N 6

```
NNN         NNN     CCCCCCCCCCCC  PPPPPPPPPPPP
NNN         NNN     CCCCCCCCCCCC  PPPPPPPPPPPP
NNN         NNN     CCCCCCCCCCCC  PPPPPPPPPPPP
NNN         NNN  CCC              PPP         PPP
NNN         NNN  CCC              PPP         PPP
NNN         NNN  CCC              PPP         PPP
NNNNNN      NNN  CCC              PPP         PPP
NNNNNN      NNN  CCC              PPP         PPP
NNNNNN      NNN  CCC              PPP         PPP
NNN   NNN   NNN  CCC              PPPPPPPPPPPP
NNN   NNN   NNN  CCC              PPPPPPPPPPPP
NNN   NNN   NNN  CCC              PPPPPPPPPPPP
NNN      NNNNNN  CCC              PPP
NNN      NNNNNN  CCC              PPP
NNN      NNNNNN  CCC              PPP
NNN         NNN  CCC              PPP
NNN         NNN  CCC              PPP
NNN         NNN  CCC              PPP
NNN         NNN     CCCCCCCCCCCC  PPP
NNN         NNN     CCCCCCCCCCCC  PPP
NNN         NNN     CCCCCCCCCCCC  PPP
```

M 12

```
**FILE**ID**NMADEF
```

```
NN      NN  MM      MM    AAAAAA    DDDDDDDD    EEEEEEEEEE  FFFFFFFFFF
NN      NN  MM      MM    AAAAAA    DDDDDDDD    EEEEEEEEEE  FFFFFFFFFF
NN      NN  MMMM  MMMM  AA      AA  DD      DD  EE          FF
NN      NN  MMMM  MMMM  AA      AA  DD      DD  EE          FF
NNNN    NN  MM  MM  MM  AA      AA  DD      DD  EE          FF
NNNN    NN  MM  MM  MM  AA      AA  DD      DD  EE          FF
NN  NN  NN  MM      MM  AA      AA  DD      DD  EEEEEEEE    FFFFFFFF
NN  NN  NN  MM      MM  AA      AA  DD      DD  EEEEEEEE    FFFFFFFF
NN    NNNN  MM      MM  AAAAAAAAAA  DD      DD  EE          FF
NN    NNNN  MM      MM  AAAAAAAAAA  DD      DD  EE          FF
NN      NN  MM      MM  AA      AA  DD      DD  EE          FF              ....
NN      NN  MM      MM  AA      AA  DD      DD  EE          FF              ....
NN      NN  MM      MM  AA      AA  DDDDDDDD    EEEEEEEEEE  FF              ....
NN      NN  MM      MM  AA      AA  DDDDDDDD    EEEEEEEEEE  FF              ....


  SSSSSSSS  DDDDDDDD    LL
  SSSSSSSS  DDDDDDDD    LL
SS          DD      DD  LL
SS          DD      DD  LL
SS          DD      DD  LL
SS          DD      DD  LL
  SSSSSS    DD      DD  LL
  SSSSSS    DD      DD  LL
        SS  DD      DD  LL
        SS  DD      DD  LL
        SS  DD      DD  LL
        SS  DD      DD  LL
SSSSSSSS    DDDDDDDD    LLLLLLLLLL
SSSSSSSS    DDDDDDDD    LLLLLLLLLL
```

```
{       .TITLE  NMADEF          Network Management Definitions
{
{ Version:      'V04-000'
{
{****************************************************************************
{*                                                                          *
{*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
{*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
{*  ALL RIGHTS RESERVED.                                                    *
{*                                                                          *
{*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
{*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
{*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
{*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
{*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
{*  TRANSFERRED.                                                            *
{*                                                                          *
{*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
{*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
{*  CORPORATION.                                                            *
{*                                                                          *
{*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
{*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
{*                                                                          *
{*                                                                          *
{****************************************************************************

{++
{ FACILITY:     DECnet-VAX Network Management Components
{
{ ABSTRACT:
{
{       Common Definitions for Network Management Components
{
{ ENVIRONMENT:  VAX/VMS Operating System
{
{ AUTHOR:       Darrell Duffy   , CREATION DATE: 3-October-1979
{
{ MODIFIED BY:
{
{       V03-043 PRD0092         Paul R. DeStefano       06-Apr-1984
{               Added values for operation failure error detail.
{
{       V03-042 PRD0085         Paul R. DeStefano       29-Mar-1984
{               Correct values for X.25 Access module parameters.
{               Add data type of NMA$C_PTY_H4.
{
{       V03-041 TMK0003         Todd M. Katz            17-Jan-1984
{               Add NMA$C_LINMC_SDF. This address qualifier specifies
{               that the physical address for the ETHERnet controller
{               should be setup as the default ETHERnet address.
{
{       V03-040 PRD0044         Paul R. DeStefano       05-Jan-1984
{               Add SERVICE NODE VERSION parameter.
{
{       V03-039 TMK0002         Todd M. Katz            11-Nov-1983
{               Remove NMA$C_LINCN_ILO as a valid line controller mode.
{               Instead add the line parameter NMA$C_PCLI_ILP. This line
{               parameter can only be used to set the loopback mode of
{               a DELUA to internal.
{
{       V03-038 TMK0001         Todd M. Katz            08-Nov-1983
{               Add NMA$C_LINCN_ILO (internal loop) as a valid line
{               controller mode. This mode is used only by the DELUA
{               at the present time.
{
{       V03-037 TMH0037         Tim Halvorsen           13-Jul-1983
{               Add EXECUTOR ALIAS parameter (VMS specific).
{
{       V03-036 RPG0034         Bob Grosso              23-Jun-1983
{               Add a bunch of codes for Meg, such as protocol
{               Bisync.
{
{       V03-035 MKP0001         Kathy Perko             30-April-1983
{               Add PCCI_SBB (circuit service substate) and PCCI_SPY
{               (circuit service physical address).
{
{       V03-034 RPG0034         Bob Grosso              22-Mar-1983
{               Add PCNO_LPN and PCNO_LAN for LOOP CIRCUIT
{
{       V03-033 RNG0033         Rod Gamache             14-Mar-1983
{               Changed value on PCLI_DES parameter.
{
{       V03-032 RPG0032         Bob Grosso              10-Mar-1983
{               Add PCLI_DES.
{
{       V03-031 RPG0031         Bob Grosso              25-Feb-1983
{               Add OPN_CNF for the NML CONFIGURATOR data base.
{
{       V03-030 RPG0030         Bob Grosso              07-Feb-1983
{               Add "REQUIRED" to node access.
{               Add symbols for RSX and SERVER BASE system specific
{               parameters.
{               Change LOOP_DSIZ from 128 to 40.
{               Add NMA$C_SOFT_DIAG.
{               Change PTY_TYP to 15 bits from 12 to permit the
{               large parameter IDs of CONFIGURATOR MODULE.
{               Add codes for months of the year.
{
{       V03-029 RPG0029         Bob Grosso              11-Jan-1983
{               Correct value for PCLI_BSZ, Device Buffer Size
{
{       V03-028 RPG0028         Bob Grosso              05-Jan-1983
{               Add PCLI_BSZ, Device Buffer Size
{
{       V03-027 RPG0027         Bob Grosso              15-Dec-1982
{               Add VMS-specific parameter, PCLI_EPT, LINE ETHERNET
{               protocol type.
{               Add symbol for data type HEX WORD.
{
{       V03-026 RPG0026         Bob Grosso              19-Nov-1982
{               Supply extraction macros for node area and address.
{               Remove LINPR_X25, CIRTY_LAPB, CIRTY_LAP.
{
{       V03-025 RPG0025         Bob Grosso              12-Nov-1982
{               Rename endnodes to nonrouters.
{               Correct values of CIRTY_X25, CIRTY_LAPB and LINTY_X25,
{               LINTY_LAPB.
{
{       V03-024 RPG0023         Bob Grosso              11-Oct-1982
{               Reinstate CIRXPT_PH3.
{
{       V03-023 RPG0023         Bob Grosso              28-Sep-1982
{               Add code for Area
{
{       V03-022 RPG0022         Bob Grosso              14-Sep-1982
{               Re-activate CIRTY_X25 and LINPR_X25.
{
{       V03-021 RPG0021         Bob Grosso              03-Sep-1982
{               Add coded values for circuit type.
{               Fix up circuit and line type values and comments.
{               Change DTE substate values to match circuit/line substate
{               values so that show/list works synergisticly.
{
{       V03-020 TMH0020         Tim Halvorsen           18-Aug-1982
{               Add coded values for DTE substate.
{               Add UNA Echo Mode line parameter.
{
{       V03-019 RPG0019         Bob Grosso              02-Aug-1982
{               Add line counter flags.
{               Add X25-Protocol DTE Maximum Circuits code.
{               Add X25-Protocol DTE Substate code.
{               Add permanent database file ID codes OPN_X25 and OPN_X29.
{
{       V018    TMH0018         Tim Halvorsen   17-Jun-1982
{               Fix typo in PCL "Secondary" protocol value.
{
{       V017    TMH0017         Tim Halvorsen   07-Jun-1982
{               Add NI protocol sharing parameter, and values.
{
{       V016    RNG001          Rod Gamache     03-Jun-1982
{               Add extra Ethernet parameters.
{
{       V015    TMH0015         Tim Halvorsen   29-Mar-1982
{               Expand parameter data type definitions.
{               Add parameters to support Ethernet.
{
{       V014    TMH0014         Tim Halvorsen   25-Feb-1982
{               Add extra parameters for X.25 support.
{
{       V013    TMH0013         Tim Halvorsen   20-Jan-1982
{               Fix classification of MST circuit parameter to
{               correctly indicate that it is a datalink only
{               parameter, rather than a NICE parameter.
{               Document the format of each coded parameter.
{
{       V012    TMH0012         Tim Halvorsen   31-Dec-1981
{               Add DMF-32 as a service device.
```

```
{
{       V011    TMH0011         Tim Halvorsen   28-Dec-1981
{               Add PCL datalink parameters and counters.  Remove
{               previous PCL parameters which are now obsolete.
{
{       V010    TMH0010         Tim Halvorsen   1-Dec-1981
{               Add proxy parameters to executor, node and object
{               entities.
{
{       V009    TMH0009         Tim Halvorsen   11-Nov-1981
{               Add LINE RETRANSMIT timer parameter.
{               Add LINK REMOTE IDENTIFICATION parameter.
{
{       V008    TMH0008         Tim Halvorsen   04-Nov-1981
{               Add circuit transport type parameter.
{               Add UNA driver datalink-only parameter/counter ID's.
{
{       V007    RNG0007         Rod N. Gamache  28-Sep-1981
{               Add Maintenance state as P2 parameter.
{
{       V006    LMK0006         Len Kawell      27-Sep-1981
{               Modify for Network Management V3.0.
{
{       V005    TMH0005         Tim Halvorsen   28-Aug-1981
{               Add VMS-specific line parameters BFS, NMS.
{
{       V004    TMH0004         Tim Halvorsen   15-Aug-1981
{               Add DMP, DMV, DPV for MOP device classes.
{               Add system-specific link parameters.
{
{       V003    TMH0003         Tim Halvorsen   05-Aug-1981
{               Change RETRANSMIT TIMER from a line parameter to a
{               circuit parameter.
{
{       V002    TMH0002         Tim Halvorsen   27-Jul-1981
{               Fix misc. typos and re-interpretations from the network
{               management spec.  Add PCL11-B line counters.
{               Add new permanent database IDs.
{               Add CIRCUIT VERIFICATION, NODE ACCESS, DEFAULT ACCESS,
{               and PIPELINE QUOTA, all VMS system-specific parameters.
{
{       V001    TMH0001         Tim Halvorsen   10-Jun-1981
{               Add definitions for DNA V2.2 NICE.  Renamed to NMADEF.MDL
{               to allow concurrent development of 2.0 and 2.2 software.
{--
```

```
{
{       Symbols for the Network Management Layer of DECnet-VAX
{

module $NMADEF;


/*
/* Object type
/*

constant OBJ_NIC          equals 19  prefix NMA tag $C;   /* Nice listener

/*
/* Function codes
/*

constant FNC_LOA          equals 15  prefix NMA tag $C;   /* Request down-line load
constant FNC_DUM          equals 16  prefix NMA tag $C;   /* Request up-line dump
constant FNC_TRI          equals 17  prefix NMA tag $C;   /* Trigger bootstrap
constant FNC_TES          equals 18  prefix NMA tag $C;   /* Test
constant FNC_CHA          equals 19  prefix NMA tag $C;   /* Change parameter
constant FNC_REA          equals 20  prefix NMA tag $C;   /* Read information
constant FNC_ZER          equals 21  prefix NMA tag $C;   /* Zero counters
constant FNC_SYS          equals 22  prefix NMA tag $C;   /* System-specific function
```

```
/*
/* Option byte
/*
/*     common to change parameter, read information and zero counters
/*


aggregate NMADEF  union fill prefix NMA$;
    NMADEF_BITS0 structure fill;
        OPT_ENT bitfield mask length 3;                 /* Entity type

        FILL_1 bitfield length 3 fill prefix NMADEF tag $$;

/*
/*     change parameter
/*

        OPT_CLE bitfield mask;                          /* Clear parameter

/*
/*     common to change parameter or read information
/*

        OPT_PER bitfield mask;                          /* Permanent parameters

    end NMADEF_BITS0;

/*
/*     read information
/*

    NMADEF_BITS1 structure fill;
        FILL_2 bitfield length 4 fill prefix NMADEF tag $$;
        OPT_INF bitfield mask length 3;                 /* Information type mask

    end NMADEF_BITS1;


    constant OPINF_SUM  equals 0  prefix NMA tag $C;    /* Summary
    constant OPINF_STA  equals 1  prefix NMA tag $C;    /* Status
    constant OPINF_CHA  equals 2  prefix NMA tag $C;    /* Characteristics
    constant OPINF_COU  equals 3  prefix NMA tag $C;    /* Counters
    constant OPINF_EVE  equals 4  prefix NMA tag $C;    /* Events


/*
/*     test
/*

    NMADEF_BITS2 structure fill;
        FILL_3 bitfield length 7 fill prefix NMADEF tag $$;
        OPT_ACC bitfield mask;                          /* Access control included
    end NMADEF_BITS2;

/*
/*    zero
/*

    NMADEF_BITS3 structure fill;
        FILL_4 bitfield length 7 fill prefix NMADEF tag $$;
        OPT_REA bitfield mask;                          /* Read and zero

    end NMADEF_BITS3;
```

```
/*
/* System types
/*

    constant SYS_RST     equals 1  prefix NMA tag $C;     /* Rsts
    constant SYS_RSX     equals 2  prefix NMA tag $C;     /* Rsx family
    constant SYS_TOP     equals 3  prefix NMA tag $C;     /* Tops-20
    constant SYS_VMS     equals 4  prefix NMA tag $C;     /* Vms
    constant SYS_RT      equals 5  prefix NMA tag $C;     /* RT-11

/*
/* Entity types.  This numbering scheme must be used in non-system-specific
/* NICE messages.  (See below for conflicting system-specific entities).
