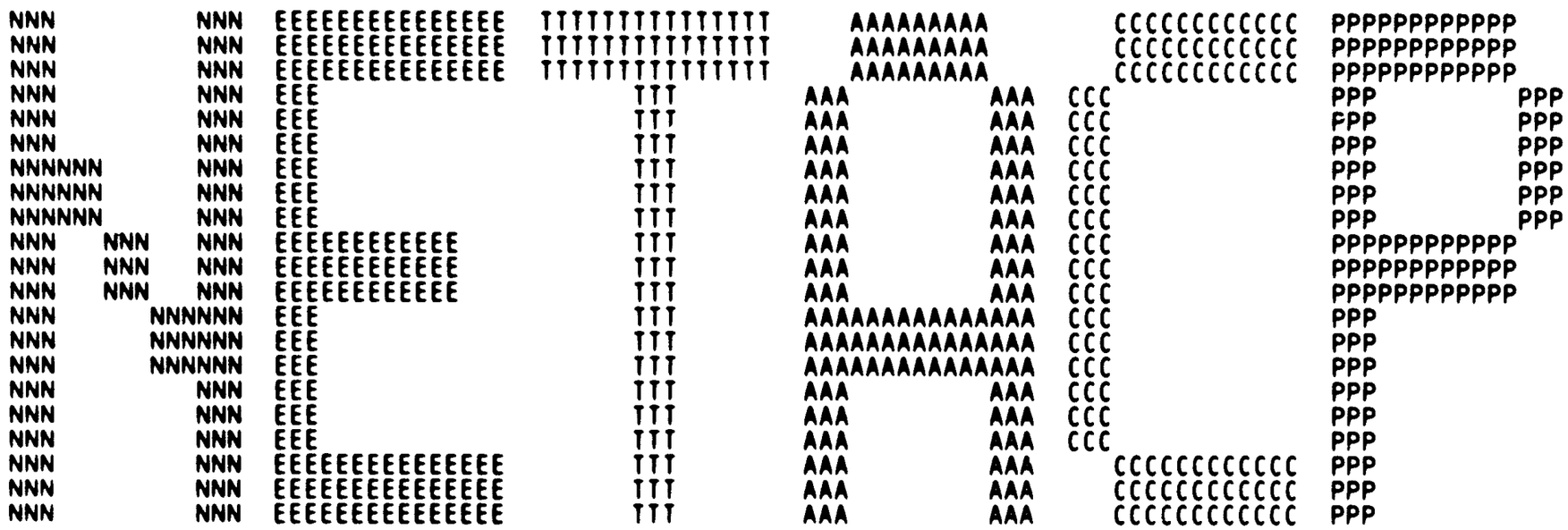
NNN NNN EEEEEEEEEEEEEEE TTTTTTTTTTTTTTT AAAAAAAAA CCCCCCCCCCC PPPPPPPPPPPP
NNN NNN EEEEEEEEEEEEEEE TTTTTTTTTTTTTTT AAAAAAAAA CCCCCCCCCCC PPPPPPPPPPPP
NNN NNN EEEEEEEEEEEEEEE TTTTTTTTTTTTTTT AAAAAAAAA CCCCCCCCCCC PPPPPPPPPPPP
NNN NNN EEE TTT AAA AAA CCC PPP PPP
NNN NNN EEE TTT AAA AAA CCC PPP PPP
NNN NNN EEE TTT AAA AAA CCC PPP PPP
NNNNNN NNN EEE TTT AAA AAA CCC PPP PPP
NNNNNN NNN EEE TTT AAA AAA CCC PPP PPP
NNNNNN NNN EEE TTT AAA AAA CCC PPP PPP
NNN NNN NNN EEEEEEEEEEE TTT AAA AAA CCC PPPPPPPPPPPP
NNN NNN NNN EEEEEEEEEEE TTT AAA AAA CCC PPPPPPPPPPPP
NNN NNN NNN EEEEEEEEEEE TTT AAA AAA CCC PPPPPPPPPPPP
NNN NNNNNN EEE TTT AAAAAAAAAAAAAAA CCC PPP
NNN NNNNNN EEE TTT AAAAAAAAAAAAAAA CCC PPP
NNN NNNNNN EEE TTT AAAAAAAAAAAAAAA CCC PPP
NNN NNN EEE TTT AAA AAA CCC PPP
NNN NNN EEE TTT AAA AAA CCC PPP
NNN NNN EEE TTT AAA AAA CCC PPP
NNN NNN EEEEEEEEEEEEEEE TTT AAA AAA CCCCCCCCCCC PPP
NNN NNN EEEEEEEEEEEEEEE TTT AAA AAA CCCCCCCCCCC PPP
NNN NNN EEEEEEEEEEEEEEE TTT AAA AAA CCCCCCCCCCC PPP

FILEID**NETCTL

```
NN      NN  EEEEEEEEEE  TTTTTTTTTT    CCCCCCCC  TTTTTTTTTT  LL
NN      NN  EEEEEEEEEE  TTTTTTTTTT    CCCCCCCC  TTTTTTTTTT  LL
NN      NN  EE              TT      CC              TT      LL
NN      NN  EE              TT      CC              TT      LL
NNNN    NN  EE              TT      CC              TT      LL
NNNN    NN  EE              TT      CC              TT      LL
NN  NN  NN  EEEEEEEE        TT      CC              TT      LL
NN  NN  NN  EEEEEEEE        TT      CC              TT      LL
NN    NNNN  EE              TT      CC              TT      LL
NN    NNNN  EE              TT      CC              TT      LL
NN      NN  EE              TT      CC              TT      LL              ....
