```
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
```

```
NN      NN  MM      MM  LL          IIIIII   NN      NN   IIIIII     SSSSSSSS  TTTTTTTTTT    AAAAAA
NN      NN  MM      MM  LL          IIIIII   NN      NN   IIIIII     SSSSSSSS  TTTTTTTTTT    AAAAAA
NN      NN  MMMM  MMMM  LL            II     NN      NN     II     SS            TT        AA    AA
NN      NN  MMMM  MMMM  LL            II     NN      NN     II     SS            TT        AA    AA
NNNN    NN  MM  MM  MM  LL            II     NNNN    NN     II     SS            TT        AA    AA
NNNN    NN  MM  MM  MM  LL            II     NNNN    NN     II     SS            TT        AA    AA
NN  NN  NN  MM      MM  LL            II     NN  NN  NN     II       SSSSSS      TT        AA    AA
NN  NN  NN  MM      MM  LL            II     NN  NN  NN     II       SSSSSS      TT        AA    AA
NN    NNNN  MM      MM  LL            II     NN    NNNN     II             SS    TT        AAAAAAAAAA
NN    NNNN  MM      MM  LL            II     NN    NNNN     II             SS    TT        AAAAAAAAAA
NN      NN  MM      MM  LL            II     NN      NN     II             SS    TT        AA    AA      ....
NN      NN  MM      MM  LL            II     NN      NN     II             SS    TT        AA    AA      ....
NN      NN  MM      MM  LLLLLLLLLL  IIIIII   NN      NN   IIIIII     SSSSSSSS    TT        AA    AA      ....
NN      NN  MM      MM  LLLLLLLLLL  IIIIII   NN      NN   IIIIII     SSSSSSSS    TT        AA    AA      ....

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
```

