```
MMM         MMM     AAAAAAAAA     NNN         NNN     AAAAAAAAA        GGGGGGGGGGGG  EEEEEEEEEEEEEEE
MMM         MMM     AAAAAAAAA     NNN         NNN     AAAAAAAAA        GGGGGGGGGGGG  EEEEEEEEEEEEEEE
MMM         MMM     AAAAAAAAA     NNN         NNN     AAAAAAAAA        GGGGGGGGGGGG  EEEEEEEEEEEEEEE
MMMMMM   MMMMMM  AAA         AAA  NNN         NNN  AAA         AAA  GGG              EEE
MMMMMM   MMMMMM  AAA         AAA  NNN         NNN  AAA         AAA  GGG              EEE
MMMMMM   MMMMMM  AAA         AAA  NNN         NNN  AAA         AAA  GGG              EEE
MMM   MMM   MMM  AAA         AAA  NNNNNN      NNN  AAA         AAA  GGG              EEE
MMM   MMM   MMM  AAA         AAA  NNNNNN      NNN  AAA         AAA  GGG              EEE
MMM   MMM   MMM  AAA         AAA  NNNNNN      NNN  AAA         AAA  GGG              EEE
MMM         MMM  AAA         AAA  NNN   NNN   NNN  AAA         AAA  GGG              EEEEEEEEEEEE
MMM         MMM  AAA         AAA  NNN   NNN   NNN  AAA         AAA  GGG              EEEEEEEEEEEE
MMM         MMM  AAA         AAA  NNN   NNN   NNN  AAA         AAA  GGG              EEEEEEEEEEEE
MMM         MMM  AAAAAAAAAAAAAAA  NNN      NNNNNN  AAAAAAAAAAAAAAA  GGG   GGGGGGGGG  EEE
MMM         MMM  AAAAAAAAAAAAAAA  NNN      NNNNNN  AAAAAAAAAAAAAAA  GGG   GGGGGGGGG  EEE
MMM         MMM  AAAAAAAAAAAAAAA  NNN      NNNNNN  AAAAAAAAAAAAAAA  GGG   GGGGGGGGG  EEE
MMM         MMM  AAA         AAA  NNN         NNN  AAA         AAA  GGG         GGG  EEE
MMM         MMM  AAA         AAA  NNN         NNN  AAA         AAA  GGG         GGG  EEE
MMM         MMM  AAA         AAA  NNN         NNN  AAA         AAA  GGG         GGG  EEE
MMM         MMM  AAA         AAA  NNN         NNN  AAA         AAA     GGGGGGGGG     EEEEEEEEEEEEEEE
MMM         MMM  AAA         AAA  NNN         NNN  AAA         AAA     GGGGGGGGG     EEEEEEEEEEEEEEE
MMM         MMM  AAA         AAA  NNN         NNN  AAA         AAA     GGGGGGGGG     EEEEEEEEEEEEEEE
```