NN      NN  EE              TT      CC              TT      LL              ....
NN      NN  EEEEEEEEEE      TT        CCCCCCCC      TT      LLLLLLLLLL      ....
NN      NN  EEEEEEEEEE      TT        CCCCCCCC      TT      LLLLLLLLLL      ....

  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
```

```
{
{       NETCTL.SDL
{
{ 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.                  *
{*                                                                           *
{*                                                                           *
{*****************************************************************************

{ MODIFIED BY:
{
{       V03-018 RNG0018         Rod Gamache             10-Feb-1984
{               Add real LLI database and NDC counter block.
{               Fix up definition of a WQE.
{
{       V03-017 RNG0017         Rod Gamache             6-Feb-1984
{               Add NDI SNV (system node version) parameter.
{
{       V03-016 PRB0313         Paul Beck       6-Feb-1984  12:14
{               Fix duplicate definition of CNR$S_SEM_TAB (didn't occur with MDL)
{
{       V015    TMH0015         Tim Halvorsen           11-Jul-1983
{               Add LNI ALI (cluster node address) parameter.
{
{       V014    TMH0014         Tim Halvorsen           26-Apr-1983
{               Add PPUNA device code.
{
{       V013    RNG0013         Rod Gamache             29-Mar-1983
{               Add new CNR routine locations.
{
{       V012    TMH0012         Tim Halvorsen           14-Feb-1983
{               Add line buffer size parameter.
{
{       V011    TMH0011         Tim Halvorsen           16-Dec-1982
{               Add Ethernet protocol type to the line database.
{
{       V010    TMH0010         Tim Halvorsen           31-Aug-1982
```

```
{               Remove cell for circuit "designated router".  It is
{               now an action routine.
{
{       V009    TMH0009         Tim Halvorsen           30-Jun-1982
{               Add Phase IV line, circuit and node parameters.
{               Add UNA device type code.
{               Remove listen timer, since it's now read-only.
{               Remove NDI$C_DTY_xxx symbols, since they are now replaced
{               by ADJ$C_PTY_xxx symbols.
{
{       V008    TMH0008         Tim Halvorsen           16-Jun-1982
{               Add SPI database.
{               Add ACC_NE access code for "no external access at all".
{
{       V007    TMH0007         Tim Halvorsen           04-Apr-1982
{               Move $NDBDEF from NETUSR to here.
{               Remove spurious comments about X.25 database parameters.
{               Add "DEFAULT" action routine to CNR block.
{               Add X.25 mode byte to PLI block.
{               Remove obsolete SEM_$ constant, which is no longer
{               needed after cleanup of CNR creation macros in NETCONFIG.
{               Rename CNF$T_MASK to CNF$L_MASK.
{               Rename CNR$T_SEM_TAB to CNR$L_SEM_TAB.
{               Add KMX and X25 to DEVTRN type codes.
{
{       V03-006 ADE0033         A.Eldridge              15-Feb-1982
{               Added LNI$W_PIQ.
{
{       V03-005 ADE0032         A.Eldridge              06-Jan-1981
{               Removed the RTT parameter from the circuit database.
{
{       V03-004 ADE0031         A.Eldridge              15-Dec-1981
{               Added the CNF$V_NOEXT bit to the CNF definitions.
{
{       V03-003 ADE0030         A.Eldridge              30-Nov-1981
{               Added support for proxy logins the OBI, NDI, and LNI
{               structures.
{
```

```
{
{ Configuration Data Base Root Block  (CNR)
{
{ This block serves as the listhead for the CNFs of a particular
{ component in the configuration data base.  It contains all of the
{ component's semantics.
{
module $CNRDEF;

aggregate CNRDEF  union fill prefix CNR$;
    CNRDEF_BITS0 structure fill;
        SEM_OFF bitfield length 8;                      /* Byte offset from top of CNF to field
        SEM_TYP bitfield length 3;                      /* Field type (bit,string,byte,word etc)
        SEM_ACC bitfield length 3;                      /* Field access control
        SEM_RT bitfield;                                /* "Field" is actually an index of a routine to call
        SEM_Z bitfield;                                 /* Set if field may be zero
        SEM_MAX bitfield length 16;                     /* Max value byte or word may be assigned
    end CNRDEF_BITS0;
    CNRDEF_BITS1 structure fill;
        FILL_1 bitfield length 16 fill prefix CNRDEF tag $$;/* Advance to SEM_MAX
        SEM_SMX bitfield length 12;                     /* Field to store max size string
        SEM_TAB bitfield length 4;                      /* Holds i.d. of string parse table
    end CNRDEF_BITS1;
                                                        /*
                                                        /* Define CNF$x_SEM_TYP values
                                                        /*
    constant SEM_BIT    equals 0  prefix CNR tag $C;    /* Type = bit
    constant SEM_B      equals 1  prefix CNR tag $C;    /* Type = byte
    constant SEM_W      equals 2  prefix CNR tag $C;    /* Type = word
    constant SEM_L      equals 3  prefix CNR tag $C;    /* Type = longword
    constant SEM_STR    equals 4  prefix CNR tag $C;    /* Type = string descriptor
                                                        /*
                                                        /* Define field access control
                                                        /*
    constant ACC_RW     equals 0  prefix CNR tag $C;    /* General read/write
    constant ACC_RO     equals 1  prefix CNR tag $C;    /* Read only
    constant ACC_R      equals 1  prefix CNR tag $C;    /* Read only
    constant ACC_WO     equals 2  prefix CNR tag $C;    /* Write only (for passwords, etc)
    constant ACC_W      equals 2  prefix CNR tag $C;    /* Write only (for passwords, etc)
    constant ACC_CW     equals 3  prefix CNR tag $C;    /* Conditionally writable
