# DEC Network Integration Server

# Configuration and Loading

Order Number: AA-Q61CD-TE

**Revision/Update Information:** This is a revised manual

**Software Version:** DECNIS™ V4.0

**First Printing, March 1994**
**Revised, October 1996**



This document was prepared using VAX DOCUMENT, Version 2.1.

# Contents

## 4 Using the DECNIS Text-Based Configurator

## 5 Creating the Configuration Files

## 8 Nonvolatile (Flash) Memory Loading



## 9 Using the DECNIS as a Proxy Load Host

## 10 Using the DECNIS as a BOOTP Gateway

## Part III Appendixes

## A Information You Need for Configuration

## B Configuration and Load Files



## C Using DECdns and the Local Namespace with the DECNIS



## D Loading from a DECnet–VAX Phase IV Load Host

## Index

## Figures



## Tables

# Preface

This manual explains how to:

- Configure the DEC™ Network Integration Server (DECNIS) software, using the DECNIS text-based configurator.
- Load the DECNIS hardware unit.
- Copy configuration files between load hosts.

This manual does not describe how to use the clearVISN™ DECNIS configurator. Refer to the manual *clearVISN DECNIS Configurator User Guide* for information about using the clearVISN DECNIS configurator.

## Audience

This manual is intended for network managers. It assumes that you understand and have some experience of:

- Local Area Networks (LANs)
- Wide Area Networks (WANs)
- DECnet™/OSI® routing
- IP routing (if using the Internet protocols)
- X.25 (if using the CCITT X.25 protocols)
- OpenVMS™ (if using an OpenVMS load host)
- Digital™ UNIX® (if using a Digital UNIX load host). (Digital UNIX was formerly called DEC OSF/1.)
- IBM®-compatible Personal Computers running MS–DOS® (PCs) (if using an MS–DOS PC load host)

## Associated Documentation

### Product Documentation

- *DEC Network Integration Server Configuration and Management from PCs*
- *DEC Network Integration Server Installation and Configuration for OpenVMS and Digital UNIX*
- *DEC Network Integration Server Introduction and Glossary*
- *DEC Network Integration Server Management*
- *clearVISN™ DECNIS Configurator User Guide*

  This manual describes how to install and configure DECNIS software on a Windows NT™ or Windows 95® PC.
- *DEC Network Integration Server Problem Solving*

  This is only available on line, as follows:

  – On DECnet/OSI systems, in Bookreader™ format.

  – On Windows 95/NT PCs, as a Windows® help file.

  – On MS–DOS PCs (non-Windows), as a series of text files.
- *DEC Network Integration Server Event Messages* (supplied on line)
- *DEC Network Integration Server Release Notes* (supplied on line)

### Hardware Documentation

The following documents are supplied with the DECNIS hardware:

- *Installation and Service Manual*
- *Configuration* card

The following documents are supplied with each Network Interface Card:

- *Cabling Instructions and Specifications* card
- *Problem Solving* card
- *Configuration* card

### Related Documentation

- Network management documentation for the load-host operating system you are using.
- *Common Trace Facility (CTF) Use* manual

This manual is part of the OpenVMS documentation set, and describes how to use the Common Trace Facility for problem solving.

- *Network Information* (supplied on line)

  This supplies profile information about all the public Packet Switching Data Networks that Digital supports.

- *X.25 Security* manual

  This manual explains the underlying concepts of X.25 security. You can order this manual through your local Digital office.

- *Bridge and Extended LAN Reference* manual

  This manual provides a general description of bridging and extended LANs. You can order this manual through your local Digital office.

- RFCs (for IP routing)

  RFCs are the working notes for the Internet research and development community. These notes are available in a three-volume set, the *DDN Protocol Handbook*, which can be ordered from the following address:

  Network Solutions, Inc.
  Attn: InterNIC Registration Service
  505 Huntmar Park Drive
  Herndon, VA 22070, USA
  Tel. 1-800-444-4345 or 619-455-4600

## Returning Comments About this Documentation

**We would like to know what you think about the DECNIS documentation set and online help.**

If you have any comments, or suggestions, please return them in any of the following ways:

- Send an electronic mail message to the Internet address books@reo.mts.dec.com
- Send an electronic mail message to the X.400 address S=IDC BOOKS; O=digital; OU1=reo; P=digital; A=CWMail; C=gb
- Send a fax to (+44)1734 206018

## Conventions

The following conventions are used in this manual:

| | |
|---|---|
| Return | Key names are shown enclosed to indicate that you press a key on the keyboard. |
| Ctrl/*x* | This symbol indicates that you press the Ctrl key at the same time as you press another key. For example, Ctrl/C, Ctrl/Y, and so on. |
| *Italics* | This indicates variable information. |
| DECNIS | DEC Network Integration Server |
| *decnis* | This indicates that you should substitute the node name of the DECNIS. If you are using a DECdns namespace or a local namespace, enter the name registered in the namespace. |
| Prompts | The following prompts precede commands that you enter:<br>For MS–DOS: `C:\`<br>For OpenVMS: `$`<br>For Digital UNIX: #<br>For NCL: `NCL>`<br>For NCP: `NCP>` |

# 1
# Introduction

## 1.1 About This Manual

This manual describes the following:

- General information about using the DECNIS text-based configurator.
- The configuration files used by the DECNIS.
- The methods you can use to load the software and configuration files to the DECNIS.

**Note**

If you wish to configure, load and manage a DECNIS from a Windows NT or Windows 95 PC, or use the clearVISN DECNIS configurator, refer to the manual *clearVISN DECNIS Configurator User Guide* and the clearVISN DECNIS configurator online help. This manual does not provide a detailed description of the clearVISN DECNIS configurator

### 1.1.1 Prerequisite Tasks

This manual assumes you have already done the following tasks:

1. Installed the DECNIS software on a supported load host.
2. Run the load-host configurator on the load host, and set up load information for your DECNIS systems.

For more information about these tasks, consult one of the following manuals:

- *DECNIS Installation and Configuration for OpenVMS and Digital UNIX*, which covers installation, load-host configuration and starting the DECNIS text-based configurator on OpenVMS and Digital UNIX load hosts.

- *DECNIS Configuration and Management from MS–DOS PCs*, which covers installation, load-host configuration, and starting the DECNIS text-based configurator on MS–DOS PC load hosts.

## 1.2 Required Steps

In order to set up your DECNIS hardware unit as a working system, carry out the steps shown in Table 1–1.

**Table 1–1 Steps to Set Up the DECNIS**

| Step | Action | Refer to: |
|---|---|---|
| 1 | Install the DECNIS software on a load host | *DECNIS Installation and Configuration for OpenVMS and Digital UNIX, DECNIS Configuration and Management from MS–DOS PCs* or *clearVISN DECNIS Configurator User Guide* |
| 2 | Make a note of the information you will need when running the text-based configurators | Appendix A |
| | If installing on OpenVMS, Digital UNIX and MS–DOS PC hosts, do Steps 3 and 4. If installing on Windows NT/95 hosts, skip to Step 5 | |
| 3 | Run the load-host configurator on a load host, and enter the required information (for Windows NT/95 hosts go to Step 5) | *DECNIS Installation and Configuration for OpenVMS and Digital UNIX* or *DECNIS Configuration and Management from MS–DOS PCs* |
| 4 | Run the DECNIS text-based configurator on a load host, and enter the required information | Chapter 2 to Chapter 4 |
| 5 | Run the clearVISN DECNIS configurator on a load host, and enter the required information | *clearVISN DECNIS Configurator User Guide* |
| 6 | Create the DECNIS configuration files | Chapter 5 |
| 7 | Downline load the configured software onto the DECNIS hardware unit | Chapter 7 to Chapter 10 and *clearVISN DECNIS Configurator User Guide* |

**Note**

This manual does not give detailed guidance on using the DECNIS text-based configurator. However, you can obtain extensive online Help while running the DECNIS text-based configurator.

See Appendix A for a complete list of all the information you may require when running the load-host and DECNIS text-based configurators.

# Part I

## Configuring the DECNIS

This part describes how to use the DECNIS text-based configurator. It also describes how to create configuration files using the DECNIS text-based configurator and the clearVISN DECNIS configurator.

It contains the following chapters:

- Chapter 2 describes how to start the DECNIS text-based configurator on supported load hosts.
- Chapter 3 gives an overview of the sections in the DECNIS text-based configurator.
- Chapter 4 describes how to use the DECNIS text-based configurator.
- Chapter 6 describes how to modify a DECNIS configuration in the DECNIS text-based configurator.
- Chapter 5 describes how to create the configuration files.

# 2

# Starting the DECNIS Text-Based Configurator

This chapter describes how to start the DECNIS text-based configurator on PC, OpenVMS, and Digital UNIX load hosts.

On OpenVMS and Digital UNIX systems, you need to set up your terminal as described in the manual *DECNIS Installation and Configuration for OpenVMS and Digital UNIX*.

## 2.1 Before You Start

You must run the load-host configurator before you run the DECNIS text-based configurator.

## 2.2 Starting the DECNIS Text-Based Configurator: PC Load Hosts

To start the DECNIS text-based configurator, do the following:

1. Use the installation directory as the current directory.
2. Start the Router Management menu by entering the following at the command prompt:

   ```
   DECROU
   ```

3. Select `DECNIS Config` from the menu.

## 2.3 Starting the DECNIS Text-Based Configurator: OpenVMS Load Hosts

To start the DECNIS text-based configurator, follow these steps:

1. Log into any account which has OPER and SYSPRV privileges.

2. Enter the command:

```
$ @SYS$MANAGER:NIS$DECNIS_CONFIG
```

## 2.4 Starting the DECNIS Text-Based Configurator: Digital UNIX Load Hosts

To start the DECNIS text-based configurator, follow these steps:

1. Log in as a superuser.
2. Enter the command:

```
# /usr/lib/dnet/nis_decnis_config
```

# 3

# DECNIS Text-Based Configurator Sections

When you run the DECNIS text-based configurator, you go through a series of sections, each corresponding to a type of information used for configuration. Sections 3.1 to 3.21 describe the purpose of each section in the configurator. Figure 3–1 gives an overview of the sections.

## 3.1 Identify the DECNIS Node

In this section, you choose which DECNIS to configure. The screen displays a menu of the DECNIS hardware units listed in the load-host data file. These are the DECNIS systems previously set up with the load-host configurator.

In the menu, each DECNIS is designated by a load client name. This is a name entered during load-host configuration. It is used simply to identify the DECNIS for the purpose of downline loading and upline dumping.

## 3.2 Select Network Interface Cards

In this section, you indicate which type of Network Interface Card will occupy each slot on the DECNIS hardware unit.

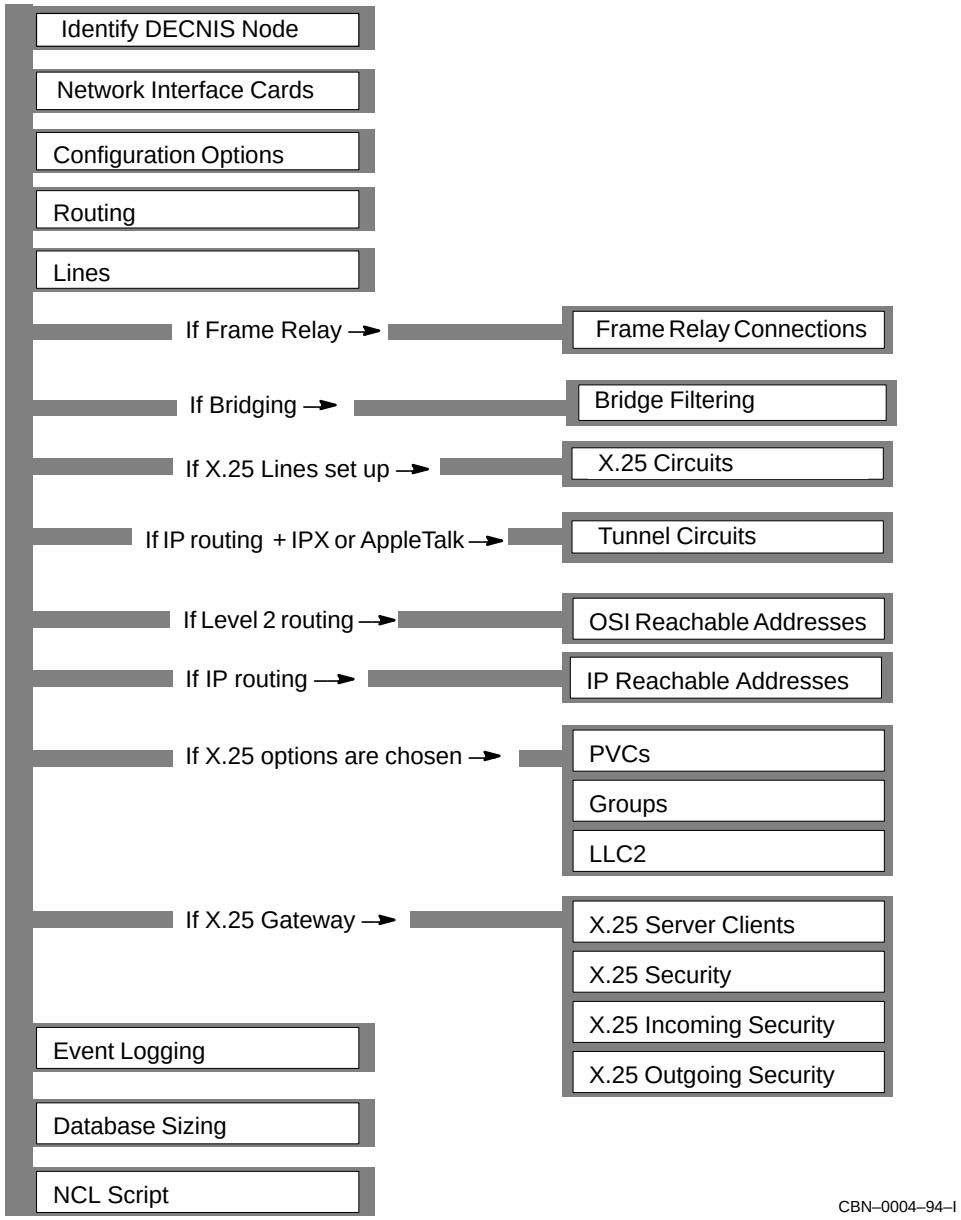
You can find the types of card supported on your DECNIS in the System Support Addendum (SSA) for your DECNIS.

## 3.3 Configuration Options

In this section, you choose whether you want to use:

- Internet Protocol (IP) routing
- Bridging
- X.25 gateway functions
- Special X.25 gateway options

**Figure 3–1 DECNIS Configurator Sections**



- NetWare® IPX routing
- AppleTalk®routing
- VCP (Vitalink® Communications Protocol)

You also supply other basic information.

## 3.4 Routing

In this section, you provide information to set up the DECNIS for OSI routing, DECnet routing and IP routing.

The information you are asked for depends on the options you have selected in the Configuration Options section. For example, you are only asked for IP information if you selected IP routing in Configuration Options.

### 3.4.1 System IP Address

If you supplied an IP address in the load-host configurator, the DECNIS will use it as a **system IP address**.

All IP packets transmitted by the DECNIS must contain a source IP address. The DECNIS uses the system IP address as the source IP address for any IP packets that do not have their own source IP address.

## 3.5 Lines

You must set up at least one line in this section. A line corresponds to a DECNIS hardware port.

The lines you configure can be used for:

- DECnet routing
- IP routing protocols:
  - Integrated IS–IS (only if the DECNIS is running the Phase V routing algorithm at one or both levels
  - RIP
  - EGP (only if the DECNIS is a Level 2 router)
  - OSPF
- NetWare IPX routing (only CSMA/CD, FDDI, VCP, PPP or CHDLC lines)
- AppleTalk routing (only CSMA/CD, FDDI or VCP lines)
- X.25 routing circuits (only synchronous lines)

- X.25 gateway switched virtual circuits (only synchronous lines)
- LLC2 communications (only CSMA/CD or FDDI lines)
- Bridging ports, on the following Network Interface Cards and lines:
  - For local bridging: CSMA/CD or FDDI lines.
  - For remote bridging on W622 cards: HDLC, PPP, CHDLC or VCP lines.
  - For remote bridging on W614 and W618 cards: HDLC, PPP or CHDLC lines.
- Frame relay channels (only on W622 cards)

## 3.6 X.25 Circuits

If you selected X.25 as the protocol for any of the lines, you are asked if you want to set up X.25 routing circuits. You can use these circuits for any or all of the following:

- DECnet routing
- IP routing
- Connecting to another OSI domain, using static addresses
- Connecting to another IP network, using static addresses

There are four types of X.25 circuit:

- Static Outgoing, to call a DTE on a DECnet router.
- Static Incoming, to receive calls from a DTE on a DECnet router.
- Permanent, to connect to a DECnet Phase IV system, using a Permanent Virtual Circuit (PVC).
- Dynamically Assigned (DA), to connect to DTEs in other OSI routing domains and/or other IP networks.

Each circuit defines a link between specified DTEs on the DECNIS and a DTE on another system.

### 3.6.1 Special Points About X.25 DA Circuits

Note that:

- You can only create X.25 DA circuits on the DECNIS if it is a Level 2 router.
- For every DA circuit to an OSI domain, you must set up an OSI reachable address (see Section 3.10).
- For every DA circuit to an IP host, subnet or network, you must set up an IP reachable address (see Section 3.11).
- You can set up DA circuits regardless of whether the DECNIS is using the link state (Phase V) or the routing vector (Phase IV) algorithm.

## 3.7 Tunnel Circuits

Tunnel circuits encapsulate NetWare IPX and/or AppleTalk packets inside IP packets. You need to set up tunnel circuits if you want the DECNIS to do either of the following:

- Forward AppleTalk or NetWare IPX packets over its synchronous lines.
- Forward NetWare IPX packets to a router that does not support NetWare IPX.
- Forward AppleTalk packets to a router that does not support AppleTalk.

You will only see the Tunnel Circuits section if you have requested the following in the Configuration Options section:

- IP Routing
- NetWare IPX and/or AppleTalk routing

## 3.8 Frame Relay Connections

In the Lines section, you can set up a line as a frame relay channel—that is, a physical connection to a frame relay network. In this section, you create frame relay connections on the frame relay channels.

Frame relay connections are virtual circuits that operate over a frame relay channel. You must create these connections in order to be able to use a channel. On the DECNIS, you can create up to 32 frame relay connections on each frame relay channel.

You can use a frame relay connection for DECnet Phase IV, OSI IP and IPX routing, and for bridging.

## 3.9 Bridge Filtering

In this section, you define the way the DECNIS does bridge filtering. You can specify that the DECNIS only forwards named protocol types, or that it only blocks named protocol types.

This is the only type of filtering you can specify in the configurator. For information about other kinds of bridge filtering, see the *DECNIS Management* manual.

## 3.10 OSI Reachable Addresses

In this section, you can specify static routes to other OSI routing domains.

You only see this section if you have specified Level 2 routing for the DECNIS.

You must set up an OSI reachable address for each X.25 Dynamically Assigned (DA) circuit with an OSI template that you have set up.

## 3.11 IP Reachable Addresses

In this section, you can specify static routes to other IP hosts, subnets or networks.

You must set up an IP reachable address for each X.25 Dynamically Assigned (DA) circuit with an IP template that you have set up.

## 3.12 PVCs

In this section, you can set up nonrouting Permanent Virtual Circuits (PVCs). A PVC is a permanent association between two specific DTEs. Two DTEs connected by a PVC can communicate without the need for call clearing or call setup.

Complete this section only if you have subscribed to this facility from a PSDN.

## 3.13 Groups

If a DTE belongs to a Closed User Group (CUG), it can communicate freely with remote DTEs that are also members of that CUG. Its communications with DTEs outside the group may be restricted, depending on your PSDN subscription options.

Complete this section only if you have subscribed to this facility from a PSDN.

## 3.14 LLC2

LLC2 is a data link protocol which enables the X.25 packet-level protocol to run over an ISO 8802-2 LAN, rather than a synchronous line. On the DECNIS, LLC2 links use CSMA/CD or FDDI ports.

Normally, you only set up the DECNIS to use LLC2 if you want the DECNIS to act as a CONS LAN/WAN relay. When acting as a CONS LAN/WAN relay, the DECNIS switches calls between LLC2 systems on the LAN and one or more PSDNs. The LLC2 systems must be capable of running the ISO 8802-2 Class II protocol over the ISO 8802-3 protocol.

If you want to use LLC2, you must set up an LLC2 DTE for each remote system you want to connect to on the LAN.

## 3.15 X.25 Server Clients

X.25 server clients identify Client systems using the DECNIS system as an X.25 gateway. You must set up X.25 server clients if you want Client systems to be able to use the DECNIS as an X.25 gateway.

### 3.15.1 Definitions of X.25 Gateway Systems and Client Systems

An X.25 gateway is a Connector system allowing Client systems to connect to a PSDN or communicate across an X.25 point-to-point link.

Examples of Client systems are VAX™ P.S.I. Access systems and DEC X.25 for ULTRIX Gateway Client systems.

You do not need to set up X.25 server clients for systems on the network that are only using LLC2 to communicate with the DECNIS.

### 3.15.2 Filters

You must set up at least one filter to associate with each X.25 server client. A filter is a mechanism for matching incoming calls to Client systems.

Each filter contains a list of characteristics corresponding to fields in an incoming call request packet. You assign values to the characteristics you want matched. If the characteristics in an incoming call match those listed in a filter, then the call is passed to the server client associated with that filter.

## 3.16 X.25 Security

In this section, you specify whether or not you want to set up X.25 security. You only see this section if you are using the X.25 gateway function.

If you choose to set up X.25 security, you will need to define X.25 security in detail for incoming and/or outgoing calls; see Section 3.17 and Section 3.18.

If you choose not to set up X.25 security, the configurator will set up open security. This means that the DECNIS will:

- Accept all incoming calls, provided that they match a filter that is in use.
- Allow all outgoing calls from any of the Client systems using the DECNIS to any remote DTE.

If you set up open security, you do not see the other security sections described below.

## 3.17 Incoming Security for X.25 Server Clients

You use incoming X.25 security to prevent unauthorized incoming calls to either or both of the following:

- X.25 server clients served by the DECNIS.
- PVCs set up on the DECNIS.

You only set up incoming security if you are using the X.25 gateway function.

To set up incoming security, you specify the remote DTEs that you expect to send calls to X.25 server client systems. You also specify the type of access: ALL, NONE or REMOTE CHARGE.

If setting up PVC security, you specify remote DTEs that are using DECNIS PVCs. In this case, the only types of access are ALL or NONE.

You are only asked to supply the remote DTEs that have ALL access; any you do not supply have NONE.

## 3.18 Outgoing Security for Client Systems

You use outgoing X.25 security to prevent unauthorized outgoing calls from either or both of the following:

- Client systems using the DECNIS.
- PVCs set up on the DECNIS.

You only set up outgoing security if you are using the X.25 gateway function.

In this section, you specify Client systems using the DECNIS. You also specify the type of access they should have to remote DTEs: ALL, NONE or REMOTE CHARGE.

If setting up PVC security, you specify local DECNIS PVCs. In this case, the types of access are ALL or NONE.

## 3.19 Event Logging

Event logging is used to monitor your system and help in problem solving. In this section, you may set up event streams and event sinks. An event stream contains events generated by the DECNIS. An event sink is a node to which event streams are sent.

The configurator produces event streams with a standard, predefined set of events. You cannot alter these event streams within the configurator. To alter the event streams, you modify the user NCL script files produced by the configurator.

## 3.20 Database Sizing

In this section, you can adjust the amount of memory resources allocated to DECNIS database components

The routing database on the DECNIS is split into a number of components that interoperate to provide routing information. Each component has a value which specifies how much memory is allocated to it.

The configurator provides a default value for each component. This section allows you to change those values.

Note that if you increase some values, you may need to reduce others, so as not to use up the total amount of memory.

## 3.21 NCL Script

When you have entered all your configuration information, the configurator creates an NCL script using this information. Refer to Chapter 5 for details.

# 4

# Using the DECNIS Text-Based Configurator

This chapter describes how to run the DECNIS text-based configurator. You should read Chapter 2 before you read this chapter.

## 4.1 Introduction and Main Menu

When you start the DECNIS text-based configurator, you will see first the copyright screen, and then a brief explanation of the configurator. Press Return. The Main Menu appears:

```
Create a new configuration

Modify an existing configuration

Exit from this procedure
```

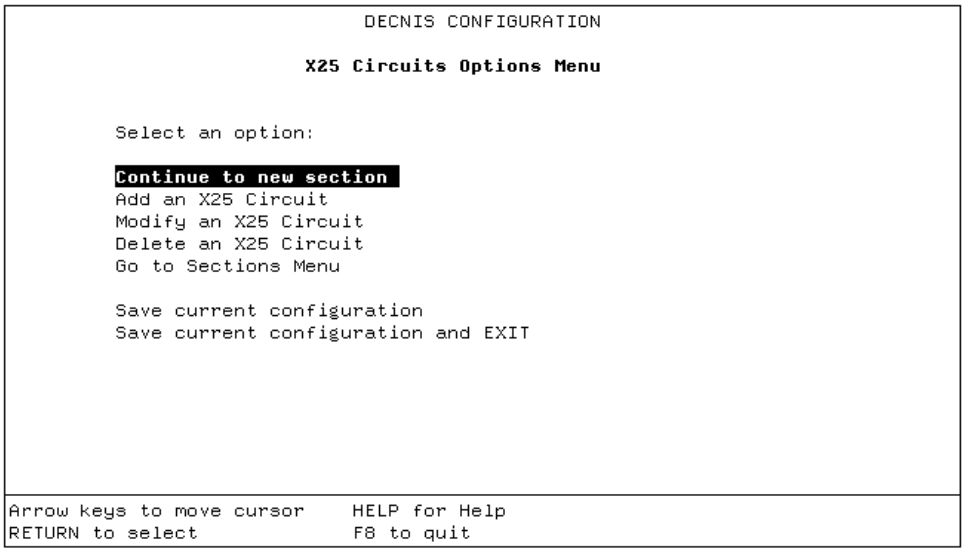
If you are configuring a DECNIS for the first time, choose `Create a new configuration`. You then see the first configurator sections.

