EIT$B_DATABASE  = 5,0,8,0%;     ! Volatile data base ID
;    0564  0       MACRO           EIT$L_SRCH_ID1  = 6,0,32,0%;    ! Volatile data base search key one ID for one entity
;    0565  0       MACRO           EIT$L_SRCH_ID2  = 10,0,32,0%;   ! Volatile data base search key two ID for one entity
;    0566  0       !
;    0567  0       ! SHOW KNOWN search key ID, length, value, and operator.
;    0568  0       !
;    0569  0       MACRO           EIT$L_KNO_SRCH_ID1      = 14,0,32,0%;   ! Search key one ID
;    0570  0       MACRO           EIT$L_KNO_SRCH_LEN1     = 18,0,32,0%;   ! Search key one length
;    0571  0       MACRO           EIT$L_KNO_SRCH_VAL1     = 22,0,32,0%;   ! Search key one value
;    0572  0       MACRO           EIT$B_KNO_OPERT = 26,0,8,0%;    ! Sense search one operator (EQL, NEQ, etc.)
;    0573  0
;    0574  0       !
;    0575  0       ! SHOW ACTIVE search key ID, length, value, and operator.
;    0576  0       !
;    0577  0       MACRO           EIT$L_ACT_SRCH_ID1      = 27,0,32,0%;   ! Search key one ID
;    0578  0       MACRO           EIT$L_ACT_SRCH_LEN1     = 31,0,32,0%;   ! Search key one length
;    0579  0       MACRO           EIT$L_ACT_SRCH_VAL1     = 35,0,32,0%;   ! Search key one value
;    0580  0       MACRO           EIT$B_ACT_OPERT = 39,0,8,0%;    ! Sense of search one operator (EQL, NEQ, etc.)
;    0581  0
;    0582  0       MACRO           EIT$A_ALLTAB    = 40,0,32,0%;           ! Parameter table for SET ALL
;    0583  0       LITERAL         EIT$C_ENTRYLEN  = 44;
;    0584  0       LITERAL         EIT$K_ENTRYLEN  = 44;                   ! Entry length
;    0585  0
;    0586  0
;    0587  0       !
;    0588  0       ! Change parameter table definitions.
;    0589  0       !
;    0590  0       !...$CPTDEF
;    0591  0
;    0592  0       MACRO           CPT$W_PSTINDEX  = 0,0,16,0%;            ! Parameter semantic table index
;    0593  0       MACRO           CPT$A_DEFINE_RTN        = 2,0,32,0%;            ! Define routine address
;    0594  0       MACRO           CPT$A_PURGE_RTN = 6,0,32,0%;            ! Purge routine address
;    0595  0       !   F SET_RTN,A                     ! Set routine address
;    0596  0       !   F CLEAR_RTN,A               ! Clear routine address
;    0597  0       LITERAL         CPT$C_ENTRYLEN  = 10;
;    0598  0       LITERAL         CPT$K_ENTRYLEN  = 10;                           ! Length of table entry
;    0599  0
```