    constant ACC_C      equals 3  prefix CNR tag $C;    /* Conditionally writable
    constant ACC_ER     equals 4  prefix CNR tag $C;    /* External read only (e.g., if from QIO)
    constant ACC_E      equals 4  prefix CNR tag $C;    /* External read only (e.g., if from QIO)
    constant ACC_NE     equals 5  prefix CNR tag $C;    /* No external read or write access
    constant ACC_N      equals 5  prefix CNR tag $C;    /* No external read or write access
                                                        /*
                                                        /* Define string parse table i.d.'s
                                                        /*
    constant SEM_T      equals 0  prefix CNR tag $C;    /* Transparent - all characters are legal
    constant SEM_A      equals 1  prefix CNR tag $C;    /* Upper case alpha or numerics only
    constant SEM_F      equals 2  prefix CNR tag $C;    /* Parse string as if file specification
end CNRDEF;

    aggregate CNRDEF1 structure fill prefix CNR$;
    FLINK_OVERLAY union fill;
```

```
        FLINK longword unsigned;                    /* Forward link
        COLBTE longword unsigned;                   /* Pointer to the collating tree entry for NDI CNR
    end FLINK_OVERLAY;
    BLINK_OVERLAY union fill;
        BLINK longword unsigned;                    /* Backward link
        NAMEBTE longword unsigned;                  /* Pointer to the name tree entry for NDI CNR
    end BLINK_OVERLAY;
    SIZE word unsigned;                             /* Block size
    TYPE byte unsigned;                             /* Block type (one of the NFB$C_DB_... codes)
    FLG byte unsigned;                              /* Flag bits
    SIZ_CNF word unsigned;                          /* Size of associated CNF without any string storage
    MAX_INX word unsigned;                          /* Maximum field index defined for this database
    FLD_LOCK longword unsigned;                     /* Storage for bit id of conditional write gate
    FLD_COLL longword unsigned;                     /* Storage for collating bit i.d.
    ACT_QIO longword unsigned;                      /* Ptr to QIO preprocessor for this database
    ACT_SHOW longword unsigned;                     /* Ptr to "show" QIO to a specific CNF
    ACT_DFLT longword unsigned;                     /* Ptr to "defaulting" action routine
    ACT_INSERT longword unsigned;                   /* Ptr to CNF "pre-insert" action routine
    ACT_DELETE longword unsigned;                   /* Ptr to CNF "mark-for-delete" action routine
    ACT_REMOVE longword unsigned;                   /* Ptr to CNF "post-remove" action routine
    SCANNER longword unsigned;                      /* Ptr to CNF scanner
    INSERT longword unsigned;                       /* Ptr to CNF real insertion routine
    SPCSCAN longword unsigned;                      /* Ptr to CNF special scan routine
    VEC_ACT longword unsigned dimension 16;         /* Vector of action routine pointers
    END_ACT longword unsigned;                      /* Mark the end of the vector
    VEC_MAND longword unsigned dimension 24;        /* Vector of mandatory field i.d.'s
    END_MAND longword unsigned;                     /*
    VEC_UNIQ longword unsigned dimension 16;        /* Vector of i.d.'s of fields required to be
    END_UNIQ longword unsigned;                     /* unique
    constant MAX_INX    equals 95  prefix CNR tag $C;   /* Maximum field index possible (0 indexed => 96 indexes)
    SEM_TAB_OVERLAY union fill;                     /* (Avoid duplicate definition of CNR$S_SEM_TAB)
        SEM_TAB longword unsigned;                  /* Semantic table -- 4 bytes for each of 96 indexes
        SEM_TABLE longword unsigned dimension 96;   /* ...synonym to reserve the space...
    end SEM_TAB_OVERLAY;
    constant ''LENGTH'' equals . prefix CNR$ tag K;     /* Structure size
    constant ''LENGTH'' equals . prefix CNR$ tag C;     /* Structure size
                                                        /*F         SEM_TAB,L,4*CNR$C_MAX_INX        /* Semantic table
                                                        /*-also see CNF$L_MASK definition
                                                        /*
                                                        /*  The following defines the format of each entry in the semantic table
                                                        /*
    end CNRDEF1;

end_module $CNRDEF;

module $CNFDEF;
```

```
/*
/* Configuration Data Block (CNF)
/*
/* This is a general block structure used to carry a sub-block in the
/* configuration data base of the NETACP.  The  CNF  and sub-block
/* semantics for each component type are store in the associate CNR
/* described above.
/*


aggregate CNFDEF structure fill prefix CNF$;
    FLINK_OVERLAY union fill;
        FLINK longword unsigned;                    /* Forward link
        COLBTE longword unsigned;                   /* Pointer to the collating tree entry for NDI CNF
    end FLINK_OVERLAY;
    BLINK_OVERLAY union fill;
        BLINK longword unsigned;                    /* Backward link
        NAMEBTE longword unsigned;                  /* Pointer to the name tree entry for NDI CNF
    end BLINK_OVERLAY;
    SIZE word unsigned;                             /* Block size
    TYPE byte unsigned;                             /* Block type
    FLG_OVERLAY union fill;
        FLG byte unsigned;                          /* Flags defined as follows:
        FLG_BITS structure fill;
            FLG_CNR bitfield mask;                  /* Block is actually a CNR
            FLG_DELETE bitfield mask;               /* Block is a temporary CNF or marked for delete
            FLG_ACP bitfield mask;                  /* Block is a catch-all used by the ACP
            FLG_NOEXT bitfield mask;                /* Block is used internally only. It is not to be
                                                    /* displayed to the "external" world, i.e., above the
                                                    /* $QIO interface.
