Remote Office Client for NetRider

PC/TCP OnNet Applications Use

Part Number: AA-QQAEA-TE

November 1995

Revision/Update Information:	This is a new document.
Software Version:	Remote Office client for NetRider V2.2 for MS-DOS and Windows
Operating System:	MS-DOS Version 5.0 minimum Windows Version 3.1 minimum
Reader's Comments:	Your comments and suggestions help us improve the quality and usefulness of our documentation. You can respond to the documentation manager at: **Email:** doc_quality@lkg.mts.dec.com **FAX:** 508 486 5655

November 1995

Contents

4 OnNet MS-DOS Applications

5 Configuring the Kernels

6 Configuring NetBIOS

A PC/TCP Configuration Parameters Reference

B Command Reference

Preface

Overview

Purpose

This book explains how to use the PC/TCP OnNet applications that are supplied with the Remote Office Client for NetRider for MS-DOS and Windows.

Intended Audience

This book is written for individuals who need to use the PC/TCP OnNet applications. This book assumes that the reader has installed and is familiar with the use of the Remote Office Client for NetRider for MS-DOS and Windows. The book also assumes that readers are familiar with the other hardware and software installed on their personal computer.

Conventions

This book uses the following conventions:

Convention	Description
boldface text	Boldface text represents the name of a button, icon, or menu option.
`monospaced text`	Monospaced text represents commands that you type.
italic text	Italic text in a command indicates variable information that you provide.

Associated Documents

Contact your Digital reseller or Digital representative to order additional documentation. The following documents can be of use:

Book	Description
NetRider Client for Macintosh Installation and Use	Explains how to install and use the NetRider Remote client in a Macintosh environment.
Remote Office for NetRider MS-DOS Installation and Use	Explains how to install and use the Remote Office Client in an MS-DOS environment.
Remote Office for NetRider Windows Installation and Use	Explains how to install and use the Remote Office Client in a Windows environment.
Remote Office Client for NetRider RCHost Installation and Use	Explains how to install and use the Remote Office Client RCHost remote control program with an IPX network.
NetRider Remote Access Server Quick Start	Explains how to install the NetRider Remote Access Server, the Digital Network Access Server software, and automate the PC client installation.

Book	Description
DECserver Network Access Software Installation	Explains how to install the network access software on your operating system.
DECserver Owner's Manual (90 or 900 series)	Explains how to install and operate the DECserver hardware.

Chapter 1

Introduction to OnNet with Remote Office

Overview

In This Chapter

This chapter describes the following:

- ROSTART.BAT for OnNet
- Choosing a kernel

ROSTART.BAT for OnNet

Introduction

The Remote Office install procedure creates a file called ROSTART.BAT in the Remote Office directory. This BAT file may be used to load the Remote Office client. If you wish, the modules loading ROSTART.BAT can be included in another BAT file.

ROSTART.BAT illustrates how to run Remote Office Gold using both the DOS TSR (terminate-and-stay-resident program) version, and the Windows VXD versions of FTP's PC/TCP product.

OnNet always uses a Packet Driver interface for TCP connectivity. This is possible even in multi-protocol environments because the Remote Office kernel can support multiple MAC layer interfaces concurrently. This gives the flexibility, for instance, to load an ODI driver for NetWare while simultaneously loading a packet driver for OnNet. See the OnNet & NetWare configuration that follows for an example of the load order.

If You Select OnNet

If you select OnNet as your LAN OS, the following ROSTART.BAT is created:

ROFFICE	**{ASYNCH KERNEL }**
PKROF-E	**{PACKET DRIVER }**
ETHDRV or VXDINIT	**{FTP KERNEL / VXD Interface }**
WIN WROCLNT	**{LOAD WINDOWS / DIALER }**

If You Select OnNet and NetWare

If you select OnNet and NetWare as your LAN OS, the following ROSTART.BAT is created:

ROFFICE	**{ASYNCH KERNEL }**
LSL	**{ODI LINK SUPPORT LAYER }**
ODIROF-E	**{RO ODI DRIVER FOR IPX }**
IPXODI	**{NETWARE IPX PROTOCOL }**
PKROF-E	**{PACKET DRIVER }**
ETHDRV or VXDINIT	**{FTP KERNEL / VXD Interface }**
VLM	**{NETWARE REDIRECTOR }**
WIN WROCLNT	**{LOAD WINDOWS / DIALER }**

Addresses

During the Remote Office installation, you are asked to provide an Internet address and a Domain Name Server address. These addresses are used as defaults when building phonebook records, and can be modified with different addresses for various phonebook records at any time.

After the Remote Office Client is assigned an IP address at PPP negotiation time, the Remote Office kernel will dynamically set the IP address for the TCP kernel as well as the Domain Name Server (DNS) address configured for the phonebook record used to establish the connection.

In order for the IP address and DNS address to get dynamically configured, the following conditions must be met:

Logging in using RO command line or ROCLIENT:

1. ETHDRV must be loaded.

Logging in using Remote Office Windows Utility (WROCLNT):

1. ETHDRV or VXDINIT must be loaded.
2. PCTCPAPI.DLL must be found in your PATH or the Windows/system directory.
3. WNET386.DLL must be found in your PATH or the Windows/system directory.
4. If VXDINIT is not loaded, VPCTCP.386 must initialize properly.
5. If VXDINIT is loaded, VXDPCTCP.386 must initialize properly.

Remote Office does not change FTP's configuration file PCTCP.INI. The IP addresses are changed in memory only.

Choosing a Kernel

Description

OnNet PC/TCP provides both VxD and TSR kernel architectures. The type of OnNet PC/TCP kernel architecture is transparent to the PC/TCP applications. The installation program automatically configures the appropriate kernel for your operating environment.

- If you operate mainly in a Windows environment, use the VxD kernel and Windows applications.
- If you work primarily in DOS, if you need to interoperate with TSRs (or third-party applications that rely on the TSR kernel), or if you need network connectivity with TSRs that you load before you start Windows, use the ETHDRV TSR kernel and DOS based applications.

The PC/TCP TSR kernel is required if you run DOS only. Windows users can choose between the TSR and the VxD kernels.

NETBIOS Considerations

If your system is configured to use the VxD kernel and NetBIOS, run the vxdinit command prior to loading windows.

Before you start Windows, create an entry in your PCTCP.INI file for starting the NetBIOS VxD.

```
[pctcp vxdinit]
   vnbep=yes
```

You can override the PCTCP.INI parameter setting by using the following switch on the VXDINIT command line to automatically load the VXD netbios driver.

```
VXDINIT +n
```

Chapter 2

Introduction to PC/TCP Applications in Windows

Overview

In This Chapter

This chapter contains the following information:

- Product description
- Getting started using PC/TCP windows applications

Product Description

With PC/TCP® Network Software and PC/TCP® OnNet™ applications for Windows, you can take advantage of a variety of services on your network. A network is a system of interconnected computers, also known as "hosts."

With applications in the Remote Office Gold group, you can communicate with other users, share files, and find other users and hosts on the network. Because you are working in Windows, you can simultaneously perform tasks on network hosts and on your PC.

PC/TCP software also includes a library for use by any application written to the standard Windows Sockets application programming interface. If you have applications that rely on this interface, you can run them using the Windows Sockets support that is supplied automatically with PC/TCP.

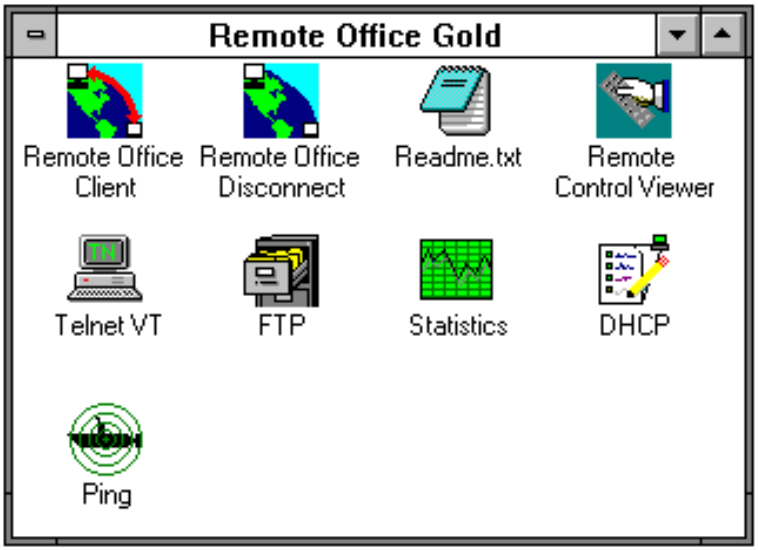
OnNet With DOS

You can use any of the PC/TCP DOS-based applications from a DOS session in Windows. You can add these applications by creating a program item for them or by executing a command at the DOS prompt. The DOS applications offer many of the same functions as the Windows applications, except that they run in a DOS session.

Getting Started Using PC/TCP Windows Applications

Overview

Installing Remote Office installs the PC/TCP Applications in the Remote Office Gold group in Windows Program Manager. The following illustration shows a WinApps program group with Reference Desk installed:



Application Description

You can use applications in the Remote Office Gold program group to perform a variety of networking tasks. Refer to the following list to decide which applications you want to use.

Use this application:	To do this:
 FTP	Transfer files between remote hosts and your PC, and transfer files between two remote hosts. For more information, refer to the online help for FTP.
 Ping	Test the availability of network hosts and network gateways, or trace the path that a message takes from one host to another. For more information, refer to the online help for Ping.
 Statistics	Display information about your PC/TCP configuration, view network configuration information for your PC, and monitor network traffic. For more information, refer to the online help for Statistics.
 Telnet VT	Log in to a remote host that uses DEC VT terminal emulation. For more information, refer to the online help for TNVT.
 DHCP	Send DHCP or BOOTP request to configure the TCP stack.

Using Online Help for PC/TCP in Windows

The online help for PC/TCP applications gives you quick access to information about the task that you are performing. You can display online help by choosing the Help buttons in the dialog boxes, or by selecting Help on the menu bar. You can also use the help search facility to look for topics.

The help for each application:

- Provides a brief introduction to the application.
- Describes how to use each dialog box and each menu.
- Describes the steps that you need to follow to complete a task.

Viewing Information

To view general information about an application:

- From an application Help menu, choose Contents.
- Select the topic under Get Started that explains how to use the application.

Displaying Dialog Box Help

To display help for a dialog box where you are working, choose the Help button. The help topic provides the information that you need to complete entries in that dialog box.

Working with PC/TCP Applications in Windows

To start a PC/TCP Windows application, choose or double-click an application icon. You can use the mouse or keyboard shortcuts to work in a WinApps application. You can exit from an application by choosing Exit from a File or Session menu, or from an icon bar.

PC/TCP Windows applications use the status bar at the bottom of dialog boxes to provide information about screen options that you select. Some applications display additional information in the status bar, such as the state of the application or of a connection.

You can customize how you use an application by choosing options on the application's Settings menu. Use the options on this menu to display or hide an icon bar, display or hide a status bar, or select preferences particular to an application, such as choosing fonts for FTP or Dialog.

You can simultaneously perform many networking tasks with PC/TCP Windows applications. You can make multiple connections with a single application by starting several instances of the application. For example, you can run more than one instance of TNVT and use each instance to connect to a different remote host. FTP works the same way.

Using Sessions

Many of the PC/TCP applications for Windows support “sessions.” A session is the set of interactions between your system and a remote host during a connection. Applications that use sessions let you customize and save information about a session so that the application can use the same information for future connections. In this manual, the term session refers to the application settings that you establish for a remote connection.

You can create session definitions for several remote hosts and then choose which session or sessions you want to use when you start the application. You also can change characteristics of an existing session and save the changes. The application that you are using determines the information saved in a session definition.

FTP and TNVT save session definitions to a file named SESSION.INI in the directory that contains your PC/TCP files. Because these applications share the same session definitions, when you create a new session, the session becomes available to both applications. Likewise, when you delete a session, it is no longer available to any of these applications.

Customizing Application Startup

Using Windows Program Manager, you can configure some PC/TCP Windows applications to automatically connect to or contact a remote host when you choose the application’s icon. You can do this by specifying options on the startup command line in the Program Item Properties dialog box. Command line options also can establish other behavior or specify characteristics that go into effect every time that you start the application. For details about the command line options that each application supports, see the online help for that application.

Customizing An Application

Sometimes it is convenient to customize an application so that you can use several sessions simultaneously. To do this, create and save the session definitions, then create a new icon for each session that you want to use. For example, you can create several TELNET icons and customize each one to connect to a different host.

Customizing An Icon

To customize an existing icon, do the following:

Step	Action
1	Select the icon for the application that you want to customize.
2	From the File menu in Program Manager, choose Properties.
3	In the Command Line box, specify the name of the Windows application executable file, followed by any command line options that you want to use.

Examples

- To log in with WTNVT to a VAX mainframe for which you previously defined a session named VAX, you can enter the following command line:

  ```
  wtnvt.exe -S VAX
  ```

  You can change the name of the icon to reflect how you have customized it. For instance, you might rename the icon in the previous example LOGIN_VAX.

- To connect to an FTP server for which you have a previously defined session named ENGR_FTP, you can enter the following command line:

  ```
  WFTP.EXE  -s  FTP_ENGR
  ```

  Again, you can customize the name of the ICON. For instance, you might rename the icon in the previous example FTP ENGR.

Chapter 3

OnNet Windows Applications

Overview

In This Chapter

This chapter describes the following PC/TCP Windows applications:

- PING
- Statistics
- FTP
- Telnet VT
- DHCP

PING

Introduction

Use PING to verify that your PC can contact a host on the network. If you receive a reply to a Ping request, you know that a network path between your PC and the host is open and that the host is active. If a Ping request fails, a message box provides information about the failure. You can also use Ping to help you isolate the source of network problems.

Get Hostname Or IP Address

Before you try to contact another computer on the network with Ping, you need the hostname or IP address of that computer.

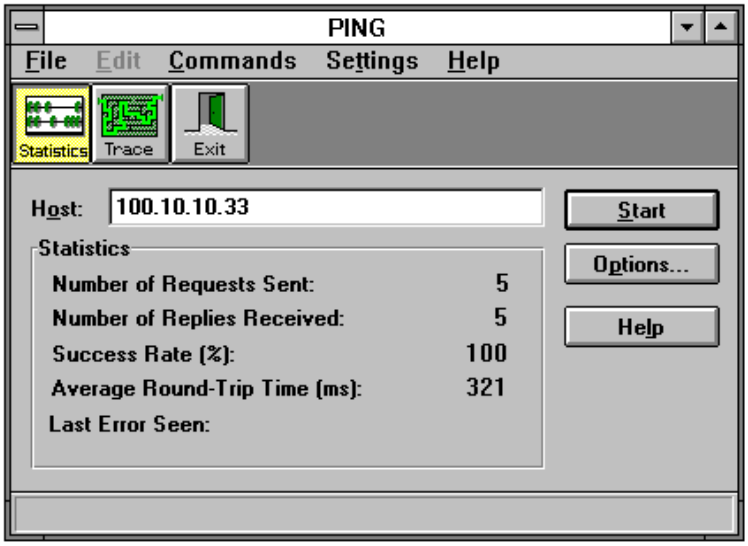
Procedure

To test the accessibility of a remote host, do the following:

Step	Action
1	Choose the Ping icon.
2	From the Commands menu, or from the icon bar, choose Statistics.
3	In the Host box, type the name or IP address of the host.
4	Choose Start to send a request for a reply from the remote host.

Example

The following example resulted from a series of ping requests to the host 100.10.10.33. The statistics indicate that all of the ping requests were successful.



You also can use Ping to view the path that a Ping request takes to get to another host. If you have difficulty connecting to a remote host, using the Traceroute command lets you see where the network path fails.

To display the path that a Ping request takes to a remote host, do the following:

Step	Action
1	Do one of the following: a) From the Commands menu, choose Traceroute. b) From the icon bar, choose the Trace icon.
2	In the Host box, type the name or IP address of the host.
3	Choose Start.

More Information

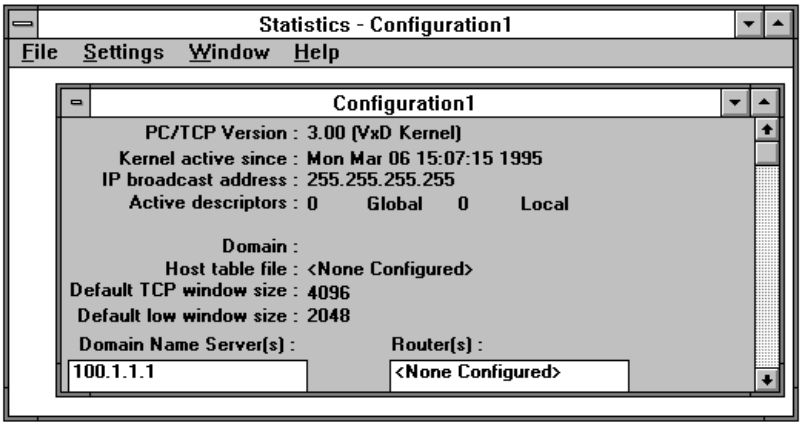
For additional step-by-step procedures, refer to the application's online Help.

Statistics

Introduction

Use Statistics to display information about the network configuration for your PC. If you have difficulty connecting to remote hosts, you can find out if you have correctly configured the IP addresses of your domain name servers that convert hostnames to IP addresses, and the IP addresses of the computers (called routers or gateways) that connect your network to other networks.

The following illustration shows configuration displays in Statistics:



Description

Statistics lets you display information in a text format and in a graphical format. To monitor several kinds of data at the same time, you can open, view, and minimize multiple graphs and information displays. With the graphical displays in Statistics, you can set the colors, graph type, and data collection method for each graph. Setting different colors for each graph lets you easily distinguish these graphs from each other on your PC.

Advanced users can use the more sophisticated features of Statistics to monitor network traffic and identify local network problems.

More Information

For additional step-by-step procedures, refer to the application's online Help.

FTP

Introduction

The FTP (File Transfer Protocol) application transfers files between your PC and another computer on the network, or between two remote hosts. You can also rename, view, and delete files that reside on other computers and on your PC. Use FTP if you do not know the exact location of the files that you want to transfer, or want to perform file transfer tasks other than just copying files.

Use FTP to work with files over the network. Sessions in FTP let you manage the connections between your PC and other computers. Using FTP to copy files from one computer to another is similar to using Windows File Manager to copy files from one directory to another.

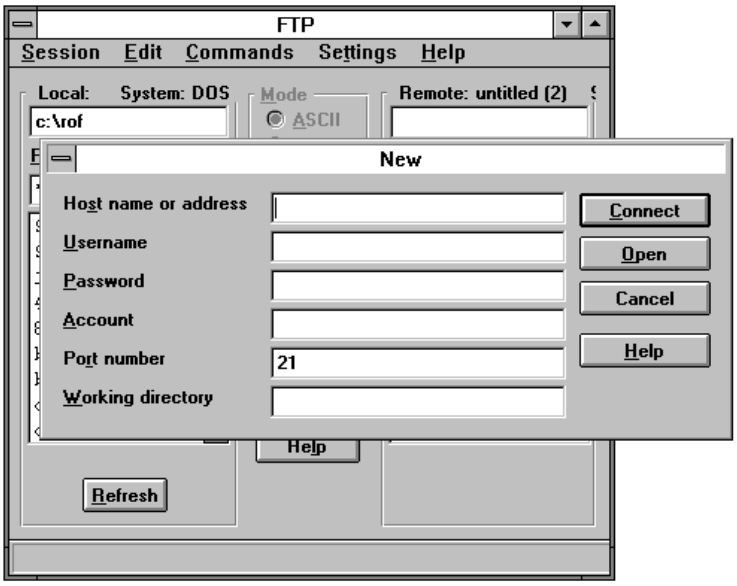
Using FTP

To use FTP, do the following:

Step	Action
1	Open a session to a remote host.
2	Customize the session.
3	Set the file transfer mode.
4	Perform your file transfer tasks, such as copying, viewing, renaming or deleting files.
5	Close the session.

Opening an FTP Session

When you start FTP, the FTP main window appears. The first time that you use FTP, the New dialog box appears with the FTP main window. The New dialog box lets you define session information for the remote host that you want to access for file transfers.



Previously Saved Sessions

If you previously saved sessions for TNVT or FTP, the Open dialog box appears with the FTP main window when you start the application. You can connect to remote hosts listed in the Open dialog box. OnNet Applications configures FTP and TNVT to share the same session information, so that the same sessions are available to both applications.

Defining a New Session

To define a new session and connect to a remote host, do the following:

Step	Action
1	Do one of the following: a) Use the New dialog box that appears when you start FTP. b) From the Session menu, choose New to display the New dialog box.

Step	Action
2	In the Hostname or Address box, type the name or Internet Protocol (IP) address of the remote host that you want to connect to for your file transfers.
3	Type the username, password, and account information that was established for you on the remote host. If you are logging in to a remote host that supports anonymous FTP, you use anonymous as your username, and usually enter your username or e-mail address as the password.
4	If you want to connect to a specific directory on the remote host, type that directory path in the Working Directory box.
5	Choose Connect.

Open Previous Session

To open a previously defined session, do the following:

Step	Action
1	From the Session menu, choose Open.
2	Under Sessions Available, select a session.
3	Choose Connect.

Customizing an FTP Session

The Settings menu lets you customize how you use FTP. You can use the FTP default settings, but should set the host type (remote operating system) for each session. Setting the host type ensures that FTP correctly displays the directories and filenames as they appear on the remote host.

To set the Host Type, do the following:

Step	Action
1	From the Settings menu of the FTP main window, choose Host Type.
2	Under Session Name, select a session.
3	Under Operating Systems, select the host operating system.
4	Choose OK.

If you want to customize other aspects of the session, choose other options from the Settings menu. Refer to the FTP online Help for an explanation of these options.

Sending and Receiving Files

After you establish and customize a session, you are ready to work with files and directories over the network. In FTP, you can list and change both local and remote directories to locate files in preparation for file transfers.

When you transfer files with the FTP application, a copy progress bar shows the percentage of files transferred. If you copy large files (over 6.3 MB) the percentage increments correctly, and the file is transferred, but the black bar disappears and then a while later starts again at the far left of the bar frame.

If you transfer files from a UNIX host, FTP includes newline and formfeed characters in this calculation. Due to the addition of these characters, a transfer may continue after the graph indicates that 100% of the files have been transferred.

Canceling a copy of a very large file can take several minutes.

To change directories on the local or remote system, do the following:

Step	Action
1	From the Commands menu of the FTP main window, choose Change Directory.
2	In the Directory Name box, type the name of the directory.
3	Under Current Sessions, select the host for that directory.
4	Choose OK.

You can also change directories by double-clicking a directory name in a file listing box of the FTP main window.

The FTP main window displays the contents of the current remote and local directories.

Copying Files

To copy files, do the following in the FTP main window:

Step	Action
1	Do one of the following: a) Select the name of the file in one of the file lists, and choose the Copy button. b) Drag and drop files: • From Windows File Manager to a remote directory displayed in the FTP main window. • From Windows File Manager to the minimized FTP icon (files are copied to the current remote directory). • From a remote directory to your local directory, and from your local directory to a remote directory. The FTP main window displays these directories. • From a remote directory on one system to a remote directory on another system if you connect to two remote hosts with FTP. The FTP main window displays these directories.

Viewing Files

If you want to see what a file contains, but do not want to copy the file to your PC, you can use FTP to view a file's contents.

To view the contents of a file, do the following:

Step	Action
1	In a file listing on the FTP main window, select a file.
2	Choose the Show button.

Windows Notepad, or a text editor that you selected, displays the file contents.

Changing the Text Editor

To change the text editor that FTP uses, do the following:

Step	Action
1	From the Settings menu, choose Editor for Show.
2	Select the path and filename of the executable file for your editor.

Using a Command File

If you frequently use FTP to transfer the same files between computers, you can create a command file (sometimes called a “take” file) that identifies the FTP procedures that you use. A command file is a text file that contains commands for FTP.

To use a Command File, do the following:

Step	Action
1	Create a text file that contains commands for FTP. (See the online Help for information about these commands.) Use a number sign (#) to indicate comment lines.
2	From the Commands menu of the FTP window, choose Run Command File.
3	In the FTP Command File box, select the command file that you want to use.

Example

The following example shows how you can use a command file:

```
#change the remote directory
cd /trees/flowering/fruit
#copy all files from the remote directory
mget *.*
#close the connection
quit
```

Closing an FTP Session

To end the current session and continue using FTP, do the following:

Step	Action
1	From the Session menu, choose Close.
2	If you have two sessions open, select the session that you want to close in the Choose Session to lose dialog box.

To end a session and exit from FTP, choose Exit from the Session menu.

Sessions that you save for FTP are also available to TNVT, unless you configured the applications to save session information in different files.

Troubleshooting File Transfers

The table that follows lists common problems that you might encounter with FTP file transfer and describes how to respond.

Problem	Response
A binary file is not usable after a transfer. You cannot view or use the file as expected.	Verify that you selected the appropriate file type in the Mode box of the FTP main window.
The connection closes after it has been idle for some time.	The FTP connection has been idle for a period of time, and the remote host has closed the connection. Close the inactive session; then reopen it.
An FTP command to the remote server fails.	The remote FTP server might not support the FTP command. Not all servers support the full set of FTP commands. Verify that the remote host supports the command.

More Information

For additional step-by-step procedures, refer to the application's online Help.

Telnet VT

Introduction

Use Telnet VT (TNVT) to connect your PC to a remote UNIX or other system host using DEC VT terminal emulation.

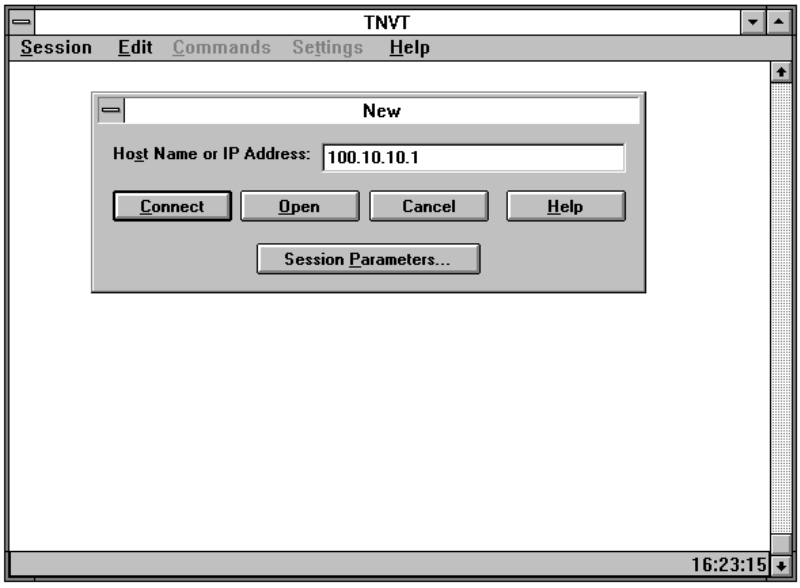
Using TNVT

To use TNVT, do the following:

Step	Action
1	Define a new session.
2	Change session parameters.
3	Perform your remote login tasks.
4	Close the session.

Defining a New Session

When you start TNVT, its main window appears.



The first time that you use TNVT, the New dialog box appears with the main window (if you did not previously save sessions using the FTP application). The New dialog box lets you define session information for the remote host to which you want to connect.

If you previously saved TNVT or FTP sessions (these applications share the same session information), the Open dialog box appears with the main window when you start the application.

To define a new session and connect to a remote host, do the following:

Step	Action
1	Do one of the following: a) Use the New dialog box that appears when you start TNVT. b) From the Session menu, choose New.
2	In the Host Name or IP Address box, type the name or Internet Protocol (IP) address of the remote host to which you want to connect.

Step	Action
3	Connect to the remote host by choosing the Connect or Open button as described in the following table.

Choose this button:	To do this:
Connect	Open the session and immediately connect to the remote host.
Open	Open the session without connecting to the remote host. Use Open to customize session characteristics before connecting; for example, you may want to request a particular terminal type (for example, one that you are familiar with).

Changing Session Parameters

Use the Settings menu to change session parameters. Among the things that you can change from the Settings menu are:

- Terminal type and characteristics. For example, you can increase the number of columns on your screen so that you can easily read text that spans 132 columns.
- Screen colors. For example, you can change the background and foreground colors so that your screen is easier to read.
- Fonts. For example, you can change text so that it is italicized.
- The data that the application sends to the remote host when you press a key (this process is known as keyboard remapping). You can map your keyboard so that when you press a key on your PC, it sends a value that you define (for example, an international character that is not represented on your keyboard, a series of keystrokes, or text).

Closing a Session

When you finish your work on a remote host, choose commands on the Session menu to close the session.

Choose this command:	To do this:
Close	Close the current session, but continue using TNVT.
Exit	Close the session and exit from TNVT.

Troubleshooting

This section lists common situations that you might encounter with TNVT and describes how to respond.

Unrecognized Terminal

The remote UNIX system host does not recognize your terminal type.
If you have system management privileges, verify that the RFC 1060 standard terminal type designations (DEC-VT220, DEC-VT100, DEC-VT52) exist as aliases in the host's /etc/termcap file or /usr/lib/terminfo directory. If they do not exist, add them to the appropriate terminal definitions.

If you do not have system management privileges, you have several options:

- Ask the system manager on the remote host to add the appropriate aliases to the TERMCAP or TERMINFO file.
- Find a VT100, VT220, or IBMPC terminal ID that does exist in the remote host's terminal information file. From the Settings menu, choose Terminal Emulation. Specify the terminal ID in the appropriate Terminal Type String box. This value allows TNVT to pass nonstandard terminal type names when negotiating the terminal type with the remote host.
- Include a tset command in your .login file on the remote host that sets the terminal type, or translates between the tn and UNIX

terminal naming conventions. You also can enter a tset command after you log in to set the terminal type for the duration of that session only.

- Include a command in your .login file that sets the TERM environment variable to a terminal type recognized by the UNIX system. You also can set the environment variable after you log in to last for the duration of your session.

Arrow Keys Not Working

Arrow keys do not work properly in a connection emulating a DEC VT100 or VT220 terminal.
If you have system management privileges, change the VT100 or VT220 parameters in the appropriate terminal definition file on the remote host. If the host uses a TERMCAP file, make sure the :ks variable looks like the following:

```
:ks=\E[?1h\E>
```

If the host uses a TERMINFO database, make sure the smkx variable is set to `\E[?1h\E>`.

More Information

For additional step-by-step procedures, refer to the application's online Help.

DHCP

Description

The Remote Office client receives its IP address at PPP negotiation time. The Remote Office kernel will then dynamically set the IP address for the TCP kernel. In addition, a phonebook record can also be configured to dynamically set a DNS address for a given dial-up session, thus making the use of DHCP unnecessary.

If your Comm Server is configured to respond to BOOTP or DHCP requests, then DHCP can be used to set your local IP and Domain Name Server addresses.

Chapter 4

OnNet MS-DOS Applications

Overview

In This Chapter

This chapter contains the following information:

- PC/TCP networking tasks
- Before you start using PC/TCP commands in DOS
- Using PC/TCP commands from the DOS command line
- PING
- INET
- FTP
- TFTP
- TELNET
- DHCP and BOOTP
- NETBIOS

PC/TCP Networking Tasks

Introduction

PC/TCP® Network Software and PC/TCP® OnNet™ let your PC exchange information and services with other computers that connect to a TCP/IP network. Once you install PC/TCP, you can begin to use it if you know the names and locations of "servers," systems that provide services to other systems in a network. Your system administrator can provide information about the servers at your site. In most cases, your PC is a "client" of the TCP/IP network servers. With PC/TCP, you can connect to many types of servers on your network to extend your PC's computing capabilities.

Networking Tasks

This section explains how to perform many networking tasks. The following table lists these tasks and the PC/TCP commands that you can use for the tasks:

Networking Task	PC/TCP DOS Commands
Transferring files	ftp, tftp
Terminal Emulation	tn
Configuring and tuning PC/TCP	bootp, dhcp, inet, ethdrv, netbios, pctcpcfg, vxdinit
Troubleshooting PC/TCP	inet, ping

Before You Start Using PC/TCP Commands in DOS

Wildcard Formats

Some PC/TCP commands, such as ftp, take command line arguments in a wildcard format. Since the remote host interprets the wildcard syntax, you need to use the wildcard characters that the remote host interprets. For example, you can use the asterisk (*) as a wildcard, as in ls *.doc, only if the remote host interprets the asterisk as a wildcard.

Case Sensitivity in Commands

Many operating systems interpret capital letters in files or commands differently than lowercase letters. If you are using PC/TCP to connect to another DOS host, case sensitivity is not an issue; DOS is not case sensitive. However, if you connect to a UNIX host, be aware that the UNIX host may be case sensitive.

Using PC/TCP Commands from the DOS Command Line

Command Line Syntax Conventions

You enter PC/TCP commands at the DOS prompt. When you enter PC/TCP commands, use the following command line syntax conventions:

Enter	Initiates the command. You can press Enter immediately after the command name for a brief display of command-line options.
Esc or Ctrl+C	Aborts most PC/TCP commands.
-options	Modifies the behavior of the command in the way that you choose. You generally enter a hyphen (-) before command line options. You enter one hyphen followed by one or more options.
command -?	Displays the command syntax and command-line options.
command -version	Displays the version number of the PC/TCP command.

The descriptions of individual commands in Appendix A explain any exceptions to these conventions.

PC/TCP Interactive Commands

After you enter a PC/TCP command, it generally executes a task and then returns control to DOS—you immediately see the DOS prompt. Some PC/TCP commands, however, start an application that provides its own command-line or graphical user interface. For example, when you enter the ftp command, you start an FTP session—you see an ftp command-line prompt. Until you exit from the application, the ftp command interprets any command that you enter. Similarly, when you enter the idnet command, you start a DOS application with a graphical user interface—you work with menus and dialog boxes in the application until you exit from it.

PC/TCP TSR Commands

Other PC/TCP commands start up terminate-and-stay-resident (TSR) programs. TSR programs continue to run, in the background, for as long as your computer remains on, or until you explicitly unload the TSR or restart your PC. PC/TCP provides several TSR programs including the kernel and NetBIOS.

PING

Introduction

The easiest way to test the network is to use the ping command. Ping tests the connection between two hosts by sending an "echo request." An echo request is a message to the remote host, requesting a response. The ping command also displays the time in milliseconds for a response to arrive, and debugging information about the network interface.

If the remote host is active and sends a reply, you know that the network media and IP routers forming the path to that host are working. You can then use other TCP/IP applications on that path.

If a host fails to respond to a network request, it means there has been a failure at one of several points from your PC to the remote host. The host may not be working and is unable to respond, some network or gateway in the path from the user to the host may not be working, or the host may not implement the service that you are requesting.

Example: Testing Remote Host Availability

```
C:\>ping 128.127.50.128
host responding, time = 25 ms
Debugging information for interface ifcust
interrupts: 1145 (0 receive, 0 transmit)
packets received: 682, transmitted: 463
receive errors: 0, unknown types: 198
runts: 0, aligns: 0, CRC: 0, parity: 0, overflow: 0
too big: 0, out of buffers: 0, rcv timeout: 0, rcv reset: 0
transmit errors: 0
collisions: 0, underflows: 0, timeouts: 0, resets: 0
lost crs: 0, heartbeat failed: 463
ARP statistics:
arps received: 13 (10 requests, 3 replies)
bad: opcodes: 0, hardware type: 0, protocol type: 0
arps transmitted: 3 (3 requests, 0 replies)
3 large buffers; 2 free now; minimum of 0 free
3 small buffers; 3 free now; minimum of 1 free
```

Test PING Installation

After you install Remote Office, send an echo request to yourself to make sure that the ping command is installed correctly. Use your IP address as an argument, since hostname translation introduces another variable into the test. Once you have established that the ping command is functional, you can send an echo request to a remote host to test your network connection.

More Information

Refer to Appendix B for a detailed description of supported command line options.

INET

Description

Use the inet command to monitor network traffic and to troubleshoot local network problems.

The inet command requires an argument, several of which are useful to the network administrator in diagnosing network performance problems: config, debug, route, stats, tcp, and unload.

Specify the help argument (inet help) for a list of arguments to the inet command. Specify the version option (inet version) if you want to display the version number of your kernel.

More Information

Refer to Appendix B for a detailed description of supported command line options.

FTP

Introduction

Use the ftp command to transfer text (ASCII) files and binary files between hosts on a network.

It implements the client TCP/IP standard File Transfer Protocol (FTP) on a PC. If you specify a single command, ftp executes that command and quits. Otherwise, it starts up a command interpreter known as an FTP session. During this session, you can execute interactive FTP commands, such as directory commands and file transfer commands.

The ftp client command lets you:

- Open and close an FTP session on a remote host.
- Transfer single and multiple files.
- Transfer binary files
- Select options during a file transfer
- Execute DOS commands during an FTP session.
- Manage directories during an FTP session.
- Display file contents during an FTP session.

Starting a Session

Do the following to start an FTP session on a remote host:

Step	Action
1	At the DOS prompt, enter ftp followed by the name or IP address of the remote host. For example, to connect to a remote host named paris, enter `C:\> ftp paris`
2	To accept the default user ID supplied, press Enter or enter another user ID at the prompt. For example, if your user ID is pat, enter the following: Userid for logging in on paris.comp.com (default)? `pat`
3	Enter your password at the prompt from the remote host. Password for logging in as (pat) on paris.comp.com? If you entered an incorrect password, you are still in an FTP session. You can restart the login procedure by typing `user` at the prompt, or you can close the session by typing `bye` or `quit.`

After you open an FTP session, you see the ftp prompt. This prompt differs in appearance depending on how you specified the remote host. In the following examples, the prompt is abbreviated to `ftp>.`

Starting the FTP Client Without Connecting to a Host

Enter ftp without specifying a host. Doing this allows you to use FTP command options, such as local directory commands or getting FTP help, without being connected to a remote host. Enter `open host` to connect to a remote host.

Opening an FTP Session

With the FTP client application running, enter open followed by the name of a remote host.

For example, to open a connection on a host named London, enter

```
ftp>open london
```

Closing an FTP Session

To close an FTP session without leaving the FTP client, enter `close` or `disconnect`.

Ending an FTP Session

At the session prompt, type `bye`, `exit`, or `quit`.

Transferring Files

Once you open your FTP session, you can

- Get a single file from the remote host.
- Get multiple files from the remote host.
- Put a single file on the remote host.
- Put multiple files on the remote host.

Get a Single File From the Remote Host

Enter `get` followed by the filename of the remote file and the filename that you want to give the file on your local PC.

For example, to transfer the file BUBBLES.MSS from the remote host and store it under the same name on the local PC, enter the following:

```
ftp>get bubbles.mss bubbles.mss
```

To transfer the file SOAP.TXT and assign it the name BATH.TXT, enter the following:

```
ftp>get soap.txt bath.txt
```

The FTP session prompts you for these filenames if you do not supply them. Case sensitivity rules on the remote host may differ from those on your PC.

Get Multiple Files From the Remote Host

Enter `mget` followed by the filenames to be transferred. Specify parts of the filename common to all the needed files, and use wildcards to stand for other characters.

The remote host interprets the wildcard syntax. After transfer, the files retain their original names on your PC. For example, to transfer all files named JUICE with any extension, such as the files JUICE.MSS and JUICE.BAT, enter the following:

```
ftp>mget juice.*
```

To transfer all files that have an .MSS extension, enter the following:

```
ftp>mget *.mss
```

These examples may not work as described if the remote host's wildcard syntax is different.

Put a Single File on the Remote Host

Enter `put` followed by the filename of the local file and the filename that you want to give the file on the remote host.

For example, to put SHOWER.TXT on the remote host, enter the following:

```
ftp>put shower.txt shower.txt
```

Put Multiple Files on the Remote Host

Enter `mput`, followed by wildcard syntax for the files to be transferred from your PC. Use PC wildcard syntax, since it is your PC that interprets the syntax. The mput command works like the mget command.

Transferring Files in Binary Mode

The FTP default transfer mode for transferring files is ASCII. Some files (for example, executable files) must be transferred in binary mode rather than in ASCII. You can set binary mode as the prevailing transfer mode, or you can specify binary mode transfer for individual files. Binary modes are Binary, Image, Tenex, or Local *n* (where *n* is the byte size of the remote machine).

To set the transfer mode to binary for all file transfers

Enter `binary`, `image`, `tenex`, or `local n` at the ftp prompt. You can then use the get or put command for binary files as you do for ASCII files.

To reset the transfer mode for ASCII file transfers, enter `ascii` at the ftp prompt:

```
ftp>ascii
```

Executing DOS Commands During an FTP Session

DOS commands are useful during an FTP session if you want to view or edit files on your PC before transferring them, or if you want to create directories on the PC to receive files from the remote host.

Enter an exclamation point (!) followed by a space, followed by the command.

To request a display of the file LEMONS.MSS before sending it to the remote host, enter the following:

```
ftp>! type lemons.mss
```

To escape to a DOS session enter only the exclamation point:

```
ftp>!
```

Your PC displays the DOS prompt, from which you can issue any number of DOS commands.

To return to the FTP session from a DOS session enter exit at the DOS prompt.

More Information

Refer to Appendix B for a detailed description of supported command line options.

TFTP

Introduction

The tftp command, an implementation of the Trivial File Transfer Protocol (TFTP), lets you transfer single files without any user authentication between hosts on a network. Not all hosts implement TFTP service. It is currently available on most networked Multics, 4BSD UNIX, DOS, and TOPS-20 machines.

Syntax

Use the following command syntax for each file that you want to transfer:

```
tftp (get | overwrite | put) local_filename
host remote_filename [image]
```

Options

Variable	Description
local_filename	Specifies the filename on the PC issuing the transfer command; it is always the first filename argument required.
host	Specifies the name of the machine running the TFTP server.
remote_filename	Specifies the name of the file on the remote host running the TFTP server. You need to specify the complete pathname if it is different from the default TFTP directory. For remote UNIX hosts, use the slash pathname syntax (/path/filename).

Choosing Options

Use the following table to determine which option to use:

To do this:	Enter:
Copy a file from the server	`get`
Place a file on the server.	`put`
Overwrite a file on the local system	`overwrite`
Use binary mode transfer	`image`

Command Line Information

Refer to Appendix B for a detailed description of supported command line options.

TELNET

Introduction

Use the TN command to log in to a remote host using the TCP/IP Telnet protocol. A Telnet server must be running on the remote host.

Description

The TN program negotiates with the remote host to determine the appropriate terminal to emulate (in accordance with the Telnet Terminal Type Negotiation protocol described in RFC 1091). If the remote host does not comply with RFC 1091, tn offers only the default DEC VT220 terminal type, unless you use the `-t` or `-x` command line options to request a specific terminal type or a range of types.

Once you connect to a remote host, you can use escape commands to change features of the connection, create connections to other hosts, display information about the connection, and transfer data between the remote host and your PC.

An advantage of the TN program is that it implements the TCP/IP Telnet protocol, which is widely available on systems such as UNIX and VMS hosts.

Examples

- To open a Telnet connection to the host xyz.com and allow tn to negotiate the terminal type, enter

  ```
  C:\> tn xyz.com
  ```

- To request emulation of a DEC VT100 terminal with a 43- or 50-line screen on the host xyz.com, enter

  ```
  C:\> tn -h -x vt100 xyz.com
  ```

- To request emulation of a DEC VT100 terminal with a 43- or 50-line screen on the a host with internet address 100.10.10.111, enter

```
C:\> tn -h -x vt100  100.10.10.111
```

More Information

Refer to Appendix B for a detailed description of supported command line options.

DHCP and BOOTP

Description

The Remote Office client receives its IP address at PPP negotiation time. The Remote Office kernel will then dynamically set the IP address for the TCP kernel. In addition, a phonebook record can also be configured to dynamically set a DNS address for a given dial-up session, thus making the use of DHCP or BOOTP unnecessary.

If your Communications Server is configured to respond to BOOTP or DHCP requests, then these programs can be used to set your local IP and Domain Name Server addresses.

NETBIOS

Description

The netbios command starts and unloads the PC/TCP NetBIOS (NETBIOS.COM) TSR program.

PC/TCP NetBIOS configuration option defaults are defined in the [pctcp netbios] section of your PCTCP.INI configuration file.

More Information

Refer to Appendix B for a detailed description of supported command line options.

Chapter 5

Configuring the Kernels

Overview

In This Chapter

This chapter describes the following:

- Configuring PC/TCP VxDs
- Configuring TSRs

Configuring PC/TCP VxDs

Introduction

The PC/TCP VxD kernel supports TCP/IP applications in a Windows operating environment. Because the VxD 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.

When to Use Kernel

You use the PC/TCP VxD kernel when you:

- Require network connectivity only with Windows and DOS applications initiated within Windows.
- Need to conserve conventional memory. (The VxD kernel requires approximately 2.5K of conventional memory.)

How to Load

You instruct Windows to load PC/TCP VxDs using the vxdinit command that you enter at a DOS prompt before you start Windows. This command loads a small TSR that instructs Windows to load the PC/TCP VxD kernel and optionally other PC/TCP VxD applications (such as NetBIOS). The VXDINIT TSR loads high by default.

How to Configure

The installation procedures generally configure your system with defaults that suffice for most operating environments. However, before you issue the vxdinit command, you should review the configured defaults in your PCTCP.INI file to verify that the configuration is correct.

You specify VxD loader configuration option defaults in the [pctcp vxdinit] section of the PCTCP.INI file. (You can override these default settings using vxdinit command line options.)

You can configure the PC/TCP VxDs using standard PC/TCP configuration utilities (for example, the Configure application under Windows to configure the kernel).

Any vxdinit command line options specified in your AUTOEXEC.BAT file override these [pctcp vxdinit] default settings.

If you do not override the PCTCP.INI parameters, PC/TCP VxDs use defaults that are appropriate for most operating environments.

Configuring TSRs

Introduction

You select a TSR kernel when you require network connectivity with TSRs that you load before you start Windows, if you interoperate with third-party applications that work only with the TSR kernel, or if you need to operate in both DOS and Windows environments.

When you use the TSR kernel, you must also use the TSR versions of these applications.

All TSRs reside in memory and remain active until you explicitly unload them.

TSR Commands

This table lists the PC/TCP TSRs with their corresponding load and unload commands:

For this TSR:	Use this load command: Use this unload command:
kernel	`C:\> ethdrv (for example` `C:\> inet unload`
NetBIOS	`C:\> netbios.com` `C:\> netbios -u`

How to Configure

The PC/TCP installation procedures generally configure your system with defaults that suffice for most operating environments. However, before you issue the kernel command, you should review the configured defaults to verify that the configuration is correct.

You can configure the PC/TCP TSRs using the standard PC/TCP configuration utilities. You can override configuration file settings for PC/TCP TSRs using the appropriate command line options.

Load the PC/TCP kernel TSR into memory last if you intend to unload it frequently. To reuse memory space occupied by any TSR, you must first unload every TSR that you loaded after it.

The order that you load TSRs into memory may affect PC/TCP performance. You must load the PC/TCP kernel into memory before starting any other PC/TCP network program or third-party program that uses PC/TCP.

If you do not load the kernel into memory first, you will get this error message:

PC/TCP resident module is not loaded; aborting program...

When you use PC/TCP TSRs, you must configure the kernel with the DNS option on (enter `kernel-does-dns=yes` in the [pctcp kernel] section).

Chapter 6

Configuring NetBIOS

Overview

In This Chapter

This chapter contains the following information:

- NetBIOS introduction
- Starting the NetBIOS VxD
- Tuning PC/TCP for NetBIOS
- Communicating across routers
- Creating and using a name file
- Using a NetBIOS name

NetBIOS Introduction

About NetBIOS

PC/TCP 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. Note that unless you plan to run a network operating system that uses the NetBIOS transport protocol, you do not need to load PC/TCP NetBIOS.

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

Using NetBIOS

PC/TCP provides a NetBIOS TSR (terminate-and-stay-resident program) and a NetBIOS VxD (virtual device driver). The PC/TCP installation procedures install and configure the appropriate PC/TCP NetBIOS for your system environment.

Both PC/TCP NetBIOS implementations provide the same functionality.

NetBIOS Configuration

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.

You must unload and reload PC/TCP NetBIOS for new configuration option settings to take effect.

Starting the NetBIOS VxD

Introduction

If your system is configured to use the VxD kernel and NetBIOS, run the vxdinit command prior to loading windows.

Before you start Windows, create an entry in your PCTCP.INI file for starting the NetBIOS VxD.

```
[pctcp vxdinit]
   vnbep=yes
```

Procedure

To start the NetBIOS VxD, do the following:

Step	Action
1	Verify that the vnbep=yes parameter is set in the [pctcp vxdinit] section of your PCTCP.INI file.
2	If the vnbep=yes parameter does not exist, create it.
3	Enter `inet unload` to unload the PC/TCP VxD kernel.
4	Enter `vxdinit` to restart the kernel with the new configuration.
5	Restart Windows. The kernel loads the NetBIOS VxD.

Loading and Unloading the NetBIOS TSR

Use these procedures to load and unload the NetBIOS TSR. PC/TCP NetBIOS loads into upper memory by default. You need to unload and reload NetBIOS for configuration changes to take effect

When you load the PC/TCP NetBIOS TSR with other applications, you should load NetBIOS:

- After you load the PC/TCP kernel.

- Before you load Windows (if you are using Windows).
- Before you start any NetBIOS application (e.g. before you start a LAN Manager workstation).

To unload NetBIOS (NETBIOS.COM), enter netbios -u at the system prompt:

```
C:\> netbios -u
```

Setting PC/TCP NetBIOS Configuration Options

PC/TCP NetBIOS configuration options are applied when you load PC/TCP NetBIOS.

A set of standard configuration option default values are supplied with PC/TCP NetBIOS. For most network environments these defaults should be sufficient, and you should not need to reconfigure NetBIOS.

PC/TCP NetBIOS configuration option defaults are specified in the [pctcp netbios] section of your PC/TCP configuration file (C:\PCTCP\PCTCP.INI, by default).

Example

This is a sample PC/TCP NetBIOS configuration file section. When you do not specify a value for a configuration parameter, NetBIOS uses a default value.

```
[pctcp netbios]
adapter-number =
broadcastfile =
cache-elements =
domain-scope =
names =
loadhigh =
namefile =
ncbs =
scope =
sessions =
timeout =
```

More Information

Refer to Appendix A for a complete listing of {PCTCP Netbios] parameters and descriptions.

Tuning PC/TCP for NetBIOS

Description

Out of the total number of TCP connections configured for the PC/TCP kernel, NetBIOS must allocate 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.

You may need to increase the number of TCP connections for the TCP Kernel in the PCTCP.INI file.

Communicating Across Routers

Introduction

Generally, NetBIOS B-node implementations do not broadcast across IP routers. PC/TCP NetBIOS provides the following mechanisms to transmit broadcasts across routers:

- Broadcast file
- Name file

Configuring PC/TCP NetBIOS

To configure PC/TCP NetBIOS, do the following:

Step	Action
1	Create a broadcast file that identifies the remote hosts to receive NetBIOS messages.
2	Create a NetBIOS name file to associate host Internet addresses with a name.
3	Edit the [pctcp netbios] section of your PCTCP.INI configuration file to identify your NetBIOS broadcast and name files.

Using a Broadcast File

A “broadcast file” is an ASCII file containing a list of Internet host names 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 unload and reload PC/TCP NetBIOS.

Specify the path- or filename of the broadcast file by editing the broadcastfile= parameter in the [pctcp netbios] section of your PCTCP.INI configuration file.

```
broadcastfile=C:\bfiles\miller
```

Example

This is a sample PC/TCP NetBIOS broadcast file:

```
128.127.55.103
lanmanserver1
128.127.55.120
alpha
```

Procedure

To create a broadcast file, do the following:

Step	Action
1	Make a list of the names or IP addresses that you want to add to a broadcast file.
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.
	
	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 broadcastfile= parameter in your [pctcp netbios] section of your PCTCP.INI configuration file specifies the appropriate file or pathname.

If you specify the 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.

Creating and Using a Name File

Introduction

A “name file” is an ASCII file that associates a list of NetBIOS names with their corresponding fully qualified hostnames or IP addresses.

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

Specify the path- or filename of the name file by editing the namefile= parameter in the [pctcp netbios] section of your PCTCP.INI configuration file. This entry identifies the C:\NFILES\PYTHON name file:

```
namefile=C:\nfiles\python
```

Example

This is a sample NetBIOS name file:

```
LMN          128.127.55.103
STUV      128.127.55.120
```

Creating a Name File

To create a name file, do the following:

Step	Action
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.
2	Using a text editor, create a name file entry for each NetBIOS name that you want to add.

Step	Action

3	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. 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 128.127.55.100`
	`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 namefile= parameter in your [pctcp netbios] section of your PCTCP.INI configuration file specifies the appropriate pathname. `namefile=C:\python`

Using a NetBIOS name

Procedure

Enter your command (such as net view), specifying the NetBIOS name associated with the remote host.

Example

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 using the cleese NetBIOS name defined in the PYTHON name file:

```
C:\> net view \\cleese
```

Appendix A

PC/TCP Configuration Parameters Reference

Overview

In This Appendix

This appendix contains detailed information about PCTCP.INI configuration file sections and the parameters that apply to each section. It contains the following information:

- [PCTCP ADDRESSES]
- [PCTCP FTP]
- [PCTCP interface n]
- [PCTCP KERNEL]
- [PCTCP KERNEL-VXD]
- [PCTCP NETBIOS]
- [PCTCP TN]
- [PCTCP PING]

[PCTCP ADDRESSES]

Introduction

This section contains the IP addresses of network servers.

domain-name-server= ip_address

Specifies the IP address of a domain name server, which maps IP addresses to hostnames. You can define up to three domain name servers.

Specifies the fully qualified domain name or IP address of a host acting as the time server in your local network. This parameter is used by the setclock program. You can define as many as five time servers. Each server is tried in the order specified until a valid response is received.

[PCTCP FTP]

Introduction

This section specifies the parameter that points to an initialization file used by the ftp command.

ftpinit= drive:\path\filename

Specifies the name and location of a command file for FTP. After prompting you for a username and a password, FTP takes commands from this file. You can create and use this command file to preset the option command and file transfer modes.

hash-mark-size= number

Sets how often hash marks are displayed by specifying an integer value from 512 to 65536 bytes.

Default: 4096

[PCTCP interface n]

Introduction

This section contains parameters that apply to a specific network interface. In the [pctcp interface n] section, replace the word interface with the name that represents your network driver interface (ifcust). Replace n with a number that uniquely identifies the particular network interface. For example, [pctcp ifcust 0] typically identifies the configuration file section that describes your PC/TCP network interface. For each interface card, you must create a new [pctcp interface n] section.

The values of interface and n are specified in the interface= parameter in the [pctcp kernel] section.

broadcast-address= broadcast_address

Specifies the IP broadcast address for the kernel to use to send packets to all hosts on the local network. Servers use a broadcast address to advertise their existence; clients use a broadcast address to search for available resources.

Default: 255.255.255.255

frame-type= DIX-Ethernet

Will always be DIX-ETHERNET using Remote Office with PKROF-E.

interface-type= PKTDRV

Will always be PKTDRV using Remote Office with PKROF-E.

ip-address= ip_address

Specifies the Internet address of this interface. Your IP address must be unique and must be assigned by your network administrator. The Remote Office Client software will dynamically set the IP address to PPP negotiated address.

odi-pkts=value

Specifies the number of buffers allocated to the network interface driver to receive incoming packets (this means that the driver may queue buffers before returning them to the kernel). This parameter applies to network interfaces using Packet, NDIS, or ODI drivers.

Range: 1 - 8 *Default*: 3 (TSR), 2 (VxD)

router= ip_addresses

Specifies the default IP router addresses (up to a maximum of three). If all your hosts are on a single LAN, you can ignore this parameter. Otherwise, ask your network administrator for the value for this parameter. This parameter is ignored for Remote Office connections.

subnet-mask= 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. This parameter is ignored for Remote Office connections.

[PCTCP KERNEL]

Introduction

This section contains parameters that apply to the operation of the PC/TCP kernel.

bsd-urgent=[no|yes]

Specifies the type of out-of-band, urgent pointers that TCP uses. Yes sets the type to BSD urgent pointers. No sets the type to RFC 1122 urgent pointers.

Default: no

do-slow-start=[no|yes]

Specifies if slow start is enabled or not. Set `do-slow-start=yes` if you experience problems with some Windows applications.

Default: no (yes for SLIP and PPP kernels)

host-table= drive:\path\filename

Specifies a formatted file that contains the Internet addresses and corresponding hostnames of hosts connected to your network. If you have such a file in your directory, type the drive, path, and name of the file (such as C:\ETC\HOSTS.TXT). Note that if you do not have such a file available to you, PC/TCP can still resolve hostnames to the correct Internet address by accessing one or more domain name servers on the network (if such servers are available).

huge-packets= number

Specifies the number of very big packet buffers to allocate. These packet buffers are used by InterDrive and NFS to reassemble big (8K) UDP datagrams.

Default: 0 (TSR), 25 (VxD)

huge-packet-size= number

Specifies the size of each huge packet in bytes.

Range: 4096 – 9216

Default: 9216

interface= interface n

Identifies the [pctcp interface n] section that describes your network interface. For example, [pctcp ifcust 0] usually specifies your PC/TCP network interface.

ip-delay=[high|low]

Specifies the IP type of service with regard to the likelihood of delay in the flow of data packets.

Default: high

ip-precedence= value

Specifies the type of IP precedence (defined by RFC 791 as superseded by RFCs 1195 and 1349). You can use one of the following values:

routine override priority internet-control

immediate critical flash network-control

Default: routine

[PCTCP KERNEL]

ip-precedence-matching= value

Specifies the type of IP precedence checking. This parameter defines whether or not packets coming in from a remote host need the same IP precedence as those defined by your local ip-precedence= parameter.

Value	Description
lax	Specifies that packets coming in from a remote host need not match your local level of IP precedence.
strict	Specifies that packets coming in from a remote host require strict conformance to your local level of IP precedence.

Default: lax

ip-reliability=[low|high]

Specifies the IP type of service with regard to the reliability of data packet delivery.

Default: low

ip-security= value

Specifies the type of IP security.

Value	Description
basic	Uses the values from the basic-classification= and basic-authority= parameters in the [pctcp ip-security] section.
extended	Uses the values from the basic-classification= , the basic-authority= , and the extended= parameters in the [pctcp ip-security] section.
none	Uses no IP security. Disabling this option can reduce the size of the TSR kernel by approximately 1.3K.

Default: none

ip-throughput=[low|high]

Specifies the type of IP service with regard to the throughput of data packets.

Default: low

ip-ttl= number

Specifies the time-to-live number for a packet. This figure indicates the number of gateway hops a packet makes before expiring.

Default: 64

keepalive-timeout= number

Specifies the number of milliseconds a TCP connection remains active before it expires.

Default: 60000

kernel-does-dns=[no|yes]

Specifies whether the TSR kernel loads DNS support or relies on the network for resolving domain names. If you use PC/TCP TSRs (such as NetBIOS and the InterDrive NFS client), you must use kernel-does-dns=yes.

This option does not apply to the VxD kernel.

Default: no

kernel-int= interrupt_vector

Specifies a different software interrupt for the kernel. This parameter is useful when there is a conflict with other terminate-and-stay-resident (TSR) programs. PC/TCP applications automatically determine which software interrupt to use.

Default: 61 (in hexadecimal)

[PCTCP KERNEL]

large-packets= number

Specifies the number of large packets that the PC/TCP kernel allocates. Large packets are of equal size to the maximum packet size on the local network. The memory required for a large packet buffer defaults to the maximum size allowed on the local network. For example, an Ethernet large packet buffer contains 1514 bytes.

Default: 5 (TSR), 20 (VxD)

loadhigh=[yes|no]

Specifies if the PC/TCP TSR kernel loads into the upper memory block (UMB). Yes loads the kernel into the UMB. No loads the kernel into conventional memory.

This option does not apply to the VxD kernel. For information about loading the PC/TCP VxD kernel, see the [pctcp vxdinit] section.

Default: yes

mtu-discovery=[yes|no]

Specifies if the kernel negotiates with the remote host for the number of TCP connections, and the Maximum Transmission Units (MTUs) that the path allows. If you set mtu-discovery=no, the kernel defaults to a TCP maximum segment size of 536 bytes when communicating with machines on remote subnets.

Default: yes

multicast=[no|yes]

Specifies if receiving multicast packets from this host is allowed. Disabling this option can reduce the size of the TSR kernel by approximately 1.9K.

Default: no

pktdrv-loopback=[yes|no]

Specifies if multicast packets loop back to the application that sent them. RFC 1112 requires that there must be a mechanism to turn off packet loopback.

Default: yes

router-discovery=[no|yes]

Specifies if the kernel sends out router solicitation packets on startup, requesting the IP addresses of reachable routers.

Default: no

serial-number= number

Specifies your unique PC/TCP serial number, which is supplied with your PC/TCP distribution disks.

Some installations do not require a serial number.

small-packets= number

Specifies the number of small packets that the PC/TCP kernel allocates.

Default: 5 (TSR), 80 (VxD)

small-packet-size= number

Optimizes your kernel performance when you specify this parameter appropriately for your network environment. For greater optimization, FTP recommends a minimum of 60 bytes.

Default: 160

[PCTCP KERNEL]

tcp-connections= number

Specifies the number of TCP connections. If the kernel does not have enough memory for the number of TCP connections that you request, you will receive an error message.

Default: 5 (TSR), 30 (VxD)

udp-connections= number

Specifies the number of UDP connections.

Default: 5 (TSR), 30 (VxD)

use-emm=[yes|no]

Specifies if the PC/TCP TSR kernel attempts to load into LIM EMS 4.0 memory.

The TSR kernel loads some of its code into LIM EMS 4.0 memory, if you have an expanded memory manager (such as EMM386.EXE) loaded.

This option does not apply to the PC/TCP VxD kernel.

Default: yes

window= window_size

Sets the TCP window size in bytes. A common window size is either 1024 or 8192 bytes. If the window size is set to 0, programs using this parameter default to 1024 bytes on an Ethernet network. This setting allows the other host to send one maximum-size packet. This parameter remedies a performance problem often found with machines running a BSD UNIX operating system.

Default: 1024 (TSR), 4096 (VxD)

[PCTCP KERNEL-VXD]

Introduction

This section specifies default values for certain configuration parameters for the VxD kernel. For defaults for the other configuration options, and for detailed descriptions of each of these parameters, see the [pctcp kernel] section.

Defaults

The following values are the VxD defaults for the [pctcp kernel] parameters:

huge-packets=25

large-packets=20

small-packets=80

tcp-connections=30

udp-connections=30

window=4096

[PCTCP NETBIOS]

Introduction

This section contains PC/TCP NetBIOS configuration parameters. For more information about how to configure PC/TCP NetBIOS, see Chapter 6, Configuring NetBIOS.

broadcastfile=[pathname|filename]

Specifies the file- or pathname of the broadcast file that lists the hosts that you want contacted whenever you send broadcasts, name lookups, or registration packets to another network. Any hosts listed in this file logically become part of the NetBIOS broadcast domain. The broadcast file is read into memory when you load NetBIOS.

cache-elements= number

Sets the number of the NetBIOS name cache elements.

Default: 10 (TSR), 256(VxD)

domain-scope= name

Specifies an additional name (up to 19 characters) to use when a NetBIOS name is unresolved.

Default: no domain scope

loadhigh=[yes|no]

Specifies if the PC/TCP TSR NetBIOS attempts to load into the upper memory block (UMB).

Yes loads into the UMB. No loads into conventional memory.

Default: yes

name-broadcast-retry= number

Sets the internal retry counter according to the RFC 1001 and RFC 1002 protocols. Specify the number of times to retry a name broadcast request.

Range: 1 – 16

Default: 3

name-broadcast-timeout= number

Sets the internal timeout between the next name broadcast requests (see RFC 1001 and RFC 1002). Specify the timeout number in 250-ms time units for the NetBIOS VxD and in 500-ms units for the NetBIOS TSR.

Range: 2 – 254

Default: 2

name-delete-retry= number

Specifies how many times NetBIOS retries to perform a name delete request. (See the name-broadcast-timeout entry to specify the timeout between retries.) If you experience long delays while exiting Windows, set this parameter to 1.

Default: 3

namefile=[pathname|filename]

Specifies the name of a file used to resolve NetBIOS names. Each name file parameter associates a NetBIOS name with a corresponding hostname or Internet address.

names= number

Sets the maximum number of local names (up to a maximum of 254).

Default: 16 (TSR), 16(VxD)

[PCTCP NETBIOS]

ncbs= number

Sets the maximum allowable number of outstanding NCBs (Network Control Blocks). The maximum is 169.

Default: 24 (TSR), 32(VxD)

scope= string

Specifies the name scope string (up to 19 characters). If the string contains spaces, do not use double or single quotation marks around the string.

sessions= number

Sets the maximum number of active NetBIOS sessions up to the number of kernel TCP connections.

Default: 16 (TSR), 16(VxD)

timeout= number

Sets the application timeout interval (in 500-ms NetBIOS time units. Use this parameter to override the application default.

A timeout value of 0 specifies that the application will never time out.

This option is particularly useful if you run the IBM PC LAN program, for which the time out sequence is 15 seconds. Foreground activity can often take longer (so an infinite timeout can improve performance).

Range: 0–254

[PCTCP TN]

Introduction

This section contains parameters that apply to the tn command.

authentication= state

Specifies:

- Whether to provide Kerberos authentication services for a Telnet session.
- How to proceed if authentication synchronization fails.

State Keywords

Possible state keywords are as follows:

This keyword:	Indicates this:
enabled	Turn Kerberos authentication on. If authentication synchronization with the Kerberos server fails, keep the Telnet session open and prompt for the username and the password.
disabled	Turn Kerberos authentication off.
required	Turn Kerberos authentication on. If authentication synchronization with the Kerberos server fails, close the Telnet session.

Default: enabled

[PCTCP TN]

back-arrow-key=[del|bs]

Specifies if pressing the Backspace (<-) key on your keyboard sends an ASCII backspace character or an ASCII delete character during a Telnet session. A delete character erases characters that you have typed. A backspace character travels back over characters that you have typed without erasing them.

Default: del

cmdline=[options] hostname

Specifies a default remote host to connect to, as well as any other tn command line options (each field separated by a space). This lets you start tn or create a new connection without having to specify command line options. If you do specify other options on the command line, those options override this parameter.

encryption= state

Specifies:

- Whether to encrypt Telnet session data.
- How to proceed if encryption synchronization fails.

State Keywords

Possible state keywords are as follows:

This keyword:	Indicates this:
disabled	Turn encryption off.
enabled	Turn encryption on. If encryption synchronization with the Kerberos server fails, send Telnet data unencrypted.
required	Turn encryption on. If encryption synchronization with the Kerberos server fails, close the Telnet session.

Default: disabled

ftpsrv=[off|on|never]

Specifies if you enable an embedded FTP server automatically on your PC when you start tn. If you specify on, other users can establish an FTP connection and transfer files to and from your machine. Specifying never prevents enabling of the FTP server from within a Telnet session.

Default: off

kerberos4-auth=[on|off]

Specifies if your PC uses Kerberos Version 4 authentication, if requested by the remote host. If you specify on, your PC uses this authentication.

Default: on

kerberos4-only=[off|on]

Specifies if your Telnet connection closes without attempting a normal login if Kerberos Version 4 authentication fails.

Default: off

service-name= service

Specifies a Kerberos service name other than rcmd. The tn program requests the RFC 1411-compliant name rcmd; hosts using various Telnet server implementations may use a different name.

screen-saver=[no|yes]

Specifies if a screen saver runs on your PC while you use tn. If the value is yes, tn passes your keystrokes to DOS, as well as to the remote host. This prevents the screen saver from activating during a Telnet session. If set to no, tn passes your keystrokes to the remote host only.

Default: no

[PCTCP TN]

status=[on|off]

Specifies if the status line displays at the bottom of the screen in a tn connection. The status line displays the remote hostname, the date and time, and any messages resulting from certain escape commands. If this parameter is set to off, the status line does not display.

Default: on

x-display= hostname:server.screen

Specifies the host name of your display hardware for the X Window System. This lets you set your display environment on the local, rather than the remote host.

This variable:	Indicates this:
hostname	The hostname of display hardware. You can specify a hostname or an IP address.
server	The server number. (Refers to a collection of monitors that share a common device such a keyboard.) Each server is assigned a number beginning at 0.
screen	The screen number (refers to the collection of screens in a multiple server configuration). Each screen is assigned a number beginning at 0.

[PCTCP PING]

Introduction

This section contains parameters that apply to the Ping program. Each parameter can be set by editing the PCTCP.INI file manually or by using Ping program menus and options.

host= hostname

Specifies the name of the host to contact when the Ping program starts. Use the full hostname when naming a remote host. For example, `host=spud.xyz.com`.

Default: no hostname

icmp-data-length= number

Sets the value for the length (in bytes) of data packets. The minimum value is 18; the maximum value depends on the Maximum Transmission Unit (MTU) supported by your network.

Range: 18 – maximum MTU

Default: 56

icon-bar=[yes|no]

Specifies if the icon bar is displayed when the Ping program starts.

Default: yes

mode=[stats|trace]

Specifies which command the Ping program displays at startup.

Default: stats

[PCTCP PING]

ping-on-startup=[no|yes]

Specifies if the Ping program automatically contacts a host (specified by the host= hostnameparameter) at startup.

Default: no

savesettings=[yes|no]

Specifies if the Ping program saves the values set during the most recent Ping session.

Default: yes

status-bar=[yes|no]

Specifies if the status bar is displayed when the Ping program starts.

Default: yes

time-interval= number

Sets the value for the number of seconds between each echo request.

Range: 1 – 9999

Default: 1

trace-show-host=[yes|no]

Specifies if the Trace command displays the remote hostname (specified by the host= hostnameparameter).

Default: yes

Appendix B

Command Reference

Overview

In This Appendix

This appendix contains a summary of the PC/TCP OnNet commands. It describes commands for the following:

- ftp
- Interactive ftp
- inet
- ethdrv
- netbios
- pctcpcfg
- ping
- tftp
- tn

ftp

Usage

ftp [-d] [**-u** userid password] [**-p** port_no] [host] [command]

ftp [-? | -version]

Description

The ftp command transfers text (ASCII) files and binary files between hosts on a network.

It implements the client TCP/IP standard File Transfer Protocol (FTP) on a PC. If you specify a single command, ftp executes that command and quits. Otherwise, it starts up a command interpreter known as an FTP session. During this session, you can execute interactive FTP commands, such as directory commands and file transfer commands.

The default transfer mode is ASCII text. To send or receive binary files, you log in to the remote host and set the transfer mode for the session by using the **binary, tenex**, or **image** interactive command. Then you use the **get** or **put** command for each file you transfer.

You must choose the file mode based on the byte size of the system to or from which you are transferring the file. If the system has the same word size as the IBM PC (8, 16, or 32 bits), you can use **image**. If it does not, you should probably use **tenex** (binary) mode. The most common systems in use that do not have the same word size as the PC are LISP and TOPS-20 systems.

To transfer a single binary file, you can log in to the remote host and use the **iget, tget, tput,** or **iput** command for each binary file you transfer.

Command Line Options

Command line options are as follows:

Option	Meaning
command	Performs the specified operation. See the "Interactive FTP Commands" section that follows.
-d	Displays all FTP network commands and responses that are sent over the FTP control connection. This option displays your password after you enter it.
host	Specifies the name of the host on which to open an FTP connection.
-p *port_no*	Specifies the number of the remote port that the FTP server monitors.
-u *userid, password*	Automatically logs you in to the remote host using your username and password.
-?	Displays and explains the usage line of the command.
-version	Displays the version and patch level of the command. Refer to this information if you call Technical Support.

Interactive FTP Commands

Introduction

During an FTP session, you can use a large number of interactive commands. Unambiguous abbreviations of these commands often work in place of the full command name. For example, **di** is the same as **dir**, but **d** is ambiguous; it could mean **debug, delete, dir**, or **drive**.

Some older systems may support FTP servers that do not implement certain commands like **mkdir, rmdir, pwd,** and **parent**. You should upgrade the FTP server program on these systems.

Interactive Commands

When you get the FTP session prompt, you can enter any of the following interactive commands:

Command	Effect
! [*command*]	With no argument specified, switches to a command interpreter. Otherwise it executes the specified DOS command and returns your ftp prompt.
? [*command*]	With no argument specified, displays a list of available commands. Typing **?** and a command (like **acct**) displays a brief help message for the specified command.
abort	Causes the server to abruptly stop the previous FTP service command.
allocate *n*	Allocates storage on a server for a file transfer. Specify the number of bytes to set aside.
append [*local remote*]	Adds local file to a remote file on the remote host in the current mode.
ascii	Enables the transfer of ASCII text files.

Command	Effect
binary	Synonymous with **image.**
bye	Synonymous with **exit.**
cd	Synonymous with **fcd.**
close	Disconnects from the FTP server without exiting from the FTP client program.
debug (on | off)	Sets debugging either on or off. This command is not normally needed.
delete filename	Deletes a specified file on the remote machine. To delete a directory, use the **rmdir** *remote_directory* command.
dir [*t[filename*]]	Lists the current directory on the remote machine. You can use the first option, *argument*, to specify a pathname, filename, or argument for the server to use as it gets a directory listing.
disconnect	Disconnects from the FTP server without exiting from the FTP client program.
drive *driveletter*	Changes the current drive on the PC to the specified drive.
exit	Exits from the FTP session and returns your PC's DOS prompt.
fcd [*path*]	Changes the current working directory on the remote machine.
fdir	Synonymous with **dir**.
fpwd	Shows the name of the current working directory on the remote machine.
get [*remote local*]	Copies a file from the remote machine to the PC in the current mode (ASCII, Image, Tenex, or Local n).

Command	Effect
hash [on | off | *integer*]	Displays a number sign (hash mark #) everytime FTP sends or receives a specified number of bytes.
help	Lists commands supported by FTP. Synonymous with **?.**
iget [*remote local*]	Like **get**, but transfers a remote file to a local directory using Image (binary) mode (for just this one transfer).
image	Enables the transfer of binary files between similar machines. Use the **local** *n* command to transfer files correctly between two different kinds of machines.
iput [*local remote*]	Like **put**, but copies local_file to remote_file using Image (binary) mode. (Subsequent transfers revert back to the default ASCII mode.)
lcd [*local_directory*]	Changes the current working directory on the PC to the specified directory. (Typing a drive letter in the directory name does not change the current drive. Use the **drive** command to change the current drive.)
ldir [*path*]	Displays a local directory listing. Specify *path*, to view the contents of a specific directory.
lmkdir [*local_directory*]	Creates a new directory on the PC with the specified name. You must return to the DOS session to remove the directory. (There is no **lrmdir** command.)
login *user*	Sets your username to *user* on the remote machine.
lpwd	Shows the name of the current directory on your PC.

Command	Effect
ls [*argument* [*filename*]]	Like **dir**, but does a short listing of the current remote directory.
mdelete *wildcard_name*	Deletes multiple files on the remote machine.
mget *wildcard_name*	Transfers multiple files to the local machine using wildcard syntax.
mkdir *remote_directory*	Creates a new directory on the remote machine.
mount *pathname*	Changes the server's file system mount information.
mput *wildcard_name*	Transfers multiple files to the remote machine using wildcard syntax.
open *[host_name]* [*port*]	Opens an FTP connection. Use this command after the FTP client has been started without a connection, or after disconnecting from an FTP server.
ask on	Prompts you for file transfer instructions. For examples, see the description for **mget** or **mdelete.**
casehack off	Uses the case you specify to generate filenames. If this is a wildcard transfer, FTP generates uppercase default names.
page [on | *n***]**	Prevents the results of a command (such as **dir**) from scrolling off the screen.
pathhack off	Uses the path you specify to generate filenames. Otherwise, when FTP generates a default filename, it tries to strip off the path for the output file.
prompt [on | off]	Makes FTP prompt the user for file transfer instructions. For examples, see the description for **mget** or **mdelete**.

Command	Effect
parent	Changes the current directory on the remote machine to the parent directory. Not all servers support this command.
passive	Executes the next transfer in passive mode (instead of the server opening the data connection, the PC opens the data connection).
prompt [on | off]	Makes FTP prompt the user for confirmation of actions to be taken on each file. For examples, see the description for **mget** or **mdelete**.
put [*local remote*]	Copies *local_file* to *remote_file* in the current mode (ascii, image, tenex, or local *n*).
pwd	Synonymous with **fpwd.**
quit	Synonymous with **exit.**
quote *command*	Sends an ftp command directly to the remote machine's server. For example, you can send a command that changes the remote directory to a directory with spaces in its name (**quote cd my new directory**). The complete command and argument are sent together after the **quote** command, which reads the spaces as actual characters.
reinit	Reinitializes the connection to the startup state following an in-progress transfer.
rename [o*ld new*]	Renames an existing filename to a new filename on the remote host. You can rename files on the PC by using the **!** command to escape to a ommand interpreter.
r**etrieve** [*remote local*]	Synonymous with **get.**

Command	Effect
rmdir *remote_dir*	Deletes a specified directory on the remote machine.
send [*local remote*]	Synonymous with **put.**
server *command*	Synonymous with **quote.**
show *filename*	Displays the text of a specified file (from the remote machine only) on your screen.
site free	Returns the number of bytes available on the FTP server.
stat	Returns the current state of the remote machine's server. Not all servers support this command.
take *local_file*	Reads commands for FTP from a local file.
tenex	Enables the transfer of binary files to or from a TOPS-20 machine and from most LISP machines. The Tenex mode is equivalent to Local 8.
tget [*remote local*]	Like **get**, but transfers a remote file to a local directory using tenex (binary) mode (for just this one transfer).
tput [*local remote*]	Like **put**, but transfers a local file to a remote directory using Tenex (binary) mode (for just this one transfer).
type *type*	Displays the current file transfer mode. Set the transfer mode by specifying type. The command accepts the following transfer mode types: **image, ascii, enex, local** *n*, **ascii_nonprint, ascii_telnet**, or **ascii_carriage**. For example, the command **type image** is equivalent to **image.**
user name	Synonymous with **login**.

Command	Effect
verbose (on | off)	Synonymous with **debug.**
version	Displays the version and patch level of the command.

inet

Usage

inet [arp] [debug] [pap] [ppp] [route] [slip] [stats] [tcp] [version] [unload]

inet [config [advanced | security]]

inet [ipcp (stats | config)]

inet [lcp (stats | config)]

inet [-? | -version]

Description

The inet command displays network statistics from the PC/TCP kernel, and can unload the PC/TCP kernel TSR.

Command Line Options

Option	Meaning
advanced	Displays information about advanced kernel configuration parameters such as time to live, type of service, precedence, and so on.
arp	Displays the current contents of the kernel's address resolution protocol (ARP) cache.
config	Displays the date and time the current kernel began running, the hardware configuration used by the kernel, network addresses, and other configuration information.
debug	Displays information about packet receipt and transmission, interrupts, ARP statistics, and so on.

Option	Meaning
stats	Displays the IPCP layer statistics. The IPCP layer is part of the Point-to-Point Protocol (PPP) kernel. It handles IP addresses and Van Jacobson Compression for a PPP link.
config	Displays the IPCP layer configuration, including the initial, intermediate, and final working values.
route	Displays the current contents of the kernel's routing cache.
stats	Displays network statistics about the network interface. The statistics include the name of the interface, its network (IP) address, the subnet mask, and the total number of packets sent out and received on this interface.
tcp	Displays the contents of the kernel's TCP connection table.
unload	Unloads the PC/TCP kernel TSR from memory.
version	Displays the version number of the active kernel.
-?	Displays and explains the usage line of the command.
-version	Displays the version and patch level of the command. Refer to this information if you call Technical Support.

ethdrv

Usage

ethdrv **[-B]** [**-b** broadcast_addr] [**-i** interrupt] [**-m**]

[**-p** lg_pkt_count] [**-s** sm_pkt_count] [**-t** max_tcp_conn] [**-u** max_udp_conn]

ethdrv[**-? | -version**]

Description

PC/TCP TSR ethdrv into memory. Use inet unload to unload the TSR ethdrv from memory.

Do not load or unload the PC/TCP TSR ethdrv while you run Microsoft Windows.

The ethdrv is the central part of the PC/TCP software. Loading the PC/TCP ethdrv lets you use PC/TCP network applications.

The PC/TCP ethdrv supports upper memory (UMB) and expanded memory loading options that you can specify on the command line or in the [pctcp ethdrv] section of the PCTCP.INI configuration file. (The ethdrv loads its code and data segments into upper memory by default.)

PC/TCP ethdrv command line options override configuration parameter settings in the [pctcp ethdrv] and [pctcp interface n] sections of the PCTCP.INI file.

Command Line Options

Option	Meaning
-B	Specifies that Berkeley UNIX-type urgent pointers are used.
-b *bcast_addr*	Specifies an address other than 255.255.255.255 as the broadcast address.

Option	Meaning
-i *interrupt*	Specifies a different PC/TCP software interrupt in hexadecimal. The default is 6.
-m	Allows the ethdrv to use expanded memory if a properly configured expanded memory manager is also running.
-p *lg_pkt_count*	Specifies the number of large packet buffers to reserve when loading the ethdrv into memory. The size of a large packet is equivalent to the Maximum Transmission Unit (MTU) allowed in your network. The maximum number of packet buffers is variable because the size of the MTU changes on networks other than Ethernet or StarLAN. Default: 5 or the number of TCP connections plus 1 (whichever is larger).
-s *sm_pkt_count*	Specifies the number of small packets to reserve when loading the ethdrv into memory. Typically, small packets carry protocol or other handshaking data between systems, and arge packets carry application data. Default: 5 or the number of TCP connections plus 1 (whichever is larger).
-t *max_tcp_con*	Specifies the maximum number of simultaneous TCP connections to allow. The maximum number of TCP connections is 64. Default: 4.
-u *max_udp_con*	Specifies the maximum number of simultaneous UDP connections to allow. Default: 4.
-?	Displays and explains the usage line of the command.
-version	Displays the version and patch level of the command. Refer to this information if you call Technical Support.

Examples

- To unload the PC/TCP TSR ethdrv, enter

  ```
  C:\> inet unload
  ```

- To load the DIX Ethernet TSR ethdrv TSR into conventional memory with 8 TCP and 8 UDP connections, enter

  ```
  C:\> ethdrv -t 8 -u 8
  ```

netbios

Usage

netbios [-a] [-u]

netbios [-? | -version]

Description

The netbios command starts and unloads the PC/TCP NetBIOS (NETBIOS.COM) TSR program. To load the corresponding VxD, see the **vxdinit** command.

PC/TCP NetBIOS configuration option defaults are defined in the [pctcp netbios] section of your PCTCP.INI configuration file.

Command Line Options

Option	Meaning
-a	Uses the alternate interrupt interface (interrupt 0x2A) to the transport layer. By default PC/TCP NetBIOS uses interrupt 0x5C (and does not use the alternate interrupt interface).
-u	Unloads PC/TCP TSR NetBIOS.
-?	Displays and explains the usage line of the command.
-version	Displays the version and patch level of the command. Refer to this information if you call Technical Support.

pctcpcfg

Usage

pctcpcfg [**-k** | **-K**] [**-b**] [**-i** ini-file] [**-p** package] section [**-s** subsection]

("" | variable ["" | value ...])

pctcpcfg [-help]

pctcpcfg [-? | -version]

Description

The pctcpcfg command sets new values in the PCTCP.INI configuration file. If you use the **-k** or **-K** options, you can set new values in the active kernel. You can set more than one value at a time. You can also use this command in DOS batch files.

Command Line Options

Option	Meaning
""	Displays all variables for the section or subsection.
-b	Specifies that a backup file not be created.
-help	Displays a list of kernel variables you can set using the **-k** or **-K** options.
-i *ini-file*	pathname of an alternative PC/TCP configuration file.
-K	Sets a value in the kernel only.
-k	Sets a value in the kernel and the PCTCP.INI file.
-p *package*	Specifies the name of the product package you want to configure. (The default is PCTCP.)
s- *subsection*	Specifies the configuration file subsection.
section	Specifies the configuration file section.

Option	Meaning
value	Specifies the new value, or values, for the variable.
variable	Displays the variable specified (for example, sec-arg).
variable ""	Clears the value for the specified variable.
variable value	Changes the value for the specified variable.
-?	Displays and explains the usage line of the command.
-version	Displays the version and patch level of the command.

Examples

- To display the list of variables and their values for a specific section, enter

  ```
  C:\> pctcpcfg tn ""
  ```

  [pctcp tn] - cmdline = tunes

  [pctcp tn] - ftpsrv = on

- To set two domain-name-server addresses in the [pctcp addresses] section of your PCTCP.INI file and the kernel, enter a command similar to the following. Note that this command removes previous addresses and replaces them with specified IP addresses.

  ```
  C:\> pctcpcfg -k addresses domain-name-server
  128.127.50.101 128.127.50.102
  ```

- To display the value of the subnet-mask= in the [pctcp ifcust 0] configuration file section, enter

  ```
  C:\> pctcpcfg ifcust -s 0 subnet-mask
  ```

  [pctcp ifcust 0] - subnet-mask = 255.255.0.0

ping

Usage

ping [-options] host

ping [-? | -version]

Description

Use ping to determine if a host is active and to isolate host connection problems. The ping command sends an echo request to another host and waits for a response, using the Internet Control Message Protocol (ICMP).

The ping command reports success with a Host responding message followed by statistics for the host that initiated the connection. The command might also report failure with a Ping failed or cannot resolve hostname message followed by statistics about the host that initiated the connection.

Command Line Options

When typing the following options on the command line, leave a space between the option and any values or arguments that correspond to it.

Option	Meaning
-d *[bytes]*	Displays header and debugging information about the incoming packet.
-d# *[bytes]*	Displays header and debugging information about the outgoing packet.
host	Specifies the name or Internet address of the remote host.

Option	Meaning
-i *number*	Sets the IP Time-to-Live (TTL) value on the outgoing packet and displays the TTL value for the incoming packet. The range for number is 1–255; the default is 64.
-l *length*	Sets the length in bytes of the data in a packet. The default length is 256 bytes.
-n *times*	Sends a specific number of echo requests and then stops.
-Q	Turns on the Trace Route option, which performs like the **-q** option except that IP addresses are not translated to domain names.
-q	Turns on the Trace Route option. The option increments the TTL to identify all of the routers encountered when trying to reach the target host, and denotes each router by its IP address and domain name.
-t	Continuously sends echo requests to the target host, each time waiting for a response before sending the next request.
-w *seconds*	Specifies a number of seconds to wait for a response. The default value is 6.
-x	Turns on the IP Timestamp option.
-z	Specifies quiet mode, which reports success or failure rather than full statistics.
-?	Displays and explains the usage line of the command.
-version	Displays the version and patch level of the command. Refer to this information if you call Technical Support.

tftp

Usage

tftp (**get** | **overwrite** | **put**)local_file host remote_file **[image**]

tftp serve

tftp [**-?** | **-version**]

Description

The tftp command lets you transfer one file at a time between hosts on a network, without requiring you to enter a password. The tftp program uses the Trivial File Transfer Protocol (RFC 783) on PCs.

Since the command does not provide authentication of the user, you can usually transfer only publicly accessible files.

The tftp program can also act as a file transfer server that lets remote users transfer files to and from your PC. The TFTP server provides no access control whatever; that is, it lets a remote host initiate any TFTP operation for any file on any accessible disk.

When you have the TFTP server running, DOS transfers files to and from the current working directory. The user on the remote machine, however, can specify a full pathname.

The TFTP server permits only one file transfer at any time. If any host requests a transfer while one is already in operation, the TFTP server refuses the second request.

Command Line Options

Option	Meaning
get	Transfers a file from a remote machine to your PC.
host	Specifies the name or Internet address of the remote host.
image	Transfers binary files literally byte-by-byte from one computer to the other.
local_file	Specifies the name of the file on your PC file system.
overwrite	Overwrites an existing local file with the contents of a different remote file.
put	Transfers a file from your PC to a remote machine. You usually cannot transfer a file to overwrite an already existing file of the same name.
remote_file	Specifies the name of the file in the remote machine file system.
serve	Starts up a TFTP server, letting other hosts transfer files to and from your PC. While the TFTP server is running, you cannot use the PC. Type **q** to end the server program.
-?	Displays and explains the usage line of the command.
-version	Displays the version and patch level of the command. Refer to this information if you call Technical Support.

tn

Usage

tn [-a] [-b] [-D] [-h] [-k realm] [**-n**] [**-p** service-name] [**-t** ttype | **-x** ttype]

[**-w**] host [port]

tn [-? | version]

Description

The tn command lets you log in to a remote host using the TCP/IP Telnet protocol. A Telnet server must be running on the remote host.

The tn program negotiates with the remote host to determine the appropriate terminal to emulate (in accordance with the Telnet Terminal Type Negotiation protocol described in RFC 1091). If the remote host does not comply with RFC 1091, tn offers only the default DEC VT220 terminal type, unless you use the -t or -x command line options to request a specific terminal type or a range of types.

Once you connect to a remote host, you can use escape commands to change features of the connection, create connections to other hosts, display information about the connection, and transfer data between the remote host and your PC. See the “Escape Commands” section for more information.

To end a Telnet connection, log out of the remote host (for example, by pressing Ctrl+D or typing exit). If you cannot log out normally, use the c or q escape command (see the “Escape Commands” section for details).

Command Line Options

Option	Meaning
-a	Causes tn to use the alternate screen size for the chosen terminal type.
-b	Specifies that binary mode not be negotiated during VT emulation.
-D	Turns on Kerberos debugging.
-h	Increases the number of screen lines displayed. The screen height varies depending on the type of terminal being emulated and the capabilities of your PC's video card and monitor:
host	Specifies the name or Internet address of the remote host.
-n	Requests that tn negotiate only for terminal types without extended attributes.
port	Specifies the number of a port on the remote host machine, letting you connect to a remote host through a port other than the default port (port23).

Option	Meaning
-t *ttype*	Requests that tn offer a range of terminal types, beginning with the specified type, during negotiation with the remote host. The tn program sends a name from an ordered list in response to each request from the remote host, cycling through the list until the remote host accepts a terminal type or stops requesting terminal types. The default order in which tn negotiates for terminal types is: DEC VT220 DEC VT100 DEC VT52 IBM PC IBM 3278 Model 2 with extended attribute support IBM 3278 Model 2 You can change this default list by specifying the **-t** option. Use one of these *ttype* keywords to indicate the beginning point in the list: **vt vt220 vt100 vt52 ibmpc 3277 3277 3278 3279**
-w	Enables a wide screen display, if tn negotiates 3270 emulation and if your video card and monitor support wide screen display. This option is available only with the 3278 terminal type (the default IBM terminal type).
-x *ttype*	Requests negotiation of a specific terminal type. Use one of the following keywords: **vt vt220 vt100 vt52 ibmpc 3277 3278 3279** If the terminal you request is a specific type, tn negotiates for that terminal type only. If you use the keyword **vt**, tn offers the three DEC VT types from the list of terminal types in the following order: VT220, VT100, and VT52. This is synonymous with **-t vt.**
-?	Displays and explains the usage line of the command.
-version	Displays the version and patch level of the command.

Escape Commands

To invoke an escape command in a tn session, press the escape key; then enter the escape command at the Command: prompt. The escape key is Alt+F10. Escape commands are case sensitive.

Command	Meaning
?	Displays a help message listing the escape commands.
!	Invokes a nested DOS command interpreter, but does not close the Telnet connection. To return to the remote host, type **exit** at the DOS prompt.
0– 9	Creates a new connection, or switches to an existing connection identified by the number you supply. To see a list of existing connections, use the **Ctrl+s** escape command. To create a new connection, press the escape key and enter any number 0 through 9 that is not in use for an existing connection (use the **Ctrl+s** escape command to see a list of existing connections).
a	Sends a Telnet protocol IAC Are-You-There command to the remote host to determine if the Telnet connection is active. Most hosts respond *yes*.
b	Sends a Telnet protocol IAC Interrupt Process command to the remote host.
c	Closes the current Telnet connection gracefully.
Enter key	Switches from the current connection to the next open connection in the list. As many as 10 connections can be open at a time.
I	Displays the local host's IP address on the status line.
i	Sends the contents of a PC file as input to the remote host command line. You are prompted for the name of the file, whose contents are then sent to the remote host exactly as if you typed and entered them there. The IBM 3270 terminal emulator does not accept this command.

Command	Meaning
l	Enables local echo mode. The remote host receives your keystrokes, but does not send them back to your PC. DOS displays the keystrokes on your PC screen. IBM mainframes do not accept the **l** escape command, nor do some other hosts.
o	Records your keystrokes and the resulting remote system output in a file on your PC.
Q	Ends all existing connections by resetting them instead of closing them gracefully.
q	Ends the current remote login session by resetting the connection instead of closing it gracefully.
r	Enables remote echo mode. The remote host echoes keystrokes back to the local host over the Telnet connection.
t	Sends a Telnet break signal to the remote host.
U	Turns on the display of the status line at the bottom of the terminal screen (this is the default).

Index

E

F

G

H

I

K

L

M

N

O

P

Q

R

S

T

U

V

W

X