# TOPS–10
# ANF–10 Networks Software Installation Guide

AA–W557A–TB

**July 1984**

This manual provides information on the assembly, installation, and use of ANF–10 networks software.

This manual supersedes the *TOPS–10 Networks Software Installation Guide*, AA–5156E–TB and its update AD–5156E–T1.

**OPERATING SYSTEM:** TOPS–10 V7.02

Software and manuals should be ordered by title and order number. In the United States, send orders to the nearest distribution center. Outside the United States, orders should be directed to the nearest DIGITAL Field Sales Office or representative.

**Northeast/Mid–Atlantic Region**

Digital Equipment Corporation
PO Box CS2008
Nashua, New Hampshire 03061
Telephone:(603)884–6660

**Central Region**

Digital Equipment Corporation
Accessories and Supplies Center
1050 East Remington Road
Schaumburg, Illinois 60195
Telephone:(312)640–5612

**Western Region**

Digital Equipment Corporation
Accessories and Supplies Center
632 Caribbean Drive
Sunnyvale, California 94086
Telephone:(408)734–4915

digital equipment corporation • marlboro, massachusetts

**First Printing, July 1984**



The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license and may only be used or copied in accordance with the terms of such license.

No responsibility is assumed for the use or reliability of software on equipment that is not supplied by DIGITAL or its affiliated companies.

The following are trademarks of Digital Equipment Corporation:

digital™

| | | |
|---|---|---|
| DEC | MASSBUS | RSX |
| DECmate | PDP | RT |
| DECsystem–10 | P/OS | UNIBUS |
| DECSYSTEM–20 | Professional | VAX |
| DECUS | Q–BUS | VMS |
| DECwriter | Rainbow | VT |
| DIBOL | RSTS | Work Processor |

The postage-prepaid READER'S COMMENTS form on the last page of this document requests the user's critical evaluation to assist us in preparing future documentation.

CONTENTS

FIGURES



TABLES

## PREFACE

This manual is intended for software installers and system operators responsible for installing the Advanced Network Functions software for the TOPS-10 operating system (TOPS-10 ANF-10) on communications front ends and remote nodes. It assumes that the reader knows how to

- log in on a TOPS-10 host
- use BACKUP to restore data from tapes to disk
- use an editor such as SOS or TECO
- run MONGEN

In order to install network software, you must have a running TOPS-10 operating system. Use the instructions in the TOPS-10 Software Installation Guide before using this manual. For information on installing TOPS-10 DECnet-10 network software, see the manual entitled DECnet-10 Network Generation and Installation Procedures. For information on installing IBM communications software, see the manual entitled TOPS-10 IBM Emulation/Termination.

Not all the hardware mentioned in this manual is currently supported by TOPS-10 ANF-10. For current support status of hardware and software, please refer to the TOPS-10 Version 7.02 Software Product Description (SPD).

This publication does not cover the design of a network. It assumes that the network topology has already been decided upon and that the hardware and DIGITAL-supplied software are available.

This installation guide is organized as follows:

- Chapter 1 provides an introduction to TOPS-10 ANF-10 network configurations and an overview of the installation procedures.
- Chapter 2 describes the DIGITAL-supplied software and the procedures for copying it to your system's storage area.
- Chapter 3 describes the generation of a configuration file for each node in the network.
- Chapter 4 describes the assembly of the software and the selection of the source modules used as input to the assembly.

- Chapter 5 describes the loading procedures for the communications front ends and the remote nodes.
- Chapter 6 describes the initial hardware check as the node is loaded and started.
- Appendix A contains a comprehensive list of all available configuration switches.

Documents referenced in this manual or useful during the installation and checkout procedures are:

TOPS-10 Software Installation Guide

TOPS-10 Operating System Commands Manual

TOPS-10 Operator's Guide

DECnet-10 Network Generation and Installation Procedures

BOOT11 Specification

DTELDR Specification

INITIA Specification

KDPLDR Specification

NETLDR Specification

REACT Specification

TOPS-10 Remote Station Guide

TOPS-10 CHK11 Manual

TOPS-10/TOPS-20 DDT11 Manual

TOPS-10 IBM Emulation/Termination

DN92 User's Guide

PDP-11 Peripherals Handbooks (1976)

Terminals and Communications Handbook (1979)

Large Systems Product Summary (1980)

## NOTATION USED IN THIS MANUAL

| Abbreviation | Meaning |
|---|---|
| addr | memory address |
| bll | type ball for 2741-type terminals |
| c | memory location (lower bound) |
| CPUtype | type of processor (11 for PDP-11 or 8 for PDP-8) |
| CTLR | KMC11 controller number |
| CTY | console terminal |
| d | memory location (upper bound) |
| len | argument length (NSPLST macro) |
| mn | PDP-11 model number (34, 40) |
| mmm | TTYmmm: terminal number |
| mx | maximum |
| n | asynchronous line number (0 to 177 octal) |
| nn | node address (1 to 77 octal) |
| nodeid | either node name or node number |
| nodename | node name, up to 6 alphanumeric characters, starting with an alphabetic character |
| portnum | port number (0 to 7) |
| PTY | pseudoterminal |
| RET | the RETURN key |
| s | speed of line (baud rate) |
| ser | serial number of remote CPU |
| syn# | synchronous line number (0 to 12) |
| typ | type of node (for example, D82, 87S) |
| type | node type (for example, DN82, DN92) |
| tt | front end type by number only (for example, 20, 87) |
| TTY | terminal |

| | |
|---|---|
| w | width of line (in columns) |
| x | CPU number (DTELDR) |
| xxx | feature identifier (for example, RNN,TAB) |
| y | DTE number (DTELDR) |
| #as | number of available asynchronous lines (0 to 177 octal; TTYN equals this value) |
| #sy | number of synchronous lines (0 to 12 octal; NLINES equals this value) |

# CHAPTER 1

# THE TOPS-10 ANF-10 NETWORK

This chapter provides an overview of the TOPS-10 Advanced Network Functions (ANF-10) network hardware and an outline of the software installation procedures.

## 1.1 NETWORK OVERVIEW

The TOPS-10 ANF-10 network is a configuration of TOPS-10 central processors (hosts), communications control systems (front ends), and remote stations interconnected over communications lines. The lines, also called links, can be as short as a few feet when connecting a front end to its central processor, or many miles long when connecting remote stations to the central site over telephone lines or radio relay links.

DECnet-10, a separate product, allows a TOPS-10 system to be part of a DECnet network and to communicate with VAX/VMS, TOPS-20, RSX-11, RSTS, and RT systems. DECnet-10 also replaces the DECnet Compatible Port (DCP).

The computer systems that attach to a line are called nodes, whether hosts, front ends, or remote stations. Each node is identified by a unique node number and a unique node name. The node number is a nonzero two-digit octal number, limiting the maximum number of nodes in a TOPS-10 ANF-10 network to 63 (77 octal). The node name must begin with a letter and can contain up to six alphanumeric characters.

### 1.1.1 Host Processors

When you create a monitor that is to support a TOPS-10 ANF-10 network, you must run MONGEN and answer the network-related questions in its HDWGEN, TTYGEN, NETGEN, and FGEN segments appropriately. Those questions are described below.

In the **HDWGEN** segment, respond to the network-related prompts as follows:

| Prompt | Response |
|---|---|
| Allow jobs to be locked in core? | Y |
| # high priority queues (0,0-15) | Use a value greater than 0 so that you can set a HPQ switch in NETLDR.INI. |
| MSGSER (Y,N) | Y to use DTELDR. (You must answer this question with Y if you have nodes connected to DTEs, such as the DN20 or DN87S.) |
| # PTYs (20,0-510) | Use any value in the allowed range. (The total of PTYs plus TTYs declared in all MONGEN segments must not exceed 511.) |

In the **TTYGEN** segment of MONGEN, specify that lines 0-777 are LINES WHICH RUN INITIA AT STARTUP. For lines on your console front end, and lines on any 2020 (KS10), you must also specify their line speeds (and other characteristics) in TTY.INI.

In the **NETGEN** segment of MONGEN, specify that you want network support and give information on the following:

number of local CPUs:

number of DL10-interfaced nodes on CPU0:

number of DTE20-interfaced nodes on each CPU:

host node number:

host node name:

number of remote TTYs:

whether you have

network virtual terminals:

remote card readers:

remote line printers:

remote data entry terminals:

remote task-to-task:

number of connects:

See the NETGEN dialogue in the TOPS-10 Software Installation Guide for more information.

In the **FGEN** segment of MONGEN, specify either KIFULL, KLFULL, or KSFULL to the Feature Set prompt. Only these monitors support TOPS-10 ANF-10 networks.

If you have a 2020 (KS10), you need only run MONGEN, and reassemble and load your monitor to place your processor as a host in a TOPS-10 ANF-10 network. All the communications software resides in the 2020 itself. Your 2020 can only be connected to a network over a synchronous line, and it cannot support a DN92. Whenever you power up the system, the KMC used in the KS10 may need to be reloaded. The monitor runs KDPLDR for this purpose.

You can specify the characteristics of terminals in your network in TTY.INI. For example, the following entries in SYS:TTY.INI set the characteristics for TTYs on two network nodes:

```
NOVA_TTY13-15:          TYPE:VT52 PAGE:0

CTCH22_TTY1-40:         TYPE:VT52 FILL:0 TAB WIDTH:80
```

The node name and TTY number must be joined by an underline (_) in your TTY.INI file. If you do not assemble line speeds into your communications front end software and you do not want your lines to autobaud, you should use TTY.INI to set line speeds for your terminals. Always use TTY.INI to set line speeds for terminals on your console front end (the PDP-11 that runs RSX-20F on your 1091) and on the KS10. For more information on the characteristics you can set in TTY.INI, see the TOPS-10 Software Installation Guide. For more information on setting line speeds in your front-end software (the preferred method for non-KS10 nodes), see Chapter 3 in this manual.

### 1.1.2 Communications Front Ends

A communications front end is a dedicated communications processor that acts as an interface between a central processing system and a network. The software that runs in a communications front end is tailored to the protocol (or set of rules) governing the transfer of information within that particular network.

A communications front end is required for each TOPS-10 KI10 or KL10 host that is part of a network. A front end with asynchronous line support can relieve the host processor of a significant portion of the processing requirements for terminals. TOPS-10 communications processors are the DN85 Synchronous Front End and the DN20, DN87, and DN87S Universal Synchronous/Asynchronous Front Ends (see Table 1-1).

### 1.1.3 Remote Stations

A remote station is a small computer system that allows access to the network from locations that are distant from a central processing system. Remote stations usually support three classes of input/output devices: terminals (hard copy or video), line printers, and card readers. A remote station also runs software supporting the communications protocol of the network. Table 1-1 describes the characteristics of the remote stations that you may have in your ANF-10 network. All the nodes listed in this table are PDP-11-based, except for the DN92, which is PDP-8-based.

Every remote station has an operator's console. To have TOPS-10 recognize this remote operator's console, the TOPS-10 system administrator must run REACT to specify the remote node as having remote operator privileges. For more information on REACT, see the REACT Specification in the TOPS-10 Software Notebook Set.

**Table 1-1: Characteristics of Communications Nodes**

| Node Type | Node Use | No. of Synch. Lines | No. of Asynch. Lines | Printer | Card Reader |
|---|---|---|---|---|---|
| DN20 | Front End[1] | 0-12 | 0-128 | 0 | 0 |
| DN82 | RJE & Conc. | 1-4 | 1-16 | 1 | 1 |
| DN87 | Front End | 0-10 | 0-96 | 0 | 0 |
| DN87S | Front End | 0-12 | 0-128 | 0 | 0 |
| DN92 | RJE & Conc. | 1 | 1-16[2] | 1 | 1 |
| DN200 | RJE & Conc. | 1-2 | 1-32 | 1 | 1 |
| 2020 | Host | 1-2 | 1-32 | 1 | 1 |

[1] The allowable number of lines depends on the mix of high-speed synchronous lines, low-speed synchronous lines, asynchronous lines, and activated feature-test switches.

[2] The DN92 can have one of the following combinations: 16 asynchronous lines, or 1 printer or card reader with 12 asynchronous lines, or 1 printer with 1 card reader and 8 asynchronous lines.

### 1.1.4 Network Configurations

Network configurations (topologies) are generally determined by the

- geographical distribution of the nodes
- volume and scheduling of communications traffic
- cost of lines and hardware

Configurations supported by TOPS-10 ANF-10 include simple network topologies such as point-to-point and star. They also include more complex multilink and multipath configurations with features such as multiple hosts and dynamic topologies.

Complex topologies are composed of multiple links and multiple nodes. A multilink configuration can include multiple TOPS-10 hosts in the same network. A multihost configuration permits the user at a terminal of a remote station to select his host with the SET HOST command. The route-through capability allows communication between two nodes that are indirectly connected through one or more intermediate nodes (for example, see nodes FOUR and SEVEN in Figure 1-1).

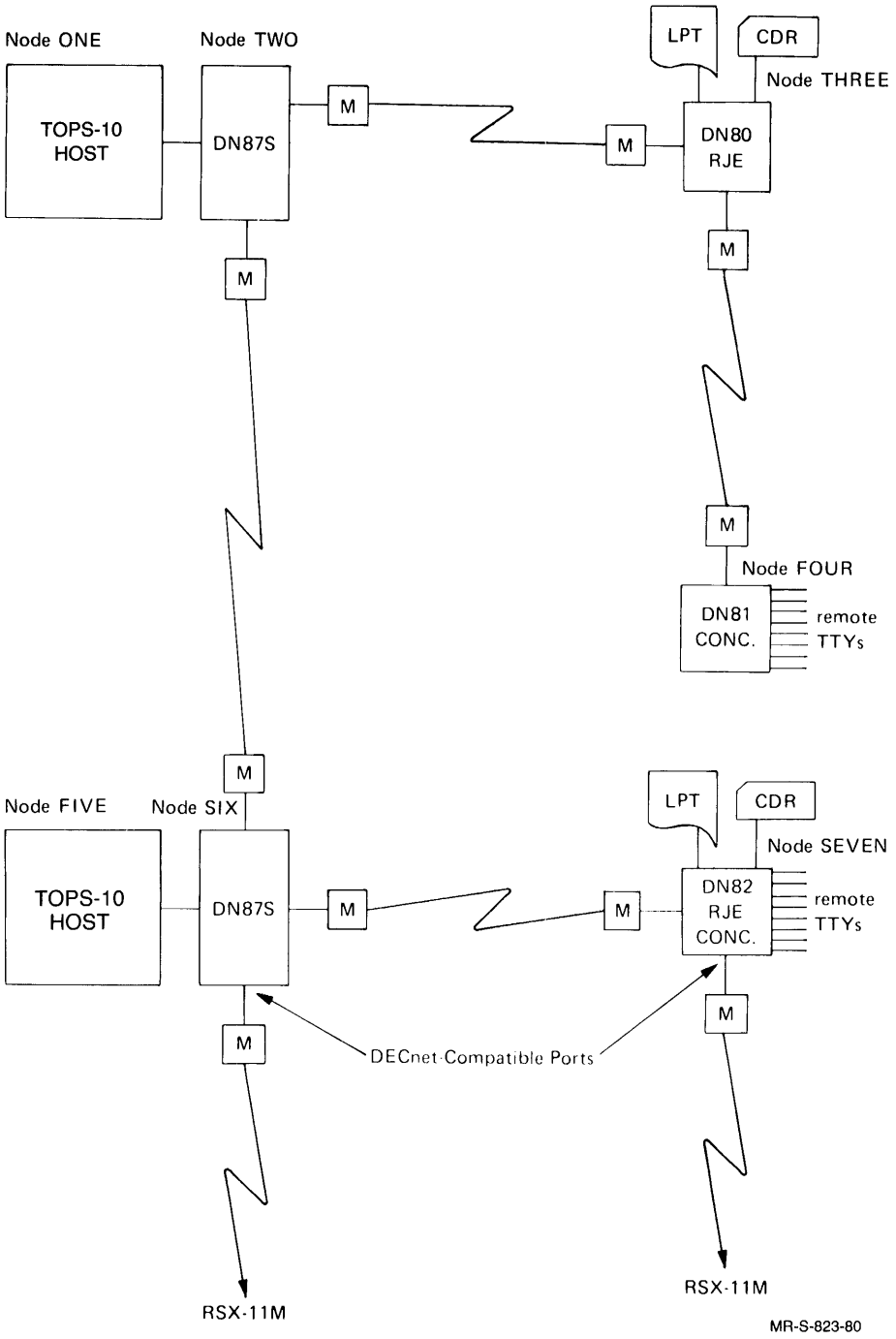


Figure 1-1: Multilink Configuration

## 1.2 INSTALLATION OVERVIEW

This manual describes the generation and installation of ANF-10 network software. You must have a fully operational TOPS-10 monitor and be familiar with its use to perform the procedures contained in the following chapters. You may need to create or update a new TOPS-10 monitor to support TOPS-10 ANF-10 network software. To create or update a TOPS-10 operating system, please refer to the TOPS-10 Software Installation Guide before following the instructions in this manual.