```
;  0600  0
;  0601  0
;  0602  0     !
;  0603  0     ! Change dispatch table definitions.  This table is used by NMLCHANGE or
;  0604  0     ! NMLREAD to dispatch to the correct change routine for the entity.
;  0605  0     !
;  0606  0     !...$DTDEF
;  0607  0
;  0608  0     MACRO           DT$L_DISPATCH   = 0,0,32,0%;            ! Dispatch routine for this entity
;  0609  0
;  0610  0     !
;  0611  0     ! Dispatch table definitions for NICE change operations.
;  0612  0     !
;  0613  0     MACRO           DT$L_SEDE_PARSE = 4,0,32,0%;            ! Set/Define NPARSE table for parsing the
;  0614  0                                     !       NICE command parameters
;  0615  0     MACRO           DT$L_CLPU_PARSE = 8,0,32,0%;            ! Clear/Purge NPARSE table for parsing the
;  0616  0                                     !       NICE command parameters.
;  0617  0     MACRO           DT$A_SET_ROUTINES       = 12,0,0,0%;            ! SET routines for entity.
;  0618  0     LITERAL         DT$S_SET_ROUTINES       = 16;
;  0619  0     MACRO           DT$A_CLEAR_ROUTINES     = 28,0,0,0%;    ! CLEAR routines for entity.
;  0620  0     LITERAL         DT$S_CLEAR_ROUTINES     = 16;
;  0621  0     MACRO           DT$A_DEFINE_ROUTINES    = 44,0,0,0%;    ! DEFINE routines for entity.
;  0622  0     LITERAL         DT$S_DEFINE_ROUTINES    = 16;
;  0623  0     MACRO           DT$A_PURGE_ROUTINES     = 60,0,0,0%;    ! PURGE routines for entity.
;  0624  0     LITERAL         DT$S_PURGE_ROUTINES     = 16;
;  0625  0
;  0626  0
;  0627  0     !
;  0628  0     ! Dispatch table definitions for NICE read operations.
;  0629  0     !
;  0630  0     MACRO           DT$A_SHOW_ROUTINES      = 4,0,0,0%;             ! SHOW routines for entity.
;  0631  0     LITERAL         DT$S_SHOW_ROUTINES      = 20;
;  0632  0     MACRO           DT$A_LIST_ROUTINES      = 24,0,0,0%;            ! LIST routines for entity
;  0633  0     LITERAL         DT$S_LIST_ROUTINES      = 20;
;  0634  0
;  0635  0
;  0636  0
;  0637  0     !...$CHGDEF
;  0638  0          !
;  0639  0          ! Each of the DT$ change ROUTINES fields breaks down into the following
;  0640  0          ! routine offsets.  They are offsets in order to make NMLSHR PIC.
;  0641  0          !
;  0642  0     MACRO           CHG$L_ENTITY    = 0,0,32,0%;                    ! Offset to change routine for single entity
;  0643  0     MACRO           CHG$L_ENTITY_W_QUAL     = 4,0,32,0%;            ! Offset to change routine for single entity
;  0644  0                                     !       with a qualifier
;  0645  0     MACRO           CHG$L_KNOWN     = 8,0,32,0%;                    ! Offset to change routine for KNOWN entities.
;  0646  0     MACRO           CHG$L_KNOWN_W_QUAL      = 12,0,32,0%;           ! Offset to change routine for KNOWN entities
;  0647  0                                     !       with qualifier
;  0648  0     LITERAL         CHG$C_CHG_TABLEN        = 16;
;  0649  0     LITERAL         CHG$K_CHG_TABLEN        = 16;                   ! Length of dispatch table
;  0650  0
;  0651  0
;  0652  0
;  0653  0     !...$RDDEF
;  0654  0          !
;  0655  0          ! Each of the DT$ read ROUTINES fields breaks down into the following
;  0656  0          ! routine offsets.  They are offsets in order to make NMLSHR PIC.
```

```
;    0657  0          !
;    0658  0      MACRO           RD$L_ENTITY     = 0,0,32,0%;                    ! Offset to read routine for single entity
;    0659  0      MACRO           RD$L_ENTITY_W_QUAL      = 4,0,32,0%;            ! Offset to read routine for single entity
;    0660  0                                      !          with a qualifier
;    0661  0      MACRO           RD$L_KNOWN      = 8,0,32,0%;                    ! Offset to read routine for KNOWN entities.
;    0662  0      MACRO           RD$L_KNOWN_W_QUAL       = 12,0,32,0%;           ! Offset to read routine for KNOWN entities
;    0663  0                                      !          with qualifier
;    0664  0      MACRO           RD$L_ACTIVE     = 16,0,32,0%;                   ! Offset to read routine for ACTIVE entities.
;    0665  0      LITERAL         RD$C_RD_TABLEN  = 20;
;    0666  0      LITERAL         RD$K_RD_TABLEN  = 20;                           ! Length of dispatch table
;    0667  0
;    0668  0
;    0669  0
;    0670  0      !...$ZERDEF
;    0671  0          !
;    0672  0          ! Each of the ZER$ zero ROUTINES fields breaks down into the following
;    0673  0          ! routine offsets.  They are offsets in order to make NMLSHR PIC.
;    0674  0          !
;    0675  0      MACRO           ZER$L_DISPATCH  = 0,0,32,0%;            ! Dispatch routine for this entity
;    0676  0      MACRO           ZER$L_ENTITY    = 4,0,32,0%;                    ! Offset to zero routine for single entities.
;    0677  0      MACRO           ZER$L_KNOWN     = 8,0,32,0%;                    ! Offset to zero routine for KNOWN entities.
;    0678  0      LITERAL         ZER$C_ZER_TABLEN        = 12;
;    0679  0      LITERAL         ZER$K_ZER_TABLEN        = 12;                   ! Length of dispatch table.
;    0680  0
```