/*

    constant ENT_NOD     equals 0  prefix NMA tag $C;     /* Node
    constant ENT_LIN     equals 1  prefix NMA tag $C;     /* Line
    constant ENT_LOG     equals 2  prefix NMA tag $C;     /* Logging
    constant ENT_CIR     equals 3  prefix NMA tag $C;     /* Circuit
    constant ENT_MOD     equals 4  prefix NMA tag $C;     /* Module
    constant ENT_ARE     equals 5  prefix NMA tag $C;     /* Area

/*
/* System-specific (function 22) entity types.  This numbering scheme
/* for objects must be used in any entity type in system-specific NICE
/* messages.
/*
    constant SENT_ALI    equals 3  prefix NMA tag $C;     /* Alias
    constant SENT_OBJ    equals 4  prefix NMA tag $C;     /* Object
    constant SENT_PRO    equals 5  prefix NMA tag $C;     /* Process
    constant SENT_SYS    equals 6  prefix NMA tag $C;     /* System
    constant SENT_LNK    equals 7  prefix NMA tag $C;     /* Links

    NMADEF_BITS4 structure fill;
        FILL_5 bitfield length 7 fill prefix NMADEF tag $$;
        ENT_EXE bitfield mask;                            /* Executor indicator flag for response messages

    end NMADEF_BITS4;

/*
/* Entity identification format types
/*


    constant ENT_ADJ     equals -4  prefix NMA tag $C;    /* Adjacent
    constant ENT_ACT     equals -2  prefix NMA tag $C;    /* Active
    constant ENT_KNO     equals -1  prefix NMA tag $C;    /* Known
    constant ENT_ADD     equals 0  prefix NMA tag $C;     /* Node address
    constant ENT_ALL     equals -3  prefix NMA tag $C;    /* All
    constant ENT_LOO     equals -3  prefix NMA tag $C;    /* Loop
```

```
/*
/* Logging sink types
/*


    constant SNK_CON     equals 1  prefix NMA tag $C;     /* Console
    constant SNK_FIL     equals 2  prefix NMA tag $C;     /* File
    constant SNK_MON     equals 3  prefix NMA tag $C;     /* Monitor


/*
/* Counter data type values
/*

    NMADEF_BITS5 structure fill;
        CNT_TYP bitfield mask length 12;                  /* Type mask
        CNT_MAP bitfield mask;                            /* Bitmapped indicator
        CNT_WID bitfield mask length 2;                   /* Width field mask
        CNT_COU bitfield mask;                            /* Counter indicator

    end NMADEF_BITS5;


    NMADEF_BITS6 structure fill;
        FILL_6 bitfield length 13 fill prefix NMADEF tag $$;
        CNT_WIL bitfield mask;                            /* Width field low bit
        CNT_WIH bitfield mask;                            /* Width field high bit

    end NMADEF_BITS6;
```

```
/*
/* Node area and address extraction
/*
end NMADEF;

    aggregate NMADEF1 union fill prefix NMA$;
    NODE word unsigned;
        NODE_BITS0 structure fill;
            ADDR bitfield length 10;
            AREA bitfield length 6;
        end NODE_BITS0;

/*
/* Parameter ID word (DATA ID)
/*

        NODE_BITS1 structure fill;
            PTY_TYP bitfield mask length 15;            /* Type mask
        end NODE_BITS1;

/*
/* Parameter data type byte (DATA TYPE)
/*

        constant PTY_MAX         equals 15  prefix NMA tag $C;/* Maximum fields within coded multiple

        NODE_BITS2 structure fill;
            PTY_CLE bitfield mask length 6;             /* Coded length mask
            PTY_MUL bitfield mask;                      /* Coded multiple indicator
            PTY_COD bitfield mask;                      /* Coded indicator
        end NODE_BITS2;

        NODE_BITS3 structure fill;
            FILL_7 bitfield length 6 fill prefix NMADEF tag $$;
            PTY_CMU bitfield mask length 2;             /* Coded multiple
        end NODE_BITS3;

        NODE_BITS4 structure fill;
            PTY_NLE bitfield mask length 4;             /* Number length mask
            PTY_NTY bitfield mask length 2;             /* Number type mask
            PTY_ASC bitfield mask;                      /* Ascii image indicator
        end NODE_BITS4;

                                                        /* NTY values (how to display number):
        constant NTY_DU equals 0  prefix NMA tag $C;    /* Unsigned decimal
        constant NTY_DS equals 1  prefix NMA tag $C;    /* Signed decimal
        constant NTY_H  equals 2  prefix NMA tag $C;    /* Hexidecimal
        constant NTY_O  equals 3  prefix NMA tag $C;    /* Octal

                                                        /* NLE values (length of number):
        constant NLE_IMAGE       equals 0  prefix NMA tag $C;/* Image field (byte-counted)
        constant NLE_BYTE        equals 1  prefix NMA tag $C;/* Byte
        constant NLE_WORD        equals 2  prefix NMA tag $C;/* Word
        constant NLE_LONG        equals 4  prefix NMA tag $C;/* Longword
        constant NLE_QUAD        equals 8  prefix NMA tag $C;/* Quadword

/*
/* Define standard values for the DATA TYPE byte
/*

        constant PTY_AI equals 64  prefix NMA tag $C;    /* ASCII image (ASC=1)
        constant PTY_HI equals 32  prefix NMA tag $C;    /* Hex image (NTY=H, NLE=IMAGE)
        constant PTY_H1 equals 33  prefix NMA tag $C;    /* Hex byte (NTY=H, NLE=BYTE)
        constant PTY_H2 equals 34  prefix NMA tag $C;    /* Hex word (NTY=H, NLE=WORD)
        constant PTY_H4 equals 36  prefix NMA tag $C;    /* Hex byte (NTY=H, NLE=LONG)
        constant PTY_DU1         equals 1  prefix NMA tag $C;/* Decimal unsigned byte (NTY=DU,NLE=BYTE)
        constant PTY_DU2         equals 2  prefix NMA tag $C;/* Decimal unsigned word (NTY=DU,NLE=WORD)
        constant PTY_CD1         equals 129  prefix NMA tag $C;/* Coded decimal byte (COD=1, 1 byte)
        constant PTY_CM2         equals 194  prefix NMA tag $C;/* Coded multiple, 2 fields
        constant PTY_CM3         equals 195  prefix NMA tag $C;/* Coded multiple, 3 fields
        constant PTY_CM4         equals 196  prefix NMA tag $C;/* Coded multiple, 4 fields
        constant PTY_CM5         equals 197  prefix NMA tag $C;/* Coded multiple, 5 fields
```

```
/*
/*      Circuit parameters
/*

        constant PCCI_STA       equals 0  prefix NMA tag $C;/* State (coded byte of NMA$C_STATE_)
        constant PCCI_SUB       equals 1  prefix NMA tag $C;/* Substate (coded byte of NMA$C_LINSS_)
        constant PCCI_SER       equals 100  prefix NMA tag $C;/* Service (coded byte of NMA$C_LINSV_)
        constant PCCI_LCT       equals 110  prefix NMA tag $C;/* Counter timer (word)
        constant PCCI_SPY       equals 120  prefix NMA tag $C;/* Service physical address (NI address)
        constant PCCI_SSB       equals 121  prefix NMA tag $C;/* Service substate (coded byte of NMA$C_LINSS_)
        constant PCCI_CNO       equals 200  prefix NMA tag $C;/* Connected node
        constant PCCI_COB       equals 201  prefix NMA tag $C;/* Connected object
        constant PCCI_LOO       equals 400  prefix NMA tag $C;/* Loopback name (ascic)
        constant PCCI_ADJ       equals 800  prefix NMA tag $C;/* Adjacent node
        constant PCCI_DRT       equals 801  prefix NMA tag $C;/* Designated router on NI
        constant PCCI_BLO       equals 810  prefix NMA tag $C;/* Block size (word)
        constant PCCI_COS       equals 900  prefix NMA tag $C;/* Cost (byte)
        constant PCCI_MRT       equals 901  prefix NMA tag $C;/* Maximum routers on NI (byte)
        constant PCCI_RPR       equals 902  prefix NMA tag $C;/* Router priority on NI (byte)
        constant PCCI_HET       equals 906  prefix NMA tag $C;/* Hello timer (word)
        constant PCCI_LIT       equals 907  prefix NMA tag $C;/* Listen timer (word)
        constant PCCI_BLK       equals 910  prefix NMA tag $C;/* Blocking (coded byte of NMA$C_CIRBLK_)
        constant PCCI_MRC       equals 920  prefix NMA tag $C;/* Maximum recalls (byte)
        constant PCCI_RCT       equals 921  prefix NMA tag $C;/* Recall timer (word)
        constant PCCI_NUM       equals 930  prefix NMA tag $C;/* Number (ascic)
        constant PCCI_USR       equals 1000  prefix NMA tag $C;/* User entity identification
        constant PCCI_POL       equals 1010  prefix NMA tag $C;/* Polling state (coded byte of NMA$C_CIRPST_)
        constant PCCI_PLS       equals 1011  prefix NMA tag $C;/* Polling substate (coded byte)
        constant PCCI_OWN       equals 1100  prefix NMA tag $C;/* Owner entity identification
        constant PCCI_LIN       equals 1110  prefix NMA tag $C;/* Line (ascic)
        constant PCCI_USE       equals 1111  prefix NMA tag $C;/* Usage (coded byte of NMA$C_CIRUS_)
        constant PCCI_TYP       equals 1112  prefix NMA tag $C;/* Type (coded byte of NMA$C_CIRTY_)
        constant PCCI_DTE       equals 1120  prefix NMA tag $C;/* DTE (ascic)
        constant PCCI_CHN       equals 1121  prefix NMA tag $C;/* Channel (word)
        constant PCCI_MBL       equals 1122  prefix NMA tag $C;/* Maximum data (word)
        constant PCCI_MWI       equals 1123  prefix NMA tag $C;/* Maximum window (byte)
        constant PCCI_TRI       equals 1140  prefix NMA tag $C;/* Tributary (byte)
        constant PCCI_BBT       equals 1141  prefix NMA tag $C;/* Babble timer (word)
        constant PCCI_TRT       equals 1142  prefix NMA tag $C;/* Transmit timer (word)
        constant PCCI_RTT       equals 1143  prefix NMA tag $C;/* Retransmit timer (word)
        constant PCCI_MRB       equals 1145  prefix NMA tag $C;/* Maximum receive buffers (coded byte)
                                                          /* 0-254 is value, 255 = UNLIMITED
        constant PCCI_MTR       equals 1146  prefix NMA tag $C;/* Maximum transmits (byte)
        constant PCCI_ACB       equals 1150  prefix NMA tag $C;/* Active base (byte)
        constant PCCI_ACI       equals 1151  prefix NMA tag $C;/* Active increment (byte)
        constant PCCI_IAB       equals 1152  prefix NMA tag $C;/* Inactive base (byte)
        constant PCCI_IAI       equals 1153  prefix NMA tag $C;/* Inactive increment (byte)
        constant PCCI_IAT       equals 1154  prefix NMA tag $C;/* Inactive threshold (byte)
        constant PCCI_DYB       equals 1155  prefix NMA tag $C;/* Dying base (byte)
        constant PCCI_DYI       equals 1156  prefix NMA tag $C;/* Dying increment (byte)
        constant PCCI_DYT       equals 1157  prefix NMA tag $C;/* Dying threshold (byte)
        constant PCCI_DTH       equals 1158  prefix NMA tag $C;/* Dead threshold (byte)
```

```
/*
/* RSX-specific circuit parameters
/*

        constant PCCI_RSX_MAC   equals 2320  prefix NMA tag $C;/* Multipoint active ratio
        constant PCCI_RSX_LOG   equals 2380  prefix NMA tag $C;/* Logical name
        constant PCCI_RSX_DLG   equals 2385  prefix NMA tag $C;/* Designated name
        constant PCCI_RSX_ACT   equals 2390  prefix NMA tag $C;/* Actual name

/*
/* VMS-specific circuit NICE parameters [2700 - 2799]
/*
        constant PCCI_VER       equals 2700  prefix NMA tag $C;/* Verification (coded byte of NMA$C_CIRVE_)
        constant PCCI_XPT       equals 2720  prefix NMA tag $C;/* Transport type (coded byte of NMA$C_CIRXPT_)


/*
/* VMS-specific datalink only circuit parameters   [2800 - 2899]
/*
/* (these will never be used in NICE messages).