### 1.2.1 Network Installation Requirements

To perform the appropriate network generation and installation procedures, you should have access to the system programs listed in Table 1-2 that apply to your system.

**Table 1-2: System Software**

| Program Name | Used to Install | Function |
|---|---|---|
| BACKUP | all nodes | Copies network software from the distribution tape to the system disk. |
| BOOT11 | DN87 | Loads network software into a communications front end over a DL10. |
| CREF | DN92 | Prepares a cross-reference listing, following assembly of software. |
| DDT11 | all nodes except 2020 | Allows you to examine and deposit code and data in a running PDP-8 or PDP-11 node. You can also use it to read memory dumps from these machines. |
| DTELDR | DN20, DN87S | Loads network software into a communications front end over a DTE20. |
| KDPLDR | 2020 | Loads microcode into the KMC. |
| MACDLX | DN20,DN80-series, DN200 | Assembles PDP-11 software on a TOPS-10 host. |
| NETLDR | Remote nodes | Loads software downline into a remote node over a synchronous line. |
| PAL10 | DN92 | Assembles PDP-8 software on the TOPS-10 host. |

Network programs listed in Table 1-3 are found on the second saveset of the Distribution Tape or on the customer-supported tape.

**Table 1-3: Network Programs**

| Program Name | Used to Install | Function |
|---|---|---|
| source code | all nodes except 2020 | Used to assemble network software for each network node. Source files for DN20, DN200, and DN80-series nodes are *.P11 files; for DN92 nodes, DN92.PAL files. See Section 2.1 for descriptions of these source modules. |
| TSKSER | all host nodes | Provides task-to-task capabilities, such as file-transfer facilities (with PIP) between nodes in the network. |

### 1.2.2 Installation Summary

The network software installation procedure contains six operations:

1. Copy the files from the CUSP tape to your disk area. Do this first so that the updated CUSP files on the main monitor Distribution Tape will supersede the CUSP tape files when you perform step 2.

2. Copy the network software files from the Distribution Tape to your disk area.

3. For each node, create a configuration file that reflects the environment in which the node will operate.

4. Assemble the network software for each node.

5. Load each node with its assembled software. The load procedures vary according to whether the node is remote (for example, a remote station) or local (for example, a communications front end). If the node is local, loading procedures vary according to the type of interface to the TOPS-10 host.

6. Examine your initial system output. This is either CHK11 output (if the node you load is a PDP-11) or SYSCHK output (if the node is a PDP-8). This hardware check program runs whenever a node is loaded and started (but not when it is restarted manually). The space occupied by the check program is reclaimed and used as buffer space when the node is running.

During the software installation of a remote node and subsequently when the node is running, you may need to examine the node software. DDT11 is a remote debugging program that allows you to examine a running node. For information on DDT11, see the TOPS-10/TOPS-20 DDT11 Manual.

# CHAPTER 2

# COPY THE DISTRIBUTED SOFTWARE TO DISK

The procedures in this chapter enable you to copy the DIGITAL-supplied network software from the Distribution Tape to your disk. The network software is used in subsequent chapters to generate customized code for each network node.

## 2.1 NETWORK SOURCE MODULES

When you are ready to assemble the software for each node, you must supply the MACDLX or PAL10 assembler with a list of source modules, a listing file name, a binary file name, and switches.

### 2.1.1 Required PDP-11 Modules

The following modules apply to all PDP-11-based nodes and include all modules in each assembly. The modules are described in the order in which they must appear when you do your assembly.

filename.P11 is the node configuration file you create according to the instructions in Chapter 3. The filename may be C.P11 or representative of the node (for example, CN8222.P11). This file must be the first file specified at assembly time.

S.P11 contains the symbol definitions used by the network software. This file must be the second file specified at assembly time.

MACROS.P11 contains system macro definitions used by the network software. This file must be the third file specified at assembly time.

DNCNFG.P11 processes the configuration parameters and feature test switches that you entered in the filename.P11 file. This file must be the fourth file specified at assembly time.

DNCOMM.P11 contains common data and code such as the main loop and clock routines.

DNNCL.P11 contains the Network Control Language (NCL) routines.

DNDCMP.P11 contains the Digital Data Communications Message Protocol (DDCMP) code.

DNLBLK.P11 contains line block definitions and the CHK11 interface. This file must be the next-to-last specified at assembly time.

CHK11.P11 contains the code that performs the initial hardware check of each device present on the node. This file must be the last specified at assembly time.

### 2.1.2 PDP-11 Device Modules

The following modules are device drivers; their inclusion in the assembly file list depends on each node's configuration. For more information on the hardware components, see either the PDP-11 Peripherals Handbook (1976) or the Terminals and Communications Handbook (1979). In the list below, entries in brackets [ ] indicate the unit as described in the Large Systems Product Summary (1980).

DNDL10.P11 contains driver code for the DL10 interface [DN87].

DNDTE.P11 contains driver code for the DTE20 interface [DN87S,DN20].

DNCDDQ.P11 contains synchronous line driver code for the DQ11 communications interface on the DN80-series [DN8x-H or -J].

DNCDMC.P11 contains synchronous line driver code for the DMC11 communications interface on the DN20 and DN200 [DN21-BA, -BB, -HA and DNSXX-AA, -AB].

DNCDDH.P11 contains line driver code for the DH11 16-line asynchronous serial line multiplexer. Use this module with point-to-point or multidrop configurations, and be sure to use DNDH11.P11 with it. Use this module when the asynchronous line uses DDCMP to communicate with RDX-type devices or other nodes [DN8x-EA to -ED; used with DM11s].

DNCDUP.P11 contains synchronous line driver code for the DUP11 communications interface on the DN20 [DN20-BA or -BB].

DNDM11.P11 contains the DM11 modem control routines [DN8x-EA to -ED; used with DH11s].

DNDH11.P11 contains the DH11 asynchronous line interface code. Use this module when communicating with TTYs [DN8x-EA to -ED]. This module can be used alone without other device drivers.

DNDZ11.P11 contains the DZ11 asynchronous line interface code, the DZ11 modem control routines, and the line driver code for the DZ11 8-line asynchronous serial line multiplexer [DN25-AA, -AB, or -BA]

DNLPT.P11 contains the code for the line printer routines; used with the LP11 interface.

DNCRD.P11 contains the code for the card reader routines; used with the CR11 interface.

DNDN11.P11 contains the code to support the DN11 automatic dialing interface device.

### 2.1.3 PDP-11 Debugging Modules

The following modules are useful when examining dumps of nodes, or when testing running nodes.

DNTRCE.P11 contains the code to support the optional tracing facility. This is a debugging tool.

DNDBG.P11 contains the debugging storage blocks.

### 2.1.4 Miscellaneous PDP-11 Modules

The following modules perform other network functions. Their inclusion in the assembly file list depends on the node configuration and any special operating environment that may be required.

DNDEV.P11 contains the NCL (Network Control Language) interface code to handle device access for line printers, card readers, and terminals.

DNTTY.P11 contains the terminal routines.

DN2741.P11 contains the BCD translation tables and code to support IBM 2741 terminals.

DNCTAB.P11 contains special-character tables for TTYs and line printers.

DNRDA.P11 contains code that supports ASCII remote data entry devices.

DNRDE.P11 contains code that supports remote data entry terminals on multidrop lines.

DNTSK.P11 contains code that allows the scheduling of special purpose user tasks.

### 2.1.5 DCP Modules

The DECnet Compatible Port (DCP; not to be confused with DECnet-10 Version 3.0) requires the following modules:

DNNSP.P11 contains code that allows communication with an RSX-11M node running DECnet Phase I.

DNNSP3.P11 contains code that allows communication with an RSX-11M node running DECnet Phase II.

DNDCP4.P11 contains code that allows communication with an RSX-11M node running DECnet Phase III.

### 2.1.6 PDP-8 (DN92) Modules

The following modules apply to the PDP-8-based DN92.

| | |
|---|---|
| DN92.CTL | contains a configuration file for a node numbered 73, with one LP05 line printer, one card reader, and 8 TTYs. |
| DN92.PAL | contains the DN92 source program. |
| DN92.DOC | contains the DN92 internals documentation. |
| DN9210.DOC | contains the changes between DN92 Versions 7 and 10. |

### 2.1.7 Supplementary Files

In addition to the source modules described above, a number of other files are distributed. These files contain useful information, examples of network control files, and executable DDTl1 files, which are described below in alphabetical order. Most of these modules, except DDTl1, contain unsupported code. For more information on DDTl1, please see the TOPS-10/TOPS-20 DDTl1 Manual.

| File | Contents |
|---|---|
| DDTl1.EXE | The executable DDTl1 file. |
| DDTl1.MAC | The source code for DDTl1. |
| DDTl1.DOC | A description of DDTl1. |
| NETTST.MEM | A description of the NETTST program (unsupported). |
| NETTST.MAC | The source code for NETTST, which is a program that performs demonstration functions. |
| NETTST.EXE | The executable NETTST code. |
| NETLIB.MAC | The source code for NETLIB (used by NETTST). |
| MACLIB.MAC | The library file used by NETTST. |
| TSTTSK.Pl1 | A diagnostic module that exercises each of the task interface calls in DNTSK.Pl1. |
| TULIP.MAC | The source code for the TULIP I/O package (used with NETTST). |
| TULIP.MEM | A description of the TULIP I/O package. |
| TULLIB.MAC | The source code for the I/O subroutine library (used with NETTST). |
| DNxxxx.CTL | The control files used to build software (DN8x-series and others) for certain systems. These files serve as examples for other network-building control files. |
| NETBLD.CTL | The file that starts the DNxxxx.CTL files. |

| File | Contents |
|---|---|
| NETLDR.CTL | A control file used to create NETLDR.EXE (the executable file) from source code, NETLDR.MAC, and the intermediate relocatable file NETLDR.REL. This creation also produces a memory map and a cross-reference (CREF) listing. |
| NETLDR.DOC | A brief description of NETLDR. |
| NETLDR.EXE | The stored executable binary file for NETLDR. |
| NETLDR.HLP | The NETLDR help file. |
| NETLDR.MAC | The NETLDR source file. |
| 702NET.DIR | A directory of files on the network saveset of the Distribution Tape. |

## 2.2 COPY PROCEDURE

The first step in the installation procedure is to copy the software from the CUSP tape to your system area. The second step is to copy the software from the Distribution Tape to your system area. To copy the tapes, use the BACKUP program. On the TOPS-10 host, the disk area that is allocated for DIGITAL-supplied software is [10,7], which can also be referred to as pseudo device DEC:.

The following BACKUP command sequence can be used to copy the tapes to DEC:. The slash (/) shown below at the beginning of each line is the BACKUP prompt character.

```
.R BACKUP               ;load the BACKUP program
/TAPE MTxnnn            ;use drive nnn on magnetic tape
                          ;controller x
/REWIND                 ;rewind tape to load point
/DENSITY 800            ;specify tape density (800 or 1600 bpi)
/FILES                  ;print each file name being copied
/SSNAME ALL             ;specify all savesets
/RESTORE DEC:=DSK:      ;restore to device DEC: ([10,7])
/^C
.
```

# CHAPTER 3

# CREATE A CONFIGURATION FILE

One of the source modules used in the assembly of the node software is the node-specific configuration file. You must create a node-specific configuration file for each node in the network that is not a host processor. Network software for a host processor is assembled during monitor installation (see the TOPS-10 Software Installation Guide).

This chapter covers the selection of configuration file entries, their allowable values and defaults, and shows several representative configuration files.

## 3.1 SELECT YOUR FILE ENTRIES

This section describes the entries you can put in your control files. Section 3.2 describes macros that are either required for specified features or that you can use to facilitate the definition of lines and terminals.

Use an editor to create a file whose name reflects the particular node you are configuring. The file-naming convention used in this manual is:

```
filename = DNttnn
```

where:

tt denotes the type of node (20 for a DN20, 00 for a DN200, 82 for a DN82, or 92 for a DN92).

nn represents the node number of the node.

The configuration file contains parameters that are used by MACDLX when you build a node. For lists of configuration files that are on the Software Distribution Tape, see Section 2.1. For sample configuration files, see Section 3.4.

For configuration file entries that have values other than 0 or 1 (OFF or ON), you can enter the number in either octal or decimal. A number followed by a decimal point is taken as decimal; without the decimal point, it is taken as octal. A number can also be specified as decimal by preceding it with the characters ^D (circumflex D).

Not all entries are appropriate for all types of nodes. Applicable entries for each type of node are shown in Table 3-1. Configuration file entries are also described in Appendix A. Examine the distributed control files to see what they contain. For example, a configuration file for a DN87S node might contain the entries below to specify 96 TTYs, 4 synchronous lines, with the width of TTY4 equal to 132 columns:

```
TTYN   = 96.
NLINES = 4
T4WID  = 132.
```

(The distributed DN92.CTL configuration file, supplied with the software, defines the remote station with one LP05 line printer, one card reader, and 8 TTYs at node number 73.)

**Table 3-1: Configuration File Entries by Node Type**

| Entries (macros) | Node Type | | | | |
|---|---|---|---|---|---|
| | DN20 | DN82 | DN87 | DN87S | DN200 |
| OURNNM | x | x | x | x | x |
| node name | x | x | x | x | x |
| PDP11 | o | o | o | o | o |
| FT.D20 | x | | | | |
| FT.D82 | | x | | | |
| FT.D87 | | | x | | |
| FT.87S | | | | x | |
| FT.200 | | | | | x |
| SCBMAX | x | x | x | x | x |
| NGHMAX | o | o | o | o | o |
| SEQNGH | x | x | x | x | x |
| NLINES | o | o | o | o | o |
| TTYN | o | o | o | o | o |
| (TDEF) | o | o | o | o | o |
| (DHCNFG) | | x | x | x | x |
| (DHUSE) | | x | x | x | x |
| TnDSL* | o | o | o | o | o |
| TnWID* | o | o | o | o | o |
| TnXS* | o | o | o | o | o |
| TnRS* | o | o | o | o | o |
| TnTAB* | o | o | o | o | o |
| FT.RNN | o | o | o | o | o |
| TnPFH | o | o | o | o | o |
| FT.PFH | o | o | o | o | o |
| TnRNN | o | o | o | o | o |
| FT2741 | o | o | o | o | o |
| DEFBCD | o | o | o | o | o |
| DMCN | o | | | | o |
| DUPN | o | | | | o |
| FT.RDA | o | o | o | o | o |
| FT.RDE | o | o | o | o | o |
| FTDN11N | o | o | o | o | o |
| FTHOST | o | o | o | o | o |

x = required entries
o = optional entries
blank = not applicable

*can be defined by TDEF macro
**one of a pair of entries: if one entry is 1, the other must be 0.

**Table 3-1: Configuration File Entries by Node Type (Cont.)**

| Entries (macros) | Node Type | | | | |
|---|---|---|---|---|---|
| | DN20 | DN82 | DN87 | DN87S | DN200 |
| FT.DCP** | o | o | o | o | |
| (NSPLST) | o | o | o | o | o |
| FT.MPT** | o | o | o | o | o |
| FT.CTY | | | o | | o |
| DGUTS | o | o | o | o | o |
| DEBUG | o | o | o | o | o |
| FT.TSK | o | o | o | o | o |
| FT2BIT | o | o | o | o | o |
| FTRACE | o | o | o | o | o |
| CTYWID | | | | | |
| CTYTAB | | | | | |
| DEFINE | | | | | |
| DL10SR | | | o | | |
| LPTN | | o | | | o |
| LA180 | | | | | |
| FTLPLC | | | | | |
| LPTWID | | o | | | o |
| CDRN | | o | | | o |
| DELROM | | | | | |
| FTOLDC | o | o | o | o | |
| NEILEN | | | | | |
| DFLXMT | | | | | |
| DFLRCV | | | | | |
| TnnRCV | | | | | |
| TnnXMT | | | | | |

x = required entries
o = optional entries
blank = not applicable

*can be defined by TDEF macro
**one of a pair of entries: if one entry is 1, the other must be 0.

If none of the optional entries shown in Table 3-2 is specified, the defaults are used. They also appear on the first page of the module DNCNFG.P11.

**Table 3-2: Option Macro Defaults**

| Node Type | PDP-11 | DL10 | DTE20 | SYNC I/O | NLINES | TTYs | CDR | LPT |
|---|---|---|---|---|---|---|---|---|
| 20 | 34 | 0 | 1 | DUP11 [1] | 4 | 0 | 0 | 0 |
| 82 | 40 | 0 | 0 | DQ11 [2] | 4 | 32 | 1 | 1 |
| 87 | 40 | 1 | 0 | DQ11 [2] | 4 | 32 | 0 | 0 |
| 87S | 40 | 0 | 1 | DQ11 [2] | 4 | 32 | 0 | 0 |
| 200 | 34 | 0 | 0 | DMC11 [3] | 1 | 32 | 1 | 1 |

[1] DN20-BA or -BB

[2] DN8x-H or J

[3] DN21-8A

### 3.1.1 Required Entries for All PDP-11 Nodes

The following entries are mandatory for all PDP-11 nodes:

OURNNM=nn — This is the node number declaration, where nn is a two-digit octal number with a range of 01-77. Each node's number must be unique in the network.

```
.MACRO  NODE  MNAME
        MNAME <nodename>
.ENDM
```

This macro sets up the symbolic name for the node. Nodename must be unique in the network and must consist of one to six uppercase alphanumeric characters. The first character must be alphabetic.

FT.typ=1 — This entry declares the type of node, where:

typ = D20 for a DN20
= D80 for a DN80
= D81 for a DN81
= D82 for a DN82
= D85 for a DN85
= 200 for a DN200
= D87 for a DN87
= 87S for a DN87S

NGHMAX=n — This entry specifies the maximum number of neighbors for each node in the network. If NGHMAX is less than the largest number of neighbors of any one node in the network, an error in communication may result. The default for n is one plus the number of DDCMP lines (NLINES) on the node being defined.

SCBMAX=mx — This entry specifies the maximum number of nodes that the network will support. The number can be greater than the actual number of nodes, to allow for expansion of the network at the cost of currently unused storage. **The value of mx must not be less than the actual number of nodes, or the system will fail erratically.** The number mx includes the node being configured as well as any TOPS-10 host processors in the network. The default for mx is twice the number of DDCMP lines (NLINES).

SEQNGH=n — This entry specifies the number of sequential nodes that are neighbors of this node. A sequential node is a boundary node that is "at the end of the line." For example, a DN92, a DC72, and a DECnet node connected through a DECnet-Compatible Port are sequential nodes. The default for n is zero.

### 3.1.2 Synchronous Line Entry

The following entry defines the synchronous lines on each node:

NLINES=#sy — This entry specifies the number of synchronous lines attached to the node. If omitted, any PDP-11 based node is built for four synchronous lines. NLINES can be 0 (for a DN20, DN87, or DN87S front end supporting only asynchronous lines) to a maximum of 10 for a DN87, or 12 for a DN20, DN200, DN85, or DN87S. This entry does not apply to the DN92 since it can have only one synchronous line. If a node has more than one type of synchronous line, line numbers are assigned in the following order:

DQ11
DMC11/DMR11
DUP11
DS11
DU11
DP11
DV11

For example, if a DN20 has 2 DMC11/DMR11s and 2 DUP11s, the DMC11s are lines 0 and 1 and the DUP11s are lines 2 and 3.

### 3.1.3 Asynchronous Line and Terminal Entries

The following entries define the characteristics of asynchronous lines on each node and of terminals attached to each line:

TTYN=#as — This entry specifies the total number of asynchronous terminal lines attached to the node. Code is generated for a maximum number of asynchronous lines (#as) numbered 0 to #as-1, excluding the CTY. The number of #as includes the CTY if you specify FT.CTY. If there is a CTY, it is on line 0. If this entry is omitted, the default number of lines, shown in Table 3-2, will be used. The allowable values for TTYN vary from 0 to 96 for a DN87, and 0 to 128 for a DN20, DN200, or DN87S.

Line numbers, used in the context of this manual, refer to local lines on the network nodes, and can be in the range 0 to 177. These local line numbers are not the same as the TTYmmm numbers that appear in the configuration messages sent by a host's front end at startup time. TTYmmm numbers are assigned dynamically by the host when the terminal connects to the host. When you give a line number, omit leading zeroes.

Each of the terminals declared in the TTYN entry has the following default characteristics:

- hard-wired (as opposed to dataset line)
- 72-column width
- autobaud detection
- no hardware tabs
- may be assigned by any network node
- is not an IBM 2741-type terminal

> NOTE
>
> To indicate that a terminal is an IBM 2741-type, give its speed as 134. baud; turn on FT2741=1 and include module DN2741.P11 (see Section 3.1.4).

To override any of the above defaults, use one or more of the entries listed below. Any of the following entries, of the form Tnxxx, can also be specified using generic terms. When the entry applies to the console terminal, use CTY.

TnDSL=1 — This entry specifies that line n is a dataset line and connects to a terminal with a modem rather than being hardwired. One such entry is required for each dataset line.

TnWID=w — This entry specifies that the terminal connected to line n has a column width of w characters rather than the default of 72. One entry is required for each terminal with a nondefault column width. Maximum column width is 255 columns.

TnXS=s
TnRS=s — This pair of entries specifies the transmit speed (XS) and receive speed (RS) for an asynchronous line. Transmit speed is the speed from the node processor to the terminal; receive speed is the speed from the terminal to the node processor. The line number, n, must be in octal. The speed of the line in baud, s, is usually entered in decimal. If both speeds are the same (as in DZ11 lines), you need only specify one speed.

Acceptable line speeds (in baud) are:

| | |
|---|---|
| 50. | 600. |
| 75. | 1200. |
| 110. | 1800. |
| 134.* | 2400. |
| 150. | 4800. |
| 200. | 9600. |
| 300. | |

* Use this line speed only for IBM 2741-type terminals.

If only the transmit speed is defined, the receive speed defaults to the same value. If the terminal is a split-speed terminal, both TnXS and TnRS must be specified for that line.

> NOTE
>
> Different transmit and receive speeds cannot be defined for terminals assigned to asynchronous DZ11 lines. DZ11 lines are usually formed in DN20 and DN200 nodes.

If the terminal is hardwired and is always set for a particular line speed, you may want to use these entries to set the line speed to the specified value when the node is reloaded.

TnTAB=1 This entry specifies that the terminal connected to line n has hardware tabs. One entry is needed for each such terminal.

### 3.1.4 Global Entries Defining Terminals

The following entries provide special restrictions or capabilities for terminals. When the entry applies to all the lines specified in the TTYN entry, use TTYxxx. Use CTYxxx for the console terminal.

FT.RNN=1 This entry specifies that code to support restricted terminal devices is to be generated. A restricted terminal device is one that can initially connect to the host specified for it. This entry generates only the support code; a TnRNN entry, specifying the appropriate control host, is required for each line so restricted.

TnPFH=nn This entry specifies the node number of the preferred host as node nn for terminal n. To use this entry, set FT.PFH and FTHOST to 1. (See FT.PFH below.) Use TTYPFH=nn for all TTYs and CTYPFH=nn for CTY. If the preferred host is available, you will always connect to it; if the preferred host is not available, you will connect to the first available host.

FT.PFH=0 This entry is set to 1 to include the preferred host code. If FT.PFH is 1, FTHOST must also be 1. When FT.PFH is set, the node will attempt to connect terminals to the preferred host. If the preferred host is not up, the node will connect to another host in the network.

TnRNN=nn This entry specifies that the terminal on line n can connect only to the host specified by nodenum(nn). (TTYRNN=nn will change all TTYs. CTYRNN=nn will change CTY.) One entry is required for each restricted terminal. If any terminal on this node is to be restricted, you must also specify an FT.RNN=1 entry. If the specified host is not available, the terminal will not connect to any other host. This does not prevent your doing a SET HOST command after connecting to a host.

FT2741=1 This entry specifies that code to support IBM 2741-type terminals is to be generated. If this switch is omitted, FT2741 assumes a value of 0.

DEFBCD=bll. This entry specifies the default type element (ball) for all the IBM 2741-type terminals on this node. Acceptable values for bll. are:

| | |
|---|---|
| 938. | BCD |
| 963. | EBCDIC |
| 987. | APL correspondence |
| 988. | APL (EBCDIC) |

Individual 2741-type terminals on this node can use elements other than the default by invoking the SET TTY ELEMENT monitor command. The default element is 988..

### 3.1.5 Global Entries that Change the Network Environment

The following entries to the configuration file are optional and are used to create special operating environments, set rules of protocol, or invoke special network features.

DL10SR=164000
This entry enables CHK11 to find a DL10 in a DN87 front end with 32K-word memory. This is the address at which DDT11 stops scanning for a DL10. Set it equal to or higher than the DL10 address.

DMCN=n This entry specifies the number of DMC11s connected to a DN20 or a DN200.

DUPN=n This entry specifies the number of DUP11s connected to a DN20 or a DN200.

FT.RDA=1 This entry specifies that code to support ASCII remote data entry devices is to be generated. If this entry is not present, the code is not generated.

FT.RDE=1 This entry specifies that code to support multidrop remote data entry pseudo devices is to be generated. If this entry is not present, the code is not generated.

FTDM11=1 This entry is set to include the modem code for DM11/DH11 lines. If there are no dial-in lines on this node, set FTDM11 to 0. You should set this to 1 when you use DH11s. The default value is 1 if DH11s are defined.

FTDN11=1 This entry specifies that code to support the DN11 automatic-dialing interface device is to be generated. If this entry is not present, the code is not generated.

FTHOST=0 This entry suppresses code needed for the SET HOST monitor command. If this entry is not present, code to support the SET HOST command is generated.

FT.MPT=1 This entry specifies that code to support multidrop (multipoint) lines is to be generated. If this entry is not present, the code will not be generated. Use this entry with RDX devices.

> NOTE
>
> To generate multidrop support, include the source modules DNCDDH.P11 and DNRDE.P11 in the MACDLX assembler input list.

FT.CTY=1 This entry specifies that code to support the use of the timesharing terminal on the DL11 asynchronous line interface as a TOPS-10 terminal is to be generated. CHK11 output generated at node startup time is always output on the DL11.

DGUTS=1 This entry specifies that error recovery code is to be generated. When this code is active, the node will attempt to recover from "soft" errors. Soft errors include lack of buffer or table space and incorrect message formats. Hard errors, such as functionally inoperative hardware, are still fatal. Note that when this recovery feature is activated, certain debugging facilities such as the ASSERT and TWIDDL macros are disabled. If this entry is not present, all errors are fatal, and no error recovery is attempted; however, full debugging facilities are available. It is recommended that this feature test switch be set off (DGUTS=0).

FT.TSK=1 This entry generates code to support the scheduling of special-purpose user tasks in the PDP-11 while the node is running. If this entry is not present, the code is not generated.

FT2BIT=1 This entry determines the minimum length of the stop bit on terminal lines operating at 300 baud or faster. The default value of 1 (ON) sets the stop bit to twice the length of a data bit. A value of 0 (OFF) sets the stop bit to the length of a data bit. At line speeds under 300 baud, the minimum length of the stop bit is always twice the length of a data bit. FT2BIT=0 may provide more throughput than FT2BIT=1 but this setting is incompatible with some terminals.

FTOLDC=1 This entry specifies that this node is to be generated with the version of DDCMP protocol used prior to Version 6.03 of TOPS-10. The default value is 0 (OFF). **If you are running Version 6.03 or later, do not include this entry.**

PDP11=nn This entry specifies the model number of the node's PDP-11 processor. For example, use PDP11=40 for a DN80-series node, or PDP11=34 for a DN20 and DN200 node (see Table 3-2).

Other entries for PDP-11-based nodes are described in Appendix A.

### 3.1.6 Entries for PDP-8-based DN92 Nodes

The following entries for a DN92 configuration file define parameters for a line printer and a card reader, and specify certain special conditions.

| Entry | Meaning |
|---|---|
| OURNNM=nn | Node number declaration; nn is a two-digit octal value 01 to 77. Each node number in the network must be unique. This entry is required. |
| DEFINE DN92ID <nodename> | Node name declaration. The nodename can be one to six uppercase alphanumeric characters. Each character must be preceded by a double quote (") and separated from others by a semicolon (;). (See Example 4 in Section 3.4.) The first character must be alphabetic. The default for this entry is DN92. |
| LPTN=0 | No line printer. |
| LPTN=1 | LP05 line printer (defaults are 132-column line, uppercase only). This is also the default with no LPTN entry. |
| LA180=1 | LA180 line printer (defaults are 132-column line, uppercase and lowercase). |
| FTLPLC=0 | Printer is uppercase only. FTLPLC=1 Printer is both uppercase and lowercase. |
| LPTWID=204 | Printer has 132-column line width (204 octal). |
| LPTWID=120 | Printer has 80-column line width (120 octal). |
| CDRN=0 | No card reader. |
| NEILEN=n | Maximum number of nodes, excluding the DN92. Cannot be greater than 64. |
| DFLXMT=m | Default transmit speed; m is octal equivalent of baud rate. |
| DFLRCV=m | Default receive speed; m is octal equivalent of baud rate (for example, 156=110 baud). |
| TnnRCV=m | Receive speed m for terminal nn. Informs host of speed; does not set speed. |
| TnnXMT=m | Transmit speed m for terminal nn. Informs host of speed; does not set speed. |

| Entry | Meaning |
|---|---|
| DELROM=1 | System should print an error message and halt if an error occurs in the DN92; if this line is not placed in the configuration file, the DN92 is rebooted from its ROM. |
| FTOLDC=1 | This entry specifies that this node is to be generated with the version of DDCMP protocol used prior to Release 6.03 of TOPS-10. The default value is 0 (OFF). **If you are running Version 6.03 or later, do not include this entry.** |

**3.1.6.1 DN92 Configuration File Defaults** - If no configuration file is specified for a DN92 remote station assembly, the software automatically assembles a remote node with node number=72; 8 TTYs; one card reader; and one 132-column, uppercase-only, LP05 line printer. The default switches are summarized below.

| Switch | Default |
|---|---|
| OURNNM= | 72 |
| DEFINE DN92ID< > | DN92 |
| NEILEN= | 16 |
| TTYN= | 10 (octal) (8. TTYs) |
| TnnWID= | 72 |
| CTYWID= | 72 |
| LPTN= | 1 |
| LA180= | 0 |
| FTLPLC= | 0 |
| LPTWID= | 204 (octal) (132. columns) |
| CDRN= | 1 |
| DFLXMT | If omitted, 110 baud. |
| DFLRCV | If omitted, default is DFLXMT. |

## 3.2 CONFIGURATION-DEFINING MACROS FOR PDP-11-BASED NODES

Place TDEF, DHCNFG, and DHUSE macros in your C.P11 module to provide special definitions for your asynchronous lines and terminals. These macros are defined in the DNCNFG.P11 module.

Use the NSPLST macro in C.P11 whenever a node supports one or more DECnet-Compatible Ports (synchronous lines). The NSPLST macro is defined in the C.P11 module.

### 3.2.1 The TDEF Macro

The TDEF macro is an alternate way to define the terminal characteristics described in Section 3.1.3. Use it only if the node includes asynchronous terminals on DH11 or DZ11 lines, and only if you are defining terminals with other than the default attributes, which are listed in Section 3.1.3. The form of the macro is:

```
TDEF index,<list>
```

where:

index — is a symbol representing the local line number n being defined. The value is in octal. Set this symbol initially equal to the first line you want to define. Each additional TDEF macro increments the index. TINDX is the recommended symbol name.

list — is a list of entries of the form xxx or <xxx,value> where:

- xxx — is the two- or three-character identifier in the Tnxxx entry.
- value — is the value to be assigned to the Tnxxx entry. If value is omitted, the entry is given the value 1.

For example,

```
TINDX=4
TDEF TINDX,<<RS,2400.>,<WID,80.>,TAB>
TDEF TINDX,<DSL,<RNN,10>>
```

defines the following entries:

```
T4RS=2400.
T4WID=80.
T4TAB=1
T5DSL=1
T5RNN=10
```

> NOTE
>
> An asynchronous terminal assigned to a DZ11 line will not function properly if the defined transmit and receive speeds differ. The DZ11 interface does not support split speeds.

### 3.2.2 The DHCNFG Macro

Place a DHCNFG macro definition in your configuration file (C.P11) to define the attributes of DH11 or DZ11 lines by using calls to the TDEF macro.

Note the following example:

```
.MACRO DHCNFG
TINDX=6                                 ;set index to 6
TDEF TINDX,<<RS,150.>,<XS,2400.>>       ;line 6
TDEF TINDX,<<RS,150.>,<XS,2400.>>       ;line 7
TDEF TINDX,<<RS,150.>,<XS,2400.>>       ;line 10 (octal)
TINDX=12                                ;set index to 12 (octal)
TDEF TINDX,<<RS,2400.>>                 ;line 12
TDEF TINDX,<<RS,2400.>>                 ;line 13
TINDX=50                                ;set index to 50
TDEF TINDX,<<RS,300.>>                  ;line 50
TDEF TINDX,<<RS,300.>>                  ;line 51
TDEF TINDX,<<RS,300.>>                  ;line 52
TDEF TINDX,<<WID,80.>,TAB,DSL>          ;line 53
DHUSE (NTT,NAL,NMPT,TRIB,RDPN,RDAN)     ;see Section 3.2.3
.ENDM
```

The above entries define the following asynchronous lines:

| Line No. | Nondefault Attributes |
|---|---|
| 6 | split baud, 150 receive/2400 transmit |
| 7 | split baud, 150 receive/2400 transmit |
| 10 | split baud, 150 receive/2400 transmit |
| 12 | 2400 baud, receive and transmit |
| 13 | 2400 baud, receive and transmit |
| 50 | 300 baud, receive and transmit |
| 51 | 300 baud, receive and transmit |
| 52 | 300 baud, receive and transmit |
| 53 | dataset line, hardware tabs, and 80-column line width |

### 3.2.3 The DHUSE Macro

Whenever you have asynchronous lines attached to a node, you must define a DHUSE macro in your DHCNFG macro. This is true even if all the lines and terminals are to assume the default characteristics. All but the first and last arguments in the list apply only to DDCMP asynchronous lines and are usually set to 0. RDAN is for RDA devices.

The format of the DHUSE macro is as follows:

DHUSE (NTT,NAL,NMPT,TRIB,RDPN,RDAN) where:

NTT — is the number of terminals. This argument can be specified as TTYN to use the defaults given in Table 3-2.

NAL — is the number of asynchronous point-to-point DDCMP lines.

NMPT — is the number of multipoint DH11 or DZ11 lines.

TRIB — is the number of multipoint tributary DH11 or DZ11 lines.

RDPN — is the number of point-to-point DDCMP remote data entry devices.

RDAN — is the number of ASCII remote data entry devices.

For example, a node with 64 (decimal) terminals, all taking default characteristics, can be defined in the configuration file with the following entries:

```
.MACRO DHCNFG
DHUSE (64.,0,0,0,0,0)     ;64. is number of terminals in decimal
.ENDM
```

## 3.3 SAVE THE CONFIGURATION FILE

When you have entered all the applicable entries to the configuration file, save it in your disk area. Chapter 4 describes how the configuration file is used in the assembly command string. If the editor is one that provides line numbering, such as SOS, save the file without the line numbers.

## 3.4 EXAMPLES OF CONFIGURATION FILES

The following examples are included to provide you with reference material as you generate your configuration files.

Example 1

Create a configuration file for a DN20 front end at node 16 that supports 4 low-speed synchronous lines (DUP11s) and 64 terminal lines (DZ11s). (The $ used to exit from SOS is the echo of the ESCape (ALTMODE) key.)

```
.SOS DN2016.P11
Input: DN2016.P11
00100   OURNNM=16                ;This is node #16
00200   .MACRO NODE MNAME
00300           MNAME   <SEN>    ;This is node name SEN
00400   .ENDM
00500   FT.D20=1                 ;This is a DN20 with a DTE20
00600   SCBMAX=25.               ;Maximum of 26 nodes in the network
00700   DUPN=4                   ;4 DUP11 synchronous lines
00800   NLINES=DUPN              ;Number of synchronous lines
00900   FTHOST=1                 ;Turn on SET HOST capability
01000   FT.CTY=0                 ;No DN20 CTY, DL11 goes to 20F CTY
01100   FT.RNN=1                 ;Turn on the Restricted Node code
01200   TTYRNN=26                ;All TTYs are restricted to node 26
01300   CTYRNN=26                ;on the initial connect, so is CTY
01400   TTYN=64.                 ;64 DZ11 TTY lines
01500   .MACRO DHCNFG
01600   TINDX=0
01700   TDEF TINDX,<<XS,9600.>,<WID,80.>>
01800   TDEF TINDX,<DSL>
01900   TINDX=40
02000   TDEF TINDX,<<XS,2400.>,<WID,80.>>
02100   TINDX=46
02200   TDEF TINDX,<DSL>
02300   TDEF TINDX,<DSL>
02400   TDEF TINDX,<DSL>
02500   DHUSE (TTYN,0,0,0,0,0)
02600   .ENDM DHCNFG
02700   $
*ES

[DSKC:DN2016.P11[30,5520]]

.
```

Example 2

Create a configuration file for a DN87S front end at node 16 to support one synchronous line and a maximum of 60 asynchronous lines for local terminals. Some of the local terminals are to have other than default characteristics.

```
.SOS DN8747.P11
Input: DN8747.P11
00100   OURNNM=47;                ;This is node number 47
00200   .MACRO  NODE    MNAME
00300           MNAME   <HANLEY> ;This is node name HANLEY
00400   .ENDM
00500   FT.87S=1                  ;This is a DN87S
00600   SCBMAX=25.                ;Maximum of 26 nodes in the network
00700   FTHOST=1                  ;Turn on SET HOST capability
00800   TTYN=64.                  ;64 TTYs on DH11s
00900   FT.PFH=1                  ;Turn on Preferred Host code
01000   TTYPFH=26                 ;All TTYs connect to node 26 if free
01100   CTYPFH=26                 ;Same for CTY
01200   FTDM11=0                  ;No "DH11/DM11" datasets
01300   DMCN=1                    ;One DMC11 synchronous line
01400   DQN=2                     ;Two DQ11 synchronous lines
01500   NLINES=DQN+DMCN           ;Number of synchronous lines
01600   FT.CTY=0                  ;DL11 is connected to the 20F CTY
01700   FT2741=0                  ;No 2741 TTYs
01800   .MACRO  DHCNFG
01900   TINDX=0                   ;Starting at DH #0, line #0
02000   .REPT 60                  ;48. lines
02100           TDEF    TINDX,<<XS,300.>,<RS,300.>>     ;all 300 baud
02200   .ENDR
02300   .REPT 4                   ;We have 4 lines
02400           TDEF    TINDX,<<XS,9600.>,<RS,9600.>>   ;of 9600 baud
02500   .ENDR
02600   .REPT   4                 ;We then have 4 lines
02700           TDEF    TINDX,<<XS,4800.>,<RS,4800.>>   ;of 4800 baud
02800   .ENDR
02900   .REPT   8.                ;And finally, we have 8 lines
03000           TDEF    TINDX,<<XS,2400.>,<RS,2400.>>   ;of 2400 baud
03100   .ENDR
03200           DHUSE   TTYN,0,0,0,0,0
03300   .ENDM   DHCNFG
03400   $
*ES

[DSKC:DN8747.P11[30,5520]]

.
```

Example 3

Create a configuration file for a DN200 remote station at node 70 to support two synchronous lines (DMC11s), 16 asynchronous lines, a line printer, and a card reader.

```
.SOS DN0070.P11
Input: DN0070.P11
00100   OURNNM=70                ;This is node number 70
00200   .MACRO NODE MNAME
00300           MNAME   <COMET>  ;This is node name COMET
00400   .ENDM
00500   FT.200=1                 ;It is a DN200
00600   SCBMAX=25.
00700   FT.CTY=1                 ;We have a physical CTY
00800   NLINES=2                 ;2 synchronous lines (DMC11s)
00900   TTYN=16.                 ;16 terminal lines (DZ11s)
01000   LP11N=1                  ;We have a lineprinter
01100   CDRN=1                   ;We also have a card reader
01200   .MACRO DHCNFG
01300   TINDX=0
01400   TDEF    TINDX,<<XS,9600.>,<RS,9600.>>   ;CTY is 9600 baud
01500   TINDX=3
01600   TDEF    TINDX,<<XS,9600.>,<RS,9600.>>   ;Line 3 is 9600 baud
01700   TDEF    TINDX,<<XS,2400.>,<RS,2400.>>   ;Line 4 is 2400 baud
01800   DHUSE TTYN,0,0,0,0,0     ;Asynchronous lines take defaults
01900   .ENDM DHCNFG
02000   $
*ES

[DSKC:DN0070.P11[30,5520]]

.
```

Example 4

Create a configuration file for a DN92 remote station at node 44 to support eight TTYs and one LP05 line printer.

```
.R SOS


FILE: DN9244.PAL
INPUT:  DN9244.PAL
00100   TTYN=10                                 ;10 (octal) terminals
00200   OURNNM=44                               ;Node number = 44
00300   DEFINE DN92ID<"N;"E;"W;"N;"O;"D>        ;Name = NEWNOD
00400   CDRN=0                                  ;No card reader
00500   LPTN=1                                  ;One line printer
00600   $
*ES

[DSKC:DN9244.PAL]

.
```

# CHAPTER 4

# ASSEMBLE THE SOFTWARE

The software for the PDP-11 or PDP-8-based node is assembled on the TOPS-10 host. Use the MACDLX assembler to generate PDP-11 code; use the PAL10 assembler to generate PDP-8 code.

## 4.1 SELECT ASSEMBLY SOURCE MODULES FOR PDP-11-BASED PROCESSORS

All your source modules should be in your job's search list.

The modules you select depend on the devices attached to the node, the type of communication lines, the protocol to be followed, and any special features needed. Table 4-1 provides a cross-reference of modules and network node types. For any particular type of node, some modules are required, some are optional, and others do not apply.

The inclusion of any module noted as being optional in Table 4-1 is dependent upon one or more feature test switches specified in Table 3-1 or on specific hardware components. The relationships are as follows:

| Module | Include if: |
|---|---|
| DNCDDH.P11 | FT.RDE not zero, FT.MPT not zero, or when using asynchronous (DH11 or DZ11) lines on DN8x-series nodes in a DDCMP point-to-point configuration. |
| DN2741.P11 | FT2741=1 |
| DNDN11.P11 | FTDN11=1 |
| DNRDE.P11 | FT.RDE=1 |
| DNTSK.P11 | FT.TSK=1 |
| DNTRCE.P11 | FTRACE=1 |
| DNDH11.P11 | TTYN not zero for a DN87 or DN87S |
| DNDZ11.P11 | FT.RDE not zero, when using DZ11 lines on a DN20 or DN200 node in a DDCMP point-to-point configuration, or for regular asynchronous lines on DN20s or DN200s. |
| DNCTAB.P11 | TTYN or LPTN not zero. |
| DNDM11.P11 | FT.DM11=1 |

**Table 4-1: Assembly Modules by Node Type**

| Module | Type of Node | | | | |
|---|---|---|---|---|---|
| | DN20 | DN82 | DN87 | DN87S | DN200 |
| C.P11 | x | x | x | x | x |
| S.P11 | x | x | x | x | x |
| MACROS.P11 | x | x | x | x | x |
| DNCNFG.P11 | x | x | x | x | x |
| DNCOMM.P11 | x | x | x | x | x |
| DNNCL.P11 | x | x | x | x | x |
| DNDEV.P11 | x | x | x | x | x |
| DNDCMP.P11 | o | x | x | x | x |
| DNDL10.P11 | | | x | | |
| DNDTE.P11 | x | | | x | |
| DNCDDQ.P11 | | x | o | o | |
| DNCDDH.P11 | | o | o | o | |
| DNCDUP.P11 | o | | | | x |
| DNDM11.P11 | | o | o | o | |
| DNDH11.P11 | | x | o | o | |
| DNDZ11.P11 | o | | | | o |
| DNTTY.P11 | x | x | o | o | o |
| DN2741.P11 | o | o | o | o | o |
| DNCTAB.P11 | o | x | o | o | o |
| DNDN11.P11 | o | o | o | o | o |
| DNRDA.P11 | o | o | o | o | o |
| DNRDE.P11 | o | o | o | o | o |
| DNLPT.P11 | | x | | | o |
| DNCRD.P11 | | x | | | o |
| DNTSK.P11 | o | o | o | o | o |
| DNTRCE.P11 | o | o | o | o | o |
| DNDBG.P11 | o | o | o | o | o |
| DNLBLK.P11 | x | x | x | x | x |
| CHK11.P11 | x | x | x | x | x |

x These modules are required for this type of node.
o These modules are optional for this type of node.
blank These modules do not apply to this type of node.

## 4.2 ASSEMBLE SOFTWARE FOR PDP-11-BASED PROCESSORS

The general MACDLX command string for assembling PDP-11 node software is:

```
*dev:binfile.ext,dev:lstfile.ext/CRF=dev:srcfile.ext,..
```

where:

dev: is the physical or logical device on which the input or output files are or will be located. If dev: is omitted, DSK: is assumed.

binfile.ext is the output binary program file. If .ext is omitted, .BIN is assumed.

lstfile.ext is the output program listing file. If .ext is omitted, .LST is assumed.

| | |
|---|---|
| /CRF | is the switch that includes cross-reference information in the listing file. |
| srcfile.ext,... | are the input source module files. The first four and last two modules shown in Table 4-1 must be in the order listed. Other modules are entered between these two groups and may be in any order. Source module files are separated by commas. If you do not include the .ext portion of srcfile.ext, MACDLX looks for .MAC and .M11 first, and then for .P11. Your configuration file (built in Chapter 3) must be the first one. |

The output binary program file is used when loading the node from the host processor.

The output program listing file with cross-reference information is used to generate a file-specific DDT11 file for on-line checkout, testing of node software, and debugging crashes.

## 4.3 CHECK PROGRAM SIZE FOR PDP-11-BASED PROCESSORS

At this point, it is advisable to ensure that the size of the assembled program does not exceed the memory size of the node into which it will be loaded. **If the program size is greater than the node size, you must reassemble the software, omitting some of the source module files or features.**

The size of the program appears as the third value in the second line of the MACDLX terminal output. For example, the MACDLX output, shown in Example 1 of Section 4.4, shows the size of the program as 000023 (octal). This program will fit in a DN20 node (memory size=32K), but will not fit in a DN87 node (memory size=20K).

## 4.4 EXAMPLES FOR PDP-11-BASED PROCESSORS

The following examples show the successful assembly of the software for three PDP-11-based nodes. When you give the MACDLX command string, do not type a carriage return (RET) until after you have entered the last source module file name (CHK11.P11).

Example 1

Assemble the software for a DN20 front end at node 16. The configuration file DN2016 is taken from Example 1 of Section 3.4.

```
.R MACDLX (RET)

*DN2016,DN2016.CRF/CRF=DN2016,S,MACROS,DNCNFG,DNCOMM,DNNCL,
 DNDEV,DNDCMP,DNDTE,DNCDUP,DNDZ11,DNTTY,DNCTAB,DNLBLK,CHK11 (RET)

18218   077132  000020          .PRINT  Q       ;size of program
in octal K words
22187   112232  000023          .PRINT CHKSIZ   ;size of program
in octal K words

ERRORS DETECTED:  0

DN2016,DN2016.CRF/CRF=DN2016,S,MACROS,DNCNFG,DNCOMM,DNNCL,DNDEV
DNDCMP,DNDTE,DNCDUP,DNDZ11,DNTTY,DNCTAB,DNLBLK,CHK11

RUN-TIME: 56 71     11 SECONDS
RUN-TIME RATIO:     1284/140=9.1
CORE USED:  42K      (83 PAGES)
```

Example 2

Assemble the software for a DN87S front end at node 47. The configuration file DN8747 is taken from Example 2 of Section 3.4.

```
.R MACDLX (RET)

*DN8747,DN8747.CRF/CRF=DN8747,S,MACROS,DNCNFG,DNCOMM,DNNCL,DNDEV,
 DNDCMP,DNDTE,DNCDDQ,DNCDMC,DNDH11,DNTTY,DNCTAB,DNLBLK,CHK11 (RET)

18443   106134  000022          .PRINT Q        ;size of program
in octal K words
22412   122040  000025          .PRINT CHKSIZ   ;size of program
in octal K words.

ERRORS DETECTED:  0

DN8747,DN8747.CRF/CRF=DN8747,S,MACROS,DNCNFG,DNCOMM,DNNCL,DNDEV,
DNDCMP,DNDTE,DNCDDQ,DNCDMC,DNDH11,DNTTY,DNCTAB,DNLBLK,CHK11

RUN-TIME: 63 78 12 SECONDS
RUN-TIME RATIO: 979/153=6.3
CORE USED:  44K  (87 PAGES)
```

Example 3

Assemble the software for a DN200 front end at node 27. The configuration file DN0070 is taken from Example 3 of Section 3.4.

```
.R MACDLX (RET)

[S:MACDLX 31(1065) + ]
*DN0070,DN0070.CRF/CRF=DN0070,S,MACROS,DNCNFG,DNCOMM,DNNCL,
 DNDCMP,DNCDMC,DNDEV,DNLPT,DNCRD,DNDZ11,DNTTY,DNRAM,DNVFU,DNCTAB,
 DNLBLK,CHK11 (RET)

18086   073010  000017          .PRINT  Q       ;size of program
in octal K words
22055   106144  000022          .PRINT CHKSIZ   ;size of program
in octal K words.

ERRORS DETECTED:  0

DN0070,DN0070.CRF/CRF=DN0070,S,MACROS,DNCNFG,DNCOMM,DNNCL,
DNDCMP,DNCDMC,DNDEV,DNLPT,DNCRD,DNDZ11,DNTTY,DNRAM,DNVFU,DNCTAB,
DNLBLK,CHK11

RUN-TIME: 40 56 7 SECONDS
RUN-TIME RATIO: 1365/104=13.1
CORE USED:  32K  (63 PAGES)
```

## 4.5 EXAMPLES OF SAVING SYMBOLS WITH DDT11

The following examples show the use of DDT11 to save symbols for ANF-10 nodes. Saving symbols with DDT11 allows you to delete the .CRF file and still be able to examine dump files symbolically. For further information on using DDT11, please refer to the TOPS-10/TOPS-20 DDT11 Manual.

Example 1

Extract symbols from the .CRF file for the DN20 node.

```
.R DDT11 (RET)

DDT11 7E(106)

Input:  DN2016.CRF/SYMBOL (RET)
[46p core]
[47p core]
[48p core]
[49p core]
[50p core]
[51p core]
[52p core]
[53p core]
[54p core]
[55p core]
[56p core]
[57p core]
[58p core]
[59p core]
[60p core]
[61p core]
[62p core]
[63p core]
[64p core]
[65p core]
[66p core]
[67p core]
[68p core]
[69p core]
[70p core]
%Loaded 3118 symbols.

Input:  ^Z

.SSAVE DN2016 (RET)
DN2016 saved
```

Example 2

Extract symbols from the .CRF file for the DN87S node.

```
.R DDT11 (RET)

DDT11 7E(106)

Input:  DN8747.CRF/SYMBOL (RET)
[46p core]
[47p core]
[48p core]
[49p core]
[50p core]
[51p core]
[52p core]
[53p core]
[54p core]
[55p core]
[56p core]
[57p core]
[58p core]
[59p core]
[60p core]
[61p core]
[62p core]
[63p core]
[64p core]
[65p core]
[66p core]
[67p core]
[68p core]
[69p core]
[70p core]
[71p core]
%Loaded 3271 symbols.

Input:  ^Z

.SSAVE DN8747 (RET)
DN8747 saved
```

Example 3

Extract symbols from the .CRF file for the DN200 node.

```
.R DDT11 (RET)

DDT11 7E(106)

Input:  DN0070.CRF/SYMBOL (RET)
[46p core]
[47p core]
[48p core]
[49p core]
[50p core]
[51p core]
[52p core]
[53p core]
[54p core]
[55p core]
[56p core]
[57p core]
[58p core]
[59p core]
[60p core]
[61p core]
[62p core]
[63p core]
[64p core]
[65p core]
%Loaded 2444 symbols.

Input:  ^Z

.SSAVE DN0070 (RET)
DN0070 saved
```

## 4.6 ASSEMBLE SOFTWARE FOR PDP-8-BASED PROCESSORS

Use the PAL10 assembler to generate the PDP-8 code. The PAL10 assembler can be found on the third saveset of the TOPS-10 Distribution Tape. It is best to run both PAL10 and CREF.

To run the assembler, you must have:

| | |
|---|---|
| DN92.PAL | DN92 source modules. |
| PAL10.EXE | The PAL10 assembler. |

You can also have:

| | |
|---|---|
| a configuration file | You can use DN92.CTL from the distribution tape if it serves your purposes; otherwise, create your own or alter DN92.CTL to reflect your installation. Without a configuration file, a default node is assembled (see Section 3.1.6.1). |
| CREF | The TOPS-10 cross-reference program should be on your system. Without the CREF program, you cannot obtain a cross-reference listing. |

To assemble the software, specify an input string to PAL10 in the form:

```
output-file.BIN,CREF-file.CRF/C=configuration-file.CTL,DN92.PAL
```

File extensions should be as indicated. You cannot run the cross-reference program unless you specify a CREF filename.

The following example illustrates a successful assembly of the DN92 software. This example illustrates use of a C.PAL configuration file, creation of a binary file called DN9273.BIN, and an output file that can be used to run CREF. It is useful to print the CREF file (as shown in the example) to have a cross-referenced listing for your records. You need this file if you wish to run DDT11. The C.PAL configuration file (part of DN92.CTL) contains the following definitions:

```
LPTN   = 1
CDRN   = 1
TTYN   = 10
OURNNM = 73
```

➡ Step 1: Run the PAL10 assembler.

```
.R PAL10 (RET)
*DN9273.BIN,DN9273.CRF/C=C.PAL,DN92.PAL
 ERRORS DETECTED:  0
 LINKS GENERATED:  629
 RUN-TIME:  21 SECONDS
 6K CORE USED
*^C
```

To exit from the PAL10 assembler, use CTRL/C.

➡ Step 2: Run the cross-reference program:

```
.R CREF (RET)

*DN9273=DN9273.CRF (RET)
[CRFXKC 22K core]
*^C
```

To exit from CREF, use CTRL/C.

When CREF is run, it creates a file called DN9273.LST which can be printed.

➡ Step 3: (optional) Print the CREF output file:

```
.PRINT DN9273.LST (RET)
[LPT:DN9273= /Seq:4562/Limit:433, 1 File]
```

Once this sequence has been completed, you are ready to down-line load the DN92.

### 4.6.1 Submitting a DN92 Control File

The DN92 distributed software contains a control file called DN92.CTL, which you can use to perform the above assembly automatically.

To use DN92.CTL, use the SUBMIT command:

```
.SUBMIT DN92.CTL (RET)
[Batch job DN92 queued, request  727, limit 0:05:00]
```

You should check your log file to be sure that everything is assembled correctly. Once the control file has executed correctly, you are ready to down-line load the DN92.

# CHAPTER 5

# LOAD THE SOFTWARE

This chapter covers the loading of network software into the following types of network nodes:

- local DL10-interfaced nodes
- local DTE20-interfaced nodes
- remote nodes

It also gives information on loading the KMC11 for the 2020.

Installation of software on hosts is covered in the TOPS-10 Software Installation Guide. The operation of and output from the hardware test programs CHK11 (for PDP-11-based nodes) and SYSCHK (for PDP-8-based remote nodes) are also described. CHK11 (part of DNx code) is invoked whenever a PDP-11-based node is loaded and started; SYSCHK (part of the DN92 code) is invoked whenever a PDP-8-based node is loaded and started. These programs perform initial hardware surveys of the node to ensure that node devices and node memory are functioning correctly.

All network nodes except those adjacent to a KS10 host can be loaded from any KL10 host in the TOPS-10 ANF-10 network. A node adjacent to a KS10 must be loaded from the KS10. To load a remote node, all intervening nodes between it and the host node must be running their respective software. Local nodes (communications front ends) are loaded by either BOOT11, if a DL10 interface is used, or DTELDR, if a DTE20 is used. Remote nodes are loaded by NETLDR. NETLDR loads assembled software through a communications front end (for example, a DN87S) down a specified synchronous line to a remote station (see Figure 5-1). This process is called down-line loading.

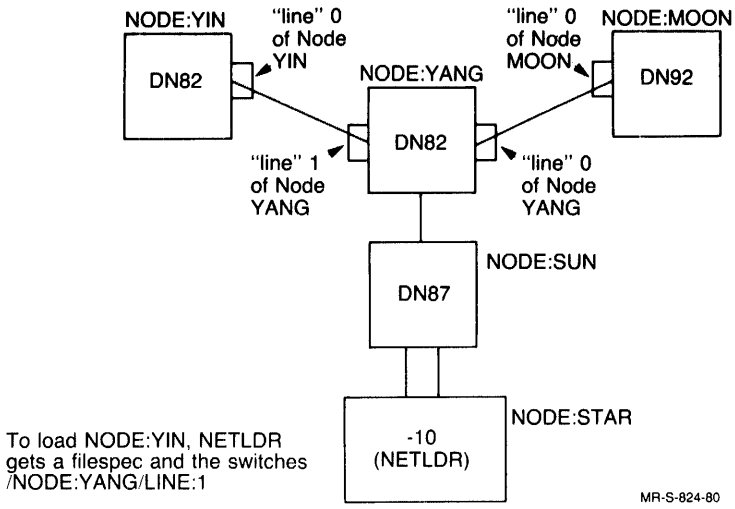


**Figure 5-1: Typical ANF-10 Network Configuration**

## 5.1 LOAD CODE FOR A LOCAL NODE

Local nodes in a TOPS-10 network are any of the communications front ends attached to the TOPS-10 hosts. The loading procedure for these nodes varies according to the node's interface to the host. DN85s and DN87s are attached through the DL10 interface. The program used to load these nodes is BOOT11. The DN20 and the DN87S are attached to the host through the DTE20 interface. The program used to load both of these front ends is DTELDR.

### 5.1.1 Load Over a DL10 (BOOT11)

To load a communications front end over the DL10, use the BOOT11 program found in your SYS: area.

Start BOOT11 by typing:

```
.R BOOT11 (RET)
```

TOPS-10 loads and starts BOOT11, which then prompts you with:

```
FILE:
```

As a response, BOOT11 expects a standard TOPS-10 file specification optionally followed by one or more of the following switches:

/CLEAR:addr — Zero PDP-11 memory from 0 to addr-1. If :addr is not present, the default is to zero all of core.

/START:addr — Load the PDP-11 from the specified TOPS-10 file and start the PDP-11 at the octal address addr. If :addr is not present, the default is the starting address of the file that was loaded. If :addr is present, it must be an even octal value.

/PORTNO:p Load the PDP-11 that is attached to port number p on the DL10. (DL10 0 has its ports numbered 0 through 3; DL10 1 has its ports numbered 4 through 7.) BOOT11 requires that the port number be specified if the total number of ports on this TOPS-10 host is greater than one.

The following response to BOOT11's prompt loads and starts the PDP-11 on port number 2, with the program located in file DSK:DN8747.BIN (the result of the assembly operation in Section 4.4):

```
FILE:SYS:DN8747/PORT:2
```

If you reply to the BOOT11 prompt with a carriage return, the following default file specification and switches are assumed:

```
FILE:DSK:PDPXIn.BIN/START/PORT:0
```

where n is the port number. The default applies when there is only one port to load with BOOT11.

BOOT11 informs you of its progress with the following messages:

```
"PDP-11 loading from file: DSK:DN8747.BIN[1,4]

"PDP-11 started   (PDP-11 now executing)
```

For additional information on BOOT11 (including progress, warning, and error messages), see the BOOT11 specification. To get out of BOOT11, type CTRL/Z.

### 5.1.2 Load Over a DTE20 (DTELDR)

To load a communications front end through the DTE20 interface, use the DTELDR program found in the SYS: disk area.

Start DTELDR by typing:

```
.R DTELDR (RET)
```

TOPS-10 loads and starts DTELDR which, in turn, prompts you with:

```
*
```

DTELDR expects you to reply with a file specification (or assume a default), an action switch, and optional modification switches. The default file specification is:

```
DSK:DTELxy.BIN
```

where:

x is the CPU number (0 or 1). If x is omitted, 0 is assumed.

y is the DTE number (0 through 3) of the PDP-11 interface to the TOPS-10 host. Note that DTE 0 is reserved for the RSX-20F front end.

The action switch to load a communications front end is:

/RELOAD:xy

where x and y have the same definitions as above. The /RELOAD switch dumps the appropriate front end and then loads it with the specified file; prints CHK11 output on the terminal; and starts the primary protocol. The front end is then enabled for communication.

The following action switches can also be used with DTELDR:

| | |
|---|---|
| /HELP | Provides information about DTELDR. |
| /IGNORE:xy:xy:...:xy | Ignores the specified front end(s) when in automatic mode. |
| /INITIALIZE:xy | Starts primary protocol on the specified front end. |
| /NOLISTEN:xy:xy:...:xy | Ignores CTY output from the front end except when reloading. |
| /TALK:xy:xy:...:xy | Talks to the CTY on the specified front end. |
| /TERMINATE:xy | Shuts down the specified front end. |

Another action switch available with DTELDR is /AUTO.

| | |
|---|---|
| /AUTO | Automatically reloads a crashed front end with its unique default file (DTEaxy.Bzz). With the /AUTO switch, the default load device is SYS: and the default dump device is XPN:; variable a is either L (load) or D (dump); x and y are as defined above; and zz is the first unused file extension in the sequence IN,00,01,...,99. The TOPS-10 monitor command SET SCHED 1000 inhibits the automatic running of DTELDR. |

To take advantage of the /AUTO feature, you must ensure that the default files contain the appropriate programs.

For example:

| | |
|---|---|
| SYS:DTEL01.BIN | must contain the program for the front end on port 1. |
| SYS:DTEL02.BIN | must contain the program for the front end on port 2. |

Modification switches that may be applicable are:

| | |
|---|---|
| /EXIT | This switch causes a return to the monitor. |
| /IMAGE | Specifies that the .BIN file is in image format; that is, the formatted binary output is unpacked (one 8-bit frame per TOPS-10 word). The default is packed format, or four 8-bit frames per TOPS-10 word. The output of MACDLX is packed format. |

/NODUMP Cancels the dump of PDP-11 memory that is automatically taken when /RELOAD loads a file. The file used for a dump is XPN:DTEDxy.Bzz, where zz is the first unused file in the sequence IN, 00, 01, ..., 99.

/NOLOG Cancels the automatic error logging that is in effect whenever you load a front end.

To load a DN87S on DTE number 2, with a program located in DSK:DN8721.BIN, and skip the automatic dump, answer the prompt with:

```
*DN8721.BIN/RELOAD:2/NODUMP
```

If the program has been located in the default location DSK:DTEL02.BIN, you need only type:

```
*/RELOAD:2/NODUMP
```

By including the following sequence in the OPR.ATO file, you can provide for both the automatic loading of a front end at TOPS-10 initialization time, as well as the subsequent automatic reloading of any front end that might crash.

```
:SLOG                          ;log in a subjob
:DEF DTE=                      ;name it DTE
DTE-R DTELDR                   ;load and start DTELDR
DTE-/INIT:1                    ;start communication with CPU0, front
                               ;end #1
DTE-/AUTO                      ;set automatic reload
```

For additional information on DTELDR, see the DTELDR Specification.

### 5.1.3 Load a KMC11 (KDPLDR)

KDPLDR works with the KMC11 synchronous line controller. **It applies only to a KS10 and loads only the controller, not a node in a network.**

The KMC11 is always loaded automatically whenever the KS10 monitor is brought up for normal timesharing.

KDPLDR can do the following:

- Start, stop, and clear a KMC11 synchronous-line controller attached to a KS10 processor with a KS10 UNIBUS adapter.
- Load and verify KMC11 microcode.
- Initialize or halt DDCMP on the DUP11 synchronous lines that the KMC11 controls.

KDPLDR has the following switches:

/AUTO The most commonly used switch. It does the following:
/MCLEAR:ALL (initializes the KMC11)
/LOAD:ALL (loads the microcode)
/VERIFY:ALL (verifies the CRAM in the KMC11)
/USTART:ALL (starts the microcode)
/START:ALL/KMC:ALL (starts DDCMP on all DUP11s)

| | |
|---|---|
| /KMC: {ALL / Ctlr} | Required with a /START and /STOP switch; Ctlr is the KMC11 controller number (always 0). |
| /LOAD: {ALL / Ctlr} | Directs KDPLDR to load the control RAMS of the given KMC11 controllers. KDPLDR loads its own version of the COMIOP/DUP microcode. |
| /MCLEAR: {ALL / Ctlr} | Clears the given KMC11 controllers. |
| /START: {ALL / line-no.} | Starts DDCMP on the specified DUP11 line(s); use with a /KMC switch that gives the KMC11 controller number. |
| /STOP: {ALL / line-no.} | Stops DDCMP on the specified DUP11 line(s); use with a /KMC switch that gives the KMC11 controller number. |
| /USTART: {ALL / Ctlr} | Starts the microcode of the specified KMC11s. |
| /VERIFY: {ALL / Ctlr} | Compares the microcode in the CRAM of the given KMC11 with the internal copy in KDPLDR. Any differences appear on the terminal where KDPLDR is run. |

Examples:

| | |
|---|---|
| /MCLEAR:ALL | Clears all KMC11s on the KS10. |
| /VERIFY:0 | Compares microcode in the CRAM of KMC11 number 0 with the internal copy in KDPLDR. Differences appear on the user's terminal. |
| /START:ALL/KMC:0 | Starts DDCMP on all DUP11 lines controlled by KMC11 number 0. |
| /STOP:1/KMC:ALL | Halts DDCMP on line 1 on the KMC11 controller. |

For more information on KDPLDR, see the KDPLDR Specification.

## 5.2 LOAD A REMOTE NODE (NETLDR)

NETLDR is the TOPS-10 utility that is used to down-line load remote nodes (DN82, DN200, and DN92). NETLDR appears on the third saveset of the Distribution Tape and should be placed on SYS:. For NETLDR to run properly, there must be a bootstrap program running in the remote node. The detailed discussion of down-line loading is divided into three sections:

- Activating the remote node
- Automatically executing NETLDR in response to a load request from the remote station
- Manually running NETLDR from the host system

### 5.2.1 Activate the Remote Node

Activating the remote node consists of powering up the unit and starting the bootstrap program that is stored in the bootstrap ROM. These are two distinct operations in a PDP-11/40 node. PDP-11/34 nodes start the ROM after powering up. In a DN92 node, the bootstrap ROM starts automatically when the power is turned on. The remote PDP-11-based node may be equipped with either a BM873 or an M9301 ROM. Consult your Field Service Representative to determine which is present on any particular node.

**5.2.1.1 Activate the Remote PDP-11 Node** - If the remote system contains an M9301 ROM, enter 173002 into the address switches, and press the LOAD ADDRESS and START switches. The bootstrap program will type a carriage return and a line feed on the terminal, and then will wait for operator input. If there is no TTY action, the system probably has a BM873 ROM. If this is the case, enter 173000 into the address switches, press LOAD ADDRESS and START, and refer to Section 5.2.2 or Section 5.2.3.

For PDP-11-based nodes which have M9301 ROMs (see Figure 5-2), operator input consists of two parts:

- M9301 switches to the ROM program
- a command string to be forwarded to NETLDR

Since the NETDLR command string can also contain switches, enter all M9301 switches first, before any part of the NETLDR input.

In the case where M9301 switches are immediately followed by NETLDR switches, the two types must be separated by some character other than a carriage return, control character, or rubout. The character used is usually a blank. A carriage return is echoed as a carriage return/line feed and then ignored. A control character or rubout deletes the current switch (but previously entered switch values remain in effect). The first carriage return after any NETLDR input terminates the console operation. The ROM then begins to send load requests to an adjacent node.

There are three M9301 switches:

/Lsyn# tells the ROM to transmit the load requests on line syn#. If either syn# or the entire switch is not present, the load requests are sent out on line 0.

/Nnn tells the ROM to designate node nn as the system (which must be a TOPS-10 host) to receive the load requests. In a multihost network, this switch will determine which host is to perform the down-line load. If the entire switch is not present, the node accepts loading from any host.

/Sser tells the ROM to override the default serial number in the load request with the number ser. If either ser or the entire switch is omitted, the serial number defaults to zero.

If you issue a carriage return in place of entering M9301 switches or a NETLDR command string, the load request will be the same as for a BM873 node. It will specify a PDP-11 node, serial number 0, and will be sent on line 0. If line 0 is inoperable, the node will not be loaded.

However, the M9301 switches give you substantial flexibility regarding alternate load paths. In Figure 5-2, if line 0 on node THREE is inoperable, you can route the load request through node FOUR by specifying the M9301 switch:

/L1

Node FOUR converts the load request to a station control message, adds its own node name, adds the number of the synchronous line leading back to the node requesting the load, and then forwards the message to the host at node SIX. Also, node SIX can load node TWO and node ONE can load node FIVE.

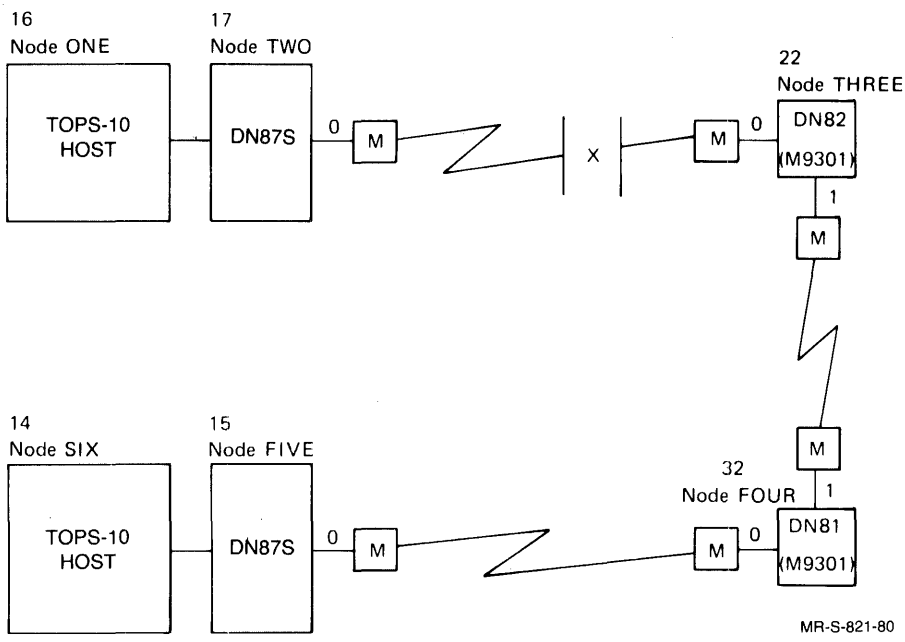


Figure 5-2: Loading an M9301-Equipped Remote Node

If the network consists of more than one host, the /N switch allows you to specify the host that is to do the loading.

If the network contains multidrop nodes, you can use the M9301 switch /S to identify the node on the synchronous line that is requesting the load.

For example, the following operator input generates a load request. This request specifies that node 10 is to load the remote node with the program DIAG1.BIN:

```
/N10 DIAG1.BIN/LOAD
```

If the NETLDR command string is included, it has the following format:

```
filespec/switch/switch...
```

where:

| | |
|---|---|
| filespec | is the dev:filename.ext[P,PN] designation of the file to be loaded into the remote node. If a filespec is specified, filename must be present. The rest of the filespec defaults to a .BIN file on your user disk area. |

The optional NETLDR switches that are applicable in bootstrap messages are as follows:

| | |
|---|---|
| /IMAGE | specifies that the file to be read is in image (unpacked) mode. This is the default mode for a PDP-8 node. |
| /LOAD | specifies that NETLDR is to load a file into the remote node but not start the program. |
| /PACKED | specifies that the file to be read is in packed mode (four 8-bit frames per TOPS-10 word). This is the default mode for a PDP-11 node. |
| /START:addr | specifies that the program that was just loaded is to be started at address addr. If the command string did not include a filespec, NETLDR starts the program that is currently in memory. This switch is not necessary if filespec is present and if the program to be loaded is to start at its default start address. |

If the remote station contains an M9312 ROM, the new M9312 ROM works like the old M9312 ROM, as outlined below:

1. Reliably boots DMC11 interfaces connected to other DMC11 interfaces (DMC can be read as DMR as well).

2. Has the following built-in error messages:

| | |
|---|---|
| ?DMC | ;An error occurred in trying to use the DMC11. It can be generated by a user who may be plugging or unplugging the EIA or V35 interface cables while the ROM is running. |
| ?NXM | ;The ROM experienced a nonexistent memory trap. This indicates a definite hardware problem, such as a misaddressed DMC11 or memory, or a broken memory. |

?CHK ;The ROM error checking code has detected a checksum error in the ROM itself. The checksum is of the sliding variety to detect ROM chips placed in wrong sockets.

The operation of the new M9312 is also the same as that of the old M9312, as follows:

1. Halt the PDP-11/34.
2. Reset the PDP-11/34.
3. Boot the PDP-11/34.
4. The console types out an underline (_) prompt. Enter the desired load file's specification and any switches that are desired. Rubout or CTRL/U will kill the current line and it will have to be typed in from the beginning.
5. Press RETURN.

The 9301WB version of the M9312 ROM has the capability of booting DMC11 interfaces (as described above), but has no error messages because the DN20 does not have a console. The old M9301WB ROM only booted from the first configured DMC11. The new 9301WB ROM boots from DMC/DMR11s 0 through 7. The old and new ROMs are operated using the steps outlined below:

1. Halt the PDP-11/34.
2. Reset the PDP-11/34.
3. Enter the number of the DMC you wish to boot in the switch register.
4. Load the address 165002 into the address register.
5. Start the PDP-11/34 with the start button.

**5.2.1.2 Activate the Remote PDP-8 Node** - If the remote node is a DN92, its ROM bootstrap program automatically begins execution when the station is turned on. As a network node that needs to be loaded, the DN92 sends a 'load request' over its synchronous line. The node that receives the request then transmits the request to the TOPS-10 host. Once the TOPS-10 host has received the request, it automatically runs NETLDR. (Unless the SCHED bit has been set to 1000. See Section 5.2.2.) NETLDR then loads the DN92 station.

You can use three switches with the DN92 ROM which are similar to those available for the M9301 ROM. These switches are:

| Switch | Meaning |
|---|---|
| /Nnn | Specifies node nn as the host to receive the load requests. |
| /Rd | Specifies that program execution is to begin at location d in remote station memory (d is an octal value of 1 to 5 digits). |
| /Sser | Specifies the serial number to be sent in the load request as ser. This is the serial number of the node to be loaded. |

As the request is sent and down-line loading occurs, the RUN and BUS lamps on the DN92 operator's console are lit and the ADDRS and DISP indicators change. If these indicators do not change after a minute or two, down-line loading has not occurred and the station will not be operable. When the "%" appears, the ROM has begun execution and is waiting for input from the operator's console. If no operator input occurs within one minute, the ROM issues the default load request. Messages that appear as successful loading occurs represent initialization and the host system prompt. For example:

```
[INITIALIZING DN92 V0.7 NODE "DN92"]
%%TTY NO XMT FLAG
     .
     .
     .
RS300 KL10 SYS#1242 14:37:36
```

Any messages about TTYs with "NO XMT FLAG" indicate that the specified device is not connected to the DN92 processor. Once the system message and period (.) prompt appear, the terminals are connected and users can log in to the TOPS-10 host.

> NOTE
>
> You can also start the DN92 by pressing the following keys on the operator's console.
>
> <INIT>
>
> 70<LXA>
>
> 0<LA>
>
> <INIT>
>
> <RUN>
>
> (See the DN92 User's Guide for more information about these keys.)

### 5.2.2 Alternate Path Loading a Front End

If you have a multiple-CPU system, your communications front ends should have special bootstrap ROMs. Only with these special ROMs can you load a front end when the CPU it is connected to is down. On a DN20, the ROM is an M9301WB; on a DN87S, it is an M9301WA. Using these ROMs, you can load your front end, not over the DTE20, but over a synchronous line, as though it were a remote station.

To do this, place an entry in NETLDR.INI for each front end of your multiple-CPU system. In the entry, specify the front end nodes using the NODE and LINE switches. With the NODE switch, give the name of the adjacent node that is to load your front end; with the LINE switch, give the number of the line on the adjacent node that is connected to line 0 on your front end. For example, given the configuration in Figure 5-3, with KL#1 down, you can load node 3 (the communications front end of KL#1) from the adjacent node (12) attached to line 0 of node 3.

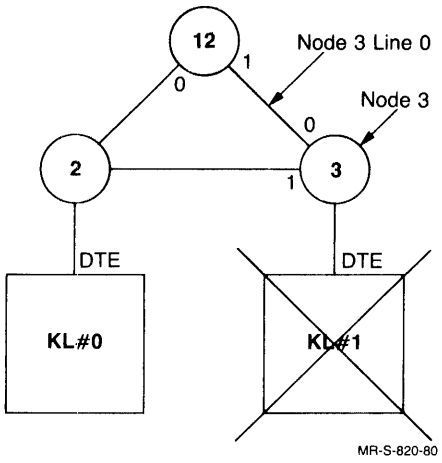


Figure 5-3: Dual-Processor Nodes

An entry in NETLDR.INI (on KL#0) to load node 3 from node 12 over line 1 of node 12 will do the job. The entry in NETLDR.INI is in the following format:

filespec/CPUtype/NODE:nodeid/LINE:syn#

For example:

PDP-11 is CPUtype /11 and PDP-8 is CPUtype /8.

For example:

DN2003/11/NODE:12/LINE:1

Once you place the entry in NETLDR.INI, start the M9301WA or M9301WB ROM. To start the ROM, enter 165002 at the start address and start the front end. The ROM then sends a boot request to the host system. This boot request activates NETLDR, which then reloads and starts your front end.

Both the M9301WA and M9301WB ROMs have an alternate starting address you can use to pass boot-request arguments to the ROM. You can use this alternate starting address to start a communications front end over a line other than line 0, or to start a communications front end directly from its programmer's console (not using a terminal). The alternate starting address is 165000. When you start the front end at this address, the ROM reads register 0. Register 0 points to a 4-word argument block of the following form:

| | |
|---|---|
| .WORD SERIAL | serial number of the local node you wish to start |
| .WORD NODE | node number of the host to receive the boot request |
| .WORD LINE | line over which you send the boot request |
| .WORD 0 | either 0 or a pointer to an ASCIZ boot string (commonly 0) |

This argument block is defined in the GO.ROM routine of the DNCOMM.P11 communications module. For example, assume you are standing at node 3 in Figure 5-3. Node 3 has serial number 70, and you wish to start it from KL#0 (node number 20). Node 3 is connected over line 1 to an adjacent node attached to KL#0.

Enter 70 followed by 20 followed by 1 followed by 0 and start your DN20. Your node will be loaded from KL#0.

### 5.2.3 Invoke NETLDR Automatically

When a remote node is activated, the bootstrap program in the ROM sends load requests over one of its synchronous lines to an adjacent node (the adjacent node must be running). A load request contains the type and serial number of the node to be loaded and, optionally, the number of the host node to do the loading as well as a NETLDR command string.

The adjacent node converts the load request into an NCL station control message and then adds its node name and the number of the line that leads back to the node to be loaded. The station control message is then sent to the host processor. If SCHEDule bit 1000 is not set, the TOPS-10 host then automatically loads and runs the NETLDR program. The TOPS-10 operating system command, SET SCHED 1000, inhibits the automatic running of NETLDR.

If the original load request included a file specification, NETLDR loads that file into the specified node directly.

If NETLDR decides that the request does not contain sufficient information, it searches the SYS:NETLDR.INI file for an entry matching the load request information supplied.

Each entry in the NETLDR.INI file consists of two parts separated by an equal sign. The left half of each entry is of the form:

```
/NODE:nodeid/LINE:syn#/TYPE:type/SERIAL:ser
```

where:

| | |
|---|---|
| /NODE:nodeid | specifies the node name or node number of the node adjacent to the node to be loaded. |
| /LINE:syn# | specifies the number of the synchronous line on the adjacent node that leads to the node to be loaded. |
| /TYPE:type | specifies the type and standard memory size of the node to be loaded. Use DN82 for DN80-series nodes. Use DN92 for DN92 stations. |
| /SERIAL:ser | specifies the serial number of the node to be loaded. This switch can be used if the remote node has an M9301 ROM or is a DN92 remote station (see Section 5.2.1.1 for an M9301 ROM, or Section 5.2.1.2 for a DN92). You can also use this switch with any value of ser when dumping PDP-11 nodes. |

If the contents of the left half contain less information than the load request, the system makes a match according to the information provided.