Table of contents

```
0000     1
0000     2          .TITLE  NML$INISTA      NML Initial parsing state table
0000     3          .IDENT  'V04-000'
0000     4 ;
0000     5 ;*****************************************************************************
0000     6 ;*                                                                           *
0000     7 ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
0000     8 ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
0000     9 ;*  ALL RIGHTS RESERVED.                                                     *
0000    10 ;*                                                                           *
0000    11 ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED    *
0000    12 ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0000    13 ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER    *
0000    14 ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY    *
0000    15 ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY    *
0000    16 ;*  TRANSFERRED.                                                             *
0000    17 ;*                                                                           *
0000    18 ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE    *
0000    19 ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0000    20 ;*  CORPORATION.                                                             *
0000    21 ;*                                                                           *
0000    22 ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS    *
0000    23 ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
0000    24 ;*                                                                           *
0000    25 ;*                                                                           *
0000    26 ;*****************************************************************************
0000    27 ;
0000    28
0000    29 ;++
0000    30 ; FACILITY:  DECnet-VAX Network Management Listener
0000    31
0000    32 ;
0000    33 ; ABSTRACT:
0000    34 ;
0000    35 ;       This module contains the NPARSE state tables for handling the
0000    36 ;       initial processing of all NICE command messages.
0000    37 ;
0000    38 ; ENVIRONMENT:  VAX/VMS Operating System
0000    39 ;
0000    40 ; AUTHOR:  Distributed Systems Software Engineering
0000    41 ;
0000    42 ; CREATION DATE: 8-Sep-1979
0000    43 ;
0000    44 ; MODIFIED BY:
0000    45 ;       V03-011 MKP0013         Kathy Perko     21-Mar-1984
0000    46 ;               Fix parsing of node number to change area 0 to area 1
0000    47 ;               for Phase IV NCPs and to change area 0 to the exec's area
0000    48 ;               for Phase III NCPs.  This to fix the area 1 probem.
0000    49 ;
0000    50 ;       V03-010 MKP0012         Kathy Perko     6-Jan-1984
0000    51 ;               Add X25-Access Module entity.
0000    52 ;
0000    53 ;       V03-009 MKP0011         Kathy Perko     20-April-1983
0000    54 ;               Add support to call MOM if I get a system specific down
0000    55 ;               line load command.
0000    56 ;
0000    57 ;       V03-008 MKP0010         Kathy Perko     17-Jan-1983
```

```
0000     58 ;                  Add support for NI CONFIGURATOR module.
0000     59 ;
0000     60 ;       V03-007 MKP0009         Kathy Perko     12-Nov-1982
0000     61 ;               Allow CIRCUIT qualifier on any SHOW NODE command.
0000     62 ;
0000     63 ;       V03-006 MKP0008         Kathy Perko     7-Nov-1982
0000     64 ;               Add area entity.
0000     65 ;
0000     66 ;       V03-005 MKP0007         Kathy Perko     1-Oct-1982
0000     67 ;               Add SHOW ADJACENT NODES and SHOW CIRCUIT ADJACENT NODE
0000     68 ;               parsing.
0000     69 ;
0000     70 ;       V03-004 MKP0006         Kathy Perko     20-Sept-1982
0000     71 ;               Fix node parsing so that the internal entity ID for
0000     72 ;               NML is properly set up.
0000     73 ;
0000     74 ;       V03-003 MKP0005         Kathy Perko     9-Sept-1982
0000     75 ;               For errors in X25-Protocol qualifiers, return
0000     76 ;               detail = module instead of the qualifier parameter ID.
0000     77 ;               The architecture needs to be changed to make this better.
0000     78 ;
0000     79 ;       V03-002 MKP0004         Kathy Perko     17-June-1982
0000     80 ;               Fix parsing for X25-Protocol networks to default
0000     81 ;               to using null string as entity id - indicates the
0000     82 ;               currently active network.
0000     83 ;               Add SHOW/LIST GROUP parsing for DTE qualifier.
0000     84 ;               Add X29 Server and Trace modules.
0000     85 ;
0000     86 ;       V03-001 MKP0003         Kathy Perko     16-June-1982
0000     87 ;               Add parsing for qualifiers, and clean up X25-PROTOCOL
0000     88 ;               Module parsing.
0000     89 ;
0000     90 ;       V02-002 MKP0002         Kathy Perko     14-Nov-1981
0000     91 ;               Fix circuit id parsing so same checks are made as
0000     92 ;               for line id parsing.
0000     93 ;
0000     94 ;       V02-001 MKP0001         Kathy Perko     19-July-1981
0000     95 ;               Add Circuit and Module entities.
0000     96 ;
0000     97 ;--
```

```
0000     99          .SBTTL  Declarations
0000    100 ;
0000    101 ; INCLUDE FILES:
0000    102 ;
0000    103
0000    104 $NMADEF                     ; Network Management Layer symbols
0000    105 $NMLDEF                     ; Network Management Listener definitions
0000    106
0000    107 ;
0000    108 ; MACROS:
0000    109 ;
0000    110
0000    111 ;
0000    112 ; EQUATED SYMBOLS:
0000    113 ;
0000    114
0000    115 ;
0000    116 ; OWN STORAGE:
0000    117 ;
```

```
0000   119          .SBTTL  NML$NPA_INIT - NICE Phase III message parse tables
0000   120 ;+
0000   121 ; Phase III parse tables
0000   122 ;-
0000   123          IMSG$   NML$NPA_INIT
0000   124 ;
0000   125 ; Get the function code
0000   126 ;
0000   127          FIELD$
0000   128          $BYTE   NMA$C_FNC_CHA,NML_CHA,NML$PRSFNC ; Change
0000   129          $BYTE   NMA$C_FNC_REA,NML_REA,NML$PRSFNC ; Read
0000   130          $BYTE   NMA$C_FNC_ZER,NML_ZER,NML$PRSFNC ; Zero
0000   131          $BYTE   NMA$C_FNC_TES,NML_TES,NML$PRSFNC ; Test
0000   132          $BYTE   NMA$C_FNC_LOA,NML_LOA,NML$PRSFNC ; Load
0000   133          $BYTE   NMA$C_FNC_DUM,NML_DUM,NML$PRSFNC ; Dump
0000   134          $BYTE   NMA$C_FNC_TRI,NML_TRI,NML$PRSFNC ; Trigger