### 4.1.1 The Sections

The DECNIS text-based configurator is divided into sections, each corresponding to a logical group of information. Each section contains a series of screens on which you enter information.

### 4.1.2 The Options Menu

When you complete the last screen in a section, the screen displays an **Options Menu**. For example:




### 4.1.3 Meaning of the Options in the Options Menu

The following list gives a brief explanation of each option in Options Menus.

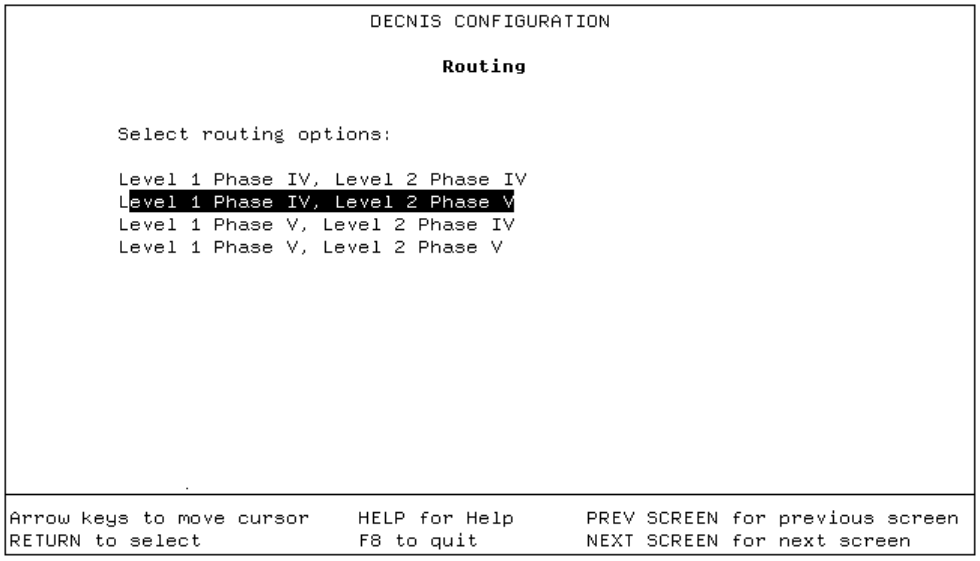
- `Continue to new section` takes you to the next uncompleted section.
- `Add` lets you set up an item. For example, an additional X.25 routing circuit.
- `Modify` lets you change information previously entered.
- `Delete` lets you delete an item set up previously.
- `Sections Menu` takes you to a menu of completed sections. If you select a section, you go to the Options Menu for that section.
- `Save the current configuration` lets you save your configuration so far.
- `Save the current configuration and EXIT` is the same as Save, except that when the configuration is saved, you exit from the configurator.

## 4.2 Entering Information

This section describes how to enter information in the DECNIS configurator.

### 4.2.1 Selecting from a Menu

On some screens, you select from a menu. For example:




1. Move the cursor to the item you want, using the up or down arrow keys.
2. Press Return.

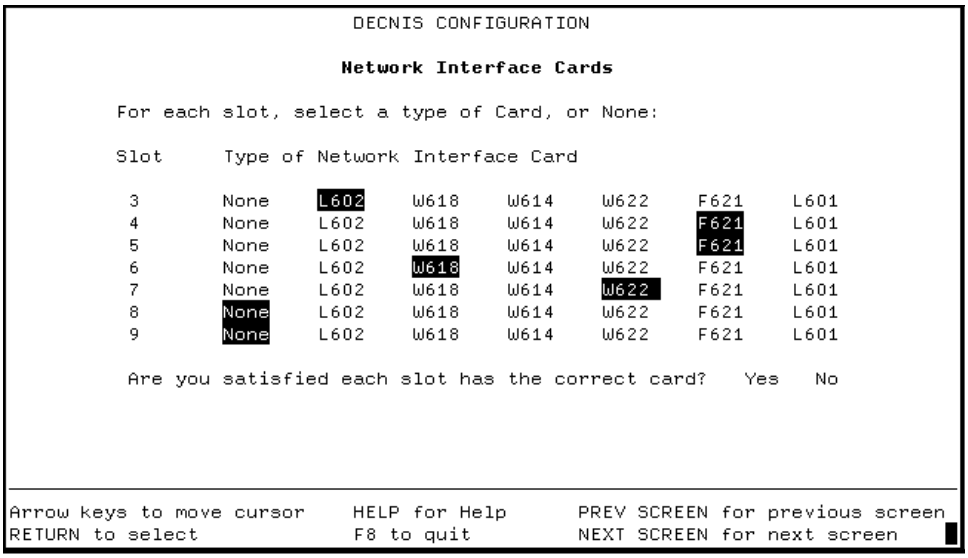
### 4.2.2 Horizontal Menus

Some screens have a horizontal menu:

1. Move the cursor to the item you want, using the left or right arrow keys.
2. Press Return.

### 4.2.2.1 Series of Horizontal Menus

A more complicated example of horizontal menus is in the Network Interface Card section. Here, you select from a series of horizontal menus, as shown in the following example screen:




On each line, you do the following:

1. Move the cursor to the card you want, using the left or right arrow key.
2. Press Return. The cursor moves to the first column on the next line.
3. Repeat for every line.

#### Changing a Selection

To change a selection on a previous line, use the up arrow key to get to the line. You can then use the left or right arrow key to move to a new selection.

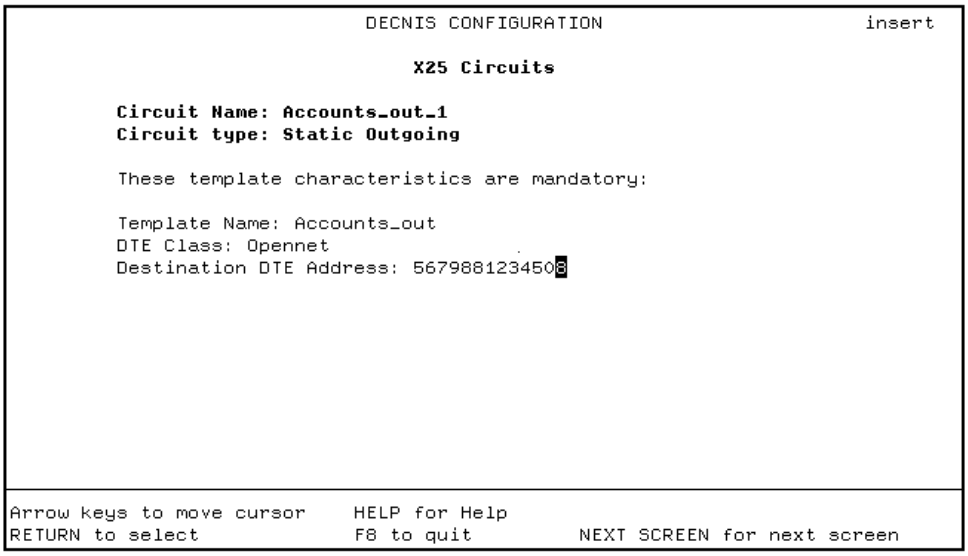
#### Restrictions on Leaving a Line

If you have moved the cursor horizontally on a line, you must press Return before you can move to another line, or move to the next screen.

For example, on line 6, you move the cursor from None to W622. You then decide to move back to line 5 to change your previous selection. You cannot do this until you have pressed Return on line 6.

### 4.2.3 Typing in Data

On some screens, you type information into a field. For example:




Type data into the field, and press Return.

#### Horizontal Scrolling

Usually, when you type in data, you can see the entire field. However, sometimes the maximum length you are allowed to type in is too long to fit into the field – for example, a node name, which may be up to 400 characters. In such cases, the field horizontally scrolls as you enter data.

Horizontal scrolling only works if the keyboard is in "Insert" mode, not if it is in "Overstrike" mode.

The words "Insert" or "Overstrike" appear in the upper righthand corner of the screen. To change between modes:

- On OpenVMS and Digital UNIX systems, press Ctrl/A
- On PCs, press Insert

## 4.3 Moving Within and Between Sections

The next sections describe how you can move within and between configurator sections.

### 4.3.1 Moving Forward Within a Section

When you have filled in the required fields on a screen, a new screen automatically appears. You cannot move forward until you have completed the required fields.

If the fields are filled in already, or are optional, you can move to the next screen by pressing:

- On OpenVMS and Digital UNIX systems, Next Screen
- On PCs, PG DN

### 4.3.2 Moving Back Within a Section

To move back within a section, do the following:

- On OpenVMS and Digital UNIX systems, press Prev Screen
- On PCs, press PG UP

You can move backwards only as far as the first screen of the section. To get to another section, select `Go to Sections Menu` from any Options Menu. Then select a section.

### 4.3.3 Moving to a New Section

From the Options Menu, choose:

`Continue to new section`

You move to the first screen of the next section you have not seen. If you have completed all the sections, you move to the final section, `Create NCL Script`.

If you have modified previously completed sections, always use this option after you have finished making your changes.

### 4.3.4 Moving to a Previous Section

You can move to any section previously completed. This includes optional sections you previously chose to skip.

From the Options Menu, choose:

`Go to Sections Menu`

You see a menu listing the completed sections. When you select a section, you go to the Options Menu for that section.

### 4.3.5 Restriction on Leaving a Section

You cannot jump to the Options Menu from the middle of a section, even if the section was previously completed. You must go through all the screens.

If the section is complete, you can move quickly through the screens, by pressing:

- On OpenVMS and Digital UNIX systems, Next Screen
- On PCs, PG DN

## 4.4 What You Can Do from the Options Menu

### 4.4.1 Adding an Item

You use `Add` to create several of the same type of item—for example, lines, X.25 circuits, reachable addresses. From the Options menu, choose:

`Add an` *`item`*

You go to the first screen on which you can enter data.

### 4.4.2 Modifying a Completed Section

You can change information previously entered. From the Options Menu, choose:

`Modify an` *`item`*

You go to the first screen on which you can enter data. All screens will display the information previously entered.

### 4.4.3 Deleting an Item

You can delete an item previously created. From the Options Menu, choose:

`Delete an` *`item`*

### 4.4.4 Using the Sections Menu

You use the Sections Menu to:

- Go back to a section you previously completed.
- Go back to a section you previously chose to skip.
- Go to the `NCL Script` section. This section only appears if you have completed all other sections.

When you select a section from the Sections Menu, you go to the Options Menu for that section. The only exception is when you select the NCL Script section. In that case, you go to the first screen of the section.

### 4.4.5 Saving an Incomplete Configuration

You can save a configuration without completing it. To do this, select either of the following from any Options Menu:

- Save the current configuration. This saves the information you have entered so far, and leaves you on the Options Menu.
- Save the current configuration and EXIT. This saves the information you have entered so far, and then returns you to the operating system.

Both options save your configuration information to a special data file. They do not create a new NCL script file. See Section 5.3.5 for details.

Once you have saved your incomplete configuration, you can complete it by using the Modify option; see Section 6.2.1.

## 4.5 Exiting and Quitting the DECNIS Configurator

To leave the DECNIS configurator, you can either exit or quit.

### 4.5.1 Definition of Exiting

**Exiting** means that the configurator saves all the information you have entered since starting the configurator and then returns you to the operating system.

### 4.5.2 How to Exit from the Configurator

You can save your configuration and exit at any of the following points:

- On any Options Menu, by selecting Save the current configuration and EXIT.
- In the NCL Script section, after you have created an NCL script file.
- In the NCL Script section, after you have created a CMIP file.

### 4.5.3 Definition of Quitting

**Quitting** means that:

- You leave the configurator and return to the operating system.
- The configurator does not save any information you have entered since the last time you saved your configuration or created an NCL script file.

### 4.5.4 How to Quit the Configurator

To quit the DECNIS text-based configurator, press F8. This will delete all of the information you have entered since the last time you selected a Save option, or created an NCL script file.

## 4.6 Errors when Running the DECNIS Configurator

If there are any errors when you are running the DECNIS text-based configurator, they will be recorded in the following log files:

- On PCs:

  C:\ *install-directory* \ NIS \ NIS_DNIS.LOG

  where *install-directory* is the installation directory.

- On OpenVMS systems:

  MOM$SYSTEM:NIS_DECNIS.LOG

- On Digital UNIX systems:

  /usr/lib/dnet/nis_decnis.log

# 5

# Creating the Configuration Files

## 5.1 Introduction

When you finish configuring the DECNIS, you need to do the following:

1. Create a master NCL script. See Section 5.3.1.

   This script holds the commands needed to configure your DECNIS. You create this within the DECNIS text-based configurator or clearVISN DECNIS configurator.

2. As an option, edit the user NCL script files. See Section 5.2.2.

   These are empty NCL script files. You use them to add NCL commands to modify your configuration (for example, if you want to add facilities that are not in the configurator).

3. Create a binary configuration file. See Section 5.2.3.

   This is the file that will be downline loaded to the DECNIS.

This chapter describes these files, how to create them using the DECNIS text-based configurator or the clearVISN DECNIS configurator, and how they are used.

The clearVISN DECNIS configurator is a Windows graphical user interface (GUI) program, running on Windows 95/NT PCs. For more information, refer to the *clearVISN DECNIS Configurator User Guide*.

## 5.2 Configuration Files: Introduction

This section describes:

- The master NCL script
- User NCL script files
- The CMIP file
- The combined file

### 5.2.1 The Master NCL Script

Both the DECNIS configurators use the information you enter, together with system defaults, to create an initialization script for the DECNIS. This initialization script contains Network Control Language (NCL) commands, and is known as the master NCL script. This script contains the commands necessary for configuring the DECNIS.

#### 5.2.1.1 The Master NCL Script in the Configurators

The master NCL script produced by the DECNIS text-based configurator is an ASCII text file of NCL commands.

The master NCL script produced by the clearVISN DECNIS configurator is not an ASCII text file. Instead, it is held entirely within the configurator. It is not possible to edit the master NCL script within the clearVISN DECNIS configurator; you can only view it. However, you can save the master NCL script to a file.

### 5.2.2 User NCL Script Files

In addition to generating the master NCL script, the configurator generates three **user NCL script files**.

When the user NCL script files are first generated, they are empty files. You can enter additional NCL commands in the user NCL script files. The DECNIS configurator compiles the user NCL script files and the master NCL script to create a loadable CMIP file.

#### 5.2.2.1 Purpose of the User NCL Script Files

The purpose of the user NCL script files is to allow you to change your DECNIS configuration without editing the master NCL script. Edit the user NCL script files if you want to:

- Change default information that you cannot change within the configurator, for example, some timer values.
- Set up facilities that you cannot set up within the configurator, for example, setting up the DECNIS as a CONS LAN/WAN Relay.

#### 5.2.2.2 Why Use the User NCL Script Files?

The master NCL script file is recreated whenever you run the DECNIS configurator. If you edit the master NCL script, and then rerun the configurator, any changes you have made will be lost. If you insert NCL commands in the user NCL script instead, your changes will be preserved.

Note that in any case you cannot edit the master NCL script produced by the clearVISN DECNIS configurator.

#### 5.2.2.3 Do Not Delete the User NCL Script Files

Do not delete the user script files, even if you will not use them. On all load hosts except Windows 95/NT, the master NCL script contains calls to the user script files. The CMIP file will not compile if the user script files are not present.

On Windows 95/NT hosts, you cannot delete the user script files, as they are held within the configurator.

#### 5.2.2.4 User NCL Script File Names

Table 5–1 lists the user NCL script file names. In the table, *client-name* is the load client name of the DECNIS.

See Section B.1 for the full file specifications of these files.

**Table 5–1 User NCL Script File Names**

| File Names: OpenVMS and Digital UNIX | File Names: MS–DOS PC | Contain | Where Called in Master NCL Script File |
|---|---|---|---|
| NIS_*client-name*_ EXTRA_CREATE.NCL | CREATE.NCL | CREATE commands | After standard entities have been created |
| NIS_*client-name*_ EXTRA_SET.NCL | SET.NCL | SET commands | After standard entities have been set |
| NIS_*client-name*_ EXTRA_ENABLE.NCL | ENABLE.NCL | ENABLE commands | After standard entities have been enabled |

On Windows 95/NT PC hosts, you can only edit the user NCL script files within the clearVISN DECNIS configurator.

Refer to Section 5.7 for special recommendations you should follow when editing the User NCL script files.

### 5.2.3 CMIP Files and Combined Files

Before you can downline load the DECNIS software, you must compile the NCL script and user NCL files into a loadable configuration file. This can be either a separate **CMIP file** or a **combined file**.

- A CMIP (Common Management Information Protocol) file is the binary, loadable version of the NCL script files. It can be loaded as a separate file, together with the software image and profile files.
- A combined file consists of the CMIP file, software image, and profile files combined into one file. Create this file if you want the DECNIS to reload all files from DECNIS nonvolatile (flash) memory.

It is possible to include more than one CMIP file in the combined file; see Chapter 1 in the *DECNIS Management* manual for details.

## 5.3 Using the DECNIS Text-Based Configurator

This section describes how to use the DECNIS text-based configurator to create a master NCL script, a CMIP file and a combined file.

### 5.3.1 Creating the Master NCL Script File

You create a master NCL script file in the final section in the DECNIS text-based configurator, `Create NCL Script`.

#### 5.3.1.1 Create NCL Script Section

**Requirement**

Before you can go to the `Create NCL Script` section, you must complete all the configurator sections. This means that for each section, you must do one of the following:

- Supply the required information on the screens.
- Select No on the introduction screen, to say that you want to skip that section.

**How to Reach the Create NCL Script Section**

Once you have completed all the configurator sections, you will go to the `Create NCL Script` section if you select either of the following:

- `Continue to a new section` from any Option Menu.
- `NCL Script` from any Sections Menu.

The `Create NCL Script` section will not appear on the Sections Menu unless you have completed all the sections.

#### 5.3.1.2 How to Create the Master NCL Script File

In the `Create NCL Script` section, follow these steps:

1. The first menu asks you to select an option:

   ```
   Create an NCL Script
   Go to Sections Menu
   ```

2. Select `Create an NCL Script`.

   Only select `Go to Sections Menu` if you want to modify any information you have already entered.

#### 5.3.1.3 Errors when Creating the NCL Script File

If the configurator cannot create the master NCL script file, a failure message appears at the foot of the screen, and the cursor stays on the option `Create an NCL Script`. You must correct the problem before reselecting the option.

### 5.3.2 Creating the CMIP File

You can create a separate CMIP file in the DECNIS text-based configurator only if you requested load-host loading in the load-host configurator. If you requested nonvolatile memory loading, go to Section 5.3.3.

To create a CMIP file in the DECNIS text-based configurator, follow these steps:

1. Go to the `Create NCL Script` section.
2. Create the NCL script file.
3. After the NCL script file is created, the following menu is displayed:

```
Create a CMIP file from the NCL script
Return to Sections Menu
Return to Main Menu
Exit from the configurator
```

4. Select `Create a CMIP file`.

### 5.3.3 Creating the Combined File

You can create a combined file in the DECNIS text-based configurator only if you requested nonvolatile memory loading in the load-host configurator. If you requested load-host loading, go to Section 5.3.2.

To create a combined file in the DECNIS text-based configurator, follow these steps:

1. Go to the `Create NCL Script` section of the DECNIS text-based configurator.
2. Create the NCL script file.
3. After the NCL script file is created, the following menu is displayed:

```
Create a combined image/CMIP/profile file
Return to Sections Menu
Return to Main Menu
Exit from the configurator
```

4. Select `Create a combined image/CMIP/profile file`.

### 5.3.4 Requirement to Create CMIP file or Combined File

If you edit the user NCL script files after exiting from the configurator, you must create either a new separate CMIP file or a new combined file before you reload the DECNIS.

### 5.3.5 DECNIS Data Files

The DECNIS text-based configurator saves each DECNIS configuration in its own DECNIS data file. When you modify a configuration, the DECNIS text-based configurator uses the data file to show the data previously entered. This data file is independent of any NCL script files or CMIP files.

The DECNIS data file name is:

- On MS–DOS PC load hosts:

  *client-name*.DAT

- On OpenVMS and Digital UNIX load hosts:

  NIS_*client-name*.DAT

where *client-name* is the load client name of the DECNIS.

For the full file specifications of these files, see Section B.1.

The DECNIS data file is also used for saving an incomplete configuration; see Section 4.4.5.

Do not delete the DECNIS data files. You must have a data file in order to use the DECNIS text-based configurator to modify a configuration or complete an incomplete configuration.

#### 5.3.5.1 Saved Version of the DECNIS Data File

When the DECNIS text-based configurator creates a new DECNIS data file, it saves the old one, with a different file name extension.

Normally, the previous DECNIS data file is saved with the file name:

- On MS–DOS PC load hosts:

  *client-name*.BAK

- On OpenVMS and Digital UNIX load hosts:

  NIS_*client-name*.BAK

However, if you install a new version of DECNIS software, and use the DECNIS configurator to modify an existing configuration, the configurator saves the previous DECNIS data file with a different file name:

- On MS–DOS PC load hosts:
  *client-name*.V*nn*
- On OpenVMS and Digital UNIX load hosts:
  NIS_*client-name*.DAT_V*nn*

where: *client-name* is the load client name for the DECNIS.
*nn* is the version number of the previous version of DECNIS software.

Refer to Table B–1 to Table B–3 for the full file specifications.

## 5.4 Using the clearVISN DECNIS Configurator

This section describes how to use the clearVISN DECNIS configurator to create a master NCL script, edit the user NCL script files, and create a separate CMIP file or a combined file.

### 5.4.1 Creating a Master NCL Script

The clearVISN DECNIS configurator automatically enters NCL commands in a master NCL script as you configure your DECNIS. To view this script, click the **NCL scripts** button on the Main Navigation window. The script is on the Generated NCL tab page,

Note that you cannot edit the NCL commands on the Generated NCL tab page.

### 5.4.2 Editing User NCL Script Files

On Windows 95/NT hosts, you can only edit the user NCL script files within the clearVISN DECNIS configurator. Follow these steps:

1. Click the **NCL scripts** button on the Main Navigation window.
2. You will go to the NCL Scripts window. You can now add NCL commands on the three tab pages:
   - Extra Creates
   - Extra Sets
   - Extra Enables
3. If you wish, insert an existing script file on one or more of the tab pages. To do this, follow these steps:
   a. Select the tab page.
   b. One the **NCL** menu on the menu bar, select **Insert**.
      The **Open** window appears. You can now select a directory and file.

c. When you have selected a file, click **OK**. The file will be inserted on the tab page.

### 5.4.3 Creating a Separate CMIP File

To create a separate CMIP file, follow these steps:

1. On the Main Navigation window, click the **System** button.
2. On the Load Options tab page, do the following:
   a. Under Flash Contents, make sure the CMIP script box is **not** checked.
   b. If you want both the software image and the CMIP file to be always reloaded from a load host, select **From Network** under Image Loads.

      If you want the CMIP file to be reloaded from a load host, but the software image to be reloaded from flash memory, select **From Flash** under Image Loads.
   c. Click **OK**.
   d. On the Main Navigation Window, click the **Compile** button.

### 5.4.4 Creating a Combined File

To create a combined file, follow these steps:

1. On the Main Navigation window, click the **System** button.
2. On the Load Options tab page, do the following:
   a. Under Flash Contents, make sure the CMIP script box is checked.
   b. Under Image Loads, select **From Flash**.
   c. Click **OK**.
   d. On the Main Navigation Window, click the **Compile** button.

## 5.5 Using Commands

This section describes how to create a separate CMIP file and a combined file from the command line. Note that you can only do this on OpenVMS, Digital UNIX or MS–DOS PC load hosts.

### 5.5.1 Creating a CMIP File

To create a CMIP file after exiting from the DECNIS text-based configurator, follow these steps:

1. In the Create NCL Script section of the configurator, create an NCL script file.
2. On the CMIP file menu, select Exit from the configurator.
3. Edit the user NCL script files if you wish, as described in Section 5.2.2.
4. Enter the command to create a CMIP file:
   - On MS–DOS PC load hosts:

     C:*install-directory*\NIS\NIS_SCPL *client-name*
   - On OpenVMS load hosts:

     $ @SYS$MANAGER:NIS$SCRIPT_COMPILE NIS_*client-name*.NCL
   - On Digital UNIX load hosts:

     # /usr/lib/dnet/nis_script_compile nis_*client-name*

   where *install-directory* is the directory where the DECNIS software is installed
   *client-name* is the DECNIS load client name

### 5.5.2 Creating a Combined File

To create a combined file after exiting from the DECNIS text-based configurator, follow these steps:

1. Create a CMIP file, either within the DECNIS text-based configurator, or as described in Section 5.5.1.
2. Run the combine procedure, as described in Section 8.5.2.

## 5.6 Logging Errors During CMIP Compilation

When you create a CMIP file, the CMIP compiler checks the NCL script for errors. If there are any errors, they are written to a log file.

The type of log file varies according to the type of load host:

- On OpenVMS VAX load hosts, the log file contains both syntax errors and semantic errors. If there are errors of either type, no CMIP file is created.
- On OpenVMS Alpha, Digital UNIX, MS–DOS PC and Windows 95/NT load hosts, the log file only contains syntax errors. If there are syntax errors, no CMIP file is created.

However, once you have created a CMIP file, a separate **NCL checking** utility can be run. This checks the NCL script again, and produces a log file containing any semantic errors.