```
;  0681  0
;  0682  0
;  0683  0       !
;  0684  0       ! This file defines the event data base ures.
;  0685  0       !
;  0686  0       !
;  0687  0       ! Event source block definitions.
;  0688  0       !
;  0689  0       !...$SRCDEF
;  0690  0
;  0691  0       MACRO           SRC$W_LENGTH    = 0,0,16,0%;            ! Byte count of entire source block
;  0692  0       !
;  0693  0       MACRO           SRC$B_SINKTYPE  = 2,0,8,0%;             ! Sink type code:  NMA$C_SNK_CON
;  0694  0       !                                                               NMA$C_SNK_FIL
;  0695  0       !                                                               NMA$C_SNK_MON
;  0696  0       !
;  0697  0       MACRO           SRC$B_SRCTYPE   = 3,0,8,0%;             ! Source type code:  NMA$C_ENT_KNO
;  0698  0       !                                                         NMA$C_ENT_NOD
;  0699  0       !                                                         NMA$C_ENT_CIR
;  0700  0       !                                                         NMA$C_ENT_LIN
;  0701  0       !
;  0702  0       MACRO           SRC$T_SOURCE    = 4,0,0,0%;             ! Source id string
;  0703  0       LITERAL         SRC$S_SOURCE    = 18;
;  0704  0       !
;  0705  0       ! Source = node (NMA$C_ENT_NOD)
;  0706  0       MACRO           SRC$W_NODADR    = 4,0,16,0%;            !   Node address
;  0707  0       ! Source = line or circuit (NMA$C_ENT_LIN or
;  0708  0       !                           NMA$C_ENT_CIR)
;  0709  0       MACRO           SRC$B_IDLENGTH  = 4,0,8,0%;             !   Source id string length
;  0710  0       MACRO           SRC$T_ID        = 5,0,0,0%;             !   Source id string
;  0711  0       LITERAL         SRC$S_ID        = 17;
;  0712  0       !
;  0713  0       MACRO           SRC$W_MSKCOUNT  = 22,0,16,0%;           ! Count of event blocks (mask sets)
;  0714  0
;  0715  0       LITERAL         SRC$C_LENGTH    = 24;
;  0716  0       LITERAL         SRC$K_LENGTH    = 24;                           ! Length of fixed part of source block
;  0717  0
;  0718  0       !
;  0719  0       ! Event block definitions.
;  0720  0       !
;  0721  0       !...$EVTDEF
;  0722  0
;  0723  0       MACRO           EVT$W_CLASS     = 0,0,16,0%;                    ! Event class
;  0724  0       MACRO           EVT$Q_LOGMSK    = 4,0,0,0%;             ! Event log mask
;  0725  0       LITERAL         EVT$S_LOGMSK    = 8;
;  0726  0       MACRO           EVT$Q_FILTERMSK = 12,0,0,0%;            ! Event filter mask
;  0727  0       LITERAL         EVT$S_FILTERMSK = 8;
;  0728  0       LITERAL         EVT$C_LENGTH    = 20;
;  0729  0       LITERAL         EVT$K_LENGTH    = 20;                           ! Length of event block
;  0730  0
;  0731  0       !
;  0732  0       ! Event table definitions.
;  0733  0       !
;  0734  0       !...$ETBDEF
;  0735  0
;  0736  0       MACRO           ETB$W_CLASS     = 0,0,16,0%;                    ! Event class
;  0737  0       MACRO           ETB$A_GLOBAL    = 2,0,32,0%;            ! Global filter mask
```

```
;  0738  0       MACRO           ETB$A_NODE      = 6,0,32,0%;            ! Node filter mask
;  0739  0       MACRO           ETB$A_CIRCUIT   = 10,0,32,0%;           ! Circuit filter mask
;  0740  0       MACRO           ETB$A_LINE      = 14,0,32,0%;           ! Line filter mask
;  0741  0       MACRO           ETB$A_MODULE    = 18,0,32,0%;           ! Module filter mask
;  0742  0       LITERAL         ETB$C_ENTRYLEN  = 22;
;  0743  0       LITERAL         ETB$K_ENTRYLEN  = 22;                   ! Length of event table entry
;  0744  0
;  0745  0       !
;  0746  0       ! End of NMLDEF.MDL
;  0747  0       !
```

```
0748  0
0749  0        %TITLE 'NMLDDL - NML Data Definitions'
0750  0        IDENT = 'V04-000'
0751  0
0752  0  ****************************************************************************
0753  0  *                                                                          *
0754  0  *  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
0755  0  *  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
0756  0  *  ALL RIGHTS RESERVED.                                                    *
0757  0  *                                                                          *
0758  0  *  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
0759  0  *  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0760  0  *  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
0761  0  *  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
0762  0  *  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
0763  0  *  TRANSFERRED.                                                            *
0764  0  *                                                                          *
0765  0  *  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
0766  0  *  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0767  0  *  CORPORATION.                                                            *
0768  0  *                                                                          *
0769  0  *  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
0770  0  *  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
0771  0  *                                                                          *
0772  0  *                                                                          *
0773  0  ****************************************************************************
0774  0
0775  0
0776  0  !++
0777  0  ! FACILITY:  DECnet-VAX Network Management Listener
0778  0  !
0779  0  ! ABSTRACT:
0780  0  !
0781  0  !
0782  0  ! This module contains macro and symbol definitions used by all NML modules.
0783  0  !
0784  0  ! ENVIRONMENT:  VAX/VMS Operating System
0785  0  !
0786  0  ! AUTHOR:  Distributed Systems Software Engineering
0787  0  !
0788  0  ! CREATION DATE:  30-DEC-1979
0789  0  !
0790  0  ! MODIFIED BY:
0791  0  !       V03-006 MKP0008         Kathy Perko     24-June-1984
0792  0  !               Increase the size of the QIO P4 buffer to the minimum
0793  0  !               value SYSGEN allows for MAX BUFFER.  This is a slight
0794  0  !               improvement on the limit for the number of sources which
0795  0  !               can be logged for a single sink node.
0796  0  !
0797  0  !       V03-005 MKP0007         Kathy Perko     9-Aprit-1983
0798  0  !               Add globals for executor address in the volatile and
0799  0  !               permanent databases.
0800  0  !
0801  0  !       V03-004 MKP0006         Kathy Perko     19-Sept-1983
0802  0  !               Convert node permanent database to multiple ISAM keys for better
0803  0  !               performance.  Also, make NCP response message entity buffer
0804  0  !               size bigger and global - for X25 tracepoint names.
```