0000   135          $BYTE   NMA$C_FNC_SYS,NML_SYS            ; System-specific function
0000   136          $NULL   ,NML_FUN_ERR                     ; Unrecognized function
0000   137 ;
0000   138 ; Down-line load
0000   139 ;
0000   140          FIELD$  NML_LOA
0000   141          $SBEXP  NML_CHKOPT_SUB,NPA$_EXIT         ; Parse the option byte
0000   142 ;
0000   143 ; Up-line dump
0000   144 ;
0000   145          FIELD$  NML_DUM
0000   146          $SBEXP  NML_CHKOPT_SUB,NPA$_EXIT         ; Parse the option byte
0000   147 ;
0000   148 ; Trigger bootstrap
0000   149 ;
0000   150          FIELD$  NML_TRI
0000   151          $SBEXP  NML_CHKOPT_SUB,NPA$_EXIT         ; Parse the option byte
0000   152 ;
0000   153 ; Send test message
0000   154 ;
0000   155          FIELD$  NML_TES
0000   156          $SBEXP  NML_CHKOPT_SUB,NPA$_EXIT         ; Parse the option byte
0000   157
0000   158 ;
0000   159 ; Change parameter
0000   160 ;
0000   161          FIELD$  NML_CHA                          ; Check change parameter option
0000   162          $SBEXP  NML_CHKOPT_SUB                   ; Parse the option byte
0000   163          FIELD$
0000   164          $SBEXP  NML_CHKENT_SUB,NPA$_EXIT         ; Parse entity id
0000   165
0000   166 ;
0000   167 ; Read information
0000   168 ;
0000   169          FIELD$  NML_REA                          ; Check read information option
0000   170          $SBEXP  NML_CHKOPT_SUB                   ; Parse the option byte
0000   171          FIELD$
0000   172          $SBEXP  NML_CHK_INFO_SUB                 ; Check information type
0000   173          FIELD$
0000   174          $SBEXP  NML_CHKENT_SUB,NPA$_EXIT         ; Parse entity id
0000   175
```

```
0000    176 ;
0000    177 ; Zero counters
0000    178 ;
0000    179         FIELD$  NML_ZER                         ; Check zero counters option
0000    180         $SBEXP  NML_CHKOPT_SUB                  ; Parse the option byte
0000    181         FIELD$                                  ; Parse the entity ID
0000    182         $EXTZV  <NMA$C_ENT_NOD, -               ; Node
0000    183                  NMA$V_OPT_ENT, -
0000    184                  NMA$S_OPT_ENT, -
0000    185                  NPA$_ADVANCE>, NML_NOD_ENT,NML$PRSENT
0000    186         $EXTZV  <NMA$C_ENT_LIN, -               ; Line
0000    187                  NMA$V_OPT_ENT, -
0000    188                  NMA$S_OPT_ENT, -
0000    189                  NPA$_ADVANCE>, NML_LIN_ENT,NML$PRSENT
0000    190         $EXTZV  <NMA$C_ENT_CIR, -               ; Circuit
0000    191                  NMA$V_OPT_ENT, -
0000    192                  NMA$S_OPT_ENT, -
0000    193                  NPA$_ADVANCE>, NML_CIRCUIT_ENT,NML$PRSENT
0000    194         $EXTZV  <NMA$C_ENT_MOD, -               ; Module
0000    195                  NMA$V_OPT_ENT, -
0000    196                  NMA$S_OPT_ENT, -
0000    197                  NPA$_ADVANCE>, NML_MODULE_ENT,NML$PRSENT
0000    198         $NULL   ,NML_FUN_ERR                    ; Unrecognized option
0000    199
0000    200 ;
0000    201 ; System-specific functions
0000    202 ;
0000    203         FIELD$  NML_SYS                         ; System-specific function
0000    204         $EOM    ,NML_FOR_ERR                    ; Message format error
0000    205         $BYTE   NMA$C_SYS_VMS,NML_VMS,,NML$M_PRS_VMS,NML$GL_PRS_FLGS
0000    206         $NULL   ,NML_SYS_ERR                    ; Only VMS is valid
0000    207
0000    208         FIELD$  NML_VMS                         ; VMS-specific function
0000    209         $EOM    ,NML_FOR_ERR                    ; Message format error
0000    210         $BYTE   NMA$C_FNC_CHA,NML_VMS_CHA,NML$PRSFNC ; Change
0000    211         $BYTE   NMA$C_FNC_REA,NML_VMS_REA,NML$PRSFNC ; Read
0000    212         $BYTE   NMA$C_FNC_LOA,NPA$_EXIT,NML$PRSFNC   ; Load
0000    213         $NULL   ,NML_FUN_ERR                    ; Unrecognized function
0000    214
0000    215 ;
0000    216 ; Change parameter for VMS specific function
0000    217 ;
0000    218         FIELD$  NML_VMS_CHA                     ; Check change parameter option
0000    219         $SBEXP  NML_CHKOPT_SUB                  ; Parse the option byte
0000    220         FIELD$
0000    221         $SBEXP  NML_CHK_VMS_ENT_SUB,NPA$_EXIT   ; Parse entity id for VMS
0000    222                                                 ;       specific
0000    223 ;
0000    224 ; Read information for VMS specific function
0000    225 ;
0000    226         FIELD$  NML_VMS_REA                     ; Check read information option
0000    227         $SBEXP  NML_CHKOPT_SUB                  ; Parse the option byte
0000    228         FIELD$
0000    229         $SBEXP  NML_CHK_INFO_SUB                ; Check information type
0000    230         FIELD$
0000    231         $SBEXP  NML_CHK_VMS_ENT_SUB,NPA$_EXIT   ; Parse entity id for VMS
0000    232                                                 ;       specific
```

```
0000   234 ;
0000   235 ; Parse information type subexpression
0000   236 ;
0000   237         FIELD$  NML_CHK_INFO_SUB                 ; Match the information type
0000   238         $EXTZV  <NMA$C_OPINF_SUM, -              ; Summary
0000   239                  NMA$V_OPT_INF, -
0000   240                  NMA$S_OPT_INF  -
0000   241                                 >,NPA$_EXIT,NML$PRSINF
0000   242         $EXTZV  <NMA$C_OPINF_STA, -              ; Status
0000   243                  NMA$V_OPT_INF, -
0000   244                  NMA$S_OPT_INF  -
0000   245                                 >,NPA$_EXIT,NML$PRSINF
0000   246         $EXTZV  <NMA$C_OPINF_CHA, -              ; Characteristics
0000   247                  NMA$V_OPT_INF, -
0000   248                  NMA$S_OPT_INF  -
0000   249                                 >,NPA$_EXIT,NML$PRSINF
0000   250         $EXTZV  <NMA$C_OPINF_COU, -              ; Counters
0000   251                  NMA$V_OPT_INF, -
0000   252                  NMA$S_OPT_INF  -
0000   253                                 >,NPA$_EXIT,NML$PRSINF
0000   254         $EXTZV  <NMA$C_OPINF_EVE, -              ; Events
0000   255                  NMA$V_OPT_INF, -
0000   256                  NMA$S_OPT_INF  -
0000   257                                 >,NPA$_EXIT,NML$PRSINF
0000   258         $NULL   ,NML_FUN_ERR                     ; Unrecognized option
0000   259
```

```
0000   261          .SBTTL  Entity parsing
0000   262 ;
0000   263 ; Parse the entity for a non VMS specific command, according to the option byte
0000   264 ;
0000   265          FIELD$  NML_CHKENT_SUB
0000   266          $EXTZV  <NMA$C_ENT_NOD, -                   ; Node