            FLG_MRK1 bitfield mask;                 /* Special flags with different meanings for each
            FLG_MRK2 bitfield mask;                 /* database. These flags are defined locally in module
            FLG_MRK3 bitfield mask;                 /* NETCNFACT.MAR.
        end FLG_BITS;
    end FLG_OVERLAY;
    OFF_FREE word unsigned;                         /* Self-relative byte offset to free storage
    SIZ_FREE word unsigned;                         /* Bytes left in CNF free storage
    SIZ_USED word unsigned;                         /* Number of bytes used for storing strings
    ID word unsigned;                               /* Database dependent indentification data
    "BOOLEAN" word unsigned;                        /* Storage for values of parameters of type "bit"
    FILL_1 word fill prefix CNFDEF tag $$;          /* Spare -- reserved for future use
    "MASK" longword unsigned dimension 3;           /*see CNR$C_MAX_INX definition
    constant "LENGTH" equals . prefix CNF$ tag K;   /* Structure size
    constant "LENGTH" equals . prefix CNF$ tag C;   /* Structure size
                                                    /*F       MASK,L,CNR$C_MAX_INX/32       /* One bit in mask for each possib
                                                    /* The bit is set while the field is active.

end CNFDEF;

end_module $CNFDEF;

module $NDIDEF;
```

```
/*
/* REMOTE NODE INFORMATION
/*

                                                        /*
                                                        /*       The following are commonly defined for all nodes
                                                        /*

aggregate NDIDEF structure fill prefix NDI$;
    ADD word unsigned;                                  /* Node address - zero if NDI is for local node
    CTI word unsigned;                                  /* Counter timer (units = sec)
    CTA longword unsigned;                              /* Abslute due timer for counters to be logged
    S_NNA longword unsigned;                            /* Name
                                                        /*
                                                        /*       The following are defined for some, but not most, nodes
                                                        /*
    S_NLI longword unsigned;                            /* Line used if NDI is a "loopback" node
    S_PUS longword unsigned;                            /* Priv user id
    S_PAC longword unsigned;                            /* Priv account
    S_PPW longword unsigned;                            /* Priv password
    S_NUS longword unsigned;                            /* NonPriv user id
    S_NAC longword unsigned;                            /* NonPriv account
    S_NPW longword unsigned;                            /* NonPriv psw
    S_RPA longword unsigned;                            /* Receive password
    S_TPA longword unsigned;                            /* Transmit password
    ACC byte unsigned;                                  /* Access switch (inbound,outbound,etc)
    PRX byte unsigned;                                  /* Proxy access switch (inbound, outbound, etc)
    SNV byte unsigned;                                  /* System node version
    FILL_1 byte fill prefix NDIDEF tag $$;              /* Spare used for alignment. Reserved for future use
                                                        /*
                                                        /*       The following are for nodes to be downline-loaded or upline-dumped
                                                        /*
    FILL_2 byte fill prefix NDIDEF tag $$;              /* Spare used for alignment. Reserved for future use
    SDV byte unsigned;                                  /* Service device type
    CPU byte unsigned;                                  /* CPU type
    STY byte unsigned;                                  /* Software type
    IHO word unsigned;                                  /* Host address (input)
    OHO word unsigned;                                  /* Host address (output)
    DAD longword unsigned;                              /* Dump address
    DCT longword unsigned;                              /* Dump count
    S_SLI longword unsigned;                            /* Service line
    S_SPA longword unsigned;                            /* Service password
    S_LOA longword unsigned;                            /* Load file
    S_SLO longword unsigned;                            /* Secondary loader
    S_TLO longword unsigned;                            /* Tertiary loader
    S_SID longword unsigned;                            /* Software ID
    S_DUM longword unsigned;                            /* Dump file
    S_SDU longword unsigned;                            /* Secondary dumper
    S_DFL longword unsigned;                            /* Diagnostic load file
    S_HWA longword unsigned;                            /* NI hardware address for node
    constant 'LENGTH' equals . prefix NDI$ tag K;       /* Structure size
    constant 'LENGTH' equals . prefix NDI$ tag C;       /* Structure size
                                                        /* (used to down-line load a node which isn't
                                                        /*  up on the network yet)
end NDIDEF;

end_module $NDIDEF;
```

```
module $LNIDEF;
```

```
/*
/* LOCAL NODE INFORMATION
/*
                                                        /*
                                                        /*      Define local node states
                                                        /*
constant STA_ON equals 0  prefix LNI tag $C;            /* Node available for general use
constant STA_OFF        equals 1  prefix LNI tag $C;    /* Node shutting down, no connects allowed
constant STA_SHUT       equals 2  prefix LNI tag $C;    /* No new connects allowed - shutting down
constant STA_RSTR       equals 3  prefix LNI tag $C;    /* Node available for outbound connects only
constant STA_INIT       equals 4  prefix LNI tag $C;    /* State used for ACP initialization
                                                        /*
                                                        /*      Define the CNF structure
                                                        /*

aggregate LNIDEF structure fill prefix LNI$;
    ADD word unsigned;                                  /* Node address
    STA byte unsigned;                                  /* State
    ETY byte unsigned;                                  /* Local node type

    MLK word unsigned;                                  /* Maximum links allowed
    MAD word unsigned;                                  /* Maximum node address
    MBU word unsigned;                                  /* Maximum transport buffers
    MCO word unsigned;                                  /* Maximum cost

    MHO byte unsigned;                                  /* Maximum hops
    MVI byte unsigned;                                  /* Maximum visits
    MLN byte unsigned;                                  /* Maximum circuits (used to be called lines)
    LPD byte unsigned;                                  /* Default LOOP data

    LPC word unsigned;                                  /* Default LOOP count
    LPL word unsigned;                                  /* Default LOOP length
    LPH byte unsigned;                                  /* Default LOOP help type
    FILL_1 byte fill prefix LNIDEF tag $$;              /* spare for alignment

    BUS word unsigned;                                  /* Transport forwarding buffer size
                                                        /* (maximum size that we will receive and forward)