```
0805  0     !
0806  0     !       V03-003 MKP0005         Kathy Perko     19-April-1983
0807  0     !               Delete service functions from NML.
0808  0     !
0809  0     !       V03-002 MKP0004         Kathy Perko     28-June-1982
0810  0     !               Shrink P4 buffer size from 730 bytes to 512 to get rid
0811  0     !               of QIO quota exceeded errors.
0812  0     !               Add macro for generating Search Key IDs for Entity Table.
0813  0     !               Rename qualifier to use its CPT index instead of the Network
0814  0     !               Management parameter code.
0815  0     !
0816  0     !       V03-001 MKP0003         Kathy Perko     17-Mar-1982
0817  0     !               Rename some global fields so the names mean more.
0818  0     !
0819  0     !       V02-002 MKP0002         Kathy Perko     2-Nov-1981
0820  0     !               Delete NML$GW_CMD_CHAN
0821  0     !
0822  0     !       V02-001 MKP0001         Kathy Perko     14-Sept-1981
0823  0     !               Make P4 buffer size smaller so systems with SYSGEN parameter,
0824  0     !               MAXBUF, don't get buffer quota exceeded for SHOW CIRCUIT
0825  0     !               CHARACTERSITICS.
0826  0     !
0827  0     !--
0828  0
0829  0     !
0830  0     ! Miscellaneous symbols
0831  0     !
0832  0     LITERAL
0833  0         FALSE = 0,
0834  0         TRUE  = 1;
0835  0     !
0836  0     ! The following symbols are internal parameter codes.  The values all have
0837  0     ! bit 15 set, indicating a counter value, to avoid conflicts with other
0838  0     ! network management parameter codes.
0839  0     !
0840  0     LITERAL
0841  0         NMA$C_PCNO_A$$ = 1 ^ 15 OR 0,       ! Loop node address
0842  0         NMA$C_PCLI_LC$ = 1 ^ 15 OR 1,       ! Line counters
0843  0         NMA$C_PCNO_EC$ = 1 ^ 15 OR 2,       ! Executor node counters
0844  0         NMA$C_PCNO_NC$ = 1 ^ 15 OR 3,       ! Node counters
0845  0         NMA$C_PCCI_CC$ = 1 ^ 15 OR 4,       ! Circuit counters
0846  0         NMA$C_PCXP_PC$ = 1 ^ 15 OR 5,       ! X-25 Protocol DTE counters.
0847  0         NMA$C_PCXS_SC$ = 1 ^ 15 OR 6;       ! X-25 Server counters
0848  0     !
0849  0     ! Structure declarations used for system defined structures to
0850  0     ! save typing.
0851  0     !
0852  0     STRUCTURE
0853  0         BBLOCK [O, P, S, E; N] =
0854  0             [N]
0855  0             (BBLOCK+O)<P,S,E>,
0856  0
0857  0         BBLOCKVECTOR [I, O, P, S, E; N, BS] =
0858  0             [N*BS]
0859  0             ((BBLOCKVECTOR+I*BS)+O)<P,S,E>;
0860  0
0861  0     !
```