0000   267                   NMA$V_OPT_ENT, -
0000   268                   NMA$S_OPT_ENT, -
0000   269                   NPA$_ADVANCE>, NML_NOD_ENT,NML$PRSENT
0000   270          $EXTZV  <NMA$C_ENT_LIN, -                   ; Line
0000   271                   NMA$V_OPT_ENT, -
0000   272                   NMA$S_OPT_ENT, -
0000   273                   NPA$_ADVANCE>, NML_LIN_ENT,NML$PRSENT
0000   274          $EXTZV  <NMA$C_ENT_LOG, -                   ; Logging
0000   275                   NMA$V_OPT_ENT, -
0000   276                   NMA$S_OPT_ENT, -
0000   277                   NPA$_ADVANCE>, NML_LOG_ENT,NML$PRSENT
0000   278          $EXTZV  <NMA$C_ENT_CIR, -                   ; Circuit
0000   279                   NMA$V_OPT_ENT, -
0000   280                   NMA$S_OPT_ENT, -
0000   281                   NPA$_ADVANCE>, NML_CIRCUIT_ENT,NML$PRSENT
0000   282          $EXTZV  <NMA$C_ENT_MOD, -                   ; Module
0000   283                   NMA$V_OPT_ENT, -
0000   284                   NMA$S_OPT_ENT, -
0000   285                   NPA$_ADVANCE>, NML_MODULE_ENT,NML$PRSENT
0000   286          $EXTZV  <NMA$C_ENT_ARE, -                   ; Area
0000   287                   NMA$V_OPT_ENT, -
0000   288                   NMA$S_OPT_ENT, -
0000   289                   NPA$_ADVANCE>, NML_AREA_ENT,NML$PRSENT
0000   290          $NULL   ,NML_FUN_ERR                        ; Unrecognized function
0000   291
0000   292
0000   293 ;
0000   294 ; Parse VMS specific entities.
0000   295 ;
0000   296          FIELD$  NML_CHK_VMS_ENT_SUB
0000   297          $EXTZV  <NMA$C_SENT_OBJ, -                  ; Object
0000   298                   NMA$V_OPT_ENT, -
0000   299                   NMA$S_OPT_ENT, -
0000   300                   NPA$_ADVANCE>, NML_OBJ_ENT,NML$PRSENT
0000   301          $EXTZV  <NMA$C_SENT_LNK, -                  ; Link
0000   302                   NMA$V_OPT_ENT, -
0000   303                   NMA$S_OPT_ENT, -
0000   304                   NPA$_ADVANCE>, NML_LNK_ENT,NML$PRSENT
0000   305          $NULL   ,NML_FUN_ERR                        ; Unrecognized option
0000   306
0000   307 ;
0000   308 ; Parse a node entity
0000   309 ;
0000   310          FIELD$  NML_NOD_ENT                         ; Node-id entity
0000   311          $EOM     ,NML_FOR_ERR                       ; Message format error
0000   312          $LOOK   0,NML_NOD_NUM                       ; 3 bytes if node number
0000   313          $BYTE   NMA$C_ENT_LOO,NML_CHKEOM,NML$PRSIDLEQ, -
0000   314                           NML$C_LOOPNODE,NML$GL_NML_ENTITY
0000   315          $BYTE   NMA$C_ENT_KNO,NML_NODE_CIRC_QUAL,NML$PRSIDLEQ, -
0000   316                           NML$C_NODE,NML$GL_NML_ENTITY
0000   317          $BYTE   NMA$C_ENT_ACT,NML_NODE_CIRC_QUAL,NML$PRSIDLEQ, -
```

```
0000   318                          NML$C_NODE,NML$GL_NML_ENTITY
0000   319          $BYTE   NMA$C_ENT_ADJ,NML_NODE_CIRC_QUAL,NML$PRSIDLEQ, -
0000   320                          NML$C_ADJACENT_NODE,NML$GL_NML_ENTITY
0000   321          $IMAGE  6,NML_NODE_CIRC_QUAL,NML$PRSNODNAM,NML$C_NODEBYNAME, -; n bytes (nam
0000   322                          NML$GL_NML_ENTITY
0000   323          $ERROR  NML$_STS_IDE,,NML$PRSIDERR,,,NMA$C_ENT_NOD ; Illegal node name
0000   324
0000   325          FIELD$  NML_NOD_NUM                             ; Node-id number
0000   326          $SBEXP  NML_NOD_EXE,NPA$_EXIT,NML$PRSID
0000   327          $MATCH  3,NML_NODE_CIRC_QUAL, -                 ; Skip 2 bytes (node number)
0000   328                          NML$PRS_NODE_NUM_ENTITY, -
0000   329                          NML$C_NODE,NML$GL_NML_ENTITY
0000   330          $NULL   ,NML_FOR_ERR                            ; Must be at least 2 bytes
0000   331
0000   332          FIELD$  NML_NOD_EXE                             ; Zero address is executor node
0000   333          $MATCH  1
0000   334          FIELD$
0000   335          $WORD   0,NPA$_EXIT,,NML$C_EXECUTOR, -  ; Executor node
0000   336                                  NML$GL_NML_ENTITY
0000   337
0000   338 ;
0000   339 ; Parse a circuit qualifier (for SHOW NODE[S] CIRC <circuit id>)
0000   340 ;
0000   341          FIELD$  NML_NODE_CIRC_QUAL
0000   342          $EOM    ,NPA$_EXIT                                      ; No qualifier is present
0000   343          $LOOK   ,NPA$_EXIT,NML$PRS_NOREAD                       ; Get out if not a read.
0000   344          $WORD   NMA$C_PCNO_DLI,NML_CHECK_QUAL
0000   345          $ERROR  NML$_STS_IDE,,NML$PRSIDERR,,,NMA$C_ENT_NOD ; Illegal node name
0000   346
0000   347          FIELD$  NML_CHECK_QUAL
0000   348          $EOM    ,NML_FOR_ERR
0000   349          $MATCH  1,NPA$_EXIT,NML$PRSIDLEQ                        ; KNOWN circuits is the
0000   350                                                                  ;       default.
0000   351          $NULL   ,NML_CIRC_QUAL,,NML$M_PRS_QUALIFIER, -  ; Flag qualifier
0000   352                                          NML$GL_PRS_FLGS ;       present.
0000   353          FIELD$  NML_CIRC_QUAL                           ; Parse circuit qualifier
0000   354          $IMAGE  16,NPA$_EXIT,NML$PRSIDN,,,CPT$GK_PCNO_DLI
0000   355          $ERROR  NML$_STS_IDE,,NML$PRSIDERR,,,NMA$C_ENT_CIR; Illegal circuit name
0000   356
0000   357 ;
0000   358 ; Parse a line entity
0000   359 ;
0000   360          FIELD$  NML_LIN_ENT                             ; Line-id entity
0000   361          $EOM    ,NML_FOR_ERR                            ; Message format error
0000   362          $MATCH  1,NML_LINE,NML$PRSIDLEQ ; 1 byte if zero or negative
0000   363          $IMAGE  16,NML_LINE,NML$PRSDEVICE               ; n bytes (line name)
0000   364          $ERROR  NML$_STS_IDE,,NML$PRSIDERR,,,NMA$C_ENT_LIN ; Illegal line-id
0000   365
0000   366          FIELD$  NML_LINE                                ; Set internal NML entity type
0000   367          $NULL   ,NPA$_EXIT,,NML$C_LINE,NML$GL_NML_ENTITY
0000   368
0000   369 ;
0000   370 ; Parse a logging entity
0000   371 ;
0000   372          FIELD$  NML_LOG_ENT                             ; Logging entity
0000   373          $MATCH  1,NML_LOG_SIN,NML$PRSID, -              ; 1 byte, always.
0000   374                  NML$C_LOGGING,NML$GL_NML_ENTITY ; Set internal NML entity type.
```

```
0000   375          $NULL   ,NML_FOR_ERR                          ; Message format error
0000   376
0000   377          FIELD$  NML_LOG_SIN                          ; If READ then sink node
0000   378          $NULL   ,NPA$_EXIT,NML$PRSLOGSIN             ;  must be specified
