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
```

```
**FILE**ID**NCPDEF
```

```
NN      NN    CCCCCCCC  PPPPPPPP    DDDDDDDD    EEEEEEEEEE  FFFFFFFFFF
NN      NN    CCCCCCCC  PPPPPPPP    DDDDDDDD    EEEEEEEEEE  FFFFFFFFFF
NN      NN  CC          PP      PP  DD      DD  EE          FF
NN      NN  CC          PP      PP  DD      DD  EE          FF
NNNN    NN  CC          PP      PP  DD      DD  EE          FF
NNNN    NN  CC          PP      PP  DD      DD  EE          FF
NN  NN  NN  CC          PPPPPPPP    DD      DD  EEEEEEEE    FFFFFFFF
NN  NN  NN  CC          PPPPPPPP    DD      DD  EEEEEEEE    FFFFFFFF
NN    NNNN  CC          PP          DD      DD  EE          FF
NN    NNNN  CC          PP          DD      DD  EE          FF
NN      NN  CC          PP          DD      DD  EE          FF                ....
NN      NN  CC          PP          DD      DD  EE          FF                ....
NN      NN    CCCCCCCC  PP          DDDDDDDD    EEEEEEEEEE  FF                ....
NN      NN    CCCCCCCC  PP          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  NCPDEF  NCP 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: 4-October-1979
{
{ MODIFIED BY:
{
{       V03-006 RPG0006         Bob Grosso              08-Feb-1983
{               Add new data type MODPRM, so ACT$SAVPRM can store
{               MODULE type in PDB$G_VRB_EVE for NCP$TALOG.
{               Add new data types to support SHOW CONFIGURATOR.
{
{       V03-005 RPG0005         Bob Grosso              09-Nov-1982
{               Add AADR type to reflect that it is a
{               node address with an area.
{
{       V03-004 RPG0004         Bob Grosso              29-Sep-1982
{               Add new parameter type, AREA for node areas.
{
{       V03-003 RPG0003         Bob Grosso              14-Jul-1982
{               Add new parameter type, HEX for hexidecimal numbers
{               which are not padded out like hex passwords.
{
```

```
{       V03-002 RPG0002         Bob Grosso              14-Jul-1982
{               Add codes to return index for
{               Module X29-Server, X25-Trace, Configurator, Console,
{               Loader, and Looper.
{
{       V03-001 RPG0001         Bob Grosso              17-Jun-1982
{               Add codes to return index for
{               Module X25-Protocol, X25-Server, and X25-Access.
{               Add new parameter type, RNGL = range lists.
{
{       V004    TMH0004         Tim Halvorsen   11-Jan-1982
{               Add 3 byte field in LCB for NML version number.
{
{       V003    TMH0003         Tim Halvorsen   11-Nov-1981
{               Add ESCI parameter type - store source circuit
{
{       V002    TMH0002         Tim Halvorsen   20-Jul-1981
{               Add new parameter types:
{                       SAD = Subaddress range
{                       OBJ = Object ID
{
{       V001    TMH0001         Tim Halvorsen   17-Jun-1981
{               Add SDB convention to indicate system-specific entity type,
{               in order to distinguish between the two entity type numbering
{               schemes.
{               Add new parameter type ENT for multiply-coded circuit user,
{               entity type and ID.
{--
```

```
module $PBKDEF;


aggregate PBKDEF structure fill prefix PBK$;
    TYPECODE byte unsigned;                             /* Type of parameter to store
    PDB_ADR longword unsigned;                          /* Address of parameter data block
    PARAM longword unsigned;                            /* Parameter for savparam routine
    constant SIZE equals . prefix PBK$ tag K;           /* Size of the structure
    constant SIZE equals . prefix PBK$ tag C;           /* Size of the structure

