# *Advanced User's Guide*

**OnNet®16 version 2.5 (January 1997)**

*Advanced User's Guide*

**OnNet®16 version 2.5 (January 1997)**

*Advanced User's Guide*

**OnNet®16 version 2.5 (January 1997)**

*Advanced User's Guide*

**OnNet®16 version 2.5 (January 1997)**

*Advanced User's Guide*

**OnNet®16 version 2.5 (January 1997)**

*Advanced User's Guide*

**OnNet®16 version 2.5 (January 1997)**

*Advanced User's Guide*

**OnNet®16 version 2.5 (January 1997)**

# Chapter 1
# Introduction

The *Advanced User's Guide* describes how to install, configure, and manage OnNet ®16. This product provides a TCP/IP stack (also called a "kernel"), network software services and tools, and user-level network applications.

This manual describes how to

- Install and configure OnNet16.
- Manage kernel and other network services.
- Configure Kerberos and SOCKS security.

The FTP Software® kernel is the part of OnNet16 that provides the TCP/IP protocol stack. The FTP Software kernel is a virtual device (VxD) kernel, which provides network access from Windows and from DOS sessions run from Windows. The OnNet16 kernel uses Windows memory management, so it operates with essentially unlimited memory resources.

**Note:** On computers that run the Microsoft Windows for Workgroups operating system, you can install and run OnNet16 applications over the Microsoft TCP/IP protocol stack (Wolverine); in this case, the FTP Software kernel is not installed. For more information, see Section 1.1, Supported Product Configurations.

## 1.1 Supported Product Configurations

Table 1-1 shows the OnNet16 product configuration. Note the following:

- The OnNet16 product includes the FTP Software kernel, NetBIOS, and InterDrive®, some network applications that run in DOS, and network applications that run in Windows. However, not all DOS applications are available on all distributions.
- When you install the FTP Software kernel, you receive a 16-bit version of the Windows Sockets (WinSock) DLL, WINSOCK.DLL.
- Some of the Windows applications are written to the WinSock API, while others are not. If you do not install the FTP Software kernel, you receive only the WinSock applications.

Table 1-1 OnNet16 Product Summary

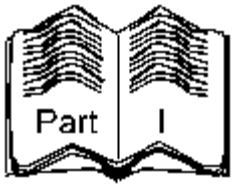
| **Platform** | **Contents** |
|---|---|
| Windows 3.*x* | The kernel, NetBIOS, and InterDrive; all Windows applications and some DOS network applications. |
| Windows for Workgroups with the Microsoft TCP/IP stack | InterDrive VxD and WinSock network applications. |

## 1.2 Structure of the Advanced User's Guide

The *Advanced User's Guide* is divided into three parts:

- Installing and Configuring OnNet16
- Managing OnNet16 Kernel and Network Services
- Using Network Security

The following illustration describes the contents of each part:



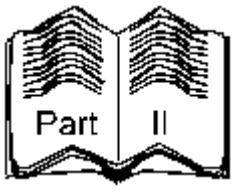

Preparing for Installation

Installing OnNet16

Configuring OnNet16

Verifying Your Network Installation

Customizing a Centralized Installation and Configuration




Introduction to VxD Kernel and Network Services

Configuring and Tuning the Kernel

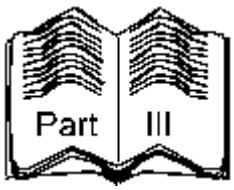
Configuring NetBIOS

Configuring Your PC Network Remotely Using DHCP

Configuring Your PC as an SNMP Agent

Monitoring Kernel Activity with IPTrace

Troubleshooting Network Connectivity




Introduction to Network Security

Configuring Kerberos security

Configuring SOCKS security

Use Part I if the OnNet16 installation card, *Installing OnNet16*, does not describe your type of installation. Use Part II if your installation includes the FTP Software kernel. Use Part III if your installation uses network security.

# Chapter 2
# Preparing for Installation

This chapter prepares you to install OnNet16 with the FTP Software TCP/IP protocol stack, also known as the kernel.

## 2.1 Overview of Installation Steps

To install OnNet16 software, perform the following steps.

Verify the hardware and software installation requirements defined in Section 2.2, Before You Begin.

Express
or
Custom

Choose an installation method from those described in Section 2.3, Choosing an Installation Method.

Run the Setup program and provide required information, as described in Section 3.1, Running the Setup Program.

Verify installation changes to system files and your connection to the network, as described in Section 3.3, Troubleshooting the Installation.


If you installed an interface that uses a modem or ISDN line, arrange for Internet service if you have not already done so. Then, use the Dialer application to establish a connection to a network.

**Related Installation Information**

If you plan to install OnNet16 to work with another network operating system (NOS), review the specific requirements in Chapter 5, Verifying Your Network Installation. If you are a network administrator who wants to customize the installation for a group or site, read Chapter 6, Customizing a Centralized Installation and Configuration.

## 2.2 Before You Begin

Before you install OnNet16, you should review the installation requirements defined in this section. You might need the help of your network administrator or your Internet service provider to obtain or verify installation information.

You can use the Installation Checklist located in the OnNet16 installation card, *Installing OnNet16*, to record required installation information.

### 2.2.1 Verifying System Prerequisites

Ensure that your system meets the hardware and software requirements.

Note that you should leave your existing FTP Software protocol stack (if any), other network operating systems, and drivers loaded so that the Setup program can detect your environment, install the appropriate files, and configure your computer correctly.

Table 2-1 shows the system requirements for OnNet16.

Table 2-1 Requirements for OnNet16

| **Component** | **Requirement** |
|---|---|
| Processor | Intel 80386 (or higher) |
| Disk Space | At least 10 MB. The actual disk space required is determined by the type of installation and the components selected for installation. When you select an installation method, the Setup program displays the disk space required for a default set of components. In addition, the Components dialog box, viewable if you select a Custom installation, displays the disk space required for the components you select. |
| Memory | Standard Mode: 1 MB<br>386 Enhanced Mode: 4 MB |
| Operating Systems | Microsoft Windows version 3.1<br>Microsoft Windows for Workgroups version3.11 |

### 2.2.2 Installing a Serial Interface

To use OnNet16 with a SLIP or a PPP connection, do one of the following:

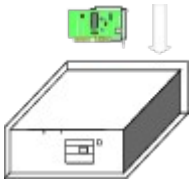
- Install a modem or an ISDN card that supports a WinISDN interface and connect the phone or ISDN line to your PC.
- Connect your PC with a serial line.
- Set up a wireless modem to use a cellular connection.

To obtain Internet service over a telephone or over an ISDN line (also provided by your telephone company), you need to arrange for such service with an Internet service provider or with your place of business.

### 2.2.3 Installing a LAN Interface

To use OnNet16 on a local area network (LAN), install a network card. You can install a related driver or let the OnNet16 Setup program do so for you. You might also need to install and verify the operation of another NOS. You can install OnNet16 on a LAN that contains TCP/IP servers, or that provides a gateway to the Internet, or both.

**Installing Your Network Card**



The network interface card provides the physical connection between your PC and the network. The type of network card that you install depends on the type of network that your system is connected to, such as Ethernet or Token Ring.

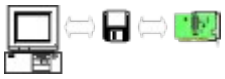
To operate in a LAN environment, you must install a network interface card on your PC before you can install OnNet16.

The Setup program uses the name and type of your network interface card to determine the appropriate driver and protocol stack to install on your system.

**To install your network interface card**

1. Refer to the manual that came with your network interface card and set any required jumpers or switches on the card.
2. Run the diagnostics software that accompanied the network card. The card must pass the manufacturer's diagnostics, if any, before you install OnNet16 software.
3. Compare the card's hardware and software settings with the memory and interrupt settings of the other devices installed in your PC. If any of these settings conflict, refer to the respective manuals for alternate settings.
4. Write down the name of your network card and other network interface card settings for use when running the Setup program.

**Verifying Network Interface Drivers**



The network interface driver is an intermediate piece of software that allows network software to communicate with network hardware (your network interface card). Typically, the manufacturer of the network card supplies a disk containing a network driver.

Drivers are usually built and distributed by the vendors who make your network interface cards. FTP Software provides some drivers for you on the OnNet16 disks or CD-ROM. The /support directory on the anonymous FTP server, ftp.ftp.com, also contains a repository of drivers.

There are several types of drivers available on the market today, each type defined or developed

by a different networking company. Table 2-2 lists the drivers that you can use with OnNet16 and the network interface type(s) that the driver supports.

Table 2-2 OnNet16 Supported Drivers and Network Interfaces

| Drivers | Network Interface |
|---|---|
| Packet Driver, defined by FTP Software in 1987 | Ethernet (802.2, 802.3)<br>Token Ring (802.5)<br>SLIP or PPP |
| NDIS (Network Driver Interface Specification), versions 2.0 and 3.0, developed by Microsoft to work with LAN Manager and now with Windows for Workgroups | Ethernet (802.2, 802.3)<br>Token Ring (802.5) |
| ODI (Open DataLink Interface) drivers, developed by Novell | Ethernet (802.2, 802.3)<br>Token Ring (802.5)<br>FDDI |
| ASI (Adapter Support Interface) drivers, defined by IBM for their 802.5 Token-Ring and LAN Support program | Token Ring (802.5) |
| Banyan VINES native driver | Ethernet (802.2, 802.3) |

These drivers are called shared drivers. OnNet16 can use the same network hardware that another NOS, like NetWare or Windows for Workgroups, is using. This lets your PC connect to more than one kind of NOS at the same time, which is useful if you need to access resources or computers on both a TCP/IP and another type of network.

When the Setup program detects a network driver, from either a previous version of OnNet16 or from other network software, the program configures your system appropriately for use with that driver and shares it with other network software.

Depending on the type of driver that you intend to use, you might need to install that driver before you run the Setup program. If you intend to use another NOS concurrently with OnNet16, install and run that NOS so that the Setup program can detect the network driver.

If you intend to use a Packet Driver, NDIS driver, or ODI driver, you can let the Setup program select the appropriate driver for you, or you can provide an updated Packet Driver disk during installation.

If you intend to use any of the following network interface drivers, install that driver before you run the Setup program:

- ASI (Adapter Support Interface), usually used with Token Ring
- A VINES driver, used with Banyan VINES


If you intend to use OnNet16 software with any other NOS, install that system *before* you install OnNet16.

When you have properly installed another NOS, the Setup program detects that system and shares the appropriate driver.

You can use OnNet16 software with the following network operating systems:

- Microsoft Windows for Workgroups
- Novell NetWare
- Microsoft LAN Manager
- Banyan VINES
- Other network operating systems that use standard drivers (such as NDIS)

Note that you cannot run two TCP/IP protocol stacks simultaneously over a single network card and driver. Both stacks require the same packets (such as IP and ARP) and the driver can supply these packets to only one stack at a time. Because of this, you cannot use OnNet16 concurrently over a single network card with another TCP/IP protocol stack, such as Microsoft TCP/IP.

For more information about installing OnNet16 with another NOS, refer to Chapter 5, Verifying Your Network Installation.

## 2.3 Choosing an Installation Method

OnNet16 includes the following installation methods:

- Express, which installs a default configuration.
- Custom, which allows you customize your configuration.
- New, which allows you to install OnNet16 without using previously configured information. (This method is included only if you are upgrading from a previous release.)

You choose the method you want during installation. For all installation methods, the Setup program

- Detects your network environment, including the operating system, any previous version of OnNet16, and drivers used with other network operating systems already installed on your system.
- Copies the files needed for basic network connectivity, including the FTP Software protocol stack (unless a Microsoft stack is already installed).
- Copies the files needed for either a default set of components (Express) or for the components you select (Custom or New).
- Updates your system files and configures the software so that you can use it alone or with other network operating systems.

# Chapter 3
# Installing OnNet16

This chapter describes how to install OnNet16 when the installation includes the FTP Software protocol stack. The chapter explains how to

- Run a Custom or New installation. (See the OnNet16 installation card, *Installing OnNet16*, for Express installation instructions.)
- Test network connectivity.
- Verify the installation.
- Reinstall individual components

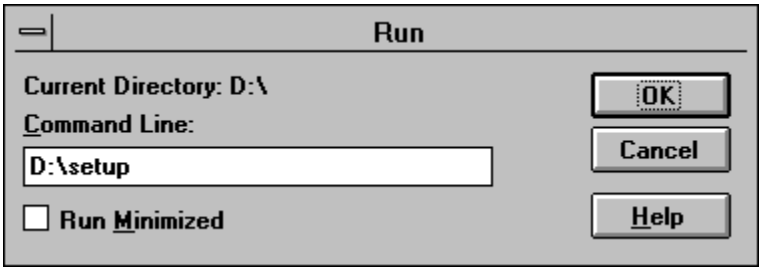
Ensure that you reviewed Chapter 2, Preparing for Installation, before running the Setup program.

## 3.1 Running the Setup Program

Run the Setup program from CD-ROM or disks.

**To start the Setup program**

1. Place the CD-ROM into the CD-ROM player or Disk 1 in the disk drive.
2. From the Program Manager File menu, choose Run; then type the drive letter, a colon, a backslash, and `setup` and then choose OK.




3. In the Welcome dialog box, choose Continue. Then read and accept the license agreement.

**To use the Setup programs**

Type information in the dialog boxes that the Setup program displays. The following buttons help you perform the installation:

- The Continue and OK buttons accept the information displayed or typed on a screen.
- The Back button displays the previous dialog box so you can review or change installation choices. You can choose Back repeatedly to move back to the beginning of the installation procedure.
- The Exit button stops the installation. If you exit from the installation, you need to run it again to install OnNet16.
- The Help button displays online Help for the current dialog box. For additional help, choose topics from the Related Topics section of the Help window.

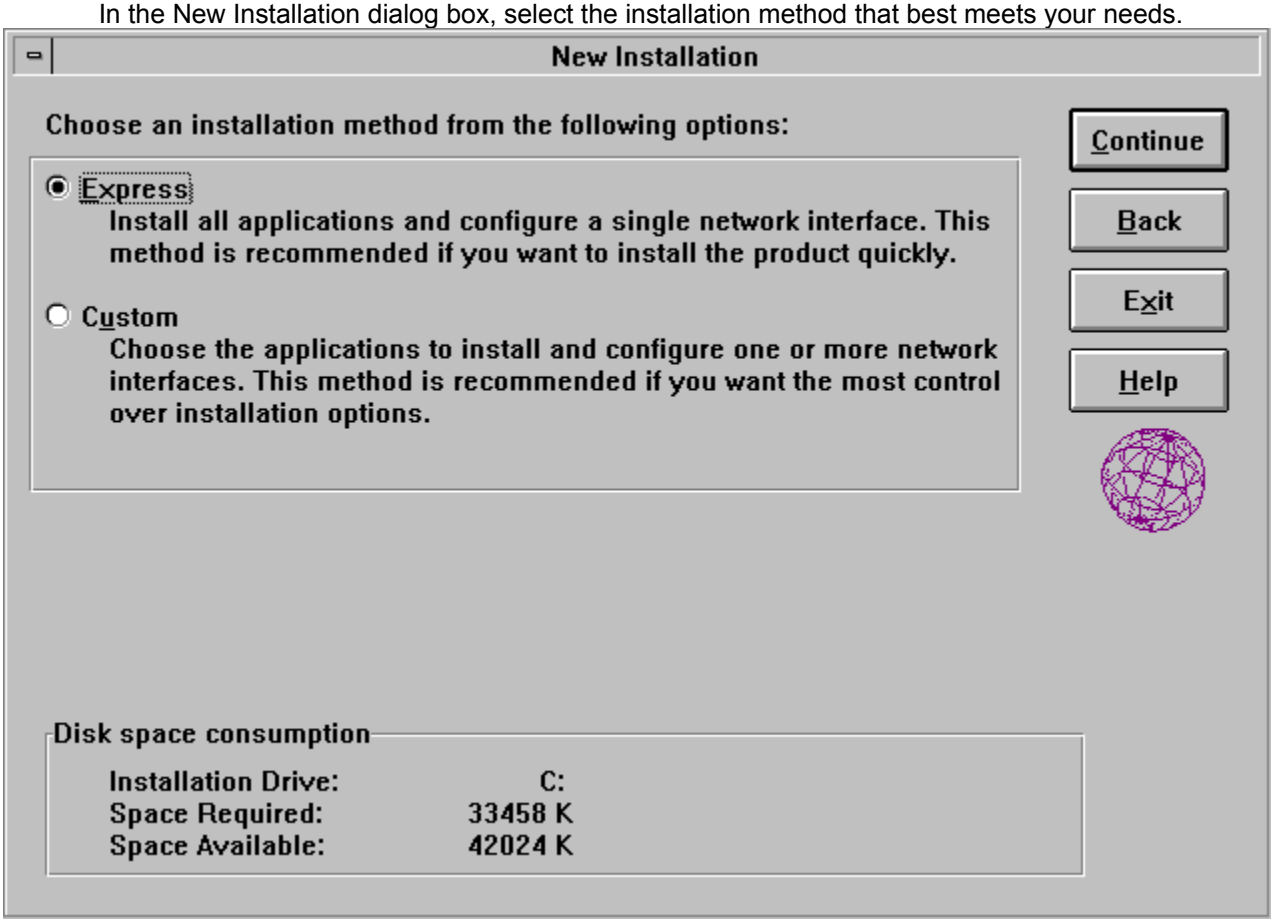
The following sections explain how to complete the installation process. You might not see every dialog box that is described because the process depends on your computer's configuration and your installation choices.

### 3.1.1 Choosing an Installation Method

Your choice of installation methods depends on whether a previous version of OnNet16 is currently installed on your computer.

**Installing on a Computer That Does Not Have OnNet or PC/TCP**

In the New Installation dialog box, select the installation method that best meets your needs.




Select Express to quickly install OnNet16. This is the fastest installation option and requires the least user input. Express quickly installs a default set of applications and configures one network interface for your computer.

Select Custom to receive more installation options. The Custom installation method detects your computer's network environment and drivers, but gives you control over various aspects of the installation. A Custom installation lets you control what is installed and lets you create multiple network interfaces; for example, one interface for a LAN and one for a Point-to-Point Protocol (PPP) or Serial Line Interface Protocol (SLIP) connection.

**Upgrading from a Previous Version**

If you are upgrading from a previous version of OnNet or PC/TCP, you can select the Express, Custom, or New installation method.

Express and Custom display your existing configuration information as you move through the installation.

Select Express to quickly install OnNet16. This is the fastest installation option and requires the least user input. Express quickly installs a default set of applications and configures one network interface for your computer. Note that, when you upgrade by selecting Express, you cannot change all of your previous configuration information.

Select Custom to receive more installation options. The Custom installation method detects your computer's network environment and drivers, but gives you control over various aspects of the installation. You can use Custom to create multiple network interfaces; for example, one interface for a LAN and one for a Point-to-Point Protocol (PPP) or Serial Line Interface Protocol (SLIP) connection. You can also use Custom on an Upgrade to choose a destination directory.

Select New to begin a new installation that does not use or display your existing configuration information, but shows Setup program default settings instead. If you select New, you receive the same installation options as with Custom.

Each option copies new files and updates (overwrites) existing files.

### 3.1.2 Duplicate MAPI.DLL

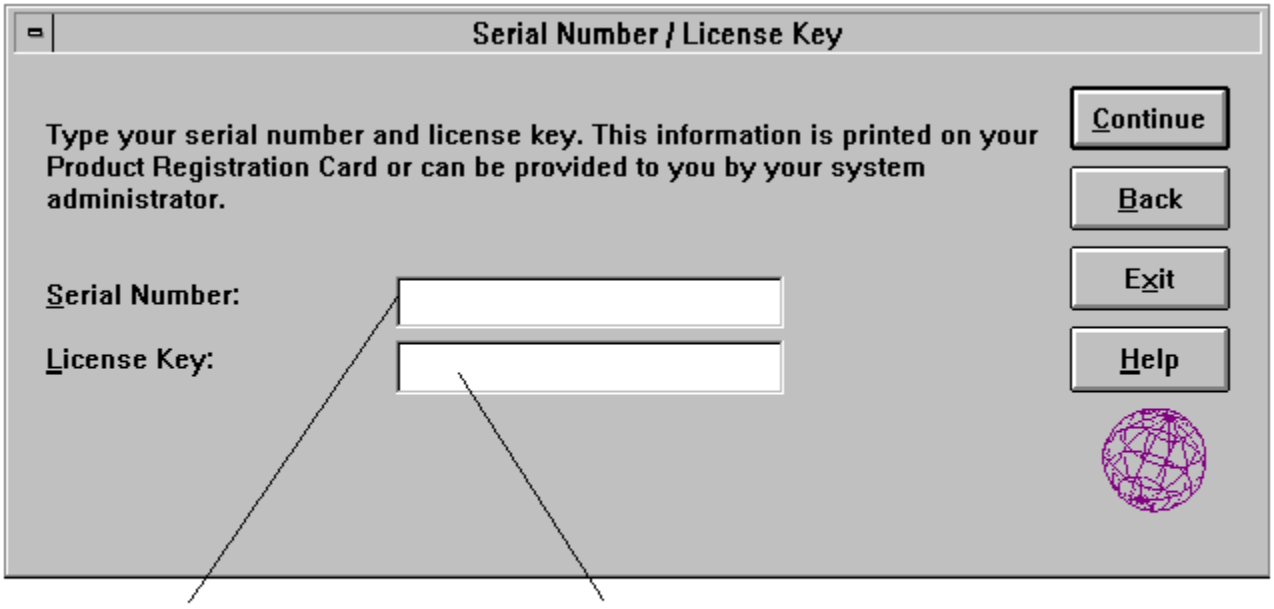
If you have a 16-bit MAPI.DLL already installed on your computer, the Setup program prompts you to choose whether or not to install Mail OnNet. If you choose to install Mail OnNet, the Setup program renames the current MAPI.DLL. Do not install Mail OnNet if you want to continue to use programs, such as Microsoft Mail, that depend on the current 16-bit MAPI.DLL.

### 3.1.3 Specifying the Serial Number and License Key

In the Serial Number/License Key dialog box, type the license key and, optionally, the serial number printed on the FTP Software License Package. These numbers uniquely identify your copy of the software.

If you did not receive, or do not type, a license key, you will be able to use OnNet16 for only 30 days. To upgrade from an evaluation copy to the full product, contact your sales representative to obtain a serial number and license key. Use the Licensing Upgrade program, in the OnNet16 2.5 program group, to enter the new number or numbers.

**Note:** You cannot use the serial number and authentication key from previous releases of OnNet16 or PC/TCP.




If you have a serial number, type it in the format *nnnn-nnnn-nnnn*. This number is not required.

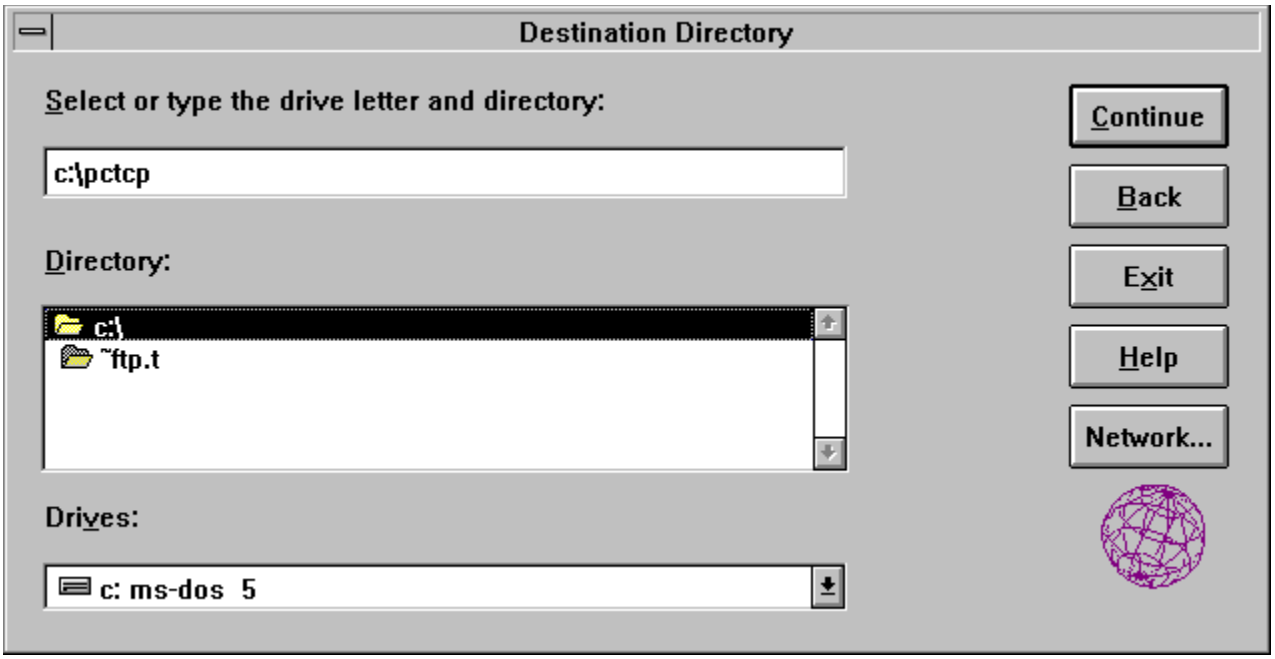
If you have a license key, type it in the format *nnnn-nnnnn-nnnn*. This number is required unless you are installing OnNet16 for a 30-day evaluation.

### 3.1.4 Specifying the Destination Directory

*For Custom installations only or a New installation when you are upgrading from a previous version of OnNet16.*

In the Destination Directory dialog box, type the full pathname for the location where you want to install the software, including any drivers and related configuration files that you add during the installation. If you want to install to a directory other than the default (C:\PCTCP), type a directory name in place of the default.

If you are upgrading and installing to the same destination directory, the Setup program updates your existing OnNet16 files. If you upgrade and install to a different directory, the Setup program does not overwrite your existing files.




**To set a destination directory**

Accept the default directory displayed in the edit box.
**-or-**
Type a directory name. If the directory does not exist, the Setup program will create it. You must use 8.3 file naming conventions.
**-or-**
Use the Drives and Directory list boxes to find and select a destination directory.

Note that the Setup program creates several subdirectories under the main directory. The \ETC directory contains files that define general system information and the SERVICES and PROTOCOL files used by the Windows Sockets DLL. The \SAMPLES directory contains files pertaining to PPP or SLIP connections (if you install a PPP or SLIP interface).

### 3.1.5 Reference Desk Options

*For installations from a network or from a CD-ROM only.*

Choose between copying the Reference Desk files to your hard drive and leaving them on the network drive or CD-ROM. (You can choose which books to install during a Custom installation, from the Components dialog box.)

| **Select** | **To** |
|---|---|
| Copy Books To My Hard Drive | Install the online documentation on your hard drive. |
| Leave Books On The CD-ROM Or Network Drive | Configure your computer to access the books from the network drive or CD-ROM. |

### 3.1.6 Choosing a TCP/IP Protocol Stack

The TCP/IP Stack Vendor dialog box appears only if your Windows for Workgroups computer has the Microsoft TCP/IP protocol stack. If your computer does not have this stack, go to Section 3.1.7, Selecting Components.

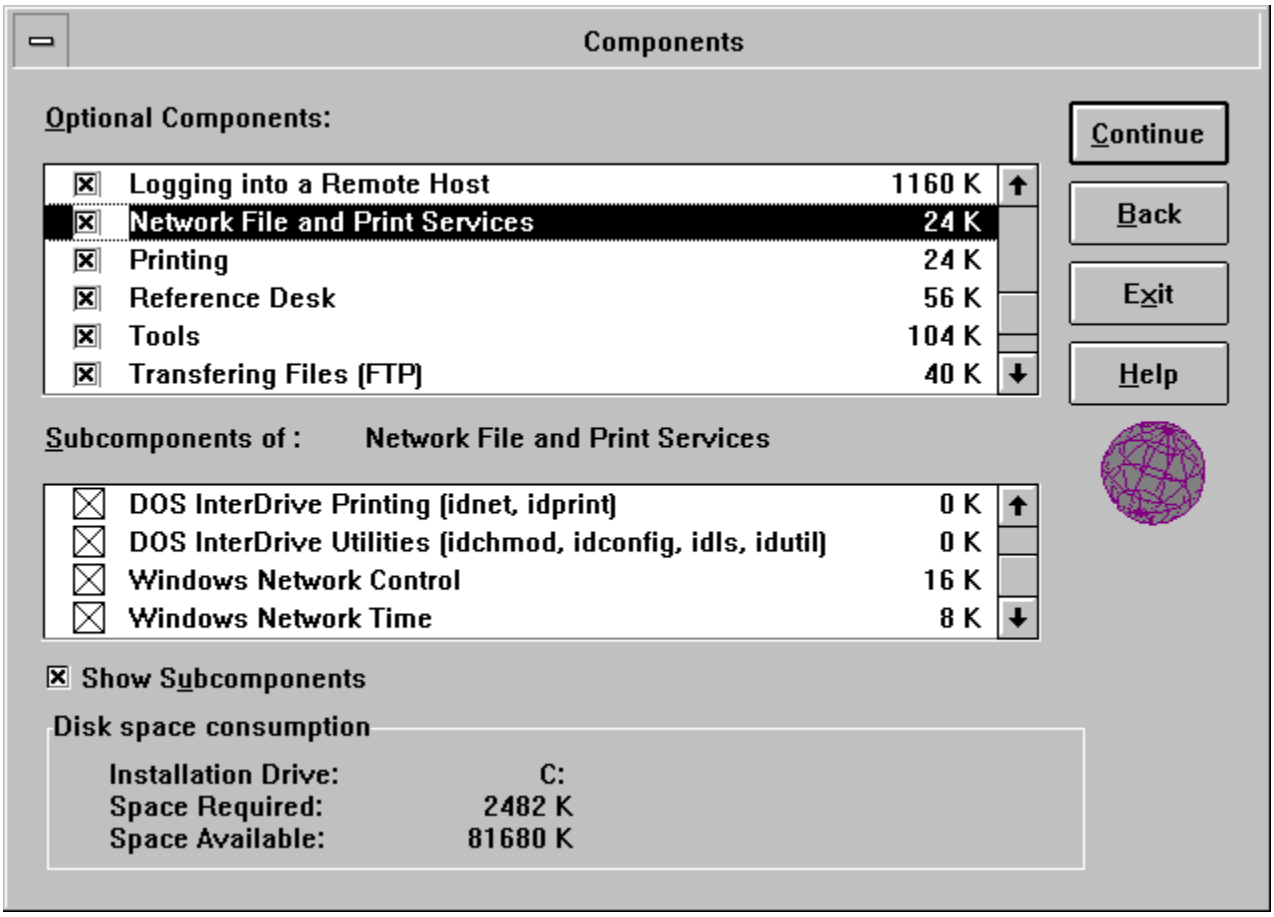
In the TCP/IP Stack Vendor dialog box, specify whether you want to continue to use the Microsoft TCP/IP protocol stack already in use on your computer, or whether you want the Setup program to remove that stack and replace it with the FTP Software stack. You cannot use both the Microsoft and the FTP Software TCP/IP stacks on your computer simultaneously.