    SBS word unsigned;                                  /* Transport segment buffer size
                                                        /* (maximum size that we will transmit)
    RSI word unsigned;                                  /* Routine supression interval (units = sec)
    IAT word unsigned;                                  /* Inactivity timer (units = sec)

    ITI word unsigned;                                  /* Incoming timer (units = sec)
    OTI word unsigned;                                  /* Outgoing timer (units = sec)
    RTI word unsigned;                                  /* Routing timer (units = sec)
    BRT word unsigned;                                  /* Broadcast routing timer (units = sec)

    MBE word unsigned;                                  /* Maximum broadcast endnodes
    MBR word unsigned;                                  /* Maximum broadcast routers

    DFA byte unsigned;                                  /* Delay factor
    DWE byte unsigned;                                  /* Delay weight
    RFA byte unsigned;                                  /* Retransmit factor
    DAC byte unsigned;                                  /* Default access (inbound,outbound,etc)

    DPX byte unsigned;                                  /* Default proxy access (inbound,outbound,etc)
```

```
    PIQ word unsigned;                          /* Pipeline quota
    FILL_2 byte fill prefix LNIDEF tag $$;      /* Spare used for alignment. Reserved for future use.

    SAD longword unsigned;                      /* X.25 sub-address range

    MAR byte unsigned;                          /* Maximum areas
    AMH byte unsigned;                          /* Area maximum hops
    AMC word unsigned;                          /* Area maximum cost
    ALI word unsigned;                          /* Alias local address (cluster node address)

    S_NAM longword unsigned;                    /* Node name
    S_IDE longword unsigned;                    /* System identification
    S_NVE longword unsigned;                    /* NSP version
    S_RVE longword unsigned;                    /* Routing version
    S_MVE longword unsigned;                    /* Network Management version
    constant "LENGTH" equals . prefix LNI$ tag K;   /* Structure size
    constant "LENGTH" equals . prefix LNI$ tag C;   /* Structure size

end LNIDEF;

end_module $LNIDEF;

module $LLIDEF;
```

```
/*
/* LOGICAL LINK INFORMATION
/*
                                                        /*
                                                        /*       The following are commonly defined for all node counter blocks
                                                        /*

/*
/* Node Counter Block  (NDC)
/*
/* The following data block is used to maintain statistics for each node in the
/* network.  A hash of these structures is contained in NETACP.
/*

aggregate NDCDEF structure fill prefix NDC$;
    ABS_TIM longword unsigned;                          /* Absolute time counter block was last zeroed
                                                        /*
                                                        /*      Network services layer counters
                                                        /*
    RSE word unsigned;                                  /* Trasmitted connect rejects due to resource errors
    RTO word unsigned;                                  /* Response timeouts
    CRC word unsigned;                                  /* Connects received
    CSN word unsigned;                                  /* Connects sent
    BRC longword unsigned;                              /* Bytes received
    BSN longword unsigned;                              /* Bytes sent
    PRC longword unsigned;                              /* Packets received
    PSN longword unsigned;                              /* Packets sent
    constant 'LENGTH' equals . prefix NDC$ tag K;       /* Structure size
    constant 'LENGTH' equals . prefix NDC$ tag C;       /* Structure size

end NDCDEF;

                                                        /*
                                                        /*       The following are commonly defined for all logical links
                                                        /*
aggregate LLIDEF structure fill prefix LLI$;
    XWB longword unsigned;                              /* Pointer to XWB
    LLN word unsigned;                                  /* Local link number
    PNA word unsigned;                                  /* Partner's node address
                                                        /*
                                                        /*      Network services layer Running Total counters
                                                        /*
    NDC_RT byte dimension NDC$C_LENGTH tag Z;           /* Running total counters
                                                        /*
                                                        /*      Network services layer Last Zeroed counters
                                                        /*
    NDC_LZ byte dimension NDC$C_LENGTH tag Z;           /* Last zeroed counters

    constant 'LENGTH' equals . prefix LLI$ tag K;       /* Structure size
    constant 'LENGTH' equals . prefix LLI$ tag C;       /* Structure size

end LLIDEF;

end_module $LLIDEF;

module $OBIDEF;
```

```
/*
/* NETWORK OBJECT INFORMATION
/*
                                                        /*
                                                        /*      Define CNF storage
                                                        /*

aggregate OBIDEF structure fill prefix OBI$;
    NUM byte unsigned;                                  /* Object number
    PRX byte unsigned;                                  /* Proxy login switch (inbound, outbound, etc)
    CHN word unsigned;                                  /* Channel over which declaration occurred
    LPR longword unsigned;                              /* Low order privilege mask
    HPR longword unsigned;                              /* High order privilege mask
    UCB longword unsigned;                              /* Associated NET UCB if declared task
    PID longword unsigned;                              /* Associated process i.d. if declared task

    S_NAM longword unsigned;                            /* Name
    S_FID longword unsigned;                            /* File id
    S_USR longword unsigned;                            /* User id
    S_ACC longword unsigned;                            /* Account
    S_PSW longword unsigned;                            /* Password
    constant "LENGTH" equals . prefix OBI$ tag K;       /* Structure size
    constant "LENGTH" equals . prefix OBI$ tag C;       /* Structure size
end OBIDEF;

end_module $OBIDEF;

module $CRIDEF;
```

```
/*
/* CIRCUIT INFORMATION
/*
                                                        /*
                                                        /*      Define CNF storage for fields used internally
                                                        /*

aggregate CRIDEF structure fill prefix CRI$;
    S_NAM longword unsigned;                            /* Circuit name
    OWPID longword unsigned;                            /* PID of temporary owner of line in service state
    CTA longword unsigned;                              /* Absolute due time for counter logging
                                                        /*
                                                        /*      Define CNF storage for fields defined by the NICE protocol
                                                        /*
    STA byte unsigned;                                  /* State
    CHN byte unsigned;                                  /* X.25 Channel No.