                                                        /* Parameter type values

    constant LOW        equals 1  prefix PBK tag $K;    /* Lowest value here
    constant LITB       equals 1  prefix PBK tag $K;    /* Literal byte
    constant NUMB       equals 2  prefix PBK tag $K;    /* Numeric byte
    constant NUMW       equals 3  prefix PBK tag $K;    /* Numeric word
    constant NUML       equals 4  prefix PBK tag $K;    /* Numeric longword
    constant TKN        equals 5  prefix PBK tag $K;    /* Token string
    constant TKNQ       equals 6  prefix PBK tag $K;    /* Quoted token
    constant NADR       equals 7  prefix PBK tag $K;    /* Node address
    constant HXPS       equals 8  prefix PBK tag $K;    /* Hex password
    constant STRQ       equals 9  prefix PBK tag $K;    /* Quoted string
    constant TRIPL      equals 10  prefix PBK tag $K;   /* Version triple
    constant LITL       equals 11  prefix PBK tag $K;   /* Long word literal
    constant PRVL       equals 12  prefix PBK tag $K;   /* Privilege list
    constant PRVC       equals 13  prefix PBK tag $K;   /* Privilege list clear
    constant ESET       equals 14  prefix PBK tag $K;   /* Setup event parameter
    constant ECLS       equals 15  prefix PBK tag $K;   /* Store event class
    constant EMSK       equals 16  prefix PBK tag $K;   /* Store single event
    constant ERNG       equals 17  prefix PBK tag $K;   /* Store event type range
    constant EWLD       equals 18  prefix PBK tag $K;
    constant ESNO       equals 19  prefix PBK tag $K;   /* Store source node
    constant ESLI       equals 20  prefix PBK tag $K;   /* Store source line
/*         MODPRM,   added at bottom        /* Store module name
    constant ESEX       equals 21  prefix PBK tag $K;   /* Source as executor node
    constant ENT        equals 22  prefix PBK tag $K;   /* Entity type and ID
    constant "END"      equals 23  prefix PBK tag $K;   /* End of PCL list
    constant SAD        equals 24  prefix PBK tag $K;   /* Subaddress range
    constant OBJ        equals 25  prefix PBK tag $K;   /* Object ID
    constant ESCI       equals 26  prefix PBK tag $K;   /* Store source circuit
    constant RNGL       equals 27  prefix PBK tag $K;   /* Range lists
    constant HEX        equals 28  prefix PBK tag $K;   /* Hexidecimal numbers
    constant AREA       equals 29  prefix PBK tag $K;   /* byte of zero and byte of Node Area
    constant AADR       equals 30  prefix PBK tag $K;   /* Node Area and Address
                                                        /*  NOTE: Used instead of NUMW to avoid hassle of handling area by action ro
    constant NIADR      equals 31  prefix PBK tag $K;   /* NI address, HEX image printed backwardss
    constant DELTIM     equals 32  prefix PBK tag $K;   /* Delta time, (Hours, Minutes, Seconds)
    constant DAYTIM     equals 33  prefix PBK tag $K;   /* Day and time (Day, Month, Hour, Minutes, Seconds)
    constant LITLST     equals 34  prefix PBK tag $K;   /* Variable length list of coded data
    constant MODPRM     equals 35  prefix PBK tag $K;   /* Store module name
    constant HIGH       equals 35  prefix PBK tag $K;   /* Highest value here


end PBKDEF;
```

```
end_module $PBKDEF;

module $PDBDEF;


aggregate PDBDEF structure fill prefix PDB$;
    STS_FLG byte unsigned;                              /* Status flag
    DATA character;                                     /* Data is here
    constant SIZE equals . prefix PDB$ tag K;           /* Size of the structure
    constant SIZE equals . prefix PDB$ tag C;           /* Size of the structure


end PDBDEF;

end_module $PDBDEF;

module $SDBDEF;


aggregate SDBDEF structure fill prefix SDB$;
    ENT_TYP byte;                                       /* Entity type. If negative,
                                                        /*    then system-specific entity type.
    ENT_ADR longword unsigned;                          /* Entity parameter address
    PCL_ADR longword unsigned;                          /* Parameter control list address
    constant SIZE equals . prefix SDB$ tag K;
    constant SIZE equals . prefix SDB$ tag C;


end SDBDEF;

end_module $SDBDEF;

module $PCLDEF;
```

```
aggregate PCLDEF structure fill prefix PCL$;
    PRM_TYP byte unsigned;                                     /* Type of parameter
    PRM_ID word unsigned;                                      /* Code value for parameter
    PDB_ADR longword unsigned;                                 /* Address of PDB for parameter
    constant SIZE equals . prefix PCL$ tag K;                  /* Size of the structure
    constant SIZE equals . prefix PCL$ tag C;                  /* Size of the structure


end PCLDEF;

end_module $PCLDEF;

module $LCBDEF;


aggregate LCBDEF structure fill prefix LCB$;
    STS byte unsigned;                                         /* Status, true for link open
    PH2 byte unsigned;                                         /* Phase II, true for phase II NML
    CHAN word unsigned;                                        /* Link channel number
    MBXCHN word unsigned;                                      /* Mailbox channel number
    NMLVERS byte unsigned dimension 3;                         /* NML version number (3 bytes)
    FILL_1 byte fill prefix LCBDEF tag $$;                     /* Spare
    NCBCNT longword unsigned;                                  /* Descriptor for NCB
    NCBPTR longword unsigned;
    constant NCBSIZE    equals 100  prefix LCB tag $C;         /* Size of NCB
    NCB character length 100;                                  /* Network Control block
    constant SIZE equals . prefix LCB$ tag K;                  /* Size of structure
    constant SIZE equals . prefix LCB$ tag C;                  /* Size of structure


end LCBDEF;

end_module $LCBDEF;

module $NCPDEF;

/*
/*   Index the MODULE entities
/*


constant ENT_MODCNF      equals 1  prefix NCP tag $C;     /* Module Configurator
constant ENT_MODCNS      equals 2  prefix NCP tag $C;     /* Module Console
constant ENT_MODLOA      equals 3  prefix NCP tag $C;     /* Module Loader
constant ENT_MODLOO      equals 4  prefix NCP tag $C;     /* Module Looper
constant ENT_MODACC      equals 5  prefix NCP tag $C;     /* Module X25-Access
constant ENT_MODPRO      equals 6  prefix NCP tag $C;     /* Module X25-Protocol
constant ENT_MODSER      equals 7  prefix NCP tag $C;     /* Module X25-Server
constant ENT_MODTRC      equals 8  prefix NCP tag $C;     /* Module X25-Trace
constant ENT_MOD29S      equals 9  prefix NCP tag $C;     /* Module X29-Server
```

```
end_module $NCPDEF;
```

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

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY