User's Guide

HP ARPA Services 2.1/MS-DOS

HEWLETT
PACKARD

D1812-90016
Printed in U.S.A. 05/92

Notice

Hewlett-Packard makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Hewlett-Packard assumes no responsibility for the use or reliability of its software on equipment that is not furnished by Hewlett-Packard.

HP LAN Manager is based on Microsoft® OS/2 LAN Manager and on 3+Open®.

Colorado Networks Division
3404 East Harmony Road
Fort Collins, Colorado 80525-9599, U.S.A.

About this Guide...

This guide is part of a three-manual documentation set for the HP ARPA Services 2.1/MS-DOS product. It describes how to use each of the HP ARPA Services 2.1/MS-DOS®.

You do not need prior knowledge about the HP ARPA Services to use this guide. However, you should be familiar with your PC's operating system.

If you need information about installing or configuring the HP ARPA Services 2.1, refer to the *Installation Guide – HP ARPA Services 2.1/MS-DOS*.

How this Guide is Organized

This guide includes the following chapters and appendix.

Chapter 1	The "Introduction" describes the HP ARPA Services and what you can do with them, as well as what you need to know before using the services.
Chapter 2	"Using TELNET for Remote Login" describes how to use the TELNET service.
Chapter 3	"Using FTP to Transfer Files" describes how to use the FTP service. It also includes a reference section for the FTP commands.
Chapter 4	"Using Remote Command Services" describes how to use the RCP and RSH services.
Chapter 5	"Troubleshooting with PING" describes how to use the PING command to troubleshoot ARPA Services problems.
Appendix A	"Making Services Transient" describes how to make an ARPA Service transient, so that the service is loaded in your PC workstation's memory only when you want to use the service.
Appendix B	"Using ARPA Services with Windows 3.0" describes how ARPA Services 2.1 works with Microsoft® Windows 3.0, including limitations and special considerations.

Conventions Used in this Guide

This guide uses the following notational conventions:

Notation	Description
Boldface	Boldface type is used when a term is defined.
`Computer Text`	Computer type is used for commands and keyboard entries that you must type exactly as shown. It is also used for on-screen prompts and messages.
italics	Italic type is used for emphasis and for titles of manuals and publications. Italic type is also used to represent a variable, such as *fileename*.
Key	This font is used to indicate a key on the computer's keyboard. When two or more keys appear together with dashes separating them, such as Ctrl-D, press those keys simultaneously to execute the command.
Softkey	This font is used to represent function softkeys that appear at the bottom of your screen.
[ ]	An element inside brackets in a syntax statement is optional. Several elements stacked inside brackets means you may select any one or none of these elements.

Contents

Chapter 5: Troubleshooting with PING

Appendix A: Making Services Transient

Appendix B: Using ARPA Services with Windows 3.0

Glossary

Index

Introduction

Chapter Overview

The HP **ARPA Services** 2.1/MS-DOS product allows your HP Vectra PC or any other supported PC to transfer files with and log on remote computers.

This chapter covers the following information:

- What you need to do before you can use ARPA Services.
- An overview of the ARPA Services.
- How computers are identified on the network.
- An overview on using the DOS 5.0 Task Manager shell with the ARPA Services.

Before You Begin

Before you use ARPA Services, check the following items:

- The ARPA Services software must be installed and configured for use with your **local PC**. For information about how to install and configure the software, refer to the *Installation Guide: HP ARPA Services 2.1/MS-DOS*.
- The ARPA Services software must be loaded on your local PC. Netsetup configures the ARPA Services software to load automatically each time you turn on your PC.
- You must know the names or IP addresses of the remote computers with which you want to communicate. The "Identifying Computers on the Network" section in this chapter has more information about remote computernames and IP addresses.

The ARPA Services

The HP **ARPA Services** product allows you to do the following:

- Use **TELNET** to log on remote computers. Once you log on a remote computer, it is as though you have a direct terminal connection to it.
- Use **FTP** (file transfer protocol) to transfer files between your local PC and a remote computer (for example, HP-UX, UNIX®, or non-UNIX computers) that is also running ARPA Services.
- Use **RCP** (remote copy protocol) to copy files between your PC and an HP-UX or UNIX remote computer.
- Use **RSH** (remote shell) to connect to an HP-UX or UNIX remote computer and execute commands.
- Use **PING** to verify a physical connection to a remote computer to isolate trouble in network communications.
- Use third-party networking products, such as a data base or a terminal emulator, that work with HP ARPA Services 2.1.

Identifying Computers on the Network

Your local PC was assigned a unique IP address when the ARPA Services software was installed and configured. Every other computer on the network also has a unique name and IP address.

Computernames

You can identify a remote computer either by its **computername** or its IP address. A computername is often more convenient to use than an IP address, but a networked system must be able to match this name with a corresponding IP address in order to find the computer on the network. The ARPA Services software includes a computername resolver that matches names with IP addresses. (See the "Computername Resolution" section on the next page for more information.)

Use the exact name of the remote computer with which you want to communicate. If you try to use an ARPA Service with a name that cannot be resolved, you will receive an error message. If you are not sure of the remote computername, check with your network administrator.

IP Addresses

If you choose to supply an **IP address** instead of a computername, it must be the correct IP address of the remote computer or you will receive an error message. An IP address is four decimal integers separated by periods. For example, an IP address might be something like `18.9.0.87`.

Computername Resolution

A networked system has two ways to resolve a computername to its corresponding IP address:

- Over the network using the **domain name resolver** or HP PROBE.
- By using a local host table file.

Domain Name Resolution

When you specify a computername, the system first uses the domain name resolver (DNR) or HP PROBE to try to resolve the name to a matching IP address. The domain name resolver and HP PROBE are dynamic ways to resolve names to IP addresses, which means that they obtain current information about computers on the network from the network itself. The information is always up-to-date.

Note

If you are running NewWave applications that require HP's proprietary NetIPC application support, then read this note.

In the default configuration, HP PROBE only allows you to resolve computernames that are on the local LAN. In order to establish connections to computers not on the local LAN, you need to use the INETNAME utility. If you need this support, see Appendix D in the *Installation Guide: HP ARPA Services 2.1/MS-DOS*, which describes INETNAME in detail.

Host Tables

If the domain name resolver and HP PROBE are not configured or fail to resolve a computername to an IP address, the networked system tries a local host table file. This file is often created for local PC's at the time the ARPA Services software is installed and configured. It is often created by copying a host file from a UNIX system (usually in the `c:\etc\hosts` directory) and then editing the file for your needs.

The host table file is a local text file that lists the IP addresses and names of remote computers with which you want to communicate. If a host table file is used for name resolution, it should list all the names and IP addresses of the computers that you regularly access.

A host table file is static, which means that its information about computers on the network stays the same until the next time it is updated with new information.

Host Table Format

The ARPA Services host table format is the same as the format for host tables in 4.2 BSD UNIX. For example, the host file entry for a node with an address of `192.45.36.5` and a host name of `hpdxsg` looks like

```
192.45.36.5          hpdxsg
```

Your network administrator keeps host tables up-to-date and organized by:

- Updating a host table when a computer is changed, added, or removed.
- Listing the remote computers in priority order, with the ones used most often at the top of the file. This increases the speed of searches for the most often used host names.

DOS 5.0 Task Manager Considerations

If you are using DOS 5.0, there is no change in the way you use HP ARPA Services 2.1 for MS-DOS *unless* you plan to use the DOS 5.0 Task Manager. The key change with using the Task Manager shell is that you must load all files for networking services that you plan to use **before** you start the DOS 5.0 shell. For more information, see "Restrictions When Using the Task Manager" below.

Using the Task Manager

Using the DOS 5.0 Task Manager enables you to start one or more ARPA Services sessions, and then "hot key" back and forth between these sessions or other tasks using Ctrl-Esc. If you leave an ARPA Services session, it remains suspended (no networking activity takes place) until you return to it.

Restrictions When Using the Task Manager

The following restrictions apply when using the DOS 5.0 Task Manager:

- You may not load any TSRs from within the shell. If you try to do this, the following error message is displayed when the files try to load:

  ```
  NET0131: <program_name> must be loaded before the DOS 5.0 shell is started.
  ```

- If you selected the Demand Load option during ARPA installation with Netsetup, reconfigure the option to `no` before you try to load any networking application files (i.e., FTP) from within the shell. This is because you cannot demand load the ARPA Services from within the shell.

- You may not unload network files from within the shell. However, you may check the status of network files with the `hpunload /status` command.

- If using Kermit with the DOS 5.0 Task Manager shell, you may manage only one Kermit session at a time. Within the single Kermit session, you can establish multiple connections at one time, using MUI (Minimal User Interface). (Refer to the "Multiple Session Management" section later in this chapter for more information on MUI.)

Note Terminal emulators that have internal session management allow you to establish multiple connections at the same time. Terminal emulators without internal session management require you to manage sessions from the MUI (Minimal User Interface) prompt.

Using Memory Management Utilities

Use of memory management utilities with DOS 5.0 is the same as with previous versions of DOS. Refer to the *Installation Guide: HP ARPA Services 2.1/MS-DOS* for more information.

2

Using TELNET for Remote Login

Chapter Overview

TELNET allows you to log on a remote computer as though the PC is a terminal attached to the computer.

This chapter describes the following information:

- What you need to know before you use TELNET.
- How to use TELNET including how to connect, disconnect, and manage multiple sessions.

Before You Begin

Before you can use TELNET, the HP ARPA software must be correctly installed, configured, and loaded on your local PC. In addition, the remote computer must be running the TELNET service.

TELNET uses the Bridge Application Program Interface (BAPI) interface that allows emulators and other applications to communicate with computers on a network.

When you specify a remote computer to connect to, you need to refer to its IP address or computername. If you specify a computername, the software queries the network for the remote computer IP address. If the IP address is not found, the local host table is checked for a matching name and IP address.

Supported Emulators

The Kermit 3.0 terminal emulator is included with the ARPA Services 2.1 product as a convenience to our customers. Kermit is a public domain VT100 emulator that comes from Columbia University. Since it is included with ARPA 2.1, you do not need to purchase another terminal emulator unless more features are needed. However, Kermit software is not supported by HP.

Note If you are using Kermit with the DOS 5.0 Task Manager shell, or in the Windows 3.*x* environment, you may manage only one Kermit session at a time. Within the single Kermit session, you can establish multiple connections at one time, using the MUI (Minimal User Interface). Refer to the "Multiple Session Management" section later in this chapter for more information on the MUI.

If you prefer, you may use other emulators that run over the BAPI interface. HP supported terminal emulators include:

- HP Advancelink for DOS (Product Number 68333F).
- HP Advancelink for Windows/NewWave (Product Number D2104B, Version A.03.02 or greater).

In addition, numerous third-party terminal emulators will work. However, third-party products are not supported by HP.

For more information on running terminal emulators in Windows 3.*x*, see "Using Terminal Emulators" in Appendix B, "Using ARPA Services with WIndows 3.*x*."

Using TELNET

This section describes how to use TELNET to connect to a remote computer, manage multiple sessions, and disconnect from a remote computer.

The information in this section describes use of TELNET with the Kermit terminal emulator. If you are using another emulator, operation may be different than described here; refer to the manual for your terminal emulator for complete instructions.

If you need to change the configuration of the Kermit emulator, refer to the file called `KERMIT.DOC`.

Connecting to a Remote Computer

To connect to a remote computer, follow these steps:

1. Do one of the following:

   a. If you are using the TELNET batch file that invokes the Kermit terminal emulator, type a command of the following format:

      `telnet (`*`computername`* `or` *`IP address`*`)`

      The *computername* or *IP address* are the computername and IP address of the remote computer.

   b. If you are using an emulator other than Kermit, or if you start Kermit without a batch file, start the terminal emulator.

2. Log on the remote computer.

Multiple Session Management

For Kermit and many other terminal emulators, TELNET provides multiple session management through MUI (Minimal User Interface). Session management commands are issued at the MUI prompt. To display the MUI prompt, use the Enter Command Mode (ECM) keystroke. The default ECM keystroke is Ctrl-^.

Note

The ECM keystroke is configured during installation. The default, Ctrl-^, is used in these instructions. If the ECM keystroke is different than this for your PC, you should replace it with whatever was set up by the person that configured the ARPA Services software.

The ECM keystroke is contained as part of the `PROTOCOL.INI` file. For more information about this file, refer to *Installation and Tuning Guide: HP ARPA Services 2.1/MS-DOS*.

Once the MUI prompt is displayed, you can open sessions to other remote computers while still maintaining the current session to the remote computer.

Type ? at the MUI prompt to display the MUI help menu. The MUI help menu looks like the following:

```
Connect       hostname or hostaddress
Disconnect    session number
Resume        session number
Modify        session number
Show Session
To quit, type the exit command appropriate for your emulator.
```

Descriptions of each of these menu options follow. Note that you need only type the first letter of each option.

Option	Description
C[onnect]	Connects your PC workstation to a remote computer, as specified by a *hostname* or *hostaddress*.
D[isconnect]	Requests that the specified *session number* is disconnected. If you do not specify a session number, the current session is disconnected.
R[esume]	Requests that communications with the specified *session number* be resumed. If you do not specify a session number, the current session is resumed.
M[odify]	Displays configuration information about the specified *session number*, that you may modify. Press the SPACE BAR to choose new choices for End-of-line mapping. When the desired choice is displayed, press Enter. The configuration is changed, and the MUI prompt is re-displayed. If you do not specify a session number, the current session is modified.
S[how Sessions]	Displays a list of active connections between your PC workstation and other computers. The current session is identified by an asterisk (*) next to it.

If your emulator uses MUI, you may access it by issuing the Ctrl-^ (or appropriate ECM) keystroke. You can then open another session, as described above. If the emulator you are using handles its own session management, you must refer to the user's guide for the emulator for information about establishing multiple connections.

Disconnecting from a Remote Computer

You disconnect from a remote computer in one of two ways:

- By logging off from the remote computer.
- By accessing MUI and disconnecting from the remote computer.

The following steps describe how to disconnect from the remote computer from MUI.

Note Depending on the emulator you are using, when you log off, you may or may not be exited from the emulator.

---

1. At the remote computer's prompt, press Ctrl-^ (or the appropriate ECM keystroke).

   The MUI prompt is displayed.

2. (This step is optional.) To show the current sessions and their corresponding session numbers, type the following at the MUI prompt:

   ```
   Show Sessions
   ```

   OR

   ```
   S
   ```

3. Disconnect from the remote computer by typing a command of the following format:

   ```
   Disconnect nn
   ```

   OR

   ```
   D nn
   ```

   nn is the number of the session you want to close. If you do not specify *session number*, the current session is closed.

   The specified connection is closed.

Exiting the Emulator

The process to exit an emulator is dependent on the emulator you are using. These steps describe the process if you are using Kermit.

To exit the terminal emulator, follow these steps:

1. Disconnect all open sessions, as described in the previous section.
2. Exit the terminal emulator. If you used Kermit, press ALT-X.
   a. If you used the TELNET batch file to start TELNET, you are now returned to DOS.
   b. If you did *not* use the batch file to start TELNET, return to DOS by typing:

      ```
      exit
      ```

3

Using FTP to Transfer Files

Chapter Overview

The FTP ARPA Service uses the standard ARPA Service file transfer protocol. This chapter explains how to use FTP to perform the following tasks:

- Transfer text and binary files between your local PC and other computers that support ARPA Services.
- List directories on a remote computer.
- Delete and rename files on a remote computer.
- Create and remove directories on a remote computer.
- Display remote files on your local PC.

This chapter describes how to use FTP. It also includes an "FTP Command Reference" section that describes each command in detail.

Before You Begin

Before you can use FTP, the ARPA Services software must be loaded.

Ways to Use FTP

You can issue an FTP command in one of two ways:

- Specify an FTP command from the DOS prompt. This is called FTP command-line mode.
- Open an FTP connection to a remote computer and interactively issue commands until closing the connection. This is called FTP interactive mode.

Filenames

When possible, choose filenames that are legal for both DOS and the remote computer(s) with which you are communicating.

Using FTP Command-line Mode

If you want to issue an `FTP` command without opening an interactive connection, use the `FTP` command followed by command-line options and an FTP command to perform a function. This section describes how to use FTP command-line mode.

Use a command of the following format to issue the `FTP` command in command-line mode:

`ftp` [*command_line options*] *remote_computer* [*command*]

command-line options can be specified after the `FTP` command. They can be in any order and must each be separated by at least one space. Command-line options consist of a hyphen (-) followed by a single letter (case-sensitive) and arguments.

Option	Description
remote_computer	The hostname or IP address of a remote computer.
command	The FTP command you want to issue, including the correct syntax. If a command is specified, FTP assumes command-line mode. The requested command is executed and then the DOS prompt is displayed. If an FTP command is *not* specified or if the FTP command is invalid, FTP enters interactive mode and the FTP prompt is displayed.
`-a`	Sets the file transfer type to ASCII (this is the default setting).
`-b`	Sets the transfer type to binary.
`-c`	Sets `CTRLC` on to enable Ctrl-C interruption of file transfers.
`-d`	Sets `DEBUG` on to show debugging information.
`-g`	Sets `GLOB` off to disable wildcard expansion.
`-h`	Sets `HASH` on to display hash marks during file transfer.
`-i`	Sets `PROMPT` off to disable prompting for multiple file transfers and file deletions.
`-l`	Sets `BELL` on to sound the bell after a file transfer is complete.
`-p`	Sets `SENDPORT` off so `PORT` commands are not sent to the server.
`-s`	Sets `SLASHFLIP` off so backslashes (\) are *not* converted to slashes (/) on outgoing commands.
`-u` *username password*	The `-u` option must be followed immediately by *username* and *password*, each separated by a space. This login information establishes a connection to the specified remote computer. The *username* and *password* are those on the remote computer. If you do not specify the `-u` option, FTP prompts for your username and password. If you do not want other users to see your password, you can omit it and FTP prompts for it.
`-v`	Sets `VERBOSE` off so FTP function status messages are not displayed.

Command-line Example

Bob wants to transfer the file `test.c` from the remote computer, `hpcndl`, to a file called `test2.c` on his local PC. Bob's username is `bob` and his password is `oct8`. To transfer the file, Bob types:

```
ftp -u bob oct8 hpcndl GET test.c test2.c
```

The remote file `test.c` is transferred to the current directory of Bob's local PC in the file called `test2.c`.

The screen displays the steps FTP goes through while transferring the file. Once the file is transferred, the screen displays the following:

```
Transfer complete.
Goodbye.
```

The DOS prompt is then displayed on the screen.

Using FTP Interactive Mode

To begin an interactive FTP session with a remote computer, use the following command format:

```
ftp [command_line options] remote_computer
```

command-line options can be specified after the FTP command. They can be in any order and must each be separated by at least one space. Command-line options consist of a hyphen (-) followed by a single letter (case-sensitive) and arguments.

Option	Description
remote_computer	The hostname or IP address of a remote computer.
-a	Sets the file transfer type to ASCII (this is the default setting).
-b	Sets the transfer type to binary.
-c	Sets CTRLC on to enable Ctrl-C interruption of file transfers.
-d	Sets DEBUG on to show debugging information.
-g	Sets GLOB off to disable wildcard expansion.
-h	Sets HASH on to display hash marks during file transfer.
-i	Sets PROMPT off to disable prompting for multiple file transfers and file deletions.
-l	Sets BELL on to sound the bell after a file transfer is complete.
-p	Sets SENDPORT off so PORT commands are not sent to the server.
-s	Sets SLASHFLIP off so backslashes (\) are *not* converted to slashes (/) on outgoing commands.

`-u` *username* *password*

The `-u` option must be followed immediately by *username* and *password*, each separated by a space. This login information establishes a connection to the specified remote computer. The *username* and *password* are those on the remote computer. If you do not specify the `-u` option, FTP prompts for your username and password.

If you do not want other users to see your password, you can omit it and FTP prompts for it.

`-v`

Sets `VERBOSE` off so FTP function status messages are not displayed.

If you do not specify the remote computername (or IP address), username or password, FTP prompts for them. Once you are connected to the remote computer, information similar to the following is displayed:

```
HP FTP Version B.00.00, 03/01/91
Copyright (c) Hewlett-Packard Co., 1989, 1990, 1991. All rights reserved.

Connected to hostname
```

When FTP starts, if you have not already supplied a login name and password, you are prompted for them. Some remote computers may also require an account name, and FTP prompts for one if necessary.

When FTP is ready to accept commands, it displays a prompt similar to one of the following:

`ftp:`*COMPUTER_NAME*`>`

or

`ftp:`*nnn.nnn.nnn.nnn*`>`

or

`ftp:>`

In the above display, *nnn.nnn.nnn.nnn* is the remote computer's IP address.

Using FTP Commands

This section describes the FTP commands and their uses.

Some FTP commands such as MKDIR, RMDIR and PWD, may not work with some remote computers. This is usually because the remote computer does not implement these commands. Contact the system administrator of the remote computer for more information.

Some commands require filename arguments. If the arguments are not typed on the command line, FTP prompts for them. If a command takes two filename arguments, they must be separated by a space. For more information on use of commands, see the "FTP Command Reference" section of this chapter.

Transferring Binary Files

If you are sending or receiving binary files, you need to do either of the following:

- Use the BINARY command to change the file type after you log on the remote computer. Or specify the -b command-line option.
- Use BPUT or BGET to transfer the file. This temporarily changes the file type to binary for the duration of the transfer.

If you don't use the BINARY, BPUT, or BGET commands, some files may not transfer properly.

Getting Help

You can display a summary of FTP commands with either the FTP HELP command or the ? command. You can also get information about just one command.

Listing All Commands

At the FTP prompt, type:

```
help
```

or

```
?
```

The list of commands is displayed.

Information About An Individual Command

To get information about an individual command, at the FTP prompt, type the HELP command in the following format:

```
help command
```

or

```
? command
```

command represents the FTP command for which you want information. The screen displays a line of information about the command.

Example

Carole wants information about the CD command. To get the information, she types:

```
help cd
```

The screen displays:

```
cd directory change remote working directory
```

Quick Reference to Commands

Table 3-1 lists frequently used FTP commands and their uses. For a complete listing of and detailed information about FTP commands, refer to the "FTP Command Reference" section later in this chapter.

Table 3-1. FTP Commands Quick Reference

To do this	Use this command
Change file type to ASCII.	ASCII, TYPE ASCII
Change file type to binary.	BINARY, TYPE BINARY
Change local working directory.	DRIVE, LCD
Change remote working directory.	CD
Create directory on remote computer	MKDIR
Delete file on remote computer	DELETE, MDELETE
Display contents of a remote text file.	SHOW
Display help information.	HELP, ?
Exit from FTP.	QUIT, BYE
List files on local PC.	LDIR, LLS
List files on remote computer.	DIR, MDIR, LS, MLS
List working directory on local PC.	LPWD
List working directory on remote computer.	PWD
Remove directory on remote computer.	RMDIR
Rename file or directory on remote computer.	RENAME
Temporarily return to DOS command interpreter.	!
Transfer file to local PC.	GET, BGET, MGET, LAPPEND
Transfer file to remote computer.	APPEND, PUT, BPUT, MPUT

Transferring Files

Files can be transferred to and from the remote computer.

Note File timestamps are not preserved in either direction. The time the file is received is the time that is stamped.

Filename Information

If you do not specify a second filename in a file transfer command operation, FTP uses the base name of the file. That means that only the filename and not the path and drive specifiers are used for the filename. The file is then placed in the working directory of the remote computer to which you are transferring the file.

File Transfer Types

The file type is specified with either the `ASCII` or `BINARY` command.

- The `ASCII` command sets the file transfer type to ASCII. This is the default.
- The `BINARY` command sets the file transfer type to binary.

The file transfer type must match the type of file being transferred.

Displaying the Current File Type

To find out the current file transfer type, use the `TYPE` command. At the FTP prompt, type:

```
type
```

The screen displays the current file type.

Setting the File Type to Binary

To set the type to binary, at the FTP prompt, type:

```
binary
```

Transferring a File from a Remote Computer

The following commands can be used to transfer files from the remote computer to your local PC:

- BGET changes the type to binary for the duration of the transfer and changes it back on completion.
- GET or RECV copies a file in the current type (ASCII or binary).
- LAPPEND appends a remote file to a local file.
- MGET copies multiple files from the remote computer to the local PC, in the current type. The wildcard syntax is interpreted by the remote computer.

To transfer a file, use the following command format, at the FTP prompt:

command [*remote_filename*] [*local_filename*]

Each option is described as follows:

Option	Description
command	One of the FTP file transfer commands used to transfer a file from the remote computer to your local PC: BGET, GET, LAPPEND, MGET, or RECV.
remote_filename	The name of the file on the remote computer, including the pathname to the file. The pathname should be either the absolute pathname or the pathname relative to your current working directory on the remote computer.
local_filename	The name of the file on your local PC, including the pathname to the file or relative path to the current directory.

You are prompted for the remote filename if you do not provide it. Spaces are not allowed in filenames. The default filename for the local file is the same name used for the *remote_filename*, excluding path and directory specifications.

Example

Julia wants to transfer the ASCII file, `chart.asc`, from the remote computer to her local PC. The type is already set to ASCII.

She types:

```
get chart.asc chart1.asc
```

If `VERBOSE` is on, once the transfer is complete, the screen displays:

```
Transfer complete.
```

After the message, the FTP prompt is displayed.

Example

Ted wants to transfer a binary file called `work.2` from the remote computer to his local PC. The type is set to ASCII, and he doesn't want to change the type.

He types:

```
bget work.2
```

Because Ted didn't specify a local filename, FTP assumes the default local filename of `work.2`.

Example

Ted now wants to transfer all binary files with the extension `.EXE` from the remote computer to his local PC. To do so, he first changes the file transfer type to binary by typing:

```
binary
```

He transfers the files by typing:

```
mget *.exe
```

Transferring a File to a Remote Computer

The commands used to transfer files from a local PC to a remote computer are the following:

- APPEND appends a local file to a file on the remote computer, according to the current type.
- BPUT changes the type to binary for the duration of the transfer and changes it back on completion.
- MPUT copies multiple files to the remote computer from the local PC, in the current type. The wildcard syntax is interpreted by the local PC, using the PC standard wildcard syntax.
- PUT or SEND copies the file in the current type (ASCII or binary).

To transfer a file, use the following command format:

command [*local_filename*] [*remote_filename*]

Option	Description
command	One of the FTP file transfer commands for transferring a file from your local PC to the remote computer: APPEND, BPUT, MPUT, PUT, or SEND.
local_filename	The name of the file on the local PC, including the path to the file. You are prompted for the local filename if you do not provide it. Spaces are not allowed in filenames.
remote_filename	The name of the file on the remote computer, including the path to the file. If the filename is not specified, then the default filename is the same as the *local_filename*, excluding path and directory specifications. Spaces are not allowed in filenames.

Example

Julia wants to transfer an ASCII file called `plan.asc` from her local PC to the remote computer. The type is already set to ASCII.

Julia types:

```
put plan.asc plan2.asc
```

When the transfer is complete, the screen displays:

```
Transfer complete.
```

The FTP prompt is displayed after the message.

Example

Carole wants to transfer a binary file called `map.3` from her local PC to the remote computer. The type is set to ASCII, so she needs to change the file type by typing:

```
binary
```

To transfer the file, Carole types:

```
put map.3
```

Because Carole didn't specify a remote filename, FTP assumes the default filename of `map.3`.

The following is displayed on the screen once the transfer is complete:

```
Transfer complete.
```

Deleting Remote Files

The `DELETE` and `MDELETE` commands delete a file or multiple files on the remote computer.

To delete a file, type the `DELETE` command in the following format:

`delete` *filename*

The *filename* is the name of the file you want to delete.

`MDELETE` deletes multiple files at once. If prompting is set `on`, you are prompted (`y` or `n`) prior to the deletion of each file. If you do not specify a filename, FTP prompts for one. `MDELETE` allows the use of the wildcard syntax of the remote computer.

Note To delete a local file, you must return to the DOS prompt.

Example

Bob wants to delete a file called `plan.2` from the remote computer.

He types:

```
delete plan.2
```

Because prompting is set on, Bob is prompted with:

```
Delete plan.2?
```

Once the file is deleted, the screen displays:

```
DELE command okay.
```

Example

Ted wants to delete all `C` source files in the current directory on the remote computer.

He types:

```
mdelete *.c
```

Because prompting is set on, Ted is prompted with:

```
Delete prog1.c?
Delete prog2.c?
```

(For information about prompting, see the "Comments" section of the MDELETE command in the FTP Command Reference section later in this chapter.)

After each file is deleted, the screen displays:

```
DELE command okay.
```

Performing Directory Operations

FTP has commands to change the default working directory and to create and remove directories. More information about the syntax and use of these commands can be found in the "FTP Command Reference" section later in this chapter.

Changing the Current Working Directory

The CD command changes the current working directory on the remote computer. The LCD command changes the current directory on the specified drive of the local PC.

Displaying the Current Working Directory

The PWD command displays the current working directory on the remote computer. The LPWD command displays the current working directory on the local PC.

Creating Directories and Removing

The MKDIR and RMDIR commands respectively create and remove directories on the remote computer.

To create and remove local directories, temporarily return to DOS, as described in the next section, "Temporarily Returning to DOS," and use the DOS MKDIR and RMDIR commands.

Temporarily Returning to DOS

Once you open an FTP connection, you might want to perform a DOS command (for example to create or remove a directory on the local PC without exiting FTP). To do this, you can use the FTP ! command to temporarily return to DOS.

To temporarily return to DOS, type:

```
!
```

The following message is displayed on the screen:

```
Warning, some programs will interfere
with network communicator and can cause
lost connections. Do not run any
network programs from this DOS shell.
Type 'EXIT' to return to FTP.
```

After this message, the DOS prompt is displayed. You can now invoke DOS commands. When completed, return to FTP by typing:

```
exit
```

The FTP prompt is displayed on the screen.

Exiting FTP

To exit and disconnect from FTP, type either of the following:

```
quit
```

or

```
bye
```

This disconnects from the remote computer, exits from FTP, and returns to DOS.

Additional FTP Features

The following additional FTP features are described in this section:

- Redirecting remote computer information.
- Using the FTP TAKE command.

Redirecting Remote Computer Information

Redirection allows you to take the output resulting from one command and capture it in a local file. You can use redirection with any of the FTP commands that send information to your local PC screen. This means you can redirect information from the following commands to a local file: LS, DIR, MDIR, MLS, and SHOW.

For example, if you want to redirect information from the LS command, type:

```
ls > listfile
```

The *listfile* option is the name of the local file for which you want the information listed.

If you want to redirect and append to an existing file, type:

```
ls >> listfile
```

Using the FTP TAKE Command

You can use the FTP TAKE command to get FTP to read commands from a local file instead of from the keyboard.

This section shows an example of doing backups using the TAKE command. With the TAKE command, MGET and MPUT commands can be issued to transfer many files at a time. In a TAKE file, lines with a # in the first column are read as comments and are not executed.

Issuing the TAKE Command From the Local PC Prompt

Use the TAKE command from the local PC prompt command line with the following options:

```
ftp -u username password remote_computer take filename
```

HP Computer Museum
www.hpmuseum.net

For research and education purposes only.

A description of each option follows:

- `-u` indicates that you are supplying the *username* and *password* on the command line.
- *username* is your username.
- *password* is your password.
- *remote_computer* is the remote computer on which you want to issue the command.
- *filename* is the name of the `TAKE` file.

Issuing the TAKE Command From FTP

Use the `TAKE` command with the following option from the FTP prompt:

```
take filename
```

The *filename* is the name of the `TAKE` file.

Example

A sample TAKE file, called BACKUP.TAK, used for doing backups, might look like the following:

```
################################################
#     Changing remote directory to backup      #
################################################
cd /usr/backup
################################################
#     Changing local directory to backup       #
################################################
lcd /backup
################################################
#     Transferring all files ending in .h to   #
#     remote computer                          #
################################################
prompt off
mput *.h
################################################
#     Changing remote directory to bttcp       #
################################################
cd bttcp
################################################
#     Changing local directory to bttcp        #
################################################
lcd bttcp
################################################
#     Transferring all files ending in .h to   #
#     remote computer                          #
################################################
prompt off
mput *.h
```

Ctrl-C and Ctrl-Break Handling in FTP

When using FTP in interactive mode (entering commands at the FTP prompt), the processing of Ctrl-C and CTRL-Break keystrokes is different than in many standard DOS programs. FTP intercepts these keystrokes, interrupts the command it is currently executing, and returns to the FTP command prompt. It does *not* abort the program. A Ctrl-C or CTRL-Break at the command line causes the command line to be reset to the basic FTP prompt. By default, FTP does not interrupt file transfer when a Ctrl-C is detected. FTP interrupts commands that operate on multiple files, such as MPUT and MGET by interrupting these commands in between the transfer of files.

If you want to be able to interrupt FTP during file transfer, specify the -c command-line option at the FTP prompt or use the FTP CTRLC command. This option enables Ctrl-C processing in the middle of a file transfer. It allows you to interrupt FTP file transfer, but it also may cause some problems, as described below:

- If a GET, BGET, MGET, RECV, or LAPPEND file transfer command is in progress, a Ctrl-C keystroke interrupts the file transfer and causes the FTP session to be aborted. You would then be returned to the FTP command prompt, and you would need to reconnect to the remote computer.
- If a PUT, BPUT, MPUT, SEND, or APPEND file transfer command is interrupted with Ctrl-C, the remote computer might report an error, but the FTP session should not be disrupted.

Caution Interrupting FTP in the middle of ANY file transfer causes the destination file to be left in some indeterminate state. Therefore, any file transfer interrupted by Ctrl-C must be considered to have corrupted the destination file.

The ability to interrupt FTP while a file transfer is in progress is useful when FTP is being used to transfer large files, or when FTP is being used to transfer files to or from a remote computer with a slow rate of data transfer. In general use, FTP transfers files quickly enough that it is unnecessary to interrupt file transfer.

Exit Code Information

FTP normally returns to DOS with an exit code of 0. If an error occurs that causes FTP to return to DOS, an exit code of 1 is returned. This is primarily useful if FTP is used in batch files, where the exit code can be evaluated by using the ERRORLEVEL construct. Refer to the DOS user's guide for your PC for more information about exit codes.

FTP Command Reference

This section contains an alphabetical listing of the FTP commands with their syntax and purpose. Examples of using the commands are included.

You can use unambiguous abbreviations of commands instead of the full command name. For example, `DI` is the same as the `DIR` command, but `D` is ambiguous because it could be an abbreviation of `DEBUG`, `DELETE`, or `DIR`.

You can issue FTP commands in either uppercase or lowercase.

APPEND

Purpose

APPEND copies and appends a local file to a remote file.

Syntax

```
append [local_filename] [remote_filename]
```

Option	Description
local_filename	The name of the local file you want to append to the remote file. FTP prompts for this filename if you do not specify it.
remote_filename	The name of the remote file to which you want the local file appended. FTP prompts for this filename if you do not specify it.

Comments

The file is transferred in the current file transfer type.

Example

If Bob wants to append the local file called `info.doc` to the remote file called `old.doc`, he types:

```
append info.doc old.doc
```

This appends `info.doc` to `old.doc`.

ASCII

Purpose

The `ASCII` command changes the file transfer type to ASCII. This type must be used for ASCII text files. When you start FTP, the transfer type defaults to ASCII.

Syntax

```
ascii
```

Example

If Carole wants to transfer an ASCII file, but needs to first change the file type to ASCII (she had previously set it to binary) she types:

```
ascii
```

BELL

Purpose

The BELL command turns the bell sound off and on. When BELL is set on, FTP beeps when file transfers are complete.

Syntax

```
bell [on | off]
```

Option	Description
on	Turns on the bell sound. When you select on, FTP beeps when file transfers are complete.
off	Turns off the bell sound. On startup, BELL defaults to off.

Comments

If you do not specify on or off, the command turns the bell off if it is on and turns the bell on if it is off.

Example

Ted is transferring long files from a remote computer to his local PC. He wants to work on other things while this is going on, but also wants to know when the transfers are complete. If he turns the BELL command on, his local PC will beep at him after a file transfer is complete. Ted types:

```
bell on
```

BGET

Purpose

The BGET command copies a remote file to a local file using the binary transfer type.

Syntax

```
bget [remote_filename] [local_filename]
```

Option	Description
remote_filename	The name of the remote file that you want to transfer to your local PC. FTP prompts for this filename if you do not provide it.
local_filename	The name of the local file to which you want the file transferred. If you do not specify this filename, then FTP uses the same name as the *remote_filename*, removing any path or drive specifiers.

Comments

After the file transfer is complete, the file transfer type reverts to what it was prior to execution of the BGET command.

Example

Bob needs to transfer a binary file from the remote computer, but he does not want to first execute the BINARY command. He can do this by using the BGET command instead. Bob types:

```
bget samp.exe
```

Because Bob did not specify a *local_filename*, it will default to the same name as the remote filename (samp.exe).

See Also

The GET command. The BGET command works the same as the GET command except that it copies *remote_filename* to *local_filename* using BINARY transfer type.

BINARY

Purpose

The `BINARY` command changes the file transfer type to binary.

Syntax

```
binary
```

Comments

This transfer type must be used for binary files, but it may not transfer text (ASCII) files correctly between two different types of computers (for example an HP-UX system and a PC).

Example

Carole wants to change the file type to binary. She types:

```
binary
```

BPUT

Purpose

The BPUT command copies a local file to a remote computer using the binary transfer type.

Syntax

```
bput [local_filename] [remote_filename]
```

Option	Description
local_filename	The name of the local file that you want to transfer to the remote computer. FTP prompts for the local filename if one is not specified.
remote_filename	The name for the file once it is transferred to the remote computer. If you do not specify this filename, FTP uses the same name as the *local_filename*, removing any path or drive specifiers.

Comments

After the transfer is complete, the transfer type reverts to what it was prior to execution of BGET.

Example

Alice is transferring a binary file from her local PC to the remote computer. She types the following command:

```
bput code.exe code1.exe
```

See Also

The PUT command. The BPUT command works the same as the PUT command except that it copies *remote_filename* to *local_filename* using BINARY transfer type.

BYE

Purpose

The BYE command exits FTP and returns to the local PC operating system.

Syntax

```
bye
```

Comments

If there is an open connection to a remote computer, it will be closed before FTP is exited.

Example

Ted wants to exit FTP and return to DOS. He types:

```
bye
```

See Also

The QUIT command. The BYE command works the same as the QUIT command.

CD

Purpose

The CD command changes the current working directory on the remote computer.

Syntax

```
cd [pathname]
```

Option	Description
pathname	A directory on the remote computer. If *pathname* is not specified, FTP prompts for it.

Example

Julia wants to change the current working directory on the remote computer to another subdirectory under the current parent directory. To do this, she types the following:

```
cd .. /data
```

CLOSE

Purpose

The CLOSE command closes a connection to a remote computer without exiting FTP.

Syntax

```
close
```

Comments

Terminates the currently open session to the remote computer without exiting FTP. When you do this, you are still in FTP, but you have closed the virtual connection to the remote computer. Use this command to close a connection to one remote computer and then use the OPEN command to open a connection to another remote computer.

Example

Bob wants to close the current FTP session. To do this, he types:

```
close
```

The screen displays:

```
Goodbye.
```

The FTP prompt is displayed on the screen:

```
ftp>
```

Bob can now use the OPEN command to open a connection to another remote computer.

See Also

The OPEN command. After using the CLOSE command to close the connection, use the OPEN command to open a connection to another remote computer.

CTRLC

Purpose

The CTRLC command, when set on, enables Ctrl-C interruption of file transfers.

Syntax

```
ctrlc [on | off]
```

Option	Description
on	Specifies that you want to enable Ctrl-C interruption of file transfers.
off	Specifies that you want to disable Ctrl-C interruption of file transfers. On startup, CTRLC defaults to off.

Comments

The default of this command is off. That means that any Ctrl-C interrupt that occurs during a file transfer is deferred until after the data transfer is complete. If the option is set on, a Ctrl-C interrupt immediately interrupts data transfer.

If you do not specify on or off, the command turns CTRLC on if it is off and turns CTRLC off if it is on.

Example

Bob wants to allow the Ctrl-C keys to interrupt a file transfer in progress. He types the following command to enable CTRLC:

```
ctrlc on
```

See Also

The "Ctrl-C and CTRL-Break Handling in FTP" section in this chapter.

DEBUG

Purpose

The DEBUG command enables you to examine the FTP protocol commands being sent between your local PC and a remote computer.

Syntax

```
debug [on | off]
```

Option	Description
on	Specifies that DEBUG be turned on.
off	Specifies that DEBUG be turned off. On startup DEBUG defaults to off.

Comments

The information you receive from this command is dependent on the types of computers being used for the FTP transfer. It can be helpful when trying to debug a problem.

If you do not specify on or off, the command turns DEBUG off if it is on and turns DEBUG on if it is off.

Example

If Carole wants to view the FTP protocol commands being sent by her local PC, she can use the DEBUG command. This allows her to see if, for example, a command is being sent that the remote computer does not recognize. Carole types:

```
debug on
```

This sets DEBUG on. Now Carole issues the LS command:

```
LS
```

Output similar to the following is displayed prior to the directory listing:

```
--->PORT 255,255,001,028,200,1
200 PORT command okay.
--->NLST
150 Opening data connection for /bin/ls (255.255.001.028,51201) (0 bytes).
```

DELETE

Purpose

The DELETE command deletes a remote file.

Syntax

```
delete [filename]
```

Option	Description
filename	The name of the remote file that you want to delete.

Comments

If you do not specify a filename, FTP asks for it. If prompting is on, you are asked to verify the file that should be deleted.

Example

Ted wants to delete the file called prog.exe. To do this, he types:

```
delete prog.exe
```

Because prompting is set on, he is prompted with:

```
Delete prog.exe?
```

To which he types Y.

See Also

The MDELETE command for information about deleting multiple files.

DIR

Purpose

The DIR command displays a listing of a remote directory or file on the local PC screen.

Syntax

```
dir [pathname]
```

Option	Description
pathname	A directory or file on the remote computer that you want listed on your local PC screen.

Comments

If you do not specify a *pathname*, FTP lists the contents of the current working directory.

Example

If Bob wants to list the contents of his current remote working directory, he types:

```
dir
```

A listing of the directory is displayed.

See Also

The CD command for information about changing the current working directory on the remote computer.

DRIVE

Purpose

The DRIVE command changes the current disk drive on the local PC to a drive you specify.

Syntax

```
drive [drivespec]
```

Option	Description
drivespec	The drive to which you want to change.

Comments

If *drivespec* is not specified the DRIVE command displays the current local PC drive specification.

Example

Carole wants to find out what the current drive is on her local PC. To do this, she types:

```
drive
```

The screen displays:

```
Current drive is A:
```

See Also

The LCD command for information about changing the current working directory on the local PC.

GET

Purpose

The GET command copies a file from the remote computer to the local PC in the current file transfer type (ASCII or binary).

Syntax

```
get [remote_filename] [local_filename]
```

Option	Description
remote_filename	The name of the remote file that you want to transfer to the local PC. If you do not provide the name of the remote filename, FTP prompts for it.
local_filename	The name of the file on the local PC. If you do not specify this filename, then FTP uses the same name as the *remote_filename*, removing any path or drive specifiers.

Example

Alice wants to transfer the file new.dat from the remote computer to her local PC. To do this, she types:

```
get new.dat some.dat
```

This transfers the remote file new.dat to a file called some.dat on her local PC.

See Also

The RECV command. The GET command works the same as the RECV command.

GLOB

Purpose

The GLOB command enables you to specify whether or not wildcard expansion should be enabled for filenames for the multiple transfer and delete commands (MGET and MDELETE). Note that wildcard expansion is *always* enabled for directory commands (DIR, LDIR, LS, LLS, MDIR, and MLS).

Syntax

```
glob [on | off]
```

Option	Description
on	When you specify GLOB on, wildcards are expanded. On startup, GLOB defaults to on.
off	When you specify GLOB off, wildcards are not expanded.

Comments

If you do not specify on or off, the command turns GLOB off if it is on and turns GLOB on if it is off.

Example

Ted does not want wildcards to be expanded. To do this, he types the following:

```
glob off
```

The screen displays the following message:

```
Globbing off.
```

HASH

Purpose

The HASH command enables you to specify if hash marks (#) are printed during file transfers. Hash marks are used to indicate the progress of the transfer. Hash marks are especially useful during the transfer of long files.

Syntax

```
hash [on | off]
```

Option	Description
on	When you specify HASH on, hash marks are printed during file transfers.
off	When you specify HASH off, hash marks are not printed during file transfers. On startup, HASH defaults to off.

Comments

If you do not specify on or off, the command turns HASH off if it is on and turns HASH on if it is off.

Example

Carole is transferring large files. To be sure the transfers are progressing, she wants to see something on the screen as they are transferring. She sets HASH on by typing the following:

```
hash on
```

The following is displayed on the screen:

```
Hash mark printing on.
```

HELP

Purpose

The HELP command provides a list of available FTP commands or brief information about one or several FTP commands.

Syntax

```
help [command]
```

Option	Description
command	The name or names of FTP commands about which you want information. If you specify more than one command, they must be separated by spaces.

Comments

If you do not specify a *command*, a list of available commands is displayed.

Example

Julia wants to see a list of available FTP commands. To do this, she types:

```
help
```

A list of available commands is displayed on her local PC screen. She then decides to get help on the APPEND command. She types:

```
help append
```

The following is displayed on the local PC screen:

```
append local_file  remote_file append a local file to a remote file
```

See Also

The ? command. HELP works the same as the ? command.

LAPPEND

Purpose

The LAPPEND command copies and appends a remote file to a local file in the current file transfer type (ASCII or binary).

Syntax

```
lappend [remote_filename] [local_filename]
```

Option	Description
remote_filename	The name of the remote file that you want to append to the local file. If you do not specify a filename, you are asked for it.
local_filename	The name of the local file that you want the remote file appended to. If you do not specify a filename, you are asked for it.

Example

Bob wants to append the remote file `info.dat` to the local file `old.dat`. To do this, he types:

```
lappend info.dat c:\old.dat
```

See Also

The APPEND command for information about appending a local file to a remote file.

LCD

Purpose

The LCD command changes the current directory on the local PC.

Syntax

```
lcd [local_directory]
```

Option	Description
local_directory	The directory to which you want to change. If you do not provide *local_directory*, FTP prompts for one.

Comments

If you type a drive letter with the directory name, it changes the current working directory on that drive but does not change the current drive.

Example

Ted wants to change the current directory on the local PC. He types:

```
lcd
```

FTP prompts with:

```
To:
```

Ted types:

```
c:\work
```

The screen displays:

```
Local directory now C:\work
```

See Also

The DRIVE command for information about changing the current drive.

LDIR

Purpose

The LDIR command displays a local directory listing.

Syntax

```
ldir [local_directory]
```

Option	Description
local_directory	Specifies the directory you want to list.

Comments

If you do not specify a directory name, FTP lists the contents of the current local working directory.

Example

Carole wants to display a listing of the memos directory on her local PC. She types:

```
ldir c:\memos
```

A listing of the directory is displayed on the screen.

See Also

The DIR command for information on how to display a listing of a remote directory or file.

The LLS command for information on displaying a short directory listing of a local directory.

LLS

Purpose

The LLS command displays a short directory listing of a local directory.

Syntax

```
llS [local_directory]
```

Option	Description
local_directory	Specifies the local directory you want to list.

Comments

If you do not specify a local directory name, FTP lists the contents of the current local working directory.

Example

Carole now wants to display a short listing of the memos directory. She types:

```
llS memos
```

See Also

The LDIR command for information about displaying a local directory listing.

LPWD

Purpose

The LPWD command displays the drive and pathname of the current working directory on the local PC.

Syntax

```
lpwd
```

Example

Alice wants to display the drive and pathname of the current working directory for her local PC. She types:

```
lpwd
```

The screen displays:

```
Current directory is G:\alice
```

See Also

The LCD command for information about how to change the current directory on the local PC.

LS

Purpose

The LS command displays a short listing of a remote directory.

Syntax

```
ls [pathname]
```

Option	Description
pathname	Specifies the directory and/or files about which you want information listed.

Comments

If a directory name is specified, FTP lists the contents of that directory. If a filename is specified, FTP shows the remote filename. If no *pathname* is specified, FTP lists the contents of the current remote working directory.

Example

Ted wants to list all files with a .c extension in the current working directory. He types:

```
ls *.c
```

All the files in the current directory with the extension .c are listed.

See Also

The DIR command for information about displaying a listing of a remote directory or file.

The LLS command for information about displaying a short listing of a local directory.

MDELETE

Purpose

The MDELETE command deletes multiple files on the remote computer.

Syntax

```
mdelete [filenames]
```

Option	Description
filenames	A file or files that you want to delete. Multiple files can be specified using both wildcard syntax and/or by listing each file separately. Multiple filenames following the MDELETE command must be separated by at least one space. When specifying a set of files with wildcard syntax, the wildcard syntax is interpreted by the remote computer if the GLOB command is enabled. If no *filenames* are specified, FTP prompts for them.

Comments

If PROMPT is on, you are prompted before the deletion of each specified file. The valid responses follow:

Response	Description
s	No, do not delete the file.
p	Proceed, delete the file and all remaining ones without asking.
q	Quit, stop immediately, and do not delete the file. The FTP prompt is again displayed.
y	Yes, delete the file, but continue prompting for subsequent files.

Example

Bob wants to delete all files with the extension .obj in the current directory on the remote computer. The GLOB and PROMPT commands are both set on. Bob types:

```
mdelete *.obj
```

Because the PROMPT command is set on, FTP prompts before each file is deleted.

See Also

The GLOB command for information about wildcard syntax.

The PROMPT command for information about how to be prompted before files are deleted.

The DELETE command for information about how to delete a single file.

MDIR

Purpose

The MDIR command displays a listing of multiple remote files and/or directories.

Syntax

```
mdir [filenames]
```

Option	Description
filenames	Specifies the file, files, or directories you want to display. If you do not specify *filenames*, FTP prompts for it.

Example

Julia wants to display the remote files with the extensions .mem and .not and the directory \notes. To do this, she types:

```
mdir   \notes \*.mem   \notes\*.not
```

See Also

The DIR command for information about displaying a listing of a remote directory or file.

MGET

Purpose

The MGET command transfers multiple files to your local PC from the remote computer using the current file transfer type (ASCII or binary).

Syntax

```
mget [filenames]
```

Option	Description
filenames	The file or files you want to transfer from the remote computer to your local PC.

Comments

Multiple files can be specified by using both wildcard syntax and/or by listing each file separately. If a set of files are specified by wildcard syntax, the wildcard syntax is interpreted by the remote computer if the GLOB command is enabled.

Files or wildcard filenames listed after the MGET command must be separated by at least one space. If no filenames are specified, FTP prompts for them.

Destination filenames default to the basename of the file being transferred.

If PROMPT is on, you are prompted before the transfer of each specified file. The valid responses follow:

Response	Description
n	No, do not transfer the file.
p	Proceed, complete this transfer and all remaining ones without asking.
q	Quit, do not transfer the file, stop immediately.
r	Rename, ask for a new name for the local filename.
y	Yes, transfer the file, and continue to prompt for subsequent files.

As each file is transferred, information about the transfer is displayed on the screen.

Example

Ted wants to transfer all the files with the extensions .c or .h in the current working directory on the remote computer to his local PC. (GLOB and PROMPT are both set on.) To do this, he types:

```
mget *.h *.c
```

Because PROMPT is set on, FTP prompts before each file is transferred.

See Also

The GLOB command for information about wildcard syntax.

The PROMPT command for information about how to be prompted before files are deleted.

The GET command for information about transferring a single file to your local PC from the remote computer.

MKDIR

Purpose

The MKDIR command creates a directory on the remote computer.

Syntax

```
mkdir [remote_directory]
```

Option	Description
remote_directory	The name of the directory you want to create on the remote computer. If you do not specify the remote directory name, you are prompted for it.

Comments

Not all remote computers support this command.

Example

Alice wants to make a directory called data on the remote computer. To do this, she types:

```
mkdir data
```

The screen displays:

```
MKDIR command okay.
```

MLS

Purpose

The MLS command displays a short listing of specified remote files and/or directories.

Syntax

```
mls [remote_files]
```

Option	Description
remote_files	The files or directories you want displayed. If *remote_files* is not specified, FTP prompts for it.

Example

Carole wants to display a short listing of the files with the extensions .c and .h. To do this, she types:

```
mls *.c *.h
```

See Also

The DIR command for information about displaying a listing of a remote directory or file.
The LS command for information about displaying a short listing of a remote directory.

MPUT

Purpose

The MPUT command transfers multiple files from your local PC to the remote computer using the current file transfer type (ASCII or binary).

Syntax

```
mput [filenames]
```

Option	Description
filenames	A file or files you want to transfer from your local PC to the remote computer.

Comments

Multiple files may be specified using both wildcard syntax and/or by listing each file separately.

Files or wildcard filenames listed after the MPUT command must be separated by at least one space. If no filenames are specified, FTP prompts for them.

Destination filenames default to the basename of the file being transferred.

If PROMPT is on, you are prompted before the transfer of each specified file. The valid responses are:

Response	Description
n	No, do not transfer the file.
p	Proceed, complete this transfer and all remaining ones without asking.
q	Quit, do not transfer the file, stop immediately.
r	Rename, ask for a new name for the remote filename.
y	Yes, transfer the file.

As each file is transferred, information about the transfer is displayed on the screen.

Example

Bob wants to transfer all files with the extension .EXE from the current directory on his local PC to the remote computer. Since these are binary files, he first must set the file transfer type to binary using the BINARY command. The PROMPT command is set on.

To set the file transfer type to binary, he types:

```
binary
```

To transfer the files, he types:

```
MPUT *.exe
```

Because PROMPT is set on, he is prompted before each file is transferred.

See Also

The PROMPT command for information about how to be prompted before files are deleted.

The PUT command for information about transferring a single file from your local PC to the remote computer.

OPEN

Purpose

The OPEN command opens a session to a remote computer other than the one specified at startup.

Syntax

```
open [remote_hostname | IP_address]
```

Option	Description
remote_hostname	The name of the remote computer to which you want to open a connection. If you do not specify a *remote_hostname*, FTP prompts you to enter this information.
IP_address	The IP address of the remote computer to which you want to open a connection. If you do not specify an *IP_address*, FTP prompts you to enter this information.

Comments

Before you can use the OPEN command, you must use the CLOSE command to close the current session, if a current session is active.

Example

Ted wants to open a connection to the remote computer with the IP address 14.6.222.79. To do this, he types:

```
open 14.6.222.79
```

The connection is then opened and he is asked for his username and password (if one is required).

See Also

The CLOSE command for information on how to close an FTP connection.

PROMPT

Purpose

If PROMPT is on, you receive prompts prior to performing functions such as file deletions and multiple file transfers.

Syntax

```
prompt [on | off]
```

Option	Description
on	Specifies that you want to receive prompts prior to FTP performing functions, such as file deletions and multiple file transfers.
off	Specifies that you do not want to receive any prompts. On startup, PROMPT defaults to on.

Comments

If you do not specify on or off, the command turns PROMPT on if it is off and turns PROMPT off if it is on.

Example

Bob knows that the PROMPT command defaults to on when FTP is first started. However, he does not want to be prompted when making multiple transfers. To disable the prompting, he types:

```
prompt  off
```

The following is displayed on the screen:

```
Interactive prompting off.
```

PUT

Purpose

The PUT command copies a local file to the remote computer in the current file transfer type (ASCII or binary).

Syntax

```
put [local_filename] [remote_filename]
```

Option	Description
local_filename	The name of the file you want to transfer to the remote computer. FTP prompts for the local filename if it is not specified.
remote_filename	The name you want for the file once it has been transferred to the remote computer. If you do not specify this filename, FTP uses the same name as the *local_filename*, removing any path or drive specifiers.

Example

Julia wants to transfer the local file `file.dat` to the remote computer. She wants to change the file on the remote computer to `file1.dat`. To do this, she types:

```
put c:\data\file.dat file1.dat
```

See Also

The SEND command. PUT works the same as the SEND command.

PWD

Purpose

The PWD command displays the pathname of the current remote working directory.

Syntax

```
pwd
```

Comments

Not all remote computers support this command.

Example

Ted wants to display the pathname of the current working directory on the remote computer. To do this, he types:

```
pwd
```

The following is displayed:

```
"/users/ted" is the current working directory.
```

See Also

The LPWD command for information about displaying the drive and pathname of the current working directory on the local PC.

QUIT

Purpose

The QUIT command closes the current connection, exits FTP, and returns to DOS.

Syntax

```
quit
```

Example

Alice wants to close the current FTP connection and exit FTP. To do this, she types:

```
quit
```

See Also

The BYE command. QUIT works the same as the BYE command.

QUOTE

Purpose

The QUOTE command sends an FTP protocol command directly to the remote server.

Syntax

```
quote [protocol_command]
```

Option	Description
protocol_command	This specifies the protocol command and syntax you want to send directly to the remote server. If you do not specify a *protocol_command*, FTP prompts for one.

Comments

Before you execute this command, you should be familiar with FTP protocol and communications.

Example

Bob wants to execute the server NOOP protocol command. To do this, he types:

```
quote noop
```

The screen displays:

```
NOOP command okay.
```

See Also

The REMOTEHELP command to get a a listing of available protocol commands.

RECV

Purpose

The RECV command copies a file from the remote computer to the local PC in the current file transfer type (ASCII or binary).

Syntax

```
recv [remote_filename] [local_filename]
```

Option	Description
remote_filename	The name of the remote file that you want to transfer to the local PC. If you do not provide the name of the remote filename, FTP prompts for it.
local_filename	The name of the file on the local PC. If you do not specify this filename, FTP uses the same name as the *remote_filename*, removing any path or drive specifiers.

Example

Julia wants to transfer the file code.c from the remote computer to her local PC. To do this, she types:

```
recv code.c
```

This transfers the remote file code.c to a file of the same name on her local PC. (Because she did not specify a local filename, it defaulted to the same name as the remote filename.)

See Also

The GET command. RECV works the same as the GET command.

REMOTEHELP

Purpose

The REMOTEHELP command displays a list of the FTP protocol commands supported by the remote computer. This information is useful if you want to issue FTP protocol commands using the QUOTE command, and for debugging any compatibility problems.

Syntax

```
remotehelp [protocol_command]
```

Option	Description
protocol_command	Specifies that you want displayed a brief description of the specific protocol command.

Comments

If a protocol command is not specified, a list of all available commands is displayed. The REMOTEHELP command is not supported by all remote computers.

Example

Carole wants to list the protocol commands supported by the remote computer she is currently using. To find out, she types:

```
remotehelp
```

A list of supported commands is displayed.

See Also

The QUOTE command for information on issuing protocol commands.

RENAME

Purpose

The RENAME command renames a remote file.

Syntax

```
rename [old_name] [new_name]
```

Option	Description
old_name	Specifies the name of the remote file you want to change. FTP prompts for this filename if you do not provide it.
new_name	Specifies the new name for the file. FTP prompts for this filename if you do not specify it.

Comments

You can rename files on the local PC by using the ! command to temporarily return to DOS without exiting FTP, and then use system commands to rename the file.

Example

Alice wants to rename a remote file from `spec.dat` to `spec1.dat`. She types:

```
rename spec.dat spec1.dat
```

RMDIR

Purpose

The RMDIR command deletes a directory on the remote computer. The remote directory must be empty before you can remove it.

Syntax

```
rmdir [remote_directory]
```

Option	Description
remote_directory	Deletes *remote_directory* on the remote computer.

Comments

The remote directory must be empty before you can remove it. If you do not supply the name of the directory to be removed, you are prompted for it. Not all remote computers support this command.

Example

Ted wants to delete the tempo directory on the remote computer. He types:

```
rmdir tempo
```

SEND

Purpose

The SEND command copies a local file to the remote computer in the current file transfer type (ASCII or binary).

Syntax

```
send [local_filename] [remote_filename]
```

Option	Description
local_filename	The name of the file you want to transfer to the remote computer. FTP prompts for the local filename if it is not specified.
remote_filename	The name of the file you want for the file once it has been transferred to the remote computer. If you do not specify this filename, then FTP uses the same name as the *local_filename*, removing any path or drive specifiers.

Example

Julia wants to transfer the local file `more.dat` to the remote computer. To do this, she types:

```
send c:\more.dat add.dat
```

The file called `more.dat` is transferred to the remote computer in a file called `add.dat`.

See Also

The PUT command. SEND works the same as the PUT command.

SENDPORT

Purpose

The SENDPORT command allows you to determine whether or not PORT commands are sent for each data connection. By default, FTP attempts to use a *port* command when establishing a connection for each data transfer.

Note The SENDPORT command is not for general use. It is intended for solving problems with specific FTP implementations.

Syntax

```
sendport [on | off]
```

Option	Description
on	Specifies that SENDPORT should be turned on. This is the default.
off	Specifies that SENDPORT should be turned off, and the remote computer uses a default network port when setting up a connection for data transfer.

Comments

If you do not specify on or off, the command turns SENDPORT off if it is on, and turns SENDPORT on if it is off.

When the use of port commands is disabled, FTP makes no attempt to use port commands for each data transfer. This is useful when trying to communicate with certain FTP implementations on remote computers.

When you set SENDPORT on, it tells the remote computer to use a non-default network port when setting up a network connection for data transfer.

SHOW

Purpose

The SHOW command displays a file from the remote computer on the local PC screen. This command is only intended to be used to display the contents of ASCII files.

Syntax

```
show [remote_file]
```

Option	Description
remote_file	The name of the file you want displayed on the screen. If *remote_file* is not specified, FTP prompts for it.

Example

Bob wants to display the remote file called `info.dat` on his local PC screen. To do this, he types:

```
show  info.dat
```

The file contents are displayed on the screen.

SLASHFLIP

Purpose

The SLASHFLIP command determines whether or not backslashes (\) are converted to slashes (/) on outgoing commands. This feature is convenient if you are used to using DOS and specifying pathnames with backslashes.

Syntax

```
slashflip [on | off]
```

Option	Description
on	Specifies that you want backslashes converted to slashes. This is the default configuration.
off	Specifies that you do not want backslashes converted to slashes.

Comments

If you do not specify on or off, the command turns SLASHFLIP off if it is on and turns SLASHFLIP on if it is off.

Example

Carole wants FTP to convert slashes to backslashes. To implement this feature, Carole types:

```
slashflip on
```

The screen displays:

```
Slash translation on.
```

STATUS

Purpose

The STATUS command displays the connection status and settings of local options (for example, VERBOSE and SLASHFLIP) and gives a brief status report. The result of this command is partially dependent on the remote computer.

Syntax

```
status
```

Comments

UNIX FTP servers do not usually implement this command. Therefore, the remote section of the status is not shown when you are connected to a UNIX remote computer. The information about your local PC is still displayed.

Example

Ted wants to know the connection status and settings of the various FTP options. To get a display of these, he types:

```
status
```

The following is displayed on the local PC screen:

```
Connected to abcco
Transfer type is ascii.
Bell off.
Debugging off.
Filename globbing on.
Hash-mark printing off.
Interactive prompting off.
Sending of PORT commands on.
Slash translation on.
Verbose mode on.

Remote Status:
      STAT command not implemented.
```

TAKE

Purpose

The TAKE command causes FTP to read FTP commands from a file instead of from the keyboard. Questions and password responses still come from the keyboard. This command is useful for those cases where a standard set of files are transferred between computers on a regular basis.

Syntax

```
take [filename]
```

Option	Description
filename	The name of the file from which you want FTP to read commands. If you do not specify a *filename*, FTP prompts for it.

Comments

Although not required, you may want to use the extension .tak for TAKE files.

Example

Bob has a filename called trans.ftp that he uses on a regular basis to transfer files to and from his local PC and the remote computer. To use this file, he types:

```
take trans.ftp
```

See Also

The section in this chapter called "Using the FTP TAKE Command."

TYPE

Purpose

The TYPE command specifies the file transfer type, either ASCII or binary.

Syntax

```
type [type]
```

Option	Description
type	Specifies the transfer type (ASCII or binary).

Comments

If you do not specify a transfer type, the current type is displayed.

The command TYPE ASCII is equivalent to the ASCII command. The command TYPE BINARY is equivalent to the BINARY command.

Example

Alice wants to change the file type to binary. To do this, she types:

```
type binary
```

The screen displays:

```
Type set to I.
```

See Also

See the ASCII command for information about the ASCII transfer type.

See the BINARY command for information about the binary transfer type.

USER

Purpose

The USER command sets your username to a different one than you used to log on the remote computer. This command is useful if the login failed or if the remote computer supports changing the username during a session.

Syntax

```
user [username]
```

Option	Description
username	Specifies the *username* you want to use on the remote computer. You are prompted for *username* if you do not provide it.

Comments

You are prompted for a password or account if they are required to complete a login.

Example

Carole could not successfully log on using her usual *username*. She decides to use the *username* of guest instead. To do this, she types:

```
user  guest
```

VERBOSE

Purpose

The VERBOSE command allows you to specify whether or not informative messages about the status of FTP commands should be displayed (for example successful completion of transfers, server messages, and transfer times).

Syntax

```
verbose [on | off]
```

Option	Description
on	Sets VERBOSE mode on. This is the default configuration.
off	Sets VERBOSE mode off.

Comments

If you do not specify on or off, the command turns VERBOSE on if it is off and turns VERBOSE off if it is on.

Example

Bob wants to set the VERBOSE command off because he does not want to see all the status messages. To do this, he types:

```
verbose off
```

The screen displays:

```
Verbose mode off.
```

!

Purpose

The ! command causes FTP to temporarily return to DOS so you can issue system commands. The FTP connection to the remote computer is maintained.

Syntax

```
! [command]
```

Option	Description
command	Specifies a DOS command you want performed in the local PC environment.

Comments

Once the command is executed, you are returned to FTP.

If you do not specify *command*, the local PC prompt is displayed and you remain in the local PC environment until you issue the EXIT command.

Example

Julia wants to use a DOS command, but she does not want to exit FTP. To do this, she can temporarily return to DOS and issue the command. To do this, she types:

```
!
```

The screen displays:

```
Warning, some programs will interfere with network communication
and can cause lost connections. Do not run any network programs from this DOS shell.
Type 'EXIT' to return to FTP
```

The DOS prompt is displayed on the screen, and Julia can issue the command(s) she wants. Once she is finished, to return to FTP, at the DOS prompt she types:

```
exit
```

?

Purpose

The ? command displays a list of available commands or information about one or several commands.

Syntax

? [*command*]

Option	Description
command	The command or commands about which you want information.

Comments

If you do not specify a command or commands, a list of available FTP commands is displayed on the local PC screen.

Example

Carole wants to display information about the REMOTEHELP command. To do this, she types:

```
? remotehelp
```

The following is displayed:

```
remotehelp get help from remote server
```

See Also

The HELP command. ? works the same as the HELP command.

4

Using Remote Command Services

Chapter Overview

This chapter describes how to access remote computers using RCP and RSH.

- RCP (remote copy protocol) copies files between your PC and an HP-UX or UNIX remote computer.
- RSH (remote shell) allows you to connect to an HP-UX or UNIX remote computer and execute commands. This is useful if you need to execute one command or a series of commands in a command script on a remote computer.

(Note: Using the DOS 5.0 Task Manager shell lets you start multiple RCP or RSH sessions and "hot key" between them.)

Note In order to use RCP or RSH you must have the ARPA Services software loaded, and the remote computer must support RSH and RCP.

Before You Begin

RCP and RSH are based on the BSD 4.2 UNIX services of the same name. However, certain compatibility issues arise when using these commands on a non-UNIX host (your PC workstation).

Note Remote computers must be able to take the local PC's IP address and resolve it to the PC computername. This is traditionally done using a host table file, but the remote computer can use other methods, such as the domain name resolver.

The RCP and RSH protocols require that the local user be identified by a username. On a UNIX system, the username is the login id used to gain access to the UNIX system. Although the DOS-based PC has no such user identifier, RCP and RSH still require that the username be supplied.

When configuring the ARPA Services, you are asked for a *User Name* on the ARPA menu. This represents the ARPA local username associated with your PC computername when using the RCP and RSH commands. The local username is used by remote computers to verify the user's access rights through the use of an `.rhosts` file. For more information about `.rhosts` files, refer to the "rhosts File" section of this chapter.

Both RCP and RSH use *local username* as the default remote username when connecting to remote computers. This remote username can be supplied in the `RCP` or `RSH` command (the local username cannot be), but it is recommended for ease of use that the local username be the same as the most often used remote username.

For example, suppose that you most often connect to a UNIX system with the login of `marie`. You should set your local username to `marie` to match this remote username. If you occasionally want to connect to a UNIX system with a different login (for example, `xyz`), you may explicitly specify the remote username as `xyz` using the `user` option of the RCP command, or the `-l` option of the RSH command.

rhosts File

When you use the RCP or RSH commands, you do not need to provide login and password information to the remote computer to which you are connecting. This is because you must have a `.rhosts` file in your **home directory** on the remote computer. This `.rhosts` file lists the computernames and associated local usernames (for UNIX systems, the local username is your login ID) that have access to that remote computer via RCP or RSH. This means that if you want to use RCP or RSH, your PC's computername and username must be in the remote computer's `.rhosts` file.

Note For further information about a remote computer's specific implementation, refer to that computers' reference documentation.

rhosts File Format

The `.rhosts` file is a text file in which each line is an entry. An entry consists of the PC computername, the PC local username, and any comments about the entry. The computername and PC local username are separated by tabs and/or spaces, and comments begin with a # character.

Example rhosts Entry

An example entry in the `.rhosts` file of the remote computer, *hostB*, could be:

```
PC5   marie
```

Once this entry is in the `.rhosts` file in the home directory of Marie's account on remote computer *hostB*, she can use RCP and RSH from her PC to *hostB*.

Security Problems

If you list PCs in your `.rhosts` file, you are leaving your PC workstation open to potential security problems. PCs should never be added to `/.rhosts` (the root's `.rhosts` file) or `/etc/hosts.equiv` (the list of all hosts "equivalent" to this UNIX computer) on any UNIX system. This is because when you use the `.rhosts` file in your home directory, the `.rhosts` file lists sites you are allowed to login from without a password. It would be easy for another PC to login as a different network computer, and the protection in the RCP and RSH protocols depends on computers not being able to do this.

Using RCP

RCP is an implementation of the 4.2 BSD UNIX Remote Copy Protocol. It allows you to copy files between your PC and a remote computer.

You cannot use RCP to perform local transfers. That is, you cannot copy a file from your local PC to your local PC.

When a file is transferred from a remote computer to the PC, the file's attributes will be the same as that file's owners permissions on the remote computer.

Using the DOS 5.0 Task Manager

Using the DOS 5.0 Task Manager enables you to start one or more `rcp` sessions, and then "hot key" back and forth between these sessions or other tasks using Ctrl - Esc. If you leave an `rcp` session, it remains suspended (no networking activity takes place) until you return to it.

Legal Filenames

Different operating systems can have different filename syntax requirements. This means that when you use RCP to transfer a file from a remote computer to your PC you may encounter some problems with legal filenames. For example, the filenames `April_13_1988` and `data.all.1` are legal filenames on UNIX systems but are illegal DOS filenames.

If you are transferring files with filenames that are not legal DOS filenames, it is recommended that you provide the target filename rather than use the default filename.

For example, you would enter the first example RCP command and *not* the second example:

```
rcp hpcomp:not.legal.filename newname.fil
```

DO NOT ENTER:

```
rcp hpcomp:not.legal.filename.
```

When possible, choose filenames that are legal for both DOS and the remote computer(s) with which you are communicating. If you do transfer a file from a remote computer to the PC and the requested target filename is not a legal DOS filename, RCP converts the filename to a legal DOS filename by truncating extra characters. Refer to the DOS manual for your PC for more information about legal DOS filenames.

RCP Command

The `RCP` command is used in the following format:

`rcp` [*options*] [[*user@*]*host:*]*source* [[[*user@*]*host:*]*source* ...] [[*user@*]*host:*]*target*

Option	Description
`-a`	Specifies that the file to be transferred is ASCII and requires translation between UNIX Newline format and the DOS Carriage-Return Linefeed format. This is the default transfer mode. You can specify either the `-a` or the `-b` option but not both.
`-b`	Specifies that the file is binary and disables the UNIX to DOS translation function. You can specify either the `-a` or the `-b` option but not both.
`-r`	Specifies recursive copy. This option allows you to copy files in the specified directory and any files and directories below it in the directory hierarchy.
`-p`	Specifies that the files to be transferred will have their file timestamps preserved. The default is *not* to preserve timestamps, and any files transferred will have their timestamps set to the current date and time.
user	An optional field representing the remote name of the user whose permissions and `.rhosts` file will be used. If *user* is specified, a `.rhosts` file on the user's remote home directory must contain an entry for the PC and the local username. If the *user* option is not specified, the local username specified in the ARPA Services configuration process is assumed to be the remote user name.
host	The computername of the remote computer handling the transfer.
source	The name of the file (including the pathname) or directory to be transferred. If the source is a directory, the `-r` option must be specified.

target The name of the target file (including the pathname) or directory. If the pathname you specify does not begin with a / character, it is interpreted by the remote computer to be relative to your home directory.

Example

Bob wants to copy the `/etc/hosts` file to his local PC from the remote computer called `ucbvax`. To do this, Bob types the following at the DOS prompt:

```
rcp ucbvax:/etc/hosts hosts.txt
```

The file is then copied to a local file called `hosts.txt` on the PC.

Example

Julia wants to copy the binary file `data.exe` from her local PC to her login directory on the remote computer `Wally`. To do this, at the DOS prompt, Julia types:

```
rcp -b data.exe Wally:data.exe
```

The file is copied into the file called `data.exe` on the remote computer `Wally`.

Example

Bob wants to copy the `letters` directory to his local PC from the remote computer called `remhost`. To do this, Bob types the following at the DOS prompt:

```
rcp -r remhost:letters temp
```

The `letters` directory is created in the local directory, `temp`.

Example

Ted wants to copy several files, all those with the extensions `.c` and `.h`, to his backup directory on the remote computer, `finance`. To do this, he types:

```
rcp *.c *.h   finance:mybackup
```

Example

Alice wants to copy two local directories, `c_dir` and `h_dir`, to the `mybackup` directory on the remote computer, `finance`. To do this, she types:

```
rcp -r c_dir h_dir finance:mybackup
```

Using RSH

RSH is an implementation of the BSD UNIX command that allows you to execute commands on a remote UNIX computer connected to the network and see the results on your PC. For example, you can use RSH to execute the `cat` command on a remote computer and view the output on the PC.

Using the DOS 5.0 Task Manager

Using the DOS 5.0 Task Manager enables you to start one or more `rsh` sessions, and then "hot key" back and forth between these sessions or other tasks using Ctrl - Esc. If you leave an `rsh` session, it remains suspended (no networking activity takes place) until you return to it.

RSH Command

The `RSH` command is used in the following format:

```
rsh host [-l user] commandline
```

Option	Description
host	The computername of a remote computer.
`-l` *user*	If the `-l` option is specified, *user* is passed as the remote user name to the remote computer. A `.rhosts` file in the user's remote home directory must contain an entry for the PC and the local username listed in the configuration file set by the ARPA Services configuration process. If the `-l` option is *not* specified, the local username specified in the ARPA Services configuration process is assumed to be the desired login.
commandline	The command passed to the remote computer to be executed; results are displayed on the PC. Shell metacharacters that are unquoted are interpreted on the PC, and quoted metacharacters are interpreted on the remote computer.