```
;    0862  0      ! Macro to generate Network ACP Control QIO (NFB) P1 buffer contents.  The NFB
;    0863  0      ! describes SET, SHOW, CLEAR, and ZERO operations.
;    0864  0      !
;    0865  0      MACRO
;    0866  0              $NFB (FUNC, FLAGS, DATABASE, SRCH_KEY_ONE, OPER_ONE,
;M   0867  0                                           SRCH_KEY_TWO, OPER_TWO) =
;M   0868  0
;M   0869  0              BYTE ( %IF %IDENTICAL (FUNC, 0)          ! QIO function code.
;M   0870  0                     %THEN 0
;M   0871  0                     %ELSE %NAME ('NFB$C_FC_',FUNC)
;M   0872  0                     %FI),
;M   0873  0              BYTE ( %IF %NULL (FLAGS)                 ! Error Update and Process
;M   0874  0                     %THEN 0                           !       Multiple Entries flags.
;M   0875  0                     %ELSE FLAGS
;M   0876  0                     %FI),
;M   0877  0              BYTE ( %IF %IDENTICAL (DATABASE, 0)      ! ACP database to update.
;M   0878  0                     %THEN 0
;M   0879  0                     %ELSE %NAME ('NFB$C_DB_',DATABASE)
;M   0880  0                     %FI),
;M   0881  0              BYTE (%IF %NULL (OPER_ONE)               ! Oper1
;M   0882  0                     %THEN 0
;M   0883  0                     %ELSE OPER_ONE
;M   0884  0                     %FI
;M   0885  0                     ),
;M   0886  0              $SRCH_KEY (DATABASE, SRCH_KEY_ONE),      ! Search key one ID
;M   0887  0              $SRCH_KEY (DATABASE, SRCH_KEY_TWO),      ! Search key two ID
;M   0888  0              BYTE (%IF %NULL (OPER_TWO)               ! Oper2
;M   0889  0                     %THEN 0
;M   0890  0                     %ELSE OPER_TWO
;M   0891  0                     %FI
;M   0892  0                     ),
;M   0893  0                  BYTE (0),                           ! Spare
;M   0894  0                  WORD (0),                           ! variable cell size
;M   0895  0
;M   0896  0                  %IF NOT %NULL(%REMAINING)
;M   0897  0                  %THEN $FIELD_ID_LIST (DATABASE, %REMAINING)
;M   0898  0                        ,LONG (NFB$C_ENDOFLIST) ! End delimiter for field ID list.
;M   0899  0                  %ELSE
;M   0900  0                        LONG (NFB$C_ENDOFLIST)  ! End delimiter for field ID list.
;M   0901  0                  %FI
;M   0902  0
;    0903  0                  %,
;    0904  0
;    0905  0      !
;    0906  0      ! Generate a Search Key ID for an NFB.  If the Search key is null,
;    0907  0      ! use a wildcard search key ID.
;    0908  0      !
;M   0909  0      $SRCH_KEY (DATABASE, SRCH_ID) =
;M   0910  0          LONG ( %IF %NULL (SRCH_ID)
;M   0911  0                 %THEN NFB$C_WILDCARD
;M   0912  0                 %ELSE $FIELD_ID (DATABASE, SRCH_ID)
;M   0913  0                 %FI )
;    0914  0          %,
;    0915  0      !
;    0916  0      ! Generate a list of longwords containing the NETACP field IDs for
;    0917  0      ! the parameters.  This iterative macro will generate as many
;    0918  0      ! field IDs as are supplied.
```

```
;   0919  0          !
; M 0920  0          $FIELD_ID_LIST (DATABASE) [FIELD_ID] =
; M 0921  0              LONG (%$FIELD_ID (DATABASE, FIELD_ID))
;   0922  0              %,
;   0923  0
; M 0924  0          $FIELD_ID (DATABASE, FIELD_ID) =
; M 0925  0              %IF %IDENTICAL (FIELD_ID, NFB$C_WILDCARD) OR
; M 0926  0                  %IDENTICAL (FIELD_ID, NFB$C_COLLATE)
; M 0927  0              %THEN
; M 0928  0                      FIELD_ID
; M 0929  0              %ELSE
; M 0930  0                      %IF %NULL (FIELD_ID)
; M 0931  0                      %THEN   0
; M 0932  0                      %ELSE %NAME ('NFB$C_',DATABASE,'_',FIELD_ID)
; M 0933  0                      %FI
; M 0934  0              %FI
;   0935  0              %;
;   0936  0
;   0937  0      !
;   0938  0      ! Macros to generate Network Control I/O request descriptors.
;   0939  0      !
;   0940  0      MACRO
;   0941  0              !
;   0942  0              ! Declare the NFB buffer (use the number of input parameters to figure
;   0943  0              ! out how big to make it) and set up a descriptor for it.
;   0944  0              !
; M 0945  0              $NFBDSC (NAM) =
; M 0946  0                      SWITCHES UNAMES;
; M 0947  0                      OWN
; M 0948  0                          _NFB : VECTOR [$NFB_ALLOCATION (%REMAINING)]
; M 0949  0                                       INITIAL ($NFB (%REMAINING));
; M 0950  0                      BIND
; M 0951  0                          %NAME(NAM) = UPLIT (%ALLOCATION(_NFB), _NFB);
; M 0952  0                      UNDECLARE _NFB;
; M 0953  0                      SWITCHES NOUNAMES
;   0954  0                      %,
;   0955  0
; M 0956  0                  $NFB_ALLOCATION [] =
; M 0957  0                      5+(MAX(0,%LENGTH-6))
;   0958  0                      %;
;   0959  0      !
;   0960  0      ! Macro to extract the bit number from bit field references
;   0961  0      !
;   0962  0      MACRO
; M 0963  0          $BITN (O, B, W, S) = B
;   0964  0          %;
;   0965  0      !
;   0966  0      ! Macro to signal status message
;   0967  0      !
;   0968  0      MACRO
; M 0969  0          $SIGNAL_MSG [] =
; M 0970  0              SIGNAL (NML$K_SIG_CODE, %REMAINING)
;   0971  0          %;
;   0972  0      !
;   0973  0      ! Macro to create constant string descriptor
;   0974  0      !
;   0975  0      MACRO
```