Thus, on these load hosts, there are two log files: one with syntax errors and one with semantic errors.

The term **syntax errors** refers to mistakes in the format of individual NCL commands, for example, spelling mistakes.

The term **semantic errors** refers to mistakes in the script as a whole, for example, incompatible commands or missing commands.

## 5.6.1 CMIP Error Log Files

Table 5–2 shows the names of the log files produced by CMIP compilation.

**Table 5–2 CMIP Error Log Files**

| Load Host | CMIP Log File | Error Type |
|---|---|---|
| MS–DOS PC | *install-directory*\CLIENTS*client*\NCLSCRPT.LOG | Syntax |
| OpenVMS Alpha | SYS$COMMON:[MOM$SYSTEM]NIS_*client*.LOG | Syntax |
| OpenVMS VAX | SYS$COMMON:[MOM$SYSTEM]NIS_*client*.LOG | Syntax and semantic |
| Digital UNIX | /usr/lib/dnet/nis_*client*.log | Syntax |

where *client* is the load client name of the DECNIS.

On Windows 95/NT hosts, the configurator displays CMIP compilation syntax errors in a window, when you compile the NCL script. There is no log file.

## 5.6.2 The NCL Checking Utility in the Configurator

On Digital UNIX and Windows 95/NT load hosts, the configurator automatically runs the NCL checking utility after it has created a CMIP file.

### 5.6.2.1 Log File on Digital UNIX Hosts

When it runs the NCL checking utility on Digital UNIX hosts, the DECNIS text-based configurator produces the following log file containing semantic errors:

/usr/lib/dnet/nis_*client*.lis

#### 5.6.2.2 Log File on Windows 95/NT Hosts

On Windows 95/NT hosts, you can only view the log file within the clearVISN DECNIS configurator. Follow these steps:

1. On the Main Navigation window, click the **Compile** button.
2. When the configurator has finished compiling, click the **Log** button. You will see the NCL checker log. All NCL directives are shown.

**Correcting Errors**

If there are semantic errors, the Compile window will display an alert. To correct errors, follow these steps:

1. Click the **Log** button to open the NCL checker log.
2. Find the error in the log window. NCL directives with an error are marked !ERROR.
3. On the Main Navigation window, click the **NCL Scripts** button. You can leave the NCL checker log window open while you do this.
4. On the appropriate Extra tab page, find and correct the error.

### 5.6.3 Running the NCL Checking Utility from the Command Line

You can run the NCL checking utility from the command line on MS–DOS PC, OpenVMS Alpha, and Digital UNIX load hosts. Note that you supply your own name for the log file.

#### 5.6.3.1 Procedure

To run the NCL checking utility, enter the appropriate command as shown in the table:

| On this load host... | Enter this command... |
|---|---|
| MS–DOS PC | C:\ *install-dir* \ NIS_NCHK *ncl-script* *log-file* |
| OpenVMS Alpha | NIS$NCHK := = $ SYS$SYSTEM:NIS$NCHK.EXE<br>NIS$NCHK *ncl-script* *log-file* |
| Digital UNIX | /usr/lib/dnet/nis_nchk *ncl-script* *log-file* |

where:

| | |
|---|---|
| *install-dir* | is the directory where the DECNIS was installed. |
| *ncl-script* | is the name of the NCL script to be checked. |
| *log-file* | is the name you want to use for the log file. |

#### 5.6.3.2 Result

When you run the NCL checking utility, any errors and warnings are displayed on the screen. In addition, all NCL script directives, together with errors and warnings, are written to the named log file.

### 5.6.4 Special Requirements

The NCL checking utility does not allow any words in NCL commands to be abbreviated to fewer than three characters. If any words have fewer than three characters, the NCL checking utility issues an error message.

### 5.6.5 Example Log File

The log file produced by the NCL checking utility lists the NCL script directives in a format similar to the following:

```
! script directive 1
create modem connect

! script directive 2
create csma-cd

! script directive 3
create fddi

! script directive 4
create mop
```

When an error is logged, it is in a format similar to the following:

```
! script directive 68
set routing circuit w622-0 manual data link sdu size 4492
! ERROR - set on entity not created circuit w6229-0 manual data link sdu size 4492
```

### 5.6.6 CMIP Errors Logged during Loading

If there are any errors in the NCL script which have not been corrected before you attempt to load the DECNIS, they are displayed on the DECNIS console during loading.

See Section 7.5.1 for more information about errors logged on the console.

### 5.6.7 Location of Load Files

Refer to Appendix B for the location on all load hosts of the files loaded to the DECNIS.

## 5.7 Special Recommendations for Editing the User NCL Script Files

The following list contains advice which may be useful when editing the user NCL script files.

- Do not enter hyphens at the end of comment lines.
- Insert bridge filtering commands in the ENABLE user NCL script file – including CREATE and SET commands.
- (OpenVMS load hosts only) When setting up bridge filtering, when you specify the identifiers of the FILTER TYPE and FILTER PID entities, remove all the hyphens, and insert %X at the front.

  For example, to set up filtering of Ethernet frames with a protocol identifier of 60-03, enter the following line in the ENABLE user NCL script file:

  ```
  CREATE BRIDGE FILTER TYPE %X6003
  ```

  Then specify the ports that can receive and forward frames bearing this protocol identifier, as described in the manual *DECNIS Management*.

  Note that this only applies to entering commands in the user NCL script files. Do not remove hyphens or insert the %X prefix when issuing NCL commands to a running DECNIS.
- Insert the event dispatcher commands PASS, BLOCK, IGNORE and SET into the SET user NCL script file.

  An example of such a command is:

  ```
  PASS EVENT DISPATCHER OUTBOUND STREAM stream_1
  ```
- If the DECNIS is connected to an OSI-compliant router which is not using the IS-IS routing protocol, add the following command in the SET user NCL file:

  SET ROUTING CIRCUIT *circuit-name* DNA NEIGHBOR FALSE

  where *circuit-name* is the name of the circuit connecting the DECNIS to the OSI-compliant router.

### 5.7.1 Long NCL Commands

The maximum input for NCL commands is as follows:

- On OpenVMS systems, 1024 characters
- On Digital UNIX systems and PCs, 2048 characters

The DECNIS configurator may generate NCL commands that exceed this maximum length, if you provide sufficient input for certain configuration options. If this happens, the command will fail. This failure will be reported in the configurator log file.

To correct this problem, you will need to edit the NCL master script file and replace the long command with several separate commands.

**Example**

You can replace this long NCL command in the NCL script file:

```
set routing circuit L602-3-1 -
   alternative subnet addresses -
   {{ address =  1.1.50.50, mask =  255.255.255.0 }, -
   {address =  1.1.50.51, mask =  255.255.255.0 }}
```

by the following separate commands:

```
add routing circuit L602-3-1 -
   alternative subnet addresses -
   {{ address =  1.1.50.50, mask =  255.255.255.0 }}
add routing circuit L602-3-1 -
   alternative subnet addresses -
   {{ address =  1.1.50.51, mask =  255.255.255.0 }}
```

# 6

# Modifying Your Configuration in the DECNIS Text-Based Configurator

## 6.1 Introduction

This chapter describes how to modify a completed configuration in the DECNIS text-based configurator.

## 6.2 How to Modify Your Configuration

You can use the DECNIS text-based configurator to modify an existing configuration. Follow these steps:

1. Start the DECNIS text-based configurator, as described in Chapter 2.
2. Select `Modify an existing configuration` from the Main Menu.
3. The screen shows a list of load client names. Select the DECNIS you wish to reconfigure.
4. The screen shows the Sections Menu. Select a section to modify.
5. The screen shows the Options Menu for that section. You can add, delete, or modify information in that section.
6. To make changes to another section, select `Go to Sections Menu` from any Options Menu. Then select a section.
7. When you have finished making changes, create the NCL script file. You will go to the `Create NCL Script` section if you do either of the following:
   - Select `Continue to new section` from any Options Menu.
   - Select `NCL Script` from the Sections Menu.
8. When you reach the `Create NCL Script` section, follow the instructions in Section 5.3.1.

### 6.2.1 Completing an Incomplete Configuration

If you have saved an incomplete configuration, and then want to complete it, follow these steps:

1. Start the DECNIS text-based configurator, as described in Chapter 2.
2. Select `Modify an existing configuration` from the Main Menu.
3. The screen shows a list of load client names. Select the DECNIS you want. You will go to the Options Menu for the next section you need to complete.
4. Complete the section, by selecting Add, Configure or Modify, as appropriate.
5. When you have completed the section, select `Continue to new section` from the Options Menu. You will go to the next section you need to complete.
6. Repeat steps 4 and 5 until the configurator takes you to the NCL Script section.
7. Create an NCL script, as described in Section 5.3.1.2.

## 6.3 Steps to Take After Modifying a DECNIS Configuration

Sometimes, if you delete or modify information in one section, it will change or delete information in another section. To make sure that you have entered all necessary information, do the following:

1. After you have made a modification, finish the section you are in.
2. Select `Continue to new section` on the Options Menu. This will always take you to the next uncompleted or unseen section.
3. If it takes you to another Options Menu, complete the section by selecting Add, Configure or Modify, as appropriate.
4. Select `Continue to new section` on every Options Menu until you arrive at the NCL Script section.

## 6.4 Effects of Modifying a DECNIS Configuration

Table 6–1 lists the modifications and deletions that have an important effect on the rest of your configuration.

**Table 6–1 Effect of Modifying DECNIS Information**

| Modification | In this section | Affects these sections: |
|---|---|---|
| Change type of Network Interface Card in a slot | Network Interface Cards | **Lines**, **X25 Circuits**, **OSI and IP Reachable Addresses**, **PVCs**, **Groups**, **LLC2** – Deletes all information for lines/DTEs on the previous Card |
| Change from Level 1 to Level 2 | Routing | **Routing**—Deletes IP route propagation information<br>**Lines**, **X25 Circuits**, **Reachable Addresses**– You may need to add information, as more functions are available to Level 2 routers |
| Change from Level 2 to Level 1 | Routing | **Routing**—Deletes IP route propagation information and Level 2 specific information<br>**Lines** – Deletes Level 2 specific information, such as Level 2 cost, interphase links<br>**X25 Circuits**—Deletes DA circuits<br>**OSI Reachable Addresses**—Deletes all information<br>**IP Reachable Addresses**—Deletes all IP reachable addresses for DA circuits |
| Change routing algorithm | Routing | **Routing**—Deletes route propagation information. Changing from Phase V only to Phase IV only deletes Phase V area addresses<br>**Lines**—Changing from Level 2, Phase V to Level 2, Phase IV deletes interphase links |
| Change from X.25 to another protocol | Lines | **Lines**—Deletes information about DTEs and DTE Classes for the line<br>**X25 Circuits**—Deletes X.25 routing circuits for the line<br>**Reachable Addresses (OSI or IP)**—Deletes reachable addresses for circuits using the line<br>**PVCs**—Deletes PVCs using the line<br>**CUGs**—Deletes CUGs using the line |
| Change to X.25 from another protocol | Lines | **Lines**—Deletes all HDLC/PPP circuit information<br>**Reachable Addresses (OSI or IP)**—Deletes reachable addresses for the circuit using the line |
| Deleting a line (except X.25) | Lines | **Reachable Addresses (OSI or IP)**—Deletes all reachable addresses for the circuit using the line |

(continued on next page)

**Table 6–1 (Cont.) Effect of Modifying DECNIS Information**

| Modification | In this section | Affects these sections: |
|---|---|---|
| Deleting an X.25 line | Lines | **Lines** – Deletes all DTEs/DTE Classes for the line<br>**X25 Circuits**—Deletes X.25 routing circuits using the line<br>**Reachable Addresses**—Deletes reachable addresses for circuits using the line<br>**PVCs** – Deletes PVCs using the line<br>**CUGs**—Deletes CUGs using the line |

## 6.5 Effects of Modifying Load-Host Information

If you modify information in the load-host configurator, the modifications may affect or even invalidate information entered in the DECNIS text-based configurator.

For this reason, always rerun the DECNIS text-based configurator after changing the load-host configuration for a DECNIS.

- If DECNIS information has been deleted, you will need to reenter it.
- If DECNIS information has not been deleted, you will simply need to rerun the configurator.

Follow the steps in Section 6.5.1.

### 6.5.1 Steps to Take After Modifying a Load-Host Configuration

This section describes how to update the DECNIS configuration after changing load-host information.

1. Exit the load-host configurator.
2. Run the DECNIS configurator.
3. Select the Modify option from the Main Menu.
4. Select the load client name for the DECNIS that you have just modified in the load-host configurator.
5. You will now see a list of sections. Select any section.
6. On the Options Menu for the section, select `Continue to new section`.

7. If the DECNIS text-based configurator has been able to update the DECNIS configuration automatically, you will go to the NCL script section. Go to step 10.
8. If the DECNIS text-based configurator cannot update the DECNIS configuration automatically, you will go to the Options Menu of a section where information has been deleted. Go to step 9.
9. Follow these steps:
   - If you are on the Routing Options Menu, select `Modify`.

     If you are on any other Options Menu, select `Add`, to set up new items, for example, event streams.
   - Enter the required information.
   - When you have finished modifying, select `Continue to new section` from the Options Menu.
   - If you go to an Options Menu for another section, repeat this step.
10. In the NCL Script section, follow the instructions on the screen.

### 6.5.2 How the DECNIS Configurator Uses Load-Host Information

The DECNIS text-based configurator uses the information you entered during load-host configuration to find out:

- The DECNIS hardware units set up for loading.
- The DECNIS hardware type for each hardware unit.
- The DECNIS Phase IV address (if present).
- The DECNIS system IP address (if present).
- Whether or not the DECNIS configurator will use naming service namespacesto find addressing information.

This load-host information affects the information you enter during DECNIS configuration.

### 6.5.3 Load-Host Modifications Affecting DECNIS Configuration

Table 6–2 lists the modifications in load-host configuration that will affect or invalidate your DECNIS configuration.

**Table 6–2 Effect of Modifying Load-Host Information on DECNIS Information**

| Change to Load-Host Configuration | Effect on DECNIS Configuration | What You Need to Do |
|---|---|---|
| Changing Hardware Address and/or Phase IV address | Changes the CREATE SESSION CONTROL KNOWN TOWER command for the DECNIS in the master NCL script | Rerun the DECNIS text-based configurator, as described in Section 6.5.1 |
| Deleting the Phase IV address | Invalidates Phase IV routing on the DECNIS | Rerun the DECNIS text-based configurator. Reenter information in the Routing section |
| Entering the Phase IV address (where there was none previously) | Allows selection of Phase IV routing for Level 1 routers, and of Phase IV routing at both levels for Level 2 routers | Rerun the DECNIS text-based configurator. Reenter information in the Routing section |
| Changing IP Address | Changes the DECNIS system IP address in the master NCL script | Rerun the DECNIS text-based configurator, as described in Section 6.5.1 |
| Changing from BOOTP loading to MOP only loading | Deletes the system IP address previously set up in the load-host configurator | Rerun the DECNIS text-based configurator. Reenter system IP address in the Routing section |
| Changing from MOP only loading to BOOTP loading | The DECNIS IP address that you enter during load-host configuration replaces the system IP address previously entered during DECNIS configuration | Rerun the DECNIS text-based configurator |
| Changing between Use of a naming service and Nonuse of a naming service | Deletes all information in the following sections:<br>X.25 Server Clients<br>Event Logging<br>Incoming Security<br>Outgoing Security | Rerun the DECNIS text-based configurator. Reenter the information |

## 6.6 Copying and Modifying a Configuration

You may want to use similar configurations for several DECNIS systems, for example, if they have identical hardware configurations. One way to do this by copying an existing configuration and then modifying it. This section describes how you do this.

Note that in the instructions:

- The DECNIS system from which you are copying the configuration is called the **first DECNIS**.
- The system to which you are copying is called the **second DECNIS**.

### 6.6.1 Before You Copy

Before you can copy a configuration, you need to do the following:

1. On a load host, install the DECNIS software, as described in the installation chapter for your load host.
2. Configure both the **first DECNIS** and the **second DECNIS** for loading. To do this, run the load-host configurator and set up downline loading details, as described in the installation chapter for your load host.
3. Configure the **first DECNIS**, as follows:
   - Run the DECNIS text-based configurator, and configure the DECNIS.
   - Create an NCL script (and a CMIP file, if you wish).
   - Select `Exit from the configurator`.

### 6.6.2 Copying the Configuration to Another DECNIS

To copy the configuration so that it applies to the **second DECNIS**, follow these steps:

1. Copy the DECNIS data file for the first system to a new file. In the file name of the new file, substitute the load client name of the second system for that of the first system.

   For example, the load client name is SOUTH1 for the first DECNIS and NORTH2 for the second DECNIS. On an OpenVMS system, use this command:

   ```
   $ COPY SYS$COMMON:[MOM$SYSTEM]NIS_SOUTH1.DAT -
   _$ SYS$COMMON:[MOM$SYSTEM]NIS_NORTH2.DAT
   ```

   Refer to Appendix B for the location and name of the DECNIS data file on all supported load hosts.

2. Start the DECNIS configurator.
3. Select Modify an existing configuration from the Main Menu.
4. Select the load client name of the second DECNIS. In the example above, this would be NIS_NORTH2.DAT.
5. Now, modify the configuration so that it is correct for the second DECNIS:
   - From the DECNIS Node Options Menu, select Go to Sections Menu.
   - Select the first section to be modified.

**Note**

You must modify the configuration. The addresses entered for the system you copy from will not be correct for the system you copy to.

   - When you reach the Options Menu, select the Sections Menu, and select another section to modify.
   - When you have finished modifying, select Continue to Next Section from any Options Menu. This will take you to the NCL Script section.

### 6.6.3 Sections to Check

Be sure to check the addressing information in the following sections:

- Routing
- Lines and DTEs
- X.25 Circuits (if configured)
- Tunnel Circuits (if configured)
- X.25 Server Clients (if configured)
- Event Logging
- Incoming Security (if configured)
- Outgoing Security (if configured)

It is safest to go through each section in turn, beginning with Network Interface Cards.

# Part II
## Loading

This part describes how to load the DECNIS software onto the DECNIS hardware, and how to set up various types of loading for the DECNIS.

It contains the following chapters:

- Chapter 7 describes how to load the configured software to the DECNIS.
- Chapter 8 describes nonvolatile memory loading.
- Chapter 9 describes how to set up the DECNIS as a MOP proxy load host.
- Chapter 10 describes how to set up the DECNIS as a BOOTP gateway.

# 7

# Loading a DECNIS

## 7.1 Introduction

This chapter describes the following:

- Loading the DECNIS from a load host.
- Reloading the DECNIS after the initial load.
- Restricting loading and dumping on the DECNIS.
- How the DECNIS loads and dumps its software.

The following chapters describe other aspects of loading and dumping:

- Chapter 8 describes how to set up nonvolatile memory loading (also known as flash memory loading).
- Chapter 9 describes how to set up the DECNIS as a proxy load host.
- Chapter 10 describes how to set up the DECNIS as a BOOTP gateway.

### 7.1.1 MOP and BOOTP Loading

You can downline load the DECNIS using MOP (Maintenance Operations Protocol), BOOTP/TFTP or both.

- MOP is a Digital-specific protocol used for loading and dumping.
- BOOTP and TFTP are protocols used for loading and dumping, defined in RFCs 783 and 951. BOOTP determines the IP address of a device being loaded, and the names of load files. TFTP is the protocol used for loading.

#### 7.1.1.1 Types of Connection for Loading

A load host can be connected to the DECNIS in either of two ways:

- Directly. The load host is on the same LAN as the DECNIS, or is directly connected to it by a synchronous line. Note that loading over a synchronous line is not supported by all load hosts.
- Indirectly. The load host is connected through another DECNIS acting as a proxy load host or BOOTP gateway (see Chapter 9 and Chapter 10).

## 7.2 Loading the DECNIS for the First Time

To load a DECNIS that has not been loaded previously, follow these steps:

1. Connect the DECNIS hardware unit to the LAN.
2. Plug the unit into the power supply and power up.

The DECNIS hardware unit then follows the steps described in Section 7.11.

## 7.3 Updating the DECNIS

If you have a management processor card (MPC-II or MPC-III) you can use the console to make dynamic updates to the contents of flash memory. You can add new script files, mark old script files for deletion, or clear flash memory and load a new image using the console commands. Refer to *DECNIS Management* for more information about controlling the contents of flash memory using the console.

## 7.4 Reloading the DECNIS

If the DECNIS has been loaded previously, you can use the following methods to reload it:

- Entering NCL commands.
- Powering up the DECNIS.
- Entering console commands. Refer to the manual *DECNIS Management* for more information about the DECNIS console.

Section 7.4.1 to Section 7.4.3 describe various methods of reloading.

For more information about the NCL commands described here, refer to the NCL reference manual for your operating system, or NCL help. Note that on Windows 95/NT PCs, NCL help is available in a Windows help file.

### 7.4.1 Reloading Using the Default Type of Loading

The default type of loading is the type of loading set up for the DECNIS during load-host configuration: full nonvolatile memory loading, partial nonvolatile memory loading, or load host loading.

#### 7.4.1.1 Entering the NCL LOAD Command

To reload the DECNIS using its default method, enter the following command:

- On an MS–DOS PC or Windows 95/NT load host:

  ```
  NCL> LOAD NODE decnis/username/password DEVICE UNIT MP*
  ```

- On an OpenVMS load host:

  ```
  NCL> LOAD NODE decnis"username password" DEVICE UNIT MP*
  ```

- On a Digital UNIX load host:

  ```
  NCL> LOAD NODE decnis/username/password DEVICE UNIT MP*
  ```

where: *decnis* is the node name of the DECNIS. On an MS–DOS PC load host, the node name is a DECnet Phase IV address or node name. On a Windows 95/NT PC load host, the node name is an IP address or IP node name.

*username* and *password* are the user name and password to enter when using NCL commands to manage the DECNIS.

#### 7.4.1.2 Powering Up

If you power up the DECNIS, as described in Section 7.2, this will have the same effect as using the LOAD command in Section 7.4.1.1.

#### 7.4.1.3 Using the Console

Enter the following command at the DECNIS console:

```
console> restart
```

This will have the same effect as the LOAD command in Section 7.4.1.1.

### 7.4.2 Reloading from a Load Host

Section 7.4.2.1 to Section 7.4.2.3 explain how to cause the DECNIS to reload from a load host, rather than from nonvolatile (flash) memory.

#### 7.4.2.1 Entering NCL Commands

To cause the DECNIS to reload from a load host, enter the commands shown below. In each case, the first command tells the DECNIS to load from a load host the next time it reloads, and the second command reloads the DECNIS.

- On an MS–DOS or Windows 95/NT PC load host:

```
NCL> SET NODE decnis/username/password HARDWARE -
_NCL> DEBUG FLAGS 1073741952

NCL> LOAD NODE decnis/username/password DEVICE UNIT MP*
```

- On an OpenVMS load host:

```
NCL> SET NODE decnis"username password" HARDWARE -
_NCL> DEBUG FLAGS 1073741952

NCL> LOAD NODE decnis"username password" DEVICE UNIT MP*
```

- On a Digital UNIX load host:

```
NCL> SET NODE decnis/username/password HARDWARE -
_NCL> DEBUG FLAGS 1073741952

NCL> LOAD NODE decnis/username/password DEVICE UNIT MP*
```

where *decnis*, *username* and *password* are defined as in Section 7.4.1.1.

Note that these commands do not specify which load host will be used.

#### 7.4.2.2 Powering Up

If you power up the DECNIS with the dump button pressed in, it will have the same effect as issuing the commands in Section 7.4.2.1, that is, it will attempt to load from the load host.

#### 7.4.2.3 Using the Console

Enter the following command at the DECNIS console:

```
console> load -NETWORK
```

### 7.4.3 Reloading from a Specified MOP Load Host

You can cause the DECNIS to reload from a particular MOP load host, provided that the load host is reachable from the DECNIS.

To reload the DECNIS from a particular MOP load host, enter the following NCL command:

- On an MS–DOS PC load host:

```
NCL> LOAD NODE load-host/username/password MOP CLIENT client-name
```

Although MOP is not supported on Windows 95/NT load hosts, you can still issue this command from these load hosts to specify another load host that does support MOP.

- On an OpenVMS load host:

  ```
  NCL> LOAD NODE load-host"username password" MOP CLIENT client-name
  ```

- On a Digital UNIX load host:

  ```
  NCL> LOAD NODE load-host/username/password MOP CLIENT client-name
  ```

where: *load-host* is the DECnet node name of the MOP load host.

*username* and *password* are the user name and password to enter when using NCL commands to manage the DECNIS.

*client-name* is the DECNIS MOP client name, as set up on the MOP load host.

## 7.5 Errors While Loading

If there are any errors in the NCL script files which have not been detected during CMIP conversion, then the DECNIS does the following:

- Logs errors to the console terminal, as described in Section 7.5.1.
- Logs script exception events, provided that the errors do not prevent communication with the event sink. Refer to the online manual *DECNIS Problem Solving* for a description of exception events.

### 7.5.1 NCL Script Errors Logged to the Console Terminal

If there is a console terminal connected to the DECNIS, the console will display NCL script errors in the following format:

```
Error processing script directive n, code m
```

where *n* is the NCL script directive number for the command that is in error, and *m* is a hexadecimal number that Digital can use to analyze the error.

#### The NCL Checking Utility

To help avoid NCL errors, you should run the NCL checking utility before loading the DECNIS. This utility checks the DECNIS NCL script, and produces a log file. The log file shows the directive numbers of the commands in your NCL script, together with errors and warnings.

The NCL checking utility may not find all errors. However, the log file it produces is useful in any case, because it lists each NCL script directive together with its directive number. This enables you to match the directive

numbers in console error messages against the actual commands in the NCL script, so that you can pinpoint the commands that are wrong.