FILEID**NETCONFIG

```
NN      NN  EEEEEEEEEE  TTTTTTTTTT    CCCCCCCC    OOOOOO    NN      NN  FFFFFFFFFF    IIIIII      GGGGGGGG
NN      NN  EEEEEEEEEE  TTTTTTTTTT    CCCCCCCC    OOOOOO    NN      NN  FFFFFFFFFF    IIIIII      GGGGGGGG
NN      NN  EE              TT      CC          OO      OO  NN      NN  FF              II      GG
NN      NN  EE              TT      CC          OO      OO  NN      NN  FF              II      GG
NNNN    NN  EE              TT      CC          OO      OO  NNNN    NN  FF              II      GG
NNNN    NN  EE              TT      CC          OO      OO  NNNN    NN  FF              II      GG
NN  NN  NN  EEEEEEEE        TT      CC          OO      OO  NN  NN  NN  FFFFFFFF        II      GG
NN  NN  NN  EEEEEEEE        TT      CC          OO      OO  NN  NN  NN  FFFFFFFF        II      GG
NN    NNNN  EE              TT      CC          OO      OO  NN    NNNN  FF              II      GG  GGGGGG
NN    NNNN  EE              TT      CC          OO      OO  NN    NNNN  FF              II      GG  GGGGGG
NN      NN  EE              TT      CC          OO      OO  NN      NN  FF              II      GG      GG      ....
NN      NN  EE              TT      CC          OO      OO  NN      NN  FF              II      GG      GG      ....
NN      NN  EEEEEEEEEE      TT        CCCCCCCC    OOOOOO    NN      NN  FF            IIIIII      GGGGGG        ....
NN      NN  EEEEEEEEEE      TT        CCCCCCCC    OOOOOO    NN      NN  FF            IIIIII      GGGGGG        ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
$!      IDENT V03-008
$!
$!++
$! Facility:    NETCONFIG, Automatic DECnet Database Configurator
$!
$! Abstract:    This procedure uses parameter information and the network I/O
$!               device environment in which it is run to automatically
$!               configure a DECnet permanent database.
$!
$! Environment: Executed from the system manager's account on a VAX/VMS
$!               capable of running DECnet.
$!
$! Parameters:  P1 - DECnet node name for this node, 1 to 6 alphabetic
$!                      characters (prompt for if omitted)
$!               P2 - DECnet node number, 0 to 1023 (prompt for if omitted)
$!               P3 - Do you want a default DECnet account? [YES]
$!               P4 - Do you want to run as a router? [NO]
$!               P5 - Do you want to startup DECnet? [NO]
$!
$! Author:      Tim Halvorsen
$! Created:     June 1982
$!
$! Modifications:
$!
$!      V03-008 PRB0342         Paul Beck       22-JUL-1984 21:39
$!               Fix problem with comment emerging as actual command.
$!
$!      V03-007 PRB0339         Paul Beck       27-JUN-1984 20:05
$!               Offer to configure CI if PA device is present.
$!               Offer advice about CI configuration when done.
$!               Issue warning if no datalinks were defined.
$!
$!      V03-006 PRB0333         Paul Beck       26-APR-1984 15:45
$!               Remove DISUSER flag from default DECnet account, and
$!               add /NOINTERACTIVE /NOBATCH qualifiers.
$!               Fix bad address checking and prompt defaulting for areas.
$!               Autoconfigure devices if minimum boot was done.
$!
$!      V03-005 WHM0001         Bill Matthews   16-Apr-1984
$!               Replaced reference to SCSNODEL/H with SCSNODE.
$!
$!      V03-004 PRB0315         Paul Beck       2-MAR-1984 12:32
$!               /PAGINATE changed to /PAGE
$!               Fix calculation of MAX ADDRESS to omit area portion of address
$!               Align prompts
$!
$!      V03-003 PRB0308         Paul Beck       4-FEB-1984 18:09
$!               Use TYPE/PAGE to print out list of commands.
$!               Adjust maximum address if needed.
$!               Use node name from SYSGEN as default for node name.
$!
$!      V03-002 TMH0002         Tim Halvorsen           05-Oct-1983
$!               Add option to setup as router/endnode and to start DECnet
$!               (without having to exit and run STARTNET).
$!               Purge desired node name from the remote node database
$!               before defining the executor.
$!               Add QNA to the list of known devices.
$!               Put temporary files in SYS$MANAGER rather than SYS$SYSTEM.
$!
$!      V03-001 ROW0194         Ralph O. Weber          12-JUL-1983
$!               Add module header.  Setup use of P1 through P3 parameters.
$!
$ REQUIRED_PRIVILEGES = "SYSPRV,OPER"
```

```
$ PREV_PRIVS = F$SETPRV(REQUIRED_PRIVILEGES)
$ IF .NOT. F$PRIVILEGE(REQUIRED_PRIVILEGES) THEN GOTO NO_PRIVILEGES
$ ON ERROR THEN GOTO NO_CHANGES
$ ON CONTROL_Y THEN GOTO NO_CHANGES
$ NO_PROMPT = P1 .NES. "" .AND. P2 .NES. ""
$ IF NO_PROMPT THEN GOTO SKIP_IDENTIFICATION
$ TYPE SYS$INPUT

        DECnet-VAX network configuration procedure

This procedure will help you define the parameters needed to get DECnet
running on this machine.  You will be shown the changes before they are
executed, in case you wish to perform them manually.

$
$SKIP_IDENTIFICATION:
$ SMALL_DISK_SYSTEM = (F$GETDVI("SYS$SYSDEVICE","MAXBLOCK") .LT. 53000 -
                .OR. F$LOG("LIB$SYSDEVICE") .NES. "")
$ IF   F$SEARCH(F$PARSE("NCP","SYS$SYSTEM:.EXE")) .EQS. "" -
  .OR. F$SEARCH(F$PARSE("NMLSHR","SYS$LIBRARY:.EXE")) .EQS. "" -
  .OR. F$SEARCH(F$PARSE("AUTHORIZE","SYS$SYSTEM:.EXE")) .EQS. "" THEN -
        GOTO MISSING_TOOLS
$ IF F$SEARCH("SYS$MANAGER:SHOWDEV.TMP") .NES. "" THEN -
        DELETE /NOLOG SYS$MANAGER:SHOWDEV.TMP;*          ! Delete old versions
$ OPEN/WRITE NETCONFIG$FILE SYS$MANAGER:NETCONFIG.TMP