/*

        constant PCCI_MST       equals 2810  prefix NMA tag $C;/* Maintenance state

/*
/* Server Base specific Circuit parameters
/*
        constant PCCI_SRV_LOG   equals 3380  prefix NMA tag $C;/* Logical name
        constant PCCI_SRV_DLG   equals 3385  prefix NMA tag $C;/* Designated name
        constant PCCI_SRV_ACT   equals 3390  prefix NMA tag $C;/* Actual name
```

```
/*
/*      Line parameters
/*


        constant PCLI_STA       equals 0  prefix NMA tag $C;/* State (coded byte of NMA$C_STATE_)
        constant PCLI_SUB       equals 1  prefix NMA tag $C;/* Substate (coded byte of NMA$C_LINSS_)
        constant PCLI_SER       equals 100  prefix NMA tag $C;/* Service (coded byte of NMA$C_LINSV_)
        constant PCLI_LCT       equals 110  prefix NMA tag $C;/* Counter timer (word)
        constant PCLI_LOO       equals 400  prefix NMA tag $C;/* Loopback name (ascic) [V2 only]
        constant PCLI_ADJ       equals 800  prefix NMA tag $C;/* Adjacent node [V2 only]
        constant PCLI_BLO       equals 810  prefix NMA tag $C;/* Block size (word) [V2 only]
        constant PCLI_COS       equals 900  prefix NMA tag $C;/* Cost (byte) [V2 only]
        constant PCLI_DEV       equals 1100  prefix NMA tag $C;/* Device (ascic)
        constant PCLI_BFN       equals 1105  prefix NMA tag $C;/* Receive buffers
        constant PCLI_CON       equals 1110  prefix NMA tag $C;/* Controller (coded byte of NMA$C_LINCN_)
        constant PCLI_DUP       equals 1111  prefix NMA tag $C;/* Duplex (coded byte of NMA$C_DPX_)
        constant PCLI_PRO       equals 1112  prefix NMA tag $C;/* Protocol (coded byte of NMA$C_LINPR_)
        constant PCLI_LTY       equals 1112  prefix NMA tag $C;/* Type (coded byte of NMA$C_LINTY_) [V2 only]
        constant PCLI_CLO       equals 1113  prefix NMA tag $C;/* Clock (coded byte of NMA$C_LINCL_)
        constant PCLI_STI       equals 1120  prefix NMA tag $C;/* Service timer (word)
        constant PCLI_NTI       equals 1121  prefix NMA tag $C;/* Normal timer (word) [V2 only]
        constant PCLI_RTT       equals 1121  prefix NMA tag $C;/* Retransmit timer (word)
        constant PCLI_HTI       equals 1122  prefix NMA tag $C;/* Holdback timer (word)
        constant PCLI_MBL       equals 1130  prefix NMA tag $C;/* Maximum block (word)
        constant PCLI_MRT       equals 1131  prefix NMA tag $C;/* Maximum retransmits (byte)
        constant PCLI_MWI       equals 1132  prefix NMA tag $C;/* Maximum window (byte)
        constant PCLI_TRI       equals 1140  prefix NMA tag $C;/* Tributary (byte) [V2 only]
        constant PCLI_SLT       equals 1150  prefix NMA tag $C;/* Scheduling timer (word)
        constant PCLI_DDT       equals 1151  prefix NMA tag $C;/* Dead timer (word)
        constant PCLI_DLT       equals 1152  prefix NMA tag $C;/* Delay timer (word)
        constant PCLI_SRT       equals 1153  prefix NMA tag $C;/* Stream timer (word)
        constant PCLI_HWA       equals 1160  prefix NMA tag $C;/* Hardware address (NI address)
/*
/* RSX-specific line parameters
/*

        constant PCLI_RSX_OWN   equals 2300  prefix NMA tag $C;/* Owner
        constant PCLI_RSX_CCS   equals 2310  prefix NMA tag $C;/* Controller CSR
        constant PCLI_RSX_UCS   equals 2311  prefix NMA tag $C;/* Unit CSR
        constant PCLI_RSX_VEC   equals 2312  prefix NMA tag $C;/* Vector
        constant PCLI_RSX_PRI   equals 2313  prefix NMA tag $C;/* Priority
        constant PCLI_RSX_MDE   equals 2321  prefix NMA tag $C;/* Dead polling ratio
        constant PCLI_RSX_LLO   equals 2330  prefix NMA tag $C;/* Location
                                                           /*  0, Firstfit
                                                           /*  1, Topdown
        constant PCLI_RSX_LOG   equals 2380  prefix NMA tag $C;/* Logical name
        constant PCLI_RSX_DLG   equals 2385  prefix NMA tag $C;/* Designated name
        constant PCLI_RSX_ACT   equals 2390  prefix NMA tag $C;/* Actual name


/*
/* VMS-specific line NICE parameters [2700 - 2799]
```

```
/*

        constant PCLI_MCD       equals 2701  prefix NMA tag $C;/* Micro-code dump filespec (ascic)
        constant PCLI_XMD       equals 2710  prefix NMA tag $C;/* X.25 line mode (coded byte of NMA$C_X25MD_)
        constant PCLI_EPT       equals 2720  prefix NMA tag $C;/* Ethernet Protocol Type (hex word)

/*
/* VMS-specific datalink only line parameters   [2800 - 2899]
/*
/* (these will never be used in NICE messages).
/*
        constant PCLI_BUS       equals 2801  prefix NMA tag $C;/* Buffer size (word)
        constant PCLI_NMS       equals 2810  prefix NMA tag $C;/* Number of DMP/DMF synch chars (word)
        constant PCLI_PHA       equals 2820  prefix NMA tag $C;/* Physical NI address of UNA (hex string)
        constant PCLI_DPA       equals 2821  prefix NMA tag $C;/* (same as HWA) ; Default UNA physical address (hex string)
        constant PCLI_PTY       equals 2830  prefix NMA tag $C;/* Ethernet Protocol type (word)
        constant PCLI_MCA       equals 2831  prefix NMA tag $C;/* UNA Multicast address list (special)
                                                        /*         (See NMA$C_LINMC_)
        constant PCLI_ILP       equals 2839  prefix NMA tag $C;/* DELUA Internal Loopback mode
                                                        /* (coded byte of NMA$C_STATE_)
        constant PCLI_PRM       equals 2840  prefix NMA tag $C;/* UNA Promiscuous mode (coded byte of NMA$C_STATE_)
        constant PCLI_MLT       equals 2841  prefix NMA tag $C;/* UNA Multicast address mode (coded byte of NMA$C_STATE_)
        constant PCLI_PAD       equals 2842  prefix NMA tag $C;/* UNA Padding mode (coded byte of NMA$C_STATE_)
        constant PCLI_DCH       equals 2843  prefix NMA tag $C;/* UNA Data chaining mode (coded byte of NMA$C_STATE_)
        constant PCLI_CRC       equals 2844  prefix NMA tag $C;/* UNA CRC mode (coded byte of NMA$C_STATE_)
        constant PCLI_HBQ       equals 2845  prefix NMA tag $C;/* UNA Hardware Buffer Quota (word)
        constant PCLI_ACC       equals 2846  prefix NMA tag $C;/* UNA protocol access mode (coded byte of NMA$C_ACC_)
        constant PCLI_EKO       equals 2847  prefix NMA tag $C;/* UNA Echo mode (coded byte of NMA$C_STATE_)
        constant PCLI_BSZ       equals 2848  prefix NMA tag $C;/* UNA Device Buffer size
        constant PCLI_DES       equals 2849  prefix NMA tag $C;/* UNA destination Ethernet address

        constant PCLI_RET       equals 2850  prefix NMA tag $C;/* PCL number of retries (word)
        constant PCLI_MOD       equals 2851  prefix NMA tag $C;/* PCL address mode (coded byte of NMA$C_LINMO_)
        constant PCLI_RIB       equals 2852  prefix NMA tag $C;/* PCL retry-if-busy state (coded byte of NMA$C_STATE_)

        constant PCLI_MNTL      equals 2860  prefix NMA tag $C;/* Maintenance loopback mode for devices
                                                        /*  which support several different loop back modes
        constant PCLI_INTL0     equals 2861  prefix NMA tag $C;/* Internal loopback level 0
        constant PCLI_INTL1     equals 2862  prefix NMA tag $C;/* Internal loopback level 1
        constant PCLI_INTL2     equals 2863  prefix NMA tag $C;/* Internal loopback level 2
        constant PCLI_INTL3     equals 2864  prefix NMA tag $C;/* Internal loopback level 3
        constant PCLI_FRA       equals 2865  prefix NMA tag $C;/* Framing address for Bisync
        constant PCLI_STI1      equals 2866  prefix NMA tag $C;/* State info 1st longword
        constant PCLI_STI2      equals 2867  prefix NMA tag $C;/* State info 2st longword
        constant PCLI_TMO       equals 2868  prefix NMA tag $C;/* Wait for CTS time out value for DMF sync half duplex
        constant PCLI_MCL       equals 2869  prefix NMA tag $C;/* Clear modem on deassign of channel
        constant PCLI_SYC       equals 2870  prefix NMA tag $C;/* BISYNC protocol sync char
        constant PCLI_BPC       equals 2871  prefix NMA tag $C;/* Number of bits per character
/*
/*    Server Base specific line parameters
/*

        constant PCLI_SRV_OWN   equals 3300  prefix NMA tag $C;/* Owner
        constant PCLI_SRV_UCS   equals 3311  prefix NMA tag $C;/* Unit CSR
        constant PCLI_SRV_VEC   equals 3312  prefix NMA tag $C;/* Vector
        constant PCLI_SRV_PRI   equals 3313  prefix NMA tag $C;/* Priority
        constant PCLI_SRV_LOG    equals 3380  prefix NMA tag $C;/* Logical name
        constant PCLI_SRV_DLG    equals 3385  prefix NMA tag $C;/* Designated name
        constant PCLI_SRV_ACT    equals 3390  prefix NMA tag $C;/* Actual name
```

```
/*
/*	Console module parameters
/*

	constant PCCO_RTR	equals 110  prefix NMA tag $C;/* Reservation timer (word)
```

```
/*
/*    Loader module parameters
/*

        constant PCLD_ASS        equals 10  prefix NMA tag $C;/* Assistance flag (coded byte of NMA$C_ASS_)
```

```
/*
/*     Looper module parameters
/*

        constant PCLP_ASS       equals 10  prefix NMA tag $C;/* Assistance flag (coded byte of NMA$C_ASS_)
```

```
/*
/*      Configurator module parameters
/*

        constant PCCN_CIR       equals 100  prefix NMA tag $C;/* NI circuit name (ascic)
        constant PCCN_SUR       equals 110  prefix NMA tag $C;/* Surveillance flag (coded byte of NMA$C_SUR_)
        constant PCCN_ELT       equals 111  prefix NMA tag $C;/* Elapsed time
        constant PCCN_PHA       equals 120  prefix NMA tag $C;/* Physical address (NI address)
        constant PCCN_LRP       equals 130  prefix NMA tag $C;/* Time of last report
        constant PCCN_MVR       equals 20001  prefix NMA tag $C;/* Maintenance version
        constant PCCN_FCT       equals 20002  prefix NMA tag $C;/* Function list
        constant PCCN_CUS       equals 20003  prefix NMA tag $C;/* Current console user (NI address)
        constant PCCN_RTR       equals 20004  prefix NMA tag $C;/* Reservation timer (word)
        constant PCCN_CSZ       equals 20005  prefix NMA tag $C;/* Command buffer size (word)
        constant PCCN_RSZ       equals 20006  prefix NMA tag $C;/* Response buffer size (word)
        constant PCCN_HWA       equals 20007  prefix NMA tag $C;/* Hardware address (NI address)
        constant PCCN_DTY       equals 20100  prefix NMA tag $C;/* Device type (coded byte of NMA$C_SOFD_)
        constant PCCN_SFI       equals 20200  prefix NMA tag $C;/* Software ID
        constant PCCN_SPR       equals 20300  prefix NMA tag $C;/* System processor (coded word)
        constant PCCN_DLK       equals 20400  prefix NMA tag $C;/* Data link type (coded word)
```

```
/*
/*    Logging parameters
/*

       constant PCLO_STA        equals 0  prefix NMA tag $C;/* State (coded byte of NMA$C_STATE_)
       constant PCLO_LNA        equals 100  prefix NMA tag $C;/* System/name (ascic)
       constant PCLO_SIN        equals 200  prefix NMA tag $C;/* Sink node
       constant PCLO_EVE        equals 201  prefix NMA tag $C;/* Events
```

```
/*
/*    X.25 Access module parameters
/*


        constant PCXA_NOD       equals 320  prefix NMA tag $C;/* Node
        constant PCXA_USR       equals 330  prefix NMA tag $C;/* User (ascic)
        constant PCXA_PSW       equals 331  prefix NMA tag $C;/* Password (ascic)
        constant PCXA_ACC       equals 332  prefix NMA tag $C;/* Account (ascic)
        constant PCXA_NET       equals 1110 prefix NMA tag $C;/* Network (ascic)

/*
/* RSX-specific X.25-Access module parameters
/*
        constant PCXA_RSX_ADS   equals 2310  prefix NMA tag $C;/* Destination
        constant PCXA_RSX_ANB   equals 2320  prefix NMA tag $C;/* Number
        constant PCXA_RSX_ASC   equals 2330  prefix NMA tag $C;/* Scope
/*
/* Server Base specific X.25-Access module parameters
/*
        constant PCXA_SRV_ADS   equals 3310  prefix NMA tag $C;/* Destination
        constant PCXA_SRV_ANB   equals 3320  prefix NMA tag $C;/* Number
        constant PCXA_SRV_ASC   equals 3330  prefix NMA tag $C;/* Scope
```

```
/*
/*    X.25 Protocol module parameters
/*


      constant PCXP_STA        equals 0  prefix NMA tag $C;/* State (coded byte of NMA$C_STATE_)
      constant PCXP_CTM        equals 100  prefix NMA tag $C;/* Counter timer (word)
      constant PCXP_ACH        equals 1000  prefix NMA tag $C;/* Active channels (word)
      constant PCXP_ASW        equals 1010  prefix NMA tag $C;/* Active switched (word)
      constant PCXP_DTE        equals 1100  prefix NMA tag $C;/* DTE (ascic)
      constant PCXP_GRP        equals 1101  prefix NMA tag $C;/* Group (ascic)
      constant PCXP_NET        equals 1110  prefix NMA tag $C;/* Network (ascic)
      constant PCXP_LIN        equals 1120  prefix NMA tag $C;/* Line (ascic)
      constant PCXP_CHN        equals 1130  prefix NMA tag $C;/* Channels
      constant PCXP_MCH        equals 1131  prefix NMA tag $C;/* Maximum channels (word)
      constant PCXP_DBL        equals 1140  prefix NMA tag $C;/* Default data (word)
      constant PCXP_DWI        equals 1141  prefix NMA tag $C;/* Default window (byte)
      constant PCXP_MBL        equals 1150  prefix NMA tag $C;/* Maximum data (word)
      constant PCXP_MWI        equals 1151  prefix NMA tag $C;/* Maximum window (byte)
      constant PCXP_MCL        equals 1152  prefix NMA tag $C;/* Maximum clears (byte)
      constant PCXP_MRS        equals 1153  prefix NMA tag $C;/* Maximum resets (byte)
      constant PCXP_MST        equals 1154  prefix NMA tag $C;/* Maximum restarts (byte)
      constant PCXP_CAT        equals 1160  prefix NMA tag $C;/* Call timer (byte)