If you keep the Microsoft TCP/IP stack, only the Windows Sockets applications are installed. The Setup program does not install the FTP Software protocol stack; and it does not install several programs that use the kernel services, such as Dialer, Ping, IPTrace, and Statistics.

This guide assumes that you select the FTP Software stack option.

### 3.1.7 Selecting Components

In the Components dialog box, select which components (major groupings of applications) and subcomponents (individual applications within a component) to install.




By default, everything but Firewall Pass-Through (SOCKS) is selected.

**To select or deselect components**

Click the check box beside a component. A check mark indicates that the component will be installed.

The Setup program installs a default set of subcomponents for each component that you select. You can, however, more precisely control what subcomponents are installed.

**To select or deselect subcomponents**

1. Choose Show Subcomponents.
2. Click the check box beside a subcomponent. A check mark indicates that the component will be installed.

If you select a component but none of its subcomponents, the disk space might, nevertheless, amount to more than 0, since some related files will be installed.

**Note:** The disk space required to install the components that you select appears at the bottom of the dialog box. This value can vary from computer to computer, due to the DOS file storage mechanism. The actual size depends upon the size of the hard disk partition: the larger the disk partition, the more disk space will be used.

The following table describes the tasks that you can perform with each component and subcomponent.

| Select this component | To install the files needed to |
|---|---|
| Archiving and Restoring Files | Back up and restore files. |
| Exchanging Mail, Messages and News | Read and send electronic mail; read bulletin boards and access news groups; “chat” with other users on the network. |
| Finding Hosts and Users on the Network | Verify that computers are on the network; obtain information about your computer, users, and remote hosts. |
| Firewall Pass-Through (SOCKS) | Use SOCKS security to access a remote host that is protected by a SOCKS firewall. |
| KEYview | View and print files independently of the application that created them. |
| Logging into a Remote Host | Log in to Telnet servers on remote hosts; execute commands on remote hosts. |
| Network File and Print Services | Use files on remote NFS hosts, manage files and directories on remote NFS hosts, and print to remote printers. |
| Printing | Send print jobs to network printers. |
| Reference Desk | View online versions of the document set. |
| Tools | Collect and display data about packets sent from or received by the TCP/IP protocol stack and gather information about system files to use when you contact Technical Support. |
| Transferring Files (FTP) | Transfer single or multiple files between your computer and file servers on the network or the Internet. |
| Windows Server Control | Set up FTP and Print servers; set up an SNMP MIB II agent. |

### 3.1.8 Selecting or Updating an Existing Driver

When the Setup program is able to detect a network driver on your computer, it displays the Found Existing Driver dialog box. In this dialog box, choose to continue using the installed driver or to install a new driver.

If you use other network operating systems concurrently with OnNet16, you should use your existing driver. The Setup program configures your computer to share that driver with any other network operating systems.

**To use the installed network driver**

In the dialog list box, select the driver you want to use, then choose Continue.

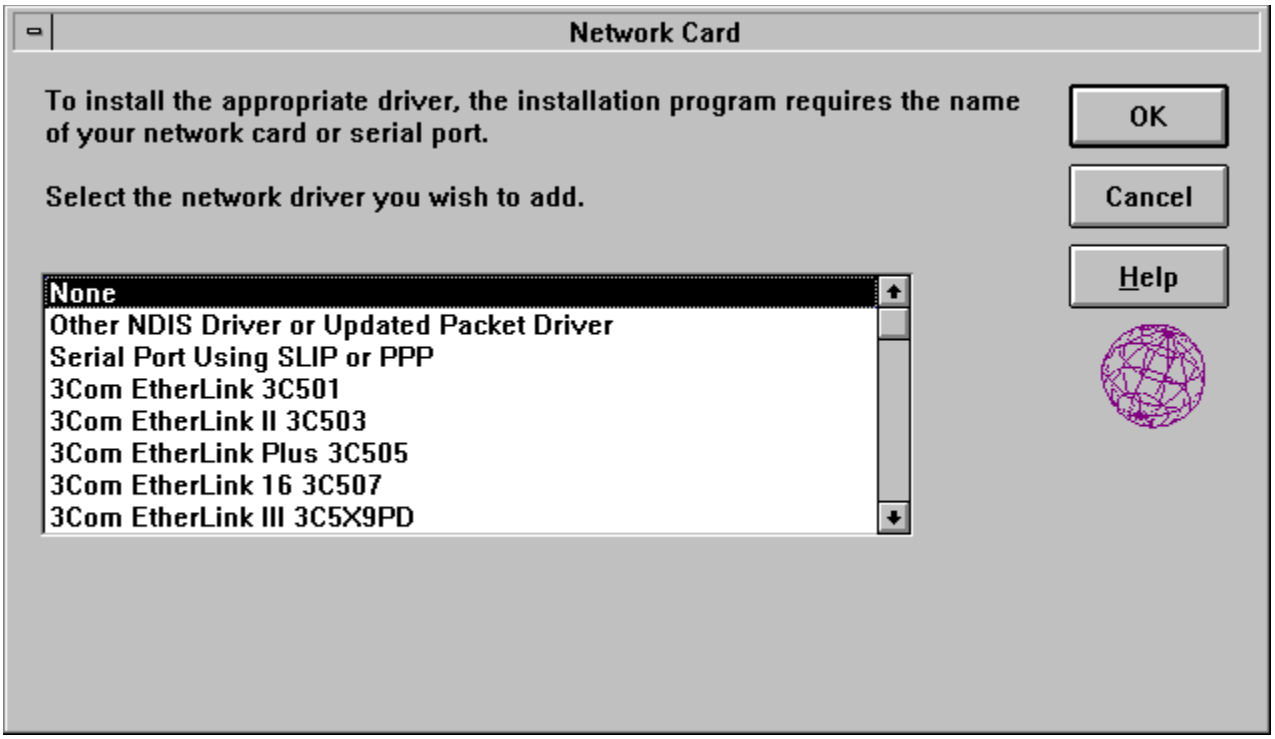
For instructions about other tasks you can perform from this dialog box, see the online Help.

### 3.1.9 Selecting a Network Card or Selecting Serial Port

If your computer has no network driver running or if you choose to install a new network driver, the Setup program displays the Network Card dialog box. Select the name of the network card you have installed in your computer, or, for any serial line, modem, or ISDN connection, select Serial Port Using SLIP or PPP.

To find the name of your network card, refer to the documentation that accompanied the card.

The Setup program installs the appropriate driver, based on the selection you make.




**To install a network card driver**

From the list box, select your card and choose OK.
**-or-**
From the list box, select Other NDIS Driver or Updated Packet Driver; when the Setup program prompts you, insert the disk containing a network card driver (supplied by the manufacturer of your network card).

If you select None; you must manually install and configure the network driver for that card after the OnNet16 installation.

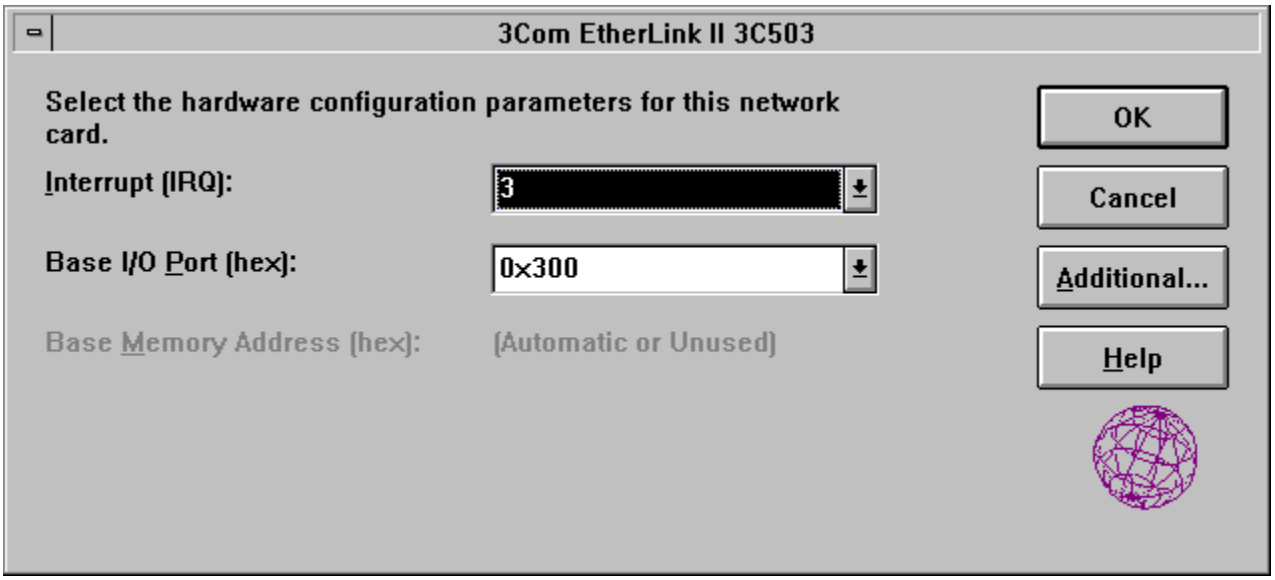
**To configure serial port connections**

In the list box, select Serial Port Using SLIP or PPP and choose OK.

### 3.1.10 Selecting Network Card Settings

*For LAN installations only*.

After you select a network card, the Setup program displays a network card settings dialog box, which is specific to your card; select or type the appropriate values to use for your network card. Refer to your network card documentation for appropriate values.




If any of the fields are grayed out on your screen, or you see the phrase `Automatic or Unused`, or both, you do not have to configure that setting.

The following table lists and describes the fields in the network card settings dialog box:

| Card Setting | Description |
|---|---|
| Interrupt (IRQ) | The interrupt vector used by your network card (an electronic signal that the card uses to communicate with your computer). For example, 3 or 5. |
| Base I/O Port (hex) | The input/output channel for your network card (the channel that your card uses to send these signals) in hexadecimal notation (for example, 0x280). |
| Base Memory Address | The base memory address for your card (the location in your computer's memory used to communicate with the card) in hexadecimal notation (for example, 0xD000). |

#### Configuring Additional Card Settings

Choose Additional to configure other network card settings (if there are any for your card). Typically, you do not need to configure additional settings, in which case the Setup program uses

the default values. If there are additional configuration settings for your network card, the resulting dialog box displays those settings and any default values.

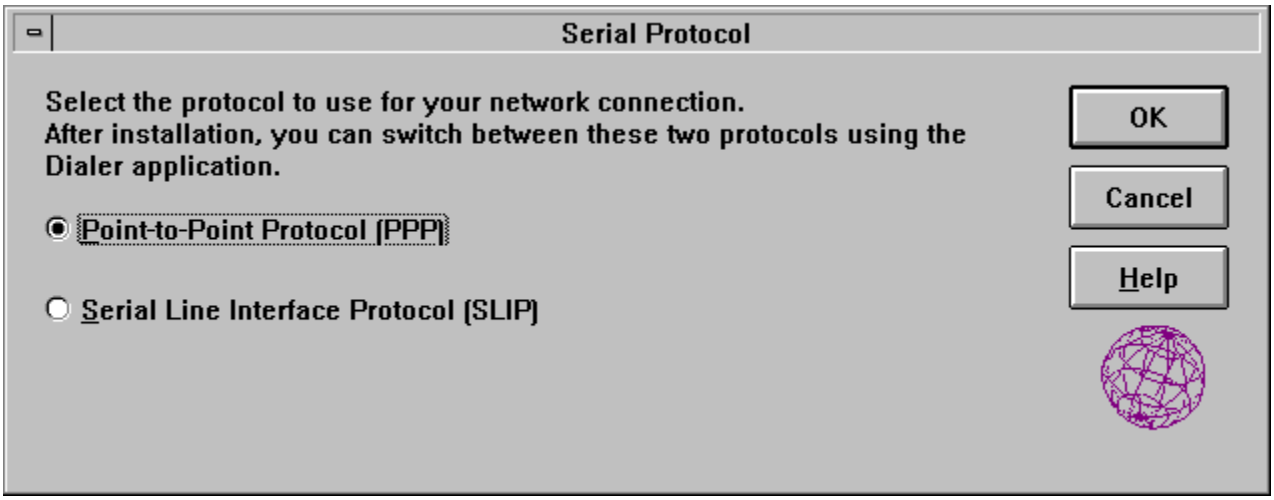
**To view or change an additional setting**

1. In the network card settings dialog box, choose Additional.
2. In the Additional Settings box, select a parameter from the Additional Settings list box.
3. Type or select the value for the parameter in the Value box.
4. To register the new value, choose Set.
   **–or–**
   To display the original value for that setting, choose Revert.

### 3.1.11 Selecting a Serial Interface Protocol

*For serial installations only*. For example, if you plan to use a modem or ISDN line.

If you selected Serial Port Using SLIP or PPP in the Network Card dialog box, the program displays the Serial Protocol dialog box. Select the protocol to use for your serial line connection: PPP (Point-to-Point Protocol) or SLIP (Serial Line Internet Protocol).




SLIP and PPP are serial line protocols that let your computer communicate with remote hosts over the Internet using a telephone line and a modem or using an ISDN interface and line. Your Internet service provider (or the network administrator at your workplace) usually specifies which serial protocol to use.

**Note:** When you communicate with remote hosts over a serial line, both hosts must use the same serial line protocol.

In general, SLIP is simpler but less reliable than PPP. Use SLIP in network environments where there is minimal risk of data corruption and where data transfer speed is critical.

Use PPP if you use an ISDN line or when you need to

- Minimize recovery time in network environments that might not take other measures to protect against data corruption.
- Use connection management options such as automatic timeouts and PAP/CHAP user authentication.
- Configure your connection with an unknown IP address.
- Transfer files larger than 4 MB.

If you select either PPP or SLIP, the Setup program installs both types of protocols. However, the protocol you select is the default. After installation, you can switch between the PPP and SLIP protocols by using the Dialer application.

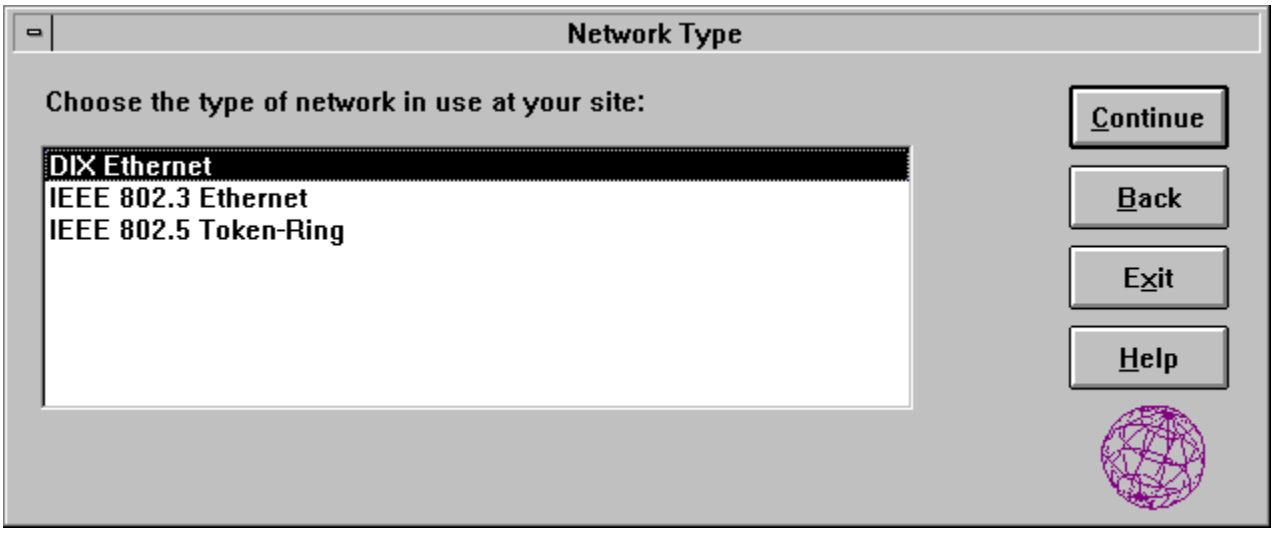
For more information about getting started with the Dialer, see its online Help or the OnNet16

*User's Guide*.

### 3.1.12 Selecting a Network Type

If you do not choose a network card or if your network card supports more than one network type, the Setup program prompts you to specify your network type.

The network type represents the kind of network to which your computer is connected, defined by the way in which transmitted data is packaged (or framed) for that network.




If you are connected to a LAN, the DIX-Ethernet network type is frequently used and is the correct choice for most FDDI (Fiber Distributed Data Interface) installations (using an ODI (Open DataLink Interface) driver). If you are unsure of what type of LAN you are connected to, see your network administrator.

Select the network type used on your network, then choose OK.

### 3.1.13 Configuring Your Internet Protocol (IP) Address

*For LAN installations only*.

In the IP Configuration dialog box, configure a permanent IP address for your computer or choose to obtain your configuration dynamically (when you start your computer) from a Dynamic Host Configuration Protocol (DHCP) or Bootp server. If you are configuring permanent IP values, obtain the appropriate information from your network administrator.

**To dynamically configure your computer's IP address**

1. Select Obtain Configuration from a DHCP Server.

   The addresses group is grayed out and you do not configure any additional information in this dialog box.

2. Choose Continue.

**To permanently configure your computer's IP address**

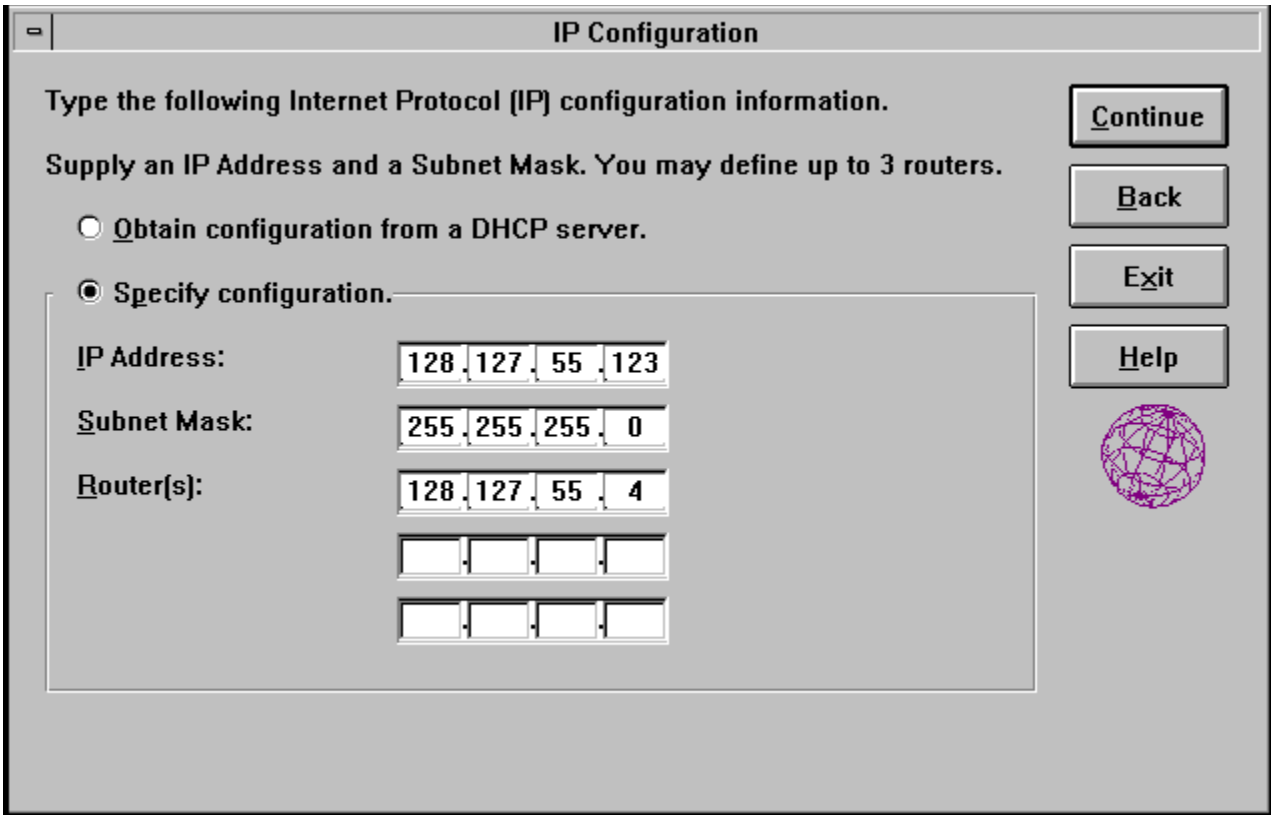
1. Select Specify Configuration.

2. Type the IP address of your computer, the subnet mask for your network, and the IP addresses of network routers (machines that direct network traffic between hosts).

   An IP address uniquely identifies your computer and servers to others on the network. IP addresses follow a standard notation of four groups of numbers separated by dots; for example, 123.75.51.125.

   In each edit box, after you type three digits in front of a dot (.), the program moves the cursor to the space before the next dot. If you type a value of fewer than three digits, you need to use the arrow key or the mouse (or other pointing device), to move to the next space. (Tab moves your cursor to the next edit box.)

3. Choose Continue.



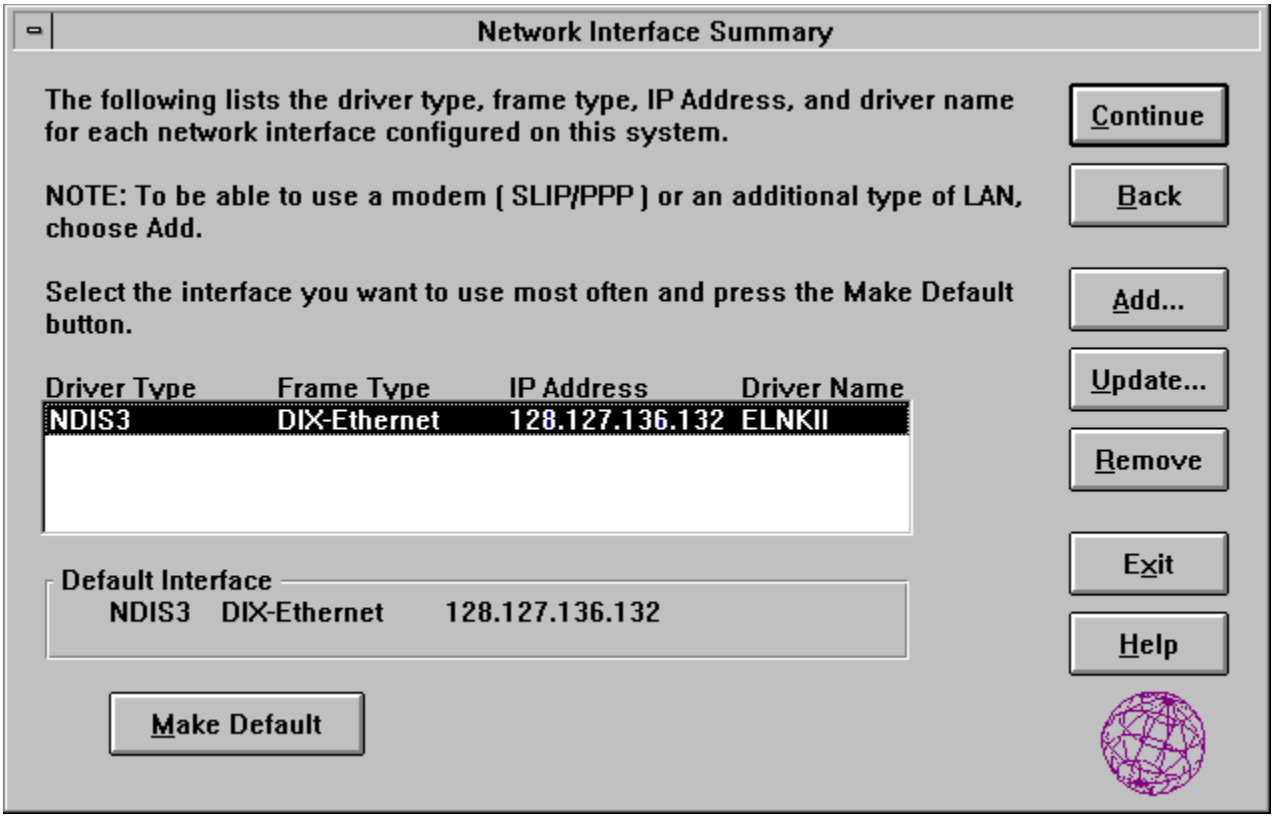

The following table lists and describes the fields in the IP Configuration dialog box:

| Address | Description |
|---|---|
| IP Address | The Internet address of your computer.<br>Required, unless you are configuring only a serial interface. For serial network connections, specify the IP address, if required, in the connection definition. |
| Subnet Mask | Specifies how much of the address to reserve for subdividing networks. The mask contains 1s for the bit positions in the 32-bit address used for the network and subnet parts, and 0s for the host part. The mask should contain at least the standard network portion.<br>Required, unless you are configuring only a serial interface. For serial network connections, specify the subnet mask, if required, in the connection definition. |
| Router(s) | The address(es) of up to three network routers.<br>Required, unless you are configuring only a serial interface. For serial network connections, specify the routers, if required, in the connection definition. |

For more information about Internet addresses, see Part II, “Managing OnNet16 Kernel and Network Services.”

### 3.1.14 Selecting Network Interfaces

In the Network Interface Summary dialog box, confirm, modify, or add network interfaces (such as a PPP or SLIP interface). The dialog box displays the configuration for your existing network interface(s): network driver type, network type, IP address, and network card.




The Setup program copies the files required for each interface to the destination directory you specified earlier in the program. The program configures your computer for the default interface. To reset the default interface, select an interface and choose Make Default.

You do not need to make your PPP or SLIP connection the default interface to use it. Instead, you can easily switch from a LAN connection to a PPP or SLIP connection by using the Dialer; when you close the PPP or SLIP connection, the computer returns to the LAN connection.

To switch between two LAN interfaces, use the Configure utility or manually edit your PCTCP.INI file and other system files.

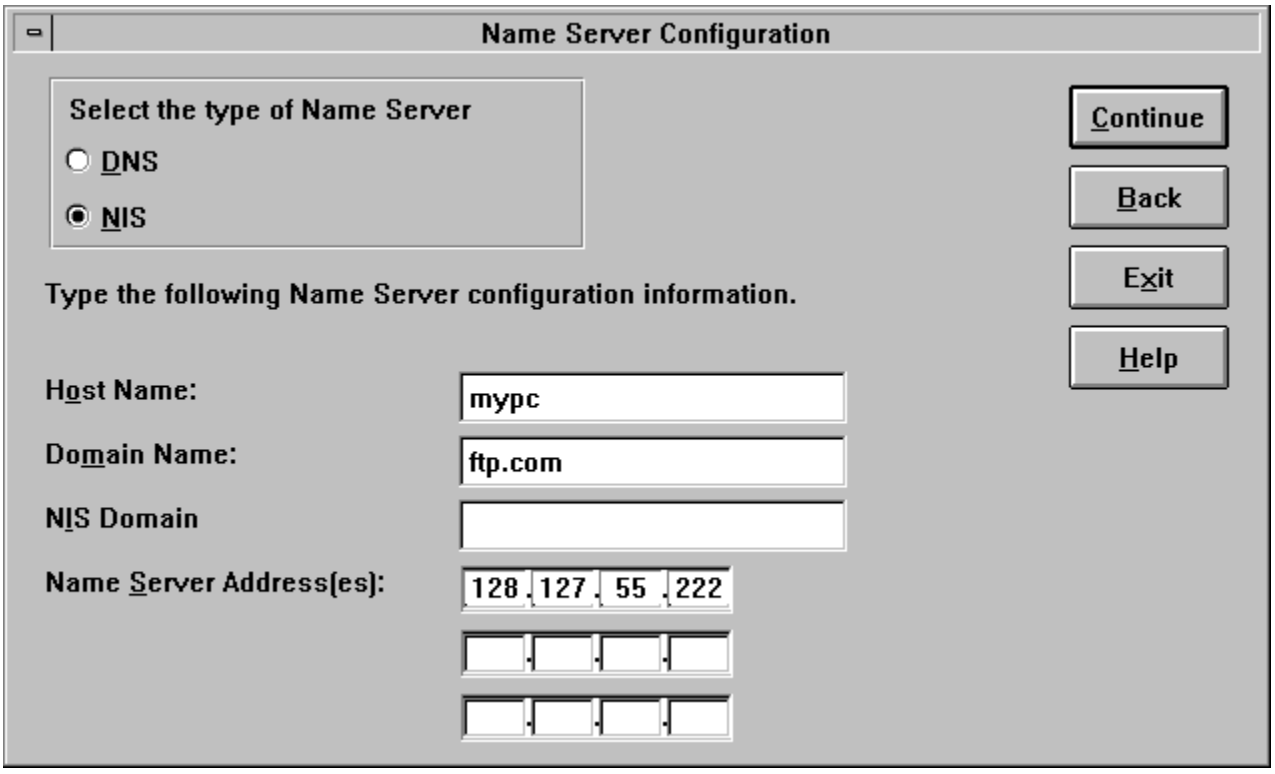
**To accept the configured interface(s) and continue with the installation**

Choose Continue.

For instructions on other tasks you can perform from this dialog box, see the online Help.

### 3.1.15 Configuring DNS or NIS Name Resolution

In the Name Server Configuration dialog box, configure the type of hostname resolution (DNS or NIS) that you use in your network and specify the computers that resolve hostnames.




**To configure name resolution**

1. Select either DNS (Domain Name System) or NIS (Network Information System) name resolution.
2. For each text box, accept the displayed information.
   **–or–**
   Specify a name or address.
3. Choose Continue.