$ WRITE NETCONFIG$FILE "$ RUN SYS$SYSTEM:NCP"
$ !
$ !     Purge out any existing database
$ !
$ IF F$SEARCH("SYS$SYSTEM:NETNODE.DAT") .NES. "" THEN -
        WRITE NETCONFIG$FILE "     PURGE EXECUTOR ALL"
$ IF F$SEARCH("SYS$SYSTEM:NETLINE.DAT") .NES. "" THEN -
        WRITE NETCONFIG$FILE "     PURGE KNOWN LINES ALL"
$ IF F$SEARCH("SYS$SYSTEM:NETCIRC.DAT") .NES. "" THEN -
        WRITE NETCONFIG$FILE "     PURGE KNOWN CIRCUITS ALL"
$ IF F$SEARCH("SYS$SYSTEM:NETLOGING.DAT") .NES. "" THEN -
        WRITE NETCONFIG$FILE "     PURGE KNOWN LOGGING ALL"
$ !
$ !     Always issue the following PURGEs, solely to create a
$ !     null database (so that NML doesn't complain about
$ !     the database not existing during STARTNET).
$ !
$ WRITE NETCONFIG$FILE "     PURGE KNOWN OBJECTS ALL"
$ WRITE NETCONFIG$FILE "     PURGE MODULE CONFIGURATOR KNOWN CIRCUITS ALL"
$ !
$ !     Prompt for executor address and name
$ !
$ NAME = P1
$ GOTO TEST_NAME
$WRONG_NAME:
$ WRITE SYS$OUTPUT "Your DECnet node name should match your VAXcluster node name."
$ PROMPT = F$FAO("!60AS","Want to reenter your DECnet node name? ")
$ INQUIRE/NOPUNC Q """PROMPT"""
$ IF Q THEN GOTO PROMPT_NAME
$ GOTO TEST_NAME
$ILLEGAL_NAME:
$ WRITE SYS$OUTPUT "Illegal node name - maximum of 6 characters"
$PROMPT_NAME:
$ DEFAULT_NAME := 'F$GETSYI("SCSNODE")
$ DEFAULT_PROMPT = "[" + DEFAULT_NAME + "]"
$ IF DEFAULT_PROMPT .EQS. "[]" THEN DEFAULT_PROMPT = ""
$ L = 60 - F$LENGTH(DEFAULT_PROMPT)
$ PROMPT = F$FAO("!""L'AS!AS",-
```

```
                    "What do you want your DECnet node name to be?", DEFAULT_PROMPT)
$ INQUIRE NAME "'PROMPT'"
$ IF NAME .EQS. "" THEN NAME = DEFAULT_NAME
$ IF F$GETSYI("CLUSTER_MEMBER") .AND. NAME .NES. DEFAULT_NAME THEN GOTO WRONG_NAME
$TEST_NAME:
$ IF NAME .EQS. "" THEN GOTO PROMPT_NAME
$ IF F$LENGTH(NAME) .GT. 6 THEN GOTO ILLEGAL_NAME
$ ADDRESS = P2
$ DEFAULT_PROMPT = ""
$ DEFAULT_ADDRESS = ""
$ IF F$GETSYI("CLUSTER_MEMBER") .AND. F$GETSYI("SCSSYSTEMID") .NE. 0 -
        THEN DEFAULT_ADDRESS = F$GETSYI("SCSSYSTEMID")
$ IF DEFAULT_ADDRESS .EQS. "" THEN GOTO TEST_ADDRESS
$ DEFAULT_AREA = DEFAULT_ADDRESS/1024
$ DEFAULT_LOC = DEFAULT_ADDRESS - DEFAULT_AREA*1024
$ IF DEFAULT_AREA .EQ. 0 THEN DEFAULT_AREA = 1
$ DEFAULT_ADDRESS = "'DEFAULT_AREA'."'DEFAULT_LOC'"
$ DEFAULT_PROMPT = "[" + DEFAULT_ADDRESS + "]"
$ GOTO TEST_ADDRESS
$WRONG_ADDRESS:
$ WRITE SYS$OUTPUT "Your DECnet node address should match your VAXcluster node address."
$ PROMPT = F$FAO("!60AS","Want to reenter your DECnet node address? ")
$ INQUIRE/NOPUNC Q "'PROMPT'"
$ IF Q THEN GOTO PROMPT_ADDRESS
$ GOTO GOT_ADDRESS
$BAD_ADDR:
$ WRITE SYS$OUTPUT "That is not a legal DECnet node address."
$ WRITE SYS$OUTPUT "Legal node address range is 1.1 - 63.1023"
$PROMPT_ADDRESS:
$ L = 60 - F$LENGTH(DEFAULT_PROMPT)
$ PROMPT = F$FAO("!"L'AS!AS",-
                    "What do you want your DECnet address to be?", DEFAULT_PROMPT)
$ INQUIRE ADDRESS "'PROMPT'"
$ IF ADDRESS .EQS. "" THEN ADDRESS = DEFAULT_ADDRESS
$TEST_ADDRESS:
$ IF ADDRESS .EQS. "" THEN GOTO PROMPT_ADDRESS
$ LOC_ADDR = ADDRESS
$ FULL_ADDR = ADDRESS
$ IF F$LOCATE(".",ADDRESS) .GE. F$LENGTH(ADDRESS) THEN GOTO GOT_LOC
$ TMP = F$EXTRACT(0,F$LOCATE(".",ADDRESS),ADDRESS)
$ LOC_ADDR = ADDRESS - TMP - "."
$ LOC_AREA = TMP - LOC_ADDR - "."
$ FULL_ADDR = LOC_AREA*1024 + LOC_ADDR
$ IF LOC_AREA .GT. 63 THEN GOTO BAD_ADDR
$ IF LOC_AREA .LT. 1 THEN GOTO BAD_ADDR
$GOT_LOC:
$ IF LOC_ADDR .GT. 1023 THEN GOTO BAD_ADDR
$ IF LOC_ADDR .LT. 1 THEN GOTO BAD_ADDR
$ IF F$GETSYI("CLUSTER_MEMBER") .AND. FULL_ADDR .NE. F$GETSYI("SCSSYSTEMID") -
        THEN GOTO WRONG_ADDRESS
$GOT_ADDRESS:
$
$ WRITE NETCONFIG$FILE "$ DEFINE/USER SYS$OUTPUT NL:"
$ WRITE NETCONFIG$FILE "$ DEFINE/USER SYS$ERROR NL:"
$ WRITE NETCONFIG$FILE "$ RUN SYS$SYSTEM:NCP     ! Remove existing entry, if any"
$ WRITE NETCONFIG$FILE "     PURGE NODE ",ADDRESS," ALL"
$ WRITE NETCONFIG$FILE "     PURGE NODE ",NAME," ALL"
$ WRITE NETCONFIG$FILE "$ RUN SYS$SYSTEM:NCP"
$ WRITE NETCONFIG$FILE "     DEFINE EXECUTOR ADDRESS ",ADDRESS," STATE ON"
$ WRITE NETCONFIG$FILE "     DEFINE EXECUTOR NAME ",NAME
$ MAX_ADDR = 255
$ IF 'LOC_ADDR' .GT. MAX_ADDR THEN MAX_ADDR = 'LOC_ADDR'
$ WRITE NETCONFIG$FILE "     DEFINE EXECUTOR MAXIMUM ADDRESS ''MAX_ADDR'"
```

```
$ !
$ !     See if we should be router or endnode (endnode preferred)
$ !
$ EXEC_TYPE = "NONROUTING IV"
$ REPLY = P4
$ IF REPLY .NES. "" THEN GOTO TEST_EXEC_TYPE
$ IF NO_PROMPT THEN GOTO SET_EXEC_TYPE
$ INQUIRE REPLY -
        "Do you want to operate as a router?        [NO (nonrouting)]"
$ IF REPLY .EQS. "" THEN REPLY = "NO"
$TEST_EXEC_TYPE:
$ IF REPLY THEN EXEC_TYPE = "ROUTING IV"
$SET_EXEC_TYPE:
$ WRITE NETCONFIG$FILE "    DEFINE EXECUTOR TYPE ",EXEC_TYP
$ !
$ !     See if default DECnet account is desired
$ !
$ IF NO_PROMPT THEN GOTO MAKE_DEF_ACCOUNT
$ REPLY = P3
$ IF REPLY .NES. "" THEN GOTO TEST_DEFACC
$ INQUIRE REPLY -
        "Do you want a default DECnet acc[illegible]                    [YES]"
$ IF REPLY .EQS. "" THEN REPLY = "YES"
$TEST_DEFACC:
$ IF .NOT. REPLY THEN GOTO NO_DEFACC
$MAKE_DEF_ACCOUNT:
$ WRITE NETCONFIG$FILE "    DE[illegible]NE E[illegible]ONPRIVILEGED USER DECNET"
$ WRITE NETCONFIG$FILE "    DEFI[illegible] NONPRIVILEGED PASSWORD DECNET"
$ WRITE NETCONFIG$FILE "$ DEFINE/US[illegible]R [illegible] SYS$SYSTEM:SYSUAF.DAT"
$ WRITE NETCONFIG$FILE "$ RUN SYS$SYSTE[illegible]UTHORIZE
$ WRITE NETCONFIG$FILE "    ADD DECNET /OWNER=""DECNET DEFAULT"" -"
$ WRITE NETCONFIG$FILE "        /PASSWORD=DECNET -"
