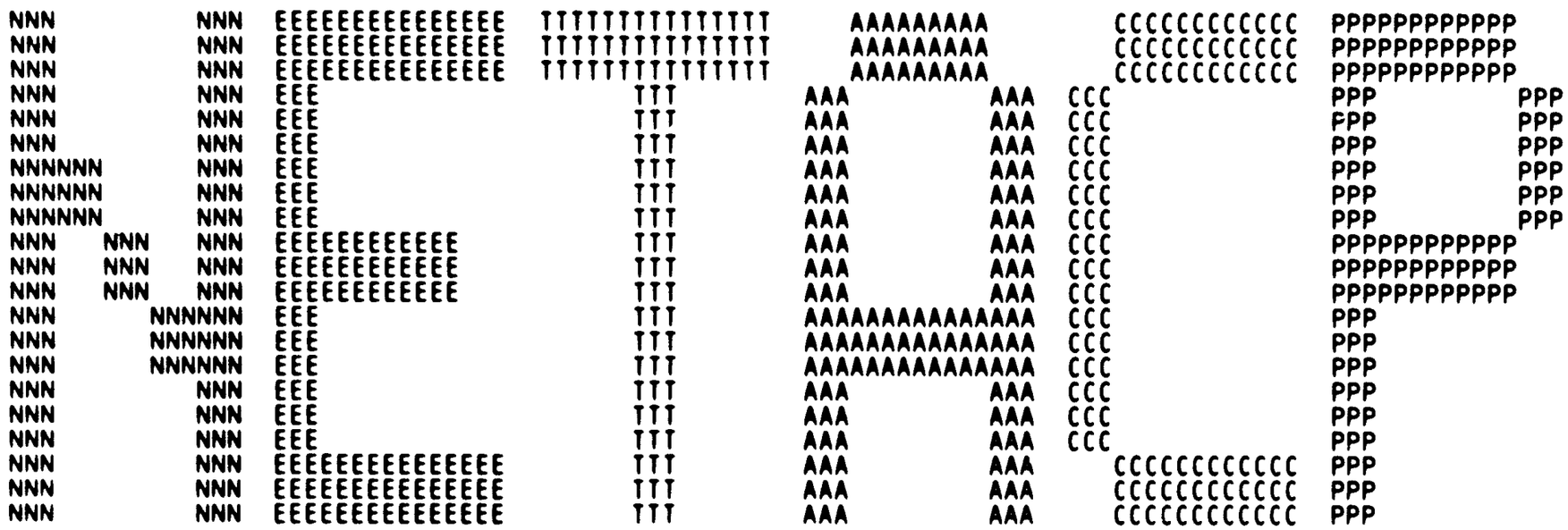
NNN NNN EEEEEEEEEEEEEEE TTTTTTTTTTTTTTT AAAAAAAAA CCCCCCCCCCC PPPPPPPPPPPP
NNN NNN EEEEEEEEEEEEEEE TTTTTTTTTTTTTTT AAAAAAAAA CCCCCCCCCCC PPPPPPPPPPPP
NNN NNN EEEEEEEEEEEEEEE TTTTTTTTTTTTTTT AAAAAAAAA CCCCCCCCCCC PPPPPPPPPPPP
NNN NNN EEE TTT AAA AAA CCC PPP PPP
NNN NNN EEE TTT AAA AAA CCC PPP PPP
NNN 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

```
NN       NN  EEEEEEEEEE  TTTTTTTTTT  UU      UU    SSSSSSSS  RRRRRRRR
NN       NN  EEEEEEEEEE  TTTTTTTTTT  UU      UU    SSSSSSSS  RRRRRRRR
NN       NN  EE              TT      UU      UU  SS          RR      RR
NN       NN  EE              TT      UU      UU  SS          RR      RR
NNNN     NN  EE              TT      UU      UU  SS          RR      RR
NNNN     NN  EE              TT      UU      UU  SS          RR      RR
NN  NN   NN  EEEEEEEE        TT      UU      UU    SSSSSS    RRRRRRRR
NN  NN   NN  EEEEEEEE        TT      UU      UU    SSSSSS    RRRRRRRR
NN     NNNN  EE              TT      UU      UU          SS  RR  RR
NN     NNNN  EE              TT      UU      UU          SS  RR  RR
NN       NN  EE              TT      UU      UU          SS  RR    RR        ....
NN       NN  EE              TT      UU      UU          SS  RR    RR        ....
NN       NN  EEEEEEEEEE      TT      UUUUUUUUUU  SSSSSSSS    RR      RR      ....
NN       NN  EEEEEEEEEE      TT      UUUUUUUUUU  SSSSSSSS    RR      RR      ....

  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
```

```
{
{ NETUSR.SDL - system definitions for NETWORK ACP interface
{
{ 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:  VAX/VMS System Macro Libraries
{
{ ABSTRACT:
{
{       This file contains the SDL source for NETWORK control blocks.
{
{ ENVIRONMENT:
{
{       n/a
{
{ AUTHOR:  The VMS Group          CREATION DATE:  1-Aug-1976
{
{ MODIFICATION HISTORY:
{
{       V024    RNG0024         Rod Gamache             28-Feb-1984
{               Add IPID field for LLI database.
{
{       V023    MKP0001         Kathy Perko             06-Dec-1983
{               Add Node parameter, SERVICE NODE VERSION, for down line
{               load.
{
{       V022    TMH0022         Tim Halvorsen           04-Jul-1983
{               Add XAI database for X.25 gateway support.
{               Add LNI ALIAS (cluster node name) parameter.
{
{       V021    TMH0021         Tim Halvorsen           20-Apr-1983
```

```
{               Add Service (DLE) database.
{
{       V020    RNG0020         Rod Gamache             18-Apr-1983
{               Reserve some more 'database' codes for PSI.
{
{       V019    TMH0019         Tim Halvorsen           04-Mar-1983
{               Add DEVNAM circuit parameter (action routine).
{
{       V018    TMH0018         Tim Halvorsen           14-Feb-1983
{               Add line buffer size parameter.
{
{       V017    TMH0017         Tim Halvorsen           07-Dec-1982
{               Add "Next node to destination" node name, as well as
{               the node number which is already a parameter.
{               Add Ethernet protocol type parameter, so that Phase IV
{               can be used with another protocol type on the NI.
{
{       V016    TMH0016         Tim Halvorsen           05-Nov-1982
{               Add area database.
{
{       V015    TMH0015         Tim Halvorsen           23-Sep-1982
{               Add adjacency database.
{
{       V014    TMH0014         Tim Halvorsen           30-Jun-1982
{               Update Phase IV line, circuit and node parameters.
{               Add line action routine to get the physical device
{               name being used by a line (including unit number).
{               Add LOOP LINE parameters and function code.
{               Change DEST SAD from a string to a longword.
{               Change DEST NODE from a string to a longword.
{
{       V013    TMH0013         Tim Halvorsen           16-Jun-1982
{               Add SPI (Server Process) database and parameters.
{               Add DECLSERV function code.
{               Make NFB$C_LENGTH equal to NFB$L_FLDID, and not be
{               4 bytes afterwards, since it should be possible to
{               issue a ACPCONTROL function with NO fields, and without
{               any ENDOFLIST field (since it isn't required anymore).
{
{       V012    TMH0012         Tim Halvorsen           04-Apr-1982
{               Remove STRT_KEY.  Remove ERR_STRT detail code.
{               Remove NFB$C_COLLATE generic field ID.
{               Increase size of P2 context area to 64 bytes.
{               Remove obsolete function codes.
{               Rename most of the symbols and fields to make the interface
{               more understandable:
{                       NFB$V_KNO    --> NFB$V_MULT
{                       NFB$V_UPD    --> NFB$V_ERRUPD
{                       NFB$V_NOUPD --> NFB$V_NOCTX
{                       NFB$C_P2STRTLNG --> NFB$C_CTX_SIZE
{               Move $NDBDEF to NETCTL.MDL
{               Add CNT (counters) parameter for XDI, XS5 and XS9 databases.
{               Remove fields which were commented out.
{               Fix classification of CRI/PLI parameters in comment field.
{               Add key and oper fields for second search key.
{               Remove obsolete NUL parameters from all databases.
```

```
{               Add Ethernet parameters.
{               Remove obsolete FNDNEXT operator, and make FNDMIN and FNDMAX
{               unavailable through the QIO interface, since they don't work.
{               Add ERR_P5 and ERR_P6 error detail codes, so that they can
{               be returned if P5 or P6 were specified.
{               Add circuit DLM flag, to indicate X.25 DLM access
{               (corresponds to the NICE OWNER parameter).
{               Add ERR_FLAGS and ERR_OPER2 error detail codes.
{               Add symbols to make accessing field IDs and values easier
{               for BLISS programs.
{               Add DTE MAXIMUM CIRCUITS parameter.
{
{       V3-011  TMH0011         Tim Halvorsen           25-Feb-1982
{               Add X.25 parameters.
{
{       V3-010  ADE0033         A.Eldridge              01-Mar-1982
{               Increased NFB$C_DB_MAX to allow access to the X25
{               databases.
{
{       V3-009  ADE0032         A.ELdridge              15-Feb-1982
{               Added 'pipeline quota' (PIQ) to the LNI database
{
{       V3-008  ADE0031         A.Eldridge              06-Jan-1982
{               Removed the 'retransmit timer' (RTT) from the circuit
{               database.
{
{       V3-007  ADE0030         Al Eldridge             30-Nov-1981
{               Added support for proxy logins
{
{       V3-006  ??
{
{       V3-005  ADE0005         Al Eldridge              2-Sep-1981
{               Update NFBDEF to get rid of overlapping symbols
{
{       V3-004  TMH0004         Tim Halvorsen           30-Aug-1981
{               Get rid of obsolete definitions and add $NFBDEF.
{
{---
```

```
{
{  Network Function Block
{
{  This is the format of the P1 buffer used for NETACP's IO$_ACPCONTROL Qio's.
{
module $NFBDEF;
                                                        /*
                                                        /*       The following generic field identifiers are defined for all databa
                                                        /*
constant ENDOFLIST      equals 0  prefix NFB tag $C;    /* Used to terminate the field i.d.
constant WILDCARD       equals 1  prefix NFB tag $C;    /* Field i.d. used for "match all" database searches
constant CTX_SIZE       equals 64  prefix NFB tag $C;   /* Length of context area in P2 buffer

                                                        /*
                                                        /*       The following codes are passed in the second IOSB longword to qual
                                                        /*       as SS$_ILLCNTRFUNC error.
                                                        /*
                                                        /* The high order word of these error codes must be 0
                                                        /* so that they won't be confused with field i.d.s
constant(
      FCT                                               /* Unrecognized NFB$B_FCT value.
    , DB                                                /* Unrecognized NFB$B_DATABASE value.
    , P1                                                /* The P1 buffer is invalid.
    , P2                                                /* The P2 buffer is invalid.
    , P3                                                /* The P3 buffer is invalid.
    , P4                                                /* The P4 buffer is invalid.
    , P5                                                /* The P5 buffer should not have been specified.
    , P6                                                /* The P6 buffer should not have been specified.
    , CELL                                              /* Unrecognized NFB$B_CELL value.
    , OPER                                              /* Unrecognized NFB$B_OPER value.
    , SRCH                                              /* Unrecognized NFB$L_SRCH_KEY field ID
    , SRCH2                                             /* Unrecognized NFB$L_SRCH2_KEY field ID
    , OPER2                                             /* Unrecognized NFB$B_OPER2 value.
    , FLAGS                                             /* Undefined bits in NFB$B_FLAGS were not zero.
    ) equals 1 increment 1  prefix NFB$_E tag RR;

                                                        /*
                                                        /*       Define the P1 buffer format
                                                        /*

aggregate NFBDEF structure fill prefix NFB$;
    FCT byte unsigned;                                  /* A function code as follows:
                                                        /* Function codes for the NFB
                                                        /*  (leaving room for 20 obsolete function codes)
    constant(
          DECLNAME                                      /* Declare name
        , DECLOBJ                                       /* Declare object
        , DECLSERV                                      /* Declare server process available
        ) equals 21 increment 1  prefix NFB tag $C;

                                                        /* Resume defining function codes
                                                        /*  (leave room for 4 obsolete function codes)
    constant(
          LOGEVENT                                      /* Log a network event
        , READEVENT                                     /* Read current raw event queue (used by EVL only)
```

```
        ) equals 28 increment 1  prefix NFB tag $C;

                                                        /* Resume defining function codes
                                                        /*  (leave room for 3 obsolete function codes)
    constant(
          FC_DELETE                                     /* Remove an entry from the data base.
        , FC_SHOW                                       /* Return specified field values.
        , FC_SET                                        /* Set/modify the field values.
        , FC_CLEAR                                      /* Clear specified field values.
        , FC_ZERCOU                                     /* Zero (and optionally read) counters
        , FC_LOOP                                       /* Loop (used only to PSI to loop an X.25 line)
        ) equals 33 increment 1  prefix NFB tag $C;     /* Maximum FCT value

    constant FC_MAX     equals NFB$C_FC_LOOP  prefix NFB tag $C;/* Maximum FCT value

    FLAGS_OVERLAY union fill;
        FLAGS byte unsigned;                            /* Miscellaneous control flags
        FLAGS_BITS structure fill;
            ERRUPD bitfield mask;                       /* Update position context, even on error
            MULT bitfield mask;                         /* Process as many entries as can be fit into P4
            NOCTX bitfield mask;                        /* Don't update position context, even if successful
                                                        /*  (used to stay on an entry for a while).  This
                                                        /*  flag overrides the ERRUPD flag.
        end FLAGS_BITS;

    end FLAGS_OVERLAY;
    DATABASE byte unsigned;                             /* A code identifying the database as follows:
                                                        /* ZERO is an illegal value for this field
    constant(
          LNI                                           /* Local node
        , NDI                                           /* Common nodes
        , OBI                                           /* Network objects
        , CRI                                           /* Circuits
        , PLI                                           /* Lines
        , EFI                                           /* Event logging filters
        , ESI                                           /* Event logging sinks
        , LLI                                           /* Logical-links
        , XNI                                           /* X.25 networks
        , XGI                                           /* X.25 groups
        , XDI                                           /* X.25 DTEs
        , XS5                                           /* X.25 server
        , XD5                                           /* X.25 destinations
        , XS9                                           /* X.29 server
        , XD9                                           /* X.29 destinations
        , XTI                                           /* X.25 trace facility
        , XTT                                           /* X.25 tracepoints
        , SPI                                           /* Server Process
        , AJI                                           /* Adjacency information
        , ARI                                           /* Area information
                                                        /* (The following codes are reserved for future PSACP
                                                        /*  databases.  These codes should only be used in the
                                                        /*  event PSIACP needs a database code before a new
                                                        /*  new NETACP can be supplied to support it).
        , PSI1                                          /* PSI reserved database
        , PSI2                                          /* PSI reserved database
        , PSI3                                          /* PSI reserved database
```

```
        , PSI4                                          /* PSI reserved database
        , PSI5                                          /* PSI reserved database
        , SDI                                           /* Service (DLE) information
        , XAI                                           /* X.25 access database
        , XXX                                           /* Last database definition for NFB$C_DB_MAX calc.
        ) equals 1 increment 1  prefix NFB$C_ tag DB;   /* Maximum DATABASE value

    constant MAX        equals NFB$C_DB_XXX-1  prefix NFB$C_ tag DB;/* Maximum DATABASE value
    OPER byte unsigned;                                 /* Specifies the sense of the search (e.g. EQL, GEQU)
                                                        /* when comparing against the SRCH_KEY field.
    constant(
          EQL                                           /* Match if SEARCH_KEY value EQL database entry field
        , GTRU                                          /* Match if SEARCH_KEY value GTRU database entry field
        , LSSU                                          /* Match if SEARCH_KEY value LSSU database entry field
        , NEQ                                           /* Match if SEARCH_KEY value NEQ database entry field
                                                        /* The following may only be used internally by NETACP
        , FNDMIN                                        /* Find entry with minimum key value
        , FNDMAX                                        /* Find entry with maximum key value
        , FNDPOS                                        /* Find entry position in database

        ) equals 0 increment 1  prefix NFB$C_ tag OP;   /* Maximum operator function

    constant MAXFCT     equals NFB$C_OP_NEQ  prefix NFB$C_ tag OP;/* Maximum operator function
    constant MAXINT     equals NFB$C_OP_FNDPOS  prefix NFB$C_ tag OP;/* Maximum internal function

    SRCH_KEY longword unsigned;                         /* Search key field identifier specifying the key used
                                                        /* to locate the entry in the database.  This search is
                                                        /* controlled by the sense of the NFB$B_OPER field.
                                                        /*
                                                        /* If this field has the value "NFB$C_WILDCARD", then
                                                        /* the very next entry in the list is assumed to be the
                                                        /* target of the search.
                                                        /*
                                                        /* If this field is not specified (zero), then it
                                                        /* is assumed to be NFB$C_WILDCARD (no search key).
                                                        /*
SRCH2_KEY longword unsigned;                            /* Secondary search key field ID specifying the key used
                                                        /* to locate the entry in the database.  This search is
                                                        /* controlled by the sense of the NFB$B_OPER2 field.
                                                        /*
                                                        /* If both SRCH_KEY and SRCH2_KEY are specified, then
                                                        /* only those database entries matching both search keys
                                                        /* will be processed.
                                                        /*
                                                        /* If this field is not specified (zero), then it
                                                        /* is assumed to be NFB$C_WILDCARD (no search key).
                                                        /*
OPER2 byte unsigned;                                    /* Specifies the sense of the search (e.g. EQL, GEQU)
                                                        /* when comparing against the SRCH2_KEY field.
MBZ1 byte unsigned;                                     /* Reserved. MBZ.
CELL_SIZE word unsigned;                                /* Some of the field values found in the P4 buffer are
constant "LENGTH" equals . prefix NFB$ tag K;           /* Minimum structure size.
constant "LENGTH" equals . prefix NFB$ tag C;           /* Minimum structure size.
                                                        /* counted strings.  If the "cell size" is non-zero, it
                                                        /* indicates the number of bytes which each string in
                                                        /* the P4 buffer occupies.  If it is zero then strings
```

```
    FLDID longword unsigned;                        /* fields are stored as variable lengthed strings.
                                                    /* Cell containing the first field ID -- the list
                                                    /* of field IDs begins here and continues to the
                                                    /* end of the structure.
                                                    /*
                                                    /* The list may be terminated before the end of the
                                                    /* structure by placing the value NFB$C_ENDOFLIST
                                                    /* in the longword following the last field ID.
                                                    /*

                                                    /*
                                                    /*       Define the "field i.d." format.
                                                    /*
end NFBDEF;

aggregate NFBDEF1 union fill prefix NFB$;
    PARAM_ID longword unsigned;                     /* Define parameter ID longword
        PARAM_ID_BITS structure fill;
            INX bitfield mask length 16;            /* Index into semantic table
            TYP bitfield mask length 2;             /* Field type (string, bit, etc.)
            SPARE bitfield mask length 6;           /* Reserved, MBZ
            DB bitfield mask length 8;              /* Data-base i.d.
        end PARAM_ID_BITS;
        constant TYP_BIT        equals 0  prefix NFB tag $C;/* Field type for bits
        constant TYP_V  equals 0  prefix NFB tag $C;    /* Field type for bits
        constant TYP_LNG        equals 1  prefix NFB tag $C;/* Field type for longwords
        constant TYP_L  equals 1  prefix NFB tag $C;    /* Field type for longwords
        constant TYP_STR        equals 2  prefix NFB tag $C;/* Field type for strings
        constant TYP_S  equals 2  prefix NFB tag $C;    /* Field type for strings

/*
/*   Define useful symbols for storing and retreiving binary and string
/*   values from the P2 and P4 buffers
/*

end NFBDEF1;

aggregate NFBDEF2 union fill prefix NFB$;
    LNG_VALUE longword unsigned;                    /* Longword value
end NFBDEF2;

aggregate NFBDEF3 union fill prefix NFB$;
    BIT_VALUE longword unsigned;                    /* Boolean value
end NFBDEF3;

aggregate NFBDEF4 union fill prefix NFB$;
    STR_COUNT word unsigned;                        /* String count field
        STR_COUNT_FIELDS structure fill;
            FILL_1 byte dimension 2 fill prefix NFBDEF tag $$;
            STR_TEXT character length 0 tag B;      /* Start of string data

/*
/*  Define identifiers for each parameter in all database
/*
/*    **  The low order 16 bits for each parameter must be unique  **
/*   ***  with respect to all other parameters in its particular   ***
```

```
/*    **  database.                                          **
```

```
                                                        /*
                                                        /* Define a field identifier index for each parameter in the NDI database.
                                                        /*
                                                        /*
                                                        /*   Boolean parameters
                                                        /*
        constant(
              LCK                                       /* Set if conditionally writable fields are not writable
            , LOO                                       /* Set if CNF is for a "loopback" node
            , REA                                       /* Set if node is reachable
            ) equals (((NFB$C_DB_NDI@24)+(NFB$C_TYP_BIT@16)+1)) increment 1  prefix NFB$C_N tag DI;

                                                        /*
                                                        /*   "Longword" Parameters
                                                        /*
        constant(
              TAD                                       /* "transformed address" - uses local node address
                                                        /* for the local NDI (instead of zero as does ADD)
            , CTA                                       /* Absolute due time for logging counters
            , ADD                                       /* Address
            , CTI                                       /* Counter timer
            , ACL                                       /* Active links
            , DEL                                       /* Delay
            , DTY                                       /* Destination Type
            , DCO                                       /* Destination Cost
            , DHO                                       /* Destination Hops
            , SDV                                       /* Service Device
            , CPU                                       /* CPU type
            , STY                                       /* Software type
            , DAD                                       /* Dump address
            , DCT                                       /* Dump count
            , OHO                                       /* Host
            , IHO                                       /* Host
            , ACC                                       /* Access switch (inbound, outbound, etc)
            , PRX                                       /* ** obsolete ** (Node proxy parameter)
            , NND                                       /* Next node address
            , SNV                                       /* Service Node Version
            ) equals (((NFB$C_DB_NDI@24)+(NFB$C_TYP_LNG@16)+16)) increment 1  prefix NFB$C_N tag DI;

                                                        /*
                                                        /*   String parameters
                                                        /*
        constant(
              COL                                       /* Collating field
            , HAC                                       /* Node address/loop linename combination
            , CNT                                       /* Counters
            , NNA                                       /* Name
            , SLI                                       /* Service line
            , SPA                                       /* Service password
            , LOA                                       /* Load file
            , SLO                                       /* Secondary loader
            , TLO                                       /* Tertiary loader
            , SID                                       /* Software ID
            , DUM                                       /* Dump file
```

```
                , SDU                                   /* Secondary dumper
                , NLI                                   /* Loopback Line
                , DLI                                   /* Destination Line
                , PUS                                   /* Privileged user id
                , PAC                                   /* Privileged account
                , PPW                                   /* Privileged password
                , NUS                                   /* Non-privileged user id
                , NAC                                   /* Non-privileged account
                , NPW                                   /* Non-privileged password
                , RPA                                   /* Receive password
                , TPA                                   /* Transmit password
                , DFL                                   /* Diagnostic load file
                , HWA                                   /* Hardware NI address (ROM address)
                , LPA                                   /* Loop assistant NI address
                , NNN                                   /* Next node name to destination (goes with NNN)
                ) equals (((NFB$C_DB_NDI@24)+(NFB$C_TYP_STR@16)+64)) increment 1  prefix NFB$C_N tag DI;
```

```
                                                /*
                                                /* Define a field identifier index for each parameter in the LNI database.
                                                /*
                                                /*
                                                /*   Boolean parameters
                                                /*
        constant(
              LCK                               /* Set if conditionally writable fields are not writable
            , SUP                               /* Set if area numbers are to be suppressed
            ) equals (((NFB$C_DB_LNI@24)+(NFB$C_TYP_BIT@16)+1)) increment 1  prefix NFB$C_L tag NI;

                                                /*
                                                /*   "Longword parameters
                                                /*
        constant(
              ADD                               /* Address
            , ACL                               /* Total number of active links
            , ITI                               /* Incoming timer
            , OTI                               /* Outgoing timer
            , STA                               /* State
            , MLK                               /* Maximum links
            , DFA                               /* Delay factor
            , DWE                               /* Delay weight
            , IAT                               /* Inactivity timer
            , RFA                               /* Retransmit factor
            , ETY                               /* Executor Type
            , RTI                               /* Routing timer
            , RSI                               /* Routing suppression timer
            , SAD                               /* Subaddress
                                                /* (lower word = lower limit, upper word = upper limit)
            , MAD                               /* Maximum address
            , MLN                               /* Maximum lines
            , MCO                               /* Maximum cost
            , MHO                               /* Maximum hops
            , MVI                               /* Maximum visits
            , MBU                               /* Maximum buffers
            , BUS                               /* Forwarding buffer size
            , LPC                               /* Loop count
            , LPL                               /* Loop length
            , LPD                               /* Loop Data type
            , DAC                               /* Default access switch (inbound, outbound, etc)
            , DPX                               /* Default proxy access (inbound, outbound, etc)
            , PIQ                               /* Pipeline quota
            , LPH                               /* Loop help type of assistance given to loop requestors
            , BRT                               /* Broadcast routing timer
            , MAR                               /* Maximum areas
            , MBE                               /* Maximum nonrouters on NI
            , MBR                               /* Maximum routers on NI
            , AMC                               /* Area maximum cost
            , AMH                               /* Area maximum hops
            , SBS                               /* Segment buffer size
            , ALI                               /* Alias local node address (cluster address)
            ) equals (((NFB$C_DB_LNI@24)+(NFB$C_TYP_LNG@16)+16)) increment 1  prefix NFB$C_L tag NI;

                                                /*
```

```
                                        /*   String parameters
                                        /*
        constant(
              COL                       /* Collating field
            , NAM                       /* Local node name
            , CNT                       /* Counters
            , IDE                       /* Identification
            , MVE                       /* Management version
            , NVE                       /* Nsp version
            , RVE                       /* Routing version
            , PHA                       /* Physical NI address (current address)
            ) equals (((NFB$C_DB_LNIa24)+(NFB$C_TYP_STRa16)+64)) increment 1  prefix NFB$C_L tag NI;
```

```
                                                        /*
                                                        /* Define a field identifier index for each parameter in the OBI database.
                                                        /*
                                                        /*
                                                        /*   Boolean Parameters
                                                        /*
        constant(
              LCK                                       /* Set if conditionally writable fields are not writable
            , SET                                       /* Set if a "set" QIO has ever modified the CNF. If
                                                        /* not then the CNF was due to a "declare name/obect"
                                                        /* only and may be deleted when the declaring process
                                                        /* breaks the channel over which the object was declared
            ) equals (((NFB$C_DB_OBI@24)+(NFB$C_TYP_BIT@16)+1)) increment 1  prefix NFB$C_O tag BI;

                                                        /*
                                                        /*   Longword Parameters
                                                        /*
        constant(
              LPR                                       /* Low order privileges
            , HPR                                       /* High order privileges
            , UCB                                       /* Owner's UCB address
            , CHN                                       /* Owner's channel
            , NUM                                       /* Number
            , PID                                       /* Process id
            , PRX                                       /* Proxy login switch (inbound, outbound, etc)
            ) equals (((NFB$C_DB_OBI@24)+(NFB$C_TYP_LNG@16)+16)) increment 1  prefix NFB$C_O tag BI;

                                                        /*
                                                        /*   String Parameters
                                                        /*
        constant(
              COL                                       /* Collating field
            , ZNA                                       /* Zero obj+name identifier
            , SFI                                       /* Parsed file i.d.
            , IAC                                       /* Default inbound combined access control string
            , NAM                                       /* Name
            , FID                                       /* File id
            , USR                                       /* User id
            , ACC                                       /* Account
            , PSW                                       /* Password
            ) equals (((NFB$C_DB_OBI@24)+(NFB$C_TYP_STR@16)+64)) increment 1  prefix NFB$C_O tag BI;
```

```
                                                /*
                                                /* Define a field identifier index for each parameter in the CRI database.
                                                /*
                                                /*
                                                /*                          /* Use
                                                /*                          /* ----
                                                /*                          C = common
                                                /*                           E = Executor (used by Transport)
                                                /*                          X = Native X.25 network management
                                                /*                          D = DECnet (not X.25)
                                                /*
                                                /*
                                                /*   Boolean Parameters
                                                /*
        constant(
              LCK                               /* D Set if conditionally writable fields are
                                                /*         not writable
            , SER                               /* D Set if Service functions not allowed
            , BLK                               /* E Blocking
            , VER                               /* D Transport verification requested if set
            , DLM                               /* E Circuit to be used as X.25 datalink, if set
                                                /*         If clear, circuit is for X.25 native use
            ) equals (((NFB$C_DB_CRI@24)+(NFB$C_TYP_BIT@16)+1)) increment 1  prefix NFB$C_C tag RI;

                                                /*
                                                /*   "Longword" parameters
                                                /*
        constant(
              OWPID                             /* D PID of temp owner of line in service state
            , CTA                               /* D Absolute due time for counter logging
            , SRV                               /* D Service substate qualifier
            , STA                               /* C State
            , SUB                               /* C Substate
            , LCT                               /* C Counter timer
            , PNA                               /* E Adjacent node address
            , BLO                               /* E Partner's receive block size
            , COS                               /* E Cost
            , HET                               /* E Hello timer
            , LIT                               /* E Listen timer
            , MRC                               /* E Maximum recalls
            , RCT                               /* E Recall timer
            , POL                               /* D Polling state
            , PLS                               /* D Polling substate
            , USE                               /* X Usage
            , TYP                               /* C Type
            , CHN                               /* X X.25 Channel
            , MBL                               /* X Maximum block
            , MWI                               /* X Maximum window
            , TRI                               /* D Tributary
            , BBT                               /* D Babble timer
            , TRT                               /* D Transmit timer
            , MRB                               /* D Maximum receive buffers
            , MTR                               /* D Maximum transmits
            , ACB                               /* D Active base
            , ACI                               /* D Active increment
```

```
            , IAB                                   /* D Inactive base
            , IAI                                   /* D Inactive increment
            , IAT                                   /* D Inactive threshold
            , DYB                                   /* D Dying base
            , DYI                                   /* D Dying increment
            , DYT                                   /* D Dying threshold
            , DTH                                   /* D Dead threshold
            , MST                                   /* D Maintenance mode state (0 => On, 1 => Off>
            , XPT                                   /* E Transport protocol to use
            , MRT                                   /* E Maximum routers on this NI
            , RPR                                   /* E Router priority
            , DRT                                   /* E Designated router on NI (node address)
            ) equals (((NFB$C_DB_CRI@24)+(NFB$C_TYP_LNG@16)+16)) increment 1  prefix NFB$C_C tag RI;

                                                    /*
                                                    /*   String Parameters
                                                    /*
        constant(
              COL                                   /* D Collating field
            , NAM                                   /* C Circuit name
            , VMSNAM                                /* D Device name in VMS format
            , CHR                                   /* D Characteristics buffer for startup control QIO
            , CNT                                   /* C Counters
            , P2P                                   /* D Line's PhaseII partner name (for loopback)
            , LOO                                   /* E Loopback name
            , PNN                                   /* E Adjacent node name
            , NUM                                   /* X Call Number
            , DTE                                   /* X DTE
            , DEVNAM                                /* D Device name in VMS format, with unit included
            ) equals (((NFB$C_DB_CRI@24)+(NFB$C_TYP_STR@16)+64)) increment 1  prefix NFB$C_C tag RI;
```

```
                                                /*
                                                /* Define a field identifier index for each parameter in the PLI database.
                                                /*
                                                /*                              C = common
                                                /*                              L = LAPB (X.25)
                                                /*                              D = DDCMP (not X.25)
                                                /*                              E = Ethernet
                                                /*
                                                /*                              /* Use
                                                /* ----
                                                /*
                                                /*   Boolean Parameters
                                                /*
        constant(
              LCK                               /* D Set if conditionally writable fields are
                                                /*       not writable
            , SER                               /* D Service
            , DUP                               /* C Duplex (set if half)
            , CON                               /* C Controller (set if loopback)
            , CLO                               /* C Clock mode (set if internal)
            ) equals (((NFB$C_DB_PLI@24)+(NFB$C_TYP_BIT@16)+1)) increment 1  prefix NFB$C_P tag LI;

                                                /*
                                                /*   "Longword"  Parameters
                                                /*
        constant(
              CTA                               /* D Absolute time for counter read and clear
            , STA                               /* C State
            , SUB                               /* C Substate
            , LCT                               /* D Counter timer
            , PRO                               /* C Protocol
            , STI                               /* D Service timer
            , HTI                               /* L Holdback timer
            , MBL                               /* L Maximum block
            , MRT                               /* L Maximum retransmits
            , MWI                               /* L Maximum window
            , SLT                               /* D Scheduling timer
            , DDT                               /* D Dead timer
            , DLT                               /* D Delay timer
            , SRT                               /* D Stream timer
            , BFN                               /* D Receive buffers
            , BUS                               /* D Action routine returns bufsiz used for line
            , PLVEC                             /* D PLVEC i.d.
            , RTT                               /* D Retransmit timer
            , MOD                               /* L X.25 mode (DCE, DTE, etc).
            , LPC                               /* L Loop count
            , LPL                               /* L Loop length
            , LPD                               /* L Loop Data type
            , EPT                               /* E Ethernet protocol type for datalink
            , BFS                               /* C Line buffer size (overrides executor bufsiz)
            ) equals (((NFB$C_DB_PLI@24)+(NFB$C_TYP_LNG@16)+16)) increment 1  prefix NFB$C_P tag LI;

                                                /*
                                                /*   String Parameters
                                                /*
```

```
        constant(
              COL                                       /* D Collating field
            , NAM                                       /* C Line name
            , VMSNAM                                    /* D Device name in VMS format
            , CHR                                       /* D Set-mode $QIO line Characteristics buffer
            , CNT                                       /* C Counters
            , MCD                                       /* L Filespec for microcode dump (initiates dump)
            , HWA                                       /* D NI hardware address (ROM address)
            , DEVNAM                                    /* D Device name in VMS format, with unit included
            ) equals (((NFB$C_DB_PLI@24)+(NFB$C_TYP_STR@16)+64)) increment 1  prefix NFB$C_P tag LI;
```

```
                                                        /*
                                                        /* Define a field identifier index for each parameter in the EFI database.
                                                        /*
                                                        /*
                                                        /*   Boolean Parameters
                                                        /*
        constant(
              LCK                                       /* Set if conditionally writable fields are not writable
            ) equals (((NFB$C_DB_EFI@24)+(NFB$C_TYP_BIT@16)+1)) increment 1  prefix NFB$C_E tag FI;

                                                        /*
                                                        /*   "Longword" Parameters
                                                        /*
        constant(
              SIN
            , SP1
            , B1
            , B2
            ) equals (((NFB$C_DB_EFI@24)+(NFB$C_TYP_LNG@16)+16)) increment 1  prefix NFB$C_E tag FI;

                                                        /*
                                                        /*   String Parameters
                                                        /*
        constant(
              COL                                       /* Collating field
            , EVE
            , SB1
            , SB2
            , SB3
            ) equals (((NFB$C_DB_EFI@24)+(NFB$C_TYP_STR@16)+64)) increment 1  prefix NFB$C_E tag FI;
```

```
                                                /*
                                                /* Define a field identifier index for each parameter in the ESI database.
                                                /*
                                                /*
                                                /*   Boolean Parameters
                                                /*
        constant(
              LCK                               /* Set if conditionally writable fields are not writable
            ) equals (((NFB$C_DB_ESI@24)+(NFB$C_TYP_BIT@16)+1)) increment 1  prefix NFB$C_E tag SI;

                                                /*
                                                /*   "Longword" Parameters
                                                /*
        constant(
              SNK
            , STA
            , SP1
            , B1
            , B2
            ) equals (((NFB$C_DB_ESI@24)+(NFB$C_TYP_LNG@16)+16)) increment 1  prefix NFB$C_E tag SI;

                                                /*
                                                /*   String Parameters
                                                /*
        constant(
              COL                               /* Collating field
            , LNA
            , SB1
            , SB2
            , SB3
            ) equals (((NFB$C_DB_ESI@24)+(NFB$C_TYP_STR@16)+64)) increment 1  prefix NFB$C_E tag SI;
```

```
                                                /*
                                                /* Define a field identifier index for each parameter in the LLI database.
                                                /*
                                                /*
                                                /*   Boolean Parameters
                                                /*
        constant(
              LCK                               /* Set if conditionally writable fields are not writable
              ) equals (((NFB$C_DB_LLI@24)+(NFB$C_TYP_BIT@16)+1)) increment 1  prefix NFB$C_L tag LI;

                                                /*
                                                /*   Longword Parameters
                                                /*
        constant(
              DLY                               /* Round trip delay time
            , STA                               /* State
            , LLN                               /* Local link number
            , RLN                               /* Remote link number
            , PNA                               /* Partner's node address
            , PID                               /* External Process I.D.