In addition, the log file can help in interpreting script exception events, as such events may include directive numbers.

Refer to Section 5.6 for more information about the NCL checking utility.

## 7.6 Disabling and Restoring Loading from a MOP Load Host

This section describes how to disable loading from one or more MOP load hosts, and restore it after it has been disabled.

Note that Windows 95/NT load hosts do not support MOP loading, so these commands do not apply to them.

### 7.6.1 Disabling MOP Loading

To prevent loading from a particular MOP load host, use NCL to delete the MOP Client entity for the DECNIS on that load host.

### 7.6.2 Restoring MOP Loading

If you have deleted one or more DECNIS MOP Client entities, you can use the load-host configurator Restore option to recreate these MOP client entities. There are two ways to use Restore:

- Use the `Restore a router` option in the load-host configurator to restore an individual DECNIS.
- Use the automatic Restore procedure to restore all DECNIS systems set up by the load-host configurator.

#### Automatic Restore

To use automatic Restore, enter the command appropriate for your load host:

- MS–DOS PC load hosts:

  `C:\`*`install-directory`*`\NIS\nis_hcfg/restore`

  where *install-directory* is the installation directory.

- OpenVMS load hosts:

  ```
  $ @SYS$MANAGER:NIS$HOST_CONFIG RESTORE
  ```

- Digital UNIX load hosts:

  ```
  # /usr/lib/dnet/nis_host_config -r
  ```

## 7.7 Enabling Dumping

The DECNIS will by default prevent dumping on all connections. This allows the DECNIS to restart quickly after failure; typically, the DECNIS takes about two minutes to restart if it does not dump, as compared to 30 minutes if it does dump.

Section 7.7.1 and Section 7.7.2 describe how to enable dumping on the DECNIS.

### 7.7.1 Enabling Dumping Temporarily

You may want the DECNIS to dump the next time it fails, but to revert to not dumping thereafter. To do this, issue the following command interactively:

```
NCL> SET HARDWARE DUMP CONTROL dump-type
```

where *dump-type* is one of the following:

FULL DUMP
SYSTEM PROCESSOR DUMP

### 7.7.2 Enabling Dumping Permanently

To enable dumping permanently, enter the SET HARDWARE DUMP CONTROL command described in Section 7.7.1 in the user SET NCL script.

### 7.7.3 Dumping Using the Dump Button

If you press the red dump button on the DECNIS hardware unit, the DECNIS will dump, regardless of whether you have enabled dumping.

## 7.8 Restricting Connections Used for Loading and Dumping

### 7.8.1 Introduction

By default, the DECNIS does the following:

- Attempts to load over all connections.
- Prevents dumping over all connections.

This section and Section 7.9 describe how to:

- Selectively restrict loading.
- Selectively restrict dumping, provided that you have first explicitly enabled dumping on the DECNIS (see Section 7.7).

### 7.8.2 Types of Restriction

You can restrict DECNIS loading and dumping by specifying either or both of the following:

- The Network Interface Cards that are not allowed to be used for loading and/or dumping
- The load protocols that are not allowed to be used for loading and/or dumping

**Example**

1. You can ensure that the DECNIS will only load through one of its CSMA/CD connections. This would avoid delays in attempting to load through synchronous connections.
2. You can prevent the DECNIS from using BOOTP to perform dumps, ensuring that a MOP load host will always receive the dump file.

### 7.8.3 Cannot Restrict Individual Connections

Although you can restrict the cards used for loading and dumping, all ports on a particular card must have the same ability to load and dump . For example, you cannot specify that one port on a DEC WANcontroller 622 card can be used for loading and dumping, but the other cannot. If the card is enabled for loading and dumping, then all ports can be used for loading and dumping.

### 7.8.4 Restrictions on Protocols Used for Loading and Dumping

The only protocols supported for loading and dumping over serial lines are DEC HDLC and PPP.

For this reason, you must disable loading and dumping on any WAN card that has lines configured to use other protocols (for example, X.25). To disable loading and dumping, use the ADD HARDWARE SLOT FUNCTIONS DISABLE command. See Section 7.9.3 for an example.

### 7.8.5 Restrictions on Cards Used for Loading and Dumping

You cannot use the DEC ATMcontroller Network Interface Card (W631) for loading or dumping.

## 7.9 Commands to Manage Loading and Dumping Restrictions

This section describes the FUNCTIONS DISABLED commands used to restrict, or remove restrictions from, loading and dumping.

### 7.9.1 Entity and Attributes Used for Restricting Loading and Dumping

To restrict loading and dumping on a card, you use the NCL entity HARDWARE SLOT, with its attribute FUNCTIONS DISABLED.

#### 7.9.1.1 HARDWARE SLOT Entity

Each card on a DECNIS has an associated HARDWARE SLOT entity. The name of this entity is the number of the slot in which the card is inserted.

#### 7.9.1.2 FUNCTIONS DISABLED Attribute of the HARDWARE SLOT Entity

The FUNCTIONS DISABLED attribute determines whether the DECNIS can load and dump over the connections on a card, and the protocols it can use.

Possible values are shown in Table 7–1.

**Table 7–1 Values of FUNCTIONS DISABLED Attribute**

| Value | Description |
|---|---|
| MOP LOAD REQUESTER | Loading using MOP over all lines associated with the backplane slot is prohibited |
| MOP DUMP REQUESTER | Dumping using MOP over all lines associated with the backplane slot is prohibited |
| IP LOAD REQUESTER | Loading using BOOTP/TFTP over all lines associated with the backplane slot is prohibited |
| IP DUMP REQUESTER | Dumping using BOOTP/TFTP over all lines associated with the backplane slot is prohibited |

### 7.9.2 Commands Used to Restrict Loading and Dumping

This section lists the commands you can use to restrict loading and dumping, or change previous restrictions on loading and dumping.

#### Do Not Include Commands in NCL Script File

The value of the FUNCTIONS DISABLED command for each slot is stored in nonvolatile memory on the DECNIS, and does not change when the DECNIS is rebooted. For this reason, do not add any FUNCTIONS DISABLED commands to the user NCL script files.

You can override the FUNCTIONS DISABLED values in nonvolatile memory by entering FUNCTIONS DISABLED commands interactively.

### 7.9.3 ADD Command

Use this command to add loading and/or dumping to the functions that are disabled for the hardware slot.

#### ADD Command Example

```
NCL> ADD NODE decnis HARDWARE SLOT 3 FUNCTIONS DISABLED -
_NCL>{MOP LOAD REQUESTER, MOP DUMP REQUESTER, -
_NCL> IP LOAD REQUESTER, IP DUMP REQUESTER}
```

**Result**: This prevents lines associated with the card in slot 3 of the backplane from being used to load and dump the DECNIS.

### 7.9.4 REMOVE Command

Use this command to remove loading and/or dumping from the functions that are disabled for the hardware slot.

#### REMOVE Command Example

```
NCL> REMOVE NODE decnis HARDWARE SLOT 3 FUNCTIONS DISABLED -
_NCL> {MOP DUMP REQUESTER}
```

**Result**: This now allows the lines associated with the card in slot 3 of the backplane to use MOP for sending upline dumps. They still cannot use BOOTP /TFTP for dumping, and they cannot load using either MOP or BOOTP/TFTP.

### 7.9.5 SHOW Command

Use this command to see what functions (if any) are disabled for a particular hardware slot.

#### SHOW Command Example

```
NCL> SHOW NODE decnis HARDWARE SLOT 3 FUNCTIONS DISABLED
```

#### Clearing Nonvolatile Memory

You can clear the nonvolatile memory of the DECNIS by powering up the hardware with the dump button held in: see the *Installation and Service Manual* for your hardware.

## 7.10 Moving a DECNIS

If your DECNIS is moved to a new site, or a Management Processor Card is moved to a different DECNIS, you must hold the dump button in when you power up the DECNIS, to clear the contents of nonvolatile memory. This will force the DECNIS to load from a load host.

For more information, refer to the *Installation and Service Manual* for your hardware unit.

## 7.11 How the DECNIS Loads Its Software

When the DECNIS hardware unit is powered up, it does the following:

1. Runs diagnostic tests on the DECNIS system.
2. Checks whether there is a valid image in DECNIS nonvolatile memory. (If you have an MPC-III card it checks the Boot Area of nonvolatile memory)
3. If there is no valid image, sends out a request to downline load the software, as described in steps 12 to 14.
4. If there is a valid image, checks its load instructions to see if it should load that image, or if it should load an image from a load host.
5. If the load instructions are to load from a load host, sends out a request to downline load the software, as described in steps 12 to 14.
6. If the load instructions are to load the image from nonvolatile memory, it does so.
7. Checks whether there is a valid CMIP file in nonvolatile memory.
8. If there is no valid CMIP file, loads the CMIP file from a load host.
9. Checks to see whether the profile files are required and, if so, are present.
10. If there are no profile files, loads them from the load host. The load is then complete.
11. If the profile files are present, the load is then complete.
12. When requesting a load from a load host, the DECNIS issues the request on the connection over which it was last loaded, using the protocol with which it was last loaded.
13. If there is no response, or if it has not been previously loaded, it broadcasts MOP and BOOTP load requests on all available connections.
14. One of the load hosts (either BOOTP or MOP) responds to this request:
    - If a MOP load host is the first to respond, the DECNIS sends it a load request. The MOP load host then loads the required files.
    - If a BOOTP load host is the first to respond, it sends the DECNIS a BOOTP message, containing the IP address of the relevant DECNIS interface, and the name and location of the load files. The DECNIS then sends it a TFTP load request, and the BOOTP load host loads the required files.

## 7.12 How the DECNIS Dumps Its Software

### 7.12.1 Load Hosts and Dumping

Each load host you configure using the load-host configurator or the clearVISN DECNIS configurator can also act as a dump sink.

### 7.12.2 How Dumping Works

When you press the dump button on the DECNIS hardware unit, the DECNIS sends a request for a load host to accept a dump of its memory:

1. It issues a request for a host system to accept the dump on the connection over which it was last dumped.

   If there is no response, or if it has not previously dumped, it broadcasts the request on all available connections.

2. One of the load hosts (MOP or BOOTP) responds by sending a message to the DECNIS. The DECNIS then sends the dump to this load host.

# 8

# Nonvolatile (Flash) Memory Loading

## 8.1 Introduction

This chapter describes how to set up a DECNIS to load from its own nonvolatile (flash) memory.

### 8.1.1 Methods for Setting Up Flash Memory Loading

The recommended method is to do either of the following:

- On OpenVMS, Digital UNIX, or non-Windows MS–DOS PC hosts, use the load-host configurator.
- On Windows NC/95 hosts, use the clearVISN DECNIS configurator.

However, you may need to set up nonvolatile loading using NCL commands. Section 8.5 describes how to do this.

### 8.1.2 How Flash Memory Loading Works

The term nonvolatile, or flash, memory refers to an area of DECNIS memory which can be used to store:

- The DECNIS software image only.
- A combined file containing the DECNIS software image, configuration file(s) and any profile files.
- The DECNIS software image and one or more separate configuration files.

#### When the DECNIS Loads From Flash Memory

The DECNIS initially loads its image and configuration files from a load host. After that, it will reload from flash memory if the following apply:

- The following NCL command is in the DECNIS configuration file.

  `SET NODE` *`decnis`* `HARDWARE DEBUG FLAGS 0`

  The DECNIS configurator automatically inserts this command in the DECNIS NCL script if you select nonvolatile or flash memory loading.

- The required file or files have been loaded into flash memory.

  The software image is always loaded into flash memory when you initially load the DECNIS from a host.

  On the next load, if there is only an image in flash memory, the DECNIS will load it, and load the CMIP/profile files from the load host. If the CMIP /profile files are in flash, the DECNIS will load them as well.

  The CMIP/profile files can be placed in flash memory in two ways:

  - By creating and loading a combined file.
  - By inserting individual CMIP files into flash memory dynamically, as described in the *DECNIS Management* manual.

However, note that there are some circumstances in which the DECNIS automatically loads from a load host; see Section 8.6 for details.

## 8.2 Setting Up Flash Memory Loading in the Load-Host Configurator

Follow these steps:

1. Run the load-host configurator. On the `Type of Loading` screen, select one of the nonvolatile memory loading options:

   ```
   Nonvolatile memory for both CMIP and image
   Load host for CMIP; nonvolatile memory for image
   Load host for both CMIP and image
   ```

2. When you have finished configuring the DECNIS for loading, exit from the load-host configurator.
3. Run the DECNIS text-based configurator, as described in Chapter 2 to Chapter 4.
4. In the `Create NCL Script` section, create a combined file (or CMIP file).
5. Load the DECNIS, as described in Chapter 7.

If you have not created a combined file, you will have to insert a CMIP file into flash memory dynamically; refer to Section 8.4.

### 8.2.1 Results of Choosing

Table 8–1 shows what happens when you choose each type of loading.

**Table 8–1 Types of Loading**

| Option | Results |
|---|---|
| Nonvolatile memory for both CMIP and image | 1. The image, the CMIP file and any profile files are combined into a single file.<br>2. The next time the DECNIS is rebooted, it loads the combined file from the load host.<br>3. On subsequent reboots, the DECNIS loads the combined file from its nonvolatile memory. |
| Load host for CMIP; nonvolatile memory for image | 1. The next time the DECNIS is rebooted, it loads the image and the CMIP and profile files from the load host.<br>2. On subsequent reboots, the DECNIS loads the image from its nonvolatile memory and the CMIP and profile files from a load host. |
| Load host for both CMIP and image | 1. The next time the DECNIS is rebooted, it loads the image and the CMIP and profile files from a load host.<br>2. On subsequent reboots, the DECNIS reloads the image, the CMIP file and the profile files from a load host. |

## 8.3 Setting Up Flash Memory Loading in the clearVISN DECNIS Configurator

If you have configured a DECNIS in the clearVISN DECNIS configurator, and you want to set up flash memory loading, follow these steps:

1. On the Main Navigation window, click the System button.
2. On the Load Options tab page, do the following:
   - Under Image Loads, select **From Flash**.
   - If you want to create a combined file, check the **CMIP Script** box Under Flash Contents.
3. Click OK.

If you have not created a combined file, you will have to insert a CMIP file into flash memory dynamically; refer to Section 8.4.

## 8.4 Modifying Flash Memory Dynamically

The rest of this chapter assumes that you wish to load a combined file into flash memory. However, you can, if you wish, use DECNIS console commands to add one or more separate CMIP files to flash memory dynamically, and designate the one to be loaded.

Once you have done this, the DECNIS will load the designated CMIP file from flash memory, provided that you have specified nonvolatile loading, as follows:

- In the load-host configurator, select **either** `Nonvolatile memory for both CMIP and image` or `Load host for CMIP; nonvolatile memory for image`.
- In the clearVISN DECNIS configurator, select **From Flash** under **Image Loads**.

Refer to the *DECNIS Management* manual for detailed instructions on modifying flash memory dynamically.

## 8.5 Setting Up Nonvolatile Memory Loading Using Commands

Section 8.5.1 to Section 8.5.3 describe how to set up flash loading without using the configurator. Note that you can only do this on OpenVMS, MS–DOS PC and Digital UNIX load hosts. On Windows 95/NT hosts, you can only set up loading information using the clearVISN DECNIS configurator.

This section assumes that you have already created a CMIP file, either in the DECNIS text-based configurator or as described in Section 5.5.1.

### 8.5.1 Method for Setting Up Flash Memory Loading

To set up the DECNIS for flash memory loading, follow these steps:

1. Create a combined file, as described in Section 8.2, Section 8.3 or Section 8.5.2.
2. Issue the command to load from the load host, as described in Section 8.5.3. This is required to load the new combined file to the DECNIS.

### 8.5.2 Creating a Combined File

To combine the software image, CMIP file and profile files into a single combined file, enter the following command:

- PC load hosts:

```
C:\install-directory\NIS\NIS_ICMP NIS040 client-name
```

- OpenVMS load hosts:

  ```
  $ @SYS$MANAGER:NIS$COMBINE.COM NIS040 client-name
  ```

- Digital UNIX load hosts:

  ```
  # /usr/lib/dnet/nis_combine nis040 client-name
  ```

where: *install-directory* is the installation directory.

*client-name* is the load client name of the DECNIS.

### 8.5.3 Issuing the Command to Load from a Load Host

Issue the following NCL comand to tell the DECNIS to load from a load host:

- PC load hosts:

  ```
  NCL> SET NODE decnis/username/password HARDWARE DEBUG FLAGS 1073741952
  ```

- OpenVMS load hosts:

  ```
  NCL> SET NODE decnis"username password" HARDWARE DEBUG FLAGS 1073741952
  ```

- Digital UNIX load hosts:

  ```
  ncl> SET NODE decnis/username/password HARDWARE DEBUG FLAGS 1073741952
  ```

where: *decnis* is the node name of the DECNIS.

*username* and *password* are the user name and password required to use NCL commands to manage the DECNIS.

#### Issuing the Network Loading Console Command

If the DECNIS has previously been loaded, you can use the DECNIS console to reload it from a load host. Issue the following console command:

```
console> load -flash
```

### 8.5.4 Example: Setting Up Flash Memory Loading

#### 8.5.4.1 Before You Begin

In this example, you have already done the following:

- Run the load-host configurator and selected `Load host for both CMIP and image`.
- Run the DECNIS text-based configurator and created an NCL script.

You have not created a CMIP file.

#### 8.5.4.2 Available Information

In this example, the following information is available:

| | |
|---|---|
| Type of load host | Digital UNIX MOP |
| Load client name of the DECNIS | load_decnis1 |
| Node name of the DECNIS | paris_decnis1 |
| Username | Rosencrantz |
| Password | Guildenstern |

#### 8.5.4.3 Procedure

To set up paris_decnis1 for flash memory loading, follow these steps:

1. Create a CMIP file by entering the command:

   ```
   # /usr/lib/dnet/nis_script_compile nis_load_decnis1
   ```

2. Create a combined file by entering the command:

   ```
   # /usr/lib/dnet/nis_combine nis040 nis_load_decnis1
   ```

3. Instruct the DECNIS to load from a load host by entering the command:

   ```
   NCL> SET NODE paris_decnis1/rosencrantz/guildenstern -
   _NCL> HARDWARE DEBUG FLAGS 1073741952
   ```

4. Reload the DECNIS. The new combined file will be loaded from the load host.

5. The next time you reboot the DECNIS, it will load from flash memory. To check that it has loaded successfully, enter the following command:

   ```
   NCL> SHOW NODE paris_decnis1/rosencrantz/guildenstern LAST REBOOT REASON
   ```

   The reason displayed should be either **Flash updated successfully** or code 12. If there is any other reason, refer to the online *DECNIS Problem Solving* manual.

## 8.6 Loading a New Image or Configuration File

In order to load new software versions and configuration files, you need to reload the DECNIS from a load host. The configurators automatically force the next load to be from the load host in the following circumstances:

- If you use the load-host configurator Update option to update one or more DECNIS systems to Version 2.1 or higher.
- If you use the automatic Update procedure to update all DECNIS systems.
- If you use the load-host configurator Add option to set up a new DECNIS.

- If you create a new CMIP file, or a new combined file, within the DECNIS configurator.

## 8.7 Forcing the DECNIS to Load from the Load Host

You may sometimes need to force the DECNIS to load from a load host, without using the configurator. Section 8.7.1 lists the circumstances in which you will need to do this.

### 8.7.1 When to Force a Load from the Load Host

You will need to force the DECNIS to load from the load host if:

- You have created a new combined file from the command line, as described in Section 8.5.2.
- The system on which you are configuring is on a different network from the load host from which you are loading.
- The HARDWARE DEBUG FLAGS command issued by the DECNIS configurator fails. This will happen if either of the following are true:
  - The DECNIS is not connected to the network and powered up.
  - The DECNIS is not reachable from the system on which the command is issued.

### 8.7.2 Methods of Forcing a Load from the Load Host

Refer to Section 7.4.2 and Section 7.4.3.

## 8.8 Version 7-07 ROMs and Flash Memory Loading

If the DECNIS attempts to load from a load host, and no load has been obtained within one hour, what happens next will depend on the version of Management Processor ROMs fitted to the DECNIS.

- With ROM versions earlier than V7-07, the DECNIS will reset and run the self-test after one hour. It will not attempt to load from flash memory.
- With ROM versions V7-07 or later, the DECNIS will attempt to load from flash memory after one hour. If there is a software image in flash memory, it will be loaded.

  On future loads, the DECNIS will again attempt to load from the load host.

### 8.8.1 Finding the ROM Version Used by the DECNIS

To check what version of Management Processor ROM your DECNIS is using, enter the following command:

```
NCL> SHOW NODE decnis DEVICE UNIT MP* FIRMWARE ID
```

The screen will display the firmware identifier, which includes the version number. For example:

```
  Status
   Firmware Identifier = "0 2-3.6 V7-04 2.7"
```

In this example, the version number is V7-04.

# 9

# Using the DECNIS as a Proxy Load Host

## 9.1 Introduction

This chapter describes how to set up the DECNIS as a proxy load host.

### 9.1.1 Definition of Proxy Load Host

A **proxy load host** is a system that can load another system or receive dumps from it, but which does not itself store the load or dump files. Instead, the load and dump files are stored on a **real load host**. They are sent to and from the proxy load host as required.

### 9.1.2 Using the DECNIS as a Proxy Load Host

When acting as a proxy load host, the DECNIS can get the load files from the real load host in either of two different ways:

- Using the Data Access Protocol (DAP). This is a DECnet protocol used for file access and transfer.
- Using the TFTP protocol. This is an IP protocol used for loading and dumping.

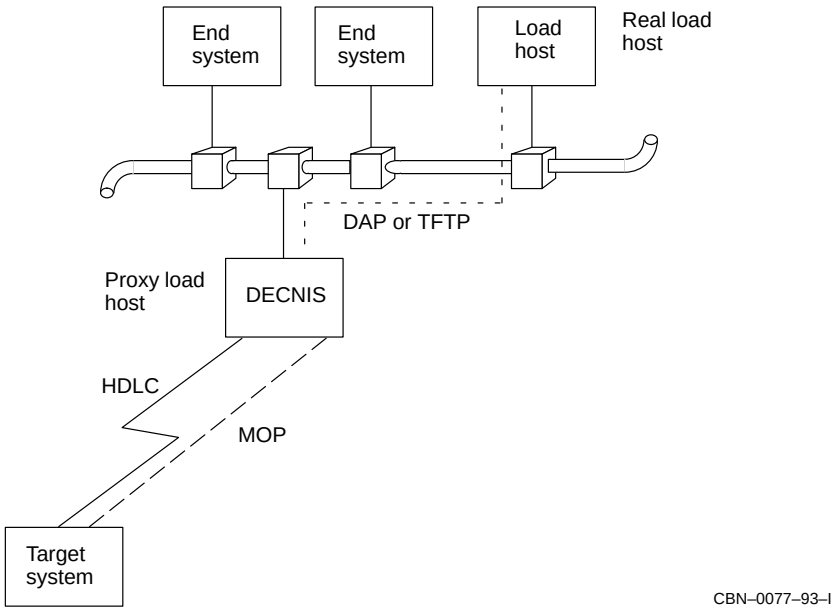
Once the DECNIS has obtained the files, it uses MOP over an HDLC or PPP link to load the target systems.

### 9.1.3 Example: Proxy Load Host

Figure 9–1 shows the DECNIS acting as a proxy load host.

The DECNIS obtains the load files from the real load host and downline loads them to the target system.

**Figure 9–1 DECNIS Acting as a Proxy Load Host**



## 9.2 Setting Up Proxy Loading: DECnet and MOP

This section describes how to set up the DECNIS so that it can do the following:

- Receive files from a real load host, using DECnet, and downline load them to other systems, using MOP.
- Receive dumps from other systems, using MOP, send them to the real load host, using DECnet.

### 9.2.1 Requirements for the Real Load Host

The real load host, which stores the files to be downline loaded, must meet the following requirements:

- It must be a DECnet system that supports DAP.
- It must be reachable from the proxy load host, using DECnet protocols.

### 9.2.2 Requirements for the DECNIS Proxy Load Host

#### 9.2.2.1 Supported Data Link

The DECNIS proxy load host can use the following types of data link to connect to the target systems:

- HDLC
- PPP

#### 9.2.2.2 Configuration: Proxy Load Host

Before you set up a DECNIS as a proxy load host, do the following:

1. Run the load-host configurator and add load information for the DECNIS proxy load host.
2. Run the DECNIS text-based configurator and configure the DECNIS proxy load host. You should:
   - In the `Network Interface Cards` section, set up a WANcontroller card, for example, a W618 card.
   - In the `Lines` section, set up an HDLC or PPP circuit on one of the lines on the WANcontroller card.
   - In the `Create NCL Script` section, create a CMIP file or a combined file.

### 9.2.3 Requirements for Target Systems

If the target system is a DECNIS, configure it, following the steps in Section 9.2.2.2.

If the target system is not a DECNIS, configure it and generate a CMIP or combined file as described in the documentation for the target system.

If you configure the target system on a system other than the real load host, copy the load files to the real load host.

### 9.2.4 Information Required

You will need to know the following:

- The node name of the real load host. This is the same as the name of the KNOWN TOWER entity for the load host. Note: The node name must have the correct format for a full DECdns node name, even if you are not using DECdns on your network.

  Refer to the *DECNIS Management* manual for a full explanation of how to create the KNOWN TOWER entity.

- The location on the real load host of the files to be loaded to the target system: either the combined file or the software image and CMIP and profile files.

Refer to Section B.2 for the standard location of load files on various load hosts.

### 9.2.5 Procedure

Follow these steps to set up the DECNIS as a proxy load host:

1. If necessary, create a MOP circuit that uses the HDLC or PPP data link.

   The DECNIS text-based configurator sets up a MOP circuit for each HDLC data link. The MOP circuit has the same name as the HDLC line. For example, if the line name is W618-4-2, the MOP circuit name is also W618-4-2.