    LCT word unsigned;                                  /* Counter timer

    S_LOO longword unsigned;                            /* Loopback name
    HET word unsigned;                                  /* Hello timer
    FILL_1 word fill prefix CRIDEF tag $$;              /* spare

    COS byte unsigned;                                  /* Cost
    MRC byte unsigned;                                  /* Maximum recalls
    RCT word unsigned;                                  /* Recall timer

    S_NUM longword unsigned;                            /* Call Number

    POL byte unsigned;                                  /* Polling state
    PLS byte unsigned;                                  /* Polling substate
    TYP byte unsigned;                                  /* Type
    FILL_2 byte fill prefix CRIDEF tag $$;              /* spare for alignment

    S_DTE longword unsigned;                            /* DTE

    MBL word unsigned;                                  /* Maximum block
    MWI byte unsigned;                                  /* Maximum window
    TRI byte unsigned;                                  /* Tributary

    BBT word unsigned;                                  /* Babble timer
    TRT word unsigned;                                  /* Transmit timer

    CHN word unsigned;                                  /* X.25 channel
    USE byte unsigned;                                  /* X.25 Usage
    MRB byte unsigned;                                  /* Maximum receive buffers

    MTR byte unsigned;                                  /* Maximum transmits
    ACB byte unsigned;                                  /* Active base
    ACI byte unsigned;                                  /* Active increment
    IAB byte unsigned;                                  /* Inactive base

    IAI byte unsigned;                                  /* Inactive increment
    IAT byte unsigned;                                  /* Inactive threshold
    DYB byte unsigned;                                  /* Dying base
    DYI byte unsigned;                                  /* Dying increment
```

```
    DYT byte unsigned;                                  /* Dying threshold
    DTH byte unsigned;                                  /* Dead threshold
    XPT byte unsigned;                                  /* Transport protocol
    MRT byte unsigned;                                  /* Maximum routers on NI

    RPR byte unsigned;                                  /* Router priority on NI
    constant "LENGTH" equals . prefix CRI$ tag K;       /* Structure size
    constant "LENGTH" equals . prefix CRI$ tag C;       /* Structure size
end CRIDEF;

end_module $CRIDEF;

module $PLIDEF;
```

```
/*
/* PHYSICAL LINE INFORMATION
/*
                                                        /*
                                                        /*      Define CNF storage for fields used internally
                                                        /*

aggregate PLIDEF structure fill prefix PLI$;
    S_NAM longword unsigned;                            /* Line name
    CTA longword unsigned;                              /* Absolute due time for counter logging
                                                        /*
                                                        /*      Define CNF storage for fields defined by the NICE protocol
                                                        /*
    BFN byte unsigned;                                  /* Number of buffers in receive pool
    STA byte unsigned;                                  /* State
    SUB byte unsigned;                                  /* Substate
    PRO byte unsigned;                                  /* Protocol

    LCT word unsigned;                                  /* Counter timer
    STI word unsigned;                                  /* Service timer

    HTI word unsigned;                                  /* Holdback timer
    RTT word unsigned;                                  /* Retransmit timer

    MBL word unsigned;                                  /* Maximum block
    MRT byte unsigned;                                  /* Maximum retransmits
    MWI byte unsigned;                                  /* Maximum window

    SLT word unsigned;                                  /* Scheduling timer
    DDT word unsigned;                                  /* Dead timer

    DLT word unsigned;                                  /* Delay timer
    SRT word unsigned;                                  /* Stream timer

    S_HWA longword unsigned;                            /* NI hardware address [READ ONLY]

    S_MCD longword unsigned;                            /* X.25 KMX microcode dump file [WRITE ONLY - ONE SHOT]

    EPT word unsigned;                                  /* Ethernet protocol type
    MOD byte unsigned;                                  /* X.25 mode (DTE, DCE, etc.)