$ WRITE NETCONFIG$FILE "        /UIC=[376,376] /ACCOUNT=DECNET -"
$ WRITE NETCONFIG$FILE "        /DEVICE=SYS$SYSDEVICE: /DIRECTORY=[DECNET] -"
$ WRITE NETCONFIG$FILE "        /PRIVILEGE=(TMPMBX,NETMBX) -"
$ WRITE NETCONFIG$FILE "        /FLAGS=(CAPTIVE) /LGICMD=NL: -"
$ WRITE NETCONFIG$FILE "        /NOBATCH /NOINTERACTIVE "
$ IF .NOT. SMALL_DISK_SYSTEM THEN -  ! If RL02 system, inhibit network .LOGs
        WRITE NETCONFIG$FILE "$ CREATE/DIRECTORY SYS$SYSDEVICE:[DECNET] /OWNER=[376,376]"
$ IF SMALL_DISK_SYSTEM THEN WRITE SYS$OUTPUT - ! and print out a message
        "This is a small disk system.  As a result, the default DECnet account"
$ IF SMALL_DISK_SYSTEM THEN WRITE SYS$OUTPUT -
        "will be setup so that .LOG files will not be created for network jobs."
$ WRITE NETCONFIG$FILE "$ RUN SYS$SYSTEM:NCP
$NO_DEFACC:
$ !
$ !     Get various communications devices (by doing a SHOW DEVICE)
$ !
$ STARTUP_P1 := F$GETSYI("STARTUP_P1")
$ TEMP = F$LOCATE("MIN",STARTUP_P1)
$ IF TEMP .EQ. F$LENGTH(STARTUP_P1) THEN GOTO CONFIGURED
$ IF TEMP .NE. 0 THEN GOTO CONFIGURED
$ WRITE SYS$OUTPUT "Minimal system was booted."
$ INQUIRE Q -
        "Do you want to perform SYSGEN AUTOCONFIGURE ALL?       [YES]"
$ IF Q .EQS. "" THEN Q = "Y"
$ IF .NOT. Q THEN GOTO CONFIGURED
$ RUN SYS$SYSTEM:SYSGEN
AUTOCONFIGURE ALL
EXIT
$CONFIGURED:
$ FOUND_DATALINK = 0
$ FOUND_PA = 0
```

```
$ FOUND_CI = 0
$ DEFINE/USER SYS$OUTPUT SYS$MANAGER:SHOWDEV.TMP
$ SHOW DEVICES                                  ! Write SHOW DEV info to a file
$ OPEN/READ NETCONFIG$SHOWDEV SYS$MANAGER:SHOWDEV.TMP -
        /ERROR=NO_MORE_DEVICES
$ READ NETCONFIG$SHOWDEV LINE                   ! Read header lines
$ READ NETCONFIG$SHOWDEV LINE
$ READ NETCONFIG$SHOWDEV LINE
$READ_NEXT_DEVICE:
$ READ NETCONFIG$SHOWDEV LINE -                 ! Read next line in file
        /END_OF_FILE=NO_MORE_DEVICES
$30:
$ IF F$EXTRACT(0,1,LINE) .NES. " " THEN GOTO 40 ! Skip leading blanks
$ LINE = F$EXTRACT(1,999,LINE)
$ GOTO 30
$40:
$ IF F$LOCATE(":",LINE) .EQ. F$LENGTH(LINE) THEN GOTO READ_NEXT_DEVICE ! Skip spurious lines
$ DEVNAME = F$EXTRACT(0,F$LOCATE(":",LINE),LINE) ! Get device name from line
$ DEVUNIT = F$GETDVI(DEVNAME,"UNIT")
$ DEVTEMP = (DEVNAME + ":") - (F$STRING(DEVUNIT) + ":")
$ IF F$LOCATE("$",DEVTEMP) .LT. F$LENGTH(DEVTEMP) THEN -
        DEVTEMP = F$EXTRACT(F$LOCATE("$",DEVTEMP)+1,999,DEVTEMP)
$ DEVTYPE = F$EXTRACT(0,F$LENGTH(DEVTEMP)-1,DEVTEMP)
$ DEVCONT = F$STRING(F$CVUI(0,8,F$EXTRACT(F$LENGTH(DEVTEMP)-1,1,DEVTEMP))-F$CVUI(0,8,"A"))
$ IF DEVTYPE .EQS. "XM" THEN GOTO DMR
$ IF DEVTYPE .EQS. "XD" THEN GOTO DMP
$ IF DEVTYPE .EQS. "XG" THEN GOTO DMF
$ IF DEVTYPE .EQS. "XE" THEN GOTO UNA
$ IF DEVTYPE .EQS. "XQ" THEN GOTO QNA
$ IF DEVTYPE .EQS. "XX" THEN GOTO UE
$ IF DEVTYPE .EQS. "CN" THEN GOTO CI
$ IF DEVTYPE .EQS. "PA" THEN GOTO PA
$ GOTO READ_NEXT_DEVICE
$DMR:
$ DATALINK = "DMC-" + DEVCONT
$ COST = 5
$ GOTO POINT_TO_POINT
$DMP:
$ DATALINK = "DMP-" + DEVCONT
$ COST = 5
$ GOTO POINT_TO_POINT
$DMF:
$ DATALINK = "DMF-" + DEVCONT
$ COST = 8
$ GOTO POINT_TO_POINT
$UNA:
$ IF DEVUNIT .NE. 0 THEN GOTO READ_NEXT_DEVICE  ! Ignore cloned UCBs
$ DATALINK = "UNA-" + DEVCONT
$ COST = 3
$ GOTO POINT_TO_POINT
$QNA:
$ IF DEVUNIT .NE. 0 THEN GOTO READ_NEXT_DEVICE  ! Ignore cloned UCBs
$ DATALINK = "QNA-" + DEVCONT
$ COST = 4
$ GOTO POINT_TO_POINT
$UE:
$ IF DEVUNIT .NE. 0 THEN GOTO READ_NEXT_DEVICE  ! Ignore cloned UCBs
$ DATALINK = "UE-" + DEVCONT
$ COST = 4
$POINT_TO_POINT:
$ WRITE NETCONFIG$FILE "    DEFINE LINE    ",DATALINK," STATE ON"
$ WRITE NETCONFIG$FILE "    DEFINE CIRCUIT ",DATALINK," STATE ON COST ",COST
$ FOUND_DATALINK = 1
```

```
$ GOTO READ_NEXT_DEVICE
$PA:
$ FOUND_PA = 1
$ GOTO READ_NEXT_DEVICE
$CI:
$ DATALINK = "CI-0"
$ WRITE NETCONFIG$FILE "    DEFINE LINE    ",DATALINK," STATE ON"
$ FOUND_DATALINK = 1
$ FOUND_CI = 1
$ GOTO READ_NEXT_DEVICE
$
$NO_MORE_DEVICES:
$ CLOSE NETCONFIG$SHOWDEV
$ DELETE /NOLOG SYS$MANAGER:SHOWDEV.TMP;*
$ !
$ !     See if we need to interrogate for CNDRIVER.
$ !
$ IF FOUND_PA .EQ. 0 THEN GOTO NO_CI
$ IF FOUND_CI .EQ. 1 THEN GOTO NO_CI
$ INQUIRE Q -
        "Do you want to use the CI as a DECnet datalink?        [NO]"
$ IF Q .EQS. "" THEN Q = "N"
$ IF .NOT. Q THEN GOTO NO_CI
$ DATALINK = "CI-0"
$ WRITE NETCONFIG$FILE "    DEFINE LINE    ",DATALINK," STATE ON"
$ FOUND_DATALINK = 1
$ FOUND_PA = 2
$ FOUND_CI = 1
$NO_CI:
$ !
$ !     Setup default logging database
$ !
$ WRITE NETCONFIG$FILE "    DEFINE LOGGING MONITOR STATE ON"
$ WRITE NETCONFIG$FILE "    DEFINE LOGGING MONITOR EVENTS 0.0-9"
$ WRITE NETCONFIG$FILE "    DEFINE LOGGING MONITOR EVENTS 2.0-1"
$ WRITE NETCONFIG$FILE "    DEFINE LOGGING MONITOR EVENTS 4.2-13,15-16,18-19"
$ WRITE NETCONFIG$FILE "    DEFINE LOGGING MONITOR EVENTS 5.0-18"
$ WRITE NETCONFIG$FILE "    DEFINE LOGGING MONITOR EVENTS 128.0-4"
$ CLOSE NETCONFIG$FILE
$ !
$ !     Ask user if he wants to do it
$ !
$ IF NO_PROMPT THEN GOTO DO_DECNET_SETUP
$ TYPE SYS$INPUT

        Here are the commands necessary to setup your system.

-----------------------------------------------------------------------
$ TYPE/PAGE SYS$MANAGER:NETCONFIG.TMP
$ TYPE SYS$INPUT
-----------------------------------------------------------------------

$ IF FOUND_DATALINK .EQ. 0 THEN -
        WRITE SYS$OUTPUT "WARNING: no lines have been defined."
$ INQUIRE REPLY "Do you want to go ahead and do it? [YES]"
$ IF REPLY .EQS. "" THEN REPLY = "YES"
$ IF .NOT. REPLY THEN GOTO NO_CHANGES
$DO_DECNET_SETUP:
$ @SYS$MANAGER:NETCONFIG.TMP
$ IF NO_PROMPT THEN GOTO EXIT
$ TYPE SYS$INPUT

The changes have been made.
```