0000   379          $EOM    ,NPA$_EXIT,NML$PRSEXESNK             ; Null is executor node
0000   380          $WORD   NMA$C_PCLO_SIN,NML_LOG_NID
0000   381
0000   382          FIELD$  NML_LOG_NID
0000   383          $SBEXP  NML_LOG_NIDSUB,NML_CHKEOM,,NML$M_PRS_SNKNOD,NML$GL_PRS_FLGS
0000   384          $ERROR  NML$_STS_IDE,,NML$PRSIDERR,,,NMA$C_ENT_NOD ; Illegal node id
0000   385
0000   386          FIELD$  NML_LOG_NIDSUB
0000   387          $BYTE   NMA$C_ENT_KNO,NPA$_EXIT,,NML$M_PRS_KNOSNK,NML$GL_PRS_FLGS ; Known si
0000   388          $LOOK   0,NML_LOG_NODNUM
0000   389          $IMAGE  6,NPA$_EXIT,NML$PRSSNKNNA
0000   390          $NULL   ,NML_FOR_ERR                         ; Message format error
0000   391
0000   392          FIELD$  NML_LOG_NODNUM
0000   393          $MATCH  3,NPA$_EXIT,NML$PRSSNKNAD
0000   394
0000   395 ;
0000   396 ; Parse a circuit entity
0000   397 ;
0000   398          FIELD$  NML_CIRCUIT_ENT                      ; Circuit-id entity
0000   399          $EOM    ,NML_FOR_ERR                         ; Message format error
0000   400          $MATCH  1,NML_CIRCUIT,NML$PRSIDLEQ, -        ; 1 byte if zero or negative
0000   401                  NML$C_CIRCUIT,NML$GL_NML_ENTITY
0000   402          $IMAGE  16,NML_CIRCUIT,NML$PRSDEVICE, -      ; n bytes (circuit name)
0000   403                  NML$C_CIRCUIT,NML$GL_NML_ENTITY
0000   404          $ERROR  NML$_STS_IDE,,NML$PRSIDERR,,,NMA$C_ENT_CIR; Illegal circuit-id
0000   405
0000   406          FIELD$  NML_CIRCUIT                          ; Set internal NML entity type
0000   407          $EOM    ,NPA$_EXIT                           ; No Adjacency qualifier - exit
0000   408          $WORD   NMA$C_PCCI_ADJ,NML_CIRC_ADJACENCY
0000   409          $NULL   ,NPA$_EXIT                           ; Check for other parameters
0000   410                                                       ;       later.
0000   411
0000   412          FIELD$  NML_CIRC_ADJACENCY
0000   413          $EOM    ,NML_FOR_ERR
0000   414          $BYTE   NMA$C_ENT_KNO,NPA$_EXIT              ; Known adjacencies is default.
0000   415          $NULL   ,NML_PRS_ADJ,,NML$M_PRS_QUALIFIER, - ; Set qualifier flag
0000   416                  NML$GL_PRS_FLGS
0000   417          FIELD$  NML_PRS_ADJ
0000   418          $LOOK   0,NML_ADJ_NOD_NUM                    ; 3 bytes if node number
0000   419          $IMAGE  16,NPA$_EXIT,NML$PRSIDN,,, -         ; Save adjacent node name.
0000   420                  CPT$GK_PCCI_AJ$
0000   421          $ERROR  NML$_STS_IDE,,NML$PRSIDERR,,,NMA$C_ENT_NOD; Illegal Adjacency id
0000   422
0000   423          FIELD$  NML_ADJ_NOD_NUM                      ; Adjacent node-id is a number
0000   424          $SBEXP  NML_ADJ_EXE,NPA$_EXIT
0000   425          $MATCH  3,NPA$_EXIT,NML$PRS_NODE_NUM,,, -    ; Save adjacent node number.
0000   426                  CPT$GK_PCCI_ADJ
0000   427
0000   428          FIELD$  NML_ADJ_EXE                          ; Zero address is executor node
0000   429          $MATCH  1
0000   430          FIELD$
0000   431          $WORD   0,NML_NOD_ERR
```

```
                0000   432
                0000   433          FIELD$  NML_NOD_ERR
                0000   434          $ERROR  NML$_STS_IDE,,NML$PRSIDERR,,,NMA$C_ENT_NOD; Illegal Adjacency id
```

```
0000   436 ;
0000   437 ; Parse a module entity
0000   438 ;
0000   439         FIELD$  NML_MODULE_ENT                      ; Module entity
0000   440         $EOM    ,NML_FOR_ERR                       ; Message format error
0000   441         $MATCH  1,NML_FUN_ERR,NML$PRSIDLEQ         ; Don't allow KNOWN or ACTIVE
0000   442                                                    ;       modules.
0000   443         $IMAGE  10,NML_MOD_X25_ACCESS,NML$PRS_MODULE,,,NML$C_X25_ACCESS
0000   444         $IMAGE  12,NML_MOD_PROTOCOL,NML$PRS_MODULE,,,NML$C_PROTOCOL
0000   445         $IMAGE  10,NML_MOD_X25_SERVER,NML$PRS_MODULE,,,NML$C_X25_SERV
0000   446         $IMAGE  10,NML_MOD_X25_TRACE,NML$PRS_MODULE,,,NML$C_TRACE
0000   447         $IMAGE  10,NML_MOD_X29_SERVER,NML$PRS_MODULE,,,NML$C_X29_SERV
0000   448         $IMAGE  12,NML_MOD_NI_CONFIG,NML$PRS_MODULE,,,NML$C_NI_CONFIG
0000   449         $ERROR  NML$_STS_IDE,,NML$PRSIDERR,,,NMA$C_ENT_MOD; Illegal module-id
0000   450
0000   451 ;
0000   452 ; Parse X25 Access module - Network qualifier is manadatory
0000   453 ;
0000   454         FIELD$  NML_MOD_X25_ACCESS
0000   455         $EOM    ,NML_INV_DEST                      ; No network ID in command
0000   456         $WORD   NMA$C_PCXA_NET,NML_X25_ACCESS_NET
0000   457         $NULL   ,NML_INV_DEST
0000   458
0000   459         FIELD$  NML_X25_ACCESS_NET                 ; Check destination ID
0000   460         $EOM    ,NML_FOR_ERR                       ; No destination ID, message
0000   461                                                    ;       format error
0000   462         $NULL   ,NML_MOD_ENT,,NML$C_X25_ACCESS, - ; Set internal NML entity type.
0000   463                 NML$GL_NML_ENTITY
0000   464
0000   465         FIELD$  NML_INV_DEST                       ; Illegal destination id
0000   466         $ERROR  NML$_STS_IDE,,NML$PRSIDERR,,,NMA$C_ENT_MOD
0000   467
0000   468 ;
0000   469 ; Parse for X25 Protocol module sub entity - DTE or group
0000   470 ;
0000   471         FIELD$  NML_MOD_PROTOCOL
0000   472         $EOM    ,NML_PROTCL_NET                    ; No sub entity.  Use network.
0000   473         $WORD   NMA$C_PCXP_DTE,NML_PROTCL_DTE      ; DTE
0000   474         $WORD   NMA$C_PCXP_GRP,NML_PROTCL_GRP      ; Group
0000   475         $NULL   ,NML_PROTCL_NET                    ; Assume network
0000   476
0000   477         FIELD$  NML_PROTCL_NET                     ; Network sub entity.
0000   478         $NULL   ,NPA$_EXIT,NML$PRS_ACTIVE_NET, -   ; Set internal NML entity type.
0000   479                 NML$C_PROT_NET, -                  ;       and active network
0000   480                 NML$GL_NML_ENTITY                  ;       entity.
0000   481
0000   482
0000   483         FIELD$  NML_PROTCL_DTE                     ; DTE sub entity.
0000   484         $EOM    ,NML_FOR_ERR                       ; No DTE ID, message format error
0000   485         $NULL   ,NML_MOD_ENT,,NML$C_PROT_DTE, -    ; Set internal NML entity type.
0000   486                 NML$GL_NML_ENTITY
0000   487
0000   488
0000   489         FIELD$  NML_PROTCL_GRP                     ; Check Group ID
0000   490         $EOM    ,NML_FOR_ERR                       ; No destination ID, message
0000   491                                                    ;       format error