   The DECNIS text-based configurator does not set up a MOP circuit for PPP data links. To set up a MOP circuit on a PPP link, enter the following NCL commands, or insert them into the user NCL scripts:

   ```
   NCL> CREATE NODE decnis MOP CIRCUIT circuit-name TYPE HDLC
   NCL> SET NODE decnis MOP CIRCUIT circuit-name -
   _NCL> LINK NAME PPP LINK link-name
   ```

   Note that the MOP circuit is of type HDLC.

2. Enable the LOAD SERVER and DUMP SERVER functions on the MOP circuit:

```
NCL> ENABLE NODE decnis MOP CIRCUIT circuit-name FUNCTION -
_NCL> {LOAD SERVER, DUMP SERVER}
```

   This command will not affect any other functions currently enabled on the MOP circuit.

3. Create a MOP client to represent the target system:

```
NCL> CREATE NODE decnis MOP CLIENT client-name
```

   where: *decnis* is the node name of the DECNIS proxy load host.
   *client-name* is the MOP client name to identify the target system.

   Note that MOP client names are also referred to as load client names in this manual.

4. Specify the MOP circuit to be used to load to the target system (the MOP client name identifies the target system).

```
NCL> SET NODE decnis MOP CLIENT client-name -
_NCL>  CIRCUIT circuit-name
```

5. Create an entry for the real load host in the Known Towers database:

```
NCL> CREATE NODE decnis SESSION CONTROL -
_NCL> KNOWN TOWER host TOWERS {(tower1), (tower2), ...}
```

   where: *host* is the node name of the real load host.
   *tower1* and *tower2* are towers that describe the protocols and NSAP addresses used to communicate with the remote node.

6. Specify the image and load files to be loaded to the target system:

   - If the target system will use nonvolatile memory loading, specify the name of the combined file:

```
NCL> SET NODE decnis MOP CLIENT client-name -
_NCL> SYSTEM IMAGE {"host::filespec"}
```

     where: *host* is the node name of the real load host. This is the name of the KNOWN TOWER entity created in step 4.
     *filespec* is the specification of the combined file on the real load host.

   - If the target system will reload from the load host, follow these steps:

     a. Specify the software image to be loaded to the target system:

```
NCL> SET NODE decnis MOP CLIENT client-name -
_NCL> SYSTEM IMAGE {"host::image"}
```

b. Specify the CMIP (script) file to be loaded to the target system:

```
NCL> SET NODE decnis MOP CLIENT client-name -
_NCL> SCRIPT FILE {"host::cmip"}
```

c. Specify the directory containing the profile files to be loaded to the target system. Enter the name of the CMIP script file, as in the previous step:

```
NCL> SET NODE decnis MOP CLIENT client-name MANAGEMENT IMAGE -
_NCL> {"host::cmip"})
```

where: *host* is the node name of the real load host. This is the name of the KNOWN TOWER entity created in step 4.
*image* is the file specification of the image on the real load host.
*cmip* is the specification of the CMIP file on the real load host.

7. Specify the name of the dump file for the target system:

```
NCL> SET NODE decnis MOP CLIENT client-name -
_NCL> DUMP FILE {"host::filespec"}
```

where: *host* is the node name of the real load host.
*filespec* is the specification of the file to which the dump will be written on the real load host.

8. Ensure that the DECNIS has read access to the load files and write access to the directory where the dump files will be written on the real load host. Do this in one of the following ways:

- Set the file protection on the load and dump files on the real load host to allow access by the default DECnet account.
- Set up a proxy account on the real load host, to allow access to these files by *proxy-name*::LES$MOP, where *proxy-name* is the node name of the DECNIS.

These steps will set up the DECNIS to load another DECNIS. However, if you want the DECNIS to load to other types of target system, you may need to specify additional MOP client characteristics for the target system.

#### 9.2.5.1 Enter Commands in the User NCL Script Files

It is recommended that you add the NCL commands in Section 9.2.5 to the appropriate user NCL script files (refer to Section 5.2.2). If you do not do this, you will have to reenter the commands each time the proxy load host is rebooted.

### 9.2.6 Example NCL Commands: DECnet/OSI for OpenVMS Real Load Host

This section gives NCL commands to configure a DECNIS to act as a proxy load host for another DECNIS. You have the following information:

| | |
|---|---|
| Name of DECnet/OSI for OpenVMS real load host | ORG:.NORTH.REAL |
| Name of DECNIS proxy load host | DECNIS_PROXY |
| MOP Client Name of DECNIS target | DECNIS_TARGET |
| Type of loading specified for target | All files from load host |

The commands are:

```
create node DECNIS_PROXY mop circuit hdlc_circ1 type hdlc
set node DECNIS_PROXY mop circuit hdlc_circ1 link name hdlc link -
   hdlc_link1 logical station hdlc_stat1
enable node DECNIS_PROXY mop circuit hdlc_circ1 function
   {load server,dump server}

create node DECNIS_PROXY mop client DECNIS_TARGET
set node DECNIS_PROXY mop client DECNIS_TARGET circuit hdlc_circ1

create node DECNIS_PROXY session control known tower org:.north.real -
   towers {([DNA_CMIP-MICE], [DNA_SESSIONCONTROLV3, NUMBER = 19], -
   [DNA_NSP], [DNA_OSINETWORK, 37:12345:02-00:08-2B-14-78-66-11:20])}

set node DECNIS_PROXY mop client DECNIS_TARGET system image
   {"org:.north.real::mom$system:nis040.sys"}
set node DECNIS_PROXY mop client DECNIS_TARGET script file
   {"org:.north.real::mom$system:nis_DECNIS_TARGET.cmip"}
set node DECNIS_PROXY mop client DECNIS_TARGET management image
   {"org:.north.real::mom$system:nis_DECNIS_TARGET.cmip"}
set node DECNIS_PROXY mop client DECNIS_TARGET dump file
   {"org:.north.real::mom$system:nis_DECNIS_TARGET.dmp"}
```

### 9.2.7 Example NCL Commands: DECnet/OSI for Digital UNIX Real Load Host

This section gives NCL commands to configure a DECNIS to act as a proxy load host for another DECNIS. You have the following information:

| | |
|---|---|
| Name of DECnet/OSI for Digital UNIX real load host | ORG:.NORTH.REAL |
| Name of DECNIS proxy load host | DECNIS_PROXY |
| MOP Client Name of DECNIS target | DECNIS_TARGET |
| Type of loading specified for target | All files from load host |

The commands are:

```
create node DECNIS_PROXY mop circuit hdlc_circ1 type hdlc
set node DECNIS_PROXY mop circuit hdlc_circ1 link name hdlc link -
   hdlc_link1 logical station hdlc_stat1
enable node DECNIS_PROXY mop circuit hdlc_circ1 function
   {load server,dump server}

create node DECNIS_PROXY mop client DECNIS_TARGET
set node DECNIS_PROXY mop client DECNIS_TARGET circuit hdlc_circ1

create node DECNIS_PROXY session control known tower ORG:.NORTH.REAL -
   towers {([DNA_CMIP-MICE], [DNA_SESSIONCONTROLV3, NUMBER = 19], -
   [DNA_NSP], [DNA_OSINETWORK, 37:12345:02-00:08-2B-14-78-66-11:20])}

set node DECNIS_PROXY mop client DECNIS_TARGET system image
   {"org:.north.real::/usr/lib/mop/nis040.sys"}
set node DECNIS_PROXY mop client DECNIS_TARGET script file
   {"org:.north.real::/usr/lib/mop/nis_DECNIS_TARGET.cmip"}
set node DECNIS_PROXY mop client DECNIS_TARGET management image
   {"org:.north.real::/usr/lib/mop/nis_DECNIS_TARGET.cmip"}
set node DECNIS_PROXY mop client DECNIS_TARGET dump file
   {"org:.north.real::/usr/lib/mop/nis_DECNIS_TARGET.dmp"}
```

## 9.3 Setting Up Proxy Loading: TFTP and MOP

This section describes how to set up the DECNIS so that it can:

- Receive files from a real load host using TFTP and downline load them to other systems, using MOP.
- Receive dumps from other systems using MOP and send them to the real load host, using TFTP.

### 9.3.1 Requirements for the Real Load Host

The real load host, which stores the files to be downline loaded, must meet the following requirements:

- It must be running the TFTP daemon.
- It must be reachable from the proxy load host.

### 9.3.2 Requirements for the DECNIS Proxy Load Host

The requirements for the DECNIS proxy load host are as described in Section 9.2.2.1 and Section 9.2.2.2.

### 9.3.3 Requirements for Target Systems

If the target system is a DECNIS, configure it, following the steps in Section 9.2.2.2.

If the target system is not a DECNIS, configure it and generate a CMIP or combined file as described in the documentation for the target system.

If you configure the target system on a system other than the real load host, copy the load files to the real load host.

### 9.3.4 Information Required

You will need to know the following:

- The IP address of the real load host.
- The location on the real load host of the files to be loaded to the target system: either the combined file or the software image and CMIP and profile files.

  Refer to Section B.2 for the standard location of load files on various load hosts.

### 9.3.5 Procedure

Follow these steps to set up the DECNIS as a proxy load host:

1. If necessary, create a MOP circuit that uses the HDLC or PPP data link.

   The DECNIS text-based configurator sets up a MOP circuit for each HDLC data link. The MOP circuit has the same name as the HDLC line. For example, if the line name is W618-4-2, the MOP circuit name is also W618-4-2.

   The DECNIS text-based configurator does not set up a MOP circuit for each PPP data link. See Section 9.2.5, step 1 for the NCL commands required to set up a MOP circuit on a PPP link.

2. Enable the LOAD SERVER and DUMP SERVER functions on the MOP circuit:

   ```
   NCL> ENABLE NODE decnis MOP CIRCUIT circuit-name FUNCTION -
   _NCL> {LOAD SERVER, DUMP SERVER}
   ```

   This command will not affect any other functions currently enabled on the MOP circuit.

3. Create a MOP client to represent the target system:

   ```
   NCL> CREATE NODE decnis MOP CLIENT client-name
   ```

   where: *decnis* is the node name of the DECNIS proxy load host.
   *client-name* is a MOP client name to identify the target system.

4. Specify the MOP circuit to be used to load to the target system (the MOP client name identifies the target system):

   ```
   NCL> SET NODE decnis MOP CLIENT client-name -
   _NCL>  CIRCUIT circuit-name
   ```

5. Specify the image and load files to be loaded to the target system:

   - If the target system will use nonvolatile memory loading, specify the name of the combined file:

     ```
     NCL> SET NODE decnis MOP CLIENT client-name -
     _NCL> SYSTEM IMAGE {"host-ip-address:filespec"}
     ```

     where: *host-ip-address* is the host IP address of the real load host.
     *filespec* is the specification of the combined file on the real load host.

- If the target system will reload from the load host, follow these steps:

  a. Specify the software image to be loaded to the target system:

  ```
  NCL> SET NODE decnis MOP CLIENT client-name -
  _NCL> SYSTEM IMAGE {"host-IP-address:image"}
  ```

  b. Specify the CMIP (script) file to be loaded to the target system:

  ```
  NCL> SET NODE decnis MOP CLIENT client-name -
  _NCL> SCRIPT FILE {"host-IP-address:cmip"}
  ```

  c. Specify the directory containing the profile files to be loaded to the target system. Enter the name of the CMIP script file, as in the previous step:

  ```
  NCL> SET NODE decnis MOP CLIENT client-name MANAGEMENT IMAGE -
  _NCL> {"host-IP-address:cmip"})
  ```

  where: *host-ip-address* is the host IP address of the real load host.
  *image* is the file specification of the image on the real load host.
  *cmip* is the specification of the CMIP file on the real load host.

6. Specify the name of the dump file for the target system:

```
NCL> SET NODE decnis MOP CLIENT client-name -
_NCL> DUMP FILE {"host-ip-address:filespec"}
```

where: *host* is the host IP address of the real load host.
*filespec* is the specification of the file to which the dump will be written on the real load host.

These steps will set up the DECNIS to load another DECNIS. However, if you want the DECNIS to load to other types of target system, you may need to specify additional MOP client characteristics for the target system.

#### 9.3.5.1 Enter Commands in the User NCL Script Files

It is recommended that you add the NCL commands in Section 9.3.5 to the appropriate user NCL script files (refer to Section 5.2.2). If you do not do this, you will have to reenter the commands each time the proxy load host is rebooted.

### 9.3.6 Example NCL Commands: PC Real Load Host

This section gives NCL commands to configure a DECNIS to act as a proxy load host for another DECNIS. You have the following information:

| | |
|---|---|
| IP address of PC real load host | 23.24.32.78 |
| Node name of DECNIS proxy load host | NISPRO |
| MOP client name of DECNIS target | TARG1 |
| Type of loading specified for target | All files from load host |

The commands are:

```
create node NISPRO mop circuit hdlc_circ1 type hdlc
set node NISPRO mop circuit hdlc_circ1 link name hdlc link -
   hdlc_link1 logical station hdlc_stat1
enable node NISPRO mop circuit hdlc_circ1 function
   {load server,dump server}

create node NISPRO mop client TARG1
set node NISPRO mop client TARG1 circuit hdlc_circ1

set node NISPRO mop client TARG1 system image
   {"23.24.32.78:c:\decrou\common\nisv40\system"}
set node NISPRO mop client TARG1 script file
   {"23.24.32.78:c:\decrou\clients\TARG1\script"}
set node NISPRO mop client TARG1 management image
   {"23.24.32.78:c:\decrou\clients\TARG1\script"}
set node NISPRO mop client TARG1 dump file
   {"23.24.32.78:c:\decrou\clients\TARG1\dump"}
```

### 9.3.7 Example NCL Commands: Digital UNIX Real Load Host

This section gives NCL commands to configure a DECNIS to act as a proxy load host for another DECNIS. You have the following information:

| | |
|---|---|
| IP address of Digital UNIX real load host | 23.24.32.78 |
| Name of DECNIS proxy load host | DECNIS_PROXY |
| MOP client name of DECNIS target | DECNIS_TARGET |
| Type of loading specified for target | All files from load host |

The commands are:

```
create node DECNIS_PROXY mop circuit hdlc_circ1 type hdlc
set node DECNIS_PROXY mop circuit hdlc_circ1 link name hdlc link -
   hdlc_link1 logical station hdlc_stat1
enable node DECNIS_PROXY mop circuit hdlc_circ1 function
   {load server,dump server}

create node DECNIS_PROXY mop client DECNIS_TARGET
set node DECNIS_PROXY mop client DECNIS_TARGET circuit hdlc_circ1

set node DECNIS_PROXY mop client DECNIS_TARGET system image
   {"23.24.32.78:/usr/lib/mop/nis040.sys"}
set node DECNIS_PROXY mop client DECNIS_TARGET script file
   {"23.24.32.78:/usr/lib/mop/nis_DECNIS_TARGET.cmip"}

set node DECNIS_PROXY mop client DECNIS_TARGET management image
   {"23.24.32.78:/usr/lib/mop/nis_DECNIS_TARGET.cmip"}

set node DECNIS_PROXY mop client DECNIS_TARGET dump file
   {"23.24.32.78:/usr/lib/mop/nis_DECNIS_TARGET.dmp"}
```

# 10

# Using the DECNIS as a BOOTP Gateway

## 10.1 Introduction

This chapter describes how to set up the DECNIS as a BOOTP gateway.

### 10.1.1 Definition of BOOTP Gateway

A BOOTP gateway can do the following:

- Downline load files located on **BOOTP servers** to **BOOTP clients**,
- Receive dumps from BOOTP clients and relay them to BOOTP servers.

These functions are known as BOOTP relay functions.

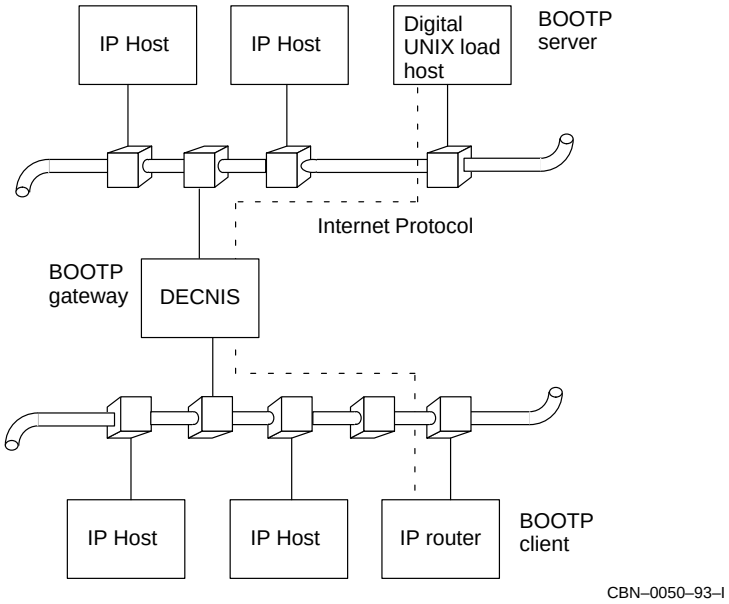
The BOOTP server is the real load host, and stores the load and dump files. The BOOTP gateway does not store load and dump files; rather, it relays BOOTP load and dump requests from the BOOTP client to the BOOTP server, and BOOTP responses from the BOOTP server to the BOOTP client.

A DECNIS BOOTP gateway can send load requests directly to one or more BOOTP servers, or can forward them to another BOOTP gateway.

### 10.1.2 Example: BOOTP Relay

Figure 10–1 shows a DECNIS BOOTP gateway. The DECNIS BOOTP gateway obtains the load files from the BOOTP server, and downline loads them to the BOOTP client.

**Figure 10–1 DECNIS Acting as a BOOTP Gateway**




## 10.2 Setting Up the DECNIS as a BOOTP Gateway

This section describes how to set up a DECNIS so that it can load BOOTP clients, and receive dumps from them.

### 10.2.1 Requirements for BOOTP Servers

You need to specify the BOOTP servers to which the DECNIS will forward BOOTP messages. You can specify multiple BOOTP servers on each routing circuit configured for BOOTP relay.

#### 10.2.1.1 Type of System

A BOOTP server can be any of the following:

- An MS–DOS PC, with BOOTP/TFTP loading set up, as described in *DECNIS Configuration and Management from MS–DOS PCs*.
- A Windows NT or Windows 95 PC, with the Microsoft TCP/IP network protocol stack installed (included in Windows NT/95).

- A Digital UNIX host, configured for BOOTP/TFTP, as described in the manual *DECNIS Installation and Configuration for OpenVMS and Digital UNIX*.
- A non-Digital UNIX® load host configured for BOOTP, as described in the manual *DECNIS Installation and Configuration for OpenVMS and Digital UNIX*.

#### 10.2.1.2 BOOTP Load File Locations

On a Digital UNIX BOOTP server, the directory and file names for the load files must be set up so that the BOOTP server can respond correctly to load and dump requests. For more information, refer to Section B.3.

You can set up load information for all BOOTP clients, and locate the files correctly, by running one of the following on the BOOTP server:

- The load-host configurator, if the BOOTP server is an MS–DOS PC or a Digital UNIX system).
- The clearVISN DECNIS configurator, if the BOOTP server is a Windows NT or Windows 95 system.

### 10.2.2 Requirements for BOOTP Gateways

#### 10.2.2.1 Type of Data Link

The DECNIS can perform BOOTP gateway functions on all types of circuits except X25 DA.

#### 10.2.2.2 Configuration of BOOTP Gateway on MS–DOS and Digital UNIX systems

Before you enter the NCL commands to set up DECNIS as a BOOTP gateway, you make sure that you have done the following in the DECNIS text-based configurator:

1. Selected Yes for IP routing in the Configuration Options section,
2. In the Lines section, set up the routing circuit(s) to be used for BOOTP gateway functions.

#### 10.2.2.3 Configuration of BOOTP Gateway on Windows NT/95 PCs

Before you enter the NCL commands to set up DECNIS as a BOOTP gateway, you make sure that you have done the following in the clearVISN DECNIS configurator:

1. Tick the IP box on the General tab page, under the **System** button.
2. On the Circuits tab page in the IP routing section, configure the routing circuit(s) to be used for BOOTP gateway functions.

If you are using a non-Digital UNIX system as a BOOTP server, you will have to copy the required files to it, as described in the manual *DECNIS Installation and Configuration for OpenVMS and Digital UNIX*.

## 10.2.3 Requirements for BOOTP Clients

### 10.2.3.1 Configuration: MS–DOS PCs or Digital UNIX Systems

If the BOOTP client is a DECNIS, do the following:

1. Run the load-host configurator and add load information. Select either `BOOTP` or `MOP or BOOTP` as the method to be used for loading.
2. Run the DECNIS text-based configurator. The requirements are the same as in Section 10.2.2.2.

### 10.2.3.2 Configuration: Windows NT or Windows 95 Systems

If the BOOTP client is a DECNIS, it needs to meet the requirements described in Section 10.2.2.3.

### 10.2.3.3 Configuration: Other Systems

If the BOOTP client is a type of system other than those in Section 10.2.3.1 and Section 10.2.3.2, configure it and generate a CMIP file as described in the documentation for the BOOTP client.

## 10.2.4 Information Required for BOOTP Relay Configuration

You will need to know the following:

- The IP address of the BOOTP server(s).
- The names of the routing circuits to be used for BOOTP relay functions on the BOOTP gateways.

## 10.2.5 Procedure

Follow these steps to set up DECNIS systems as BOOTP gateways:

1. Check that the BOOTP server or servers are configured so that they can reach the BOOTP gateway and clients.
2. If the routing circuit(s) to be used for BOOTP are enabled, disable them. For example:

   ```
   NCL> DISABLE ROUTING CIRCUIT circuit-name
   ```

3. On each DECNIS BOOTP gateway, specify the circuit(s) to be used for BOOTP functions; that is, to receive BOOTP requests and carry out the load. This circuit must be connected to the same LAN to which the BOOTP client is connected.

```
ncl> SET NODE decnis ROUTING CIRCUIT circuit-name - )
_ncl>  BOOTP SERVERS {a.a.a.a.,b.b.b.b}
```

where: *decnis* is the node name of the DECNIS BOOTP gateway.
*circuit-name* is the name of the routing circuit.
*a.a.a.a,b.b.b.b* are IP addresses of BOOTP servers, or of other BOOTP gateways through which the DECNIS is obtaining the files. You can specify up to tbs IP addresses for each circuit using BOOTP.

#### 10.2.5.1 Enter Commands in the User NCL Script Files

It is recommended that you add the NCL command in Section 10.2.5 to the SET user NCL script file (refer to Section 5.2.2). If you do not do this, you will have to reenter the commands each time the DECNIS BOOTP gateway is rebooted.

### 10.2.6 Example

Figure 10–2 shows systems to be set up for BOOTP relay.

#### 10.2.6.1 Available Information

In this example, the following information is available:

| System | Node Name | IP Address | Routing Circuit |
|---|---|---|---|
| BOOTP gateways | nis_peach | | L602-3-1 |
| | nis_mango | | L602-6-0 |
| BOOTP server | host north | 28.34.26.10 | |

#### 10.2.6.2 Procedure

You wish to configure nis_peach and nis_mango as BOOTP gateways so that they can load BOOTP clients from the BOOTP server host_north. Follow these steps:

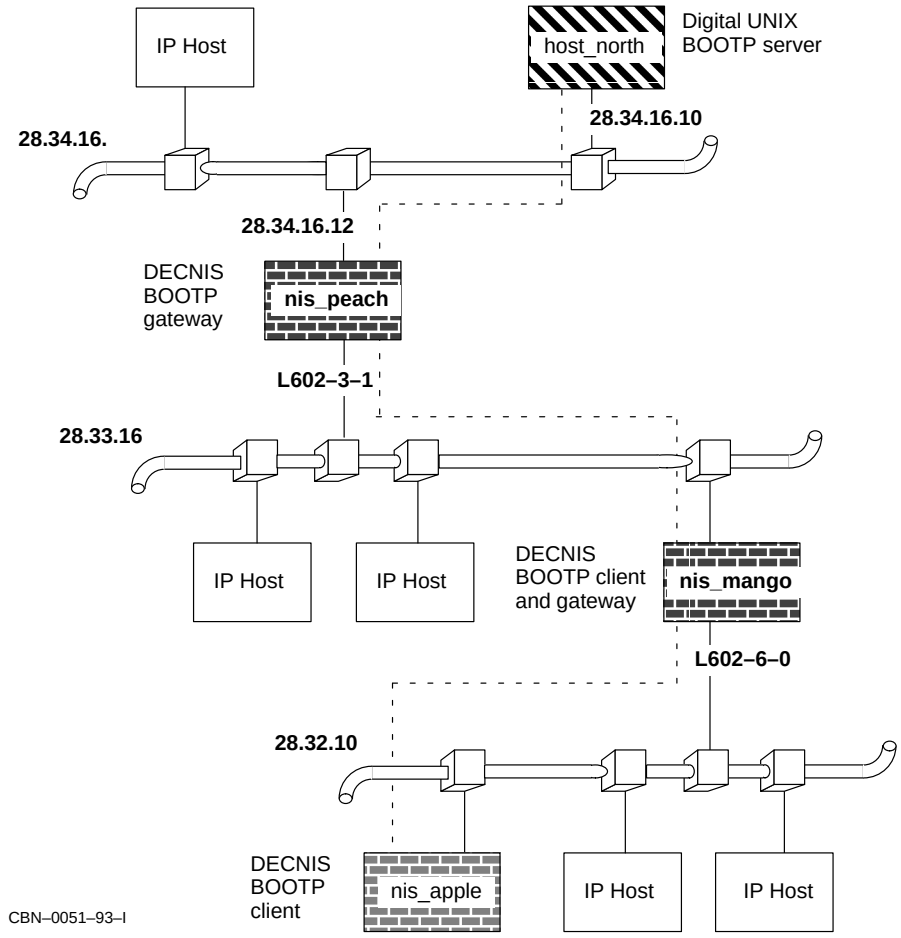
1. Disable the routing circuit that nis_peach will use (L601-3-1) to forward load requests to the BOOTP server (IP address 28.34.26.10), and downline load files from the BOOTP server.