      constant PCXP_CLT        equals 1161  prefix NMA tag $C;/* Clear timer (byte)
      constant PCXP_RST        equals 1162  prefix NMA tag $C;/* Reset timer (byte)
      constant PCXP_STT        equals 1163  prefix NMA tag $C;/* Restart timer (byte)
      constant PCXP_GDT        equals 1170  prefix NMA tag $C;/* Group DTE (ascic)
      constant PCXP_GNM        equals 1171  prefix NMA tag $C;/* Group number (word)
      constant PCXP_GTY        equals 1172  prefix NMA tag $C;/* Group type (coded byte of NMA$C_XPRTY_)
/*
/*      RSX-specific X.25-Protocol Module parameters
/*
      constant PCXP_RSX_PMC    equals 2300  prefix NMA tag $C;/* Maximum circuits

/*
/* VMS-specific X25-PROTOCOL NICE parameters [2700 - 2799]
/*

      constant PCXP_MNS        equals 2700  prefix NMA tag $C;/* Multinetwork Support flag (coded byte of NMA$C_XPRMN_) [disabled,
      constant PCXP_MCI        equals 2710  prefix NMA tag $C;/* Maximum circuits, qualified by DTE
      constant PCXP_SBS        equals 2720  prefix NMA tag $C;/* Substate, qualified by DTE (coded byte of NMA$C_XPRSB_)

/*
/* Server Base specific X.25-Protocol Module parameters
/*
      constant PCXP_SRV_PMC    equals 3300  prefix NMA tag $C;/* Maximum circuits
```

```
/*
/*    X.25 server module parameters
/*

        constant PCXS_CTM        equals 100   prefix NMA tag $C;/* Counter timer (word)
        constant PCXS_ACI        equals 200   prefix NMA tag $C;/* Active circuits (word)
        constant PCXS_DST        equals 300   prefix NMA tag $C;/* Destination (ascic)
        constant PCXS_MCI        equals 310   prefix NMA tag $C;/* Maximum circuits (word)
        constant PCXS_NOD        equals 320   prefix NMA tag $C;/* Node
        constant PCXS_USR        equals 330   prefix NMA tag $C;/* Username
        constant PCXS_SPW        equals 331   prefix NMA tag $C;/* Password to set (ascic)
        constant PCXS_RPW        equals 331   prefix NMA tag $C;/* Password to read (coded byte of NMA$C_NODPW_)
        constant PCXS_ACC        equals 332   prefix NMA tag $C;/* Account (ascic)
        constant PCXS_OBJ        equals 340   prefix NMA tag $C;/* Object
        constant PCXS_PRI        equals 350   prefix NMA tag $C;/* Priority (byte)
        constant PCXS_CMK        equals 351   prefix NMA tag $C;/* Call mask (byte-counted hex)
        constant PCXS_CVL        equals 352   prefix NMA tag $C;/* Call value (byte-counted hex)
        constant PCXS_GRP        equals 353   prefix NMA tag $C;/* Group (ascic)
        constant PCXS_NUM        equals 354   prefix NMA tag $C;/* Number (ascic)
        constant PCXS_SAD        equals 355   prefix NMA tag $C;/* Subaddresses

/*
/* RSX-specific X.25-Server Module parameters
/*
        constant PCXS_RSX_5ST    equals 2310  prefix NMA tag $C;/* State
                                                             /*  0, On
                                                             /*  1, Off

/*
/* VMS-specific X25-SERVER NICE parameters [2700 - 2799]
/*

        constant PCXS_STA        equals 2700  prefix NMA tag $C;/* Server state (coded byte of NMA$C_STATE_)
        constant PCXS_FIL        equals 2710  prefix NMA tag $C;/* Object filespec (ascic)

/*
/* Server Base specific X.25-Server Module parameters
/*
        constant PCXS_SRV_5ST    equals 3310  prefix NMA tag $C;/* State
                                                             /*  0, On
                                                             /*  1, Off
```

```
/*
/* X.25 trace module parameters (VMS-specific)
/*

        constant PCXT_STA        equals 0  prefix NMA tag $C;/* State (coded byte of NMA$C_STATE_)
        constant PCXT_BSZ        equals 100  prefix NMA tag $C;/* Buffer size (word)
        constant PCXT_MBK        equals 101  prefix NMA tag $C;/* Maximum blocks/file (word)
        constant PCXT_FNM        equals 102  prefix NMA tag $C;/* Filename (ascic)
        constant PCXT_MBF        equals 103  prefix NMA tag $C;/* Maximum number of buffers (word)
        constant PCXT_CPL        equals 104  prefix NMA tag $C;/* Global data capture limit (word)
        constant PCXT_MVR        equals 105  prefix NMA tag $C;/* Maximum trace file version (word)
        constant PCXT_TPT        equals 106  prefix NMA tag $C;/* Trace point name (ascic)
        constant PCXT_CPS        equals 110  prefix NMA tag $C;/* Per-trace capture size (word)
        constant PCXT_TST        equals 111  prefix NMA tag $C;/* Per-trace state (coded byte of NMA$C_STATE_)
```

```
/*
/*      Node parameters
/*


        constant PCNO_STA           equals 0  prefix NMA tag $C;/* State (coded byte of NMA$C_STATE_)
        constant PCNO_PHA           equals 10  prefix NMA tag $C;/* Physical address (NI address)
        constant PCNO_IDE           equals 100  prefix NMA tag $C;/* Identification (ascic)
        constant PCNO_MVE           equals 101  prefix NMA tag $C;/* Management version (3 bytes)
        constant PCNO_SLI           equals 110  prefix NMA tag $C;/* Service circuit (ascic)
        constant PCNO_SPA           equals 111  prefix NMA tag $C;/* Service password (8 bytes)
        constant PCNO_SDV           equals 112  prefix NMA tag $C;/* Service device (coded byte of NMA$C_SOFD_)
        constant PCNO_CPU           equals 113  prefix NMA tag $C;/* CPU type (coded byte of NMA$C_CPU_)
        constant PCNO_HWA           equals 114  prefix NMA tag $C;/* Hardware address (NI address)
        constant PCNO_SNV           equals 115  prefix NMA tag $C;/* Service node version (coded byte of NMA$C_SVN_)
        constant PCNO_LOA           equals 120  prefix NMA tag $C;/* Load file (ascic)
        constant PCNO_SLO           equals 121  prefix NMA tag $C;/* Secondary loader (ascic)
        constant PCNO_TLO           equals 122  prefix NMA tag $C;/* Tertiary loader (ascic)
        constant PCNO_DFL           equals 123  prefix NMA tag $C;/* Diagnostic file (ascic)
        constant PCNO_STY           equals 125  prefix NMA tag $C;/* Software type (coded byte of NMA$C_SOFT_)
        constant PCNO_SID           equals 126  prefix NMA tag $C;/* Software ID (ascic)
        constant PCNO_DUM           equals 130  prefix NMA tag $C;/* Dump file (ascic)
        constant PCNO_SDU           equals 131  prefix NMA tag $C;/* Secondary dumper (ascic)
        constant PCNO_DAD           equals 135  prefix NMA tag $C;/* Dump address (longword)
        constant PCNO_DCT           equals 136  prefix NMA tag $C;/* Dump count (longword)
        constant PCNO_OHO           equals 140  prefix NMA tag $C;/* Host (read only parameter)
        constant PCNO_IHO           equals 141  prefix NMA tag $C;/* Host (write only parameter)
        constant PCNO_LPC           equals 150  prefix NMA tag $C;/* Loop count (word)
        constant PCNO_LPL           equals 151  prefix NMA tag $C;/* Loop length (word)
        constant PCNO_LPD           equals 152  prefix NMA tag $C;/* Loop Data type (coded byte of NMA$C_LOOP_)
        constant PCNO_LPA           equals 153  prefix NMA tag $C;/* Loop assistant physical address (NI address)
        constant PCNO_LPH           equals 154  prefix NMA tag $C;/* Loop help type (coded byte)
        constant PCNO_LPN           equals 155  prefix NMA tag $C;/* Loop circuit node
        constant PCNO_LAN           equals 156  prefix NMA tag $C;/* Loop circuit assistant node
        constant PCNO_CTI           equals 160  prefix NMA tag $C;/* Counter timer (word)
        constant PCNO_NNA           equals 500  prefix NMA tag $C;/* Name
        constant PCNO_NLI           equals 501  prefix NMA tag $C;/* Circuit (ascic)
        constant PCNO_ADD           equals 502  prefix NMA tag $C;/* Address
        constant PCNO_ITI           equals 510  prefix NMA tag $C;/* Incoming timer (word)
        constant PCNO_OTI           equals 511  prefix NMA tag $C;/* Outgoing timer (word)
        constant PCNO_ACL           equals 600  prefix NMA tag $C;/* Active links (word)
        constant PCNO_DEL           equals 601  prefix NMA tag $C;/* Delay (word)
        constant PCNO_NVE           equals 700  prefix NMA tag $C;/* Nsp version (3 bytes)
        constant PCNO_MLK           equals 710  prefix NMA tag $C;/* Maximum links (word)
        constant PCNO_DFA           equals 720  prefix NMA tag $C;/* Delay factor (byte)
        constant PCNO_DWE           equals 721  prefix NMA tag $C;/* Delay weight (byte)
        constant PCNO_IAT           equals 722  prefix NMA tag $C;/* Inactivity timer (word)
        constant PCNO_RFA           equals 723  prefix NMA tag $C;/* Retransmit factor (word)
        constant PCNO_DTY           equals 810  prefix NMA tag $C;/* Destination Type (coded byte of NMA$C_XPRTY_)
        constant PCNO_DCO           equals 820  prefix NMA tag $C;/* Destination Cost (word)
        constant PCNO_DHO           equals 821  prefix NMA tag $C;/* Destination Hops (byte)
        constant PCNO_DLI           equals 822  prefix NMA tag $C;/* Destination circuit (ascic)
        constant PCNO_NND           equals 830  prefix NMA tag $C;/* Next node to destination
        constant PCNO_RVE           equals 900  prefix NMA tag $C;/* Routing version (3 bytes)
        constant PCNO_ETY           equals 901   prefix NMA tag $C;/* Executor Type (coded byte of NMA$C_NODTY_)
        constant PCNO_RTI           equals 910   prefix NMA tag $C;/* Routing timer (word)
        constant PCNO_SAD           equals 911   prefix NMA tag $C;/* Subaddress (2 words)
        constant PCNO_BRT           equals 912   prefix NMA tag $C;/* Broadcast routing timer (word)
        constant PCNO_MAD           equals 920   prefix NMA tag $C;/* Maximum address (word)
        constant PCNO_MLN           equals 921   prefix NMA tag $C;/* Maximum circuits (word)
        constant PCNO_MCO           equals 922   prefix NMA tag $C;/* Maximum cost (word)
        constant PCNO_MHO           equals 923   prefix NMA tag $C;/* Maximum hops (byte)
        constant PCNO_MVI           equals 924   prefix NMA tag $C;/* Maximum visits (byte)
        constant PCNO_MAR           equals 925   prefix NMA tag $C;/* Maximum areas (byte)
        constant PCNO_MBE           equals 926   prefix NMA tag $C;/* Maximum broadcast nonrouters (word)
        constant PCNO_MBR           equals 927   prefix NMA tag $C;/* Maximum broadcast routers (word)
        constant PCNO_AMC           equals 928   prefix NMA tag $C;/* Area maximum cost (word)
        constant PCNO_AMH           equals 929   prefix NMA tag $C;/* Area maximum hops (byte)
        constant PCNO_MBU           equals 930   prefix NMA tag $C;/* Maximum buffers (word)
        constant PCNO_BUS           equals 931   prefix NMA tag $C;/* Executor buffer size (word)
        constant PCNO_SBS           equals 932   prefix NMA tag $C;/* Segment buffer size (word)
        constant PCNO_FBS           equals 933   prefix NMA tag $C;/* Forwarding buffer size (word)
/*
/* RSX-Specific Node (Executor) parameters
/*
        constant PCNO_RSX_RPA   equals 2300  prefix NMA tag $C;/* Receive password
                                                        /*  0, Password set

        constant PCNO_RSX_TPA   equals 2301  prefix NMA tag $C;/* Transmit password
                                                        /*  0, Password set

        constant PCNO_RSX_VER   equals 2310  prefix NMA tag $C;/* Verification state
                                                        /*  0, On
                                                        /*  1, Off
/*
/* VMS-specific node parameters
/*
        constant PCNO_PUS           equals 2704  prefix NMA tag $C;/* Privileged user id
        constant PCNO_PAC           equals 2705  prefix NMA tag $C;/* Privileged account
        constant PCNO_PPW           equals 2706  prefix NMA tag $C;/* Privileged password
        constant PCNO_NUS           equals 2712  prefix NMA tag $C;/* Non-privileged user id
        constant PCNO_NAC           equals 2713  prefix NMA tag $C;/* Non-privileged account
        constant PCNO_NPW           equals 2714  prefix NMA tag $C;/* Non-privileged password
        constant PCNO_RPA           equals 2720  prefix NMA tag $C;/* Receive password
        constant PCNO_TPA           equals 2721  prefix NMA tag $C;/* Transmit password
        constant PCNO_ACC           equals 2730  prefix NMA tag $C;/* Access (coded byte of NMA$C_ACES_)
        constant PCNO_DAC           equals 2731  prefix NMA tag $C;/* Default access (coded byte of NMA$C_ACES_)
        constant PCNO_PIQ           equals 2740  prefix NMA tag $C;/* Pipeline quota (word)
        constant PCNO_ALI           equals 2741  prefix NMA tag $C;/* Alias address (word)
        constant PCNO_PRX           equals 2750  prefix NMA tag $C;/* Proxy access (coded byte of NMA$C_ACES_) !! Obsolete: Only for LIS
        constant PCNO_DPX           equals 2751  prefix NMA tag $C;/* Default proxy access (coded byte of NMA$C_ACES_)

/*
/* Server Base specific Node (Executor) parameters
/*
        constant PCNO_SRV_RPA   equals 3300  prefix NMA tag $C;/* Receive password
                                                        /*  0, Password set
        constant PCNO_SRV_TPA   equals 3301  prefix NMA tag $C;/* Transmit password
                                                        /*  0, Password set
        constant PCNO_SRV_VER   equals 3310  prefix NMA tag $C;/* Verification state
                                                          /*  0, On
                                                          /*  1, Off
        constant PCNO_SRV_ACB   equals 3402  prefix NMA tag $C;/* Active control buffers