    FILL_1 byte fill prefix PLIDEF tag $$;              /* (spare for alignment)

    BFS word unsigned;                                  /* Buffer size to override executor buffer size
    constant "LENGTH" equals . prefix PLI$ tag K;       /* Structure size
    constant "LENGTH" equals . prefix PLI$ tag C;       /* Structure size

end PLIDEF;

end_module $PLIDEF;

module $EFIDEF;
```

```
/*
/* EVENT LOGGING FILTER INFORMATION
/*
                                                        /*
                                                        /*        Define the CNF structure
                                                        /*

aggregate EFIDEF structure fill prefix EFI$;
    SIN word unsigned;                                  /* Sink node address
    SP1 word unsigned;                                  /* Spare
    B1 longword unsigned;                               /* For user defined use
    B2 longword unsigned;                               /* For user defined use

    S_EVE longword unsigned;                            /* Event list
    S_SB1 longword unsigned;                            /* For user defined use
    S_SB2 longword unsigned;                            /* For user defined use
    S_SB3 longword unsigned;                            /* For user defined use
    constant "LENGTH" equals . prefix EFI$ tag K;       /* Structure size
    constant "LENGTH" equals . prefix EFI$ tag C;       /* Structure size

end EFIDEF;

end_module $EFIDEF;

module $ESIDEF;

/*
/* EVENT LOGGING SINK INFORMATION
/*
                                                        /*
                                                        /*        Define logging sink states
                                                        /*
constant STA_ON equals 0  prefix ESI tag $C;            /* Logging is on
constant STA_OFF        equals 1  prefix ESI tag $C;    /* Logging is off
constant STA_HLD        equals 2  prefix ESI tag $C;    /* Hold events
                                                        /*
                                                        /*   Define logging sink types
                                                        /*
constant SNK_CON        equals 1  prefix ESI tag $C;    /* Console
constant SNK_FIL        equals 2  prefix ESI tag $C;    /* File
constant SNK_MON        equals 3  prefix ESI tag $C;    /* Monitor
                                                        /*
                                                        /*        Define the CNF structure
                                                        /*

aggregate ESIDEF structure fill prefix ESI$;
    SNK byte unsigned;                                  /* Sink type
    STA byte unsigned;                                  /* Sink state
    SP1 word unsigned;                                  /* Spare
    B1 longword unsigned;                               /* For user defined use
    B2 longword unsigned;                               /* For user defined use

    S_LNA longword unsigned;                            /* Sink name
    S_SB1 longword unsigned;                            /* For user defined use
    S_SB2 longword unsigned;                            /* For user defined use
    S_SB3 longword unsigned;                            /* For user defined use
```

```
    constant "LENGTH" equals . prefix ESI$ tag K;       /* Structure size
    constant "LENGTH" equals . prefix ESI$ tag C;       /* Structure size
end ESIDEF;

end_module $ESIDEF;

module $SPIDEF;
```

```
/*
/* NETWORK SERVER PROCESS INFORMATION
/*
                                                        /*
                                                        /*      Define CNF storage
                                                        /*

aggregate SPIDEF structure fill prefix SPI$;
    PID longword unsigned;                              /* Server PID
    IRP longword unsigned;                              /* IRP of waiting DECLSERV QIO. 0 if process active
    RNA word unsigned;                                  /* Remote node address which initially started server
    CHN word unsigned;                                  /* Channel associated with L_IRP (waiting DECLSERV)

    S_ACS longword unsigned;                            /* ACS used initially to start server process
    S_RID longword unsigned;                            /* Remote user ID which initially started server
    S_SFI longword unsigned;                            /* Last (current) filespec given to server
    S_NCB longword unsigned;                            /* Last (current) NCB given to server
    S_PNM longword unsigned;                            /* Last (current) process name given to server
    constant "LENGTH" equals . prefix SPI$ tag K;       /* Structure size
    constant "LENGTH" equals . prefix SPI$ tag C;       /* Structure size
end SPIDEF;

end_module $SPIDEF;

module $WQEDEF;
```

```
/*
/*  Work Queue Elements (WQE) are used by the ACP to serialize and standardize
/*  all schedulable but non-IRP oriented work.  Datalink state transition
/*  control and events originating from ASTs are examples.
/*
/*  The WQE structure is depicted below.  The WQE$B_SUB field is used to
/*  determine if any special processing is needed when the WQE is queued or
/*  dequeued as follows:
/*


aggregate WQEDEF structure fill prefix WQE$;
    FLINK longword unsigned;                        /* Queue forward link
    BLINK longword unsigned;                        /* Queue backward link
    SIZE word unsigned;                             /* Bytes allocated for the WQE
    TYPE byte unsigned;                             /* Structure type code
    SUB byte unsigned;                              /* Structure sub-type code as follows:
                                                    /*
    constant SUB_BAS    equals 0  prefix WQE tag $C;    /* The WQE is the base of a list - NOTE low bit clear
                                                        /* for this SUB constant only!!!
                                                        /*
    constant SUB_ACP    equals 1  prefix WQE tag $C;    /* The WQE was spawned during normal internal ACP
                                                        /* activity, e.g., during IO$_ACPCONTROL Qio activity.
                                                        /* No special action is required when it is queued.
                                                        /* When it is dequeued, dispatch directly to the action
                                                        /* routine which responsible for deallocating it.
                                                        /*
    constant SUB_AST    equals 3  prefix WQE tag $C;    /* The WQE is the consequence of a miscellaneous AST,
                                                        /* e.g., a datalink Qio AST.  If its the first entry
                                                        /* queued then $WAKE the ACP.  When it is dequeued,
                                                        /* dispatch directly to the action routine -  which is
                                                        /* responsible for deallocating it.
                                                        /*
    constant SUB_MBX    equals 5  prefix WQE tag $C;    /* The WQE is the consequence of a mailbox read AST. If
                                                        /* it is the first element queued then $WAKE the ACP.