The following table lists and describes the boxes in the Name Server Configuration dialog box:

| Name or Address | Description |
|---|---|
| Host Name | The hostname of your computer (for example, `mypc`).A hostname can be used interchangeably with your IP address to identify your computer on the network. |
| Domain Name | The domain name of your network, separated by periods (for example, xyz.com). The domain name identifies the work group or organization on the network to which your |

| | |
|---|---|
| | computer belongs. |
| NIS Domain Name | The NIS domain name of your network. This box is visible only if you select the NIS type of name server. |
| Name Server Address(es) | The IP address for each name server on your network, in the form 123.145.55.123. A name server translates IP addresses to domain names, and domain names to IP addresses. You can specify up to three name servers. |

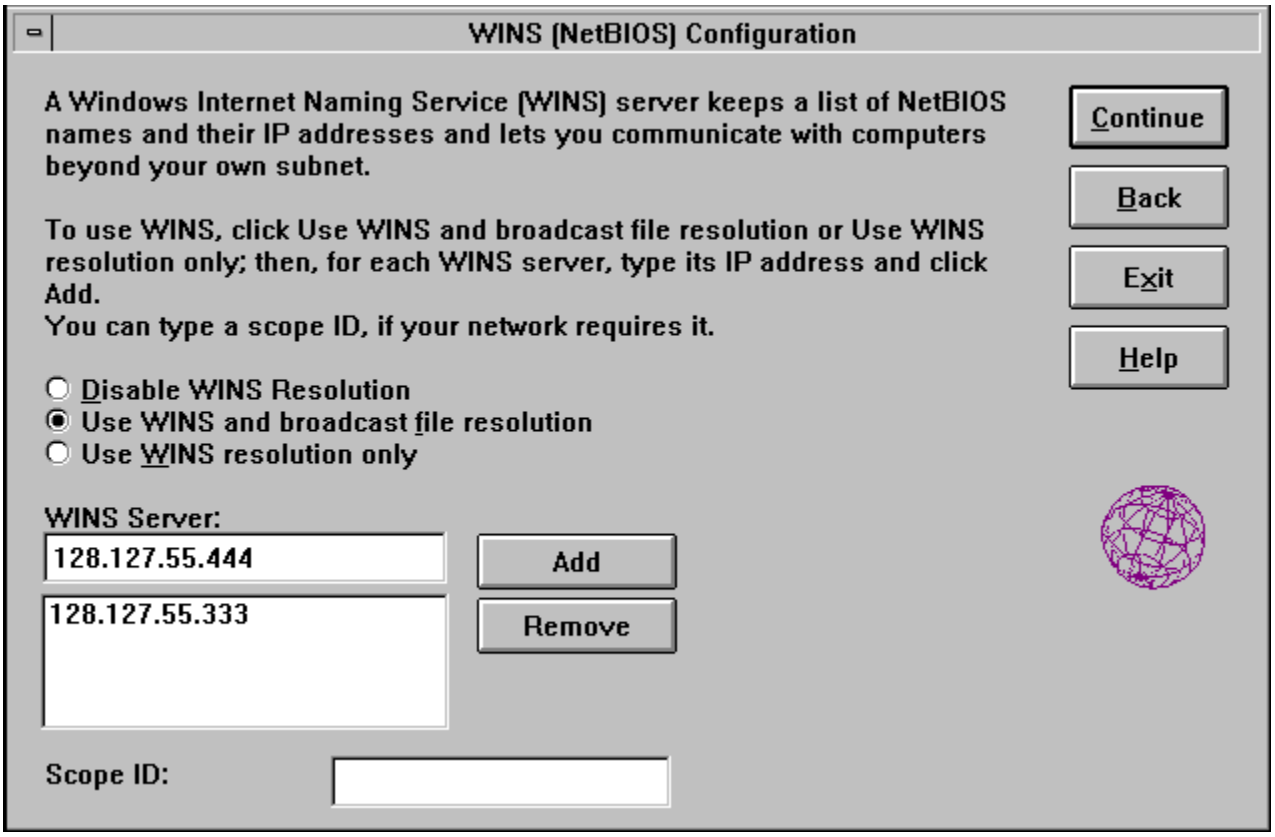
For more information about hostnames and domain names, see the OnNet16 *User's Guide*.

### 3.1.16 Configuring WINS Name Resolution

The Microsoft Client for Microsoft Networks uses the NetBIOS protocol to enable file- and printer sharing between computers running Microsoft Windows 95, Windows for Workgroups, Windows NT, LAN Manager, and other Microsoft-compatible computers. If you install the FTP Software TCP/IP protocol stack, you can continue to use NetBIOS networking in addition to TCP/IP networking.

Typically, NetBIOS relies on broadcast messages to establish communications with other NetBIOS machines. This limits NetBIOS communications to a single subnet. You can get around this limitation by using the Windows Internet Naming Service (WINS) to identify computers on other subnets. WINS maps NetBIOS names to IP addresses.

If WINS cannot resolve the NetBIOS name, the FTP Software TCP/IP protocol stack uses the Domain Name System (DNS) to resolve the name. For this method to work, each computer that will communicate using NetBIOS should have its NetBIOS computer name be the same as its IP hostname. (You specify the NetBIOS computer name by choosing Networks in Windows Control Panel.)




**To configure WINS name resolution**

1. Choose Use WINS and broadcast resolution.
   -or-

Choose Use WINS resolution only.

2. For each WINS server you plan to use, in the WINS Server box, type the server's IP address and choose Add.
3. If your network administrator has told you to do so, type a scope ID.

The following table lists and describes the fields in the WINS (NetBIOS) Configuration dialog box. If you do not know this information, ask your network administrator.

| Option | Description |
| --- | --- |
| Disable WINS resolution | Sets the NetBIOS node type to B, which means that NetBIOS resolves hostnames by using the following three resources, in this order:<br>Name cache on the local computer<br>Broadcasted request<br>NAMES or LMHOSTS file on the local computer |
| Use WINS and broadcast resolution | Sets the NetBIOS node type to H, which means that NetBIOS resolves hostnames primarily by using a WINS server, but uses broadcasts if the name cannot be resolved by a WINS server. |
| Enable WINS resolution only | Sets the NetBIOS node type to P, which means that NetBIOS resolves hostnames by using only WINS servers and does not use broadcasts. |
| Set WINS server | The IP address of up to three NT computers that provide WINS name resolution service. |
| Scope ID | Specifies the scope identifier that this host will use to communicate with other hosts using NetBIOS networking.<br>A scope ID is some arbitrary value, often a workgroup name, assigned by the network administrator. When using TCP/IP to handle NetBIOS traffic, all computers that will communicate with each other using NetBIOS broadcasts must use the same scope ID. |

### 3.1.17 Configuring Your Username

In the Username Configuration dialog box, type the name you want to use when you log in to remote hosts and choose Continue. Several programs and applications, including file transfer, mail and network news (except Mail OnNet), and remote login, use this username by default.

### 3.1.18 Selecting a Time Zone

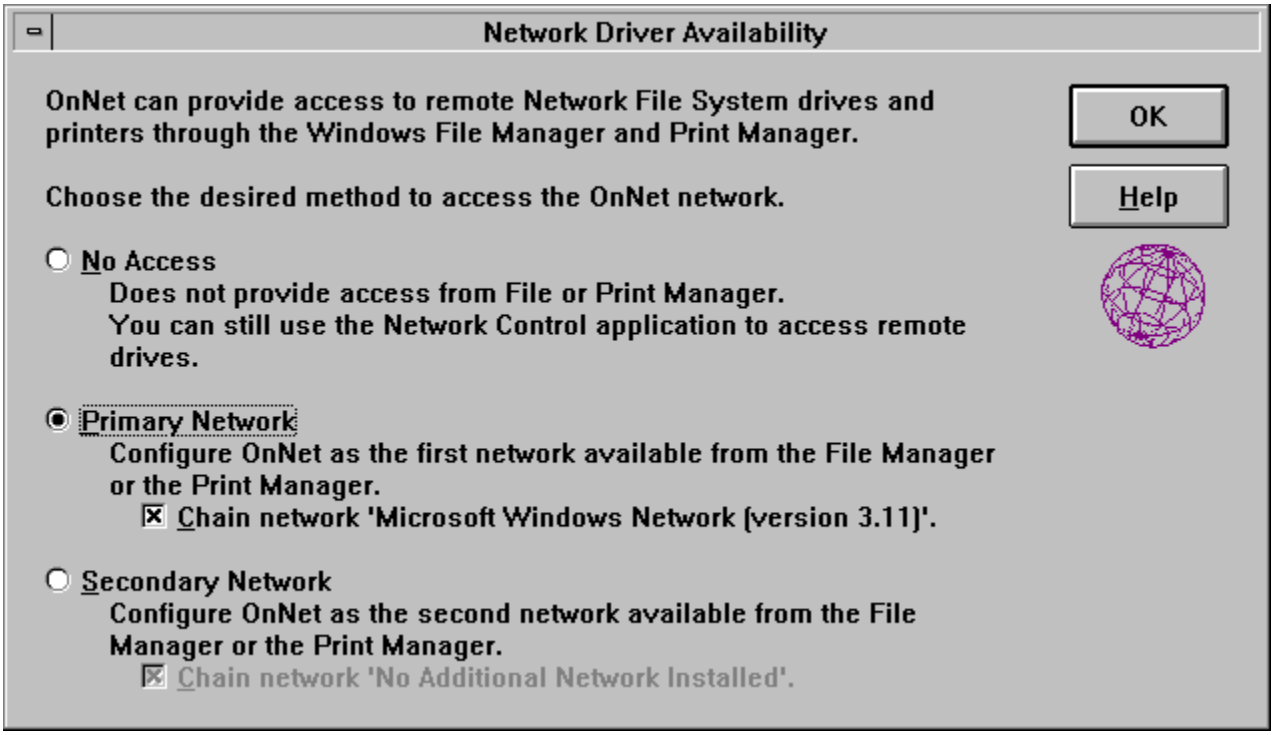
In the Time Zone Configuration dialog box, select the appropriate time zone for your geographic area, and choose Continue.

If you are upgrading, note that the time zone selected by default is determined by an entry in the Setup program file SETUP.INF, not by your previous version of OnNet16. (See Chapter 6, Customizing a Centralized Installation and Configuration.) If you are unsure which time zone to select, ask your network administrator.

**Note:** Use the Network Time application to set your computer's clock from a network time server.

### 3.1.19 Selecting Network Driver Availability

If you selected Network File and Print Services in the Components dialog box (see Section 3.1.7, Selecting Components), in the Network Driver Availability dialog box, choose whether to access remote files and services through the Windows File Manager, Print Manager, and Control Panel. If you use other network operating systems (such as Microsoft's Windows for Workgroups or Novell NetWare) in addition to FTP Software NFS, you can also choose the order in which you access the services of each network. Note that the default for this dialog box depends on what other network operating systems are already installed on your computer.




The following table lists and describes options for selecting your network driver availability:

| Select | To |
|---|---|
| No Access | Continue using another network operating system that does not chain. With this selection, you use the other network operating system from File Manager, Print Manager, or Control Panel. You can mount remote drives and printers, which use FTP Software NFS, from the FTP Software Network Control application. |
| Primary Network | Configure FTP Software NFS as the first network available from File Manager, and to chain the existing network to FTP Software NFS. |
| Secondary Network | Configure FTP Software NFS as the second network available from File Manager, chained to an existing network. |

### 3.1.20 Specifying a Firewall Pass-through Server

The Firewall Pass-Through Configuration dialog box appears only if you are doing a Custom or New installation and you selected Firewall Pass-Through (SOCKS) in the Components dialog box. If you are not configuring a firewall server, skip to Section 3.1.22, Copying Files.

In the Firewall Pass-Through Configuration dialog box, specify the hostname or, preferably, the IP address of a default SOCKS server to use. This information is required unless the SOCKS.CNF configuration file specifies a SOCKS server. The SOCKS server is your entry point into a network that is protected by a SOCKS security firewall. Any application that uses TCP and that runs over the Winsock DLL can use SOCKS security.

After the installation, edit the SOCKS.TXT file in the ONNET\ETC directory and save it as SOCKS.CNF. See Chapter 16, Configuring SOCKS Security, in Part III, “Using Network Security,” for information about the contents of this file.

### 3.1.21 Running the Windows for Workgroups Network Setup

*For installations on computers with Microsoft Network.*

If you are installing OnNet16 on a computer that uses Microsoft Network, the Setup program provides you the option to run the Windows for Workgroups Network Setup. If you choose Start, the Setup program starts up Network Setup.

Perform the following steps:

**To set up FTP Software Network Driver to work with Windows for Workgroups**

1. In the Network Settings group box, choose Drivers .
2. In the Network Drivers group box, select the driver you want to use and choose Add Protocol.
3. Select Unlisted or Updated Protocol, and choose OK.
4. Type the destination directory where you installed OnNet16. For example, `C:\PCTCP`
5. Choose OK.
6. Close the Network Drivers dialog box.
7. In the Network Settings group box, choose Networks.
8. Choose one of the following network configurations for your computer. If you selected the FTP Software Network Driver as the primary or secondary network driver during installation, you should choose one of the following matching options.

   *To configure your computer to use both Windows for Workgroups and the FTP Software Network Driver*

   Select Install Microsoft Windows Network, then choose Other and select FTP Software PC/TCP Network. Then choose OK.

   *To configure your computer to use the FTP Software Network Driver only*

   Select Install Windows Support for the Following Network, then select FTP Software Network Driver. Then choose OK.
9. Choose OK to close the Network Setup dialog box.
10. Choose Restart so that the Setup program prompts you to reboot the computer.
11. Return to the Setup program.

### 3.1.22 Copying Files

In the Copy Files dialog box, choose Continue to have the Setup program copy the appropriate OnNet16 files onto your computer. If you are installing from disks, insert new disks, as directed by the program.

After copying the files, the Setup program builds the driver information database, edits system startup files, and adds icons to the program groups. Then, the Setup program displays the Setup Complete dialog box.

Read the contents of this box and then choose OK. Note that the file INSTALL.LOG contains a list of the startup files that the Setup program edited and that the Setup program saves your original files with an FTP extension.

**Note:** If you use DOS 6.0 multiple configurations or other conditionalized system files, you must update your system files manually, then reboot, before you can use OnNet16. In this case, use the Setup program template files AUTOEXEC.FTP and CONFIG.FTP to help update your system files.

### 3.1.23 Restarting Your Computer

When the Restart System dialog box appears, choose Restart System. (Remember to remove the installation disk before restarting your computer.) The Setup program performs appropriate cleanup tasks and then restarts your computer.

**Note:** If you do not let the program restart your computer, you must manually perform the cleanup tasks and restart your computer before you can use OnNet16. Ensure that you have updated your DOS system files before rebooting.

To perform the cleanup tasks manually, exit from Windows and type the **ftpclean** command from your top-level DOS directory. (Note that you might also be prompted to run this command by FTP NFS.)

The **ftpclean** command updates some files, ensuring that only the appropriate versions of OnNet16 programs are installed on your computer.

## 3.2 Testing Network Connectivity

After you have completed the Setup program and have rebooted your computer, you need to verify that you are connected to your network.

If you installed a PPP or SLIP interface and do not have a network connection, you need to connect to an Internet service provider before you attempt to verify your network connectivity.

- If you have a PPP or SLIP network account established, use the Dialer application to establish a connection to your Internet service.
- If you do not have a PPP or SLIP network account established, contact your preferred local Internet service provider to establish your Internet service; then use the Dialer application to establish a connection to your Internet service.

**To verify network connectivity**

Use the Ping application, specifying the network address of a functioning host on your network.

If the host responds, you have a successful installation. If the host does not respond, use the procedures described in Section 3.3, Troubleshooting the Installation.

## 3.3 Troubleshooting the Installation

The Setup program generally configures your computer for use with OnNet16 and for use with any other network operating system(s) installed on your computer. If you could not connect to the network, review this section.

**To troubleshoot the installation**

1. Ensure that you have restarted your computer. Changes to your system files and to OnNet16 configuration files will not take effect until you restart your computer. Typically, you let the Setup program restart your computer at the end of the installation process.

2. Ensure that you ran **`ftpclean.exe`** if you did not let the Setup program restart your computer.

3. Review the INSTALL.LOG file in your destination directory. This file lists the actions taken by the Setup program, including changes made to system files. Depending on your installation, the Setup program creates or updates the PCTCP.INI configuration file, AUTOEXEC.BAT, and Windows system files, if necessary.

4. Ensure that the PCTCP environment variable is set. You can confirm this variable by typing **`SET`** at a DOS prompt or in a DOS session in Windows.

   The PCTCP environment variable is set if you see a line similar to the following:

   ```
   PCTCP=C:\PCTCP\PCTCP.INI
   ```

5. Verify that you have configured a valid license key and IP address. (Use the Internet Addresses dialog boxes in the Configure application to view or set IP addresses. Use the License Upgrade application to set a license key.)

6. Use the information in Chapter 5, Verifying Your Network Installation, to verify that your system files (AUTOEXEC.BAT and CONFIG.SYS) and network driver parameters and configuration files (such as NET.CFG and PROTOCOL.INI) have been correctly updated.

7. Verify that your SERVICES and PROTOCOL files are located in the directory specified by the Work (etc) directory parameter. (Use the General parameters group dialog box in the Configure application to view or set the location of the Work (etc) directory.)

8. Restart your computer and record any system error messages.

### 3.3.1 Verifying the PCTCP.INI File

Depending on your installation, the Setup program creates or updates the PCTCP.INI configuration file, DOS system files, and other network or Windows files, if necessary. The Setup program records the changes it makes to your system files in the file INSTALL.LOG, located in your destination directory.

By default, the Setup program creates a configuration file called PCTCP.INI. Regardless of your network driver or protocol stack configuration, the Setup program creates the PCTCP.INI file sections and parameters that you need for basic networking functionality.

The following example shows a sample PCTCP.INI file with the minimum required information to run OnNet16 with an Ethernet over a packet driver configuration. Your PCTCP.INI file will look similar to the following example, but might contain additional configuration information, such as sections for InterDrive and mail applications.

```
[pctcp general]
host-name = mypc
user = pat

[pctcp addresses]
domain-name-server = 128.123.54.122
domain-name = xyz.com

[pctcp kernel]
authentication-key = 5678-56789-5678
interface = ifcust 0

[pctcp ifcust 0]
router = 123.145.51.4
ip-address = 123.145.51.125
subnet-mask = 255.255.255.0
frame-type = DIX-Ethernet
interface-type = Pktdrv
```

**To use InterDrive**

```
[pctcp idrive]
[pctcp idrive-vxd]
```

**Note:** The `interface-type=` and `frame-type=` parameters are always written to PCTCP.INI, even if they are not required by the protocol stack installed on your computer.

See Chapter 5, Verifying Your Network Installation, which describes in more detail the changes to the PCTCP.INI file that the Setup program makes, or that you might need to manually make, in specific networking environments.

**To verify changes to other system files**

Depending on the driver that you selected during installation, the Setup program might update

other system files. The Setup program automatically updates the appropriate system files when you install OnNet16 to work with a packet driver (including SLIP and PPP), NDIS driver (NDIS3 or NDIS2), or ODI driver. For installations using ASI drivers, you might need to manually update your system files. See Chapter 5, Verifying Your Network Installation, for details on what system files the Setup program creates or updates.

## 3.4 Rerunning the Setup Program

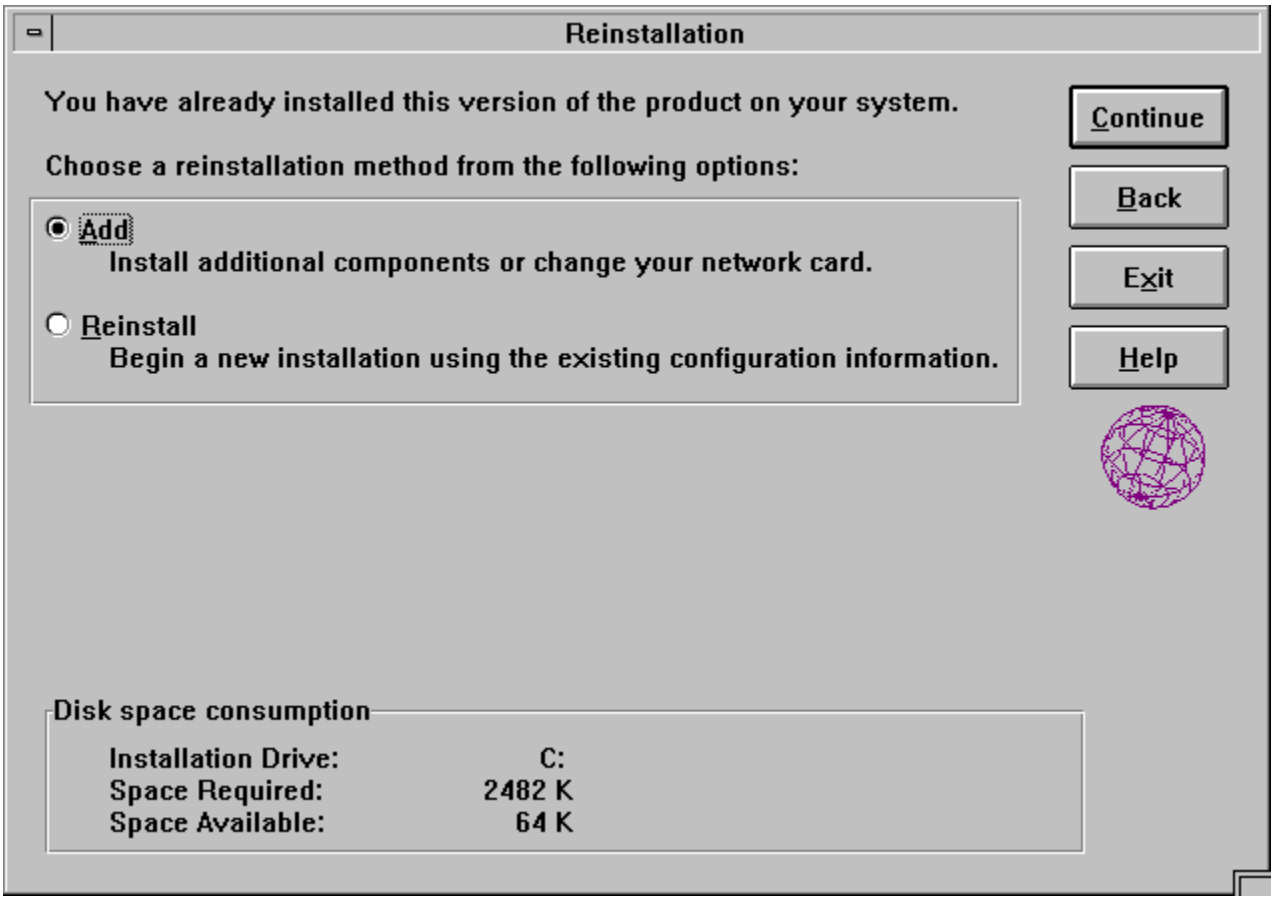
After you have installed this release, you can rerun the Setup program to make changes to your configuration. For example, you might want to

- Define a different network interface (or multiple interfaces) for your computer because you changed network cards or added a modem or ISDN line.
- Reset the time zone in response to a move.
- Add components.
- Redefine an interface to reflect a change of IP address, or to add another router.

While you can perform most of these tasks with the Configure application, you might prefer to use the step-by-step interface in the Setup program.

You can also choose to reinstall OnNet16. You might want to do this if you need to recopy files.

To rerun the Setup program, run setup as described in Section 3.1, Running the Setup Program.




From the Reinstallation dialog box, select Add to change your network driver or to configure

another network interface. You can add, change, or remove your network interface or define multiple network interfaces (although the Setup program configures your computer to use only one interface at a time).
-or-
From the Reinstallation dialog box, **select** Reinstall to start the Setup program at the beginning and use existing configuration information as you move through the program dialog boxes.Specify a different destination directory if you want to install multiple configurations.
-or-

From the Reinstallation dialog box, choose Help to view instructions about reinstalling or adding components.

Note that you cannot use the Setup program to remove files. To remove files, delete all files in the destination directory that you specified at the beginning of the program, then restore your backup files for the AUTOEXEC.BAT and SYSTEM.INI files, if applicable. You should also remove the program groups from Program Manager.

# Chapter 4
# Configuring OnNet16

As you continue to use OnNet16, you might need to customize your configuration to reflect changes in your network environment or to improve protocol stack (kernel) or network performance. This chapter introduces the different ways to configure OnNet16. For more information, see Part II, “Managing OnNet16 Kernel and Network Services.”

You can configure OnNet16 by

- Using the Configure application to revise configuration parameters, create multiple configuration files, or change your network interface.
- Using DHCP (Dynamic Host Configuration Protocol) to obtain configuration parameters from a network server.
- Manually editing the PCTCP.INI configuration file.

Whichever method you choose, the parameters that you set are stored in the PCTCP.INI configuration file, in the directory where you installed OnNet16.

In addition, you can configure most applications by setting options within the application itself. Some options that you set in the applications themselves are stored in a separate configuration file (such as SESSION.INI).

## 4.1 Using the Configure Application

The Configure application lets you set configuration parameters for the protocol stack and some applications.

The Configure application also lets you create multiple configuration files, and some Advanced mode commands let you make immediate changes to the FTP Software protocol stack.

You can choose to view and edit parameters in Basic mode or in Advanced mode. Basic mode displays only the parameters needed to provide basic configuration for the FTP Software protocol stack and the applications that you installed. Advanced mode displays all possible parameters for your system and identifies which of those parameters are currently in use.

If you are experienced with OnNet16 configuration, you can also use the Configure application to edit any PCTCP.INI file parameter.

***Caution:*** Be certain that you do not have the configuration file open in Notepad or in any other text editor while editing it in the Configure application.

You can use Configure to set values for DOS terminal emulation, mail, and news applications and Kerberos security.

You can configure InterDrive, Mail OnNet, TNVTPlus, TN3270/5250, and Print Server from within the application. In general, you should configure a Windows application from within the application itself.

**To run the Configure application**

From the OnNet16 program group, double-click the Configure icon.


**–or–**

From the Windows Program Manager File menu, choose Run, then type

```
C:\PCTCP\WCONFIG.EXE
```

and choose OK.

The main window of the Configure application appears, as shown in Figure 4-1. From this window, you can choose commands from the menu bar, display the toolbar, and select the parameters that you want to configure. Use the Configure application toolbar to move quickly between configuration modes. By default, the toolbar is enabled.

**To enable or disable the Configure toolbar**

From the Settings menu, choose Display Toolbar. By default, the Configure application displays the toolbar.

**To get online help**

From the Configure application, choose the Help button.
**–or–**
From the Configure application, choose Help from the menu bar.

From most Help topics you can choose to view a description of a parameter that appears in that dialog box and view instructions for editing these parameters. Other Help topics describe, in a general way, how to use the Configure application.

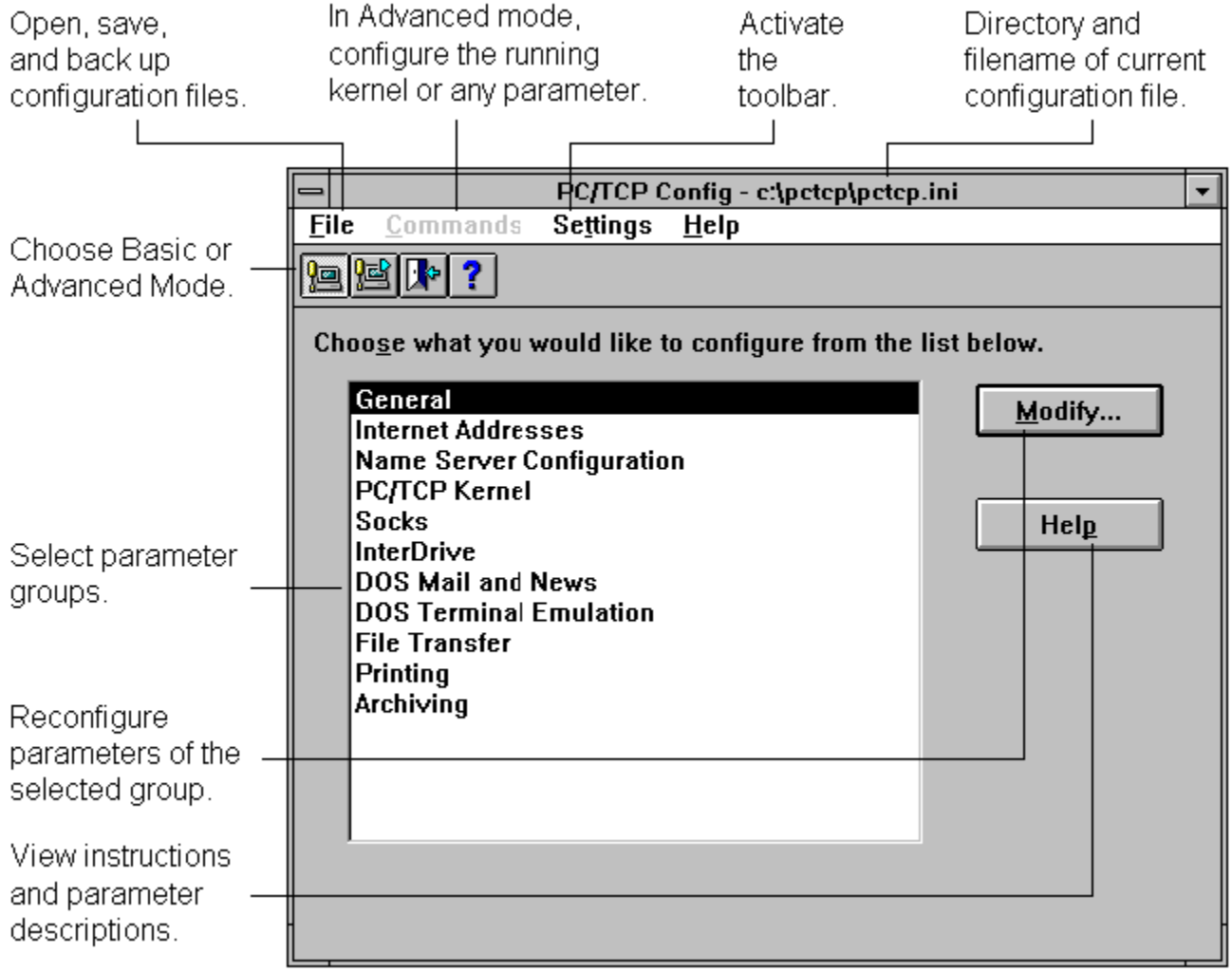



Figure 4-1 The Windows Configure Application Main Window

## 4.2 Configuring OnNet16 by Using the DHCP Client

You can also configure OnNet16 with DHCP to obtain network configuration information from a network server, such as IP addresses for your PC and network servers. Each time that you start your PC, your PC sends a message to the server requesting the values of specific parameters. This central configuration method lets a system administrator standardize and quickly change network information for a large group of users.

This method of configuration requires that your system administrator set up a DHCP or Bootp server on your network. Once a server is set up, users on the network can obtain a temporary or permanent IP address and other network information, including server addresses, default routers, and Domain Name System (DNS) server addresses.

Note that you can also configure your system to use DHCP at startup. To do so, during the installation program, select Obtain Configuration from a DHCP Server in the IP Configuration dialog box. After installation, you can use the Configure application to configure the use of DHCP and its parameters, if necessary.

Network administrators can use Custom Installation Manager to configure the installation program so that the Obtain Configuration from a DHCP Server option is the default in the IP Configuration dialog box. Refer to Chapter 6, Customizing a Centralized Installation and Configuration.

For more information about how to start and use DHCP, see Part II, “Managing OnNet16 Kernel and Network Services.”

## 4.3 Configuring OnNet16 by Editing the PCTCP.INI File

You can also configure OnNet16 software by directly editing the PCTCP.INI configuration file with an ASCII editor. This file contains parameter groupings that relate to different applications.

The PCTCP.INI file is located in the directory in which you installed OnNet16 software (C:\PCTCP if you accepted the installation program default).

For detailed information about PCTCP.INI configuration file sections and the specific parameters that apply to each section, visit the FTP Software home page on the World Wide Web at http://www.ftp.com/techsup/quick-help/docs/. Look under News and Doc in the Technical Support area for the Configuration Parameters Reference documentation.

# Chapter 5
# Verifying Your Network Installation

This chapter identifies the changes to system files resulting from installing the OnNet16 kernel (protocol stack) and describes any additional steps that must be performed in order to run the OnNet16 kernel with another NOS. You can use the information to verify what changes the installation program made to your system, or to learn alternate methods for configuring the OnNet16 kernel.

This chapter provides information about configuring the OnNet16 kernel in the following network environments:

- Banyan VINES
- Novell NetWare
- Packet Drivers, including serial line and AppleTalk
- Windows for Workgroups

For information about kernel interoperability with a NOS not listed above, contact Technical Support or your service provider.

This chapter also identifies how to use the FTP Software Network Driver for Windows with other network operating systems (chaining) and suggests ways in which to troubleshoot your installation.

Before you install the OnNet16 kernel in any network environment, review the information and perform the procedures described in Chapter 2, Preparing for Installation, and in the section of this chapter that applies to your network environment.

The samples from configuration files in this chapter use C:\PCTCP as the path designation for the OnNet16 files.

## 5.1 Installing the OnNet16 Kernel with Banyan VINES

Use the OnNet16 kernel with Banyan VINES 4.1 or later to add TCP/IP capabilities to your PC.

You can use Banyan VINES and the OnNet16 kernel concurrently, as follows:

- When sharing the Banyan Ethernet driver
- When sharing an NDIS (Network Driver Interface Specification) driver and using an Ethernet network card
- When sharing an ASI (Adapter Support Interface ) driver and using a Token Ring network card
- When encapsulating (“tunneling”) IP packets within VINES transport services

Before you install the OnNet16 kernel with Banyan VINES, initialize the interface with the VINES driver by running the Banyan **ban** command.

**To use the Banyan Ethernet driver**

When you are prompted by the installation program , select the default Banyan Ethernet driver in the Found Existing Driver dialog box.

Table 5-1 displays the system file entries that support the OnNet16 kernel and Banyan VINES interoperability with the Banyan Ethernet driver.

Table 5-1 Sharing the Banyan Ethernet Driver

| File | Entries |
|---|---|
| AUTOEXEC.BAT | `cd \drivers\banyan`<br>`ban`<br>`cd \`<br>`path=c:\pctcp;%path%`<br>`set pctcp=c:\pctcp\pctcp.ini`<br>`c:\pctcp\vxdinit` |
| PCTCP.INI | `[pctcp ifcust 0]`<br>`ip-address= 128.127.50.1`<br>`router=128.127.50.2`<br>`subnet-mask=255.255.255.0`<br>`frame-type= DIX-Ethernet`<br>`interface-type= EBANYAN` |

**To use an NDIS driver**

1. In the Found Existing Driver dialog box, select the default existing network driver.

**-or-**

Choose Install a New Driver and select Other NDIS Driver in the Network Card dialog box.

2. Typically select the DIX Ethernet network (frame) type.

Table 5-2 displays the system file entries that support the OnNet16 kernel and Banyan VINES interoperability with an NDIS driver.

Table 5-2 Sharing an NDIS Driver with Banyan VINES

| File | Entries |
|---|---|
| AUTOEXEC.BAT | `cd \drivers\banyan`<br>`ban`<br>`cd \`<br>`path=c:\pctcp;%path%`<br>`set pctcp=c:\pctcp\pctcp.ini`<br>`c:\pctcp\vxdinit` |
| CONFIG.SYS | `device=c:\directory\protman.dos /I:C:\directory`<br>`device=c:\directory\`*`your_NDIS_driver`*<br>`device=c:\pctcp\dis_pkt.gup` |
| PROTOCOL.INI | `[PKTDRV]`<br>`drivername=PKTDRV$`<br>`BINDINGS=ELNKII`<br>`intvec=0X60`<br>`chainvec=0X65`<br><br>`[ELNKII]` |
| PCTCP.INI | `[pctcp ifcust 0]`<br>`ip-address= 128.127.50.1`<br>`router=128.127.50.2`<br>`subnet-mask=255.255.255.0`<br>`frame-type= DIX-Ethernet`<br>`interface-type= PKTDRV` |

In the PROTOCOL.INI file, set `BINDINGS=` to the label used in the card-specific section in the same file (as in the sample).

**To use an ASI driver**

1. Select the default network driver in the Found Existing Driver dialog box.

2. Select the IEEE 802.5 Token Ring network (frame) type.

Table 5-3 displays the system file entries that support the OnNet16 kernel and Banyan VINES interoperability with an ASI driver and a Token Ring network card.

Table 5-3 Sharing ASI Drivers with Banyan VINES

| File | Entries |
|---|---|
| AUTOEXEC.BAT | `cd \drivers\banyan`<br>`ban`<br>`cd \`<br>`path=c:\pctcp;%path%`<br>`set pctcp=c:\pctcp\pctcp.ini`<br>`c:\pctcp\vxdinit` |
| PCTCP.INI | `[pctcp ifcust 0]`<br>`ip-address= 128.127.50.1`<br>`router=128.127.50.2`<br>`subnet-mask=255.255.255.0`<br>`frame-type= Token-Ring`<br>`interface-type= IBMTR` |
| CONFIG.SYS | `device=c:\asi\dxma0mod.sys`<br>`device=c:\asi\dxmc0mod.sys`<br>`device=c:\asi\dxmt0mod.sys ES=2 O=Y DS=2048` |

**To use IBANYAN**

After you install the OnNet16 kernel, configure your system manually.

During the installation, choose not to use existing drivers, and select None when you are prompted for a network interface card. After the installation, configure this kernel by editing the system files, as shown in Table 5-4.

Table 5-4 displays the system file entries that support the OnNet16 kernel and Banyan VINES interoperability by TCP/IP encapsulation.

Table 5-4 Tunneling TCP/IP Packets with VINES Transport

| Files | Entries |
|---|---|
| AUTOEXEC.BAT | `cd \drivers\banyan`<br>`ban`<br>`cd \`<br>`path=c:\pctcp;%path%`<br>`set pctcp=c:\pctcp\pctcp.ini`<br>`c:\pctcp\vxdinit` |
| PCTCP.INI | `[pctcp ifcust 0]`<br>`ip-address = 128.127.50.100` |

```
subnet-mask = 255.255.255.0
router = 128.127.50.6
bserver= Banyan_server_name
frame-type= IBANYAN
interface-type=IBANYAN
```

## 5.2 Installing the OnNet16 Kernel with Novell NetWare

Use the OnNet16 kernel with Novell NetWare to add TCP/IP capabilities to your PC. You can use NetWare and the OnNet16 kernel concurrently, as follows:

- When sharing ODI (Open Data Link Interface) Ethernet or token ring drivers
- When sharing ASI token ring drivers
- When sharing an Ethernet packet driver
- When sharing an NDIS driver
- With NetWare/IP using the LWPE VxD from FTP Software and ODI drivers.

  The LWPE VxD lets you run LAN WorkPlace applications, including Novell's NetWare/IP application, over the OnNet16 kernel. This installation requires you to manually copy and configure some files.

**To use ODI drivers**

1. In the Found Existing Driver dialog box, select the default existing network driver.

   **-or-**

   Choose Install a New Driver and select a driver from the Network Card dialog box.

2. If you are installing the OnNet16 kernel, typically select the DIX Ethernet network (frame) type.

3. After completing the installation, edit the AUTOEXEC.BAT, PCTCP.INI, and NET.CFG files, as shown in Table 5-5.

Table 5-5 displays the system file entries that support the OnNet16 kernel and Novell NetWare interoperability with ODI drivers.

Table 5-5 Sharing ODI Drivers with Novell NetWare

| File | Entries |
|---|---|
| AUTOEXEC.BAT | `path=c:\pctcp;%path%`<br>`set pctcp=c:\pctcp\pctcp.ini`<br>`c:\nwclient\lsl.com`<br>`c:\nwclient\cardname.com`<br>`c:\nwclient\ipxodi.com`<br>`c:\pctcp\odipkt.com`<br>`c:\nwclient\vlm.exe`<br>`c:\pctcp\vxdinit` |
| NET.CFG | `link driver` *`cardname.com`*<br>`int 3` |

```
   port 280
   mem D8000
   frame ETHERNET_802.3
   frame ETHERNET_802.2
   frame ETHERNET_II

NetWare DOS Requester ;disables packet burst support
   PB BUFFERS 0       ;(and avoids unexpected exit from
                      ;Windows)
```

PCTCP.INI

```
[pctcp ifcust 0]
ip-address=128.127.50.1
router=128.127.50.2
subnet-mask=255.255.255.0
frame-type= DIX-Ethernet
interface-type= PKTDRV

[pctcp ifcust 0]
odi-pkts=8     ;provides optimal ODI performance
```

The `frame-type=` in the PCTCP.INI file specifies your network media type.

Determine the frame type from the following table:

| If you are using this NetWare frame type | Use this OnNet16 kernel frame type |
|---|---|
| Ethernet_802.3 | DIX-Ethernet |
| Ethernet_802.3 and Ethernet_802.2 | IEEE |
| Token-Ring_802.5 or Token-Ring | Token-Ring |

**To use NetWare Connect 2.0**

1. Edit your AUTOEXEC.BAT and PCTCP.INI files as shown in Table 5-6.

Table 5-6 OnNet16 Kernel with NetWare Connect 2.0

| File | Entries |
|---|---|
| AUTOEXEC.BAT | `PATH=c:\pctcp;c:\net\bin;%PATH% ;onnet should precede ;netware`<br>`PCTCP=c:\pctcp\pctcp.ini`<br>`lsl.com /C=c:\net\net.cfg ;Novell ODI driver`<br>`nesl.com ;Novell ODI driver` |

```
ncomx.com ;Novell ODI driver
nwremote.com ;Novell ODI driver
c:\pctcp\vxdinit.exe ;the installation program adds this
;entry
c:\pctcp\lwpe.com ;add this entry
```

PCTCP.INI

```
[pctcp ifcust 0]
interface-type=SERIALODI ;add this entry
frame-type=PPP ;(or SLIP)
ip-address=xxx.xxx.xxx.xxx
subnet-mask=255.255.255.0
[pctcp vxdinit]
vlwpe=yes
```

2. Remove any instances of WCOMVXD.386 and WCOMAPI.DLL from the SYSTEM.INI file.
3. Confirm that the NET.CFG file is configured properly for a SLIP or PPP connection (refer to the NetWare Connect 2.0 documentation.)

Use the NetWare Connect 2.0 Windows Dialer program to dial your NetWare Server that is running NetWare Connect 2.0. (The FTP Software Dialer does not work in this environment.)

**To use an NDIS driver**

1. In the Found Existing Driver dialog box, select the default existing network driver.

   **-or-**

   Choose Install a New Driver and select Other NDIS Driver in the Network Card dialog box.
2. Typically select the DIX Ethernet network (frame) type.
3. After the installation, generate a new NetWare IPX shell.

   **Note:** The NetWare shell generationand linker program is called WSHGEN.EXE for NetWare 3.*x* and later.

Table 5-7 displays the system file entries that support the OnNet16 kernel and Novell NetWare interoperability with an NDIS driver.

Table 5-7 Sharing an NDIS Driver with Novell NetWare

| File | Entries |
|---|---|
| AUTOEXEC.BAT | `c:\lanman\netbind`<br>`c:\nwclient\ipx.com`<br>`c:\nwclient\vlm.com`<br>`path=c:\pctcp;%path%` |

| | |
|---|---|
| | `set pctcp=c:\pctcp\pctcp.ini`<br>`c:\pctcp\vxdinit` |
| CONFIG.SYS | `device= c:\lanman\protman.dos /I:c:\lanman`<br>`device=c:\lanman\`*`your_NDIS_driver`*<br>`device=c:\pctcp\dis_pkt.gup` |
| PCTCP.INI | `[pctcp ifcust 0]`<br>`ip-address= 128.127.50.1`<br>`router=128.127.50.2`<br>`subnet-mask=255.255.255.0`<br>`frame-type=DIX-Ethernet`<br>`interface-type=PKTDRV` |
| PROTOCOL.INI | `[PKTDRV]`<br>`drivername=PKTDRV$`<br>`BINDINGS=ELNKII`<br>`intvec=0X60`<br>`chainvec=0X65`<br><br>`[ELNKII]` |

In the PROTOCOL.INI file, set `BINDINGS=` to the label used in the card-specific section.

**To use an ASI driver over Token Ring**

1. In the Found Existing Driver dialog box, select the default existing network driver.

   **-or-**

   Choose Install a New Driver and select a driver from the Network Card dialog box.

2. Typically select the IBM 802.5 Token Ring network (frame) type.

3. After the installation, generate a new NetWare IPX shell.

   **Note:** The NetWare shell generationand linker program is called WSHGEN.EXE for NetWare 3.*x* and later.

Table 5-8 displays the system file entries that support the OnNet16 kernel and Novell NetWare interoperability with ASI drivers and Token Ring.

Table 5-8 Sharing ASI Drivers with Novell NetWare

| File | Entries |
|---|---|
| AUTOEXEC.BAT | `path=c:\pctcp;%path%`<br>`set pctcp=c:\pctcp\pctcp.ini`<br>`c:\nwclient\lsl.com`<br>`c:\asi\lansup.com` |

```
c:\nwclient\ipxodi.com
c:\pctcp\odipkt.com
c:\nwclient\vlm.exe -or-  c:\nwclient\netx.com
c:\pctcp\vxdinit
-or-
path=c:\pctcp;%path%
set pctcp=c:\pctcp\pctcp.ini
c:\nwclient\lsl.com
c:\asi\lansup.com
c:\nwclient\ipx.com
c:\nwclient\vlm.exe -or-  c:\nwclient\netx.com
c:\pctcp\vxdinit
```

CONFIG.SYS

```
device=c:\asi\dxma0mod.sys
device=c:\asi\dxmC0mod.sys
device=c:\asi\dxmt0mod.sys ES=2 O=Y ds=2048
```

PCTCP.INI

```
[pctcp ifcust 0]
ip-address= 128.127.50.1
router=128.127.50.2
subnet-mask=255.255.255.0
frame-type=Token-Ring
interface-type=PKTDRV
```

NET.CFG

```
link driver lansup
     int 3
     port 280
     mem D8000
     frame TOKEN_RING
     frame TOKEN_RING_SNAP
```

**To use an Ethernet packet driver**

1. In the Found Existing Driver dialog box, select the default existing network driver.

   **-or-**

   Choose Install a New Driver and select a driver from the Network Card dialog box.

2. Typically select the DIX Ethernet network (frame) type.

Table 5-9 displays the system file entries that support the OnNet16 kernel and Novell NetWare interoperability with an Ethernet packet driver.

Table 5-9 Sharing an Ethernet Packet Driver with Novell NetWare

| File | Entries |
|---|---|

| | |
|---|---|
| AUTOEXEC.BAT | `c:\pctcp;%path%`<br>`set pctcp=c:\pctcp\pctcp.ini`<br>`c:\pctcp\3c503.com /n 0x60 5 0x280 0xc800`<br>`c:\nwclient\ipx.com`<br>`c:\nwclient\vlm.com`<br>`c:\pctcp\vxdinit` |
| PCTCP.INI | `[pctcp ifcust 0]`<br>`ip-address= 128.127.50.1`<br>`router=128.127.50.2`<br>`subnet-mask=255.255.255.0`<br>`frame-type=DIX-Ethernet`<br>`interface-type=PKTDRV` |

**To use NetWare/IP over LWPE.COM and ODI Drivers**

1. If it is not already present, install the NetWare/IP 2.1 server module on a NetWare 4.1 server.
2. If it is not already present, install Novell LAN WorkPlace for DOS on your PC.
3. If it is not already present, install the Novell NetWare/IP 2.1 client on your PC. Confirm that the C:\WINDOWS\SYSTEM directory contains NWIPWIN.EXE and RES_SUPP.DLL.
4. After completing the installation, edit the AUTOEXEC.BAT, PCTCP.INI, and NET.CFG files, as shown in Table 5-10.
5. Configure `stacks=9,512` in your CONFIG.SYS file, as shown in Table 5-10.
6. Verify that IPCONFIG.DLL is in the C:\WINDOWS\SYSTEM directory.

Table 5-10 displays the system file entries that support the OnNet16 kernel and Novell NetWare interoperability over LWPE.COM and ODI drivers.

Table 5-10 NetWare/IP over LWPE.COM and ODI Drivers

| File | Entries |
|---|---|
| AUTOEXEC.BAT | `path=c:\pctcp;%path%`<br>`set pctcp=c:\pctcp\pctcp.ini`<br>`c:\nwclient\lsl.com`<br>`c:\nwclient\`*`cardname.com`*<br>`c:\pctcp\odipkt.com`<br>`c:\pctcp\lwpe`<br>`c:\nwclient\nwip`<br>`c:\nwclient\vlm`<br>`c:\pctcp\vxdinit` |
| CONFIG.SYS | `stacks=9, 512` |

RESOLV.CFG

```
domain abc.xyz.com
nameserver 128.127.50.123
nameserver 128.127.51.789
```

NET.CFG

```
link driver cardname.com
     int 3
     port 280
     mem D8000
     frame ETHERNET_802.3
     frame ETHERNET_802.2
     frame ETHERNET_II

NetWare ODS Requester ;Disables packet bursts. Add if
     PB BUFFERS 0     ;you are using an IBM Token Ring
                      ;network card
```

PCTCP.INI

```
[pctcp ifcust 0]
ip-address= 128.127.50.1
router=128.127.50.2
subnet-mask=255.255.255.0
frame-type=DIX-Ethernet
interface-type=PKTDRV

odi-pkts=8         ;provides optimal ODI performance

[pctcp kernel]
vlwpe=yes

[pctcp lwpe]
rcbs=40
sockets=32
loadhigh=yes
```

## 5.3 Installing the OnNet16 Kernel with a Packet Driver

Use the OnNet16 kernel with a packet driver to add TCP/IP capabilities to your PC

- On a LAN that does not also support another network environment.
- With a serial line connection (such as a modem).
- In an Apple LocalTalk network.
- In a Novell NetWare environment that uses Ethernet packet drivers.

**To install the OnNet16 kernel in a simple LAN**

1. In the Found Existing Driver dialog box, select the default existing network driver.

   **-or-**

   Choose Install a New Driver and select another driver in the Network Card dialog box.

2. Typically select the DIX Ethernet or the IEEE 802.5 Token Ring network (frame) type.

Table 5-11 displays the system file entries that support the OnNet16 kernel over a packet driver.

Table 5-11 Using a Packet Driver

| File | Entries |
|---|---|
| AUTOEXEC.BAT | `path=c:\pctcp;%path%`<br>`set pctcp=c:\pctcp\pctcp.ini`<br>`c:\pctcp\3c503 0x60 3 0x300`<br>`c:\pctcp\vxdinit.exe` |
| PCTCP.INI | `[pctcp ifcust 0]`<br>`ip-address= 128.127.50.1`<br>`router=128.127.50.2`<br>`subnet-mask=255.255.255.0`<br>`frame-type = DIX-Ethernet`<br>`interface-type = PKTDRV` |

### 5.3.1 Installing the OnNet16 Kernel with an Apple LocalTalk Packet Driver

Use the OnNet16 kernel with Apple LocalTalk and the LTDRV packet driver to add TCP/IP capabilities to your LocalTalk network.

The LocalTalk PC card must be installed with the IRQ level disabled (that is, in polled mode).

You must have an IP gateway node already on the AppleTalk network (GatorBox, or some other gateway box). The LocalTalk packet driver *does not work* without a gateway node. The IP address of the node is an optional argument you can specify when you load the driver.

**To use an AppleTalk packet driver**

When you are prompted by the installation program, choose to use the AppleTalk packet driver and the AppleTalk network type.

Table 5-12 displays the system file entries that support the OnNet16 kernel and Apple LocalTalk interoperability.

Table 5-12 Using an Apple LocalTalk Packet Driver

| File | Entries |
|---|---|
| AUTOEXEC.BAT | `path=c:\pctcp;%path%`<br>`set pctcp=c:\pctcp\pctcp.ini`<br>`c:\pctcp\localtlk 0x62`<br>`c:\pctcp\vxdinit.exe` |
| PCTCP.INI | `[pctcp ifcust 0]`<br>`ip-address= 128.127.50.1`<br>`router=128.127.50.2`<br>`subnet-mask=255.255.255.0`<br>`frame-type=Apple-Talk`<br>`interface-type=PKTDRV` |

### 5.3.2 Installing the OnNet16 Kernel with a Serial Line Packet Driver

Use the OnNet16 kernel with a serial line, ISDN line, or modem to add TCP/IP capabilities to your PC.

**To use a SLIP or PPP packet driver**

1. During a new Custom installation, a reinstallation, or an installation that is an upgrade from a previous release of OnNet16, from the Network Interface dialog box, choose Add.
2. In the Found Existing Driver dialog box, select Install a new driver.
3. In the Network Card dialog box, select Serial Port using SLIP or PPP.
4. If you are using a modem, select either Point-to-Point Protocol (PPP) or Serial Line Internet Protocol (SLIP), whichever protocol is supported by the remote site.

   If your are using an ISDN line, in the Serial Protocol dialog box, select Point-to-Point Protocol (PPP).

   If you are using a serial line, in the Serial Protocol dialog box, select Serial Line Internet Protocol (SLIP).

You can use SLIP or PPP only with Microsoft Windows 386 Enhanced mode.

Table 5-13 displays the system file entries that support the OnNet16 kernel and serial line interoperability.

Table 5-13 Using a SLIP or PPP Packet Driver

| File | Entries |
|---|---|
| AUTOEXEC.BAT | `path=c:\pctcp;%path%`<br>`set pctcp=c:\pctcp\pctcp.ini`<br>`c:\pctcp\vxdinit.exe` |
| PCTCP.INI | `[pctcp ifcust 0]`<br>`ip-address= 128.127.50.1`<br>`router=128.127.50.2`<br>`subnet-mask=255.255.255.0`<br>`frame-type= PPP`<br>`interface-type=PKTDRV`<br><br>`[pctcp kernel]`<br>`rtt-multiplier=2`<br>`slow-link-multiplier=2`<br>`do-slow-start=yes`<br><br>`[pctcp serial default]` |

SYSTEM.INI

```
[pctcp comscrpt default]
serial=default

[boot]
drivers=mmsystem.dll, wcomapi.dll

[386Enh]
device= wcomvxd.386
```

## 5.4 Installing the OnNet16 Kernel with Windows for Workgroups

Use the OnNet16 kernel with Windows for Workgroups and one of several drivers to add TCP/IP capabilities to your PC. Add the OnNet16 kernel NetBIOS to extend the use of Windows for Workgroups over a router.

Use the OnNet16 kernel with Windows for Workgroups and ODI drivers to provide TCP/IP capabilities on your network and to communicate with NetWare 386 servers (3.*x* and above).

You can use the OnNet16 kernel and Windows for Workgroups sharing either an NDIS driver or ODI drivers and using one of the following:

- NETBEUI, to run Windows for Workgroups applications only within a single LAN.
- NetBIOS, to run Windows for Workgroups applications over a router, when you do not need to support NETBEUI.
- Both NETBEUI and NetBIOS, if you need to interoperate with NETBEUI-compatible applications across routers.

**To use an NDIS driver**

1. Configure Windows for Workgroups to use the Real and Enhanced Mode (NDIS2/NDIS3) option.
2. When you are prompted by the installation program, select the existing network driver and one of the following network (frame) types: DIX Ethernet, IEEE 802.3 Ethernet, or IEEE 802.5 Token Ring.
3. Complete the installation as described in Section 3.1.21, Running the Windows for Workgroups Network Setup.

**To use NetBIOS with NETBEUI**

1. Edit your PROTOCOL.INI and PCTCP.INI files, as shown in Table 5-14.
2. To use a serial connection, edit the `SlowLanas=` parameter in the `[network]` section of the SYSTEM.INI file as shown in Table 5-14.
3. Verify your system files, as shown in Table 5-14.

Table 5-14 displays the system file entries that support the OnNet16 kernel and Windows for Workgroups interoperability with an NDIS driver.

Table 5-14 Sharing NDIS Drivers with Windows for Workgroups

| File | Entries |
|---|---|
| AUTOEXEC.BAT | `c:\windows\net start ; if using NDIS2 drivers, this` |

```
                        ; line must precede vxdinit
path=c:\pctcp;%path%
set pctcp=c:\pctcp\pctcp.ini
c:\pctcp\vxdinit
```

CONFIG.SYS

```
device= c:\windows\ifshlp.sys
```

SYSTEM.INI

```
[network drivers]
netcard= your_NDIS_driver_filename.dos
LoadRMDrivers=No

;To use NetBIOS without NETBEUI, comment out the
;netmisc= and transport= entries in the [386Enh]
;section.

[network]
SlowLanas=0,1,2 ;specify the LANA channel(s) used for
                ;NetBIOS. Regulates the flow of NetBIOS
;data to prevent the network from
;timing out over slow connections.
```

PROTOCOL.INI

```
[PKTDRV]
DriverName=ELNK3$
BINDINGS=MS$ELNK3
intvec=0X60
chainvec=0X65

[MS$ELNK3]

;To add NetBIOS, change lana0= to lanann, where n is the
;next available adapter number, in a way similar to the
;following:
[network.setup]
lana3=ms$elnk3,1,ms$netbeui
lana1=ms$elnk3,1,ms$ndishlp
lana2=ms$elnk3,1,ms$netbeui

[ms$netbeui]
```

PCTCP.INI

```
[pctcp ifcust 0]
ip-address= 128.127.50.1
router=128.127.50.2
subnet-mask=255.255.255.0
frame-type= DIX-Ethernet
interface-type= NDIS3

;To turn off NetBIOS
[pctcp vxdinit]
vnbep=no

;To use NetBIOS, add the following:
[pctcp vxdinit]
vnbep=yes

[pctcp netbios-vxd]
```

```
names=16
ncbs=32
sessions=16
cache-elements=256

[pctcp netbios]
namefile=c:\pctcp\namefile.txt
broadcastfile=c:\pctcp\broadcast.txt
```

In the AUTOEXEC.BAT file, there is no **netbind** command and the **net start** command precedes any entry that starts a TSR.

In the CONFIG.SYS file, note that there are no references to the windows device driver (DIS_PKT.GUP), to PROTMAN.DOS, or to NDIS card-specific drivers.

To review InterDrive modifications to the SYSTEM.INI file, see Section 5.5, Using the FTP Software Network Drivers with Other Network Operating Systems (Chaining).

In the PROTOCOL.INI file, set `BINDINGS=` to the label used in the card-specific section. Note that the value of `lanabase=` is `3`, corresponding to `lana3` in the `[network.setup]` section.

**To use ODI drivers**

1. When you are prompted by the installation program, select the existing network driver.
2. To use NetBIOS over a serial connection, edit the `SlowLanas=` parameter in the `[network]` section of the SYSTEM.INI file as shown in Table 5-14.
3. If you selected the OnNet16 kernel, choose one of the following network (frame) types: DIX Ethernet or IEEE 802.5 Token Ring.

Table 5-15 displays the system file entries that support the OnNet16 kernel and Windows for Workgroups interoperability with ODI drivers.

Table 5-15 Sharing ODI Drivers with Windows for Workgroups

| File | Entries |
|---|---|
| AUTOEXEC.BAT | `path=c:\pctcp;%path%`<br>`set pctcp=c:\pctcp\pctcp.ini`<br>`c:\windows\net start ; if using NDIS2 drivers, this`<br>`; line must precede vxdinit`<br>`c:\nwclient\lsl.com`<br>`c:\nwclient\cardname.com`<br>`c:\nwclient\ipxodi.com`<br>`c:\pctcp\odipkt.com`<br>`c:\nwclient\vlm.exe`<br>`c:\pctcp\vxdinit` |

```
c:\windows\odihlp.exe

;To add NetBIOS add +n to c:\pctcp\vxdinit:
c:\pctcp\vxdinit +n
```

CONFIG.SYS

```
device= c:\windows\ifshlp.sys
```

NET.CFG

```
link driver cardname.com
     int 3
     port 280
     mem D8000
     frame ETHERNET_802.3
     frame ETHERNET_802.2
     frame ETHERNET_II
```

SYSTEM.INI

```
[network drivers]
netcard=ODI_driver_filename.ext
transport=ndishlp.sys, *netbeui
LoadRMDrivers=No

:To use NetBIOS without NETBEUI, comment out the
;netmisc= and transport= entries in the [386Enh]
;section.

[network]
SlowLanas=0,1,2 ;specify the LANA channel(s) used for
                ;NetBIOS over a serial connection.
                ;Regulates the data rate to accomodate
                ;slower serial connections.
```

PCTCP.INI

```
[pctcp ifcust 0]
ip-address= 128.127.50.1
router=128.127.50.2
subnet-mask=255.255.255.0
frame-type= DIX-Ethernet
interface-type= PKTDRV

;To turn off NetBIOS, set the following:
[pctcp vxdinit]
vnbep=no

;To use NetBIOS, add the following:
[pctcp vxdinit]
vnbep=yes

[pctcp netbios-vxd]
names=16
vncbs=32
sessions=16
cache-elements=256

[pctcp netbios]
namefile=c:\pctcp\namefile.txt
broadcastfile=c:\pctcp\broadcast.txt
```

Note that, in the AUTOEXEC.BAT file, there is no **netbind** command and that the **net start** command precedes any entry that starts a TSR.

To review InterDrive modifications to the SYSTEM.INI file, see Section 5.5, Using the FTP Software Network Drivers with Other Network Operating Systems (Chaining).

## 5.5 Using the FTP Software Network Drivers with Other Network Operating Systems (Chaining)

The FTP Software Network Driver (PCTCPNET.DRV) allows you to access remote files and printing services through the Windows File Manager, the Control Panel, and Print Manager. If you use other network operating systems concurrently with the OnNet16 kernel, these systems must be connected to enable access to each system's services. The configuration of these network operating systems is referred to as "chaining" one network to or from another.

The installation program detects other network operating systems on your system and makes the appropriate changes to the `[boot]` section of your SYSTEM.INI file, as outlined in the following table. You can also use the Network Driver Availability dialog box during an OnNet16 Custom installation to configure the order in which you access network drivers.

Table 5-16 Network Driver Entries in the SYSTEM.INI File

| Network Operating System(s) | Entries |
|---|---|
| FTP Software Network Driver only in Windows 3.1 | `[boot]`<br>`network.drv=pctcpnet.drv` |
| FTP Software Network Driver and Windows for Workgroups | `[boot]`<br>`network.drv=wfwnet.drv`<br>`secondnet.drv=pctcpnet.drv` |
| FTP Software Network Driver and another network operating system in Windows 3.1 (such as Novell NetWare) | `[boot]`<br>`network.drv=pctcpnet.drv`<br>`pctcpchain=netware.drv` |
| FTP Software Network Driver and Windows for Workgroups with another network operating system (such as Novell NetWare) | `[boot]`<br>`network.drv=wfwnet.drv`<br>`secondnet.drv=pctcpnet.drv`<br>`pctcpchain=netware.drv`<br>`;Ensure that the [386Enh] section does not contain entries for WFWFTP.386 or VPCTCP.386. Remove (or comment out) these entries if they appear.` |

# Chapter 6
# Customizing a Centralized Installation and Configuration

OnNet16 offers a variety of installation options for the network administrator. Using the procedures defined in this chapter, and the Custom Install Manager program (with its graphical interface) you can customize the Setup program to meet the needs of your group or site.

Using the procedures in this chapter, you can

- Customize the Setup program with Custom Install Manager.
- Set up a network installation.
- Specify a central network configuration method.
- Manually copy OnNet16 files.

## 6.1 Setting Up Network Installations

This section defines procedures for configuring network installations. You can choose to have your users install OnNet16 from a network server to simplify and standardize the installation process.

Users who are at sites that are licensed for multiple installations, and who have a network operating system installed on their system, can perform an installation of this type.

To prepare for a network installation, the network administrator copies all the files from the distribution disks or CD-ROM to a file server, then instructs each remote user to access the remote file server and to run the Setup program.

**To set up an installation to be run from a network server**

1. On the file server, create a directory and copy the OnNet16 distribution disks into the directory.

   Create a flat directory structure and copy the files from the CD-ROM.

   **-or-**

   Create a flat directory structure; copy all files from the disks into the same directory on the server; delete the existing FILES.INF file; and rename NETFILES.INF to FILES.INF.

   **-or-**

   Create a directory structure such that you copy the files from each disk to a separate subdirectory. In this case, do not rename NETFILES.INF.

2. Instruct each remote user to access the remote file server and to run the Setup program.

**To run TN3270 from an application server**

1. Install TN3270 on the application server.
2. In the `[System Info]` section of the TN3270.INI file, set the shared version to `1`, as follows:

```
[System Info]
Shared Version=1
```

3. Copy the TN3270.INI and TN3270.CFG files from the application server to the client's WINDOWS directory. Edit the TN3270.INI file to customize directories and paths as appropriate for your environment.

## 6.2 Customizing the Setup Program for Your Site with Custom Install Manager

Using the Custom Install Manager program, you can create and manage customized installation sets for departments, groups, and individuals.

If you want to use customized installation sets you must perform a network installation as described in Section 6.1, Setting Up Network Installations. However, you can have a shared OnNet16 installation without customization.

### 6.2.1 Before You Start Using Custom Install Manager

Custom Install Manager is located on the OnNet16 CD-ROM. If you have OnNet16 on disks, you can get Custom Install Manager by sending in the order card (which is included in the distribution package).

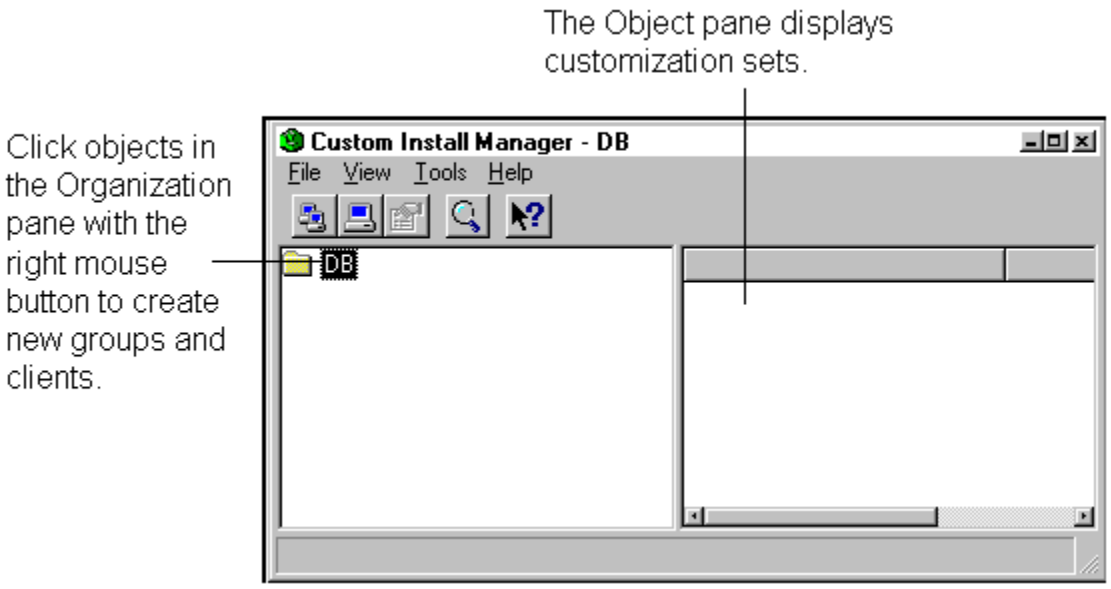
Before you start using the Custom Install Manager, you should be sure that

- You have access to a computer that is running Windows 95 or Windows NT 4.0. Custom Install Manager does not run over a Windows 3.*x* operating system.
- You have access to a shared drive.
- You are familiar with the following:
  - Setup, the installation program
  - The components and subcomponents of OnNet16
  - The FTP Software TCP/IP protocol stack

### 6.2.2 Getting Started with Custom Install Manager

**To start Custom Install Manager**

1. Create a network installation as described in Section 6.1, Setting Up Network Installations.
2. Read the instructions in INSTALL.TXT, located in the CIM folder on the OnNet16 CD-ROM, or on Disk 1 of the Custom Install Manager distribution on disks.
3. On a computer that is running Windows 95 or Windows NT 4.0, run the separate Setup program for Custom Install Manager.
4. On the Start menu, point to FTP Custom Install Manager and then click FTP Custom Install Manager.




5. Read the overview information in the Custom Install Manager online Help.

**Tips for Using Custom Install Manager**

- You can create customized installations labeled by computer name or Microsoft Network group name. Then the Setup program runs the correct installation script for these clients or groups.
- You can specify TCP/IP configuration information for each group or client.

## 6.3 Using DHCP or Bootp for Central Configuration

If you intend to set up a server from which your users can obtain network and OnNet16 configuration information, you must set up the server and use Custom Install Manager to specify a central configuration method in the `[options]` section of SETUP.INF.

**To enable central configuration**

1. Configure a DHCP or Bootp server to provide configuration information.
2. Use Custom Install Manager to specify the use of dynamic IP configuration by choosing Automatic (DHCP) configuration. For detailed instructions, see the online Help for Custom Install Manager.

For DHCP and Bootp client configuration procedures, see Part II, “Managing OnNet16 Kernel and Network Services.”

## 6.4 Manually Copying OnNet16 Files

If you need to copy additional files from the OnNet16 distribution disks, use the following procedure.

**Note:** FTP Software recommends that you use the OnNet16 Setup program to copy files.

**To copy a file from the installation disks**

1. Place Disk 1 of the installation disks in your disk drive.
2. From the Program Manager File menu, choose Run. Type the drive letter, a colon, a backslash, and `setup /z`. For example,

   **a:\setup /z a:***filename* **.ex_ c:\pctcp***filename* **.exe**

3. If prompted to insert the installation disk containing the specified file, place that disk in your disk drive.

# Chapter 7
# Introduction to OnNet16 Serial Services

The OnNet16 Dialer program provides serial services that allow a PC to connect to the Internet or to a corporate service. Dialer supports both PPP and SLIP network interface protocols; it can establish connections automatically (using scripts that the user creates or that Dialer creates with the Connection wizard). Dialer connections can use analog telephone lines and modems, wireless devices, ISDN digital telephone lines and devices, or dedicated serial lines.

## 7.1 Managing SLIP and PPP Connections

SLIP connections are made with an analog telephone line and a modem, or with a wireless serial device. PPP connections are made with an analog telephone line and a modem, or with a digital ISDN line and an ISDN device.

You can use the Dialer application to easily switch between a LAN and SLIP or PPP connections. The Dialer application includes an “Automatic Connection” feature (also known as “Dial on Demand”) that enables applications to connect to the network as needed. When you start an application that requires a network connection, that connection is automatically made. You can configure the Dialer application to end a connection after a period of inactivity.

For example, if you are working in your LAN, you can use the Dialer application to dial out of the office with a modem. When you start the Dialer application, it will switch you automatically from the LAN interface to the SLIP or PPP interface. When you terminate the connection, the Dialer will switch you back to the LAN interface. (However, if the Dialer terminates a connection after a period of inactivity, it does not automatically switch you back to the LAN; you remain in serial mode until you click Release Device.)

### 7.1.1 ISDN Support

The OnNet16 kernel supports high-speed ISDN telephone line connections using a WinISDN or CAPI v2.0 interface. You can select a WinISDN or CAPI connection in the Dialer application; however, to use this feature you must have the necessary hardware and you must have ISDN service.

The Dialer supports multilink PPP over ISDN through the CAPI 2.0 DLL or CAPI 2.0 VxD interface; it supports only single link PPP with the WinISDN interface.

### 7.1.2 Wireless Support

The Dialer supports connections to remote sites over cellular telephone channels by way of a wireless modem. The Dialer supports both Cellular Digital Packet Data (CDPD) and analog cellular (Circuit Switched Cellular) devices. For more information about using wireless modems, see the Dialer online Help.

### 7.1.3 Using Scripts

The Dialer wizard interface creates a script file the first time you establish a new connection to a network or Internet service provider. This file generally initializes the modem, dials the remote host, and logs you in with a SLIP or PPP account name and password. When you end the connection, the script file generally ends the connection, then resets your modem so that it is ready to receive the next dial-up sequence.

Usually, you do not need to edit Dialer script files because they are automatically created for you by the Dialer connection wizard. However, if you need to, you can create and edit your own Dialer script files.

In the case of CDPD, there is no need to create a script when a new CDPD connection is established because their scripts have already been created.

#### Customizing Dialer Script Files

Your FTP Software distribution includes the sample script file TOUGH.SCR that you can customize and use to connect to and disconnect from a remote host automatically. (The installation program copies this file to the C:\PCTCP directory by default.) The TOUGH.SCR file contains extensive annotations to explain how it works.

In a Windows environment you can edit TOUGH.SCR through the Dialer application in Advanced Mode. In a DOS environment edit this file with any ASCII editor. Save the edited file under a new name with the .SCR extension (for example, MYDIALUP.SCR).

**Note:** Not all .SCR files in your OnNet16 software are sample files. Your distribution also includes some .SCR files that are used with applications. These scripts can serve as useful models for you, but you should not alter their contents.

Working in a Windows environment, you select your script from the pull-down list in the Advanced Mode window of the Dialer. See the Dialer application online Help for details.

To learn more about scripts and the SLANG™ language in which they are written, visit the FTP Software home page on the World Wide Web at http://www.ftp.com/. Look under Technical Support for the SLANG documentation.

# Chapter 8
# Configuring and Tuning the Kernel

When you install OnNet16, the setup program automatically configures the FTP Software kernel with default settings that are appropriate for your operating environment. In general, the kernel performs well using the configuration option defaults. If you find that your network is not performing adequately, you can change the settings of the kernel configuration parameters to improve performance. This chapter describes how to change the settings of the OnNet16 version of the FTP Software kernel.

Use the Configure application to configure your kernel dynamically.

Defaults for most kernel options are specified in the `[pctcp kernel]` and `[pctcp` *`interface n`*`]` sections of the PCTCP.INI file.

For detailed information about PCTCP.INI configuration file sections and the specific parameters that apply to each section, visit the FTP Software home page on the World Wide Web at http://www.ftp.com/. Look under Technical Support for the Configuration Parameters Reference documentation.

This chapter shows how to

- Tune OnNet16 kernel performance.
- Configure IP security.
- Resolve configuration file problems.

## 8.1 Configuring OnNet16 Components

The OnNet16 kernel supports TCP/IP applications in a Windows operating environment. Because the OnNet16 kernel relies on Windows memory management, it operates with essentially unlimited memory resources. It provides up to 128 active connections, and can be configured to use very little conventional memory.

Windows is instructed to load the OnNet16 kernel by the **vxdinit** command, which is normally placed in your AUTOEXEC.BAT file during software installation. This command loads a small TSR that instructs Windows to load the OnNet16 kernel and optionally other OnNet16 applications (such as InterDrive, DHCP, NetBIOS, and LWPE). The VXDINIT TSR loads high by default.

Use the Advanced setting of the Configure application to review your kernel configuration. Examine the present values of PC/TCP Kernel. You can also examine the present values of NetBIOS and InterDrive to find out if they are being automatically enabled.

**Note:** Any **vxdinit** command line options (including those specified in your AUTOEXEC.BAT file) override settings displayed in the Configure application.

For more information about verifying the OnNet16 kernel configuration, see Part I, “Installing and Configuring OnNet16.”

## 8.2 Tuning Kernel Performance

Tuning the kernel can resolve performance problems such as slow transfer rates, inconsistent application behavior, or difficulty opening additional connections. You can optimize performance by adjusting

- The TCP window size.
- The number of packet buffers.
- The number of TCP and UDP connections.

### 8.2.1 Adjusting the TCP Window Size

The configured TCP window size can affect network performance. The TCP window size determines the maximum amount of TCP data (in bytes) that a remote host can send to your system before your system passes the data to an application. In general, the configured default is adequate for most environments. However, if you are experiencing performance problems, your PC might be receiving (and responding to) duplicate packets.

**To determine if your system is receiving duplicate packets**

Run the Statistics application and choose the Statistics text view.

Look for a nonzero value for “duplicates” in the TCP section of the display. If you are using **inet stats**, look for a nonzero value in the `duplicate pkts` display line. If the number of duplicate packets is large, you can sometimes improve performance by adjusting the TCP window size.

You can specify a default TCP window size with the Configure application. Use the Advanced setting, select PC/TCP Kernel, choose Modify, then modify Fine Tuning.

**To determine the best TCP window size**

1. Use the following table to determine the amount of data in a TCP packet for your network interface.

   Refer to the following information and note the Maximum Segment Size (MSS) of your network interface. If your network interface does not appear in the table, refer to the documentation for your network card.

| For this type of network | The Maximum Segment Size (MSS) is |
|---|---|
| Ethernet or StarLAN | 1460 |
| Token Ring | 1962 |
| ProNET-10 | 2000 |
| SLIP | 962 |
| PPP | 1460 |

2. Start any TCP application that will transfer data and display the transmission speed. For example, start the FTP application and, from the Settings menu, choose Verbose.
3. Connect to a remote host and select a large file (1MB or larger) to transfer.
4. Copy the selected file, giving the name of the destination file as **`nul`** This will ensure that no file is actually created on the local host and that the peculiarities of local disk speeds will not affect the results of your test.

5. When the transfer is complete, the application displays the transmission speed in bytes per second.

6. Repeat steps 3 through 5 four times and compute an average of the reported transmission speeds.

7. Use the Configure application to dynamically configure the kernel with a new TCP window size.

   Use a value that is a multiple of the MSS. For example, if you started with a window size of 1460 on your Ethernet network, use the value 2920.

8. Repeat steps 3 through 7 several more times with values three and four times the original MSS to determine the TCP window size that maximizes the transmission speed.

9. When you have determined the optimum TCP window size, reconfigure the kernel with the new window size setting.

### 8.2.2 Adjusting the Number of Packet Buffers

The number of configured packet buffers can affect network performance. In general, the defaults for the maximum number of packets provide sufficient buffering capacity for most networking environments. However, if you are experiencing network problems, the kernel might require additional buffers for optimal performance.

**To determine the best number of packet buffers**

1. Run a TCP application such as FTP.
2. Start the Statistics application to display kernel statistics.
3. Review the statistical display and note the `minimum free` values for both large and small packet buffers. If either of these numbers reaches 0, increase the number of packet buffers.
4. Review the `out of buffers` number in the statistical display to verify that it is less than 1% of the `packets received`. If it is greater than 1%, increase the number of large and small packet buffers by one or two.

   **Note:** The software uses available small packet buffers first, before it uses large packet buffers. If you are experiencing network problems, try increasing the number of small packet buffers first.
5. Use the Configure application to reconfigure the kernel with additional small, large, or huge packet buffers.
6. Unload and reload the kernel so that the configuration options take effect.
7. Repeat steps 1 through 6 to determine the best settings for your system.
8. When you have determined the optimum number of packet buffers, reconfigure the kernel with the new setting.

### 8.2.3 Adjusting the Number of TCP and UDP Connections

If you are running several simultaneous applications over the kernel, you might exhaust the number of available TCP or UDP connections. Adding more connections can improve network performance.

**Note:** Performance can degrade if you specify too many TCP connections and too few large packet buffers. Make sure that you configure enough large packet buffers to accommodate the number of TCP connections that you expect to use. The default number of large packet buffers is 40, which is usually large enough to accommodate most operating environments.

Because Windows manages OnNet16 kernel memory usage, the default number of TCP and UDP connections for the OnNet16 kernel is usually large enough to accommodate most operating environments. However, if you run out of TCP or UDP connections, you can

- Close one (or more) of the running applications to release connections for use by other applications.
- Reconfigure the kernel to increase the number of TCP or UDP connections, and try the application again.

**To increase the maximum number of TCP or UDP connections**

1. Start the Statistics application to display kernel statistics.
2. Use the Configure application to reconfigure the kernel with additional TCP or UDP connections. Combined, there can be no more than 64.

   Add one connection at a time, and retry.
3. Unload and reload the kernel for the new configuration settings to take effect.

## 8.3 Configuring IP Security

OnNet16 supports Type of Service, Precedence, and Security options in the IP header. The TCP layer interprets the Type of Service and Precedence options.

You can specify the type of IP security using the Configure application. Use the Advanced setting and select PC/TCP kernel. Then select IP Security. Disabling the IP security configuration option saves approximately 1.4K of memory.

For a discussion of FTP security features, see Part III, “Using Network Security.”

For background information about TCP/IP security standards, consult RFCs 791, 1108, 1195, and 1349.

## 8.4 Resolving Configuration File Problems

If the configuration changed—for example, if you add new drivers to the system or you tune the system to improve its performance—you might enter an incorrect value when modifying the original configuration.

If you suspect that an error in the new configuration file is the cause of the current problem, try returning the configuration parameters to their previous settings. If returning to the previous settings eliminates the problem, you must find and change the incorrect entry in the new configuration file.

Before making any additional changes, review and analyze the hardware and software configuration to clarify what entries should be specified in the configuration file.

**To examine configuration file parameters**

Use the Configure application to examine and change configuration file parameters.

The following tables show standard IP settings, sections, and file parameters.

To use the Configure application to change the settings in the following table, choose the Advanced setting. Then select Internet Addresses from the configuration list, choose Modify, select Your System, and choose Modify.

| IP Setting | PCTCP.INI Section | PCTCP.INI Parameter |
|---|---|---|
| The PC's IP Address | `[pctcp interface n]` | `ip-address=` |
| The IP router | `[pctcp interface n]` | `router=` |
| The subnet mask | `[pctcp interface n]` | `subnet-mask=` |

To use the Configure application to change the settings in the following table, choose the Advanced setting. Then select Name Server Configuration from the configuration list and choose Modify.

| IP Setting | PCTCP.INI Section | PCTCP.INI Parameter |
|---|---|---|
| IP hostname | `[pctcp general]` | `host-name=` |
| The DNS server(s) | `[pctcp addresses]` | `domain-name-server=` |
| The NIS server(s) | `[pctcp addresses]` | `nis-name-server=` |
| The domain | `[pctcp general]` | `domain=` |

**Note:** NIS is Sun's Network Information System. It is used in Sun environments to distribute system information across a network. A system configured to support NIS cannot also support DNS.

To use the Configure application to change the settings in the following table, choose the Basic setting. Then select the appropriate kernel from the configuration list, and choose Modify.

| IP Setting | PCTCP.INI Section | PCTCP.INI Parameter |
|---|---|---|
| The local host table file | `[pctcp kernel]` | `host-table=` |

**To change incorrect configuration file entries**

Use the Configure application to change incorrect configuration file entries.

# Chapter 9
# Configuring NetBIOS

NetBIOS is a session layer interface for local area networks (LANs) that lets you run applications over a variety of network protocols. It provides wide area connectivity for network operating systems (NOS), such as Microsoft LAN Manager and Windows for Workgroups. Note that unless you plan to run a NOS that uses the NetBIOS transport protocol, you do not need to load NetBIOS. A NetBIOS host can communicate only with other RFC-compliant hosts.

The FTP Software NetBIOS uses TCP/IP and is fully compliant with RFCs 1001 and 1002, and with the *IBM NetBIOS Application Development Guide*.

When you run NetBIOS with a compatible network operating system, your PC can

- Share resources with computers not only within a LAN but also across a TCP/IP network.
- Interact with non-PC hosts that implement TCP/IP NetBIOS.

## 9.1 Before You Start Using NetBIOS

Perform the following tasks before you use NetBIOS:

1. Install, configure, and load your NOS.
2. Install and configure OnNet16. (The distribution includes NetBIOS.)

   **During installation you can optionally configure Windows Internet Naming Service (WINS).**

   **Note:** To support your network environment, you might need to configure your kernel with additional TCP connections. See Section 9.5, Tuning the Kernel for NetBIOS.
3. Verify that any other NetBIOS product that you are using is RFC-compliant and is compatible with NetBIOS for OnNet16 (such as Windows for Workgroups).

## 9.2 Configuring NetBIOS

The installation procedure provides a default NetBIOS configuration. In general, this initial configuration is adequate, and you should not need to override the default settings. If you do need to change NetBIOS settings, use the Configure application.

You must restart Windows for new configuration option settings to take effect.

## 9.3 Starting NetBIOS

This section provides detailed instructions for starting the NetBIOS VxD.

**To start the NetBIOS VxD**

1. In the Configure application, select NetBIOS and examine the present value in the Load NetBIOS VxD box. If the value is set to Yes, the NetBIOS VxD is running and you can skip the rest of this procedure. If the value is set to No, set it to Yes and exit Windows.
2. Enter **`vxdinet -u`** to unload the VxD kernel.
3. Enter **`vxdinit`** to restart the kernel with the new configuration.
4. Restart Windows.

   The kernel loads the NetBIOS VxD, and any other specified VxDs.

## 9.4 Setting NetBIOS Configuration Options

To set NetBIOS configuration options, use the Configure application and choose the Advanced setting. Then select NetBIOS and choose Modify. NetBIOS configuration option defaults are stored in the `[pctcp netbios-vxd]` section of your OnNet16 configuration file (C:\PCTCP\PCTCP.INI, by default).

You can use NetBIOS configuration options to

- Use multiple NetBIOS adapters.
- Logically partition your NetBIOS subnet.
- Communicate across routers.
- Identify hosts to automatically receive broadcast messages.
- Specify NetBIOS names.
- Use Domain Name System (DNS) to resolve NetBIOS names.
- Enable Windows Internet Name Server (WINS) support.

The Configure application online Help provides detailed information about many of these and other configuration options.

### 9.4.1 Using Multiple NetBIOS Drivers

Some network operating systems allow their NetBIOS drivers to coexist with the NetBIOS you receive with OnNet16. If you are using multiple NetBIOS stacks, you must specify which adapter to use for NetBIOS. In the Configure application, choose the Advanced setting, select NetBIOS, choose Modify, choose Additional, select Driver adapter number (LANA), and set the value.

**Note:** Some NetBIOS-based network operating system installation programs insert information about their own NetBIOS drivers into the CONFIG.SYS or AUTOEXEC.BAT file. Generally, you should remove any reference to a third-party NetBIOS network driver to avoid conflicts with NetBIOS for OnNet16.

### 9.4.2 Partitioning Your Network with Scope

Scope is a naming scheme that NetBIOS uses to form a session-layer subnetwork on the LAN. This subnetwork (which differs from an IP subnet) divides hosts into discrete logical groups and limits those that receive broadcasts.

By default, NetBIOS for OnNet16 names have no scope (commonly called a “null” scope). Many NetBIOS vendors (such as 3Com, Bridge, and Excelan) also use a null scope because they design applications for single groups of users on small LANs.

If your network uses scopes (many do not), your PC can communicate only with other NetBIOS hosts that share your scope. For example, if your PC is assigned to the kafka scope and you send a print request to a print server within the yeats scope, you receive the following message:

```
Network name not found.
```

To specify a name scope with the Configure application, choose the Advanced setting, select NetBIOS, choose Modify, and type a name in the Scope Value box.

To specify the null scope, leave the Scope Value box empty.

When you attempt to contact another NetBIOS host, NetBIOS appends your scope (with a leading dot (.)) to your local NetBIOS name (as defined in the `computername=` entry in your LAN Manager LANMAN.INI file). For example, if you set `computername= chris` in your LANMAN.INI file, and typed `kafka` in the Scope Value box, your host would identify itself as chris.kafka.

### 9.4.3 Communicating Across Routers

The OnNet16 kernel supports the following end-node implementations:

- B-node and P-node implementations (defined in RFCs 1001 and 1002)
- H-node implementations (a Microsoft-specific implementation)

**Note:** When using NetBIOS over a serial line, you must configure NetBIOS as a P-node or H-node implementation.

Generally, NetBIOS end-node implementations do not broadcast across IP routers. Instead, NetBIOS provides the following mechanisms to transmit broadcasts across routers:

| | |
|---|---|
| Broadcast file | See Section 9.4.4, Creating and Using a Broadcast File. |
| Name file | See Section 9.4.5, Creating and Using a Name File. |
| Domain scope | See Section 9.4.6, Using Domain Scope. |

**To configure NetBIOS B-node implementations to communicate across routers**

1. Create a broadcast file that identifies the remote hosts to receive NetBIOS messages.

   For example, the following broadcast file (C:\PCTCP\NB-BCAST) identifies two remote hosts:

   ```
   128.127.50.11
   128.127.50.12
   ```

   For detailed information about how to create and use a NetBIOS broadcast file, see Section 9.4.4, Creating and Using a Broadcast File.

2. Create a NetBIOS name file to associate host Internet addresses with a name.

   For example, the following NetBIOS name file (C:\PCTCP\NB-NAMES) associates names with the hosts identified in the broadcast file:

   ```
   SOMEHOST 128.127.50.11
   ANOTHER1 128.127.50.12
   ```

   For detailed information about how to create and use a NetBIOS name file, see Section 9.4.5, Creating and Using a Name File.

3. In the Configure application,

   – Select NetBIOS and type the names of your broadcast and name files in the appropriate value boxes.

   – View the Internet Addresses, Your System dialog box, and confirm that a router is configured.

   – View the Name Server Configuration dialog box and confirm that the DNS servers are

configured.

To configure NetBIOS P-node or H-node implementations to communicate across routers

1. Create a file containing the IP address of a NetBIOS name server (NBNS) in your network.

   For example, this file (C:\PCTCP\NBNS-SRV) identifies an NBNS server:

   ```
   128.127.50.15
   ```

   **Note:** A Windows NT WINS server can be used as an NBNS server. For more information, see Section 9.4.7, Enabling WINS Support.

2. Use a text editor to edit the PCTCP.INI file for each NetBIOS client. In the `[pctcp netbios-vxd]` section, add two lines to identify the type of end-node and the location of the file containing the name of the NBNS server. The following example illustrates P-nodes using the server identified by the file C:\PCTCP\NBNS-SRV:

   ```
   [pctcp netbios-vxd]
   .
       .
       .
       Node-type=P
       NBNS-servers-file=c:\pctcp\nbns-srv
   ```

**To use multicasting capabilities with B-nodes and H-nodes**

If your router(s) support multicasting, you can support this capability for B-nodes and H-nodes by adding a line to the `[pctcp netbios-vxd]` section of your PCTCP.INI file that supplies the multicast IP address. This is an address in the range 224.0.0.3 - 239.0.0.0. The NetBIOS default multicast address is 225.1.1.1. The following example adds a line to the PCTCP.INI file that specifies the default multicast address:

```
[pctcp netbios-vxd]
.
    .
    .
    multicast-address=225.1.1.1
```

Also ensure that

- Multicasting is enabled in the kernel. To verify this with the Configure application in Advanced mode, choose PC/TCP Kernel, select Basic, choose Additional, select Multicast and verify that the present value is set to Yes.
- All NetBIOS nodes in the same scope are using the same multicast address.

### 9.4.4 Creating and Using a Broadcast File

A "broadcast file" is an ASCII file containing a list of Internet hostnames or addresses to contact whenever you send information to the network. The broadcast file is read into memory when you load NetBIOS. To change the list of hosts in the broadcast file, you must edit the file and then restart Windows or unload and reload NetBIOS.

Note that NetBIOS does not recognize identical entries in the broadcast file as duplicates.

**Note:** Use this file carefully. NetBIOS sends a message to all local hosts on your LAN and to all hosts in the broadcast file whenever it sends a broadcast message or attempts to resolve a name. This can cause heavy network traffic.

The following is a sample NetBIOS broadcast file:

```
128.127.55.103
lanmanserver1
lanmanserver2
128.127.55.120
alpha
```

Specify the pathname or filename of the broadcast file with the Configure application. Choose NetBIOS and type a filename or pathname (for example, `C:\bfiles\miller`) in the Broadcast file name Value box.

If you specify only a filename, NetBIOS searches the directory where your PCTCP.INI file resides.

**To create a broadcast file**

1. Make a list of the names or IP addresses of the remote hosts that you want to receive NetBIOS packets.

2. Use a text editor to create a new file.

3. Create a name file entry for each name that you want to add.

   Each broadcast file entry consists of one line that identifies the hostname or IP address. The maximum number of entries for a broadcast file is 128.

   Do not enter a local IP address (such as your LAN Manager server's address) in the NetBIOS broadcast file.

   For example, the following broadcast file (C:\PCTCP\MILLER) contains two entries. The first host is specified by IP address, and the other is specified by a fully qualified Internet hostname.

   ```
   128.127.50.1
   anais.xyz.com
   ```

4. Verify that the present value of the Broadcast file name Value box in the Configure application

specifies the appropriate filename or pathname.

If you specify the MILLER broadcast file, NetBIOS sends all broadcasts, name resolution queries, and registration packets to the address 128.127.50.1 and the host anais.xyz.com.

### 9.4.5 Creating and Using a Name File

A “name file” is an ASCII file that associates a list of NetBIOS names with their corresponding fully qualified hostnames or IP addresses. A name file does not consume network resources like a broadcast file does, because NetBIOS sends information to only one remote host at a time. The name file is read into memory when you load NetBIOS.

NetBIOS displays the number of name file entries when you load it. Note that it does not recognize identical entries as duplicates.

The following is a sample NetBIOS name file:

```
LMN               128.127.55.103
LAN1ROOT          lanmanserver1
LAN2ROOT          lanmanserver1
LAN1PUBLIC        lanmanserver2
LAN2PUBLIC        lanmanserver2
PQR               128.127.55.120
STUV              128.127.55.120
```

Use the Configure application to specify the pathname or filename of the name file. Choose NetBIOS and type a filename or pathname (for example, `C:\nfiles\python`) in the Name file Value box.

If you specify only a filename, NetBIOS searches the directory where your PCTCP.INI file resides.

**To create a name file**

1. Make a list of the hostnames or IP addresses of the remote hosts that you want to add to a name file, together with the NetBIOS names that you want to assign to each.

   A NetBIOS name is usually a synonym that you can use in place of a fully qualified hostname or Internet address.

2. Use a text editor to create a new file.

3. Create a name file entry for each NetBIOS name that you want to add.

   For each entry, enter the NetBIOS name, press one or more spaces or tabs, then enter the IP address (or fully qualified hostname). Press Enter after each full entry.

   To insert hexadecimal values in a NetBIOS name, use a text editor to insert the equivalent quoted control character. For example, to add the hexadecimal value 0x01 to a name, use your text editor to insert Ctrl+A.

   **Note:** NetBIOS names in the name file are case sensitive. If you are using this name file with LAN Manager, all names in the name file must be uppercase.

   The following name file, PYTHON, contains name file entries for two different hosts:

```
idle 128.127.55.100
cleese monty.xyz.com
```

If you are using this name file with LAN Manager, the hostnames must be uppercase:

```
IDLE 128.127.5.100
CLEESE monty.xyz.com
```

NetBIOS names that contain spaces should be enclosed in double (or single) quotation marks:

```
"MY HOST" 128.127.55.100
```

4. Verify that the present value of the Name file Value box in the Configure application specifies the appropriate filename or pathname.

**To use a NetBIOS name**

Enter your network command (such as **net view** or **net print**), specifying the NetBIOS name associated with the remote hostname.

The following example shows how a LANtastic network user (after loading LANtastic) views the available resources on a remote host. This command connects the user to the remote host MONTY.XYZ.COM by using the NetBIOS name `cleese`, as defined in the PYTHON name file:

```
net view \\cleese
```

### 9.4.6 Using Domain Scope

If NetBIOS cannot resolve a hostname on the local network or through the broadcast file or name file mechanisms, NetBIOS attempts to resolve the hostname by combining an encoded NetBIOS name with the specified domain scope string.

Using a domain scope is most practical if you use the Domain Name System (DNS) to resolve hostnames. You can store and update all the NetBIOS names for your network in one host table.

To specify a domain scope with the Configure application, select the Advanced setting, select NetBIOS from the configuration list, and choose the Modify button. In the NetBIOS window, choose the Additional button. In the NetBIOS Additional window, select Domain scope, and type a scope (for example, `.xyz.com`) in the Value box.

To specify a null domain scope, leave the Value box empty.

NetBIOS uses an extension to the DNS to resolve NetBIOS names. The following is an overview of how NetBIOS resolves names:

1. NetBIOS searches the local NetBIOS name cache.

   Initially, when you load NetBIOS, this cache is empty. When a NetBIOS name is resolved, NetBIOS adds the name to the local name cache.

2. NetBIOS queries the local network for the NetBIOS name with the scope suffix.

3. NetBIOS searches for a match for the NetBIOS name in the name file. The name file associates a NetBIOS name with a fully qualified Internet name.

4. NetBIOS queries the DNS for the encoded NetBIOS name and then the unencoded NetBIOS name (if the domain scope suffix was specified).

NetBIOS DNS queries add the domain scope suffix to the NetBIOS name. These queries do *not* use the `scope=` suffix.

Domain name servers are identified in the Name Server Configuration window of the Configure application.

If you do not have a domain name server configured, NetBIOS does *not* query the DNS server to resolve NetBIOS names.

The following example illustrates how to configure NetBIOS to communicate with the hosts, SOMEHOST and ANOTHER1 (by name) across routers.

This is a sample C:\PCTCP\NB-BCAST file:

```
128.127.50.10
128.127.50.11
```

This is a sample C:\PCTCP\NB-NAMES file:

```
SOMEHOST 128.127.50.10
ANOTHER1 128.127.50.11
```

**To use NetBIOS domain scope**

Enter your network command (such as **net print** or **net use**), specifying the NetBIOS name associated with the remote hostname.

In the following example, a LANtastic network user has defined the domain scope as .xyz.com. The following command sequence sends the report1 file to printer1, which is attached to the network print server fast.xyz.com:

```
net use lpt2 \\fast##printer1
net print report1 lpt2:
```

NetBIOS appends the domain scope .xyz.com to fast and the file is printed on the remote printer using the server fast.xyz.com. (Note that the syntax of the print command varies with each network operating system.)

### 9.4.7 Enabling WINS Support

To take advantage of NIS (Network Information Service) and WINS (Windows Internet Name Service) hostname resolution, at least one Windows NT advanced server with WINS support must be installed on the network.

To enable WINS support, add the following lines to the [`pctcp netbios-vxd`] section of the PCTCP.INI file:

```
Node-type=H
NBNS-servers-file= pathname
```

where *`pathname`* is the name of a file containing the IP addresses of up to four WINS servers. You must specify this name using a complete pathname.

When NetBIOS uses WINS to resolve a name, NetBIOS contacts WINS servers in the order in which they are listed in the `NBNS-servers-file=` parameter.

## 9.5 Tuning the Kernel for NetBIOS

Out of the total number of TCP connections configured for the OnNet16 kernel, NetBIOS must allot the following:

- One TCP connection for NetBIOS to monitor incoming session requests from remote hosts.
- One TCP connection for each *active session* with another host running NetBIOS. For example, a remote host accessing a service on your PC requires one of your TCP connections.

If you are using the PC as a file server or print server, you can rapidly consume the available TCP connections. If the following message appears when you load NetBIOS, you need to reconfigure the kernel with additional TCP connections:

```
Warning: The number of NetBIOS sessions was reduced to n.
If you want to have more sessions, configure more connections at kernel startup.
```

For a detailed description of the procedure for increasing the number of TCP connections, see Section 8.2.3, Adjusting the Number of TCP and UDP Connections.

## 9.6 Programming with NetBIOS

The following information assumes that you are a NetBIOS programmer. Read it if you are writing an application program to run over NetBIOS for OnNet16 and need to know details about the OnNet16 NetBIOS implementation.

| To do this | You need to know that |
|---|---|
| Use asynchronous I/O | TCP/IP NetBIOS allows both synchronous (blocking) and asynchronous (nonblocking) I/O modes. It indicates completion by two methods: by upcalling the post routines or by performing completion code verifications in the Network Control Block (NCB). |
| Interpret the INTERFACE_BUSY error | The kernel and NetBIOS are not fully re-entrant. The kernel cannot respond to a request while it is servicing another request. If NetBIOS receives a request while another request is in process, it rejects the NCB and returns the INTERFACE_BUSY error. If your application receives this error on a WAIT NCB, the application must try the command later. |
| Interpret nonstandard return values | Some NetBIOS applications provide utilities that let you list the status of a NetBIOS session. NetBIOS reports ADAPTER_STATUS and SESSION_STATUS values on pending sessions, packets sent, packets received, and retransmissions. (The equivalent fields displayed on the screen for these status conditions vary with the utility.) |

The following table describes the nonstandard return values in NetBIOS for OnNet16:

| NetBIOS Condition | Description |
|---|---|
| Pending sessions | The number of open and listening sessions. |
| Packets sent | The number of successful session and datagram send commands. |
| Packets received | The number of successful session and datagram packets received. |
| Retransmissions | The number of name query retries (unsuccessful queries). |

**Note:** In the Sytek NetBIOS specification, the ADAPTER_STATUS function returns the hardware address of the network interface card. However, NetBIOS returns the IP address of the local PC.

The following list outlines some NetBIOS software limitations:

- NetBIOS does not support the NetBIOS UNLINK command.
- There can be no more than one NO WAIT instance of any other function pending at one time.
- There can be no more than 31 simultaneous sessions (established through CALL or LISTEN functions).
- Datagrams can contain no more than 512 bytes of user data.
- The ADAPTER_STATUS function returns the first 335 bytes of adapter status information, which can contain no more than 16 names.

For more information about NetBIOS application programming, refer to the *IBM NETBIOS Application Development Guide*, available from IBM.

## 9.7 Troubleshooting NetBIOS

The following information provides solutions to problems that you might encounter while using NetBIOS:

NetBIOS does not load under the PC LAN Support program.

Remove the IBM NetBIOS device driver from the CONFIG.SYS file.

TCP connection messages appear.

Increase the number of TCP connections. See Section 9.5, Tuning the Kernel for NetBIOS.

You cannot access a mounted network drive.

Increase the number of TCP connections. See Section 9.5, Tuning the Kernel for NetBIOS.

You see the message `Network name not found`.

Verify that the scope value is set correctly. See Section 9.4.2, Partitioning Your Network with Scope.

You see the message `Protocol not loaded`.

Confirm that the kernel is installed/loaded.

In the Configuration application, choose NetBIOS, choose Additional, and set the values for Maximum NetBIOS sessions and for Maximum network control blocks to 32.

Increase the number of TCP connections. See Section 9.5, Tuning the Kernel for NetBIOS.

You see the message `Unable to unload NetBIOS`.

Stop any active sessions.

You see the message `Warning: The number of NetBIOS sessions was reduced to` *n*.

Configure your kernel with additional TCP connections. See Section 9.5, Tuning the Kernel for NetBIOS.

# Chapter 10
# Configuring Your PC Network Remotely Using DHCP

This chapter describes how to use the DHCP (Dynamic Host Configuration Protocol) client. The DHCP client allows a PC to obtain network configuration from either a DHCP or Bootp server, whenever the PC is started. Use the DHCP client when you want to

- Configure PCs that cannot maintain a local configuration, for either security or hardware reasons.
- Centralize configuration information by keeping it on DHCP or Bootp servers, rather than on individual PCs.

## 10.1 Before You Start Using the DHCP Client

Ensure that

- You have installed a network interface card in your PC, that the network interface card is correctly configured, and that it passes the diagnostic tests supplied by the card vendor.
- You have installed a correct network card driver.
- You or a network administrator have configured a Bootp or a DHCP server to provide network information.
- You have installed OnNet16 and you have a PCTCP.INI file.

## 10.2 Understanding the DHCP Client

DHCP provides a simple and reliable means for you to configure networked workstations from a server. DHCP centralizes TCP/IP configuration, manages the allocation of static and dynamic IP addresses, and automates much of the configuration process. In addition, DHCP simplifies workstation configuration by allowing PCs, as DHCP clients, to access network configurations with common characteristics that are shared across multiple workstations.

DHCP can configure the following parameters for a TCP/IP client:

- IP address
- Subnet mask
- Routers
- DNS domain name
- DNS servers

DHCP is an extension of the Bootstrap protocol (Bootp) and can interoperate with Bootp participants. A DHCP client can get network information from either a Bootp server or a DHCP server.

When a DHCP client program starts on your PC, the program sends a broadcast request for configuration information (such as an IP address and the addresses of network resources, such as name servers, routers, and so forth) out to the network. If a DHCP or Bootp server is available, the server responds to the client request and sends configuration information to the client

**Note:** To limit a DHCP client to accepting either DHCP server responses or Bootp server responses, add the line `protocol-accepted=` to the `[pctcp dhcp]` section of your PCTCP.INI file. Set the value of this parameter to either `dhcp` or `bootp`. The DHCP client will then accept responses only in the specified protocol.

The client request might contain values for any of the following items:

| Configuration Value | Definition |
|---|---|
| Media Access Control (MAC) address | The physical address of the PC on the network. The hardware address is guaranteed to uniquely identify the requesting client. MAC addresses are shown as hexadecimal numbers. A sample MAC address is 0x0000C094F82C.<br><br>**Note:**To find a PC's MAC address, use the Additional Configuration window under the Statistics application. |
| Client ID | A string that uniquely identifies the PC, such as a username.<br><br>This value can be used in place of the MAC address |

| | |
|---|---|
| | to identify a client configuration so that the same configuration is issued to a client regardless of the MAC address. (This supports the changing of the hardware without reconfiguring the server or assigning the PC client a different IP address.)<br><br>By default, the DHCP client uses a client ID that is based on the MAC address. |
| Subnet number | A number that is derived from the incoming DHCP client request (and not specifically entered by the client). |
| Class ID | A string that identifies the class to which the client belongs.<br><br>A system administrator can choose to group sets of PCs into separate classes, and return different configuration information for each separate class.<br><br>Examples of a class would be all PCs, or all computers belonging to a given department. |

A DHCP server compares the information provided in the client request to a database to select a defined configuration, called a “profile.”

When the server receives a request from a client, the server determines if the client has previously received configuration information and an address.

| If the client | The server then |
|---|---|
| Has previously received configuration information and an IP address | Assigns the same configuration to the client. |
| Has not requested configuration information before | Uses the information in the client request to select a configuration. |

After the server determines the configuration values and the IP address, the server sends this information back to your PC.

The information sent from the DHCP server to your PC includes the following:

- A set of configuration parameters, called “options.” Options define network resources (such as name servers) and site-specific information.
- A set of valid values for the options (such as IP addresses for servers).
- An IP address that is either a predetermined, permanent address, called a “static” address, or

an address that is dynamically allocated. (See Section 10.2.1, Assigning IP Addresses with DHCP, for information about dynamic and static address allocations.) Bootp servers can return only static addresses.

- The duration of the “lease” offered by the server (if the IP address is a dynamically allocated address) that determines the length of time that you can keep the address. (See Section 10.2.2, Assigning DHCP Leases, for information about leases.) Bootp servers do not return a lease time, because all Bootp responses are considered to be permanent.

The client uses the information it receives to update the kernel on your system. These values can change each time that you start your PC; the accuracy of the values is verified every time that the DHCP client obtains or renews a lease.

**Notes:**

- If you have an IP address and other network configuration values set in your PCTCP.INI file and you use the DHCP client program to get new information, the new information replaces the existing information in the kernel. It does not overwrite the PCTCP.INI file.
- If the domain is configured in the PCTCP.INI file, the DHCP client will not change the domain setting in the FTP Software kernel.
- If the hostname and domain are both configured in the PCTCP.INI file, the DHCP client sends the resulting fully qualified domain name to the DHCP server, which can then use that name to dynamically update a DNS server, provided both the DHCP and DNS servers support this option.

### 10.2.1 Assigning IP Addresses with DHCP

DHCP supports the following mechanisms for IP address allocation:

- Dynamic — An address allocated to the PC from a pool of available IP addresses at the time of the client's request
- Static — An address reserved for a specific PC.

The type of address you receive is determined by the DHCP server configuration.

**Dynamic Address Allocation**

Dynamic address allocation allows automatic reuse of an address that is no longer needed by the PC to which it was assigned. Dynamic allocation is particularly useful for a client that connects to the network only temporarily.

Each dynamically allocated address has a lease associated with it that determines the length of time you can keep the address. (See Section 10.2.2, Assigning DHCP Leases, for information about DHCP leases.)

**Static Address Allocation**

With static address allocation, the client always gets the same IP address identified by its MAC address or by its client ID. For dynamic address allocation, the DHCP server identifies the client configuration by its subnet number.

### 10.2.2 Assigning DHCP Leases

DHCP leases offer an automated mechanism for the safe distribution and reuse of IP addresses. A DHCP server assigns a lease to your PC when the server assigns a temporary IP address. The lease can vary in time from a few minutes to several years.

The server sets a default value for the lease time. When the DHCP client requests an address, the client also can request a lease time.

After receiving a lease time, the DHCP client saves a description of the DHCP lease to a file. This description includes the starting time of the lease and the duration of the lease. Therefore, the client can use this file to resume its network connection after a system crash, even if the server is not available. When the client restarts, it continues operating with its lease, if the lease is still valid.

If the `cache-lease=` parameter in the `[pctcp dhcp]` section of the PCTCP.INI file is set to `yes` (the default is `no`), the client automatically attempts to contact the DHCP server halfway through a lease time to renew the lease. Under most circumstances, the server will renew the lease successfully.

If the server is unavailable or will not renew the lease, the DHCP client continues to try to contact the server until the lease time has expired. When the lease expires, the client stops using the leased address and begins broadcasting to find another server willing to grant a new lease.

The new server can issue the PC a new lease, which might or might not be a lease for the PC's previous IP address. (Note that, when a lease does expire, network connections are lost. However, the DHCP client continues to broadcast over the network to discover another DHCP server that will grant it a lease.)

## 10.3 Configuring the DHCP Client

If, during installation, you select the option to obtain IP configuration from a DHCP server, the installation program configures OnNet16 so that this client starts automatically each time that you start Windows.

Each time Windows starts (on a system on which DHCP VxD has been selected), the DHCP client sends a broadcast request for configuration information to either a DHCP or a Bootp server, and waits for a reply.

When a DHCP or a Bootp server is available, the DHCP client takes the IP address and other information supplied by the server, and uses that information to configure the OnNet16 kernel.

In addition, when the lease is renewed with the server, the DHCP client reconfigures the OnNet16 kernel with any new or changed information provided by that server.

The DHCP client works in many environments without requiring additional configuration. In some environments, however, the network administrator might require the use of an alternate client ID, or might require a class ID to be supplied by the client.

**Note:** Ask your network administrator if a specific client ID should be used to identify your PC, or if a class ID is required.

To configure DHCP client parameters

1. Start the Configure application.
2. From the Settings menu, choose Advanced.
3. Select DHCP/Bootp and choose Modify to display the DHCP/Bootp configuration dialog box.
4. Make modifications as necessary, then choose OK.
5. Restart Windows.

You can use the Configure application to modify the following DHCP client parameters:

| DHCP Parameter | Description | Default Value |
|---|---|---|
| Class ID | String that identifies a set of computers to which a client belongs. | If this field is blank, no class ID is sent by the client. |
| Client ID | String that uniquely identifies the client. | If this field is blank, a default client ID is constructed, based on the MAC address of the network card. |
| Initial retry count | Number of times the DHCP client | Two times |

| | | |
|---|---|---|
| | retransmits at startup time before a timeout occurs.<br><br>If this value is increased and the initial timeout is not changed, the number of initial retries may remain the same as before. | |
| Initial timeout | Number of seconds the DHCP client waits at startup time before a timeout occurs. | 12 seconds |
| Requested lease time | Number of seconds for which a DHCP lease is desired. Note that, even if a requested lease time is set, the server is under no obligation to accept or to use the specified lease time.<br><br>**Note:** Under normal circumstances, it is not necessary to configure this parameter. | If this field is blank, the client will not send a requested lease time and accepts the lease time the server chooses to grant. |

## 10.4 Troubleshooting DHCP Clients

This section provides possible situations that you might encounter when you use the DHCP client to get network configuration information. The message is described first, followed by the recommended course of action.

```
No DHCP servers were found. The DHCP VxD will not be loaded.
```

This message appears when the DHCP client sends a broadcast request for configuration information to the DHCP server and a reply is not received within the timeout period. It might be necessary to reconfigure either the initial timeout period or the initial retry count in environments in which a DHCP (or Bootp) server takes a long time to respond. Contact your network administrator.

# Chapter 11
# Configuring Your PC as an SNMP Agent

This chapter describes how to use the SNMP (Simple Network Management Protocol) agent, when you install OnNet16. The SNMP agent allows other hosts and network management stations using SNMP to examine your PC's statistics and configuration information (similar to that produced by the Statistics application). In terms of the client/server paradigm, an SNMP agent is a server, while an SNMP management station is a client.

When you install OnNet16, the SNMP agent is an MIB-II agent. It supports most of the SNMP Management Information Base (MIB) version 2, as defined in RFC 1213.

## 11.1 Before You Configure Your PC as an SNMP Agent

Before you use the SNMP agent, ensure that

- The Server Control application is installed in your OnNet16 program group.
- The SNMP MIB-II agent is in Server Control when you start it up.
- The MIB_II.SRV file is in the same directory as the Server Control executable, CTLAPP.EXE.
- The COMMUNIT.CNF and TRAPCOMM.CNF files are in the directory defined in the `etc-dir=` parameter in the PCTCP.INI configuration file.

**Note:** If you do not have the Server Control application on your system, you can run the installation program and select the Server Control component. The SNMP MIB-II agent is installed as a subcomponent of Server Control.

An SNMP network management product, such as a PC running the PC/SNMP Tools applications, displays the object IDs (associated text names) for the MIB variables. If you are using another network management product, refer to its documentation to determine if that product supports non-numeric names for MIB variables.

## 11.2 Configuring the SNMP Agent

You can run the SNMP agent using its default configurations, or you can modify those configurations. The SNMP configuration files are

- The SNMP community file, COMMUNIT.CNF.
- The SNMP error condition (trap) file, TRAPCOMM.CNF.

The installation program installs sample TRAPCOMM.CNF and COMMUNIT.CNF files, which contain configuration instructions. If the configuration files are not formatted correctly, the SNMP agent displays various error messages and aborts.

**To configure an SNMP agent for your system**

1. Add entries to the COMMUNIT.CNF configuration file using a text editor such as Notepad or DOS Edit.

   This configuration file contains entries that define SNMP community names, IP addresses, and privileges (NONE, READ, and WRITE) used to validate incoming SNMP request packets. If a community specified in an incoming packet is not authorized for the operation that it requests, the request is discarded and an SNMP error condition message (trap) might be sent (if specified in TRAPCOMM.CNF *)*.

   **Note:** To allow an SNMP management station to set values in your RW-accessible MIB variables, you must set your community privilege type to `WRITE`.

   Each entry in the COMMUNIT.CNF file must define a community name, an IP address (in dot notation), and the appropriate privilege (in uppercase letters). For example,

   ```
   interop 128.127.53.190 READ
   ```

   **Note:** Most SNMP agents use the default community name `public`.

2. Add entries to the TRAPCOMM.CNF configuration file using a text editor such as Notepad or DOS Edit.

   The TRAPCOMM.CNF file contains entries that define the IP addresses, SNMP communities and User Datagram Protocol (UDP) port numbers to use when sending SNMP Trap packets. The SNMP agent collects three kinds of error condition messages (or traps): ColdStart (when the agent is loaded), WarmStart (when the agent is restarted), and AuthenticationFailure (when a query specifies an unknown SNMP authentication community). In order to receive AuthenticationFailure traps, you must set the variable snmpEnableAuthenTraps to 1.

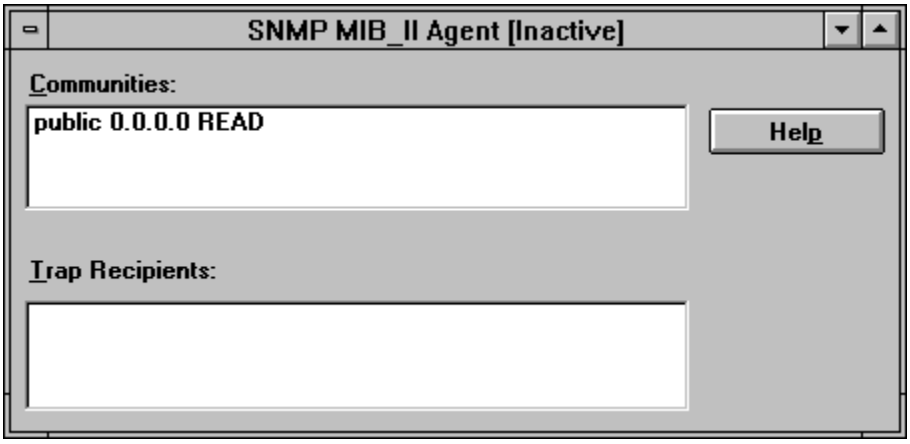
   You can specify up to five entries in this file. Each entry must define a community name, an IP address (in dot notation) and a UDP port number where you want to send error condition messages. For example,

   ```
   newcommunity 128.127.59.154 162
   ```

**To view the current configuration of the SNMP MIB-II agent**

Start the Server Control application and double-click the SNMP MIB_II Agent icon.

The SNMP MIB_II Agent window appears.



## 11.3 Starting the SNMP MIB-II Agent

**To start the SNMP MIB-II agent**

1. Start the Server Control Application.
2. Select the SNMP MIB_II Agent icon.
3. From the Commands menu, choose Start.

   **–or–**

   From the Server Control toolbar, choose the Start button.

After you start the SNMP MIB-II agent, the word Active appears under its button.

## 11.4 Stopping the SNMP MIB-II Agent

After you start the SNMP MIB-II agent, the word Active appears under the SNMP MIB_II Agent icon. You can stop the SNMP MIB-II agent if it is running.

**To stop the SNMP MIB-II agent from the Server Control application**

1. Select the SNMP MIB_II Agent icon.
2. From the Commands menu, choose Stop.

   **–or–**

   From the Server Control toolbar, choose the Stop button.

After you choose Stop, the button name contains the word Inactive.

# Chapter 12
# Monitoring Kernel Activity with IPTrace

IPTrace is a monitoring tool that captures and displays detailed information about packets that the kernel sends and receives. You use this information to analyze networking problems on a specific host. In addition, you might need to report this information to support staff if you telephone a supplier to solve networking problems.

IPTrace captures communication traffic *local* to the PC in which it is installed. IPTrace does *not* provide a way to analyze network traffic, because IPTrace receives only those packets that are touched by the installed and running OnNet16 kernel.

For those users who are familiar with the network analyzer tools LANWatch® and LANCatch™, it is important to note that the file capture facility in IPTrace is not compatible with LANWatch or with LANCatch. Trace files captured with IPTrace cannot be displayed by LANWatch or by LANCatch, and vice versa.

## 12.1 Before You Start Using IPTrace

You should ensure that

- IPTrace was installed as part of the OnNet16 installation. The installation of IPTrace is an option that is installed by default during the OnNet16 installation. However, if you did not install IPTrace, you can install it later by rerunning the OnNet16 **setup** program.
- The OnNet16 kernel is running.

If you are uncertain about this information, contact your system administrator.

When IPTrace is installed, the installation program adds an IPTrace icon to the OnNet16 WinApps group in Windows. The following files are added to the \PCTCP directory for IPTrace:

IPTRACE.EXE
IPTRACE.FON
IPTRACE.HLP
IPTRACE.INI
PRNDMP.EXE
VXDLW.386

## 12.2 Supported Protocols

The IPTrace program contains the decoding information for Ethernet and Token Ring packets, and for SLIP and PPP serial line packets.

Because IPTrace is a tool that displays only packets sent or received by the kernel, no non-TCP/IP protocols, such as Banyan VINES protocols or Novell NetWare protocols, are supported under IPTrace. In addition to the SLIP and PPP serial line protocols, IPTrace supports the following TCP/IP protocols:

| | | |
|---|---|---|
| ARP | IP | RPC |
| BOOTP | NetBIOS (RFC1001 and RFC1002) | SNMP |
| DOMAIN | | TCP |
| EGP | NFS | TELNET |
| ESIS | OSPF | TFTP |
| ICMP | RARP | UDP |
| IEN116 | RIP | |

## 12.3 Using a Mouse with IPTrace

Because IPTrace operates in a DOS environment, if you want to use a mouse you must ensure that an appropriate mouse driver is loaded before you start Windows. However, it is not necessary to use a mouse with IPTrace; all of the procedures explained in this chapter discuss alternative ways of performing tasks, with and without a mouse.

**Note:** You can use a mouse with IPTrace only when you are using IPTrace in full-screen mode and you have an appropriate DOS mouse driver loaded.

For more information about loading a mouse driver, refer to your Microsoft Windows documentation, or the documentation that came with your mouse.

## 12.4 Starting, Stopping, and Using IPTrace

IPTrace runs in an MS-DOS prompt session while the OnNet16 kernel is running, and it displays in real time the packets that the kernel is sending and receiving. You can also save this activity to a trace file for later analysis. For more information about saving data to a trace file, refer to Section 12.8, Saving Packets to a Trace File.

**To start IPTrace**

Double-click the IPTrace icon in the OnNet16 WinApps group, on a Windows 3.*x* system.

**–or–**

Start a DOS session and enter `iptrace` at the DOS prompt. There are no options or command-line arguments.

IPTrace starts an MS DOS prompt session. You can run IPTrace in the background while you do regular work or perform various network tasks.

**To use the IPTrace user interface**

1. With the mouse, choose a command name in the toolbar at the top of the IPTrace screen.

   **–or–**

   Choose a command name by simultaneously pressing Alt + the highlighted letter of the command.

   IPTrace displays a menu of subcommands.

2. Choose a subcommand from the menu with the mouse.

   **–or–**

   Use the arrow keys to move up and down the menu. Press Enter to choose the highlighted menu command.

   If you decide not to choose a command, press Esc.

After you have chosen a command by pressing Alt + the highlighted letter of the command, you can use the right and left arrow keys to move across the horizontal list of commands. As you move across, the subcommand menu is displayed for each command.

**To get online Help**

From the Help menu, choose any of the following commands: Contents, Commands, Tasks, and About IPTrace. Choosing any of the topics in the submenus provides more detailed help messages.

**–or–**

Enter **?** to get online Help.

**To exit from IPTrace**

From the File menu, choose Exit IPTrace.

**–or–**

Press the letter **q** while in the Real-Time Decode Display.

In either case, you must respond to the program's prompt to confirm that you want to exit.

## 12.5 Using IPTrace to Monitor the Kernel Operations

As soon as you start IPTrace, it begins capturing traffic to a queue, decoding the data, and displaying the data in a real-time decode display, as shown in Figure 12-1.

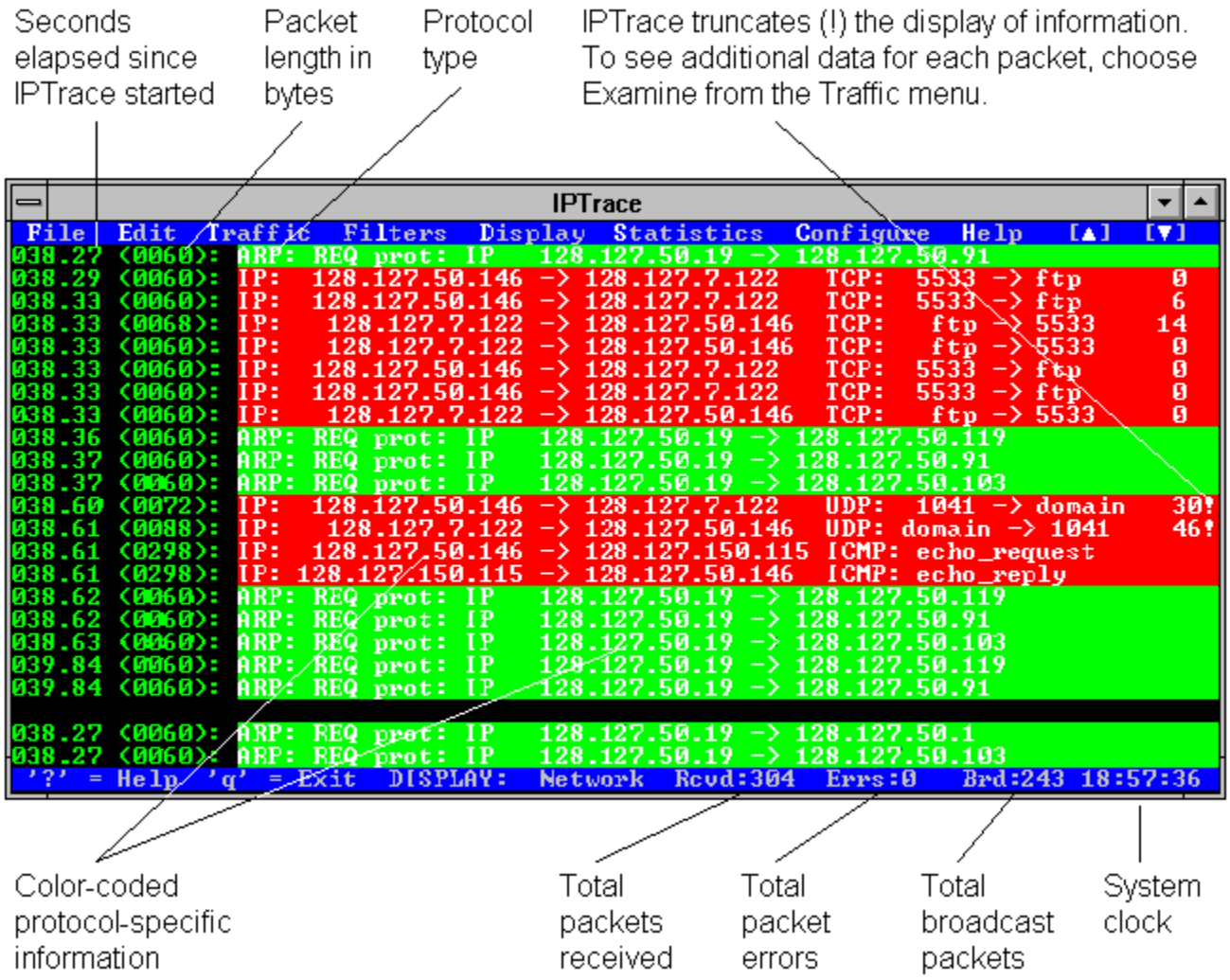



Figure 12-1 Real-time Decode Display

Each line in the display corresponds to a packet of real-time traffic currently being sent or received by the OnNet16 kernel. Packets remain in the queue until they are overwritten by new traffic. The number of packets that the queue can store depends on the configured packet size and on the available (free) conventional memory and expanded memory in your PC.

By default, IPTrace saves the first 160 bytes of each packet, which should be sufficient for looking at almost all protocol headers and some data. For example, if IPTrace captures an IP packet, it displays the protocol type (in this case, IP), the source and destination hostname or address, and information specific to the protocol. If you want IPTrace to capture more data, configure the buffer length (packet size) to a higher number by changing the `pkt-size=` parameter in your

IPTRACE.INI file. For more information, refer to Section 12.10, Changing the IPTrace Configuration.

## 12.6 Modifying the Display of Packet Data

IPTrace lets you switch between displaying packets.

- In color or in monochrome.
- With a hardware address or with a vendor ID.
- With a network address or with a hostname.
- With information at the various protocol layers shown, or with information hidden.

From the Real-time Decode and Examine displays, IPTrace lets you modify the appearance of the packet data, as shown in Table 12-1.

Table 12-1 Methods of Modifying the Display of Packet Data

| To do this | Choose this Display command |
|---|---|
| Switch between color and monochrome screen displays. | Color \| Monochrome |
| Switch between the display of hardware addresses (such as 02608cb1eb8d) and the display of vendor IDs (such as 3Com b1eb8d). | Hardware Address \| Vendor ID |
| Switch between the display of network addresses (such as 128.127.50.122) and the display of hostnames (such as flower.ftp.com). | Network Address \| Hostname |
| Display a submenu that lets you turn the display of data on and off for each protocol layer. | Protocol Layers |

## 12.7 Examining a Packet

IPTrace provides Examine commands that let you display all the captured data in the buffer on a packet-by-packet basis, as shown in Table 12-2.

Table 12-2 Methods of Examining a Packet

| To do this | Choose this Examine command |
|---|---|
| Display a snapshot of the queue, one packet per line. | Snapshot |
| Display a half-screen view of the decoded data from a selected packet. | Summary |
| Display the decoded data and the hexadecimal contents of a selected packet. | Detail |
| Display a full-screen view of the decoded data from a selected packet. | Zoom |

**To display a snapshot of the queue**

1. From the Traffic menu, choose Examine.

   IPTrace displays a menu of Examine commands.

2. From the Examine menu, choose Snapshot.

   IPTrace stops capturing packets and, instead, displays the end of the queue of saved data, highlighting the last packet in memory. If you choose Summary, Detail, or Zoom, IPTrace displays details of the last (or highlighted) packet.

**To move around the queue**

Choose either of the arrow commands to the right of the main menu, or press  (up arrow), ↓ (down arrow), Page Up, Page Down, Home, Ctrl+Home, or Ctrl+End.

Ctrl+Home always takes you to the first packet in the queue; Home takes you to the top of the last group of captured packets. For example, if you examine commands in the Snapshot display, go to the Real-time Decode display to capture more traffic, and then return to the Snapshot display, you can press Home to go to the beginning of the new group of packets.

**To examine a packet**

Select the packet by moving the  (up arrow) and ↓ (down arrow) keys and pressing Enter. From the Examine menu, choose Summary, Detail, or Zoom, as described in Table 12-2.

Note that you can switch quickly among the Summary, Detail, and Snapshot displays by repeatedly pressing Enter after selecting a packet. (To switch between displays in the reverse order, press Esc.)

**To display both the decoded data and the hexadecimal contents of a packet**

From the Examine menu, choose Detail.

For example, if you select an FTP packet that is an acknowledgment of the request to close a connection, and then choose Detail from the Examine menu, IPTrace displays the decoded data and the hexadecimal contents of that packet, as shown in Figure 12-2.


Figure 12-2 Detail Display of a Packet

Packet details depend on the hardware and protocol used to send the packet. For example, the hardware header information shown in Figure 12-2 includes the Ethernet hardware source and destination addresses and the Ethernet packet type (type: IP(0x800)).

To scroll the bottom half of the screen, press Tab. To reverse scroll, press Shift+Tab.

The IP header fields shown in Figure 12-2 are as follows:

| IP Field | Description |
|---|---|
| Internet: | The source and destination addresses (128.127.140.5 –>128.127.140.255) |
| hl: | The header length (5) in 32-bit words |
| ver: | The version (4) of the IP header |
| tos: | The type of service (0) |
| len: | The total length (132) in bytes of the IP portion |
| id: | The identification value (0xd225) |
| fragoff: | The fragment offset (0) |
| flags: | Control flags (00) |
| ttl: | The time to live, or the number of hops (30) before the packet is discarded |
| prot: | The symbolic value (if known) of the next-level protocol (UDP), and the numeric value (17) |
| xsum: | The checksum (0xb040) |

The UDP header fields shown in Figure 12-2 are as follows:

| UDP Field | Description |
|---|---|
| UDP: | The source and destination routes (111 –> 5264) |
| len: | The total length (112) in bytes of the UDP and RIP portions |
| xsum: | The checksum (0x762) |
| RIP: | A Routing Information Protocol table of 5 routes and the cost (number of hops) of each |

## 12.8 Saving Packets to a Trace File

IPTrace lets you save packets as they are captured, or as they reside in the queue, to a trace file for later analysis, as shown in Table 12-3.

Table 12-3 Tasks for Saving Packets to a Trace File

| To do this | Choose this File command |
|---|---|
| Save real-time traffic directly to a trace file. | Start Trace \| Stop Trace |
| Analyze a trace file with IPTrace. | Load Trace |
| Save previously captured packets to a trace file. | Save Packets in Memory |

When you choose a File command, IPTrace prompts you for a filename and saves data to the file in binary format. This file resides in your current directory unless you specify a different directory. After you have saved packets to a trace file, you can examine the data at any time by loading the file back into IPTrace. Note that you cannot save any markers from the Examine displays to a trace file.

**To save real-time traffic directly to a trace file**

1. From the File menu, choose Start Trace | Stop Trace.

   IPTrace prompts you for the filename.

2. Enter the filename.

   IPTrace begins copying all new packets to the file.

3. From the File menu, choose Start Trace | Stop Trace again when you have finished capturing packets.

**To save previously captured packets to a trace file**

1. From the File menu, choose Save Packets in Memory.

   IPTrace prompts you for the filename of the trace file.

2. Enter the filename.

   IPTrace prompts you for the range of packets you want to save.

3. Enter the numbers of the first and last packets in the range.

   IPTrace saves the specified range of packets to the trace file. The queue in this trace file

starts at packet number 1 and does not retain the old packet numbers.

For example, if you enter `22 500`, IPTrace creates a queue of 479 packets in the specified trace file. If you do not specify a second number, IPTrace saves all packets after the specified packet number.

If you press Enter without specifying any numbers, IPTrace writes all packets currently in memory to a file.

Note that the Save Packets in Memory command works only from the Examine displays.

**To load a trace file**

1. From the File menu, choose Load Trace.

   IPTrace prompts you for the filename of the trace file.

2. Enter the filename.

   IPTrace displays the queue of saved packets.

Note the following:

- IPTrace loads the data into your PC's packet buffer. If there is more data than the packet buffer can hold, you must press End to move to the end of the queue, and then press Page Down to load more data from the trace file. To return to the earlier data, press Home to move to the top of the queue, and then press Page Up to load the previous data from the trace file.
- If you edit the queue in a loaded trace file and you want the change to be permanent, you must save the queue again by choosing Save Packets in Memory from the File menu.
- You can load a trace file into IPTrace at startup.
- If you load a trace file over a filter, IPTrace takes extra time to filter as it loads the file.

## 12.9 Converting a Binary Trace File into an ASCII Text File

The Prndmp offline analyzer utility turns a binary trace file (dump file) into a readable and printable text file, including the time the packet was received, the length of the packet, and the packet header information. Use this utility to make archives of a trace file, or to produce hardcopy for debugging purposes. After you have ASCII output, you can import the text into various spreadsheet software programs and further refine the output.

You use the Prndmp analyzer from a DOS prompt, instead of from Windows. The format for using the **prndmp** command is as follows:

C:\>**prndmp** [**-h**] [**-l***length*] [**-m**] [**-n**] [**-s**] *tracefile* [*outfile*]

### 12.9.1 Prndmp Command Line Arguments

The Prndmp program requires one argument, *tracefile*, which is the file you want to print, and allows an optional argument, *outfile*, which names the file to hold the data. By default, Prndmp sends the output to the screen.

The options you can use in the Prndmp command line are as follows:

**-h** Displays the packet, byte-by-byte, in hexadecimal and ASCII, in addition to the packet's symbolic parsing.

**-l** *length* Lets you set the packet data length to a number other than 160 (the default). This option is useful when combined with the **-h** option. It can also affect parsing of SNMP, RIP, domain name, and other protocols where most or all of a long datagram can be parsed.

**Note:** This parameter is only useful if IPTrace is configured to capture larger packets by using the `pkt-size=` parameter in the IPTRACE.INI file. For more information, refer to Section 12.10, Changing the IPTrace Configuration.

**-m** Disables the manufacturer information and displays the hardware addresses. The **-m** option is useful only for trace files containing Ethernet or 802.5 packets. The first three bytes of an Ethernet or an 802.5 address determine what manufacturer produced the Ethernet interface associated with that address. This option displays the first three bytes of an Ethernet or an 802.5 address as hexadecimal numbers.

**-n** Disables the printing of hostnames, and displays the IP addresses if a host file exists.

*outfile* Specifies the name of the output file where you want the data stored. If you use the *outfile* option, you must specify it at the end of the command line. If you do not specify an output file, Prndmp displays the data on the screen only.

**-s** Shortens the output by printing less information. Displays one line of information per packet.

If you have a host table set in your IPTRACE.INI file, the Prndmp program uses that information when displaying hostnames.

**Example**

The following example shows output from the TRAFFIC.TRC binary trace file. The Prndmp program displays the same information as the summary display in IPTrace, including the delta time from the previous packet.

```
C:\>prndmp traffic.dmp

PRNDMP.EXE version 2.0
Allocated 1024 packets
Receive time: 1.255 (+0.000) packet length: 60 received length: 60 Ethernet: (WDgtl
4ca427 - DEC 03e7c0) type: IP(0x800) Internet: kiwi.abe.com - vex.abe.com hl: 5 ver: 4
```

tos: 0x10 len: 41 id: 0x3b34 fragoff: 0 flags: 0x2 ttl: 64 prot: TCP(6) xsum: 0xf9a3
TCP: 4358 - telnet(23) seq: 0d2013e0 ack: 5493225c win: 3867 hl: 5 xsum: 0xefbc urg: 0
flags: <ACK><PUSH> data (1/1): .

## 12.10 Changing the IPTrace Configuration

IPTrace reads configuration values from a text file called IPTRACE.INI. The installation program automatically creates this file. The first section (`[iptrace]`) lets you customize operating defaults. Note that parameters in the `[iptrace]` section are optional and that IPTrace ignores parameters preceded by a semicolon, as shown in the following example:

```
[iptrace]
; class=<packet driver class>
color=c
emm=yes
; filter=<filter file>    ; no default filter file
high-res=no
; hw-host-table=<hw host table>   ; no default hardware address table
; ip-host-table=<ip host table>   ; no default ip host table
; layers=1,3,4,5,6,7
manufacturer=yes
pkt-size=160                     ;To capture more data, set packet size to a higher
number.
; serial-type=PPP
show-names=yes
```

**To set parameter values in the IPTRACE.INI file**

1. Open the file with a text editor.
2. Replace the current value or variable (such as <filter file>) with a new value.
3. Save the IPTRACE.INI file and exit from the text editor.
4. Start IPTrace to use the new configuration.

# Chapter 13
# Troubleshooting Network Connectivity

This chapter discusses how to fix network problems that might occur when you are using the OnNet16 kernel. When you encounter problems with making connections to other network hosts, there are several things to investigate. The problem might be the result of

- Physical problems with the network; malfunctioning hardware, modems, cables, and so forth.
- Network congestion.
- Faulty host configurations.
- Faulty router configuration.
- Incorrect or missing entries in name translation tables on your PC or on the Domain Name System (DNS) servers on your network.

## 13.1 Before You Start Troubleshooting Host Connections

Investigating and solving problems with host connections is not difficult, but it might take you down several paths. You should have a basic understanding of your network, and you should be able to restore your system to its original configuration if necessary. For those reasons, ensure that you

- Make backup copies of your AUTOEXEC.BAT and PCTCP.INI files. With these copies, you can compare and test incremental changes to your configuration. In addition, you can restore your original configuration if you are not satisfied with the new configuration.
- Understand basic Internet features and addressing schemes, if you are accessing the Internet.
- Understand the basic setup and components of your local network.
- If you are using a modem or a wireless modem, understand how to use the modem and have the modem documentation on hand.
- If you are using an ISDN line, be sure that your ISDN card supports WINISDN, is connected to an ISDN line, and is working.

## 13.2 Testing Your PC's Initial Network Connection

After you install OnNet16 on your PC,

- Verify that your IP address is correctly defined in the PCTCP.INI configuration file.
- Test your PC's IP address to confirm that the kernel installed correctly.
- Test your PC's connection to other hosts running TCP/IP.

**To test your PC's IP address**

**Windows** Use the Ping application and type your PC's IP address in the Host box.

**DOS** Use the **ping** command and specify your PC's IP address. For example,

```
C:\>ping 128.127.50.182
```

Your network administrator can tell you the correct IP address for your PC.

**To examine and verify your PC's IP address**

**Windows** Use the Statistics application and open a new session. From the Text Views Available dialog box, choose the Statistics option and view your IP address, which is listed in the System Information box.

**DOS** Use the **inet stats** command. For example,

```
C:\>inet stats
```

**To correct an invalid IP address**

**Windows** Use the Configure application; from the Basic menu, select Internet Addresses. Click Modify to view or change the IP address. When you have entered the correct IP address, Save and Exit.

**DOS** Use the **pctcpcfg -k** command to change the IP address in the kernel and the `[pctcp ifcust 0]` section of the PCTCP.INI file. For example,

```
C:\>pctcpcfg -k ifcust -s 0 ip-address 134.156.50.11
```

More options are available with the **pctcpcfg** command. Enter **`pctcpcfg -?`** at the DOS prompt for a complete list of options.

## 13.3 Troubleshooting Host and Network Connections

If you have problems making a network connection, use the Ping application to send an echo request to a remote host to test host and network availability.

- If the target host is active and sends a reply to your ping request, you know that the network media forming the path to that host is working.
- If a host fails to respond to a network request, it means that there has been a failure at one of several points from your PC to the remote host. The host might not be working and is unable to respond, some network or gateway in the path between your PC and the target host might not be working, or the host might not implement the service that you are requesting.

For more information about the Ping messages, refer to Section 13.9, Interpreting Ping Messages.

**To send an ICMP echo request and receive debugging information**

**Windows** Use the Ping application and type the target host's IP address in the Host box.

**DOS** Use the **ping** *IP address* command and specify the target host's IP address. For example,

```
C:\>ping 128.127.50.128
```

**Note:** In Windows, the Ping application offers minimal debugging information. For maximum debugging information, you might want to use the **ping** command from the DOS prompt. Various options are available with the **ping** command. Enter **`ping -?`** at the DOS prompt for a list of options that might provide useful information for your specific case.

**To diagnose simple network congestion**

**Windows** Use the Ping application to send repeated echo requests to the remote host. Choose Stop when you have sent the desired number of requests.

**DOS** Use the **ping -n** *value* **-z** *hostname* | *IP address* command to send echo requests to the remote host a specific number of times. For example,

```
C:\>ping -n 2 -z vex
host responding, time = 25 ms
    host responding, time = 75 ms
    Pinged host 2 times with 0 failures,
    round-trip average (ms) = 50
```

**–or–**

Use the **ping -t** *hostname* | *IP address* command. For example,

```
C:\> ping -t vex
Pinging host vex.xyz.com repeatedly
    To exit, type q
    # of tries = 635, failures = 0, time = 60:
    round-trip average (ms) = 36
```

**To analyze packet loss and network load**

**Windows** Use the Statistics application. Start a session and choose the Statistics option from the Text Views Available dialog box to get detailed information about network traffic and errors.

**DOS** Use the **inet stats** command. For example,

```
C:\>inet stats
```

**To display header information from incoming and outgoing packets**

**DOS** Use the **ping -d -z** *hostname* | *IP address* command to display header information from the *incoming* packet. For example,

```
C:\> ping -d -z chaco
Dump of incoming packet
    Version = 4 IP header length = 5 Precedence = Routine
    Type of service = Normal
    Total length = 284 Protocol = 1 TTL = 255
```

Use the **ping -d# -z** *hostname* | *IP address* command to display header information for the *outgoing* packet. For example,

```
:\> ping -d# -z chaco
Dump of outgoing packet
    Version = 4 IP header length = 5 Precedence = Routine
    Type of service = Normal
    Total length = 284 Protocol = 1 TTL = 64
```

## 13.4 Confirming That Your PC Sends ARP Requests

The following types of computers using DIX Ethernet, IEEE 802.3 Ethernet, or 802.5 Token Ring might accept Address Resolution Protocol (ARP) requests:

- Another host running OnNet16 or OnNet
- A UNIX host
- Any other host that supports a TCP/IP protocol stack

If you are not certain that your PC is sending ARP requests to these hosts, test the transmission and receipt of requests.

**To test transmission of ARP requests**

**Windows** Use the Ping application and type the target host's IP address in the Host box. Choose Start to reach another IP host that accepts ARP requests.

Use the Statistics application and choose the ARP command from the Open Text window to examine the ARP cache.

**DOS**

1. Use the **ping** *IP address* command.
2. Use the **inet arp** command. For example,

```
C:\>inet arp

 ARP cache:
    128.127.52.141: 0080c0205516 expires: 259 sec.
    128.127.52.30: 08004c5e8c1a expires: 782 sec.
    128.127.57.30: 08006a022aea expires: 527 sec.
```

The ARP cache displays entries for each network device that has recently sent or replied to an ARP request. The entries specify each host's IP address, the corresponding hardware address, and expiration information.

Look in the ARP cache listing for your PC's IP address and the hardware address of your network interface card. If this information is in the listing, then your PC is sending ARP requests and the target computer has received those requests.

Inspect the ARP cache listing to make certain that your IP address is not listed more than once. Duplicate IP addresses in the ARP cache listing cause serious conflicts on the network.

There are many conditions that cause ARP reply failure, including the following:

- Your PC fails to send out an ARP request.

  This is most often caused by a hardware interrupt conflict in your PC. Compare the network interface card's hardware and software settings with the memory and interrupt settings of the other devices installed in your PC. If any of these settings conflict, refer to the respective manuals for alternative settings and change them.

- Your PC sent the ARP request out to the local area network (LAN), but the request did not reach the target host.

  If your local LAN segment contains any repeaters or bridges, and if any of these devices connects your PC to the target host, problems with these devices can cause ARP failure. Notify your system administrator.

- The ARP request successfully reached the target host but was not read off the network by the target host's network interface card, or your PC did not receive the ARP reply.

  This might indicate a problem with the network interface card in the target host. Contact the administrator of the remote host for help.

  **–or–**

  There might be problems with intervening repeaters or bridges. See your system administrator to investigate any suspected problems with local repeaters and bridges.

## 13.5 Troubleshooting Your Router Connections

An IP router (gateway) usually connects two *different* types of network media; for instance, Ethernet to Token Ring. The router has at least two network interface devices installed, each with a unique IP address, which allows you to access another part of the network. If there is a router on your network, your OnNet16 configuration file should list the IP address of the router through which you can access the rest of the network.

If you are having problems accessing other hosts on your network, it might be because of problems with your router configuration or service.

### 13.5.1 Testing Router Configuration

Try to reach the router(s) on your network with the Ping application or with the DOS **ping** command.

**To reach the router**

**Windows** Use the Ping application and specify the IP address assigned to a router on your network segment.

**DOS** Use the **ping** *IP address* command and specify the IP address assigned to a router on your network segment.

If you cannot reach the router, display the router entries in your configuration file and be sure that the entries are accurate. You might have to ask your network administrator for the correct router addresses.

**To display router configuration entries**

**Windows** Use the Configure application. Select Internet Addresses and choose Modify to display the configured router entries. Exit without Saving.

**DOS** Use the **pctcpcfg ifcust -s** *n* **router** command to display the entries. For example,

```
C:\> pctcpcfg ifcust -s 0 router ""

[pctcp ifcust 0] - router=128.127.50.100
[pctcp ifcust 0] - router=128.127.50.117
[pctcp ifcust 0] - router=128.127.50.109
```

If the router entries listed in the configuration file do not refer to the actual IP addresses of any network routers, you need to correct the entries or add new ones.

**Note:** Though only one is necessary, you can list a maximum of three routers in your configuration file. If the first router is disabled, the second router on the list is used instead; if the second router is disabled, the third is used.

**To configure a new router**

**Windows** Use the Configure application. From the Internet Address menu, choose Modify. Type the IP address for each router that you want to add to your configuration. You can also modify any incorrect address that might be listed. Save and Exit.

**DOS** Use the **pctcpcfg -k ifcust -s** *n* **router** *IP address* . . .command to add a new router address to your configuration, as in the following example:

```
C:\>pctcpcfg -k ifcust -s 0 router 128.127.50.13
Setting INI and kernel:[pctcp ifcust 0] -
router = 128.127.50.13
```

### 13.5.2 Testing Router Service

You can test the router service by tracing the route of a packet through the network by using the Ping Traceroute command in Windows.

**–or–**

Use the DOS **ping** command, as follows:

- The **ping -q** command traces the route of the packet and lists the number of hops, the domain name (if supported) and IP addresses of the hosts that handled the packet, and the round-trip time.
- The **ping -i** *n* command follows an Internet Control Message Protocol (ICMP) packet as it makes its way through the network. The command output summarizes the success (or failure) of the echo request, the round-trip time, and the packet's time to live (TTL).

**To test IP routing**

**Windows** Use the Ping application and choose the Traceroute command. Type the target host's name or IP address in the Host box and choose Start.

**DOS** Use the **ping -i** *n* **-z** *hostname* | *IP address* command. For example,

```
C:\>ping -i 300 -z owl.com
host responding, time = 50ms, Time to live = 236
```

Use the **ping -q -z** *hostname* | *IP address* command to get a full listing of the routers between your PC and the target host. For example,

```
C:\>ping -q -z owl
hop 1:  128.127.50.4   router-54a.ddd.com
hop 2:  128.127.5 x.1  router-2a.ddd.com
Target (128.124.5 x.23) reached on hop 3,
round-trip time 60 ms.
```

**To test IP router hops**

**Windows** Use the Ping application and choose the Traceroute command. Type the target host's name in the Host box and choose Start.

**DOS** Use the **ping -q -z** *hostname* | *IP address* command to get a full listing of the routers between your PC and the target host. For example,

```
:\>ping -q -z owl
hop 1:  128.127.50.4   router-54a.ddd.com
hop 2:  128.127.5 x.1  router-2a.ddd.com
Target (128.124.5 x.23) reached on hop 3,
round-trip time 60 ms.
```

**To see if the router is disabled**

Use the Ping application in Windows or the **ping** command in DOS. Specify the IP address assigned to a host on your local network segment. If the Ping request succeeds, use the Ping

application and specify the IP address assigned to a known host on the remote side of the router. If this request succeeds, the router is up; if this does not succeed, the router is probably down and you should contact your system administrator.

### 13.5.3 Using Output from Ping as Debugging Information

Each IP router that handles the packet must decrement the TTL (Time-to-live value) by 1. If the TTL for a packet is 0 after being decremented, the router returns a `Time To Live Exceeded` ICMP message to your PC, displaying the IP address of the router sending the ICMP message.

**To determine the point of failure**

**DOS**

1. Use the **ping -i** *n hostname* | *IP address* command with the IP address of the router returning the ICMP message.
2. Increase the **-i** *n* value by 1.
3. If you continue incrementing the TTL by 1 up to the point of failure, you can determine where your transmission fails.

## 13.6 Isolating Name Resolution Problems

If your PC can reach another host using that host's IP address but not its hostname, your system has encountered a name-to-address resolution problem. This occurs when the tables that translate names to IP addresses are either faulty or unavailable to you because of network problems. Problems can occur in your local host table and in the DNS server's table.

For example, assume that you want to reach a remote host named chaco (IP 128.127.50.32). You successfully reach host chaco using its IP address, but you get a `hostname unknown` message when using the hostname. This failure signifies a name resolution problem that might be caused by either of the following:

- Unavailable DNS (Domain Name System) servers on your network or DNS servers that are receiving/transmitting incorrect information could be the source of your problem. If you suspect that the DNS server is faulty, contact your network administrator or the DNS administrator for assistance.
- Incorrect or missing entry for the named host in the local host table file on your PC.

  **Note:** You can choose to have a local host table, which is a useful file that relates hostnames to IP addresses and defines aliases so that you can substitute short, memorable names for long addresses that are difficult to type. However, if you have a large network, or a network with a changing population, it might be difficult to maintain an accurate and up-to-date host table on your local PC. If there are well-maintained DNS servers on your network, you might find it more reliable to use those DNS servers.

  The system always searches the local host table first to resolve names, and then it refers to the DNS entry.

**To test name-to-address and address-to-name translation**

Use the Query application's Host program. To use it,

- Specify a hostname to learn the host's IP address.
- Specify the IP address of a host to learn the host's name.

**Note:** On the Internet, this number-to-hostname database is not as well maintained as the hostname-to-number database. As a result, the Query application's Host program command might not display a *canonical* (officially listed) name (`cname`) that corresponds to the IP address.

## 13.7 Testing Remote Connections Using the Finger Command

If you do not have access to a remote IP host that supports a Telnet or an FTP server, or if you do not have a valid user ID on the remote host, you can use the Query program's Finger command.

**To test remote connections**

Use the Query program's Finger command.

## 13.8 Troubleshooting the Local Network Configuration

Networking problems might be the result of an incorrectly configured kernel. For more information about configuring and tuning the kernel, refer to Chapter 8, Configuring and Tuning the Kernel.

If your examination of the kernel configuration and the configuration files does not pinpoint any obvious errors or omissions, the problem might be related to the local network configuration. You can pinpoint problems and define solutions by monitoring network traffic and displaying network statistics.

**To monitor the network configuration and statistics and analyze errors**

Use the Statistics application. Open a session, and from the Text Views Available box, choose Statistics.

**To analyze TCP connections**

Use the Statistics application. Start the application, open a session and, from the Text Views Available box, choose TCP and Statistics.

## 13.9 Interpreting Ping Messages

The Ping application prints messages in a standard form, as follows:

```
Ping failed: [error message]
```

The first part of the message specifies a Ping condition (`Ping failed`); the second part of the message defines the condition. Some messages have additional text that suggests a possible cause, as listed in the following table.

| Condition | Message and Interpretation |
|---|---|
| Cannot reach host | If the connection fails because the target host cannot be reached, you might receive the following message:<br>`Ping failed: host unreachable: Dest. Unreachable.`<br>There are three probable causes:<br>(1) This condition can occur if the IP router returns an ICMP message with a code of host unreachable. This normally means that the host's local network can be reached, but the target host is not available.<br>(2) Your default router does not have a router table entry for your target IP host.<br>(3) Your default router has a router table entry for your target IP host, but determines that it cannot find the optimal pathway for accessing your target IP host; your router sends your packet to the destination anyway. Your router also sends an ICMP redirect message back to the source node (your PC) that suggests the optimal routing path.<br>**Note:** This does *not* signal that an error has occurred, but indicates that you could more efficiently transmit your packets by using a different router. |
| No reply | If the other end of the connection does not reply, you might receive the following message:<br>`Ping failed: Timeout.`<br>Possible causes are<br>- The target host failed.<br>- The network path to the target host failed.<br>- The target host might be turned off.<br>- Conflict on the local PC due to incorrect configuration. |
| Packet not forwarded | If the IP router could not forward a packet and returned an ICMP message, you might receive the following message: |

| | |
|---|---|
| | `Ping failed: Network unreachable.`<br><br>The message indicates that the destination address is nonexistent or that a network link between your PC and the destination is not available. |
| TTL = 0 | If the IP time-to-live field in a datagram reaches zero before the datagram reaches the final destination, you might get the following message:<br><br>`Ping failed: got ICMP error: Time to Live exceeded.`<br><br>This condition can occur if the TTL value is less than 64. |

If you cannot establish a connection to the remote host by using its IP address, the cause of failure could be that the remote network or its gateway is not operating. Try using the Ping application to contact the remote host.

If you suspect that one of these problems is causing the connection failure, you probably cannot solve the problem alone. Contact the remote host's system administrator or the network administrator for help in resolving the problem.

## 13.10 Interpreting Telnet Server Failure

If you cannot use Telnet to connect to the remote host by hostname, there could be

- A name resolution problem.
- An error in your host table file.
- A problem with the domain name service.

Refer to the previous sections of this chapter for suggestions about how to diagnose and solve these problems.

If you cannot establish a connection to the remote host using its IP address, the cause of failure could be that

- Your remote host does not support a Telnet server.
- The Telnet server on the remote host is not running.
- The remote network or its gateway is not operating. Try using the Ping application to contact the remote host.

If you suspect that one of these problems is causing the connection failure, you probably cannot solve the problem alone. Contact the remote host's system administrator or the network administrator for help in resolving the problem.

## 13.11 Interpreting FTP Server Failure

If you cannot establish an FTP connection to the remote host, the cause of failure could be that

- The remote IP host does not support an FTP server.
- The remote host FTP server is not running.
- The remote network or its gateway is not running. Try reaching the remote host using the Ping application.
- There are other remote host problems, such as a problem on the line connecting to the remote host.

Refer to the previous sections of this chapter for suggestions about how to diagnose and solve these problems.

If you cannot use the FTP application to access a remote host, then the problem might originate at the remote host. Contact the remote host's system administrator for help in resolving the problem.

## 13.12 Troubleshooting SLIP or PPP Connections

When you use a SLIP or PPP connection, you can experience the same problems that have been discussed in the previous sections of this chapter. In addition, you might experience problems with your modem, your dial-up software, your Internet service provider, or your network dial-up.

To troubleshoot your physical connection, refer to the documentation that came with your modem. You need to be sure that

- Your modem and cables or ISDN card are connected correctly.
- The cable is plugged in to the correct serial port or the ISDN card.
- Your modem is powered up and operating.

If you are having trouble making a connection, sometimes it helps to turn the modem off and then turn it back on again.

**To use networking software to connect over a SLIP or PPP line**

Use the Dialer application to connect to a remote host over a modem or a serial interface. See the step-by-step procedures in online Help for detailed information about using the Dialer application.

The Dialer application includes an Automatic Connect feature that enables network applications to connect to the network as needed. This feature can provide significant savings in online charges from your service provider. See the Dialer application online Help for details.

The Dialer application can be used with scripts that will automate the process of making a connection. You can find a default script in the your \PCTCP subdirectory (with the extension .SCR). To learn more about scripts, see the Dialer application online Help and consult the FTP Software World-Wide Web home page at http://www.ftp.com.

# Chapter 14
# Introduction to Network Security

Network security enables you to work in a protected network environment. OnNet16 provides the following network security methods:

- Kerberos (available in OnNet16 only when distributed on CD-ROM)
- SOCKS
- IP security
- Proxy firewall

Kerberos security provides an authentication service for identifying network users. Kerberos can be used in the DOS Telnet and remote command applications, if your network uses Kerberos security.

SOCKS security uses a firewall host system to protect a network. SOCKS applications communicate with the firewall using the SOCKS protocol. When an application makes a request to connect to a host outside the firewall, the PCTCP.INI file and the SOCKS configuration file are examined to determine if the request should go through the firewall or go directly to the destination server.

IP security is configured using parameters in the PCTCP.INI file. IP security involves the IP header, which has several options that OnNet16 supports. The TCP layer interprets the Type of Service and Precedence options. For more information on IP security, see Part I, “Installing and Configuring OnNet16.”

Non-transparent proxy firewalls can be used with the Windows version of the FTP application. Proxy firewall software on the firewall host acts on behalf of the FTP client and, as an FTP proxy, accepts commands from FTP clients, such as your local PC. The proxy firewall determines whether the firewall user (or the IP address of your PC) is permitted to use the proxy, and if so, completes the connection to the remote destination.

## 14.1 Kerberos Security

Kerberos security grants credentials to users, then permits access to protected network service; that is, server programs that use Kerberos authentication. The Kerberos authentication system offers a complete security system that adapts to many open network computing environments. Kerberos provides an authentication service for identifying network users.

The **rexec** command is a type of authorization, which you choose when configuring Kerberos security. It protects your files by requiring a password.

Before you can use Kerberos security, you must configure it from the Advanced setting of the Configure application. Kerberos also requires a specific Kerberos directory on your PC to hold the Kerberos configuration files and Kerberos ticket files.

**Note:** Kerberos is available only in the CD-ROM version of OnNet16.

## 14.2 SOCKS Security

SOCKS is a security method that uses a firewall host system to protect a network from unauthorized users. The firewall system protects against intruders by providing only one point of access, which is itself very secure. A secure network prevents intruders from accessing local hosts.

SOCKS applications request a connection; the PCTCP.INI file and the SOCKS.CNF (SOCKS configuration file) are examined to see if the request should go through the firewall or if the request should go directly to the destination server.

Applications that go through the firewall use the SOCKS protocol to communicate with the firewall. If the request is sent to the firewall system, the SOCKS daemon on the firewall system validates the request and makes the connection request on behalf of the client application. The request passes through the firewall to the server on the other side of the firewall.

**To configure SOCKS security**

1. Use the Configure application to edit the `[pctcp socks]` section of the PCTCP.INI configuration file.
2. Edit the SOCKS.CNF configuration file.

For more information about SOCKS security, see Chapter 16, Configuring SOCKS Security.

# Chapter 15
# Configuring Kerberos Security

The following DOS applications in OnNet16 include a network security scheme called Kerberos:

- **rloginvt**
- **rcp**
- **rsh**
- **tar**
- **rmt**
- **tn**

Kerberos security lets you work in a protected network environment that provides an authentication service for identifying network users. If your network uses Kerberos security, you can use this security with applications that support Kerberos.

Kerberos grants credentials to users, then permits access to protected network services; that is, server programs that use Kerberos authentication. The Kerberos authentication system, designed at the Massachusetts Institute of Technology, offers a complete security system. Kerberos security also lets you protect sensitive data by encrypting data that is transferred over the network. To use data encryption, you must configure and use a command that supports this feature. Currently, OnNet supports data encryption for Telnet.

Kerberos security provides authentication of services and users. Authentication relies on the user and server information that is stored in the Kerberos master database. You can also use servers that do not support Kerberos security.

## 15.1 Before You Start Using Kerberos Security

You should ensure that

- Your network has services that support Kerberos security.
- You are registered in the Kerberos database and have a valid Kerberos username and password.
- You have Kerberos correctly configured on your PC.
- The time on the clock on your PC is synchronized with the time on the clock on the Kerberos server.

If you are uncertain about this information, contact your system administrator.

Kerberos security uses special terminology to discuss the authentication process. The term "principal" identifies Kerberos users and server clients. The following three elements identify a principal:

| | |
|---|---|
| Principal name | A unique Kerberos name. Typically, you and your system administrator determine the principal name. |
| Instance name | A label that gives more information about the principal. Typically, the instance for a user defines privileges. For general user privileges, the instance is usually set to null. For administrative privileges, the instance is usually set to admin. Instance names might be specific to your Kerberos environment. |
| Realm name | The Kerberos authentication domain. The realm identifies the group of users and services listed in the Kerberos master database. |

During the authentication process, a principal receives credentials, or "tickets," that permit access to Kerberos services, such as Telnet. Access to Kerberos services requires a "ticket-granting ticket" and a "service ticket" for a specific service. A ticket-granting ticket is an authorization that lets a principal obtain credentials for network services in the realm where the ticket was assigned. A service ticket is an authorization that lets the principal use a specific server program that supports Kerberos security.

## 15.2 Configuring Kerberos Security

Using Kerberos security requires configuration entries in the PCTCP.INI configuration file. Kerberos also requires a specific Kerberos directory on your PC to hold the Kerberos configuration files and the Kerberos ticket files.

**To configure Kerberos security**

1. Edit the PCTCP.INI configuration file.
2. Create a directory for the Kerberos configuration files and a directory for the Kerberos ticket file.
3. Create a KRB.CON file in the directory created for Kerberos files.
4. Create a KRB.REA file in the directory created for Kerberos files.

See the following sections for more detailed information about this procedure.

### 15.2.1 Editing the PCTCP.INI Configuration File

Kerberos security uses entries in the PCTCP.INI configuration file. Edit the `[pctcp kerberos]` section to use Kerberos commands.

| To configure | Use this section |
|---|---|
| The path of the directories that contain Kerberos configuration and ticket files on your system, your username for Kerberos applications, and the name of your Kerberos authentication realm | `[pctcp kerberos]` |
| Kerberos authentication for the **tn** command, including data encryption between hosts for Telnet sessions<br><br>**Note:** If a remote host is running Sandia National Laboratories' implementation of the Telnet server, you must define the service name as **`telnet`** (**tn** uses the RFC 1411-compliant service name rcmd; the Sandia implementation uses the service name telnet). | `[pctcp tn]` |

Kerberos also uses the username, domain name, and hostname defined in the `[pctcp general]` section of the PCTCP.INI file.

### 15.2.2 Creating Directories for Kerberos Files

The `config-directory=` parameter in the `[pctcp kerberos]` section of the PCTCP.INI file identifies the name of the directory that contains your Kerberos configuration files. The `ticket-directory=` parameter identifies the name of the directory that contains your Kerberos ticket files.

Configuring different directories for your Kerberos configuration and ticket files provides additional security by letting you manage your ticket files using RAMDRIVE.SYS.

**To change the directories**

1. Create directories for your Kerberos configuration and ticket files.
2. Create the KRB.CON and KRB.REA Kerberos configuration files in the configuration directory, as explained in following sections.

You can copy these files from the Kerberos master server on your network. Ask your system administrator for the location of the files. Typically, on a UNIX system, the files have the names krb.conf and krb.realms, and reside in the /etc directory.

### 15.2.3 Creating the KRB.CON File

Kerberos security uses a KRB.CON configuration file. This file contains the following information:

- Your Kerberos realm name
- The name of the Kerberos master server for your Kerberos realm
- The Kerberos authentication servers for other realms that you can access (optional)

**Note:** Entries for Kerberos names in the KRB.CON file are case sensitive. Typically, realm names appear in uppercase letters.

The first line of the KRB.CON file must identify the name of the default local realm. Additional lines identify a realm name followed by the hostname of a Kerberos server. The following sample KRB.CON file defines configuration information required for a user in the XYZ.COM realm:

```
XYZ.COM
XYZ.COM mach1.xyz.com admin server
DEV mach2.xyz.com
XYZ.COM mach3.xyz.com
```

| This line | Identifies |
|---|---|
| `XYZ.COM` | The default Kerberos realm as `XYZ.COM`. |
| `XYZ.COM mach1.xyz.com admin server` | The Kerberos administrative server for the realm `XYZ.COM` as `mach1.xyz.com`. |
| `DEV mach2.xyz.com` | The authentication server as `mach2.xyz.com` for the `DEV` realm, where `DEV` is the name of another realm. Using services on the DEV realm requires this entry. |
| `XYZ.COM mach3.xyz.com` | Secondary servers used for authentication in the default Kerberos realm `XYZ.COM`. Kerberos tries to contact secondary servers if the primary Kerberos server cannot be contacted. |

### 15.2.4 Creating the krb.rea File

Kerberos security also uses a KRB.REA file. The KRB.REA file translates a Kerberos realm name from a hostname. Kerberos requires the presence of this file, even if it is empty, to run applications.

Lines in the KRB.REA file appear in two formats, as follows:

| To define | Use this format |
|---|---|
| A Kerberos realm for a specified host. | *hostname kerberos_realm* |
| A Kerberos realm for a host in the specified domain. Kerberos uses this translation rule if it cannot find a hostname entry, but can match the domain name. All entries in this form begin with a dot (.). | **.***domain_name kerberos_realm* |

**Note:** Kerberos realm names are case sensitive. Internet domain names are not case sensitive. Typically, realm names appear in uppercase letters.

The following sample KRB.REA file defines information about the Kerberos realm. The first line establishes the translation rule, and lines two and three are exceptions to that rule.

```
.XYZ.COM KERB_REALM_1
mach1.xyz.com KERB_REALM_2
mach2.xyz.com KERB_REALM_3
```

| This line | Defines |
|---|---|
| `.XYZ.COM KERB_REALM_1` | Internet domain names in the domain `.XYZ.COM` as belonging to the Kerberos realm `KERB_REALM_1`. |
| `mach1.xyz.com KERB_REALM_2` | `KERB_REALM_2`, rather than KERB_REALM_1, as the default realm `for mach1.xyz.com`. |
| `mach2.xyz.com KERB_REALM_3` | `KERB_REALM_3`, rather than KERB_REALM_1, as the default realm for `mach2.xyz.com`. |

## 15.3 Obtaining Kerberos Credentials

Before you can use services that support Kerberos security on your network, you need to obtain your Kerberos credential. Use the **kinit** command to get this credential, the ticket-granting ticket. This ticket lets you obtain service tickets for connections to services (that is, programs such as Telnet) that support Kerberos security on your network.

The ticket-granting ticket expires in 8 hours (or the time that you specify with the **kinit -l** option, or the time that your network administrator specifies). During the time that the ticket-granting ticket is active, you can use Kerberos commands. After this ticket expires, you must obtain a new ticket-granting ticket with **kinit** to use Kerberos services.

### 15.3.1 Understanding the Ticketing Process

Entering the **kinit** command at your PC starts the ticketing process. This process enables you to use network services that support Kerberos security. The following steps summarize how Kerberos security provides ticket-granting tickets:

1. The **kinit** command prompts you for your Kerberos username. After you enter this username, the **kinit** command sends a request that contains your username and the name of a special service known as the “ticket-granting service” to the authentication server.

2. The authentication server verifies if you are a registered client in the database. If you are registered in the database, Kerberos security generates a random “session key” (a temporary encryption key) that it uses later between the client (your PC) and the ticket-granting server. It also creates a ticket that contains the following information for the ticket-granting server:

   – Your name
   – The name of the ticket-granting server
   – The current time
   – The ticket’s expiration time (lifetime)
   – Your PC’s Internet address
   – A new random session key

3. Kerberos security encrypts the preceding information in a key known only to the ticket-granting server and the authentication server.

4. The authentication server then sends the ticket, along with a copy of the random session key and some additional information, back to your PC. Kerberos security encrypts this response in the client’s “private key,” a number known only to Kerberos security and to the client.

5. Kerberos security prompts you for your password after the client receives the response. It then converts your password to a data encryption service (DES) key, which Kerberos security uses to decode the response from the authentication server. Kerberos security stores the ticket and the session key, along with some of the other information for future use, and erases your password and DES key from memory.

After this exchange, the PC possesses the information that it needs to prove the identity of its user for the lifetime of the ticket-granting ticket. As long as no one previously tampered with the software on the PC, no information exists that lets another user impersonate you and use the system when the system does not have a valid ticket-granting ticket.

### 15.3.2 Obtaining and Listing Kerberos Tickets

To start working in a Kerberos secure environment, use the **kinit** command to get your ticket-granting ticket. Verify that you have this ticket, as well as any service tickets, by using the **klist** command.

**To obtain a ticket-granting ticket**

1. At your system prompt, enter the **kinit** command.
2. If prompted, enter your Kerberos username.
3. At the `Password` prompt, enter your Kerberos password.

The system returns a message that verifies the success or failure of **kinit**. Use the **klist** command to verify that you have a ticket-granting ticket.

**Note:** Although your ticket expires in 8 hours, or the time that you specify with the **kinit** command, use the **kdestroy** command to delete Kerberos tickets before you leave your PC unattended. See Section 15.5, Destroying Kerberos Credentials for more information.

The following example shows a user with the Kerberos username chris logging in to the Kerberos authentication system. The user enters the Kerberos password at the prompt. The screen does not display passwords.

```
C:\>kinit

Kerberos initialization
Kerberos name: chris
Password:
```

**To list your Kerberos tickets**

At your system prompt, enter the **klist** command.

The **klist** command lists the following information about currently held Kerberos tickets:

- The pathname of the ticket file.
- The identity of the principal that owns the ticket.
- The issuance and expiration times for each ticket.
- The principal names of all Kerberos tickets that you currently hold. The system lists principal names in the form *name.instance@realm*. It does not display the dot (.) if the instance is null.

Kerberos requires that you have a Kerberos ticket-granting ticket (that is, a ticket with the principal name `krbtgt`) before you obtain service tickets for applications that support Kerberos security.

In the following example, the **klist** command verifies that user chris has a ticket-granting ticket `krbtgt`:

```
C:\>klist

Ticket file: C:\pctcp\etc\chris.tkt
Principal: chris@XYZ.COM
Issued            Expires            Principal
Jan 10 09:21:57   Jan 10 17:21:57    krbtgt.XYZ.Com@XYZ.COM
```

The fully qualified name is `chris@XYZ.COM`. The instance is null.

In the following example, the **klist** command verifies that a Kerberos ticket does not exist for user chris:

```
C:\> klist

Ticket file: C:\pctcp\etc\chris.tkt
klist: Kerberos: No ticket file.
```

### 15.3.3 Changing Your Kerberos Password

You can change your Kerberos password for the Kerberos server by using the **kpasswd** command from your PC:

```
C:\>kpasswd
```

The command prompts you for your current password and for a new password.

## 15.4 Using Commands That Support Kerberos Security

The commands that support Kerberos security provide options to specify Kerberos authentication, and to define a remote realm. To use Kerberos options for these commands, you need an active Kerberos ticket-granting ticket. If you do not have this ticket, use the **kinit** command to obtain one. The following list summarizes the commands that support Kerberos authentication:

| To do this | Use this command |
|---|---|
| Connect your PC to a remote host. | **rloginvt** |
| Copy files between computers. | **rcp** |
| Connect to a specified host and execute a specified command. | **rsh** |
| Archive and restore files between local and remote files, disks, and tapes. | **tar** |
| Control a remote tape drive. | **rmt** |
| Log in to a remote host over a Telnet connection. | **tn** |

If you use Kerberos Security for Telnet connections, you can configure Telnet to support data encryption between hosts.

## 15.5 Destroying Kerberos Credentials

When you complete your work on Kerberos servers, delete active Kerberos tickets to prevent other users from using your PC for services that require Kerberos authentication. Use the **kdestroy** command to delete your Kerberos tickets when you leave your PC unattended. The **kdestroy** command deletes a ticket-granting ticket and all service tickets. You cannot recover tickets that you delete with the **kdestroy** command. Tickets that you do not destroy expire at the expiration time set by the **kinit** command. Use the **klist** command to verify that you deleted your Kerberos tickets.

**To delete Kerberos tickets**

At your system prompt, enter **kdestroy**

The system deletes any active Kerberos tickets. Without a Kerberos ticket, you cannot access services that support Kerberos security. In the following example, the **kdestroy** command ends a Kerberos session by destroying an active ticket-granting ticket:

```
kdestroy

Tickets destroyed.
```

# Chapter 16
# Configuring SOCKS Security

SOCKS is a security method that allows you to control communication between one network and another. With SOCKS, a client must go through a “firewall” host system to communicate with a server in another network. A firewall system is a host or server that has only one gateway through which applications can send and receive data. The FTP Software version of SOCKS supports Windows TCP client (but not DOS) applications.

When SOCKS security is enabled, the FTP Software version of WinSock DLL (which you get when you install the FTP Software kernel) calls the SOCKS DLL anytime a Windows TCP client application requests a connection to a remote host. The SOCKS DLL uses the SOCKS.CNF configuration and the `[pctcp socks]` section of the PCTCP.INI file to determine how to handle the request. Based on the contents of these files, the request is either sent directly to the remote host or to the firewall.

If the request must go through the firewall, the SOCKS DLL passes the request to the SOCKS server (on the firewall) to establish the connection. If the request is valid, the SOCKS server establishes the connection. Otherwise, the SOCKS server rejects the request.

When a requested connection is made through a firewall, the SOCKS server makes the request on behalf of the client application (as a “proxy”). Thus a SOCKS server is referred to as a “proxy server.”

**Note:** If you are using OnNet16 with another vendor’s TCP/IP stack and WinSock DLL, applications will need built-in SOCKS support to traverse a firewall system using SOCKS. The OnNet16 application with built-in SOCKS support is FTP. Configure SOCKS security for FTP Client from the FTP Client itself. Also, see Section 16.2, Configuring Your PC for SOCKS Security.

## 16.1 Before You Start Using SOCKS Security

You should ensure that

- Your network uses a firewall system that supports the SOCKS protocol.
- You have SOCKS correctly configured on your PC.

Contact your network administrator for more information. The network administrator might alter the SOCKS daemon configuration; you will need the updated SOCKS.CNF file for your PC configuration.

## 16.2 Configuring Your PC for SOCKS Security

Every time a SOCKS client has to make a network connection, the client compares the pending request with the SOCKS.CNF file, one line at a time. When it finds a line with conditions that are matched by the request, the action specified on that line is taken. The remaining lines of the SOCKS.CNF file are skipped. So the order of the lines in the file is extremely important.

A SOCKS configuration file (SOCKS.CNF) resides on your PC and determines which destination addresses (hostnames) you can connect to with SOCKS through the firewall, and which destinations you must go directly to the server to access. When configuring your SOCKS configuration file, you must

- Identify your SOCKS server.
- Identify destination addresses that will use the SOCKS protocol for their applications to communicate with the firewall.

### 16.2.1 SOCKS Configuration File

The SOCKS configuration file (SOCKS.CNF) is located on your PC and is read each time a connection is requested, if you have enabled SOCKS. If your PC does not have a SOCKS.CNF file, you have the following choices:

- Get the SOCKS.CNF file from your network administrator.
- Create your own SOCKS.CNF.

**To create the SOCKS.CNF file**

1. Create the lines of the file using the following syntax. (For an explanation of each syntax line, read the remaining steps that follow and the Examples section in this chapter.)

   **direct** *dst_addr dst_mask* [*op dst_port*]

   **sockd** [**@= *server***] *dst_addr dst_mask* [*op dst_port*]

where

**direct** Indicates to use a direct connection (and bypass SOCKS).

**sockd** Indicates when to use a SOCKS connection and, optionally, which SOCKS server to connect to.

*dst_addr dst_mask* Specifies the destination IP address or the range of destination IP addresses. Both addresses are given in the usual dotted form (for example, 128.127.51.14). Bits in *dst_mask* that are set to 0 indicate the bit positions that should be masked off (that is, ignored) during comparison of *dst_addr* and the actual destination IP address. So, specifying 255.255.255.255 in *dst_mask* demands an exact match with *dst_addr*. The value 0.0.0.0 in *dst_mask* causes an address match no matter what is specified for *dst_addr*.

*op* Must be eq, neq, lt, gt, le, or ge, for the condition of equal, not equal, less than, greater than, less than or equal, and greater than or equal, respectively.

*dst_port* Can be either a port number (for example, 23) or the equivalent service name, as specified in the WINSOCK SERVICES file (for example, telnet for port number 23).

If the *op dst_port* pair is omitted, the line applies to all services.

**@= *server*** Indicates the server IP addresses (in the dotted form). (This is an optional field.)

The server, which is used only in a **sockd** line, consists of a SOCKS proxy server, which the client program should try to use for establishing a proxy connection for the defined destinations. Domain names of the servers can be used, although it is probably more prudent to specify IP addresses. If this field is omitted, the client program uses the default SOCKS proxy server, which is determined by the `socks-server=` parameter in the `[pctcp socks]` section of the PCTCP.INI file.

Create the lines in order of importance. The lines in the SOCKS.CNF file are interpreted sequentially until a match for the current connection is found. Therefore, the order of the lines is extremely important. Case is also important. Lines beginning with # are comments.

2. Place the SOCKS.CNF file in the directory specified by the `etc-dir=` parameter in the `[pctcp general]` section of the PCTCP.INI file. Alternately, you might place the SOCKS configuration file in another directory and specify the full pathname in the `config-file=` parameter of the `[pctcp socks]` section.

**Note:** This implementation of SOCKS does not implement userlists or the **deny** line (the **deny** line is like the **direct** line only it denies the action specified be done) in the SOCKS configuration file, as these concepts do not make sense on a PC. Userlists or userfiles in the SOCKS.CNF file are ignored. A line that begins with **deny**is treated like a line that begins with **direct**.

**Examples**

For example, to match the condition indicated in the following line, the destination IP address must be 128.127.52.14 exactly, and the service requested must be telnet. In that case, connection to host 128.127.52.14 should be done via a SOCKS proxy server on host 128.127.55.4.

```
sockd @=128.127.55.4 128.127.52.14 255.255.255.255 eq telnet
```

The following line defines all hosts in a local subnet as direct. The host IP address is `128.127.50.0`.

```
direct 128.127.50.0 255.255.255.0
```

This assures that network requests to the host itself are processed locally rather than being diverted through a SOCKS server. The IP address of the local machine or the subnet should be defined next as a direct connection.

The following example shows how to allow the REXEC application to bypass SOCKS and directly initiate a connection with a remote system whose IP address is 128.127.50.50:

```
direct 128.127.50.50 255.255.255.255 eq exec
```

The following example shows how to use a well-known, established port number in place of a service name:

```
direct 128.127.50.50 255.255.255.255 eq 512
```

### 16.2.2 PCTCP.INI SOCKS Configuration

The `[pctcp socks]` section in the PCTCP.INI configuration file stores configuration information for SOCKS, which is read when an application makes a request for a connection.

The following is an example of the `[pctcp socks]` section of the PCTCP.INI file:

```
#--- sample configuration entries ----

[pctcp socks]

socks-server = firewall.xyz.com

error-log = c:\3.0\socks.log

use-socks = yes

error-display = yes

#-----------------------------------------------
```

The following lists and explains each entry in the `[pctcp socks]` section of the PCTCP.INI file:

`socks-server = [hostname | ip_address]`

This parameter indicates the hostname or IP address of the default SOCKS server. Use this entry if the entries in the SOCKS.CNF configuration file do not specify a SOCKS server.

`error-log = drive:\path\filename`

This parameter indicates the full pathname of the error log file. If `error-log` is not defined, the file is created as C:\ETC\SOCKS.LOG. If the `error-log=` parameter is defined as blank, an error log is not used. This parameter is optional.

`use-socks = [no | yes]`

This parameter specifies whether applications should use SOCKS.DLL. This parameter is required.

Default: no

`error-display = [yes | no]`

This parameter specifies whether SOCKS.DLL displays message boxes when serious errors occur. This parameter is optional.

Default: yes

`config-file = drive:\path\filename`

This parameter defines the full pathname of the SOCKS configuration file. If `config-file` is not defined, SOCKS searches for a file named SOCKS.CNF in the directory specified by the `etc-dir =` parameter in the `[pctcp general]` section. This parameter is optional.

**To create a `[pctcp socks]` section**

1. Add a hostname or an IP address to the `socks-server=` parameter (for example, firewall.xyz.com).
2. Set `use-socks = yes`
3. Add a pathname to the `error-log=` parameter, indicating where the log resides (for example, C:\3.0\SOCKS.LOG). This parameter is optional.
4. Specify whether SOCKS.DLL will display message boxes by indicating `yes` or `no` in the `error-display=` parameter. This parameter is optional.
5. Specify the full pathname of the SOCKS configuration file in the `config-file=` parameter if SOCKS.CNF does not exist in the directory specified by the `etc-dir=` parameter in the `[pctcp general]` section.

## 16.3 Using X Applications on Systems with SOCKS Security

X applications cannot use the FTP Software SOCKS security to connect to remote hosts. If SOCKS is enabled, you must include entries in the SOCKS.CNF file to allow X applications to make connections directly. For example, if the remote client that an X application is going to connect to has the IP address 128.127.50.23, type the following line into your SOCKS.CNF file:

```
direct 128.127.50.23 255.255.255.255
```