0000   492         $SBEXP  NML_MOD_ENT,NML_CHK_DTE,, -        ; Set internal NML entity type.
```

```
0000   493                          NML$C_PROT_GRP, NML$GL_NML_ENTITY
0000   494          $ERROR  NML$_STS_IDE,,NML$PRSIDERR,,,NMA$C_ENT_MOD; Illegal group-id
0000   495
0000   496          FIELD$  NML_CHK_DTE
0000   497          $EOM    ,NPA$_EXIT                          ; No DTE qualifier - exit
0000   498          $WORD   NMA$C_PCXP_GDT,NML_GRP_DTE_ID       ; Go check Group DTE id.
0000   499          $ERROR  NML$_STS_IDE,,NML$PRSIDERR,,,NMA$C_ENT_MOD; Illegal group-id
0000   500
0000   501          FIELD$  NML_GRP_DTE_ID
0000   502          $EOM    ,NML_FOR_ERR
0000   503          $MATCH  1,NPA$_EXIT,NML$PRSQUALLEQ          ; Known DTEs is default.
0000   504          $NULL   ,NML_DTE_QUAL,,NML$M_PRS_QUALIFIER, -   ; Flag qualifier
0000   505                                          NML$GL_PRS_FLGS ;       present.
0000   506
0000   507          FIELD$  NML_DTE_QUAL
0000   508          $IMAGE  16,NPA$_EXIT,NML$PRSIDN,,, -        ; Save Group-DTE ID.
0000   509                                        CPT$GK_PCXP_GDT
0000   510          $ERROR  NML$_STS_IDE,,NML$PRSIDERR,,,NMA$C_ENT_MOD; Illegal DTE id
0000   511
0000   512
0000   513 ;
0000   514 ; Parse X25 Server module
0000   515 ;
0000   516          FIELD$  NML_MOD_X25_SERVER
0000   517          $EOM    ,NPA$_EXIT,,NML$C_X25_SERV, -       ; No destination sub entity
0000   518                                  NML$GL_NML_ENTITY
0000   519          $WORD   NMA$C_PCXS_DST,NML_X25_SERV_DEST; Sub entity is destination.
0000   520          $NULL   ,NPA$_EXIT,,NML$C_X25_SERV, -       ; No destination sub entity.
0000   521                                  NML$GL_NML_ENTITY;      Parameters follow.
0000   522
0000   523          FIELD$  NML_X25_SERV_DEST                   ; Sub entity is Destination.
0000   524          $EOM    ,NML_FOR_ERR                        ; No destination ID, message
0000   525                                                      ;       format error
0000   526          $NULL   ,NML_MOD_ENT,,NML$C_X25_SERV_DEST, -; Set internal NML entity
0000   527                          NML$GL_NML_ENTITY           ;       type.
0000   528
0000   529 ;
0000   530 ; Parse X25 Trace module
0000   531 ;
0000   532          FIELD$  NML_MOD_X25_TRACE
0000   533          $EOM    ,NPA$_EXIT,,NML$C_TRACE, -          ; No tracepoint sub entity
0000   534                                  NML$GL_NML_ENTITY
0000   535          $WORD   NMA$C_PCXT_TPT,NML_TRACEPOINT       ; Sub entity is tracepoint.
0000   536          $NULL   ,NPA$_EXIT,,NML$C_TRACE, -          ; No tracepoint sub entity.
0000   537                                  NML$GL_NML_ENTITY;      Parameters follow.
0000   538
0000   539          FIELD$  NML_TRACEPOINT                      ; Sub entity is tracepoint
0000   540          $EOM    ,NML_FOR_ERR                        ; No tracepoint ID, message
0000   541                                                      ;       format error
0000   542          $NULL   ,NML_TPT_ENT,,NML$C_TRACEPNT, -     ; Set internal NML entity
0000   543                          NML$GL_NML_ENTITY           ;       type.
0000   544          FIELD$  NML_TPT_ENT                         ; Finish processing module entity.
0000   545          $MATCH  1,NPA$_EXIT,NML$PRSIDLEQ            ; 1 byte if zero or negative.
0000   546          $IMAGE  31,NPA$_EXIT,NML$PRSIDN             ; 16 bytes of ID
0000   547          $ERROR  NML$_STS_IDE,,NML$PRSIDERR,,,NMA$C_ENT_MOD; Illegal module id
0000   548
0000   549
```

```
0000   550 ;
0000   551 ; Parse X29 Server module
0000   552 ;
0000   553         FIELD$  NML_MOD_X29_SERVER
0000   554         $EOM    ,NPA$_EXIT,,NML$C_X29_SERV, -    ; No destination sub entity
0000   555                                 NML$GL_NML_ENTITY
0000   556         $WORD   NMA$C_PCXS_DST,NML_X29_SERV_DEST; Sub entity is destination.
0000   557         $NULL   ,NPA$_EXIT,,NML$C_X29_SERV, -    ; No destination sub entity.
0000   558                                 NML$GL_NML_ENTITY;      Parameters follow.
0000   559
0000   560         FIELD$  NML_X29_SERV_DEST                ; Sub entity is Destination.
0000   561         $EOM    ,NML_FOR_ERR                     ; No destination ID, message
0000   562                                                  ;       format error
0000   563         $NULL   ,NML_MOD_ENT,,NML$C_X29_SERV_DEST, -; Set internal NML entity
0000   564                 NML$GL_NML_ENTITY                ;       type.
0000   565
0000   566
0000   567 ;
0000   568 ; Parse NI configurator module
0000   569 ;
0000   570         FIELD$  NML_MOD_NI_CONFIG
0000   571         $EOM    ,NPA$_EXIT,,NML$C_NI_CONFIG, -   ; No circuit qualifier - exit
0000   572                 NML$GL_NML_ENTITY
0000   573         $WORD   NMA$C_PCCN_CIR,NML_NI_CONFIG_CIRC,, -
0000   574                 NML$C_NI_CONFIG,NML$GL_NML_ENTITY
0000   575         $NULL   ,NPA$_EXIT,,NML$C_NI_CONFIG, -   ; Check for other parameters
0000   576                 NML$GL_NML_ENTITY                ;       later.
0000   577
0000   578         FIELD$  NML_NI_CONFIG_CIRC
0000   579         $EOM    ,NML_FOR_ERR
0000   580         $NEXT
094C   581
094C   582
094C   583         FIELD$  NML_MOD_ENT                      ; Finish processing module entity.
0000   584         $MATCH  1,NPA$_EXIT,NML$PRSIDLEQ         ; 1 byte if zero or negative.
0000   585         $IMAGE  16,NPA$_EXIT,NML$PRSIDN          ; 16 bytes of ID
0000   586         $ERROR  NML$_STS_IDE,,NML$PRSIDERR,,,NMA$C_ENT_MOD; Illegal module id
```

```
0000    588 ;
0000    589 ; Parse an area entity
0000    590 ;
0000    591         FIELD$  NML_AREA_ENT                        ; Area entity
0000    592         $EOM    ,NML_FOR_ERR                        ; Message format error
0000    593         $LOOK   0,NML_AREA_NUM                      ; Area number in next byte.
0000    594         $BYTE   NMA$C_ENT_RNO,NPA$_EXIT,NML$PRSIDLEQ, -
0000    595                         NML$C_AREA,NML$GL_NML_ENTITY
0000    596         $BYTE   NMA$C_ENT_ACT,NPA$_EXIT,NML$PRSIDLEQ, -
0000    597                         NML$C_AREA,NML$GL_NML_ENTITY
0000    598         $ERROR  NML$_STS_IDE,,NML$PRSIDERR,,,NMA$C_ENT_ARE; Illegal area ID
0000    599
0000    600         FIELD$  NML_AREA_NUM
0000    601         $MATCH  2,NPA$_EXIT,NML$PRSID, -            ; Save 1 byte area number
0000    602                                 NML$C_AREA,NML$GL_NML_ENTITY
0000    603         $NULL   ,NML_FOR_ERR                        ; Must be at least 2 bytes
```