            , IPID                              /* Internal Process I.D.
            , XWB                               /* Pointer to XWB
            , CNT                               /* Counters
            ) equals (((NFB$C_DB_LLI@24)+(NFB$C_TYP_LNG@16)+16)) increment 1  prefix NFB$C_L tag LI;

                                                /*
                                                /*    tring Parameters
                                                /*
        constant(
              COL                               /* Collating field
            , USR                               /* User name
            , PRC                               /* Process name
            , PNN                               /* Partner's node name
            , RID                               /* Partner's process i.d.
            ) equals (((NFB$C_DB_LLI@24)+(NFB$C_TYP_STR@16)+64)) increment 1  prefix NFB$C_L tag LI;
```

```
                                                        /*
                                                        /* X.25 network parameters (part of MODULE X25-PROTOCOL)
                                                        /*
                                                        /* Define a field identifier index for each parameter in the XNI database.
                                                        /*
                                                        /*
                                                        /*   Boolean Parameters
                                                        /*
        constant(
              LCK                                       /* Set if conditionally writable fields are not writable
            , MNS                                       /* X.25 multi-network support (set if enabled)
            ) equals (((NFB$C_DB_XNI@24)+(NFB$C_TYP_BIT@16)+1)) increment 1  prefix NFB$C_X tag NI;

                                                        /*
                                                        /*   "Longword" Parameters
                                                        /*
        constant(
              CAT                                       /* Call timer
            , CLT                                       /* Clear timer
            , DBL                                       /* Default data
            , DWI                                       /* Default window
            , MBL                                       /* Maximum data
            , MCL                                       /* Maximum clears
            , MRS                                       /* Maximum resets
            , MST                                       /* Maximum restarts
            , MWI                                       /* Maximum window
            , RST                                       /* Reset timer
            , STT                                       /* Restart timer
            ) equals (((NFB$C_DB_XNI@24)+(NFB$C_TYP_LNG@16)+16)) increment 1  prefix NFB$C_X tag NI;

                                                        /*
                                                        /*   String Parameters
                                                        /*
        constant(
              COL                                       /* Collating field
            , NET                                       /* Network
            ) equals (((NFB$C_DB_XNI@24)+(NFB$C_TYP_STR@16)+64)) increment 1  prefix NFB$C_X tag NI;
```

```
                                                /*
                                                /* X.25 DTE parameters (qualified by a given network)
                                                /*
                                                /* Define a field identifier index for each parameter in the XDI database.
                                                /*
                                                /*
                                                /*   Boolean Parameters
                                                /*
        constant(
              LCK                               /* Set if conditionally writable fields are not writable
            ) equals (((NFB$C_DB_XDI@24)+(NFB$C_TYP_BIT@16)+1)) increment 1  prefix NFB$C_X tag DI;

                                                /*
                                                /*   "Longword" Parameters
                                                /*
        constant(
              ACH                               /* Active channels
            , ASW                               /* Active switched
            , CTM                               /* Counter timer
            , MCH                               /* Maximum channels
            , STA                               /* State
            , SUB                               /* Substate
            , MCI                               /* Maximum circuits [VMS only]
            ) equals (((NFB$C_DB_XDI@24)+(NFB$C_TYP_LNG@16)+16)) increment 1  prefix NFB$C_X tag DI;

                                                /*
                                                /*   String Parameters
                                                /*
        constant(
              COL                               /* Collating field
            , DTE                               /* DTE address
            , CHN                               /* Channels
            , LIN                               /* Line
            , NET                               /* Network
            , CNT                               /* Counters
            ) equals (((NFB$C_DB_XDI@24)+(NFB$C_TYP_STR@16)+64)) increment 1  prefix NFB$C_X tag DI;
```

```
                                                /*
                                                /* X.25 group parameters (qualified by a given DTE)
                                                /*
                                                /* Define a field identifier index for each parameter in the XGI database.
                                                /*
                                                /*
                                                /*   Boolean Parameters
                                                /*
        constant(
            LCK                                 /* Set if conditionally writable fields are not writable
            ) equals (((NFB$C_DB_XGI@24)+(NFB$C_TYP_BIT@16)+1)) increment 1  prefix NFB$C_X tag GI;

                                                /*
                                                /*   "Longword" Parameters
                                                /*
        constant(
            GNM                                 /* Group number
            , GTY                               /* Group type
            ) equals (((NFB$C_DB_XGI@24)+(NFB$C_TYP_LNG@16)+16)) increment 1  prefix NFB$C_X tag GI;

                                                /*
                                                /*   String Parameters
                                                /*
        constant(
            COL                                 /* Collating field. This field must be unique across
                                                /* all entries in this database.  It consists of the
                                                /* group-name string followed by the DTE address.
            , GRP                               /* Group name
            , GDT                               /* Group DTE address
            ) equals (((NFB$C_DB_XGI@24)+(NFB$C_TYP_STR@16)+64)) increment 1  prefix NFB$C_X tag GI;
```

```
                                                /*
                                                /* X.25 server parameters (global parameters for all destinations)
                                                /*
                                                /* Define a field identifier index for each parameter in the XS5 database.
                                                /*
                                                /*
                                                /*   Boolean Parameters
                                                /*
        constant(
              LCK                               /* Set if conditionally writable fields are not writable
            ) equals (((NFB$C_DB_XS5@24)+(NFB$C_TYP_BIT@16)+1)) increment 1  prefix NFB$C_X tag S5;

                                                /*
                                                /*   "Longword" Parameters
                                                /*
        constant(
              MCI                               /* Maximum circuits allowed
            , STA                               /* State
            , ACI                               /* Active circuits
            , CTM                               /* Counter timer
            ) equals (((NFB$C_DB_XS5@24)+(NFB$C_TYP_LNG@16)+16)) increment 1  prefix NFB$C_X tag S5;

                                                /*
                                                /*   String Parameters
                                                /*
        constant(
              COL                               /* Collating field. This field must be unique across
                                                /* all entries in this database.
            , CNT                               /* Counters
            ) equals (((NFB$C_DB_XS5@24)+(NFB$C_TYP_STR@16)+64)) increment 1  prefix NFB$C_X tag S5;
```

```
                                                        /*
                                                        /* X.25 destination parameters (part of MODULE X25-SERVER)
                                                        /*
                                                        /* Define a field identifier index for each parameter in the XD5 database.
                                                        /*
                                                        /*
                                                        /*   Boolean Parameters
                                                        /*
        constant(
              LCK                                       /* Set if conditionally writable fields are not writable
            ) equals (((NFB$C_DB_XD5@24)+(NFB$C_TYP_BIT@16)+1)) increment 1  prefix NFB$C_X tag D5;

                                                        /*
                                                        /*   "Longword" Parameters
                                                        /*
        constant(
              PRI                                       /* Priority
            , SAD                                       /* Subaddress range
                                                        /* (lower word = lower limit, upper word = upper limit)
            , NOD                                       /* Remote node address containing server (gateways only)
            ) equals (((NFB$C_DB_XD5@24)+(NFB$C_TYP_LNG@16)+16)) increment 1  prefix NFB$C_X tag D5;

                                                        /*
                                                        /*   String Parameters
                                                        /*
        constant(
              COL                                       /* Collating field. This field must be unique across
                                                        /* all entries in this database.