   ```
   ncl> DISABLE NODE nis_peach ROUTING CIRCUIT L601-3-1
   ```

2. Specify the routing circuit L601-3-1 as a BOOTP RELAY circuit:

   ```
   ncl> SET NODE nis_peach ROUTING CIRCUIT L601-3-1 -
   _ncl> BOOTP RELAY 28.34.16.10
   ```

**Figure 10–2 BOOTP Relay Example**




3. Enable the routing circuit L601-3-1:

```
ncl> ENABLE NODE nis_peach ROUTING CIRCUIT L601-3-1
```

4. Disable the routing circuit that nis_mango will use (L602-6-0) to forward load requests to the BOOTP server (IP address 28.34.26.10), and downline load files from the BOOTP server.

```
ncl> DISABLE NODE nis_mango ROUTING CIRCUIT L602-6-0
```

5. Specify the routing circuit L602-6-0 as a BOOTP RELAY circuit

```
ncl> SET NODE nis_mango ROUTING CIRCUIT L602-6-0 -
_ncl> BOOTP RELAY 28.34.16.10
```

6. Enable the routing circuit L602-6-0:

```
ncl> ENABLE NODE nis_peach ROUTING CIRCUIT L602-6-0
```

# Part III

## Appendixes

This part contains the following appendixes:

- Appendix A lists the information required to configure the DECNIS in the DECNIS text-based configurator.
- Appendix B lists the files created by the DECNIS text-based configurator, and the files loaded to the DECNIS.
- Appendix C contains information about using DECdns with the DECNIS.
- Appendix D describes how to set up DECnet Phase IV system as a MOP load host for the DECNIS.

# A

# Information You Need for Configuration

This chapter lists the information you need to supply when you run the DECNIS text-based configurator.

Tables A–1 to A–21 list this information.

Write down your values in the last column, headed **Your Value**.

Note that the tables list all the information required for all cases. The information you actually need to supply depends on your configuration. For example, you do not need to supply X.25 circuit information if you do not wish to use any X.25 routing circuits.

### Default Values

The column labelled **Default** in the tables shows the default value supplied by the configurators for each item of information.

If the **Default** column shows –, this means that the configurator does not provide a default. If the value is required, you need to provide it yourself. The column labelled Required/Optional shows whether the value is required or optional.

**Table A–1 Configuration Information: DECNIS Node**

| Information Required | Notes | R(equired)/ O(ptional) | Default | Your Value |
|---|---|---|---|---|
| Load client name | Select the load client name (entered during load-host configuration) that identifies this DECNIS | R | – | |

**Table A–2 Configuration Information: Network Interface Cards**

| Information Required | Notes | R(equired)/ O(ptional) | Default | Your Value |
|---|---|---|---|---|
| **For each DECNIS slot listed on the screen:** | | | | |
| Network Interface Card | Select one of the card acronyms, or None (for an empty slot). Examples: W614, W618, W622, L601, L602, F621 | R | None | |

**Table A–3 Configuration Information: Configuration Options**

| Information Required | Notes | R(equired)/ O(ptional) | Default | Your Value |
|---|---|---|---|---|
| Use Internet Protocol (IP) routing? | Select Yes or No | R | – | |
| Use for X.25 Gateway? | Select Yes or No | R | – | |
| Use Bridging? | Select Yes or No | R | – | |
| Use NetWare IPX routing? | Select Yes or No | R | – | |
| Use AppleTalk routing? | Select Yes or No | R | – | |
| Does DECNIS have a Bridging license? | Select Yes or No | R | Yes | |
| Does DECNIS have a VCP license? | Select Yes or No | R | Yes | |
| Does DECNIS have a PC LAN license? | Select Yes or No | R | Yes | |
| Special X.25 options? | Select Yes or No for: Nonrouting PVCs; Closed User Groups; LLC2 | R | No for each | |
| Root Priority number (Bridging only) | Decimal digits. Range: 0–255. Determines whether the DECNIS will be the root bridge | R for Bridging | 128 | |
| CTF user name | Protects use of Common Trace Facility (CTF). Up to 16 characters | R | – | |
| CTF password | Protects use of CTF. Up to 16 characters | R | – | |
| Network Management user name | Protects use of NCL commands. Up to 16 characters | R | – | |
| Network Management password | Protects use of NCL commands. Up to 16 characters | R | – | |
| SNMP contact name | Name of person managing the DECNIS. Max. 255 characters | O | – | |
| SNMP domain name | Name for the DECNIS. Max. 255 characters | O | – | |
| SNMP system location | Description of physical location of the DECNIS. Max. 255 characters | O | – | |
| Type of Access to community "public" | Enter RO (read only) or RW (read and write) | R | – | |

(continued on next page)

**Table A–3 (Cont.) Configuration Information: Configuration Options**

| Information Required | Notes | R(equired)/ O(ptional) | Default | Your Value |
|---|---|---|---|---|
| Community name(s) | Additional community name(s). Max. 255 characters | O | – | |
| Type of access for communities | Enter RO (read only) or RW (read and write) | R if community name entered | – | |
| Set up SNMP traps? | Select Yes or No | R | – | |
| IP address(es) | IP address of system(s) to which the DECNIS will send traps | R for first address; O for the rest | – | |
| SNMP trap community name | Community name included in traps | R | "public" | |
| Set authentication failure trap? | Select Yes or No | R | – | |

## Table A–4 Configuration Information: Routing

| Information Required | Notes | R(equired)/ O(ptional) | Default | Your Value |
|---|---|---|---|---|
| Routing level | Select Level 1 or Level 2[1] | R | – | |
| **Level 1 Router Information** | | | | |
| Routing algorithm | Select Phase IV or Phase V | R | Phase IV | |
| Address Prefix | IDP + optional preDSP of a Phase IV compatible NSAP address, in DEC format. Up to 22 digits. Example 1: 37:12345: Example 2: 49:: | R if Phase IV address supplied | – | |
| Phase V area address (Phase V only) | The IDP, preDSP (optional) and Local Area fields of an NSAP address, in DEC format. Up to 40 digits. Example: 41:23456789:00–A5 | R if no Phase IV address (up to three). Otherwise, O (up to two) | – | |
| IP address for DECNIS [2] | System IP address for IP circuits with no IP address. *n.n.n.n* (*n* is a decimal number) | R if not entered during load-host configuration | – | |
| Use the RIP protocol?[2] | Select Yes or No | R | – | |
| Accept Default RIP route?[2] | Select Yes or No | R | Yes | |
| Announce Default RIP route?[2] | Select Yes or No | R | No | |
| Set up RIP sources?[2] | Yes or No | R | – | |
| IP address of RIP source[2] | *n.n.n.n* (*n* is a decimal number) | R | – | |
| Type of route propagation (if RIP selected)[3] | IS–IS to RIP? RIP to IS–IS? | R | No for both | |
| OSPF autonomous system boundary router? (only if OSPF selected) | Select Yes or No | R | – | |

[1]A Level 2 router acts also as a Level 1 router.

[2]Only asked if you selected IP.

[3]To use IS–IS, you must run Phase V routing at one or both levels.

(continued on next page)

**Table A–4 (Cont.) Configuration Information: Routing**

| Information Required | Notes | R(equired)/ O(ptional) | Default | Your Value |
|---|---|---|---|---|
| **Level 2 Router Information** | | | | |
| Routing algorithm | Select one of:<br>L1 Phase IV, L2 Phase IV<br>L1 Phase IV, L2 Phase V<br>L1 Phase V, L2 Phase IV<br>L1 Phase V, L2 Phase V | R | – | |
| Address Prefix | IDP + optional preDSP of a Phase IV compatible NSAP address, in DEC format. Up to 22 digits. Example 1: 37:12345: Example 2: 49:: | R if Phase IV address supplied for load-host configuration | – | |
| Phase V area address (if Phase V) | IDP, preDSP (optional), and local area fields of an NSAP, in DEC format. Up to 40 digits. Example: 41:23456789:00–A5 | R if there is no Phase IV address. Otherwise, O | – | |
| IP address for DECNIS [2] | System IP address for IP circuits with no IP address. *n.n.n.n* (*n* is a decimal number) | R if not entered during load-host configuration | – | |
| Use RIP protocol?[2] | Select Yes or No | R | Yes | |
| Use EGP protocol?[2] | Select Yes or No | R | Yes | |
| Use OSPF protocol?[2] | Select Yes or No | R | Yes | |
| Use Integrated IS-IS protocol?[2] | Select Yes or No | R | Yes | |
| Accept Default RIP route?[2] | Select Yes or No | R | Yes | |
| Announce Default RIP route?[2] | Select Yes or No | R | No | |
| Set up RIP sources? | Yes or No[2] | R | Yes | |
| IP address of each RIP source[2] | *n.n.n.n* (*n* is a decimal number) | R | – | |
| AS number (EGP only) | Number of autonomous system to which DECNIS belongs. Range 1–65535 | R | – | |

[2]Only asked if you selected IP.

(continued on next page)

**Table A–4 (Cont.) Configuration Information: Routing**

| Information Required | Notes | R(equired)/ O(ptional) | Default | Your Value |
|---|---|---|---|---|
| **Level 2 Router Information** | | | | |
| Type of route propagation (only if more than one IP protocol selected)[3] | Choice depends on protocols selected. Any or all of: IS–IS↔RIP; IS–IS↔EGP; EGP↔RIP | R | No for all | |
| OSPF autonomous system boundary router? (only if OSPF selected) | Select Yes or No | R | – | |

[3]To use IS–IS, you must run Phase V routing at one or both levels.

## Table A–5 Configuration Information: Lines

| Information Required | Notes | R(equired)/ O(ptional) | Default | Your Value |
|---|---|---|---|---|
| Select line to configure | Select from list displaying all ports set up previously (see Table A–2) | R | – | |
| Protocol (Lines on W622 NIC only) | Select one of: HDLC; PPP; Frame Relay; X.25; CHDLC; VCP | R | – | |
| Protocol (Lines on W614 or W618 NIC only) | Select one of: HDLC; PPP; DDCMP™; X.25; CHDLC | R | – | |
| | **CSMA/CD and FDDI information** | | | |
| Circuit name | Max. 32 characters | R | *port name* | |
| Enable circuit on system startup? | Select Yes or No | R | No | |
| Supply DECnet routing information? | Select Yes or No | R | No | |
| **The following information only applies if you choose to supply DECnet routing information** | | | | |
| Type of routing (Level 2 only) | Select Level 1 and 2 or Level 2 only | R | Level 1 and Level 2 | |
| Level 1 cost | Decimal number from 1–63 | R | 20 | |
| Level 2 cost | Decimal number from 1–63 | R | 20 | |
| Level 1 priority | Decimal number from 1–127 | R | 64 | |
| Level 2 priority | Decimal number from 1–127 | R | 64 | |
| **The following information only applies if you selected IP routing** | | | | |
| Run RIP on this circuit? | Select Yes or No | R | No | |
| Run EGP on this circuit? (Level 2 only) | Select Yes or No | R | No | |
| IP address | Circuit IP address. *n.n.n.n* (*n* is a decimal number) | O (but R on at least one CSMA/CD circuit) | – | |

(continued on next page)

**Table A–5 (Cont.) Configuration Information: Lines**

| Information Required | Notes | R(equired)/ O(ptional) | Default | Your Value |
|---|---|---|---|---|
| | **CSMA/CD and FDDI information** | | | |
| Subnet mask | *n.n.n.n* (*n* is a decimal number). You can use the digits 255 to show which part of the IP address is the network address. Example 1: 255.255.255.0: First three bytes are the network address. Last byte identifies the host. | R if address supplied | Depends on subnet class | |
| Alternative IP address (Only if IP address supplied) | Alternative local address(es) for this circuit. *n.n.n.n* (*n* is a decimal number) | O | – | |
| Alternative subnet mask | *n.n.n.n* (*n* is a decimal number). See IP subnet mask for more details | O | – | |
| RIP options | Only receive; Only send; Send and Receive | R | – | |
| AS number (EGP only) | AS number of an EGP neighbor | R | – | |
| IP address of EGP neighbor (EGP only) | *n.n.n.n* (*n* is a decimal number) | R | – | |
| **The following information only applies if you selected Bridging** | | | | |
| Use line as bridging port? | Select Yes or No | R | Yes | |
| Port name | Create a name. Max. 32 characters | R | Port-*n* | |
| Port number | Decimal number from 1–15 | R | Lowest available | |
| Cost | The lower the cost, the more likely that the DECNIS will be the designated bridge. Decimal number from 0–255 | R | 10 | |
| **The following information only applies if you selected NetWare IPX** | | | | |
| Run NetWare IPX? | Select Yes or No | R | – | |
| NetWare network number | Up to 8 hexadecimal digits | R | – | |
| Type of encapsulation | For CSMA/CD, select: Ethernet, 802.2, SNAP or Novell®. For FDDI, select 802.2 or SNAP | R | Ethernet | |

(continued on next page)

**Table A–5 (Cont.) Configuration Information: Lines**

| Information Required | Notes | R(equired)/ O(ptional) | Default | Your Value |
|---|---|---|---|---|
| | **CSMA/CD and FDDI information** | | | |
| Periodic update interval | Number of seconds between periodic RIP and SAP updates on this circuit. Decimal integer in range 60–65535 | R | 60 | |
| Accept NetBIOS® broadcast? | Select Yes to accept incoming NetBIOS broadcasts on this circuit | R | – | |
| **The following information only applies if you selected AppleTalk Routing** | | | | |
| Run AppleTalk? | Select Yes or No | R | – | |
| AppleTalk manual network address for the DECNIS | Network number plus node ID. Format: *number.node-id* Range: 1–65279 for network number. 128–253 for node ID. The value 0.0 means there is no network address. | O | 0.0 | |
| AppleTalk network range | Range of contiguous AppleTalk network numbers. Format: *number.number* Range: 1–65279 for each number. Example: 225–3000 | O | – | |
| AppleTalk default zone | Name of AppleTalk zone to be used for nodes with no preassigned zone or with an invalid zone name | O | – | |
| More AppleTalk zones (if default zone entered) | Name(s) of zones valid for this circuit | O | – | |

(continued on next page)

**Table A–5 (Cont.) Configuration Information: Lines**

| Information Required | Notes | R(equired)/ O(ptional) | Default | Your Value |
|---|---|---|---|---|
| | **HDLC information** | | | |
| Circuit name | Max. 32 characters | R | Line name | |
| Enable circuit on system startup? | Select Yes or No | R | No | |
| Supply DECnet routing information? | Select Yes or No | R | No | |
| **The following information only applies if you choose to supply DECnet routing information** | | | | |
| Transmit password | The characters %x followed by an even number of up to 38 hex digits | O | – | |
| Receive password | The characters %x followed by an even number of up to 38 hex digits | O | – | |
| Level 1 cost | Decimal number from 1–63 | R | 20 | |
| Level 2 cost | Decimal number from 1–63 | R | 20 | |
| Interphase link choice (only if Phase V routing at Level 2) | Choose one of: Phase IV Level 2 router; Phase V router running Phase IV routing protocols at Level 2; No interphase link | R | No interphase link | |
| Phase IV areas reachable by this circuit[1] | Enter list of area numbers. Example: 23, 30–35, 40 | R | – | |
| Path cost for Phase IV areas reachable by circuit[1] | Decimal number from 1–63 | R | 20 | |
| Other Phase IV areas reachable by DECNIS[1] | Enter list of area numbers. Example: 10, 15–22, 41–45 | R | – | |
| Path cost for other Phase IV areas[1] | Decimal number from 1–63 | R | 20 | |
| **The following information only applies if you selected IP routing** | | | | |
| Run RIP on this circuit? | Select Yes or No | R | No | |
| Run EGP on this circuit? (Level 2 only) | Select Yes or No | R | No | |

[1]You are only asked this if you are setting up an interphase link.

(continued on next page)

**Table A–5 (Cont.) Configuration Information: Lines**

| Information Required | Notes | R(equired)/ O(ptional) | Default | Your Value |
|---|---|---|---|---|
| | **HDLC information** | | | |
| Neighbor IP address | *n.n.n.n* (*n* is a decimal number). IP address of IP host to which this circuit connects. | R for RIP or EGP if no local IP address | – | |
| Local IP address | Local address for this circuit. *n.n.n.n* (*n* is a decimal number). | R for RIP or EGP if no neighbor IP address | – | |
| Local subnet mask | *n.n.n.n* (*n* is a decimal number). You can use the digits 255 to show which part of the IP address is the network address. Example: 255.255.255.0: First three bytes are the network address. Last byte identifies the host. | R if local IP address supplied | Depends on subnet class | |
| RIP options | Only receive; Only send; Send and Receive | R | – | |
| AS number (EGP only) | AS number of an EGP neighbor | R | – | |
| IP address of EGP neighbor (EGP only) | *n.n.n.n* (*n* is a decimal number) | R | – | |
| **The following information only applies if you selected Bridging** | | | | |
| Use this line as bridging port? | Select Yes or No | R | No | |
| Port name | Create a name. Max. 32 characters | R | Port-*n* | |
| Port number | Decimal number from 1–15 | R | Lowest available | |
| Cost | The lower the cost, the more likely the DECNIS is to be the designated bridge. Decimal number 0–255 | R | 10 | |
| Enable Spanning Tree? | Select Yes or No | R | Yes | |

(continued on next page)

**Table A–5 (Cont.) Configuration Information: Lines**

| Information Required | Notes | R(equired)/ O(ptional) | Default | Your Value |
|---|---|---|---|---|
| | **PPP information** | | | |
| Select routing protocols | Select any or all of: OSI routing; DECnet Phase IV routing; IP routing; IPX routing (IP and IPX appear only if selected in Configuration Options) | R | Yes for each | |
| Circuit name | Max. 32 characters | R | Line name | |
| Enable circuit on system startup? | Select Yes or No | R | No | |
| Supply DECnet routing information? | Select Yes or No | R | No | |
| **The following information only applies if you choose to supply DECnet routing information** | | | | |
| Level 1 cost | Decimal number from 1–63 | R | 20 | |
| Level 2 cost | Decimal number from 1–63 | R | 20 | |
| Interphase link choice (only if using Phase V routing at Level 2) | Choose one of: Phase IV Level 2 router; Phase V router running Phase IV routing protocols at Level 2; No interphase link | R | No interphase link | |
| Phase IV areas reachable by this circuit[1] | Enter list of area numbers. Example: 23, 30–35, 40 | R | – | |
| Path cost for Phase IV areas reachable by circuit[1] | Decimal number from 1–63 | R | 20 | |
| Other Phase IV areas reachable by DECNIS[1] | Enter list of area numbers. Example: 10, 15–22, 41–45 | R | – | |
| Path cost for other Phase IV areas[1] | Decimal number from 1–63 | R | 20 | |
| **IP information is the same as for HDLC circuits** | | | | |
| **The following information only applies if you selected NetWare IPX** | | | | |
| Run NetWare IPX? | Select Yes or No | R | – | |
| NetWare network number | Up to 8 hexadecimal digits | R | – | |

[1]You are only asked this if you are setting up an interphase link.

(continued on next page)

**Table A–5 (Cont.) Configuration Information: Lines**

| Information Required | Notes | R(equired)/ O(ptional) | Default | Your Value |
|---|---|---|---|---|
| | **PPP information** | | | |
| Periodic update interval | Number of seconds between periodic RIP and SAP updates on this circuit. Decimal integer in range 60–65535 | R | 60 | |
| Accept NetBIOS broadcast? | Select Yes to accept incoming NetBIOS broadcasts on this circuit | R | – | |
| **The following information only applies if you selected Bridging** | | | | |
| Use this line as bridging port? | Select Yes or No | R | No | |
| Port name | Create a name. Max. 32 characters | R | Port-*n* | |
| Port number | Decimal number from 1–15 | R | Lowest available | |
| Cost | The lower the cost, the more likely the DECNIS is to be the designated bridge. Decimal number 0–255 | R | 10 | |
| Enable Spanning Tree? | Select Yes or No | R | Yes | |
| Use minimum sized frame compression? | Select Yes or No | R | – | |
| | **DDCMP information** | | | |
| Circuit name | Max. 32 characters | R | Line name | |
| Enable circuit on system startup? | Select Yes or No | R | No | |
| Communications mode | Select synchronous or asynchronous | R | – | |
| Line speed (only if asynchronous selected) | Select from 1200, 2400, 4800, 9600, 19200, 38400, 56K, 64K, 128K | R | – | |
| Type of modem control (only if asynchronous selected) | Select Full modem control or Data leads only | R | Full modem control | |
| Supply DECnet routing information? | Select Yes or No | R | No | |
| **DECnet routing and Internet Protocol information are both the same as for HDLC circuits** | | | | |

(continued on next page)

**Table A–5 (Cont.) Configuration Information: Lines**

| Information Required | Notes | R(equired)/ O(ptional) | Default | Your Value |
|---|---|---|---|---|
| | **Frame Relay information** | | | |
| Select data link protocol | Select CHDLC or PPP | R | – | |
| Select management protocol | Select one of: LMI/Joint; ANSI T1.617, Annex D; CCITT Q.933, Annex A | R | – | |
| | **X.25 information** | | | |
| DTE name | Max. 32 characters | R | DTE-*slot-port* | |
| X.25 DTE address | DTE address. Max. 15 digits. Obtain from your PSDN | R | – | |
| Logical channel range | Obtain from PSDN. Numbers or range(s) of numbers. Example: 1024–1048, 30 | R | – | |
| Profile name | Name of network profile for this DTE's PSDN. Supplied by Digital. See also Network Information (NI) | R | – | |
| Flow control negotiation?[2] | Select Yes or No | R | Yes | |
| Extended packet sequence numbering?[2] | Select Yes or No | R | – | |
| Default packet size | Decimal number (power of 2). See profile and PSDN subscription | R | As in profile | |
| Maximum packet size[3] | Decimal number (power of 2). See profile and PSDN subscription | R | As in profile | |
| Minimum packet size[3] | Decimal number (power of 2). See PSDN subscription | R | As in profile | |
| Default window size | Decimal number 1–127. See PSDN subscription | R | As in profile | |
| Maximum window size[3] | Decimal number 1–127. See PSDN subscription | R | As in profile | |
| Minimum window size[3] | Decimal number 1–127. See PSDN subscription | R | As in profile | |

[2]You are only asked for this if your Profile supports it.

[3]You only need to enter values if you have chosen flow control negotiation.

(continued on next page)

**Table A–5 (Cont.) Configuration Information: Lines**

| Information Required | Notes | R(equired)/ O(ptional) | Default | Your Value |
|---|---|---|---|---|
| | **X.25 information** | | | |
| Interface mode[4] | Select DTE or DCE | R | – | |
| Window size (frame) | Decimal number. See PSDN subscription | R | As in profile | |
| DTE Class | Max. 32 characters. The name of a DTE Class to which this DTE belongs | R | Profile name | |
| | **CHDLC information** | | | |
| Select routing protocols | Select any or all of: OSI routing; DECnet Phase IV routing; IP routing; IPX routing (IP and IPX appear only if selected in Configuration Options) | R | Yes for each | |
| Circuit name | Max. 32 characters | R | Line name | |
| Enable circuit on system startup? | Select Yes or No | R | No | |
| **DECnet, IP and NetWare IPX information are all the same as for PPP Circuits** | | | | |
| **Bridging information is the same as for HDLC circuits** | | | | |
| | **VCP information** | | | |
| Select Turbo or Non-Turbo | Select type of Network Interface Card used by the Vitalink system to which the VCP line connects | R | – | |
| Circuit name | Max. 32 characters | R | *port name* | |
| Enable circuit on system startup? | Select Yes or No | R | No | |
| **DECnet, IP, NetWare IPX and AppleTalk information are all the same as for CSMA/CD and FDDI circuits** | | | | |
| **Bridging information is the same as for PPP circuits** | | | | |

[4]You are only asked for this if the Profile is ISO8208 or NPSI.