B C D E F G H I J K L M N B C D E F G H I J K L M N B C D E F G H I J K L M N B C D E F G H I

```
0000    605 ;
0000    606 ; Parse an object entity (VMS-specific)
0000    607 ;
0000    608         FIELD$  NML_OBJ_ENT                         ; Object entity
0000    609         $EOM     ,NML_FOR_ERR                       ; Message format error
0000    610         $MATCH   1,NML_OBJECT,NML$PRSIDLEQ          ; 1 byte if zero or negative
0000    611         $IMAGE   12,NML_OBJECT,NML$PRSIDN           ; n bytes (name)
0000    612         $ERROR   NML$_STS_IDE,,NML$PRSIDERR,,,NMA$C_SENT_OBJ ; Illegal object name
0000    613
0000    614         FIELD$  NML_OBJECT                          ; Set internal NML entity type
0000    615         $NULL    ,NPA$_EXIT,,NML$C_OBJECT,NML$GL_NML_ENTITY
0000    616
0000    617 ;
0000    618 ; Parse a link entity (VMS-specific)
0000    619 ;
0000    620         FIELD$  NML_LNK_ENT                         ; Link entity
0000    621         $EOM     ,NML_FOR_ERR                       ; Message format error
0000    622         $NULL    ,NML_LINK,,NML$C_LINKS,NML$GL_NML_ENTITY
0000    623
0000    624         FIELD$  NML_LINK
0000    625         $LOOK    0,NML_LNK_NUM                      ; If zero, it's a link number.
0000    626         $MATCH   1,NML_LNK_NOD,NML$PRSIDLEQ         ; 1 byte if negative
0000    627
0000    628         FIELD$  NML_LNK_NUM                         ; Link address
0000    629         $MATCH   3,NML_LNK_NOD,NML$PRSID            ; Must be byte of 0 and 2 bytes
0000    630                                                     ;   of link number.
0000    631         $ERROR   NML$_STS_IDE,,NML$PRSIDERR,,,NMA$C_SENT_LNK ; Illegal link address
0000    632
0000    633         FIELD$  NML_LNK_NOD
0000    634         $EOM     ,NPA$_EXIT
0000    635         $WORD    NMA$C_PCLK_NID,NML_LNK_NID,, -     ; Flag node qualifier
0000    636                                  NML$M_PRS_QUALIFIER,NML$GL_PRS_FLGS
0000    637         $NULL    ,NML_FOR_ERR
0000    638
0000    639         FIELD$  NML_LNK_NID
0000    640         $EOM     ,NML_FOR_ERR
0000    641         $SBEXP   NML_NODEID_SUB,NML_CHKEOM
0000    642         $ERROR   NML$_STS_IDE,,NML$PRSIDERR,,,NMA$C_ENT_NOD ; Illegal node id
0000    643
0000    644
0000    645         FIELD$  NML_NODEID_SUB
0000    646         $LOOK    0,NML_NODNUM
0000    647         $IMAGE   6,NPA$_EXIT,NML$PRSIDN,,, -        ; Qualifier is a node name.
0000    648                                  CPT$GK_PCLK_NID
0000    649
0000    650         FIELD$  NML_NODNUM
0000    651         $MATCH   3,NPA$_EXIT,NML$PRS_NODE_NUM,,, - ; Qualifier is a node address.
0000    652                                  CPT$GK_PCLK_PN$
0000    653
0000    654         FIELD$
```

```
                        .SBTTL  Common expressions
                ;
                ; Check the option byte for illegal bits and save it
                ;
                        FIELD$  NML_CHKOPT_SUB                  ; Check for valid option
                        $EOM    ,NML_FOR_ERR                    ; Message format error
                        $LOOK   ,NPA$_EXIT,NML$PRSOPT           ;
                        $NULL   ,NML_FUN_ERR                    ; Unrecognized option
                ;
                ; No more bytes allowed in message
                ;
                        FIELD$  NML_CHKEOM                      ; Check for end of message
                        $EOM    ,NPA$_EXIT                      ; If end then success
                        $NULL   ,NML_FOR_ERR                    ; Message format error
                ;
                ; Error exits
                ;
                        FIELD$  NML_FOR_ERR                     ; Message format error
                        $ERROR  NML$_STS_INV,,NML$PRSERR1,,,NMA$C_STS_INV

                        FIELD$  NML_FUN_ERR                     ; Illegal function or option
                        $ERROR  NML$_STS_FUN,,NML$PRSERR1,,,NMA$C_STS_FUN

                        FIELD$  NML_SYS_ERR                     ; Unrecognized system-specific
                        $ERROR  NML$_STS_SYS,,NML$PRSERR1,,,NMA$C_STS_SYS

                        FIELD$                                  ; End of Phase III tables
                ;
                ; End of parse tables
                ;
                        .END
```

```
CPT$GK_PCCI_ADJ                 ********  X   03
CPT$GK_PCCI_AJ$                 ********  X   03
CPT$GK_PCLK_NID                 ********  X   03
CPT$GK_PCLK_PN$                 ********  X   03
CPT$GK_PCNO_DLI                 ********  X   03
CPT$GK_PCXP_GDT                 ********  X   03
FLG$$$                        = FFFFFFFF
NMA$C_ENT_ACT                 = FFFFFFFE
NMA$C_ENT_ADJ                 = FFFFFFFC
NMA$C_ENT_ARE                 = 00000005
NMA$C_ENT_CIR                 = 00000003
NMA$C_ENT_KNO                 = FFFFFFFF
NMA$C_ENT_LIN                 = 00000001
NMA$C_ENT_LOG                 = 00000002
NMA$C_ENT_LOO                 = FFFFFFFD
NMA$C_ENT_MOD                 = 00000004
NMA$C_ENT_NOD                 = 00000000
NMA$C_FNC_CHA                 = 00000013
NMA$C_FNC_DUM                 = 00000010
NMA$C_FNC_LOA                 = 0000000F
NMA$C_FNC_REA                 = 00000014
NMA$C_FNC_SYS                 = 00000016
NMA$C_FNC_TES                 = 00000012
NMA$C_FNC_TRI                 = 00000011
NMA$C_FNC_ZER                 = 00000015
NMA$C_OPINF_CHA               = 00000002
NMA$C_OPINF_COU               = 00000003
NMA$C_OPINF_EVE               = 00000004
NMA$C_OPINF_STA               = 00000001
NMA$C_OPINF_SUM               = 00000000
NMA$C_PCCI_ADJ                = 00000320
NMA$C_PCCN_CIR                = 00000064
NMA$C_PCLK_NID                = 00000066
NMA$C_PCLO_SIN                = 000000C8
NMA$C_PCNO_DLI                = 00000336
NMA$C_PCXA_NET                = 00000456
NMA$C_PCXP_DTE                = 0000044C
NMA$C_PCXP_GDT                = 00000492
NMA$C_PCXP_GRP                = 0000044D
NMA$C_PCXS_DST                = 0000012C
NMA$C_PCXT_TPT                = 0000006A
NMA$C_SENT_LNK                = 00000007
NMA$C_SENT_OBJ                = 00000004
NMA$C_STS_FUN                 = FFFFFFFF
NMA$C_STS_INV                 = FFFFFFFE
NMA$C_STS_SYS                 = FFFFFFE6
NMA$C_SYS_VMS                 = 00000004
NMA$S_OPT_ENT                 = 00000003
NMA$S_OPT_INF                 = 00000003
NMA$V_OPT_ENT                 = 00000000
NMA$V_OPT_INF                 = 00000004
NML$C_ADJACENT_NODE           = 00000006
NML$C_AREA                    = 0000000C
NML$C_CIRCUIT                 = 00000009
NML$C_EXECUTOR                = 00000007
NML$C_LINE                    = 00000000
NML$C_LINKS                   = 00000018
NML$C_LOGGING                 = 00000001
NML$C_LOOPNODE                = 00000005
NML$C_NI_CONFIG               = 00000017
NML$C_NODE                    = 00000003
NML$C_NODEBYNAME              = 00000004
NML$C_OBJECT                  = 00000008
NML$C_PROTOCOL                = 00000019
NML$C_PROT_DTE                = 0000000F
NML$C_PROT_GRP                = 00000010
NML$C_PROT_NET                = 0000000E
NML$C_TRACE                   = 00000013
NML$C_TRACEPNT                = 00000014
NML$C_X25_ACCESS              = 0000000D
NML$C_X25_SERV                = 00000011
NML$C_X25_SERV_DEST           = 00000012
NML$C_X29_SERV                = 00000015
NML$C_X29_SERV_DEST           = 00000016
NML$GL_NML_ENTITY               ********  X   03
NML$GL_PRS_FLGS                 ********  X   03
NML$M_PRS_RNOSNK              = 00000400
NML$M_PRS_QUALIFIER           = 00000004
NML$M_PRS_SNKNOD              = 00000200
NML$M_PRS_VMS                 = 00000001
NML$NPA_INIT                    00000000 RG    03
NML$PRSDEVICE                   ********    X  03
NML$PRSENT                      ********    X  03
NML$PRSERR1                     ********    X  03
NML$PRSEXESNK                   ********    X  03
NML$PRSFNC                      ********    X  03
NML$PRSID                       ********    X  03
NML$PRSIDERR                    ********    X  03
NML$PRSIDLEQ                    ********    X  03
NML$PRSIDN                      ********    X  03
NML$PRSINF                      ********    X  03
NML$PRSLOGSIN                   ********    X  03
NML$PRSNODNAM                   ********    X  03
NML$PRSOPT                      ********    X  03
NML$PRSQUALLEQ                  ********    X  03
NML$PRSSNKNAD                   ********    X  03
NML$PRSSNKNNA                   ********    X  03
NML$PRS_ACTIVE_NET              ********    X  03
NML$PRS_MODULE                  ********    X  03
NML$PRS_NODE_NUM                ********    X  03
NML$PRS_NODE_NUM_ENTITY         ********    X  03
NML$PRS_NOREAD                  ********    X  03
NML$_STS_FUN                  = FFFFFFFE
NML$_STS_IDE                  = FFFFFFEE
NML$_STS_INV                  = FFFFFFFC
NML$_STS_SYS                  = FFFFFFCC
NML_ADJ_EXE                     00000600 R     03
NML_ADJ_NOD_NUM                 000005E0 R     03
NML_AREA_ENT                    00000980 R     03
NML_AREA_NUM                    000009D8 R     03
NML_CHA                         000000B4 R     03
NML_CHECK_QUAL                  000003D0 R     03
NML_CHKENT_SUB                  00000220 R     03
NML_CHKEOM                      00000B38 R     03
```

```
NML_CHKOPT_SUB                  00000B1C R      03
NML_CHK_DTE                     00000798 R      03
NML_CHK_INFO_SUB                000001C8 R      03
NML_CHK_VMS_ENT_SUB             00000288 R      03
NML_CIRCUIT                     0000056C R      03
NML_CIRCUIT_ENT                 00000520 R      03
NML_CIRC_ADJACENCY              00000588 R      03
NML_CIRC_QUAL                   000003F8 R      03
NML_DTE_QUAL                    000007E8 R      03
NML_DUM                         00000090 R      03
NML_FOR_ERR                     00000B48 R      03
NML_FUN_ERR                     00000B5C R      03
NML_GRP_DTE_ID                  000007C0 R      03
NML_INV_DEST                    00000700 R      03
NML_LINE                        0000045C R      03
NML_LINK                        00000A5C R      03
NML_LIN_ENT                     00000420 R      03
NML_LNK_ENT                     00000A44 R      03
NML_LNK_NID                     00000AC0 R      03
NML_LNK_NOD                     00000A9C R      03
NML_LNK_NUM                     00000A78 R      03
NML_[illegible]A                00000084 R      03
NML_LOG_ENT                     0000046C R      03
NML_LOG_NID                     000004B0 R      03
NML_LOG_NIDSUB                  000004D8 R      03
NML_LOG_NODNUM                  00000510 R      03
NML_LOG_SIN                     0000048C R      03
NML_MODULE_ENT                  00000628 R      03
NML_MOD_ENT                     0000094C R      03
NML_MOD_NI_CONFIG               00000910 R      03
NML_MOD_PROTOCOL                00000714 R      03
NML_MOD_X25_ACCESS              000006CC R      03
NML_MOD_X25_SERVER              00000810 R      03
NML_MOD_X25_TRACE               00000854 R      03
NML_MOD_X29_SERVER              000008CC R      03
NML_NI_CONFIG_CIRC              00000944 R      03
NML_NODEID_SUB                  00000AE8 R      03
NML_NODE_CIRC_QUAL              0000039C R      03
NML_NODNUM                      00000B08 R      03
NML_NOD_ENT                     000002B0 R      03
NML_NOD_ERR                     00000614 R      03
NML_NOD_EXE                     00000380 R      03
NML_NOD_NUM                     00000350 R      03
NML_OBJECT                      00000A34 R      03
NML_OBJ_ENT                     000009F8 R      03
NML_PROTCL_DTE                  00000750 R      03
NML_PROTCL_GRP                  00000768 R      03
NML_PROTCL_NET                  0000073C R      03
NML_PRS_ADJ                     000005AC R      03
NML_REA                         000000C8 R      03
NML_SYS                         00000134 R      03
NML_SYS_ERR                     00000B70 R      03
NML_TES                         000000A8 R      03
NML_TPT_ENT                     00000898 R      03
NML_TRACEPOINT                  00000880 R      03
NML_TRI                         0000009C R      03
NML_VMS                         00000158 R      03
NML_VMS_CHA                     00000198 R      03
NML_VMS_REA                     000001AC R      03
NML_X25_ACCESS_NET              000006E8 R      03
NML_X25_SERV_DEST               0000083C R      03
NML_X29_SERV_DEST               000008F8 R      03
NML_ZER                         000000E4 R      03
NPA$M_ACTION                    =  00000004
NPA$M_EXT                       =  00000001
NPA$M_LAST                      =  00008000
NPA$M_MASK                      =  00000010
NPA$M_MSKADR                    =  00000020
NPA$M_OFFSET                    =  00000040
NPA$M_PARAM                     =  00000002
NPA$M_STATE                     =  00000008
NPA$_ADVANCE                    =  00000001
NPA$_BYTE                       =  00000003
NPA$_EOM                        =  00000004
NPA$_ERROR                      =  00000007
NPA$_EXIT                       =  00000000
NPA$_EXTZV                      =  0000000A
NPA$_FAIL                       =  FFFFFFFF
NPA$_IMAGE                      =  00000000
NPA$_LOOK                       =  00000009
NPA$_MASK                       =  00000002
NPA$_MATCH                      =  00000008
NPA$_NULL                       =  00000005
NPA$_SBEXP                      =  00000006
NPA$_WORD                       =  00000001
NXT$$$                          =  00000000
```