            , DST                                       /* Destination DTE address
            , CMK                                       /* Call mask
            , CVL                                       /* Call value
            , GRP                                       /* Group name
            , NUM                                       /* DTE number
            , OBJ                                       /* && Object name
            , FIL                                       /* Command procedure to execute when starting object
            , USR                                       /* User name
            , PSW                                       /* Password
            , ACC                                       /* Account
            ) equals (((NFB$C_DB_XD5@24)+(NFB$C_TYP_STR@16)+64)) increment 1  prefix NFB$C_X tag D5;
```

```
                                                /*
                                                /* X.29 server parameters (global parameters for all destinations)
                                                /*
                                                /* Define a field identifier index for each parameter in the XS9 database.
                                                /*
                                                /*
                                                /*   Boolean Parameters
                                                /*
        constant(
              LCK                               /* Set if conditionally writable fields are not writable
            ) equals (((NFB$C_DB_XS9@24)+(NFB$C_TYP_BIT@16)+1)) increment 1  prefix NFB$C_X tag S9;

                                                /*
                                                /*   "Longword" Parameters
                                                /*
        constant(
              MCI                               /* Maximum circuits allowed
            , STA                               /* State
            , ACI                               /* Active circuits
            , CTM                               /* Counter timer
            ) equals (((NFB$C_DB_XS9@24)+(NFB$C_TYP_LNG@16)+16)) increment 1  prefix NFB$C_X tag S9;

                                                /*
                                                /*   String Parameters
                                                /*
        constant(
              COL                               /* Collating field. This field must be unique across
                                                /* all entries in this database.
            , CNT                               /* Counters
            ) equals (((NFB$C_DB_XS9@24)+(NFB$C_TYP_STR@16)+64)) increment 1  prefix NFB$C_X tag S9;
```

```
                                                /*
                                                /* X.29 destination parameters (part of MODULE X29-SERVER)
                                                /*
                                                /* Define a field identifier index for each parameter in the XD9 database.
                                                /*
                                                /*
                                                /*   Boolean Parameters
                                                /*
        constant(
              LCK                               /* Set if conditionally writable fields are not writable
            ) equals (((NFB$C_DB_XD9@24)+(NFB$C_TYP_BIT@16)+1)) increment 1  prefix NFB$C_X tag D9;

                                                /*
                                                /*   "Longword" Parameters
                                                /*
        constant(
              PRI                               /* Priority
            , SAD                               /* Subaddress range
                                                /* (lower word = lower limit, upper word = upper limit)
            , NOD                               /* Remote node address containing server (gateways only)
            ) equals (((NFB$C_DB_XD9@24)+(NFB$C_TYP_LNG@16)+16)) increment 1  prefix NFB$C_X tag D9;

                                                /*
                                                /*   String Parameters
                                                /*
        constant(
              COL                               /* Collating field. This field must be unique across
                                                /* all entries in this database.
            , DST                               /* Destination DTE address
            , CMK                               /* Call mask
            , CVL                               /* Call value
            , GRP                               /* Group name
            , NUM                               /* DTE number
            , OBJ                               /* && Object name
            , FIL                               /* Command procedure to execute when starting object
            , USR                               /* User name
            , PSW                               /* Password
            , ACC                               /* Account
            ) equals (((NFB$C_DB_XD9@24)+(NFB$C_TYP_STR@16)+64)) increment 1  prefix NFB$C_X tag D9;
```

```
                                                        /*
                                                        /* X.25 tracing facility (global) parameters.
                                                        /*
                                                        /* Define a field identifier index for each parameter in the XTI database.
                                                        /*
                                                        /*
                                                        /*   Boolean Parameters
                                                        /*
        constant(
              LCK                                       /* Set if conditionally writable fields are not writable
            ) equals (((NFB$C_DB_XTI@24)+(NFB$C_TYP_BIT@16)+1)) increment 1  prefix NFB$C_X tag TI;

                                                        /*
                                                        /*   "Longword" Parameters
                                                        /*
        constant(
              STA                                       /* State
            , BFZ                                       /* Buffer size
            , CPL                                       /* Capture limit
            , MBK                                       /* Maximum blocks/file
            , MBF                                       /* Maximum number of buffers
            , MVR                                       /* Maximum trace file version number
            ) equals (((NFB$C_DB_XTI@24)+(NFB$C_TYP_LNG@16)+16)) increment 1  prefix NFB$C_X tag TI;

                                                        /*
                                                        /*   String Parameters
                                                        /*
        constant(
              COL                                       /* Collating field. This field must be unique across
                                                        /* all entries in this database.
            , FNM                                       /* Trace file name
            ) equals (((NFB$C_DB_XTI@24)+(NFB$C_TYP_STR@16)+64)) increment 1  prefix NFB$C_X tag TI;
```

```
                                                /*
                                                /* X.25 tracpoint (local) parameters.
                                                /*
                                                /* Define a field identifier index for each parameter in the XTT database.
                                                /*
                                                /*
                                                /*   Boolean Parameters
                                                /*
        constant(
              LCK                               /* Set if conditionally writable fields are not writable
            ) equals (((NFB$C_DB_XTT@24)+(NFB$C_TYP_BIT@16)+1)) increment 1  prefix NFB$C_X tag TT;

                                                /*
                                                /*   "Longword" Parameters
                                                /*
        constant(
              TST                               /* State
            , CPS                               /* Capture size
            ) equals (((NFB$C_DB_XTT@24)+(NFB$C_TYP_LNG@16)+16)) increment 1  prefix NFB$C_X tag TT;

                                                /*
                                                /*   String Parameters
                                                /*
        constant(
              COL                               /* Collating field. This field must be unique across
                                                /* all entries in this database.
            , TPT                               /* Tracepoint name
            ) equals (((NFB$C_DB_XTT@24)+(NFB$C_TYP_STR@16)+64)) increment 1  prefix NFB$C_X tag TT;
```

```
                                                /*
                                                /* X.25 Access (qualified by a given network)
                                                /*
                                                /* Define a field identifier index for each parameter in the XAI database.