The right half of each entry is of the form:

```
=command1,command2...
```

where:

command — is a file specification and/or optional switches as described above.

Assume, for example, that the NETLDR.INI file at the host node SIX in Figure 5-2 contains the following entry:

```
/NODE:FOUR/LINE:1/TYPE:DN82=DN8222
```

A load request from node THREE, routed through node FOUR, would match on the above entry. NETLDR would then load the node connected to line 1 of node FOUR with the file SYS:DN8222.BIN. (For a load file, SYS: is the default device and .BIN is the default file extension.)

If there is no match in NETLDR.INI, the following message appears:

```
?NETNMI - Cannot find match in SYS:NETLDR.INI
```

In this case, correct the NETLDR.INI file.

In a multihost network, any host may receive the load request sent by a BM873 ROM (or an M9301 ROM started at 173000). If the host receiving the request does not have access to the appropriate load file, the node will not be loaded. You can provide for host discrimination with specific switches to NETLDR.

For the automatic down-line loading of any DN92 node, SYS:NETLDR.INI must contain a line of the following form:

```
/TYPE:DN92=DN9224.BIN
```

If the software has a name other than DN9224.BIN, that name should be used; if the software is not on SYS:, the [P,PN] of the appropriate area must be added to the file specification. For example, the following line in NETLDR.INI specifies that any DN92 node that sends a load request to NETLDR will be loaded with the file DN9224.BIN:

```
/TYPE:DN92=DN9224.BIN
```

To get an automatic up-line dump of the remote station before down-line loading, insert a /DUMP switch before the command file specification. For example, the following line in NETLDR.INI will up-line dump memory of any DN92 remote node onto the user's area on the TOPS-10 host:

```
/TYPE:DN92=/DUMP,DN9224.BIN
```

The dumped file is called nodename.LSD. You can also use DN92??.BIN (or the equivalent for other nodes) in NETLDR.INI to capture dumps for all DN92 nodes with any node number. You can get unique file names for each dump file by typing this command:

```
/DUMP:DN9224.D??
```

If a host is to disregard all load requests, the system operator can inhibit the running of NETLDR by typing the operator-privileged command:

```
.SET SCHED 1000
```

If, however, only requests from certain nodes are to be disregarded, use the NETLDR switch, /IGNORE, in NETLDR.INI file entries. For example:

```
/NODE:NODED/LINE:1=/IGNORE
```

will disregard all requests to load a node connected to line 1 of node NODED.

```
/NODE:NODED=/IGNORE
```

will disregard all requests to load any node connected to node NODED.

### 5.2.4 Operator Use of NETLDR

The NETLDR program can be run from the host node to dump, load, and execute programs in a remote node. A dump can be taken of a running node, but a node can only be loaded if it has sent an up-line load request to the host. Load the NETLDR program:

```
.R NETLDR (RET)
```

NETLDR prompts you with:

```
FILE:
```

NETLDR expects you to reply to the prompt with a command string of the form:

```
FILE:filespec/CPUtype/NODE:nodeid/LINE:syn#/switch/switch...
```

or

```
FILE:filespec/CPUtype/NODE:nodeid/SELF/switch/switch...
```

where:

| | |
|---|---|
| filespec | is of the form dev:filename.ext[P,PN]. If this is a load operation, filename is required. The other arguments default to a .BIN file on your user disk area. If this is a dump operation, the entire file specification can be omitted and will default to nodename.LSD on your disk area. |
| /CPUtype | is a required switch that specifies the type of processor at the remote node. For DN80-series nodes, enter /PDP11 or /11. For a DN92 remote station, enter /PDP8 or /8. |
| /NODE:nodeid | is a required switch used to specify the node to be operated on. The switch value, nodeid, can be either a node name or a node number. If the remote node is running, nodeid identifies the node itself. If the remote node is not running, the usual case in a load operation, nodeid identifies an adjacent node that is running. |
| /LINE:syn# | is a required switch when performing a dump or load operation on a node that is not running. This entry identifies the line that leads from the node to be operated on to an adjacent running node. |

/SELF — is a required switch (if the /LINE switch is not used) when performing a dump operation on a node that is running. This entry specifies that the dump is to be performed on the node specified by the /NODE switch.

The following NETLDR switches are optional as some apply to load operations and others to dump operations.

/DUMP — creates a full memory octal dump of the node specified by the /NODE switch.

/DUMP:c-d — creates an octal dump from address c to address d.

/DUMP:c-* — creates an octal dump from address c to the largest address of the processor.

> NOTE
>
> You can have NETLDR create successive dump files of the same name (the default is that NETLDR overwrites the old version) by including a question mark (?) as a character in the file name each time you request a dump. NETLDR replaces the question mark with 0 the first time you use this format. Each time thereafter, when you specify that file name, NETLDR creates a new dump file and replaces the question mark with an increment of 1.
>
> If you do not specify a dump file specification in the command string, the default filespec is nodename.LSD on DSK:.

/HELP — outputs the NETLDR help file.

/HPQ:n — specifies the high priority queue number. To have the running of NETLDR take precedence over other timesharing jobs, set HPQ=1 or some higher number. This switch can only be used if you have HPQ privileges and if the number of high priority queues in the system is nonzero.

/IMAGE — specifies that the file to be read is in image (unpacked) mode. This is the default mode for a PDP-8 node.

/LOAD — specifies that NETLDR is to load a file into the remote node but not start the program.

/PACKED — specifies that the file to be read is in packed mode (four 8-bit frames per TOPS-10 word). This is the default mode for a PDP-11 node.

/START:addr — specifies that the program just loaded is to be started at address addr. If the command string does not include a filespec, NETLDR starts the program that is currently in memory. This switch is not necessary if filespec is present and the program to be loaded is to start at its default start address.

### 5.2.5 Examples

To load the file NYC.BIN from DSKN: area [14,16] on the host node into the node connected to line 0 of node BOSTON and start the program at address 1000, issue the following string to NETLDR in response to the "FILE:" prompt:

```
DSKN:NYC[14,16]/NO:BOSTON/LI:0/11/START:1000
```

To dump locations 0 to 77776 of node BOSTON onto DSK: as a file named BOSTON.LSD, issue the following string to NETLDR in response to the "FILE:" prompt:

```
FILE:  /DUMP/NODE:BOSTON/SELF/PDP11
```

# CHAPTER 6

# INITIAL HARDWARE CHECK

As a node is loaded and started, a program runs to make a cursory check of the hardware. CHK11 runs in a PDP-11-based node, and SYSCHK, an integral part of DN92 code, runs in a PDP-8-based node.

## 6.1 INITIAL HARDWARE CHECK (CHK11) FOR PDP-11-BASED NODES

The last source module assembled into the node's software in Chapter 4 was CHK11.P11. CHK11 is a hardware test module for PDP-11-based nodes and is brought in and started whenever a node is loaded. CHK11 runs hardware diagnostics on the node's memory and devices to see that they are functioning correctly. The names and quantities (in octal) of each device type are printed on the system's console terminal if one is present. Representative CHK11 output for a successful load of a DN20 front end at node 65 could be:

```
Initializing DN20 V23(203) MAR 1983 - NEXT (27)

   160000 bytes of memory
     MF11-UP
     KW11-L
   1 DL11-A
   1 DTE20
   1 KMC11-DUP
   2 DUP11's
   1 KMC11-DZ
   1 DMC11
   4 DZ11's
Restarting DN20 V23(203) MAR 1983 - NEXT(27)
```

| NOTE |
| --- |
| When CHK11 runs, it displays the ASCII character set twice on all terminals set to a speed of 9600 baud and connected to the node. This appears as random data on all terminals not set to 9600 baud or connected to the node with DZ11 lines. |

If a problem is detected by CHK11, the error information is printed in the following format:

```
?  devnam#devnum(ADR=devaddr)
  ERROR AT srcaddr
    (message describing problem)
    REG/ADR=addr GD=expval BD=actval XOR=varbits
    [FATAL ERROR]
```

where:

devnam is the device name.

devnum is the device number. Device numbers are sequential, starting with 0.

devaddr is the first address of the device.

srcaddr is the address in the source listing where you should look for this error.

addr is the actual address in memory where the error occurred.

expval is the expected value in addr.

actval is the actual value in addr.

varbits are the bits resulting from the EXCLUSIVE OR operation on the expected and actual values.

The message FATAL ERROR is printed if the error is such that CHK11 is unable to continue (memory errors or KW11 clock errors). When CHK11 stops on a fatal error, a stop code is displayed in the console data lights. Table 6-1 lists the error stop codes and their meanings.

**Table 6-1: CHK11 Error Stop Codes**

| Stop Code | Meaning |
|---|---|
| 1 | Timeout or bus error (trap to address 4) |
| 2 | DL10 error |
| 5 | No console terminal |
| 6 | Memory error |
| 7 | KW11 clock error |

For additional information about CHK11 and a complete list of CHK11 error messages, refer to TOPS-10 CHK11 Reference Manual.

## 6.2 INITIAL HARDWARE CHECK (SYSCHK) FOR THE DN92

When the DN92 is down-line loaded by NETLDR, SYSCHK is executed. SYSCHK performs a check of the DN92 hardware and issues messages at the DN92 operator's console.

> NOTE
>
> SYSCHK is not executed when the DN92 is restarted manually.

SYSCHK execution takes about a minute. A SYSCHK message may be preceded by either %% or ?? (these characters have no special meaning).

All of the messages output by SYSCHK or the DN92 code are listed in Table 6-2. With the exception of the first, which is always seen when the station has been successfully loaded, all messages are in alphabetical order, ignoring the % or ? characters. 'DP8E' refers to the synchronous line interface modem controller. Generally, if these DP8E messages appear, you should contact Digital Field Service.

**Table 6-2: DN92 Software Messages**

| Message | Meaning |
|---|---|
| [INITIALIZING DN92 V#.#NODE "name"] | SYSCHK has begun. #.# contains the version number of the DN92 software; "name" is the nodename specified in the configuration file. |
| BAD MESSAGE TYPE | The DN92 received a message in an invalid format from the TOPS-10 host. |
| CARRIER BACK | The carrier from the synchronous modem has returned. |
| CARRIER LOST | The carrier from the synchronous modem has been lost. |
| DKC8 CLOCK ERROR | A fatal error. SYSCHK measured the speed of the DKC8 line frequency clock and found it either too fast or too slow. The clock may be malfunctioning or not present, or the processor speed may be incorrect. |
| DN92 CRASH PC=##### | A fatal error detected by software. The ROM is automatically restarted and issues a default load request after one minute. The DN92 operator can dump core by starting the dump program at location 201. |
| DN92 INT ERR | A hardware problem in the DN92 or an incorrect definition of the configuration at assembly time. |
| DN92 WON'T RUN ON A PDP8-I | DN92 software cannot run on a PDP8-I (SYSCHK looks for a BSW instruction). |

**Table 6-2: DN92 Software Messages (Cont.)**

| Message | Meaning |
|---|---|
| DP8E BUS ERROR | A receive or transmit bus request was not serviced within one baud; either the receive or the transmit modem clock is faulty. |
| DP8E CHAR DETECT SKIP FAILED | A hardware problem in DP8E detected by SYSCHK (SYSCHK received a sync character from the DP8E, but the special character flag did not come up). |
| DP8E FIELD SELECT FAILED | Either DP8E is not installed or there is a serious hardware failure; contact Field Service. |
| DP8E FLAG WON'T CLEAR | SYSCHK cannot clear a flag in the DP8E. |
| DP8E GT 9600 BAUD | SYSCHK found the synchronous line speed greater than 9600 baud. Throughput may be degraded with synchronous links faster than 9600 baud. |
| DP8E IS IN LOOPBACK MODE | SYSCHK sent a pattern out on the synchronous line but it came back. A modem on the line is probably in loopback mode; reset it and either reload or restart the program at PC 200. |
| DP8E MODEM NOT READY | Be sure the modem on the synchronous line is powered on and ready (SYSCHK READ STATUS 1 instruction). |
| DP8E NO CARRIER | SYSCHK attempted to initialize the DP8E but the READ STATUS 2 instruction indicates that CARRIER/AGC is not present. Verify that modems are ready and sending and that the link between the modems is working. |
| DP8E NOT CLEAR TO SEND | The modem may be off or not ready. |
| DP8E NOT RECEIVING | The synchronous line interface and modem are ready but are receiving no characters. |
| DP8E RCVD # | A non-SYNC character was received from the synchronous line. SYSCHK loops until it receives a SYNC character. |
| DP8E READ CHAR DETECTED # RIGHT 3 | A problem in DP8E hardware. (SYSCHK expects a 3, but received an unexpected character (#) after the SYNC character.) |

**Table 6-2: DN92 Software Messages (Cont.)**

| Message | Meaning |
|---|---|
| DP8E TERM NOT READY | A problem in DP8E hardware (SYSCHK set terminal ready, executed a DP8E READ STATUS 2 instruction, and found terminal ready not set). |
| DP8E WC OR CA WRONG | A problem in DP8E hardware (SYSCHK found incorrect values in data-break registers). |
| DP8E XMT NOT READY | A problem at the modem or between the modem and DP8E; the clock may not be received from the modem correctly (SYSCHK's attempt to send several SYNC's failed to complete quickly). |
| LPT FLAG WON'T CLEAR | A hardware problem in the line printer interface (SYSCHK could not clear the line printer flag). |
| LPTTRBL | The line printer has timed out and the FTTRBL flag has been set in the configuration file. |
| RESTARTING DN92 NODE"####" | This message appears on all DN92 TTYs whenever the station is restarted manually; #### indicates the node name defined in the configuration file (default = DN92). |
| RDCHK-REFEED CARD | A card-reader error in reading a card. Read the card in again. |
| TTY NOT CONNECTED | This message can appear on any TTY; it indicates that the terminal is not connected to a host. It generally indicates a problem at an intermediate node or a crash of the TOPS-10 host. This message also appears when a connection is broken. If the DN92 receives input from an unconnected TTY, it sends a connect request to the host node. |
| TTY# IS RUNNING OPEN | The interface to TTY# (# is the line number) is receiving a continuous stream of null characters. This condition is not fatal but will overload the station if the stream is running faster than 110 baud. |
| TTY# NO XMT FLAG | The indicated line # does not exist. If the line is part of the installation, contact Field Service. |
| TTY# RCV FLAG WON'T CLEAR | A fatal hardware error; contact Field Service. |
| TTY# XMT FLAG WON'T CLEAR | A hardware problem; contact Field Service. |

## APPENDIX A

## CONFIGURATION FILE SWITCHES

Entries that can be made in the configuration file are listed in Table A-1 in alphabetical order. Each entry in the configuration file must be in the form:

```
ENTRY=value
```

Spaces are allowed, except in character strings like nodename. Switches listed in this appendix that are not included in Table 3-1 are used only for debugging. Some switches apply only to certain node types. The definitive list of switches for the node for which you are assembling software can be found in the .LST listing created at assembly time. (For example, when assembling software for a DN92 remote station, all switches with their set values are listed at the end of the .LST listing.) The value given for each switch in Table A-1 is a typical value. Most values are either 0 or 1; values greater than 1 are expressed in octal.