        constant PCNO_SRV_ASB   equals 3404  prefix NMA tag $C;/* Active small buffers
        constant PCNO_SRV_ALB   equals 3406  prefix NMA tag $C;/* Active large buffers
        constant PCNO_SRV_MCB   equals 3410  prefix NMA tag $C;/* Maximum control buffers
        constant PCNO_SRV_MSB   equals 3420  prefix NMA tag $C;/* Maximum small buffers
        constant PCNO_SRV_MLB   equals 3430  prefix NMA tag $C;/* Maximum large buffers
        constant PCNO_SRV_LBS   equals 3431  prefix NMA tag $C;/* Large buffer size
        constant PCNO_SRV_NRB   equals 3440  prefix NMA tag $C;/* Minimum receive buffers
        constant PCNO_SRV_CPT   equals 3450  prefix NMA tag $C;/* CEX pool: total bytes
        constant PCNO_SRV_CPF   equals 3452  prefix NMA tag $C;/* CEX pool: number of segments
        constant PCNO_SRV_CPL   equals 3454  prefix NMA tag $C;/* CEX pool: largest segment
        constant PCNO_SRV_XPT   equals 3460  prefix NMA tag $C;/* Extended pool: total bytes
        constant PCNO_SRV_XPF   equals 3462  prefix NMA tag $C;/* Extended pool: number of segments
        constant PCNO_SRV_XPL   equals 3464  prefix NMA tag $C;/* Extended pool: largest segment
```

```
/*
/*    Area parameters
/*

        constant PCAR_STA         equals 0   prefix NMA tag $C;/* State (coded byte of NMA$C_STATE_)
        constant PCAR_COS         equals 820  prefix NMA tag $C;/* Cost (word)
        constant PCAR_HOP         equals 821  prefix NMA tag $C;/* Hops (byte)
        constant PCAR_CIR         equals 822  prefix NMA tag $C;/* Circuit (ascic)
        constant PCAR_NND         equals 830  prefix NMA tag $C;/* Next node to area
```

```
/*
/*      VMS-specific object parameters
/*

        constant PCOB_OAN           equals 400  prefix NMA tag $C;/* Active name
        constant PCOB_OAC           equals 410  prefix NMA tag $C;/* Active links
        constant PCOB_ONA           equals 500  prefix NMA tag $C;/* Name
        constant PCOB_OCO           equals 510  prefix NMA tag $C;/* Copies
        constant PCOB_OUS           equals 511  prefix NMA tag $C;/* User
        constant PCOB_OVE           equals 520  prefix NMA tag $C;/* Verification

        constant PCOB_NAM           equals 500  prefix NMA tag $C;/* Name
        constant PCOB_NUM           equals 513  prefix NMA tag $C;/* Number
        constant PCOB_FID           equals 530  prefix NMA tag $C;/* File id
        constant PCOB_PID           equals 535  prefix NMA tag $C;/* Process id
        constant PCOB_PRV           equals 540  prefix NMA tag $C;/* Privilege list
        constant PCOB_USR           equals 550  prefix NMA tag $C;/* User id
        constant PCOB_ACC           equals 551  prefix NMA tag $C;/* Account
        constant PCOB_PSW           equals 552  prefix NMA tag $C;/* Password
        constant PCOB_PRX           equals 560  prefix NMA tag $C;/* Proxy access (coded byte of NMA$C_ACES_)
```

```
/*
/*      VMS-specific link parameters
/*

        constant PCLK_STA        equals 0  prefix NMA tag $C;/* State
        constant PCLK_PID        equals 101  prefix NMA tag $C;/* Process id
        constant PCLK_NID        equals 102  prefix NMA tag $C;/* Partner Node
        constant PCLK_LAD        equals 105  prefix NMA tag $C;/* Link address [V2 only]
                                                           /* entity is node rather than link !
                                                           /* CM-1/2, DU-2 (link !), HI-4 (pid)
        constant PCLK_DLY        equals 110  prefix NMA tag $C;/* Round trip delay time (word)
        constant PCLK_RLN        equals 120  prefix NMA tag $C;/* Remote link number (word)
        constant PCLK_RID        equals 121  prefix NMA tag $C;/* Remote identification, PID or username (ascic)
        constant PCLK_USR        equals 130  prefix NMA tag $C;/* Username of link owner (ascic)
        constant PCLK_PRC        equals 131  prefix NMA tag $C;/* Process name of link owner (ascic)
```

```
/*
/*    Circuit counters
/*


        constant CTCIR_ZER      equals 0  prefix NMA tag $C;/* Seconds since last zeroed
        constant CTCIR_APR      equals 800  prefix NMA tag $C;/* Terminating packets received
        constant CTCIR_DPS      equals 801  prefix NMA tag $C;/* Originating packets sent
        constant CTCIR_ACL      equals 802  prefix NMA tag $C;/* Terminating congestion loss
        constant CTCIR_CRL      equals 805  prefix NMA tag $C;/* Corruption loss
        constant CTCIR_TPR      equals 810  prefix NMA tag $C;/* Transit packets received
        constant CTCIR_TPS      equals 811  prefix NMA tag $C;/* Transit packets sent
        constant CTCIR_TCL      equals 812  prefix NMA tag $C;/* Transit congestion loss
        constant CTCIR_LDN      equals 820  prefix NMA tag $C;/* Circuit down
        constant CTCIR_IFL      equals 821  prefix NMA tag $C;/* Initialization failure
        constant CTCIR_BRC      equals 1000  prefix NMA tag $C;/* Bytes received
        constant CTCIR_BSN      equals 1001  prefix NMA tag $C;/* Bytes sent
        constant CTCIR_MBY      equals 1002  prefix NMA tag $C;/* Multicast bytes received
        constant CTCIR_DBR      equals 1010  prefix NMA tag $C;/* Data blocks received
        constant CTCIR_DBS      equals 1011  prefix NMA tag $C;/* Data blocks sent
        constant CTCIR_DEI      equals 1020  prefix NMA tag $C;/* Data errors inbound
        constant CTCIR_DEO      equals 1021  prefix NMA tag $C;/* Data errors outbound
        constant CTCIR_RRT      equals 1030  prefix NMA tag $C;/* Remote reply timeouts
        constant CTCIR_LRT      equals 1031  prefix NMA tag $C;/* Local reply timeouts
        constant CTCIR_RBE      equals 1040  prefix NMA tag $C;/* Remote buffer errors
        constant CTCIR_LBE      equals 1041  prefix NMA tag $C;/* Local buffer errors
        constant CTCIR_SIE      equals 1050  prefix NMA tag $C;/* Selection intervals elapsed
        constant CTCIR_SLT      equals 1051  prefix NMA tag $C;/* Selection timeouts
        constant CTCIR_UBU      equals 1065  prefix NMA tag $C;/* NI user buffer unavailable
        constant CTCIR_RPE      equals 1100  prefix NMA tag $C;/* Remote process errors [V2 only]
        constant CTCIR_LPE      equals 1101  prefix NMA tag $C;/* Local process errors [V2 only]
        constant CTCIR_LIR      equals 1240  prefix NMA tag $C;/* Locally initiated resets
        constant CTCIR_RIR      equals 1241  prefix NMA tag $C;/* Remotely initiated resets
        constant CTCIR_NIR      equals 1242  prefix NMA tag $C;/* Network initiated resets

/*
/* VMS-specific circuit counters
/*

        constant CTCIR_MNE      equals 2701  prefix NMA tag $C;/* Multicast received for protocol
                                                        /* type, but not enabled
        constant CTCIR_ERI      equals 2750  prefix NMA tag $C;/* PCL Errors inbound, bit-mapped
                                                        /*        0  CRC error on receive
        constant CTCIR_ERO      equals 2751  prefix NMA tag $C;/* PCL Errors outbound, bit-mapped
                                                        /*        1  CRC on transmit
                                                        /*        2  Timeout on word
        constant CTCIR_RTO      equals 2752  prefix NMA tag $C;/* PCL Remote timeouts, bit-mapped
                                                        /*        0  Receiver busy
                                                        /*        1  Transmitter offline
                                                        /*        2  Receiver offline
        constant CTCIR_LTO      equals 2753  prefix NMA tag $C;/* PCL Local timeouts
        constant CTCIR_BER      equals 2754  prefix NMA tag $C;/* PCL Remote buffer errors
        constant CTCIR_BEL      equals 2755  prefix NMA tag $C;/* PCL Local buffer errors
```

```
/*
/*    Line counters
/*


        constant CTLIN_ZER      equals 0  prefix NMA tag $C;/* Seconds since last zeroed
        constant CTLIN_APR      equals 800  prefix NMA tag $C;/* Arriving packets received [V2 only]
        constant CTLIN_DPS      equals 801  prefix NMA tag $C;/* Departing packets sent [V2 only]
        constant CTLIN_ACL      equals 802  prefix NMA tag $C;/* Arriving congestion loss [V2 only]
        constant CTLIN_TPR      equals 810  prefix NMA tag $C;/* Transit packets received [V2 only]
        constant CTLIN_TPS      equals 811  prefix NMA tag $C;/* Transit packets sent [V2 only]
        constant CTLIN_TCL      equals 812  prefix NMA tag $C;/* Transit congestion loss [V2 only]
        constant CTLIN_LDN      equals 820  prefix NMA tag $C;/* Line down [V2 only]
        constant CTLIN_IFL      equals 821  prefix NMA tag $C;/* Initialization failure [V2 only]
        constant CTLIN_BRC      equals 1000  prefix NMA tag $C;/* Bytes received
        constant CTLIN_BSN      equals 1001  prefix NMA tag $C;/* Bytes sent
        constant CTLIN_MBY      equals 1002  prefix NMA tag $C;/* Multicast bytes received
        constant CTLIN_DBR      equals 1010  prefix NMA tag $C;/* Data blocks received
        constant CTLIN_DBS      equals 1011  prefix NMA tag $C;/* Data blocks sent
        constant CTLIN_MBL      equals 1012  prefix NMA tag $C;/* Multicast blocks received
        constant CTLIN_BID      equals 1013  prefix NMA tag $C;/* Blocks sent, initially deferred
        constant CTLIN_BS1      equals 1014  prefix NMA tag $C;/* Blocks sent, single collision
        constant CTLIN_BSM      equals 1015  prefix NMA tag $C;/* Blocks sent, multiple collisions
        constant CTLIN_DEI      equals 1020  prefix NMA tag $C;/* Data errors inbound
        constant CTLIN_DEO      equals 1021  prefix NMA tag $C;/* Data errors outbound
        constant CTLIN_RRT      equals 1030  prefix NMA tag $C;/* Remote reply timeouts
        constant CTLIN_LRT      equals 1031  prefix NMA tag $C;/* Local reply timeouts
        constant CTLIN_RBE      equals 1040  prefix NMA tag $C;/* Remote buffer errors
        constant CTLIN_LBE      equals 1041  prefix NMA tag $C;/* Local buffer errors
        constant CTLIN_SIE      equals 1050  prefix NMA tag $C;/* Selection intervals elapsed [V2 only]
        constant CTLIN_SLT      equals 1051  prefix NMA tag $C;/* Selection timeouts [V2 only]
        constant CTLIN_SFL      equals 1060  prefix NMA tag $C;/* Send failure
        constant CTLIN_CDC      equals 1061  prefix NMA tag $C;/* Collision detect check failure
        constant CTLIN_RFL      equals 1062  prefix NMA tag $C;/* Receive failure
        constant CTLIN_UFD      equals 1063  prefix NMA tag $C;/* Unrecognized frame destination
        constant CTLIN_OVR      equals 1064  prefix NMA tag $C;/* Data overrun
        constant CTLIN_SBU      equals 1065  prefix NMA tag $C;/* System buffer unavailable
        constant CTLIN_UBU      equals 1066  prefix NMA tag $C;/* User buffer unavailable
        constant CTLIN_RPE      equals 1100  prefix NMA tag $C;/* Remote process errors
        constant CTLIN_LPE      equals 1101  prefix NMA tag $C;/* Local process errors


/*
/* Line counter flags (byte offset will be 0)
/*
    end NMADEF1;

aggregate NMADEF2 union fill prefix NMA$;
    FILL_8 byte fill prefix NMADEF tag $$;                  /* byte of flags
        FILL_8_BITS structure fill;
            FILL_9 bitfield length 3 fill prefix NMADEF tag $$;/* skip bits 0,1,2
            CTLIN_BTL bitfield mask;                        /* block too long
            CTLIN_FCS bitfield mask;                        /* frame check
            CTLIN_TRJ bitfield mask;                        /* REJ sent
        end FILL_8_BITS;
end NMADEF2;

aggregate NMADEF3 union fill prefix NMA$;
    FILL_10 byte fill prefix NMADEF tag $$;                 /* byte of flags
        FILL_10_BITS structure fill;
            FILL_11 bitfield length 3 fill prefix NMADEF tag $$;/* skip bits 0,1,2
            CTLIN_RRJ bitfield mask;                        /* REJ received
        end FILL_10_BITS;
end NMADEF3;

aggregate NMADEF4 union fill prefix NMA$;
    FILL_12 byte fill prefix NMADEF tag $$;                 /* byte of flags
        FILL_12_BITS structure fill;
            FILL_13 bitfield length 2 fill prefix NMADEF tag $$;/* skip bits 0,1
            CTLIN_RRN bitfield mask;                        /* RNR received
        end FILL_12_BITS;
end NMADEF4;

aggregate NMADEF5 union fill prefix NMA$;
    FILL_14 byte fill prefix NMADEF tag $$;                 /* byte of flags
        FILL_14_BITS structure fill;
            FILL_15 bitfield length 2 fill prefix NMADEF tag $$;/* skip bits 0,1
            CTLIN_TRN bitfield mask;                        /* RNR sent
        end FILL_14_BITS;
end NMADEF5;

aggregate NMADEF6 union fill prefix NMA$;
    FILL_16 byte fill prefix NMADEF tag $$;                 /* byte of flags
        FILL_16_BITS structure fill;
            FILL_17 bitfield length 4 fill prefix NMADEF tag $$;/* skip bits 0,1,2,3
            CTLIN_INR bitfield mask;                        /* invalid N(R) received
            CTLIN_FMS bitfield mask;                        /* FRMR sent
        end FILL_16_BITS;
end NMADEF6;

aggregate NMADEF7 union fill prefix NMA$;
    FILL_18 byte fill prefix NMADEF tag $$;                 /* byte of flags
        FILL_18_BITS structure fill;
            FILL_19 bitfield length 2 fill prefix NMADEF tag $$;/* skip bits 0,1
            CTLIN_TUN bitfield mask;                        /* transmit underrun
            FILL_20 bitfield fill prefix NMADEF tag $$; /* skip bit 3
            CTLIN_RUN bitfield mask;                        /* receive underrun