If you have not already installed the DECnet-VAX license, then do so now.

After the license has been installed, you should invoke the procedure SYS$MANAGER:STARTNET.COM to startup DECnet-VAX with these changes.

```
$ REPLY = P5
$ IF REPLY .NES. "" THEN GOTO TEST_STARTNET
$ INQUIRE REPLY -
        "(If the license is already installed) Do you want DECnet started? [YES]"
$ IF REPLY .EQS. "" THEN REPLY = "YES"
$TEST_STARTNET:
$ IF .NOT. REPLY THEN GOTO NO_STARTNET
$ !
$ !     Connect CNDRIVER if it hasn't been done already.
$ !
$ IF FOUND_PA .NE. 2 THEN GOTO START_NETWORK
$ RUN SYS$SYSTEM:SYSGEN
CONNECT CNA0 /NOADAPT /DRIVER=SYS$SYSTEM:CNDRIVER.EXE
$START_NETWORK:
$ @SYS$MANAGER:STARTNET
$NO_STARTNET:
$ IF FOUND_CI .EQ. 0 THEN GOTO EXIT
$ TYPE SYS$INPUT

You will have to define a circuit for each VAXcluster node to which you
will be making DECnet connections using the CI. Do this by specifying each
node's CI port address as a tributary number. For example, to create circuits
to the nodes having CI ports 1 and 2, enter the following commands:

        NCP> DEFINE CIRCUIT CI-0.1 STATE ON TRIBUTARY 1
        NCP> DEFINE CIRCUIT CI-0.2 STATE ON TRIBUTARY 2

Also, be sure you have added the commands

        RUN SYS$SYSTEM:SYSGEN
        SYSGEN> CONNECT CNA0 /NOADAPT /DRIVER=SYS$SYSTEM:CNDRIVER.EXE

to your site-specific startup command procedure prior to calling STARTNET.COM.

$ GOTO EXIT
$
$NO_CHANGES:
$ WRITE SYS$OUTPUT "No changes have been made."
$EXIT:
$ IF F$LOGICAL("NETCONFIG$SHOWDEV") .NES. "" THEN -
        CLOSE NETCONFIG$SHOWDEV
$ IF F$SEARCH("SYS$MANAGER:SHOWDEV.TMP") .NES. "" THEN -
        DELETE /NOLOG SYS$MANAGER:SHOWDEV.TMP;*
$ IF F$LOGICAL("NETCONFIG$FILE") .NES. "" THEN -
        CLOSE NETCONFIG$FILE
$ IF F$SEARCH("SYS$MANAGER:NETCONFIG.TMP") .NES. "" THEN -
        DELETE /NOLOG SYS$MANAGER:NETCONFIG.TMP;*
$ PREV_PRIVS = F$SETPRV(PREV_PRIVS)
$ EXIT
$
$NO_PRIVILEGES:
$ WRITE SYS$OUTPUT "Insufficient privileges to run this procedure."
$ WRITE SYS$OUTPUT "Requires ",REQUIRED_PRIVILEGES," privileges."
$ GOTO NO_CHANGES
$
$MISSING_TOOLS:
$ WRITE SYS$OUTPUT "The utilities required for this procedure cannot be found."
```

```
$ WRITE SYS$OUTPUT "NCP, AUTHORIZE and SYS$SHARE:NML$HR are required."
$ IF SMALL_DISK_SYSTEM THEN WRITE SYS$OUTPUT -
  "You should tailor in the DECNET and MANAGER groups to use this procedure."
$ GOTO NO_CHANGES
```

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

AUTOGEN
LIS

NETCONFIG
LIS

STARTUP
LIS

VMSINSTAL
MEM

SWAPFILES
LIS

TRANSERR
LIS

MAKEROOT
LIS

SHUTDOWN
LIS

SPKITBLD
LIS

VMSINSTAL
LIS

VMSUPDATE
LIS