```
;  M 0976  0          $ASCID [] =
;  M 0977  0              (UPLIT (%CHARCOUNT(%STRING(%REMAINING)),
;  M 0978  0                      UPLIT BYTE (%STRING(%REMAINING))))
;    0979  0          %;
;    0980  0
;    0981  0      MACRO
;  M 0982  0          $ASCIC [] =
;  M 0983  0              UPLIT BYTE (%ASCIC %STRING (%REMAINING))
;    0984  0          %;
;    0985  0
;    0986  0      !
;    0987  0      ! Macro to move an ASCII counted string to a buffer.
;    0988  0      !
;    0989  0      MACRO
;  M 0990  0          $MOVE_ASCIC (STRING, PTR) =
;  M 0991  0              PTR = CH$MOVE ( %CHARCOUNT (%ASCIC STRING),
;  M 0992  0                              UPLIT BYTE (%ASCIC STRING),
;  M 0993  0                              .PTR)
;    0994  0          %;
;    0995  0
;    0996  0      MACRO
;  M 0997  0          DESCRIPTOR =
;  M 0998  0               BBLOCK [8]
;    0999  0          %;
;    1000  0      !
;    1001  0      ! I/O Status Block definition
;    1002  0      !
;    1003  0      FIELD
;    1004  0          IOSB_FIELDS =
;    1005  0              SET
;    1006  0              IOS$W_STATUS = [0, 0, 16, 0],    ! Status field
;    1007  0              IOS$W_COUNT  = [2, 0, 16, 0],    ! Byte count field
;    1008  0              IOS$L_INFO   = [4, 0, 32, 0]     ! Device dependent information
;    1009  0              TES;
;    1010  0
;    1011  0      MACRO
;  M 1012  0          $IOSB =
;  M 1013  0              BBLOCK [8]  FIELD (IOSB_FIELDS)
;    1014  0          %;
;    1015  0      !
;    1016  0      ! Macro to define Network Management version fields
;    1017  0      !
;    1018  0      FIELD
;    1019  0              NMV_FIELDS =
;    1020  0              SET
;    1021  0              NMV$B_VERSION = [0,0,8,0],
;    1022  0              NMV$B_DEC_ECO = [1,0,8,0],
;    1023  0              NMV$B_USER_ECO = [2,0,8,0]
;    1024  0              TES;
;    1025  0
;    1026  0      MACRO
;  M 1027  0              NMV = BBLOCK [3] FIELD (NMV_FIELDS)
;    1028  0          %;
;    1029  0      !
;    1030  0      ! Macro to define external symbols common to most of the modules.
;    1031  0      !
;  M 1032  0      MACRO $NML_EXTDEF =
```