Psect synopsis

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
. BLANK .	00000000 (0.)	01 (1.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE
ABS	00000000 (0.)	02 (2.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
NPA$STATE	00000B84 (2948.)	03 (3.)	NOPIC USR CON REL LCL NOSHR NOEXE RD NOWRT NOVEC BYTE

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	32	00:00:00.12	00:00:00.42
Command processing	137	00:00:00.91	00:00:05.23
Pass 1	635	00:00:41.54	00:01:28.34
Symbol table sort	0	00:00:01.68	00:00:02.76
Pass 2	145	00:00:07.50	00:00:16.92
Symbol table output	24	00:00:00.20	00:00:00.73
Psect synopsis output	2	00:00:00.03	00:00:00.12
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	977	00:00:51.98	00:01:54.52

The working set limit was 1950 pages.
197884 bytes (387 pages) of virtual memory were used to buffer the intermediate code.
There were 70 pages of symbol table space allocated to hold 1130 non-local and 0 local symbols.
686 source lines were read in Pass 1, producing 34 object records in Pass 2.
35 pages of virtual memory were used to define 32 macros.

Macro library statistics

Macro library name	Macros defined
_255DUA28:[SHRLIB]NMALIBRY.MLB;1	1
_255DUA28:[SYS.OBJ]LIB.MLB;1	0
_255DUA28:[NML.OBJ]NMLLIB.MLB;1	18
_255DUA28:[SYSLIB]STARLET.MLB;2	3
TOTALS (all libraries)	22

1357 GETS were required to define 22 macros.

There were no errors, warnings or information messages.

```
MACRO/LIS=LIS$:NMLINISTA/OBJ=OBJ$:NMLINISTA MSRC$:NMLINISTA/UPDATE=(ENH$:NMLINISTA)+LIB$:NMLLIB/LIB+EXECML$/LIB+SHRLIB$:NMALIBRY/LIB
```

NMLFORWRD
LIS

NMLENTRY
LIS

NMLDEFINE
LIS

NMLFILEIO
LIS

NMLLIB
LIS

NMLINISTA
LIS

NMLDISC
LIS