**Table A-1: Configuration File Switches**

| Switch | Description |
|---|---|
| ABRGM=0 | If nonzero, optimizes the text storage area for DGUTS error message (default = 0). This switch applies only to PDP-11 based nodes. |
| CDRN=1 | The number of card readers at the remote station (0 or 1 only). |
| CNKFLD=1 | The memory field which contains chunks (0 for the first field, 1 for second, 2 for third). This switch only applies to DN92 nodes. |
| CNKSIZ=100 | The size of the memory chunk; the range is 4 to 200. This value must be a power of 2 (default is 100 words = 144 octal). |
| CP.SIZ=xx | Defines the buffer size (in bytes) for the POKE debugging aid. The POKE aid is disabled if this buffer is not defined. |
| CTRLN=2 | The number of controllers using DDCMP. |
| CTYWID=110 | The width of a hardcopy terminal or CTY at the remote station (default = 72 characters = 110 octal). |

**Table A-1: Configuration File Switches (Cont.)**

| Switch | Description |
|---|---|
| DEBUG=1 | Enables debugging features. (If 0, little or no consistency checking is done; if 1, some checking occurs and reloads are done on serious errors; if -1, reloads occur on all errors.) |
| DEFBCD=bll. | The default type ball for IBM 2741 terminals:<br><br>938.=BCD<br>963.=EBCDIC<br>987.=APL correspondence<br>988.=APL (EBCDIC) |
| DELROM=1 | If 1, overrides ROM restart at the remote station when the host processor crashes (default = 0). Applies only to a DN92. |
| DGUTS=0 | If 1, causes the remote station to attempt error recovery (default = 0). If defined, FT.CHK=0, FTASRT=0, and DEBUG=0; that is, these features are disabled. |
| DHCn=0 | If 1, specifies that DHll line n has modem control (default =0). |
| DHLn=0 | Specifies how DHll line n is used, as follows:<br><br>0 = (default)<br>1 = terminal line<br>2 = point-to-point DDCMP line<br>3 = multipoint master line<br>4 = multipoint tributary line<br>5 = point-to-point DDCMP Remote Data Entry line<br>6 = ASCII Remote Data Entry line<br>7 = line is not used |
| DHSn=B.s | Specifies the transmit-and-receive speed for DHll line n. For example, DHS0=B.300 where DHll line 0 has a transmit-and-receive speed of 300 baud. |
| DIDLLS=1 | If defined, generates the code to permit the setting of DHll line parameters with DDTll. Three symbols are defined:<br><br>Location — Contents<br><br>DHHLC — DHll hardware address.<br>DHHLN — Line number from DHll.<br>DHHLO — Parameter word (value for DHll line parameter register).<br><br>(When the hardware address is assigned to DHHLC, the parameters are set and DHHLC is reset to zero.) |
| DL10SR=164000 | Allows CHKll to find a DL10 on a DN87 with 32K-word memory upgrade. |

**Table A-1: Configuration File Switches (Cont.)**

| Switch | Description |
|---|---|
| DMCN=n | The number of DMC11s/DMR11s on a DN20 or DN200. |
| DMPHDR=1 | If 1, DDCMP message headers can be selectively dumped into a circular buffer. The buffer is the size of a DMPHDR header (8 times the number of DMPHDR bytes). |
| DMPHIN=value | If DMPHDR is defined, DMPHIN can specify the number of incoming DDCMP headers to record (default = DMPHDR). |
| DMPHLN=addr | If DMPHDR is defined, DMPHLN gives the address of the 'line block' for the DDCMP line. |
| DMPHOU=0 | The number of outgoing DDCMP headers to record if DMPHDR is defined (default = 0). |
| DMPMSG=0 | Enables selective dumping of data messages on a DDCMP line. (This feature requires the FT.SNK sink debugging aid to be nonzero.) |
| DN11N=0 | The number of DN11 dial-out units at a remote station. |
| DUPN=n | Gives the number of DUP11s on a DN20 or DN200. |
| DZCn=0 | If 1, specifies that DZ11 line n has modem control (default=0). |
| DZLn=0 | Specifies how DZ11 line n is used (default=0). The values of DZLn are the same as those shown for DHLn. |
| DZSn=B.s | Specifies the transmit-and-receive speed for DZ11 line n. For example, DZS0=B.2400 where DZ11 line 0 has a transmit-and-receive speed of 2400 baud. |
| FT.ANF=0 | If 1, causes the inclusion of code to support image mode in task-to-task communications (default = 1). |
| FT.BIG=1 | If 1, causes additional code to be used for improving error reporting and line control (default=1). |
| FT.CHK=0 | If 1, uses the code for consistency checking and enables TWIDDL macros; always 0 if DGUTS=1. |
| FT.CTY=1 | If 1, allows a hardcopy terminal on DL11 to be used as a TTY (default = 0 on DC75NP, DN85, DN20; default = 1 or 3 on all others). |
| FT.DDT=0 | If 1, leaves room for PDP-11 DDT. |
| FT.DTE=0 | If 1, DTE-20 support is needed (for example, for a DN20 or a DN87S). |
| FT.HLP=1 | If 1, allows debugging messages to be sent to the remote station CTY (default = 0). |

**Table A-1: Configuration File Switches (Cont.)**

| Switch | Description |
|---|---|
| FT.MPT=0 | If 1, supports multidrop lines. (In this case, DNCDDH.P11 and DNRDE.P11 must be in the program list for MACDLX and the value for FT.DCP must be 0.) |
| FT.PFH=1 | If 1, enables the preferred-host feature (default = 0). |
| FT.PFL=0 | If 1, enables the 'profiler' feature, which samples the PC register and creates an in-core histogram of program activity (default = 0). |
| FT.QSB=1 | If 1, implements QSYNC bit in DCMP protocol. This entry is required for correct DDCMP operation when using TOPS-10 Version 6.03 or later, and is now the default. |
| FT.RDA=1 | If 1, generates the code to support ASCII remote data entry devices. |
| FT.RDE=1 | If 1, supports multidrop remote data entry devices on the network. |
| FT.RNN=1 | If 1, supports restricted devices. This switch must be 1 if TnnRNN=nodenum is used. |
| FT.ROM=1 | If 1, supports certain remote stations with bootstrap ROMs. (This switch must be 1 for remote stations not attached to the host processor by a DL10 or a DTE20.) |
| FT.SLB=1 | If 0, omits all of the debugging information from the line blocks. This switch should always be 1 (default = 1). |
| FT.SNK=0 | If 1, enables SINK debugging macros (default = 0). To set this to 1, you must use DNDBG.P11. Use this only during debugging. |
| FT.SOU=0 | If 1, enables the 'message source' feature, which permits the single or periodic insertion of predefined messages into the operating code (default = 0). To set this to 1, you must use DNDBG.P11. Use this only during debugging. |
| FT.STC=1 | If 1, DDT11 can be used to do remote examines/deposits of PDP-11 memory (default = 1). |
| FT.TSK=0 | If 1, special purpose user-supplied tasks (for example, multiprocessing) can be scheduled within the node. |
| FT.typ | The value of this switch declares the node type and includes the code for that type of node at assembly time. Allowable node types are: D20, 200, D75, D80, D81, D82, D85, D87, and 87S. |

**Table A-1: Configuration File Switches (Cont.)**

| Switch | Description |
|---|---|
| FT2BIT=1 | If 1, sets the minimum length of the stop bit for certain terminal lines at 2. This switch should be 1 for terminal lines operating at 300 baud or faster. |
| FT3.02=1 | If 1, the code is compatible with DDCMP Version 3.02 and 4.0. This switch must be 0 for operation with TOPS-10 Version 6.03 or later (default is 1). |
| FT873=1 | If 1, the BM873 bootstrap ROM is present. |
| FT2741=0 | If 1, supports IBM 2741s. |
| FTAPTH=1 | Enables alternate paths for NCL logic (default is 1). |
| FTASRT=1 | If 1, enables the ASSERT macro. |
| FTBSTC=0 | If 1, includes the code for automatic bootstrapping of remote stations with ROMs. |
| FTCLEA=0 | If 1, core is cleared before a start or restart is run (always 0). |
| FTDL10=0 | If 1, DL10 interfaces can be used. This switch should be 1 for a DN75, DN85, or DN87. |
| FTDH11=0 | If 1, DH11 asynchronous line controllers can be used. This switch should be 1 if any TTYs are present on a DN87, DN87S, DN81, or DN82 (defaults to 1 when there are TTYs on a DN8x-node). |
| FTDM11=0 | If 1, DM11BB modem controllers can be used. If any TTY lines are dataset lines on a DN87, DN87S, DN81, or DN82, this switch must be 1. |
| FTDN11=0 | If 1, uses the DN11 code. |
| FTDP11=0 | If 1, uses the code for the DP11 synchronous line controller. Applies only to DC75NPs. |
| FTDMC11=0 | If 1, uses the DMC11 code. |
| FTDQ11=0 | If 1, uses the code for the DQ11 NPR synchronous line interface. This switch must be specified for a DN82, DN85, DN87, or DN87S, except asynchronous (only DN87 and DN87S). Defaults to 1 if NLINES is greater than 0 in a DN8x-node. |
| FTDZ11=0 | If 1, the code for the DZ11 asynchronous line controllers can be used. This switch should be 1 for a DN20 supporting TTYs. Defaults to 1 if TTYN is greater than zero for a DN2x-node. |
| FTDUP11=0 | If 1, uses the code for the DUP11 synchronous line interface. |

**Table A-1: Configuration File Switches (Cont.)**

| Switch | Description |
|---|---|
| FTDS11=0 | If 1, uses the code for the DS11. This switch must be 1 for a DC75NP and only applies to a DC75NP. |
| FTDU11=0 | Always 0. |
| FTEVEN=0 | If 1, messages are padded in order to end on an even word boundary. |
| FTExxx=1 | If 1, supports the specified IBM 2741 type ball. The variable xxx can be 938, 963, 987 or 988. The default is 988. |
| FTHOST=0 | If 1, supports the SET HOST command. Default is 1. |
| FTKG11=1 | If 1, does not support KG11 devices that compute CRCs and LRCs (default is 1). |
| FTLBAC=0 | If 1, a loopback test can be used (default is 0). |
| FTLPLC=0 | If 1, the LA180 line printer can output in lowercase. This switch is applicable only to DN92 nodes. |
| FTLP11=0 | If 1, supports the LP11 line printer. |
| FTLP20=0 | If 1, supports the LP20 line printer. |
| FTNSYN=nn | The number of SYN characters sent before a DDCMP control message (must be even and greater than or equal to 2; default = 8). |
| FTOLDC=0 | Must be 0 for TOPS-10 Version 6.03 and later. |
| FTQSYN=nn | The number of SYN characters sent after a message (must be even and greater than or equal to 2; default = 2). |
| FTRACE=0 | Used as a debugging tool; causes TRACE macros in the network software to be expanded. |
| FTSILO=32. | Sets the DH11 silo alarm level; must be less than 64 and a power of 2. |
| FTSLCT=0 | Determines the selection algorithm for multidrop lines; default is 0. |
| FTSTCD=1 | Must be 1 for any remote station debugging. If 1, a user with POKE privileges can deposit in remote memory with DDT11 (default is 1). |
| FTTWID=0 | If 1, expands TWDDL macros. These macros are used for debugging. |
| JIFSEC=60. | The number of jiffies in one second; always 60 for a 60 Hz PDP-11 remote station, 50 for a 50 Hz station (default is 60. (decimal)). |

**Table A-1: Configuration File Switches (Cont.)**

| Switch | Description |
|---|---|
| LA180=1 | If 1, enables LA180s. This switch is applicable only to DN92 nodes. |
| LPTFAK=0 | If 1, discards LPT output (default is 0). |
| LPTN=0 | If 1, supports a line printer at the remote station. |
| LPTWID=204 | The width of a line printer at a remote station (80 columns = 120 octal; 120=170 octal; 132=204 octal). |
| MAXOLN=244 | The maximum output message size, in bytes. |
| NCL.LG=0 | If FTRACE is nonzero, NCL trace is enabled, as follows:<br><br>If — Action<br><br>0 Default (no action).<br>1 Traces all messages routed by the node.<br>2 Traces only device-control messages destined for the node.<br>3 Equivalent to both 1 and 2.<br><br>If this switch is nonzero, symbols BUGBF and BUGQ are defined. BUGBF points to the 64-times-10-word circular buffer for the trace. BUGQ is one word containing the address of the next entry in the trace. (Each 10-word entry in the buffer contains the top three words of the stack, the message length, and the first 12 bytes of the message. Generally, the top three words of the stack indicate the origin of the message.) |
| NLINES= sy | The number of synchronous lines attached to the node. If omitted, the default = 4. For a DN20, DN87, and DN87S with only asynchrnous lines, sy must always be 0; sy can be up to 12 with a DN20, DN200, DN87, and DN87S. |
| NUMSYN=10 | The number of SYN characters at the start of message. |
| OURNNM=nn | Our node number. This can be 01 to 77 (octal) and must be unique within the network. |
| PDP11=mn | The model number of the PDP-11 processor at the node. For example, mn must be 40 for DN8x, 15 for DC75NP, 34 for DN20 and DN200. |
| PEEKDM=0 | If 1, enables a facility to monitor modem-control state transitions on a DH11/DM11 line (default = 0). |
| PRFL.L=bound | Defines a low-boundary value for the profiler histogram (default = 1000 (octal)). |

**Table A-1: Configuration File Switches (Cont.)**

| Switch | Description |
|---|---|
| PRFL.H=bound | Defines a high-boundary value for the profiler histogram (default = 60000 (octal)). |
| PRFL.S=power | Defines the resolution of the profiler histogram. Each bar has a width of 2 to the PRFL.S power (default = 8; that is, 2 to the 8th power = 256). |
| REPDWN=36 | The number of DDCMP unanswered <REP> messages that must pass before the line is considered down. |
| REPSEC=2 | The number of seconds between DDCMP <REP> transmissions on an idle line. |
| REPTIM=170 | RET timeout in jiffies (120 jiffies = 170 octal = 2 seconds). |
| RPLPOL=1 | If nonzero, enables remotely operated line printers to send messages to the CTY when they need assistance (default = 1). |
| SCBMAX=mx | The number of nodes the network can support. Must be greater than or equal to 2 since it includes the host processor (TOPS-10) and the node being installed. This defaults to NLINES*2 for 20 and 80-series nodes; to NLINES for DC75NP. |
| SDELAY=1 | A delay in seconds, defaults to 0. |
| SLSDMP=0 | If 1, enables a facility to monitor synchronous line driver activity. If used, two buffers of SLSDMP words are defined; one is for the receiver, one for the transmitter (default is 0). |
| TnDSL=1 | Specifies that line n is a dataset line (connects to a TTY using a modem). One such entry is required for each dataset line. |
| TnRNN=nn | Defines a restricted terminal. The node number (nn) specifies which node is the default host for the specified TTY. This can only be used if FT.RNN=1. |
| TnPFH=nn | Defines a preferred host. The node number (nn) specifies which host the TTY will be connected to, if the host is available. This feature can only be used if FT.PFH=1. |
| TnRS=s | The receive speed for an asynchronous line with no autobaud detect. This switch should be used in conjunction with the TnnXS switch. Receive speed is the speed from the terminal to the node processor. Terminals on DZ11 lines must have the same transmit and receive speeds. |
| TnTAB=0 | If 1, the terminal on line n has hardware tabs. |

**Table A-1: Configuration File Switches (Cont.)**

| Switch | Description |
|---|---|
| TnWID=w | Sets the width of the terminal connected to line n. Terminals specified with this switch must be attached through DH11 interfaces. The default width is 72 characters (110 octal); allowable widths are 80 (120 octal) and 132 (204 octal). Maximum width is 255. |
| Tn801=m | Sets DN11 number m for TTY n. |
| TnXS=s | The transmit speed for a line with no autobaud detect. This switch should be used in conjunction with the TnRS switch. Transmit speed is the speed from the node processor to the terminal. Terminals on DZ11 lines must have the same transmit and receive speeds. |
| TTYMIC=120 | The maximum number of characters that can be typed in an input message from a TTY. |
| TTYN=#as | The number of terminals that can be connected, excluding the CTY. For DN87 and DN87S nodes, the number of terminals can be 16 times the number of DH11 asynchronous line interfaces that are supported; for DN20 nodes, the number of terminals can be 8 times the number of DZ11 asynchronous line interfaces. |

# INDEX

TOPS–10
ANF–10 Networks Software
Installation Guide
AA–W557A–TB

## READER'S COMMENTS

NOTE: This form is for document comments only. DIGITAL will use comments submitted on this form at the company's discretion. If you require a written reply and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Did you find this manual understandable, usable, and well-organized? Please make suggestions for improvement.

Did you find errors in this manual? If so, specify the error and the page number.

Please indicate the type of reader that you most nearly represent.

- ☐ Assembly language programmer
- ☐ Higher-level language programmer
- ☐ Occasional programmer (experienced)
- ☐ User with little programming experience
- ☐ Student programmer
- ☐ Other (please specify) _____

Name _____ Date _____

Organization _____ Telephone _____

Street _____

City _____ State _____ Zip Code _____
or Country

Do Not Tear - Fold Here and Tape

digital

No Postage Necessary if Mailed in the United States

**BUSINESS REPLY MAIL**
FIRST CLASS PERMIT NO. 33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

**SOFTWARE PUBLICATIONS**
200 FOREST STREET MRO1–2/L12
MARLBOROUGH, MA 01752

Do Not Tear - Fold Here and Tape

Cut Along Dotted Line