```
;M 1033  0     EXTERNAL
;M 1034  0     !
;M 1035  0     ! Event data
;M 1036  0     !
;M 1037  0         NML$GB_EVTSRCTYP : BYTE,            ! Event source type
;M 1038  0         NML$GQ_EVTSRCDSC : DESCRIPTOR,      ! Event source descriptor
;M 1039  0         NML$GW_EVTCLASS  : WORD,            ! Event class
;M 1040  0         NML$GB_EVTMSKTYP : BYTE,            ! Mask type
;M 1041  0         NML$GQ_EVTMSKDSC : DESCRIPTOR,      ! Mask descriptor
;M 1042  0         NML$GW_EVTSNKADR : WORD,            ! Sink node address
;M 1043  0     !
;M 1044  0         NML$GW_ACP_CHAN,
;M 1045  0         NML$GL_LOGMASK      : BITVECTOR [32],
;M 1046  0         NML$GQ_ENTSTRDSC    : DESCRIPTOR,
;M 1047  0         NML$AB_QIOBUFFER    : BBLOCK [0],
;M 1048  0         NML$GQ_QIOBFDSC     : DESCRIPTOR,
;M 1049  0         NML$AB_EXEBUFFER    : VECTOR [0, BYTE],
;M 1050  0         NML$GL_EXEDATPTR,
;M 1051  0         NML$GQ_EXEDATDSC    : DESCRIPTOR,
;M 1052  0         NML$GQ_EXEBFDSC     : DESCRIPTOR,
;M 1053  0         NML$AB_RCVBUFFER    : VECTOR [NML$K_RCVBFLEN, BYTE],
;M 1054  0         NML$GQ_RCVBFDSC     : DESCRIPTOR,
;M 1055  0         NML$AB_SNDBUFFER    : VECTOR [NML$K_SNDBFLEN, BYTE],
;M 1056  0         NML$GQ_SNDBFDSC     : DESCRIPTOR,
;M 1057  0         NML$GL_RCVDATLEN,
;M 1058  0         NML$AB_CPTABLE      : BBLOCKVECTOR [0, CPT$K_ENTRYLEN],
;M 1059  0         NML$AB_MSGBLOCK     : BBLOCK [MSB$K_LENGTH],
;M 1060  0         NML$AB_ENTITY_ID    : BBLOCK [16],
;M 1061  0         NML$AB_QUALIFIER_ID : BBLOCK [16],
;M 1062  0         NML$AB_ENTITYDATA   : BBLOCKVECTOR [, EIT$K_ENTRYLEN],
;M 1063  0         NML$AB_NML_NMV      : NMV,
;M 1064  0         NML$AB_PRMSEM       : BBLOCKVECTOR [0, PST$K_ENTRYLEN],
;M 1065  0         NML$AB_RECBUF       : BBLOCK [0],
;M 1066  0         NML$AL_ENTINFTAB    : VECTOR [0],
;M 1067  0         NML$AL_PERMINFTAB   : VECTOR [0],
;M 1068  0         NML$AW_PRM_DES      : BLOCKVECTOR [PDB$K_NUMBER, 4, WORD],
;M 1069  0         NML$GB_CMD_VER      : BYTE,
;M 1070  0         NML$GB_ENTITY_CODE  : BYTE,
;M 1071  0         NML$GB_ENTITY_FORMAT: BYTE,
;M 1072  0         NML$GL_QUALIFIER_PST,
;M 1073  0         NML$GB_QUALIFIER_FORMAT: BYTE,
;M 1074  0         NML$GB_FUNCTION     : BYTE,
;M 1075  0         NML$GB_INFO         : BYTE,
;M 1076  0         NML$GB_OPTIONS      : BYTE,
;M 1077  0         NML$GL_PRMCODE,
;M 1078  0         NML$GL_PRS_FLGS     : BLOCK [1],
;M 1079  0         NML$GL_NML_ENTITY,
;M 1080  0         NML$GQ_NETNAMDSC    : DESCRIPTOR,
;M 1081  0         NML$GQ_RECBFDSC     : DESCRIPTOR,
;M 1082  0         NML$GW_PRMDESCNT    : WORD;
   1083  0         %;
   1084  0     !
   1085  0     ! NPARSE argument block structure definitions
   1086  0     !
   1087  0     MACRO
;M 1088  0         $NPA_ARGDEF =
;M 1089  0             BUILTIN
```

```
;M 1090  0                  AP;
;M 1091  0              BIND
;M 1092  0                  NPARSE_BLOCK = AP : REF $NPA_BLKDEF;
;  1093  0          %;
;  1094  0      !
;  1095  0      ! NPARSE argument block definition macro
;  1096  0      !
;  1097  0      MACRO
;M 1098  0          $NPA_BLKDEF =
;M 1099  0              BBLOCK [NPA$K_LENGTH0]
;  1100  0          %;
;  1101  0      !
;  1102  0      ! Buffer length parameters.
;  1103  0      !
;  1104  0      LITERAL
;  1105  0          NML$K_RCVBFLEN = 512,               ! Receive buffer length
;  1106  0          NML$K_SNDBFLEN = 512,               ! Send buffer length
;  1107  0          NML$K_NFBBFLEN = 256,               ! Max size for an NFB.
;  1108  0          NML$K_QIOBFLEN = 1200,              ! QIO buffer length
;  1109  0          NML$K_P2BUFLEN = 104,               ! Max length for P2 buffers.
;  1110  0          NML$K_RECBFLEN = 1024,              ! Record buffer length
;  1111  0          NML$K_ENTBUFLEN = 64,               ! Entity name buffer size.
;  1112  0          !
;  1113  0          ! Maximum bytes of data in a permanent database record.  Leaves room
;  1114  0          ! for the node keys (which take up the most room) at the beginning of
;  1115  0          ! the record.
;  1116  0          !
;  1117  0          NML$K_MAX_REC_DATA = NML$K_RECBFLEN - NMN$K_NODE_KEYS_LEN,
;  1118  0          NML$K_PERM_KEYS_LEN = 2;            ! Key length for all permanent databases
;  1119  0                                              !       except the node database.
;  1120  0      !
;  1121  0      ! Parameter descriptor block (PDB) definitions.
;  1122  0      !
;  1123  0      LITERAL PDB$K_NUMBER    = 32;           ! Number of parameter descriptor slots
;  1124  0
;  1125  0      MACRO   PDB$W_INDEX     = 0,0,16,0%;    ! Parameter change table (CPT) index
;  1126  0      MACRO   PDB$W_COUNT     = 1,0,16,0%;    ! Parameter byte count
;  1127  0      MACRO   PDB$A_POINTER   = 2,0,32,0%;    ! Pointer to parameter value
;  1128  0
;  1129  0      LITERAL PDB$K_SIZE      = 8;            ! Size of parameter descriptor entry
;  1130  0
```