                                                /*
                                                /*
                                                /*   Boolean Parameters
                                                /*
        constant(
              LCK                               /* Set if conditionally writable fields are not writable
            ) equals (((NFB$C_DB_XAI@24)+(NFB$C_TYP_BIT@16)+1)) increment 1  prefix NFB$C_X tag AI;

                                                /*
                                                /*   "Longword" Parameters
                                                /*
        constant(
              NDA                               /* Node address
            ) equals (((NFB$C_DB_XAI@24)+(NFB$C_TYP_LNG@16)+16)) increment 1  prefix NFB$C_X tag AI;

                                                /*
                                                /*   String Parameters
                                                /*
        constant(
              COL                               /* Collating field
            , NET                               /* Network
            , USR                               /* User id
            , PSW                               /* Password
            , ACC                               /* Account
            , NOD                               /* Node id
            ) equals (((NFB$C_DB_XAI@24)+(NFB$C_TYP_STR@16)+64)) increment 1  prefix NFB$C_X tag AI;
```

```
                                                /*
                                                /* Define SPI (Server Process) parameters
                                                /*
                                                /*
                                                /*   Boolean Parameters
                                                /*
        constant(
              LCK                               /* Set if conditionally writable fields are not writable
            , PRL                               /* Proxy flag which initially started server process
            ) equals (((NFB$C_DB_SPI@24)+(NFB$C_TYP_BIT@16)+1)) increment 1  prefix NFB$C_S tag PI;

                                                /*
                                                /*   Longword Parameters
                                                /*
        constant(
              PID                               /* Server PID
            , IRP                               /* IRP of waiting DECLSERV QIO (0 if process active)
            , CHN                               /* Channel associated with DECLSERV IRP
            , RNA                               /* Remote node address which initially started server
            ) equals (((NFB$C_DB_SPI@24)+(NFB$C_TYP_LNG@16)+16)) increment 1  prefix NFB$C_S tag PI;

                                                /*
                                                /*   String Parameters
                                                /*
        constant(
              COL                               /* Collating field
            , ACS                               /* ACS used to initally start server process
            , RID                               /* Remote user ID which initially started server
            , SFI                               /* Last (current) SFI given to server process
            , NCB                               /* Last (current) NCB given to server process
            , PNM                               /* Last (current) process name given to server
            ) equals (((NFB$C_DB_SPI@24)+(NFB$C_TYP_STR@16)+64)) increment 1  prefix NFB$C_S tag PI;
```

```
                                                /*
                                                /* Define AJI (Adjacency) parameters
                                                /*
                                                /*
                                                /*   Boolean Parameters
                                                /*
        constant(
              LCK                               /* Set if conditionally writable fields are not writable
            , REA                               /* Reachable (set if two-way communication established)
            ) equals (((' FB$C_DB_AJI@24)+(NFB$C_TYP_BIT@16)+1)) increment 1  prefix NFB$C_A tag JI;

                                                /*
                                                /*   Longword Parameters
                                                /*
        constant(
              ADD                               /* Node address
            , TYP                               /* Node type
            , LIT                               /* Lister timer for this adjacency
            , BLO                               /* Partner's block size
            , RPR                               /* Partner's router priority (on NI)
            ) equals (((NFB$C_DB_AJI@24)+(NFB$C_TYP_LNG@16)+16)) increment 1  prefix NFB$C_A tag JI;

                                                /*
                                                /*   String Parameters
                                                /*
        constant(
              COL                               /* Collating field
            , NNA                               /* Node name
            , CIR                               /* Circuit name
            ) equals (((NFB$C_DB_AJI@24)+(NFB$C_TYP_STR@16)+64)) increment 1  prefix NFB$C_A tag JI;
```

```
                                                /*
                                                /* Define SDI (Service DLE) parameters
                                                /*
                                                /*
                                                /*   Boolean Parameters
                                                /*
        constant(
              LCK                               /* Set if conditionally writable fields are not writable
            ) equals (((NFB$C_DB_SDI@24)+(NFB$C_TYP_BIT@16)+1)) increment 1  prefix NFB$C_S tag DI;

                                                /*
                                                /*   Longword Parameters
                                                /*
        constant(
              SUB                               /* Service substate
            , PID                               /* PID of process owning this DLE link
            ) equals (((NFB$C_DB_SDI@24)+(NFB$C_TYP_LNG@16)+16)) increment 1  prefix NFB$C_S tag DI;

                                                /*
                                                /*   String Parameters
                                                /*
        constant(
              COL                               /* Collating field
            , CIR                               /* Circuit name
            , PHA                               /* Service physical address (BC only)
            , PRC                               /* Name of process owning this DLE link
            ) equals (((NFB$C_DB_SDI@24)+(NFB$C_TYP_STR@16)+64)) increment 1  prefix NFB$C_S tag DI;
```

```
                                                /*
                                                /* Define the AREA database (read only) for level 2 Phase IV routers only.
                                                /*
                                                /*
                                                /*   Boolean parameters
                                                /*
        constant(
              LCK                               /* Set if conditionally writable fields are not writable
            , REA                               /* Set if node is reachable
            ) equals (((NFB$C_DB_ARIa24)+(NFB$C_TYP_BITa16)+1)) increment 1  prefix NFB$C_A tag RI;

                                                /*
                                                /*   "Longword" Parameters
                                                /*
        constant(
              ADD                               /* Address
            , DCO                               /* Destination Cost
            , DHO                               /* Destination Hops
            , NND                               /* Next node address
            ) equals (((NFB$C_DB_ARIa24)+(NFB$C_TYP_LNGa16)+16)) increment 1  prefix NFB$C_A tag RI;

                                                /*
                                                /*   String parameters
                                                /*
        constant(
              COL                               /* Collating field
            , DLI                               /* Circuit used for normal traffic to area
            ) equals (((NFB$C_DB_ARIa24)+(NFB$C_TYP_STRa16)+64)) increment 1  prefix NFB$C_A tag RI;

    end STR_COUNT_FIELDS;
end NFBDEF4;

end_module $NFBDEF;

module $DRDEF;
```

```
/*
/* DISCONNECT REASONS
/*

constant DR_NORMAL      equals 0  prefix NET tag $C;    /* NO ERROR (SYNCH DISCONNECT)
constant DR_RSU equals 1  prefix NET tag $C;            /* COULDN'T ALLOCATE UCB ADDRESS
constant DR_NONODE      equals 2  prefix NET tag $C;    /* Unrecognized node name
constant DR_SHUT        equals 3  prefix NET tag $C;    /* NODE OR LINE SHUTTING DOWN
constant DR_NOBJ        equals 4  prefix NET tag $C;    /* UNKNOWN OBJECT TYPE OR PROCESS
constant DR_FMT equals 5  prefix NET tag $C;            /* ILLEGAL PROCESS NAME FIELD
constant DR_BUSY        equals 6  prefix NET tag $C;    /* Object too busy
constant DR_PROTCL      equals 7  prefix NET tag $C;    /* GENERAL PROTOCOL ERROR
constant DR_THIRD       equals 8  prefix NET tag $C;    /* THIRD PARTY DISCONNECT
constant DR_ABORT       equals 9  prefix NET tag $C;    /* DISCONNECT ABORT
constant DR_IVNODE      equals 2  prefix NET tag $C;    /* Invalid node name format
constant DR_NONZ        equals 21  prefix NET tag $C;   /* NON-ZERO DST ADDRESS
constant DR_BADLNK      equals 22  prefix NET tag $C;   /* INCONSISTENT DSTLNK
constant DR_ZERO        equals 23  prefix NET tag $C;   /* ZERO SOURCE ADDRESS
constant DR_BADFC       equals 24  prefix NET tag $C;   /* FCVAL ILLEGAL
constant DR_NOCON       equals 32  prefix NET tag $C;   /* NO CONNECT SLOTS AVAILABLE
constant DR_ACCESS      equals 34  prefix NET tag $C;   /* INVALID ACCESS CONTROL
constant DR_BADSRV      equals 35  prefix NET tag $C;   /* LOGICAL LINK SERVICES MISMATCH
constant DR_ACCNT       equals 36  prefix NET tag $C;   /* INVALID ACCOUNT INFORMATION
constant DR_SEGSIZ      equals 37  prefix NET tag $C;   /* SEGSIZE TOO SMALL
constant DR_EXIT        equals 38  prefix NET tag $C;   /* USER EXIT OR TIMEOUT
constant DR_NOPATH      equals 39  prefix NET tag $C;   /* NO PATH TO DESTINATION NODE
constant DR_LOSS        equals 40  prefix NET tag $C;   /* LOSS OF DATA HAS OCCURRED
constant DR_NOLINK      equals 41  prefix NET tag $C;   /* ILLEGAL MSG FOR LINK NOLINK STATE
constant DR_CONF        equals 42  prefix NET tag $C;   /* REAL DISCONNECT CONFIRM
constant DR_IMLONG      equals 43  prefix NET tag $C;   /* IMAGE DATA FIELD TOO LONG

end_module $DRDEF;
```

NETUSR
SDL

LIBTAIL
B32

XWBDEF
SDL

NETDEFS
MAR

PSIUSR
SDL

NETDRVMAC
MAR

NODRIVER
LIS

NSPMSGDEF
SDL

LIBHEAD
B32

NET
LIS

NETMACROS
MAR

NETACPTRN
LIS