            CTLIN_FMR bitfield mask;                        /* FRMR received
        end FILL_18_BITS;

/*
/* VMS-specific line counters
/*


        constant CTLIN_MBS      equals 2701  prefix NMA tag $C;/* Multicast packets transmitted
        constant CTLIN_MSN      equals 2702  prefix NMA tag $C;/* Multicast bytes transmitted
        constant CTLIN_RME      equals 2750  prefix NMA tag $C;/* PCL Remote errors, bit-mapped
                                                        /*      0  TDM bus busy
                                                        /*      1  Message rejected
                                                        /*        2  Message truncated
                                                        /*        3  Receiver offline
                                                        /*        4  Receiver busy
                                                        /*        5  Transmitter offline
        constant CTLIN_LCE      equals 2751  prefix NMA tag $C;/* PCL Local errors, bit-mapped
                                                        /*        0  Transmitter overrun
                                                        /*        1  CRC error on transmit
                                                        /*        2  CRC error on receive
                                                        /*        3  Timeouts
                                                        /*        4  Non-existant memory transmit
                                                        /*        5  Non-existant memory receive
                                                        /*        6  Buffer to small
                                                        /*        7  Failed to open channel
                                                        /*        8  Memory overflow
        constant CTLIN_MSE      equals 2752  prefix NMA tag $C;/* PCL master/secondary errors, bit-mapped
                                                        /*        1  Master down
                                                        /*        2  Now master
```

```
/*
/*    Node counters
/*


        constant CTNOD_ZER       equals 0  prefix NMA tag $C;/* Seconds since last zeroed
        constant CTNOD_BRC       equals 600  prefix NMA tag $C;/* Bytes received
        constant CTNOD_BSN       equals 601  prefix NMA tag $C;/* Bytes sent
        constant CTNOD_MRC       equals 610  prefix NMA tag $C;/* Messages received
        constant CTNOD_MSN       equals 611  prefix NMA tag $C;/* Messages sent
        constant CTNOD_CRC       equals 620  prefix NMA tag $C;/* Connects received
        constant CTNOD_CSN       equals 621  prefix NMA tag $C;/* Connects sent
        constant CTNOD_RTO       equals 630  prefix NMA tag $C;/* Response timeouts
        constant CTNOD_RSE       equals 640  prefix NMA tag $C;/* Received connect resource errors
        constant CTNOD_MLL       equals 700  prefix NMA tag $C;/* Maximum logical links active
        constant CTNOD_APL       equals 900  prefix NMA tag $C;/* Aged packet loss
        constant CTNOD_NUL       equals 901  prefix NMA tag $C;/* Node unreachable packet loss
        constant CTNOD_NOL       equals 902  prefix NMA tag $C;/* Node out-of-range packet loss
        constant CTNOD_OPL       equals 903  prefix NMA tag $C;/* Oversized packet loss
        constant CTNOD_PFE       equals 910  prefix NMA tag $C;/* Packet format error
        constant CTNOD_RUL       equals 920  prefix NMA tag $C;/* Partial routing update loss
        constant CTNOD_VER       equals 930  prefix NMA tag $C;/* Verification reject

/*
/* Server Base Specific Executor Node Counters
/*

        constant CTNOD_SRV_SYC  equals 3310  prefix NMA tag $C;/* Control buffer failures
        constant CTNOD_SRV_SYS  equals 3320  prefix NMA tag $C;/* Small buffer failures
        constant CTNOD_SRV_SYL  equals 3330  prefix NMA tag $C;/* Large buffer failures
        constant CTNOD_SRV_SYR  equals 3340  prefix NMA tag $C;/* Receive buffer failures
```

```
/*
/*        X.25 Protocol module counters
/*

        constant CTXP_ZER        equals 0  prefix NMA tag $C;/* Seconds since last zeroed
        constant CTXP_BRC        equals 1000  prefix NMA tag $C;/* Bytes received
        constant CTXP_BSN        equals 1001  prefix NMA tag $C;/* Bytes sent
        constant CTXP_BLR        equals 1010  prefix NMA tag $C;/* Data blocks received
        constant CTXP_BLS        equals 1011  prefix NMA tag $C;/* Data blocks sent
        constant CTXP_CRC        equals 1200  prefix NMA tag $C;/* Calls received
        constant CTXP_CSN        equals 1201  prefix NMA tag $C;/* Calls sent
        constant CTXP_FSR        equals 1210  prefix NMA tag $C;/* Fast selects received
        constant CTXP_FSS        equals 1211  prefix NMA tag $C;/* Fast selects sent
        constant CTXP_MSA        equals 1220  prefix NMA tag $C;/* Maximum switched circuits active
        constant CTXP_MCA        equals 1221  prefix NMA tag $C;/* Maximum channels active
        constant CTXP_RSE        equals 1230  prefix NMA tag $C;/* Received call resource errors
        constant CTXP_LIR        equals 1240  prefix NMA tag $C;/* Locally initiated resets
        constant CTXP_RIR        equals 1241  prefix NMA tag $C;/* Remotely initiated resets
        constant CTXP_NIR        equals 1242  prefix NMA tag $C;/* Network initiated resets
        constant CTXP_RST        equals 1250  prefix NMA tag $C;/* Restarts
```

```
/*
/*        X.25 Server module counters
/*


        constant CTXS_ZER        equals 0  prefix NMA tag $C;/* Seconds since last zeroed
        constant CTXS_MCA        equals 200  prefix NMA tag $C;/* Maximum circuits active
        constant CTXS_ICR        equals 210  prefix NMA tag $C;/* Incoming calls rejected, no resources
        constant CTXS_LLR        equals 211  prefix NMA tag $C;/* Logical links rejected, no resources
```

```
/*
/*        Coded parameter values
/*

/*
/* Loop test block type coded values
/*


        constant LOOP_MIX        equals 2  prefix NMA tag $C;/* Mixed
        constant LOOP_ONE        equals 1  prefix NMA tag $C;/* Ones
        constant LOOP_ZER        equals 0  prefix NMA tag $C;/* Zeroes

/*
/* Default values for loop functions
/*

        constant LOOP_DCNT       equals 1  prefix NMA tag $C;/* Default count
        constant LOOP_DSIZ       equals 40  prefix NMA tag $C;/* Default message size

/*
/* Values for LOOP HELP
/*

        constant LOOP_XMIT       equals 0  prefix NMA tag $C;/* Transmit
        constant LOOP_RECV       equals 1  prefix NMA tag $C;/* Receive
        constant LOOP_FULL       equals 2  prefix NMA tag $C;/* Full (both transmit and receive)
```

```
/*
/* State coded values
/*


        constant STATE_ON        equals 0  prefix NMA tag $C;/* On
        constant STATE_OFF       equals 1  prefix NMA tag $C;/* Off
/*
/*      circuit/line/process specific state values
/*
        constant STATE_SER       equals 2  prefix NMA tag $C;/* Service (circuit/line only)
        constant STATE_CLE       equals 3  prefix NMA tag $C;/* Cleared
/*
/*      logging specific state values
/*
        constant STATE_HOL       equals 2  prefix NMA tag $C;/* Hold
/*
/*      node specific state values
/*
        constant STATE_SHU       equals 2  prefix NMA tag $C;/* Shut
        constant STATE_RES       equals 3  prefix NMA tag $C;/* Restricted
        constant STATE_REA       equals 4  prefix NMA tag $C;/* Reachable
        constant STATE_UNR       equals 5  prefix NMA tag $C;/* Unreachable


/*
/* Looper/loader assistance coded values
/*


        constant ASS_ENA         equals 0  prefix NMA tag $C;/* Enabled
        constant ASS_DIS         equals 1  prefix NMA tag $C;/* Disabled


/*
/* Configurator surveillance coded values
/*


        constant SUR_ENA         equals 0  prefix NMA tag $C;/* Enabled
        constant SUR_DIS         equals 1  prefix NMA tag $C;/* Disabled
```

```
/*
/* Circuit/Line substate coded values
/*


        constant LINSS_STA       equals 0  prefix NMA tag $C;/* Starting
        constant LINSS_REF       equals 1  prefix NMA tag $C;/* Reflecting
        constant LINSS_LOO       equals 2  prefix NMA tag $C;/* Looping
        constant LINSS_LOA       equals 3  prefix NMA tag $C;/* Loading
        constant LINSS_DUM       equals 4  prefix NMA tag $C;/* Dumping
        constant LINSS_TRI       equals 5  prefix NMA tag $C;/* Triggering
        constant LINSS_ASE       equals 6  prefix NMA tag $C;/* Autoservice
        constant LINSS_ALO       equals 7  prefix NMA tag $C;/* Autoloading
        constant LINSS_ADU       equals 8  prefix NMA tag $C;/* Autodumping
        constant LINSS_ATR       equals 9  prefix NMA tag $C;/* Autotriggering
        constant LINSS_SYN       equals 10  prefix NMA tag $C;/* Synchronizing
        constant LINSS_FAI       equals 11  prefix NMA tag $C;/* Failed

        constant LINSS_RUN       equals 12  prefix NMA tag $C;/* Running
        constant LINSS_UNS       equals 13  prefix NMA tag $C;/* Unsyncronised
        constant LINSS_IDL       equals 14  prefix NMA tag $C;/* Idle (PSI-only)


/*
/* Circuit type coded values   [In V2, line type coded values]
/*


        constant CIRTY_POI       equals 0  prefix NMA tag $C;/* DDCMP Point
        constant CIRTY_CON       equals 1  prefix NMA tag $C;/* DDCMP Controller
        constant CIRTY_TRI       equals 2  prefix NMA tag $C;/* DDCMP Tributary
        constant CIRTY_X25       equals 3  prefix NMA tag $C;/* X25
        constant CIRTY_DMC       equals 4  prefix NMA tag $C;/* DDCMP DMC compatibility mode (DMP)
/*/*         CIRTY_LAPB, 5                /* LAPB  *** remove once all references have been changed to LAPB ***
        constant CIRTY_NI        equals 6  prefix NMA tag $C;/* NI


/*
/*        Circuit/Line Service
/*


        constant LINSV_ENA       equals 0  prefix NMA tag $C;/* Enabled
        constant LINSV_DIS       equals 1  prefix NMA tag $C;/* Disabled


/*
/* Circuit polling state
/*


        constant CIRPST_AUT      equals 1  prefix NMA tag $C;/* Automatic
        constant CIRPST_ACT      equals 2  prefix NMA tag $C;/* Active
        constant CIRPST_INA      equals 3  prefix NMA tag $C;/* Inactive
        constant CIRPST_DIE      equals 4  prefix NMA tag $C;/* Dying
        constant CIRPST_DED      equals 5  prefix NMA tag $C;/* Dead

/*
/* Circuit blocking values
/*


        constant CIRBLK_ENA      equals 0  prefix NMA tag $C;/* Enabled
        constant CIRBLK_DIS      equals 1  prefix NMA tag $C;/* Disabled

/*
/* Circuit usage values
/*


        constant CIRUS_PER       equals 0  prefix NMA tag $C;/* Permanent
        constant CIRUS_INC       equals 1  prefix NMA tag $C;/* Incoming
        constant CIRUS_OUT       equals 2  prefix NMA tag $C;/* Outgoing

/*
/* Circuit maximum receive buffers
/*


        constant CIRBF_UNL       equals 255  prefix NMA tag $C;/* Unlimited

/*
/* Circuit verification    [VMS only]
/*


        constant CIRVE_ENA       equals 0  prefix NMA tag $C;/* Enabled
        constant CIRVE_DIS       equals 1  prefix NMA tag $C;/* Disabled

/*
/* Circuit (desired) transport type    [VMS only]
/*


        constant CIRXPT_ZND      equals 1  prefix NMA tag $C;/* Z-node
        constant CIRXPT_PH2      equals 2  prefix NMA tag $C;/* Force Phase II on this circuit
        constant CIRXPT_PH3      equals 3  prefix NMA tag $C;/* Routing III
        constant CIRXPT_RO3      equals 3  prefix NMA tag $C;/* Routing III
        constant CIRXPT_NR4      equals 4  prefix NMA tag $C;/* Nonrouting Phase IV
```

```
/*
/* Line duplex coded values
/*


        constant DPX_FUL          equals 0  prefix NMA tag $C;/* Full
        constant DPX_HAL          equals 1  prefix NMA tag $C;/* Half


/*
/* Line controller mode
/*


        constant LINCN_NOR        equals 0  prefix NMA tag $C;/* Normal
        constant LINCN_LOO        equals 1  prefix NMA tag $C;/* Loop


/*
/* Line protocol values (same as CIRTY_)
/*


        constant LINPR_POI        equals 0  prefix NMA tag $C;/* DDCMP Point
        constant LINPR_CON        equals 1  prefix NMA tag $C;/* DDCMP Controller
        constant LINPR_TRI        equals 2  prefix NMA tag $C;/* DDCMP Tributary

        constant LINPR_DMC        equals 4  prefix NMA tag $C;/* DDCMP DMC compatibility mode (DMP)
        constant LINPR_LAPB       equals 5  prefix NMA tag $C;/* LAPB
        constant LINPR_NI         equals 6  prefix NMA tag $C;/* NI

        constant LINPR_BSY        equals 9  prefix NMA tag $C;/* BISYNC


/*
/* Line protocol values for the PCL-11B
/*


        constant LINPR_MAS        equals NMA$C_LINPR_CON  prefix NMA tag $C;/* Master (controls clock signals)
        constant LINPR_NEU        equals NMA$C_LINPR_TRI  prefix NMA tag $C;/* Neutral (uses master's clock signals)
        constant LINPR_SEC        equals NMA$C_LINPR_POI  prefix NMA tag $C;/* Secondary (backup for master failure)


/*
/* Line clock values
/*


        constant LINCL_EXT        equals 0  prefix NMA tag $C;/* External
        constant LINCL_INT        equals 1  prefix NMA tag $C;/* Internal


/*
```

```
/* Line type coded values  [V2 only]
/*


        constant LINTY_POI       equals 0  prefix NMA tag $C;/* DDCMP Point
        constant LINTY_CON       equals 1  prefix NMA tag $C;/* DDCMP Controller
        constant LINTY_TRI       equals 2  prefix NMA tag $C;/* DDCMP Tributary
        constant LINTY_DMC       equals 3  prefix NMA tag $C;/* DDCMP DMC compatibility mode (DMP)


/*
/* Line multicast address function code [VMS datalink only].