**Table A–6 Configuration Information: Bridge Filtering**

| Information Required | Notes | R(equired)/ O(ptional) | Default | Your Value |
|---|---|---|---|---|
| Method of entering protocol types | Choose either: Enter protocols to be forwarded or enter protocols to be blocked | R | – | |
| Ethernet format protocol types to be forwarded (if `forwarded` chosen) | List the protocol types the DECNIS should forward | O | – | |
| IEEE 802.2 format protocol types to be forwarded (if `forwarded` chosen) | List the protocol types the DECNIS should forward | O | – | |
| IEEE 802.2 SNAP format protocol types to be forwarded (if `forwarded` chosen) | List the protocol types the DECNIS should forward | O | – | |
| Ethernet format protocol types to be blocked (if `blocked` chosen) | List the protocol types the DECNIS should block | O | – | |
| IEEE 802.2 format protocol types to be blocked (if `blocked` chosen) | List the protocol types the DECNIS should block | O | – | |
| IEEE 802.2 SNAP format protocol types to be blocked (if `blocked` chosen) | List the protocol types the DECNIS should block | O | – | |

## Table A–7 Configuration Information: X25 Routing Circuits

| Information Required | Notes | R(equired) /O(ptional) | Default | Your Value |
|---|---|---|---|---|
| Circuit type | Select one of: X25 Static Outgoing, X25 Static Incoming, X25 Permanent, X25 DA | R | – | |
| Circuit name | Max. 32 characters | R | – | |
| **X25 Static Outgoing Circuits** | | | | |
| Template name | Max. 32 characters | R | *circuit-name*-out | |
| DTE Class | A local DTE Class containing the DTE this circuit will use for making calls. Max. 32 characters | R | – | |
| Destination DTE address | DTE address of remote system to which this circuit will connect | R | – | |
| Call data | The characters %x followed by up to 254 hex digits | O | %xff0000004445 436e65742d444c4d | |
| Packet size | Decimal number (power of 2). See PSDN subscription | O | – | |
| Window size | 1–127. See PSDN subscription | O | – | |
| Reverse Charging? | Select Yes or No | R | No | |
| Throughput Class Request | Incoming and outgoing baud rates for circuit. [*incoming..outgoing*] Example: [48..64] | O | – | |
| Supply DECnet routing information? | Select Yes or No | R | No | |
| **The following information only applies if you choose to supply DECnet routing information** | | | | |
| Transmit password | The characters %x followed by an even number of up to 38 hex digits | O | – | |
| Receive password | The characters %x followed by an even number of up to 38 hex digits | O | – | |
| Level 1 cost | Decimal number from 1–63 | R | 20 | |
| Level 2 cost | Decimal number from 1–63 | R | 20 | |

(continued on next page)

**Table A–7 (Cont.) Configuration Information: X25 Routing Circuits**

| Information Required | Notes | R(equired) /O(ptional) | Default | Your Value |
|---|---|---|---|---|
| | **X25 Static Outgoing Circuits** | | | |
| Interphase link choice (only if using Phase V (link state) routing at Level 2) | Choose one of: Phase IV Level 2 router; Phase V router running Phase IV routing protocols at Level 2; No interphase link | R | No interphase link | |
| Phase IV areas reachable by circuit[1] | Enter list of area numbers. Example: 23, 30–35, 40 | R | – | |
| Path cost for Phase IV areas reachable by circuit[1] | Decimal number from 1–63 | R | 20 | |
| Other Phase IV areas reachable by DECNIS[1] | Enter list of area numbers. Example: 10, 15–22, 41–45 | R | – | |
| Path cost for other Phase IV areas[1] | Decimal number from 1–63 | R | 20 | |
| **The following information only applies if you selected IP routing** | | | | |
| Run RIP on this circuit? | Select Yes or No | R | No | |
| Run EGP on this circuit? | Select Yes or No | R | No | |
| Neighbor IP address | *n.n.n.n* (*n* is a decimal number). IP address of IP host to which this circuit connects. | R for RIP or EGP if no local IP address | – | |
| Local IP address | Local address for this circuit. *n.n.n.n* (*n* is a decimal number). | R for RIP or EGP if no neighbor IP address | – | |
| Local subnet mask | *n.n.n.n* (*n* is a decimal number). You can use the digits 255 to show which part of the IP address is the network address. Example: 255.255.255.0: First three bytes are the network address. Last byte identifies the host. | R if local IP address supplied | Depends on subnet class | |

[1]You are only asked this if you are setting up an interphase link.

(continued on next page)

**Table A–7 (Cont.) Configuration Information: X25 Routing Circuits**

| Information Required | Notes | R(equired) /O(ptional) | Default | Your Value |
|---|---|---|---|---|
| | **X25 Static Outgoing Circuits** | | | |
| RIP options | Only receive; Only send; Send and Receive | R | – | |
| AS number (EGP only) | AS number of an EGP neighbor | R | – | |
| IP address of EGP neighbor (EGP only) | *n.n.n.n* (*n* is a decimal number) | R | – | |
| | **X25 Static Incoming Circuits** | | | |
| Template name | Create a name. Max. 32 characters | R | *circuit-name*-IN | |
| Packet size | Decimal number 16–4096 (power of 2) | O | – | |
| Window size | 1–127. See PSDN subscription | O | – | |
| Throughput Class Request | Incoming and outgoing baud rates for circuit. [*incoming..outgoing*] Example: [48..64] | O | – | |
| Filter name | Max. 32 characters | R | *circuit-name*-IN | |
| Call data value | The characters %x followed by up to 254 hex digits | O | %xff0000004445 436e65742d444c4d | |
| Call data mask | The characters %x followed by up to 254 hex digits | O | %xffffffffffff ffffffffffff | |
| Subaddress range | Range of decimal numbers from 0 to 65535. Example: [[2..24]] | O | – | |
| Sending DTE address | Calling address field of incoming call packet. Max. 15 digits | O | – | |
| DTE Class | DTE Class used for receiving call. Max. 32 characters | O | – | |
| Incoming DTE address | Called address field of incoming call packet. Max. 15 digits | O | – | |

The remaining information is the same as for X25 Static Outgoing Circuits, from the question, **Supply DECnet routing information?** onward

(continued on next page)

**Table A–7 (Cont.) Configuration Information: X25 Routing Circuits**

| Information Required | Notes | R(equired) /O(ptional) | Default | Your Value |
|---|---|---|---|---|
| | **X25 Permanent Circuits** | | | |
| Local DTE to be used | Select DTE name | R | – | |
| PVC name | Max. 32 characters | R | – | |
| Packet size | Decimal number (power of 2). See PSDN subscription | R | Default for DTE | |
| Window size | 1–127. See PSDN subscription | R | Default for DTE | |
| Channel | Channel assigned by PSDN | R | – | |
| Supply DECnet routing information? | Select Yes or No | R | No | |
| **The following information only applies if you choose to supply DECnet routing information** | | | | |
| Transmit password | The characters %x followed by an even number of up to 38 hex digits | O | – | |
| Receive password | The characters %x followed by an even number of up to 38 hex digits | O | – | |
| Level 1 cost | Decimal number from 1–63 | R | 20 | |
| Level 2 cost | Decimal number from 1–63 | R | 20 | |
| Interphase link choice (only if using Phase V (link state) routing at Level 2) | Choose one of: Phase IV Level 2 router; Phase V router running Phase IV routing protocols at Level 2; No interphase link | R | No Interphase link | |
| Phase IV areas reachable by circuit[1] | Enter list of area numbers. Example: 23, 30–35, 40 | R | – | |
| Path cost for Phase IV areas reachable by circuit[1] | Decimal number from 1–63 | R | 20 | |
| Other Phase IV areas reachable by DECNIS[1] | Enter list of area numbers. Example: 10, 15–22, 41–45 | R | – | |
| Path cost for other Phase IV areas[1] | Decimal number from 1–63 | R | 20 | |

[1]You are only asked this if you are setting up an interphase link.

(continued on next page)

**Table A–7 (Cont.) Configuration Information: X25 Routing Circuits**

| Information Required | Notes | R(equired) /O(ptional) | Default | Your Value |
|---|---|---|---|---|
| | **X25 Dynamically Assigned (DA) Circuits** | | | |
| Use of circuit? | Select one of: OSI data; IP data; Both OSI and IP | R | – | |
| OSI Template name | Max. 32 characters | R | *circuit-name*-DA-OSI | |
| IP Template name | Max. 32 characters | R | *circuit-name*-DA-IP | |
| DTE Class | Local DTE Class containing the DTE the circuit will use for making calls. Max. 32 characters | R | – | |
| IP Call Data | The characters %xcc followed by an even number of up to 252 hex digits | O | %xcc | |
| OSI Call Data | The characters %x81 followed by an even number of up to 252 hex digits | O | %x81 | |
| Packet size | Decimal number 16–4096 (power of 2). See PSDN subscription | O | – | |
| Window size | Decimal number 1–127. See PSDN subscription | O | – | |
| Reverse Charging? | Select Yes or No | R | No | |
| Throughput Class Request | Incoming and outgoing baud rates for circuit. [*incoming..outgoing*] Example: [48..64] | O | – | |
| OSI Filter name | Max. 32 characters | R | *circuit-name*-DA-OSI | |
| IP Filter name | Max. 32 characters | R | *circuit-name*-DA-IP | |
| OSI Call data value | The characters %x81 followed by an even number of up to 252 hex digits. | O | %x81 | |
| IP Call data value | The characters %xcc followed by an even number of up to 252 hex digits | O | %xcc | |
| Call data mask | The characters %xff followed by an even number of up to 252 hex digits. Same size as call data value | O | %xff | |

(continued on next page)

**Table A–7 (Cont.) Configuration Information: X25 Routing Circuits**

| Information Required | Notes | R(equired) /O(ptional) | Default | Your Value |
|---|---|---|---|---|
| | **X25 Dynamically Assigned (DA) Circuits** | | | |
| DTE Class | DTE Class used for receiving call. Max. 32 characters | O | – | |
| Incoming DTE address | Called address field of incoming call packet. Max. 15 digits | O | – | |

**Table A–8 Configuration Information: Tunnel Circuits**

| Information Required | Notes | R(equired)/ O(ptional) | Default | Your Value |
|---|---|---|---|---|
| Protocol to send on the circuit | Select Yes or No for NetWare IPX and AppleTalk | R | – | |
| Type of Circuit (Only asked if NetWare IPX is the only protocol) | Select Point-to-point or Broadcast | R | – | |
| Tunnel circuit name | Max. 32 characters | R | – | |
| Destination IP Address(es) | Point-to-point: one address. Broadcast (NetWare IPX only): no restriction on the number of addresses. *n.n.n.n* (*n* is a decimal number) | R | – | |
| NetWare network number | Up to 8 hexadecimal digits | R | – | |

## Table A–9 Configuration Information: Frame Relay Connections

| Information Required | Notes | R(equired) /O(ptional) | Default | Your Value |
|---|---|---|---|---|
| Select frame relay channel | Select from list of lines previously figured as frame relay channels | R | – | |
| Connection name | Max. 32 characters | R | – | |
| Data link connection identifier | Up to four decimal digits. Range: 16–1007 | O | – | |
| Select routing protocols | Select any or all of: OSI routing; DECnet Phase IV routing; IP routing; IPX routing (IP and IPX appear only if selected in Configuration Options) | R | Yes for each | |
| Supply DECnet routing information? | Select Yes or No | R | No | |
| **The following information only applies if you choose to supply DECnet routing information** | | | | |
| Level 1 cost | Decimal number from 1–63 | R | 20 | |
| Level 2 cost | Decimal number from 1–63 | R | 20 | |
| Interphase link choice (only if using Phase V (link state) routing at Level 2) | Choose one of: Phase IV Level 2 router; Phase V router running Phase IV routing protocols at Level 2; No interphase link | R | No interphase link | |
| Phase IV areas reachable by circuit[1] | Enter list of area numbers. Example: 23, 30–35, 40 | R | – | |
| Path cost for Phase IV areas reachable by circuit[1] | Decimal number from 1–63 | R | 20 | |
| Other Phase IV areas reachable by DECNIS[1] | Enter list of area numbers. Example: 10, 15–22, 41–45 | R | – | |
| Path cost for other Phase IV areas[1] | Decimal number from 1–63 | R | 20 | |
| **The following information only applies if you selected IP routing** | | | | |
| Run RIP on this circuit? | Select Yes or No | R | No | |

[1]You are only asked this if you are setting up an interphase link.

(continued on next page)

**Table A–9 (Cont.) Configuration Information: Frame Relay Connections**

| Information Required | Notes | R(equired) /O(ptional) | Default | Your Value |
|---|---|---|---|---|
| Run EGP on this circuit? | Select Yes or No | R | No | |
| Neighbor IP address | *n.n.n.n* (*n* is a decimal number). IP address of IP host to which this circuit connects. | R for RIP or EGP if no local IP address | – | |
| Local IP address | Local address for this circuit. *n.n.n.n* (*n* is a decimal number). | R for RIP or EGP if no neighbor IP address | – | |
| Local subnet mask | *n.n.n.n* (*n* is a decimal number). You can use the digits 255 to show which part of the IP address is the network address. Example: 255.255.255.0: First three bytes are the network address. Last byte identifies the host | R if local IP address supplied | Depends on subnet class | |
| RIP options | Only receive; Only send; Send and Receive | R | – | |
| AS number (EGP only) | AS number of an EGP neighbor | R | – | |
| IP address of EGP neighbor (EGP only) | *n.n.n.n* (*n* is a decimal number) | R | – | |
| **The following information only applies if you selected Bridging** | | | | |
| Use this connection as bridging port? | Select Yes or No | R | No | |
| Port name | Create a name. Max. 32 characters | R | Port-*n* | |
| Port number | Decimal number from 1–15 | R | Lowest available | |
| Cost | The lower the cost, the more likely the DECNIS is to be the designated bridge. Decimal number 0–255 | R | 10 | |
| Enable Spanning Tree? | Select Yes or No | R | Yes | |
| Use minimum sized frame compression? (only if PPP selected as the data link) | Select Yes or No | R | – | |

**Table A–10 Configuration Information: OSI Reachable Addresses**

| Information Required | Notes | R(equired)/ O(ptional) | Default | Your Value |
|---|---|---|---|---|
| Circuit name | Select from list | R | – | |
| Reachable Address name | Max. 32 characters | R | – | |
| Reachable Address Prefix of domain | All or leading digits of NSAP address, up to 40 digits. DEC, OSI or HRPF format. Examples: DEC format: 37:32655678:3214:<br><br>HRPF format: /37326556783214<br><br>OSI format: 3732655678+3214 | R | – | |
| Reachable Address cost | Cost of reaching the destination node. Decimal number, 1–63 | R | 20 | |
| LAN hardware address of node connecting to foreign domain (CSMA/CD circuit only) | Six pairs of hexadecimal digits, with hyphen separating each pair. Example: 08-00-2B-65-BB-43 | R | – | |
| DTE address of destination node (X.25 DA circuit only) | DTE address | R | – | |

**Table A–11 Configuration Information: IP Reachable Addresses**

| Information Required | Notes | R(equired)/ O(ptional) | Default | Your Value |
|---|---|---|---|---|
| Circuit used to reach Reachable Address | Select from list | R | – | |
| Reachable Address name | Max. 32 characters | R | – | |
| Destination address to be reached | *n.n.n.n* (*n* is a decimal number). Address of host, subnet or network | R | – | |
| Destination subnet mask | *n.n.n.n* (*n* is a decimal number) | R | Depends on subnet class | |
| IP Address of next IP router (not for X.25 DA) | *n.n.n.n* (*n* is a decimal number) | R for CSMA /CD and FDDI; O for other protocols | – | |
| DTE address of next IP router (X.25 DA only) | DTE address of the next gateway on path. Up to 15 digits | R | – | |
| IP Reachable Address cost | Cost of reaching the destination for this reachable address. Decimal number | O | 20 | |

### Table A–12 Configuration Information: PVCs

| Information Required | Notes | R(equired)/ O(ptional) | Default | Your Value |
|---|---|---|---|---|
| DTE name | Select from list | R | – | |
| PVC name | Max. 32 characters | R | PVC-*n* | |
| Channel number | Decimal number. See PSDN subscription | R | – | |
| Packet size | Decimal number (power of 2). See PSDN subscription | R | Default packet size for DTE | |
| Window size | Decimal number. See PSDN subscription | R | Default window size for DTE | |

### Table A–13 Configuration Information: Groups

| Information Required | Notes | R(equired)/ O(ptional) | Default | Your Value |
|---|---|---|---|---|
| Group name | Max. 32 characters | R | GROUP-*n* | |
| Group type | BCUG or CUG. See PSDN subscription | R | BCUG | |
| **For each DTE you want to place in the Group, enter:** | | | | |
| CUG number | Decimal number. See PSDN subscription | R | – | |
| Remote DTE address[1] | DTE address of other system in BCUG. Max. 15 digits | R | – | |

[1]You will only be asked for this information if the Group type is BCUG.

**Table A–14 Configuration Information: LLC2**

| Information Required | Notes | R(equired)/ O(ptional) | Default | Your Value |
|---|---|---|---|---|
| **For each LLC2 system you want to connect to:** | | | | |
| LAN device to be used | Select CSMA/CD port from list | R | – | |
| LLC2 DTE name | Max. 32 characters | R | DTE-*n* | |
| LLC2 DTE address | Max. 15 digits | R | – | |
| Logical channel range(s) | Number(s) or range(s) of numbers, decided in consultation with remote system. Range: From 1 to 4095. Example: 1024–1048, 30 | R | – | |
| Local LSAP | 2 hex digits | R | 7E | |
| Remote LSAP | 2 hex digits | R | 7E | |
| Remote MAC address | LAN hardware address. Example: 08-00-2B-02-AA-23 | R | – | |
| Flow control negotiation? | Select Yes or No | R | No | |
| Extended packet sequence numbering? | Select Yes or No | R | No | |
| Minimum packet size[1] | Decimal number. Power of 2 in range 16 to 4096 | R | 16 | |
| Maximum packet size[1] | Decimal number. Power of 2 in range 16 to 4096 | R | 1024 | |
| Default packet size | Decimal number. Power of 2 in range 16 to 4096 | R | 128 | |
| Minimum window size (packet level)[1] | Decimal number. Range: 1 to 127 | R | 1 | |
| Maximum window size (packet level)[1] | Decimal number. Range: 1 to 127 | R | 7 | |
| Default window size (packet level) | Decimal number. Range: 1 to 127 | R | 2 | |
| DTE class | Max. 32 characters | R | LLC2-CLASS-*n* | |

[1]You only need to enter values if you have chosen flow control negotiation.

## Table A–15 Configuration Information: X.25 Server Clients

| Information Required | Notes | R(equired)/ O(ptional) | Default | Your Value |
|---|---|---|---|---|
| **If you wish the configurator to use a naming service (DECdns or local) to find X.25 server client addresses:[1]** | | | | |
| **Supply the following for each X.25 server client system:** | | | | |
| Server client name | Create a name. Max. 32 characters | R | CLIENT-*n* | |
| Server client node name | Node name of Client system associated with this X.25 server client. Max. 400 characters | R | – | |
| **If the configurator will not use a naming service (DECdns or local) to find Client system addresses:** | | | | |
| **Supply one and only one of the following for each Client system associated with an X.25 server client**: | | | | |
| NSAP address of Client system | NSAP address format. Example: 41:23456789:00–A5:07-CA-4B-65-BB-43 | O | – | |
| Phase IV address | Phase IV address. Example: 34.3 | O | – | |

[1]Not asked on PC load hosts.

**Table A–16 Configuration Information: Filters**

| Information Required | Notes | R(equired)/ O(ptional) | Default | Your Value |
|---|---|---|---|---|
| Filter name | Max. 32 characters | R | – | |
| Priority | Decimal number, 0–65535 | R | 1 | |
| Incoming DTE address | Called address field of incoming call packet. Max. 15 digits | O | – | |
| Call data value | The characters %x followed by an even number of up to 254 hex digits | O | – | |
| Call data mask | The characters %x followed by an even number of up to 254 hex digits | O | – | |
| Subaddress range | Range of decimal numbers from 0 to 65535. Example: [[2..24]] | O | – | |
| DTE Class | DTE Class used for receiving call. Max. 32 characters | O | – | |
| Sending DTE address | Max. 15 digits | O | – | |
| Receiving DTE address | Max. 15 digits | O | – | |
| Group name | Max. 32 characters | O | – | |
| Originally called address | Max. 15 digits | O | – | |
| Redirect reason | One of: Busy; Out of order; Systematic; Not specified | O | Not specified | |
| Called address extension value | Hex digits | O | – | |
| Called address extension mask | Hex digits | O | – | |
| Called NSAP | The characters %x, followed by an even number of up to 128 hex digits | O | – | |

### Table A–17 Configuration Information: X.25 Security

| Information Required | Notes | R(equired)/ O(ptional) | Default | Your Value |
|---|---|---|---|---|
| Set up X.25 Security? | Select Yes (to set up detailed X.25 security) or No (for open X.25 security) | R | Yes | |

### Table A–18 Configuration Information: Incoming Security for X.25 Server Clients

| Information Required | Notes | R(equired)/ O(ptional) | Default | Your Value |
|---|---|---|---|---|
| X.25 Server client on which to set up security | Select from list | R | – | |
| DTE address prefixes of remote systems that can call the X.25 server client's Client system only if they pay for the call (Remote Charge access)[1] | The leading digits of a DTE address, up to 15 digits | O | – | |
| DTE addresses of systems that can call the X.25 server client's Client system irrespective of who pays for the call (All access)[1] | The leading digits of a DTE address, up to 15 digits | O | – | |
| DTE addresses of systems that are not allowed to call the X.25 server client's Client system (No Access)[1] | The leading digits of a DTE address, up to 15 digits | O | Wildcard (*)[2] | |

[1]Enter a Remote Address Prefix (RAP). This is either a full DTE address, or the leading digits of a DTE address to stand for all DTEs with an address beginning with these digits.

[2]The wildcard character (*) means all unspecified DTEs. If you enter * to stand for DTEs with Remote Charge or All access, then all DTEs will have access except those that you specify explicitly.

**Table A–19 Configuration Information: Outgoing Security for Client Systems**

| Information Required | Notes | R(equired)/ O(ptional) | Default | Your Value |
|---|---|---|---|---|
| Client system on which to set up security | Node name. Max. 256 characters | R | – | |
| Address to identify Client system (only if no naming service) | Enter either a Phase IV address or an NSAP address | R | – | |
| Security Name for Client system | Max. 32 characters | R | – | |
| DTE addresses of remote systems that can be called by this Client system only if the remote systems pay for the call (Remote Charge access)[1] | The leading digits of a DTE Address, up to 15 digits | O | – | |
| DTE addresses of systems that can be called by this Client system irrespective of who pays for the call (All access)[1] | The leading digits of a DTE address, up to 15 digits | O | – | |
| Names of PVCs that can be accessed by this Client system | Max. 32 characters | O | – | |
| DTE addresses of systems that cannot be called by this Client system (No access)[1] | The leading digits of a DTE address, up to 15 digits | O | Wildcard (*)[2] | |
| Names of PVCs that cannot be accessed by this Client system[1] | Max. 32 characters | O | – | |

[1]Enter a Remote Address Prefix (RAP). This is either a full DTE address or the leading digits of a DTE address, to stand for all DTEs with an address beginning with these digits.

[2]The wildcard character (*) means all unspecified DTEs. If you enter the * to stand for DTEs or PVCs with Remote Charge or All access, then all DTEs or PVCs can be accessed except those that you specify explicitly.

**Table A–20 Configuration Information: Event Logging**

| Information Required | Notes | R(equired)/ O(ptional) | Default | Your Value |
|---|---|---|---|---|
| Event stream name | Max. 32 characters | R | – | |
| **If you wish the configurator to use a naming service (DECdns or local) to find event sink addresses:[1]** | | | | |
| **Supply the following for each event sink:** | | | | |
| Event sink name | Node name | R | – | |
| Type of sink name | DECdns node name or DECdns object name | R | Node name | |
| **If the configurator will not use a naming service (DECdns or local) to find event sink addresses:** | | | | |
| **Supply one and only one of the following for each event sink:** | | | | |
| NSAP address of event sink | NSAP address format. Example: 41:23456789:00–A5:07-CA-4B-65-BB-43 | O | – | |
| Phase IV address of event sink | Phase IV address. Example: 34.3 | O | – | |

[1]Not asked on PC load hosts.

**Table A–21 Configuration Information: Database Sizing**

| Information Required | Notes | R(equired) /O(ptional) | Default | Your Value |
|---|---|---|---|---|
| LAN end system adjacencies | For each LAN circuit displayed, enter the number of end systems on the LAN | R | 2480 | |
| Number of router adjacencies | Number of routers directly connected to the DECNIS | R | 170 | |
| Number of end system adjacencies | Number of nonrouting systems reachable over all DECNIS circuits | R | 5120 | |
| Number of manual adjacencies | Systems in the local area that do not exchange adaptive routing information with the DECNIS | R | 60 | |
| Number of Level 1 routing destinations | Unique NSAP addresses in the local area | R | 5280 | |
| Number of Level 1 routers in local area | Number of OSI Level 1 routers | R | 100 | |
| Level 1 average connectivity | Average number of routers from which the DECNIS learns about nodes in the local area, multiplied by 10 | R | 20 | |
| Number of IP local adjacencies (IP only) | Number of IP subnets to which the DECNIS is directly connected, plus neighbor IP addresses | R | 50 | |
| Number of IP reachable destination (IP only) | IP Reachable Addresses set up on the DECNIS | R | 200 | |
| Number of IP Level 1 destinations (IP only) | Number of unique IP subnet addresses in the same area as the DECNIS | R | 250 | |
| IP area connectivity (IP only) | Average number of routers from which the DECNIS learns about each IP host within the local area, multiplied by 10 | R | 20 | |
| Number of IP external destinations (IP only) | Maximum number of IP destinations the DECNIS will learn from protocols other than Integrated IS–IS | R | 500 | |
| **If the DECNIS is a Level 2 Router, you will also see the following:** | | | | |
| Number of DA adjacencies | Number of X.25 DA circuits created on the DECNIS | R | 160 | |
| Number of Level 2 routing destinations | DECnet/OSI area addresses plus OSI reachable addresses | R | 456 | |

(continued on next page)

**Table A–21 (Cont.) Configuration Information: Database Sizing**

| Information Required | Notes | R(equired) /O(ptional) | Default | Your Value |
|---|---|---|---|---|
| Number of Level 2 routers in the domain | Number of Level 2 routers in the same routing domain as the DECNIS | R | 512 | |
| Level 2 average connectivity | Average number of routers from which the DECNIS learns about nodes in the domain, multiplied by 10 | R | 20 | |
| Number of OSI reachable addresses | OSI reachable addresses set up on the DECNIS | R | 200 | |
| Number of IP Level 2 destinations (IP only) | Number of unique IP subnet addresses within the routing domain | R | 890 | |
| IP domain connectivity (IP only) | Average number of routers from which the DECNIS learns about each IP host within the routing domain, multiplied by 10 | R | 20 | |
| **If you selected OSPF, you will also see the following:** | | | | |
| OSPF maximum connected areas | Max. number of OSPF areas that the DECNIS can connect to directly | R | 2 | |
| OSPF average connected routers | Average number of OSPF routers in each area that the DECNIS is connected to directly | R | 10 | |
| OSPF maximum area interfaces | Max. number of OSPF interfaces to a single area on any OSPF router in a connected area | R | 3 | |
| OSPF average area networks | Average number of OSPF transit and stub networks in each area that the DECNIS is connected to directly | R | 40 | |
| OSPF maximum network routers | Max. number of OSPF routers in any OSPF network in a connected area | R | 5 | |
| OSPF maximum system networks | Max. number of OSPF networks in the autonomous system (AS) | R | 25 | |
| OSPF maximum boundary routers | Max. number of OSPF autonomous system boundary routers in the AS | R | 4 | |
| OSPF maximum external routes | Max. number of OSPF external routes in the AS | R | 50 | |

(continued on next page)

**Table A–21 (Cont.) Configuration Information: Database Sizing**

| Information Required | Notes | R(equired) /O(ptional) | Default | Your Value |
|---|---|---|---|---|
| OSPF average external connectivity | Average number of discrete forwarding addresses provided by each boundary router for OSPF external routes | R | 2 | |
| OSPF maximum destinations | Max. number of destinations the OSPF protocol can have in the DECNIS routing table | R | 300 | |
| OSPF maximum adjacencies | Max. number of adjacencies the OSPF protocol can form | R | 25 | |

# B

# Configuration and Load Files

This appendix contains tables listing the files created by the DECNIS text-based configurator, and the files loaded to the DECNIS.