```
;    1131  0
;    1132  0        Version:     'V04-000'
;    1133  0
;    1134  0        ****************************************************************************
;    1135  0        *                                                                          *
;    1136  0        *  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
;    1137  0        *  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
;    1138  0        *  ALL RIGHTS RESERVED.                                                    *
;    1139  0        *                                                                          *
;    1140  0        *  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
;    1141  0        *  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
;    1142  0        *  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER  *
;    1143  0        *  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  *
;    1144  0        *  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY  *
;    1145  0        *  TRANSFERRED.                                                            *
;    1146  0        *                                                                          *
;    1147  0        *  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
;    1148  0        *  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
;    1149  0        *  CORPORATION.                                                            *
;    1150  0        *                                                                          *
;    1151  0        *  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
;    1152  0        *  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
;    1153  0        *                                                                          *
;    1154  0        *                                                                          *
;    1155  0        ****************************************************************************
;    1156  0
;    1157  0        NPARSE argument block field definitions
;    1158  0
;    1159  0        ...$NPADEF
;    1160  0
;    1161  0        MACRO           NPA$L_COUNT     = 0,0,32,0%;            ! Argument count (NPA$K_COUNT0)
;    1162  0        LITERAL         NPA$K_COUNT0    = 8;                ! Argument count value
;    1163  0        MACRO           NPA$L_MSGCNT    = 4,0,32,0%;            ! Count of bytes remaining in message
;    1164  0        MACRO           NPA$L_MSGPTR    = 8,0,32,0%;            ! Pointer to remaining message
;    1165  0        MACRO           NPA$L_OPTIONS   = 12,0,32,0%;           ! Options (not used)
;    1166  0        MACRO           NPA$L_FLDCNT    = 16,0,32,0%;           ! Count of bytes in matched field
;    1167  0        MACRO           NPA$L_FLDPTR    = 20,0,32,0%;           ! Pointer to matched field
;    1168  0        MACRO           NPA$L_LONG      = 24,0,32,0%;           ! Matched longword value
;    1169  0        MACRO           NPA$B_BYTE      = 24,0,8,0%;    !         byte value
;    1170  0        MACRO           NPA$W_WORD      = 24,0,16,0%;   !         word value
;    1171  0        MACRO           NPA$L_NUMBER    = 28,0,32,0%;           ! Matched signed value (not used)
;    1172  0        MACRO           NPA$L_PARAM     = 32,0,32,0%;           ! Action routine parameter value
;    1173  0
;    1174  0        LITERAL         NPA$C_LENGTH0   = 36;
;    1175  0        LITERAL         NPA$K_LENGTH0   = 36;                   ! Size of argument block structure
;    1176  0
```

```
;    1177  0      !
;    1178  0      ! Version:      'V04-000'
;    1179  0      !
;    1180  0      !++
;    1181  0      !       NMATAIL.B32
;    1182  0      !
;    1183  0      !       Source to undeclare the macros required for the precompile of
;    1184  0      !       NMALIBRY.B32 so they do not appear in the library.
;    1185  0      !--
;    1186  0
;    1187  0
;    1188  0      UNDECLARE %QUOTE $EQULST,
;    1189  0              %QUOTE GET1ST_,
;    1190  0              %QUOTE GET2ND_,
;    1191  0              %QUOTE NUL2ND_
;    1192  0              ;
;    1193  0
;    1194  0      !
;    1195  0      !       End of NMATAIL.B32
;    1196  0      !



;                                       COMMAND QUALIFIERS

;       BLISS/LIBRARY=LIB$:NMLLIB/LIST=LIS$:NMLLIB SRC$:NMAHEAD+LIB$:NMLDEF+SRC$:NMLDDL+LIB$:NPADEF+SRC$:NMATAIL

; Run Time:         00:13.2
; Elapsed Time:     00:17.8
; Lines/CPU Min:     5424
; Lexemes/CPU-Min: 54721
; Memory Used:  87 pages
; Library Precompilation Complete
```

NMLFORWRD
LIS

NMLENTRY
LIS

NMLDEFINE
LIS

NMLFILEIO
LIS

NMLLIB
LIS

NMLINISTA
LIS

NMLDISC
LIS

NMLMSG
LIS

NMLLOGOPS
LIS

NMLMAIN
LIS

NMLNETIO
LIS

NMLLISPRM
LIS

NMLLIST
LIS