/* Destination and physical address function codes too [VMS datalink only].
/*


        constant LINMC_SET       equals 1  prefix NMA tag $C;/* Set address(es)
        constant LINMC_CLR       equals 2  prefix NMA tag $C;/* Clear address(es)
        constant LINMC_CAL       equals 3  prefix NMA tag $C;/* Clear entire list of multicast addresses
        constant LINMC_SDF       equals 4  prefix NMA tag $C;/* Set physical address to DECnet default


/*
/* NI line protocol access mode  [VMS datalink only]
/*


        constant ACC_SHR         equals 1  prefix NMA tag $C;/* Shared access (default protocol user)
        constant ACC_LIM         equals 2  prefix NMA tag $C;/* Limited access (point-to-point conn.)
        constant ACC_EXC         equals 3  prefix NMA tag $C;/* Exclusive access (allow no others)


/*
/* PCL-11B address mode
/*


        constant LINMO_AUT       equals 1  prefix NMA tag $C;/* Auto address mode
        constant LINMO_SIL       equals 2  prefix NMA tag $C;/* Silo address mode


/*
/* X.25 line mode
/*


        constant X25MD_DTE       equals 1  prefix NMA tag $C;/* line operates as DTE
        constant X25MD_DCE       equals 2  prefix NMA tag $C;/* line operates as DCE
        constant X25MD_DTL       equals 3  prefix NMA tag $C;/* line is a DTE in loopback
        constant X25MD_DCL       equals 4  prefix NMA tag $C;/* line is a DCE in loopback
```

```
/*
/* Node type values
/*


        constant NODTY_ROU        equals 0  prefix NMA tag $C;/* Routing Phase III
        constant NODTY_NON        equals 1  prefix NMA tag $C;/* Nonrouting Phase III
        constant NODTY_PHA        equals 2  prefix NMA tag $C;/* Phase II
        constant NODTY_AREA       equals 3  prefix NMA tag $C;/* Area
        constant NODTY_RT4        equals 4  prefix NMA tag $C;/* Routing Phase IV
        constant NODTY_NR4        equals 5  prefix NMA tag $C;/* Nonrouting Phase IV


/*
/* Node password values
/*

        constant NODPW_SET        equals 0  prefix NMA tag $C;/* Password set


/*
/* Node CPU type codes
/*


        constant CPU_8  equals 0  prefix NMA tag $C;     /* PDP-8 processor
        constant CPU_11 equals 1  prefix NMA tag $C;     /* PDP-11 processor
        constant CPU_1020         equals 2  prefix NMA tag $C;/* Decsystem 10/20 processor
        constant CPU_VAX          equals 3  prefix NMA tag $C;/* Vax processor


/*
/* Service node version coded values
/*


        constant NODSNV_PH3       equals 0  prefix NMA tag $C;/* Phase III
        constant NODSNV_PH4       equals 1  prefix NMA tag $C;/* Phase IV


/*
/* Node software type code
/*


        constant SOFT_SECL        equals 0  prefix NMA tag $C;/* Secondary loader
        constant SOFT_TERL        equals 1  prefix NMA tag $C;/* Tertiary loader
        constant SOFT_OSYS        equals 2  prefix NMA tag $C;/* Operating system
        constant SOFT_DIAG        equals 3  prefix NMA tag $C;/* Diagnostics


/*
/* Node access (and default access) codes
/*
```

```
        constant ACES_NONE        equals 0  prefix NMA tag $C;/* None
        constant ACES_INCO        equals 1  prefix NMA tag $C;/* Incoming
        constant ACES_OUTG        equals 2  prefix NMA tag $C;/* Outgoing
        constant ACES_BOTH        equals 3  prefix NMA tag $C;/* Both
        constant ACES_REQU        equals 4  prefix NMA tag $C;/* Required
```

```
/*
/* X.25 Protocol type values
/*

        constant XPRTY_BIL      equals 1  prefix NMA tag $C;/* Bilateral

/*
/* X.25 protocol state values
/*

        constant XPRST_ON       equals 0  prefix NMA tag $C;/* On
        constant XPRST_OFF      equals 1  prefix NMA tag $C;/* Off
        constant XPRST_SHU      equals 2  prefix NMA tag $C;/* Shut

/*
/* X.25 protocol multi-network support flag
/*

        constant XPRMN_ENA      equals 0  prefix NMA tag $C;/* Enabled
        constant XPRMN_DIS      equals 1  prefix NMA tag $C;/* Disabled

/*
/* X.25 protocol DTE substate values
/*

        constant XPRSB_RUN      equals 12  prefix NMA tag $C;/* Running
        constant XPRSB_UNS      equals 13  prefix NMA tag $C;/* Unsynchronized
        constant XPRSB_SYN      equals 10  prefix NMA tag $C;/* Synchronizing
```

```
/*
/*  Months of the Year Codes
/*

        constant JAN    equals 1  prefix NMA tag $C;
        constant FEB    equals 2  prefix NMA tag $C;
        constant MAR    equals 3  prefix NMA tag $C;
        constant APR    equals 4  prefix NMA tag $C;
        constant MAY    equals 5  prefix NMA tag $C;
        constant JUN    equals 6  prefix NMA tag $C;
        constant JUL    equals 7  prefix NMA tag $C;
        constant AUG    equals 8  prefix NMA tag $C;
        constant SEP    equals 9  prefix NMA tag $C;
        constant OCT    equals 10  prefix NMA tag $C;
        constant NOV    equals 11  prefix NMA tag $C;
        constant DEC    equals 12  prefix NMA tag $C;
```

```
/*
/* Service device codes (MOP)
/*

        constant SOFD_DP        equals 0  prefix NMA tag $C;/* DP11
        constant SOFD_UNA       equals 1  prefix NMA tag $C;/* UNA
        constant SOFD_DU        equals 2  prefix NMA tag $C;/* DU11
        constant SOFD_DL        equals 4  prefix NMA tag $C;/* DL11
        constant SOFD_DQ        equals 6  prefix NMA tag $C;/* DQ11
        constant SOFD_DA        equals 8  prefix NMA tag $C;/* DA11
        constant SOFD_DUP       equals 10  prefix NMA tag $C;/* DUP11
        constant SOFD_DMC       equals 12  prefix NMA tag $C;/* DMC11
        constant SOFD_DMP       equals 18  prefix NMA tag $C;/* DMP11
        constant SOFD_DTE       equals 20  prefix NMA tag $C;/* DTE20
        constant SOFD_KL8       equals 32  prefix NMA tag $C;/* KL8
        constant SOFD_DMV       equals 34  prefix NMA tag $C;/* DMV
        constant SOFD_DPV       equals 36  prefix NMA tag $C;/* DPV
        constant SOFD_DMF       equals 38  prefix NMA tag $C;/* DMF32
```

```
/*
/*         Status codes for field support routines
/*


        constant(
              SUCCESS                                     /* Unqualified success
            , SUCCFLDRPL                                  /* Success with field replaced

            ) equals 1 increment 8  prefix NMA tag $;


        constant(
              BADFID                                      /* Invalid field id code
            , BADDAT                                      /* Invalid data format
            , BADOPR                                      /* Invalid operation
            , BUFTOOSMALL                                 /* Buffer too small
            , FLDNOTFND                                   /* Field not found

            ) equals 0 increment 8  prefix NMA tag $;


/*
/*         Permanent database file ID codes
/*


        constant OPN_MIN          equals 0  prefix NMA tag $C;/* Minimum !