In these tables, *client-name* is the load client name of the DECNIS.

**PC File Names**

In Table B–1 and Table B–4, *install-directory* is the installation directory. All other directories listed are subdirectories of the installation directory.

## B.1 Files Created by the Configurators

Table B–1, Table B–2 and Table B–3 give the file names and locations of the DECNIS NCL script files, configuration load files, log files, data files and dump files for MS–DOS PC, OpenVMS, and Digital UNIX load hosts, respectively.

**Table B–1 DECNIS File Names on MS–DOS PCs**

| File | Location | File Name |
|---|---|---|
| Master NCL script file | *install-directory*\CLIENTS*client-name* | NCLSCRPT.NCL |
| Master NCL script file for deleted DECNIS | *install-directory*\CLIENTS*client-name* | NCLSCRPT.OLD |
| CREATE user NCL script file | *install-directory* \CLIENTS*client-name* | CREATE.NCL |
| SET user NCL script file | *install-directory*\CLIENTS*client-name* | SET.NCL |
| ENABLE user NCL script file | *install-directory*\CLIENTS*client-name* | ENABLE.NCL |
| CMIP file | *install-directory*\CLIENTS*client-name* | SCRIPT |
| CMIP file for deleted DECNIS | *install-directory*\CLIENTS*client-name* | SCRIPT.OLD |
| Combined load file | *install-directory*\CLIENTS*client-name* | SYSTEM |
| Log file for CMIP conversion | *install-directory*\CLIENTS *client-name* | NCLSCRPT.LOG |
| Log file for configurator errors | *install-directory*\NIS | NIS_DNIS.LOG |
| DECNIS data file | *install-directory*\CLIENTS*client-name* | *client-name*.DAT |
| DECNIS data file for deleted DECNIS | *install-directory*\CLIENTS*client-name* | *client-name*.OLD |
| Previous DECNIS data file | *install-directory*\CLIENTS*client-name* | *client-name*.BAK[1] |
| DECNIS data file for last software version | *install-directory*\CLIENTS*client-name* | *client-name*.V*nn*[1] |
| Load-host data file | *install-directory*\NIS | NIS_HCFG.DAT |
| DECNIS dump file | *install-directory*\CLIENTS*client-name* | DUMP |

[1]See Section 5.3.5.1 for details.

**Table B–2 DECNIS File Names on OpenVMS Systems**

| File | Location | File Name |
|---|---|---|
| Master NCL script file | SYS$COMMON:[MOM$SYSTEM] | NIS_*client-name*.NCL |
| Master NCL script file for deleted DECNIS | SYS$COMMON:[MOM$SYSTEM] | NIS_*client-name*.NCL_OLD |
| CREATE NCL script file | SYS$COMMON:[MOM$SYSTEM] | NIS_*client-name*_EXTRA_CREATE.NCL |
| SET NCL script file | SYS$COMMON:[MOM$SYSTEM] | NIS_*client-name*_EXTRA_SET.NCL |
| ENABLE NCL script file | SYS$COMMON:[MOM$SYSTEM] | NIS_*client-name*_EXTRA_ENABLE.NCL |
| CMIP file | SYS$COMMON:[MOM$SYSTEM] | NIS_*client-name*.CMIP |
| CMIP file for deleted DECNIS | SYS$COMMON:[MOM$SYSTEM] | NIS_*client-name*.CMIP_OLD |
| Combined file | SYS$COMMON:[MOM$SYSTEM] | NIS040_*client-name*.SYS |
| Log file for CMIP conversion | MOM$SYSTEM | NIS_*client-name*.LOG |
| Log file for configurator errors | MOM$SYSTEM | NIS_DECNIS.LOG |
| DECNIS data file | SYS$COMMON:[MOM$SYSTEM] | NIS_*client-name*.DAT |
| DECNIS data file for deleted DECNIS | SYS$COMMON:[MOM$SYSTEM] | NIS_*client-name*.DAT_OLD |
| Previous DECNIS data file | SYS$COMMON:[MOM$SYSTEM] | NIS_*client-name*.BAK[1] |
| DECNIS data file for last software version | SYS$COMMON:[MOM$SYSTEM] | NIS_*client-name*.DAT_V*nn*[1] |
| Load-host data file | SYS$COMMON:[MOM$SYSTEM] | NIS_HOST_CONFIG.DAT |
| DECNIS dump file | SYS$COMMON:[MOM$SYSTEM] | NIS_*client-name*.DMP |

[1] See Section 5.3.5.1 for details.

**Table B–3 DECNIS File Names on Digital UNIX Systems**

| File | Location | File Name |
|---|---|---|
| Master NCL script | /usr/lib/dnet | nis_*client-name*.ncl |
| Master NCL script for deleted DECNIS | /usr/lib/dnet | nis_*client-name*.ncl_old |
| CREATE NCL script file | /usr/lib/dnet | nis_*client-name*_extra_create.ncl |
| SET NCL script file | /usr/lib/dnet | nis_*client-name*_extra_set.ncl |
| ENABLE NCL script file | /usr/lib/dnet | nis_*client-name*_extra_enable.ncl |
| CMIP file | /usr/lib/mop | nis_*client-name*.cmip |
| CMIP file for deleted DECNIS | /usr/lib/mop | nis_*client-name*.cmip_old |
| Combined file | /usr/lib/mop | nis040_*client-name*.sys |
| Log file for CMIP conversion | /usr/lib/dnet | nis_*client-name*.log |
| Log file for NCL checking | /usr/lib/dnet | nis_*client-name*.lis |
| Log file for configurator errors | /usr/lib/dnet | nis_decnis.log |
| DECNIS data file | /usr/lib/dnet | nis_*client-name*.dat |
| DECNIS data file for deleted DECNIS | /usr/lib/dnet | nis_*client-name*.dat_old |
| Previous DECNIS data file | /usr/lib/dnet | nis_*client-name*.bak[1] |
| DECNIS data file for last software version | /usr/lib/dnet | nis_*client-name*.dat_v*nn*[1] |
| Load-host data file | /usr/lib/dnet | nis_host_config.dat |
| DECNIS dump file | /usr/lib/mop | nis_*client-name*.dmp |

[1]See Section 5.3.5.1 for details.

## B.2 DECNIS Load Files

Table B–4, Table B–6 and Table B–7 show the names and locations of the files loaded to the DECNIS on PC, OpenVMS, and Digital UNIX load hosts, respectively.

**Table B–4 Files Loaded from PC Load Hosts**

| File Name | Description |
|---|---|
| **Loading from a Load Host Selected** | |
| *install-directory*\COMMON\NISV40\SYSTEM | System image |
| *install-directory*\CLIENTS*client-name*\SCRIPT | CMIP file |
| *install-directory*\COMMON\MCNM_PRF | Modem Connect profile file (only if serial lines configured) |
| *install-directory*\COMMON\X25L2_PR<br>*install-directory*\COMMON\X25L3_PR | X.25 profile files (only if lines configured for X.25) |
| **Loading from Nonvolatile Memory Selected** | |
| *install-directory*\CLIENTS*client-name*\SYSTEM | Combined file |

**Table B–5 Files Loaded from Windows 95/NT Load Hosts**

| File Name | Description |
|---|---|
| *install-directory*\COMMON\NISV40\NIS040.SYS | System image |
| *install-directory*\CLIENTS\ *client-name**client-name*.CMP | CMIP file |
| *install-directory*\COMMON\MCNM_PRF | Modem Connect profile file (only if serial lines configured) |
| *install-directory*\COMMON\X25L2_PR<br>*install-directory*\COMMON\X25L3_PR | X.25 profile files (only if lines configured for X.25) |
| *install-directory*\CLIENTS*client-name**client-name*.SYS | Combined file |
| *install-directory*\CLIENTS*client-name**client-name*.DAT | Secure Connections Rules |

Refer to Section 5.4.3 and Section 5.4.4 for more information.

**Table B–6 Files Loaded from OpenVMS Load Hosts**

| File Name | Description |
|---|---|
| **Loading from a Load Host Selected** | |
| SYS$COMMON:[MOM$SYSTEM]NIS040.SYS | System image |
| SYS$COMMON:[MOM$SYSTEM]NIS_*client-name*.CMIP | CMIP file |
| SYS$COMMON:[MOM$SYSTEM]FCNS$MCNM_PRF.DAT | Modem Connect profile file (only if serial lines configured) |
| SYS$COMMON:[MOM$SYSTEM]FCNS$X25L2_PRF.DAT<br>SYS$COMMON:[MOM$SYSTEM]FCNS$X25L3_PRF.DAT | X.25 profile files (only if lines configured for X.25) |
| **Loading from Nonvolatile Memory Selected** | |
| SYS$COMMON:[MOM$SYSTEM]NIS040_*client-name*.SYS | Combined file |

**Table B–7 Files Loaded from Digital UNIX Load Hosts**

| File Name | Description |
|---|---|
| **Loading from a Load Host Selected** | |
| /usr/lib/mop/nis040.sys | System image |
| /usr/lib/mop/nis_*client-name*.cmip | CMIP file |
| /usr/lib/mop/digital/fcns/mcnm_prf | Modem Connect profile file (only if serial lines configured) |
| /usr/lib/mop/digital/fcns/x25l2_prf<br>/usr/lib/mop/digital/fcns/x25l3_prf | X.25 profile files (only if X.25 lines configured) |
| **Loading from Nonvolatile Memory Selected** | |
| /usr/lib/mop/nis040_*client-name*.sys | Combined file |

## B.3 Location of DECNIS Load Files on Digital UNIX BOOTP Load Hosts

This section specifies the directory and filenames that need to be in the /etc /bootptab file in order for Digital UNIX BOOTP load hosts to respond correctly to load and dump requests from the DECNIS.

### B.3.1 Directory Used for Storing Load Files

The BOOTP client database /etc/bootptab, specifies where the BOOTP load files are stored.

On Digital UNIX systems, by default, the load directory listed in /etc/bootptab is /usr/local/bootfiles.

However, the DECNIS load files are not actually installed in the directory specified in /etc/bootptab. Instead, they are installed in the directories required for MOP loading (see Table B–3).

#### B.3.1.1 Softlinks Automatically Created

The load-host configurator automatically sets up softlinks from the files in the MOP directories to the file names and directory in /etc/bootptab. This allows the same files to be loaded no matter which protocol is specified.

### B.3.2 Load files on Non-Digital BOOTP Load Hosts

If your BOOTP load host is a non-Digital UNIX system, note the following:

- If the BOOTP/TFTP implementation is compatible with that on Digital UNIX load hosts, you must use the file names in Table B–8.
- If the BOOTP/TFTP implementation is not compatible with that on Digital UNIX load hosts, you do not need to use these file names. Refer to the load host documentation for details of file names and directories.

### B.3.3 File Names Required on Digital UNIX BOOTP Load Hosts

Table B–8 lists the file names required for BOOTP loading, and the files in the MOP directories to which they are linked. Note that the load-host configurator automatically places the correct entries in /etc/bootptab.

In the table, *client-name* is the BOOTP load client name of the DECNIS. Digital recommends that you make the BOOTP load client name the same as the MOP client name (the load-host configurator does this automatically).

The directory for the files listed in column 1 is the one specified in etc/bootptab.

**Table B–8 File Names Required for Digital UNIX BOOTP Loading**

| BOOTP File Name | Linked to this MOP file | Description |
|---|---|---|
| **Loading from a Load Host Selected** | | |
| system.*client-name* | /usr/lib/mop/nis040.sys | System image |
| script.*client-name* | /usr/lib/mop/nis_*client-name*.cmip | CMIP file |
| mcnm_prf | /usr/lib/mop/digital/fcns/mcnm_prf | Modem connect profile file |
| x25l2_prf<br>x25l3_prf | /usr/lib/mop/digital/fcns/x25l2_prf<br>/usr/lib/mop/digital/fcns/x25l3_prf | X.25 profile files |
| **Loading from Nonvolatile Memory Selected** | | |
| system.*client-name* | /usr/lib/mop/nis040_*client-name*.sys | Combined file |

# C

# Using DECdns and the Local Namespace with the DECNIS

Use this appendix if the DECnet/OSI naming services, DECdns and/or the local namespace, are used on your network.

## C.1 Introduction

The DECNIS router does not use the DECnet/OSI naming services to find the location of the DECnet systems to which it sends messages (for example, event sinks). Instead, it uses node specifications contained in the master NCL script file generated by the DECNIS configurator.

However, the load-host configurator and the DECNIS text-based configurator do make use of the naming services. If you specify in the load-host configurator that you want to use a naming service, then the configurators will do the following:

- Register the DECNIS node in the local or DECdns namespace.
- Use the local or DECdns namespace to find the node specifications of the systems to which the DECNIS sends messages. It then uses those specifications to write NCL SESSION CONTROL KNOWN TOWERS commands for these systems in the DECNIS NCL script.

Note that the clearVISN DECNIS configurator does not make use of the naming services.

## C.2 Specifying the Use of a Naming Service

In the load-host configurator, you are asked whether or not you want to use a naming service to generate node specifications.

If you select Yes, then you are asked for the following:

- The node name of the DECNIS.
- The node synonym of the DECNIS. This is optional.

If you select No, you are not asked for a node name or synonym, as the configurator will not then register the DECNIS in a namespace.

Refer to the manual *DECNIS Installation and Configuration for OpenVMS and Digital UNIX* for details.

## C.3 DECdns or Local Namespace?

When you request the use of a naming service, the load-host configurator needs to decide whether to use the local namespace or the DECdns namespace.

By default, the local namespace is used.

However, you can override the default so that the configurators use the DECdns namespace. To do this, insert the following command in the decnet_register initialization command file:

```
SET DEFAULT DIRECTORY_SERVICE DECdns
```

You must do this **before** you run the load-host and DECNIS configurators.

See the manual *DECNIS Installation and Configuration for OpenVMS and Digital UNIX* for more information.

## C.4 Registering the DECNIS in a Namespace

The configurators use the DECnet/OSI utility **decnet_register** as follows:

- The load-host configurator uses the utility to register the DECNIS in a namespace.
- The DECNIS text-based configurator uses the utility to update the DECNIS node object in the namespace with the DECNIS address towers.

See the manual *DECnet/OSI Network Management* for more information about decnet_register.

### C.4.1 Requirements for Successful Registration

In order for the load-host and DECNIS configurators to successfully create and update the DECNIS object in a namespace, writeable access to the parent directory where the object is going to reside must be available.

### C.4.2 Errors When Registering the DECNIS in the Namespace

If there are any errors when the load-host configurator is trying to register the DECNIS, the following will be displayed:

```
The node name listed below could not be registered in the namespace,
press RETURN.
```

```
Node name : namespace_name:.nodename
```

where: *namespace_name* is the name of the namespace and *.nodename* is the fullname of the DECNIS.

### C.4.3 Adding Tower Sets to the DECNIS Entry

You are asked in the DECNIS text-based configurator if you want addressing information for the DECNIS to be added to the namespace. If you select Yes, the configurator will try to add the DECNIS tower set(s) to its namespace entry.

However, if you know that writeable copies of the relevant DECdns directories will not be available when the DECNIS text-based configurator is run, you should answer No to this question. The naming service commands will not then be issued, and the DECNIS configurator will go on to create the NCL script file.

When the directories become available, rerun the DECNIS text-based configurator and select Yes to the question.

### C.4.4 Errors When Adding Tower Sets to the DECNIS Entry

If the configurator cannot update the DECNIS namespace entry with tower sets for any reason, the following will be displayed:

```
Cannot add tower sets for this DECNIS.
```

```
Look in the log file log-file-name for details.
```

where *log-file-name* is the name of the log file; see Table C–1 for details.

If this error is displayed, the rest of the configuration process should complete, but there will be no address information for the DECNIS in the namespace.

## C.5 Naming Service Error Log File

If the load-host configurator cannot register the DECNIS, or the DECNIS text-based configurator cannot update the DECNIS namespace entry, any errors will be written to the log file shown in Table C–1.

**Table C–1 Naming Service Error Log Files**

| Load Host | Log File Name |
|---|---|
| OpenVMS | MOM$SYSTEM:NIS_*client-name*_DNS.LOG |
| Digital UNIX | usr/lib/dnet/nis_*client-name*_dns.log |

where *client-name* is the load client name of the DECNIS.

## C.6 Completing Name Service Registration

If the naming service registration of node name and address information fails, then you can do one of the following:

- Rerun the configurators when the master copy of the naming service directories are available. If the path to these directories is through the DECNIS that is being configured, then it may be necessary to boot the DECNIS and manage it as described in Section C.6.1 before the registration can be completed.
- Use the decnet_register utility to register the DECNIS, as described in the manual *DECnet/OSI Network Management*.

### C.6.1 Managing the DECNIS Before Registration Is Complete

It is not possible to manage a DECNIS using its node name if you are using a naming service to store node information and the naming service registration is not complete. In these circumstances, you need to specify the address of the DECNIS in NCL commands until the DECNIS has been registered.

For example:

- To manage a DECNIS with a Phase IV compatible address 1.3, you could issue the following command:

```
ncl> show node 1.3 all attributes
```

- To manage a DECNIS with only extended NSAP addresses, one of which was 49::98-76:08-00-2b-00-12-34:20, then you could issue the following command:

```
ncl> SHOW NODE %x49987608002B00123420 ALL ATTRIBUTES
```

## C.7 Swapping the DECNIS Hardware

If you replace the DECNIS hardware unit with another unit, and the DECNIS is using any extended NSAP addresses, you must do the following before you attempt to downline load:

- Run the load-host configurator and change the hardware address for the DECNIS.
- Run the DECNIS text-based configurator, and create a new NCL script and CMIP or combined file. The configurator will update the namespace entry with a new NSAP address which contains an ID field based on the new hardware address.

# D

# Loading from a DECnet–VAX Phase IV Load Host

## D.1 Introduction

This chapter describes how to set up a DECnet–VAX Phase IV system as a MOP load host for a DECNIS.

**Definition of DECnet–VAX Phase IV**

The term DECnet–VAX Phase IV refers to the version of DECnet–VAX that preceded DECnet–VAX Extensions.

## D.2 Procedure

To set up a DECnet–VAX Phase IV system for MOP loading, follow these steps:

1. Install the DECNIS software on a DECnet/OSI system.
2. Run the configurators. In the NCL Script section of the DECNIS text-based configurator, create a CMIP file or a combined image/CMIP/profile file.
3. Copy the load files (either the combined file or the software image, CMIP file and profile files) to the DECnet–VAX Phase IV system.

   See Table B–6 for the file locations.

Now, on the DECnet–VAX Phase IV system, follow these steps:

1. Run NCP by entering the following:

   ```
   $ RUN SYS$SYSTEM:NCP
   ```

2. Enter these commands:

   ```
   NCP> CLEAR NODE client-name ALL
   NCP> PURGE NODE client-name ALL
   ```

3. Enter the DECnet address of the DECNIS:

   ```
   NCP> DEFINE NODE client-name ADDRESS decnet-address
   ```

4. Enter the hardware address of the DECNIS. Section D.2.1 explains how to find the correct hardware address.

   ```
   NCP> DEFINE NODE client-name HARDWARE ADDRESS hardware-address
   ```

5. Enter the name of the Ethernet circuit to be used for downline loading:

   ```
   NCP> DEFINE NODE client-name SERVICE CIRCUIT circuit-id
   ```

6. Check that the service circuit is enabled. If it is disabled, do the following:

   a. Turn the circuit off by entering:

      ```
      NCP> SET CIRCUIT circuit-id STATE OFF
      ```

   b. Enable the service by entering:

      ```
      NCP> DEFINE CIRCUIT circuit-id SERVICE ENABLED
      NCP> SET CIRCUIT circuit-id SERVICE ENABLED
      ```

   c. Now turn the circuit on again by entering:

      ```
      NCP> SET CIRCUIT circuit-id STATE ON
      ```

7. To specify the software image, CMIP file and profile files, enter the following commands. To specify the combined file, go to Step 8.

   ```
   NCP> DEFINE NODE client-name LOAD FILE -
   _NCP> sys$common:[mom$system]nis040.SYS)
   NCP> DEFINE NODE client-name MANAGEMENT FILE -
   _NCP> sys$common:[mom$system]nis_client-name.cmip
   ```

8. To specify a combined file, enter the following:

   ```
   NCP> DEFINE NODE client-name LOAD FILE -
   _NCP> sys$common:[mom$system]nis040_client-name.sys
   ```

9. To define the DECNIS dump file, enter:

   ```
   NCP> DEFINE NODE client-name DUMP FILE -
   _NCP> sys$common:[mom$system]nis_client-name.dmp
   ```

10. Enter the following:

    ```
    NCP> SET NODE client-name ALL
    NCP> EXIT
    ```

In these commands, *client-name* is the load client name of the DECNIS.

## D.2.1 Hardware Address

The DECNIS has 16 hardware addresses available. It assigns one hardware address to each port on its CSMA/CD and FDDI Network Interface Cards. It uses a standard scheme to do this.

When installing from a DECnet–VAX Phase IV system, you must specify the hardware address of the port on the DECNIS which will receive the downline load. Section D.2.1.1 to Section D.2.1.3 describe how to do this.

### D.2.1.1 Finding the Hardware Address

The first 11 digits of the DECNIS hardware addresses are the same. The last digit depends on the type of Network Interface Card, its slot number, and the port number. Table D–1 shows how the last digit is assigned.

**Table D–1 Hardware Address Assignment**

| Card | Last Digit | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| L601 | The number of the slot in which the card is inserted | | | | | | | |
| L602 | Assigned as follows: | | | | | | | |
| | **If slot number is:** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
| | The last digit for Port 0 is: | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| | The last digit for Port 1 is: | A | B | C | D | E | F | 2 |
| F621 | DECNIS 600: the higher number of the two slots<br>DECNIS 500: the lower number of the two slots | | | | | | | |

### D.2.1.2 Procedure

To find the correct hardware address, follow these steps:

1. Take the first 11 digits of the hardware address from the hardware address on the label on the DECNIS Processor Card. Ignore the last digit (which is always zero on the label).
2. Decide which port on which card will be used to receive the downline load.
3. Use Table D–1 to find the correct last digit for the port.

### D.2.1.3 Example

Table D–2 shows examples of hardware address assignment on a DECNIS 600.

**Table D–2 Examples of Hardware Address Assignment**

| Address on Processor Card Label | Card | Slot | Port | Hardware Address |
|---|---|---|---|---|
| 08-00-2B-C3-66-12-50 | L601 | 5 | 0 | 08-00-2B-C3-66-12-55 |
| 08-00-2B-C3-66-12-50 | L602 | 5 | 1 | 08-00-2B-C3-66-12-5C |
| 08-00-2B-D4-76-22-80 | F621 | 6 and 7 | 0 | 08-00-2B-D4-76-22-87 |

## D.2.2 Example

This section gives example commands to configure a DECnet–VAX Phase IV system as a DECNIS load host. In this example, the following information is available:

| | |
|---|---|
| DECNIS load client name | SOUTH1 |
| DECnet address | 44.6 |
| Hardware address | 08-00-2B-0A-11-33 |
| Service circuit | SVA-0 |
| Combined file | SYS$COMMON:[MOM$SYSTEM]NIS040_SOUTH1.SYS |
| Dump file | SYS$COMMON:[MOM$SYSTEM]NIS040_SOUTH1.DMP |

**Example Commands**

```
$ RUN SYS$SYSTEM:NCP
CLEAR NODE south1 ALL
PURGE NODE south1 ALL
DEFINE NODE south1 ADDRESS 44.6
DEFINE NODE south1 HARDWARE ADDRESS 08-00-2B-0A-11-33
DEFINE NODE south1 SERVICE CIRCUIT sva-0
SET CIRCUIT sva-0 STATE OFF
DEFINE CIRCUIT sva-0 SERVICE ENABLED
SET CIRCUIT sva-0 SERVICE ENABLED
SET CIRCUIT sva-0 STATE ON
DEFINE NODE south1 LOAD FILE sys$common:[mom$system]nis040_south1.sys
DEFINE NODE south1 DUMP FILE sys$common:[mom$system]nis040_south1.dmp
SET NODE south1 ALL
EXIT
```

# Index

## D

## E



## F

## G



## H



## I

## K



## L

## M



## N

## O



## P

## Q



## R



## S

## T



## U



## V



## W



## X