                                                        /* When it is dequeued, it is sent to the mailbox
                                                        /* servicing routine - which permenantly owns the WQE.
                                                        /*
    constant SUB_TIM    equals 7  prefix WQE tag $C;    /* The WQE is the consequence of a timer AST. If it is
                                                        /* the first element queued then $WAKE the ACP.  When
                                                        /* dequeued, another VMS timer must be set if there are
                                                        /* any more elements in the WQE timer queue.
    ACTION longword unsigned;                       /* Action routine address
    PM1_OVERLAY union fill;
        PM1 longword unsigned;                      /* Action routine first parameter
        PM1_FIELDS structure fill;
            EVT byte unsigned;                      /* Event code - interpreted in the context of the
                                                    /* QUAL field
            QUAL byte unsigned;                     /* REQIDT qualifier as follows:
                                                    /*
            constant QUAL_DLL   equals 1  prefix WQE tag $C;/* A data link event - REQIDT is the LPD$W_PTH value
            constant QUAL_RTG   equals 2  prefix WQE tag $C;/* An ACP routing event - REQIDT is always zero
            constant QUAL_CTM   equals 3  prefix WQE tag $C;/* A counter timer event - REQIDT identifies data base
            constant QUAL_ACT   equals 4  prefix WQE tag $C;/* The ACP active timer
            REQIDT word unsigned;                   /* Request identifier - interpreted in the context of
                                                    /* the QUAL field.
```

```
        end PM1_FIELDS;
    end PM1_OVERLAY;
    PM2 longword unsigned;                          /* Action routine second parameter
    EVL_PKT longword unsigned;                      /* Ptr to the packet header if the WQE is being used
                                                    /* for event logging
    EVL_CODE word unsigned;                         /* Event logging code if WQE is used for event logging
    EVL_DT1 byte unsigned;                          /* Event logging immediate data
    EVL_DT2 byte unsigned;                          /* Event logging immediate data
    ADJ_INX word unsigned;                          /* Adjacency to which this WQE applies (0 if none)
    SPARE word unsigned;                            /* SPARE WORD
    constant "LENGTH" equals . prefix WQE$ tag K;   /* Structure size and begining of data area
    constant "LENGTH" equals . prefix WQE$ tag C;    * Structure size and begining of data area
end WQEDEF;

end_module $WQEDEF;

module $DLLQIODEF;
```

```
/*
/* DLLQIO  - Datalink $QIO parameter block
/*


aggregate DLLQIODEF structure fill prefix DLLQIO$;
    FUNC longword unsigned;                             /* Function code
    P1 longword unsigned;                               /* QIO P1 parameter
    P2 longword unsigned;                               /* QIO P2 parameter
    P3 longword unsigned;                               /* QIO P3 parameter
    P4 longword unsigned;                               /* QIO P4 parameter
    P5 longword unsigned;                               /* QIO P5 parameter
    constant "LENGTH" equals . prefix DLLQIO$ tag K;    /* Structure size
    constant "LENGTH" equals . prefix DLLQIO$ tag C;    /* Structure size
end DLLQIODEF;

end_module $DLLQIODEF;

module $DEVTRNDEF;
```

```
/*
/* DEVTRN - Device translation table
/*


aggregate DEVTRNDEF structure fill prefix DEVTRN$;
    NETMAN byte unsigned;                               /* Count of Network Management device name
    NETMAN character length 5;                          /* Network Management device name text
    VMS byte unsigned;                                  /* Count of VMS device name
    VMS character length 3;                             /* VMS device name text
    DEV byte unsigned;                                  /* Device code
                                                        /* Define device codes
    constant(
          UNK                                            * Unknown device
        , DMC                                           /* DMC-11
        , PCL                                           /* PCL-11
        , DMF                                           /* DMF-32
        , CI                                            /* CI-780
        , DMP                                           /* DMP-11
        , DUP                                           /* DUP-11 (for X.25)
        , KMS                                           /* KMC-11 (for X.25)
        , X25                                           /* X.25 datalink (datalink mapping)
        , UNA                                           /* DEUNA (Ethernet)
        , PPUNA                                         /* DEUNA operating in point-to-point mode
                                                        /* (internal testing purposes only!)
        ) equals 0 increment 1  prefix DEVTRN$C_D tag EV;

    PROT byte unsigned;                                 /* Default device protocol (NMA$C_LINPR...)
    CHAR_OVERLAY union fill;
        CHAR byte unsigned;                             /* Device characteristics
        constant "LENGTH" equals . prefix DEVTRN$ tag K;/* Structure size
        constant "LENGTH" equals . prefix DEVTRN$ tag C;/* Structure size
        CHAR_BITS structure fill;
            MULTI bitfield mask;                        /* Multi-unit device
        end CHAR_BITS;
    end CHAR_OVERLAY;
end DEVTRNDEF;

end_module $DEVTRNDEF;

module $NDBDEF;
```

```
/*+
/* NDB - DEFINE OPCOM MESSAGE CODES
/*-


                                                        /* Message codes for OPCOM
constant(
      MSG_START                                         /* DECnet starting
      MSG_SHUT                                          /* DECnet shutting down
    ) equals 1 increment 1  prefix NDB tag $C;

end_module $NDBDEF;
```

NETACP

NDDRIVER
MAP

NETDRIVER
MAP

NETSERVER
MAP

DLEDEF
SDL

NETACP
MAP

NETNPAGED
SDL

NMALIBRY
LIS

NETCTL
SDL