        constant OPN_NODE         equals 0  prefix NMA tag $C;/* Nodes
        constant OPN_LINE         equals 1  prefix NMA tag $C;/* Lines
        constant OPN_LOG          equals 2  prefix NMA tag $C;/* Logging
        constant OPN_OBJ          equals 3  prefix NMA tag $C;/* Object
        constant OPN_CIR          equals 4  prefix NMA tag $C;/* Circuit
        constant OPN_X25          equals 5  prefix NMA tag $C;/* Module X25
        constant OPN_X29          equals 6  prefix NMA tag $C;/* Module X29
        constant OPN_CNF          equals 7  prefix NMA tag $C;/* Module Configurator
        constant OPN_MAX          equals 7  prefix NMA tag $C;/* Maximum ! permanent database files
        constant OPN_ALL          equals 127  prefix NMA tag $C;/* All opened files

/*
/*         Open access codes
/*


        constant(
              OPN_AC_RO                                   /* Read Only
            , OPN_AC_RW                                   /* Read write

            ) equals 0 increment 1 prefix NMA tag $C;
```

```
/*
/*      Define Phase II NICE function codes
/*

        constant FN2_DLL        equals 2  prefix NMA tag $C;/* Down line load
        constant FN2_ULD        equals 3  prefix NMA tag $C;/* Upline Dump
        constant FN2_TRI        equals 4  prefix NMA tag $C;/* Trigger remote bootstrap
        constant FN2_LOO        equals 5  prefix NMA tag $C;/* Loop back test
        constant FN2_TES        equals 6  prefix NMA tag $C;/* Send test message to be looped
        constant FN2_SET        equals 7  prefix NMA tag $C;/* Set parameter
        constant FN2_REA        equals 8  prefix NMA tag $C;/* Read Parameter
        constant FN2_ZER        equals 9  prefix NMA tag $C;/* Zero counters
        constant FN2_LNS        equals 14  prefix NMA tag $C;/* Line service

/*
/*      Change parameters (volatile only)
/*

        constant OP2_CHNST      equals 5  prefix NMA tag $C;/* Node operational status
        constant OP2_CHLST      equals 8  prefix NMA tag $C;/* Line operational status

/*
/*      Read Information (Status and Counters only)
/*

        constant OP2_RENCT      equals 0  prefix NMA tag $C;/* Local node counters
        constant OP2_RENST      equals 1  prefix NMA tag $C;/* local node status
        constant OP2_RELCT      equals 4  prefix NMA tag $C;/* Line counters
        constant OP2_RELST      equals 5  prefix NMA tag $C;/* Line status

/*
/*      Zero counters
/*

        constant OP2_ZENCT      equals 0  prefix NMA tag $C;/* Local Node counters
        constant OP2_ZELCT      equals 2  prefix NMA tag $C;/* Line counters


/*
/*      Line entity codes
/*

        constant EN2_KNO        equals 0  prefix NMA tag $C;/* Known lines
        constant EN2_LID        equals 1  prefix NMA tag $C;/* Line id
        constant EN2_LCN        equals 2  prefix NMA tag $C;/* Line convenience name
```

```
/*
/* NML Return codes
/*


        constant STS_SUC        equals 1  prefix NMA tag $C;/* Success
        constant STS_MOR        equals 2  prefix NMA tag $C;/* Request accepted, more to come
        constant STS_PAR        equals 3  prefix NMA tag $C;/* Partial reply
/*
        constant STS_DON        equals -128  prefix NMA tag $C;/* Done
/*
        constant STS_FUN        equals -1  prefix NMA tag $C;/* Unrecognized function or option
        constant STS_INV        equals -2  prefix NMA tag $C;/* Invalid message format
        constant STS_PRI        equals -3  prefix NMA tag $C;/* Privilege violation
        constant STS_SIZ        equals -4  prefix NMA tag $C;/* Oversized management command message
        constant STS_MPR        equals -5  prefix NMA tag $C;/* Network management program error
        constant STS_PTY        equals -6  prefix NMA tag $C;/* Unrecognized parameter type
        constant STS_MVE        equals -7  prefix NMA tag $C;/* Incompatible management version
        constant STS_CMP        equals -8  prefix NMA tag $C;/* Unrecognised component
        constant STS_IDE        equals -9  prefix NMA tag $C;/* Invalid identification format
        constant STS_LCO        equals -10  prefix NMA tag $C;/* Line communication error
        constant STS_STA        equals -11  prefix NMA tag $C;/* Component in wrong state
        constant STS_FOP        equals -13  prefix NMA tag $C;/* File open error
        constant STS_FCO        equals -14  prefix NMA tag $C;/* Invalid file contents
        constant STS_RES        equals -15  prefix NMA tag $C;/* Resource error
        constant STS_PVA        equals -16  prefix NMA tag $C;/* Invalid parameter value
        constant STS_LPR        equals -17  prefix NMA tag $C;/* Line protocol error
        constant STS_FIO        equals -18  prefix NMA tag $C;/* File i/o error
        constant STS_MLD        equals -19  prefix NMA tag $C;/* Mirror link disconnected
        constant STS_ROO        equals -20  prefix NMA tag $C;/* No room for new entry
        constant STS_MCF        equals -21  prefix NMA tag $C;/* Mirror connect failed
        constant STS_PNA        equals -22  prefix NMA tag $C;/* Parameter not applicable
        constant STS_PLO        equals -23  prefix NMA tag $C;/* Parameter value too long
        constant STS_HAR        equals -24  prefix NMA tag $C;/* Hardware failure
        constant STS_OPE        equals -25  prefix NMA tag $C;/* Operation failure
        constant STS_SYS        equals -26  prefix NMA tag $C;/* System-specific management
                                                             /* function not supported
        constant STS_PGP        equals -27  prefix NMA tag $C;/* Invalid parameter grouping
        constant STS_BLR        equals -28  prefix NMA tag $C;/* Bad loopback response
        constant STS_PMS        equals -29  prefix NMA tag $C;/* Parameter missing
/*
        constant STS_ALI        equals -127  prefix NMA tag $C;/* Invalid alias identification
        constant STS_OBJ        equals -126  prefix NMA tag $C;/* Invalid object identification
        constant STS_PRO        equals -125  prefix NMA tag $C;/* Invalid process identification
        constant STS_LNK        equals -124  prefix NMA tag $C;/* Invalid link identification
```

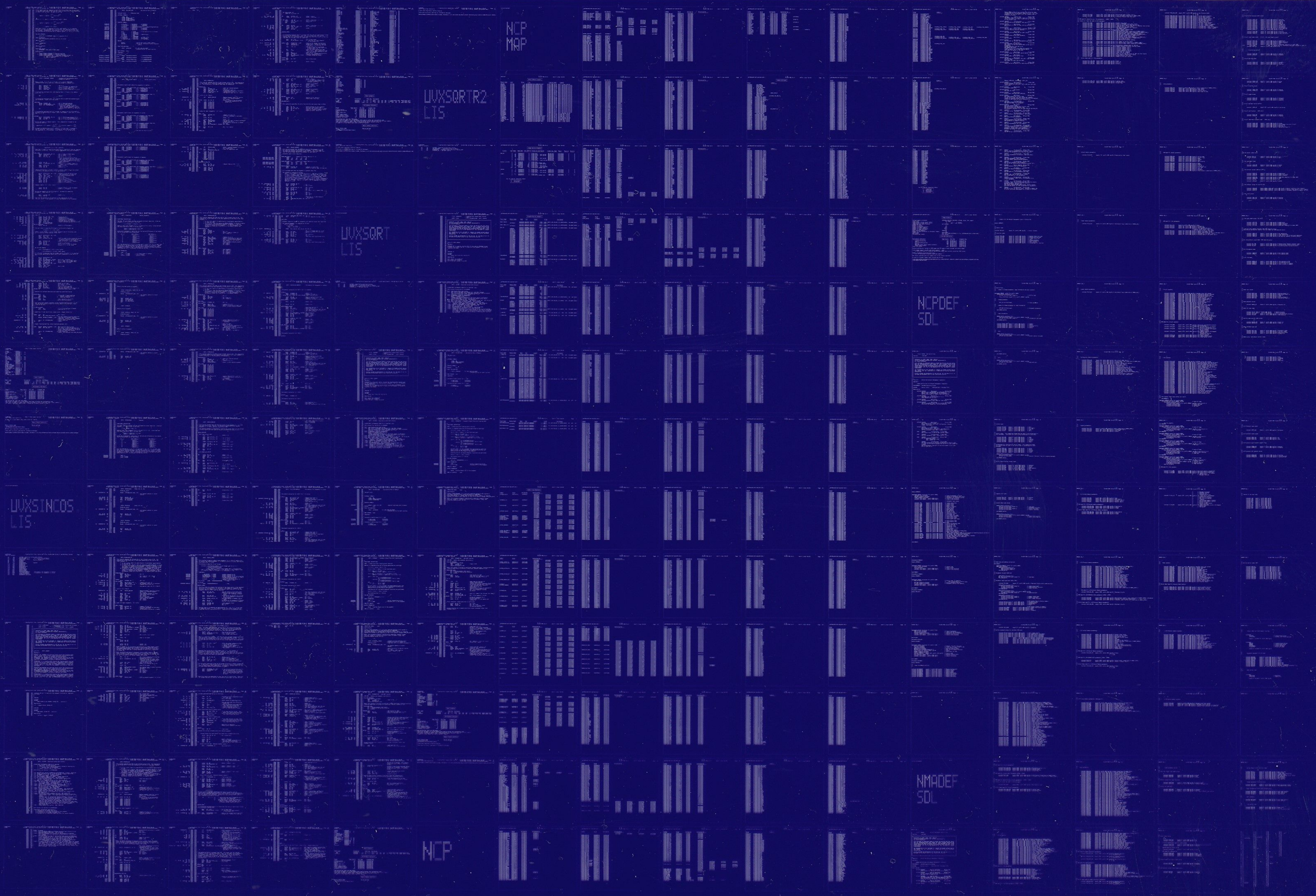
```
/*
/*      Error details
/*

/*
/*      STS_FOP and STS_FIO
/*


        constant FOPDTL_PDB     equals 0  prefix NMA tag $C;/* Permanent database
        constant FOPDTL_LFL     equals 1  prefix NMA tag $C;/* Load file
        constant FOPDTL_DFL     equals 2  prefix NMA tag $C;/* Dump file
        constant FOPDTL_SLF     equals 3  prefix NMA tag $C;/* Secondary loader
        constant FOPDTL_TLF     equals 4  prefix NMA tag $C;/* Tertiary loader
        constant FOPDTL_SDF     equals 5  prefix NMA tag $C;/* Secondary dumper

/*
/*      STS_MLD, STS_MCF
/*

        constant NCEDTL_NNA     equals 0  prefix NMA tag $C;/* No node name set
        constant NCEDTL_INN     equals 1  prefix NMA tag $C;/* Invalid node name format
        constant NCEDTL_UNA     equals 2  prefix NMA tag $C;/* Unrecognised node name
        constant NCEDTL_UNR     equals 3  prefix NMA tag $C;/* Node unreachable
        constant NCEDTL_RSC     equals 4  prefix NMA tag $C;/* Network resources
        constant NCEDTL_RJC     equals 5  prefix NMA tag $C;/* Rejected by object
        constant NCEDTL_ONA     equals 6  prefix NMA tag $C;/* Invalid object name format
        constant NCEDTL_OBJ     equals 7  prefix NMA tag $C;/* Unrecognised object
        constant NCEDTL_ACC     equals 8  prefix NMA tag $C;/* Access control rejected
        constant NCEDTL_BSY     equals 9  prefix NMA tag $C;/* Object too busy
        constant NCEDTL_NRS     equals 10  prefix NMA tag $C;/* No response from object
        constant NCEDTL_NSD     equals 11  prefix NMA tag $C;/* Node shut down
        constant NCEDTL_DIE     equals 12  prefix NMA tag $C;/* Node or object failed
        constant NCEDTL_DIS     equals 13  prefix NMA tag $C;/* Disconnect by object
        constant NCEDTL_ABO     equals 14  prefix NMA tag $C;/* Abort by object
        constant NCEDTL_ABM     equals 15  prefix NMA tag $C;/* Abort by management

/*
/*      STS_OPE
/*

        constant OPEDTL_DCH     equals  0  prefix NMA tag $C;/* Data check
        constant OPEDTL_TIM     equals  1  prefix NMA tag $C;/* Timeout
        constant OPEDTL_ORN     equals  2  prefix NMA tag $C;/* Data overrun
        constant OPEDTL_ACT     equals  3  prefix NMA tag $C;/* Unit is active
        constant OPEDTL_BAF     equals  4  prefix NMA tag $C;/* Buffer allocation failure
        constant OPEDTL_RUN     equals  5  prefix NMA tag $C;/* Protocol running
        constant OPEDTL_DSC     equals  6  prefix NMA tag $C;/* Line disconnected
        constant OPEDTL_FTL     equals  8  prefix NMA tag $C;/* Fatal hardware error
        constant OPEDTL_MNT     equals 11  prefix NMA tag $C;/* DDCMP maintainance message received
        constant OPEDTL_LST     equals 12  prefix NMA tag $C;/* Data lost due to buffer size mismatch
        constant OPEDTL_THR     equals 13  prefix NMA tag $C;/* Threshold error
        constant OPEDTL_TRB     equals 14  prefix NMA tag $C;/* Tributary malfunction
        constant OPEDTL_STA     equals 15  prefix NMA tag $C;/* DDCMP start message received
```

```
end NMADEF7;

end_module $NMADEF;
```

```
NC

!
!
!
!

MA

$S

$S

!
!
!
!

MA

$S

$S

!
```

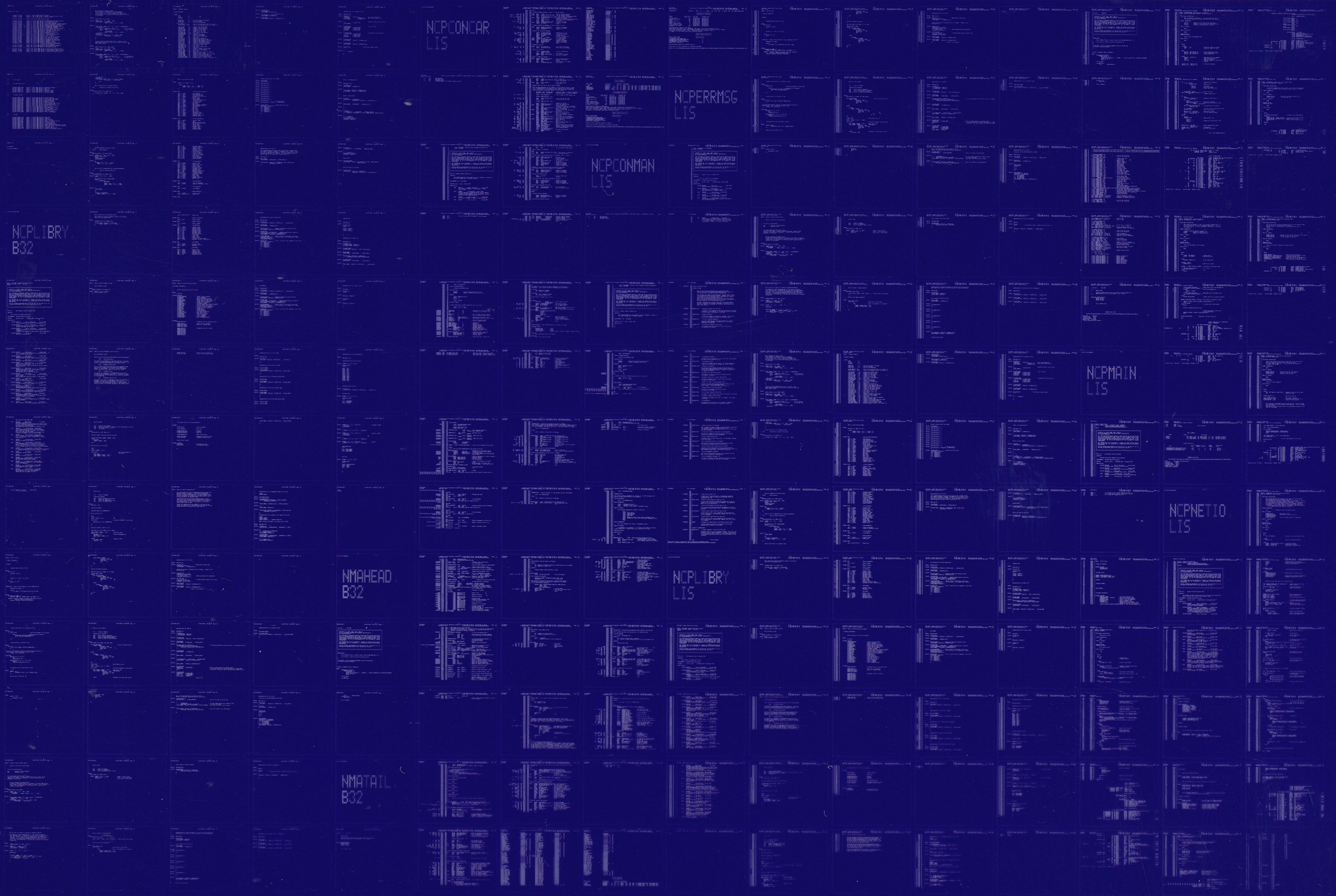
0266 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



0267 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



NCPCONCAR
LIS

NCPERRMSG
LIS

NCPCONMAN
LIS

NCPLIBRY
B32

NCPMAIN
LIS

NCPNETIO
LIS

NMAHEAD
B32

NCPLIBRY
LIS

NMATAIL
B32