Note Commands passed to the remote computer cannot require interactive input (such as a text editor). If you issue RSH with such a command, it will result in error or unpredictable behavior.

Example

Bob wants to list the contents of the file /etc/hosts on the remote computer ucbvax. To do this, Bob types the following at the DOS prompt:

```
rsh ucbvax cat /etc/hosts
```

The contents of the file are listed.

Example

Julia wants to append the remote file remfile in her home directory on the remote computer, otherhost, to the local file, locfile. To do this, Julia uses unquoted shell metacharacters because they are interpreted on the local computer.

At the DOS prompt Julia types:

```
rsh otherhost  cat remfile >> locfile
```

Example

Now Julia wants to append remfile to another remote file called newfile. To do this, Julia needs to use quoted metacharacters because they are interpreted on the remote computer.

At the DOS prompt she types:

```
rsh otherhost   cat remfile ">>" newfile
```

Exit Code Information

RCP and RSH return an exit code when either is exited. An exit code of 0 is returned if no problems are encountered and a 1 is returned if an error occurs. All error messages are sent to STDERR. All other output is sent to STDTOUT.

This is useful when using batch files. When you are using one of the services interactively, errors are displayed as they occur. Refer to the DOS user's guide for your PC for more information about exit codes.

Troubleshooting with PING

Chapter Overview

This chapter contains information about how to use the ARPA Services **PING** command. The `PING` command provides an easy way to verify the physical connection to a remote computer, using the internet control message protocol (ICMP) echo facility.

This chapter describes the following:

- A description of the **PING** command.
- Information about how to isolate a network problem.
- How to use the **PING** command.

A Description of PING

The **PING** command sends an echo request to the remote computer. The remote computer must be capable of receiving and responding to ICMP packets. ICMP is part of the IP protocol and typically is used to report errors in the processing of datagrams. It reports the number of successes or failures if the `-t` or `-n` option is selected. A PC can both send and reply to echo requests.

Isolating a Network Problem

When a remote computer fails to respond to a network request, it means there has been a failure at one of several points from the PC to the remote computer.

- The remote computer you are asking for may be down.
- A network or gateway in the path from the PC to the remote computer may be down.
- The remote computer may not implement the service you are requesting.
- A gateway may be routing incorrectly.

`PING` can usually determine which of these problems has caused the trouble.

An unsuccessful `PING` directed to another computer on the same network usually means that the original remote computer is down or not listening to the network. Failure to get echoes from any remote computer on the network means that the trouble is along the path somewhere. A `PING` directed to the gateway into the network in question can help narrow in on the problem. You can continue to work back from the target toward the originator until you find the point where communication breaks down.

The echo request sent by `PING` is dispatched using a low-level protocol that does not guarantee delivery. As a result, any one echo request can be accidentally lost, for example a temporary overload in some gateway. To check whether a particular network or gateway has failed, use a series of `PING` commands using the `-t` and `-n` options. If they consistently succeed in getting to the point in question and consistently fail to get beyond that point then you can assume there is a problem with that network or gateway.

For more information about troubleshooting, refer to the *Errors and Troubleshooting Guide: HP ARPA Services 2.1/MS-DOS*.

Using PING

The PING command uses the following format:

```
ping remote_computer  [-t [timeout_value]] [-n [num_times]]
```

Option	Description
remote_computer	The name or IP address of a remote computer.
-t [*timeout_value*]	The length of time (*timeout_value*), in seconds, that this node waits for an ICMP echo reply from a remote computer. The default *timeout_value* is 20 seconds; the maximum value is 300 seconds.
-n [*num_times*]	This option causes PING to "ping" the remote computer as many times as specified in the brackets, and then stop. The default *num_times* is 1 time.

Note There is a one-second delay between the sending of each ICMP request.

PING reports success with the following message:

```
[n] echo received from remote_computer with roundtrip xx msec
```

Variable	Description
n	The number of the echo being displayed (depending on what you specified for as *num_times*). It starts at one and counts up.
remote_computer	The name or IP address of the remote computer you specified in the PING command.
xx	The roundtrip time for the echo, in milliseconds.

Failure Messages

`PING` may report failure messages. If you receive a failure message, refer to the *Errors and Troubleshooting Guide: HP ARPA Services 2.1/MS-DOS* for possible solutions.

Making Services Transient

This appendix discusses how to make an ARPA Service *transient* so that networking modules are loaded into memory only when absolutely required.

The default installation (when you select no for demand loading in the NetSetup utility) causes various network drivers (TSR programs) to be loaded automatically in NET_STRT.BAT. Loading these modules allows you to execute FTP, RSH, and RCP services, as well as the *Kermit* terminal emulator and other third party sockets applications and terminal emulators. When these networking TSR programs are loaded, they consume DOS memory regardless of whether they are needed at the time. By choosing to *demand load* these services during installation or reconfiguration, you can keep as much DOS memory free as possible. Special batch files are available that automatically load the networking TSR modules required, execute the desired service, then automatically unload them to regain DOS memory again.

Note If you wish to use any networking services while MS Windows is running, then special rules for demand loading apply. See Appendix B, Windows 3.0, for more information.

Installing and Configuring Demand Loading

The NetSetup utility allows you to select whether the networking modules are loaded upon demand. To change the loading behavior, simply execute the NetSetup utility and select either `yes` or `no` to the demand loading question.

Note You may not load network files from within the DOS 5.0 Task Manager shell. (So, if you selected the `Demand Load` option during ARPA installation with Netsetup, reconfigure that option to `no` before trying to load any networking files from within the shell.) You **must** load all networking software before entering the DOS 5.0 Task Manager for the first time.

Using Transient Services

To use a service after selecting demand loading, you need to invoke a special batch file to invoke the service. The following table shows the original default service name and the corresponding batch file that lets you demand load the networking TSR programs.

Service	Default	Demand Load
FTP	FTP [*arguments*]	FTPD [*arguments*]
TELNET w/Kermit	TELNET [*arguments*]	TELNETD [*arguments*]
TELNET w/Advlink	ADVLINK [*arguments*]	ADVLINKT [*arguments*]
RCP	RCP [*arguments*]	RCPD [*arguments*]
RSH	RSH [*arguments*]	RSHD [*arguments*]

Use the batch file name just as you use the service itself. The trailing `D` or `T` in the batch file name indicates that the service is demand loaded, and the batch file accepts the same command-line arguments as invoking the service directly.

B

Using ARPA Services with Windows 3.x

Introduction

Windows 3.*x* introduces new opportunities for user productivity, but some networking services are affected by these new capabilities. This section describes the use and limitations of ARPA Services 2.1 in the various Windows 3.*x* modes (real, standard, and enhanced). For more information on Windows 3.*x* operating modes, consult the Windows 3.*x* User's Guide.

This appendix includes the following sections:

- Using FTP, RSH, RCP, or other sockets applications.
- Using terminal emulators.
- Executing PING and NETDIAG.
- Using demand loading.
- NewWave Office application support.

Using FTP, RSH, RCP, or Other Sockets Applications

This section describes the requirements imposed by each of the three Windows modes on your networking applications, along with the procedure you will use to run your applications.

Using Enhanced Mode

These applications, as well as sockets applications supplied by other vendors, may be used in Windows enhanced mode. They can run in either a full screen or windowed screen provided that the WINFIX utility has been executed. WINFIX is installed in NET_STRT.BAT during installation, and automatically adds appropriate entries to the Windows SYSTEM.INI file for correct networking operation. WINFIX is described in detail in the *Installation Guide – HP ARPA Services 2.1/MS-DOS.*

In enhanced mode, you may run multiple applications in different DOS windows simultaneously (provided that you have configured enough sockets connections for all simultaneous instances). For example, you can execute FTP in three different DOS boxes simultaneously.

Using Standard and Real Modes

Since these applications are not Windows applications, you should execute them using Windows icons as described in the next section or by using the `File Run` menu bar in the Windows interface. Furthermore, only one instance of the application may be executing at any time.

While executing networking applications in these modes, the Alt-Tab and Ctrl-Esc key combinations (or any other key sequence that would cause another application to run) are disabled. We recommend that you use the PIF files provided with ARPA Services 2.1 because they control this behavior. If network applications are invoked from the Windows 3.*x* DOS command prompt window, the PIF file cannot restrict their behavior. Switching to another application is very likely to cause data corruption.

Note Real mode is supported in Windows 3.0 only. Later versions of Windows do not support real mode.

Running FTP, RCP, or RSH Network Applications

Use the following procedure to run your network application. This procedure will automatically invoke the correct PIF (Program Information File) for your network application, regardless of which Windows 3*x* mode you are using. A PIF file saves you the trouble of remembering the requirements ARPA Service must satisfy in order to run under standard, real, or enhanced modes in Windows 3*x*.

1. Go to the Program Manager "desktop."
2. In this window, select the HP ARPA Services 2.1 icon. The ARPA Services window will appear.
3. In the ARPA Services window, select the icon for the ARPA Service that you want. The PIF file associated with that service will be invoked automatically.

Running Other Sockets Applications

If you are using other sockets applications, you may need to construct your own PIF file. First, check with the vendor of your sockets application to see if a PIF file has been provided. If not, here are some guidelines to assist you in constructing your own PIF file:

- Look at the existing PIF file for the FTP, RCP, or RSH applications for suggestions on what parameters to include in your own PIF file.
- Use the Microsoft Windows 3*x* PIF editor in real or standard mode. (This is described in the Microsoft Windows 3*x* User's Guide. Note that real mode is supported in Windows 3.0 only.)
- If you are using standard mode, you should select the `No Screen Exchange`, `Prevent Program Switch`, and `Reserve Shortcut Keys` check boxes. Setting the switches in this way will guarantee that real and standard mode operation will be limited appropriately, while still allowing full use in enhanced mode.

Using Terminal Emulators

The new modes introduced in Windows 3.*x* have considerable impact on terminal emulator products, due to the low-level interfaces they use to interface with networks. Regardless of whether you are using a DOS-based or Windows-based terminal emulator program, the following rules apply:

- Only a single terminal emulator that does not do its own session management can execute at any given time. These emulators will use the Minimal User Interface (MUI) feature of TELNET. See chapter 2 for more information about MUI. Examples include:
  - AdvanceLink for DOS, from Hewlett-Packard.
  - Kermit, from Columbia University.
  - Reflection, from Walker, Richer, & Quinn.
- If your terminal emulator performs its own session management, then multiple terminal emulators can be active. Examples include:
  - AdvanceLink for NewWave/Windows, from Hewlett-Packard.
  - Business Sessions, from Tymlabs.
  - Dynacomm, from FutureSoft.

To determine whether or not your terminal emulator provides its own session management, consult your manual or contact the vendor.

Additional limitations, which apply to the specific Window 3.0 modes, are included in the following *Enhanced Mode* and *Standard and Real Mode* sections.

Using Enhanced Mode

Provided that your terminal emulators meet the restrictions described earlier (single emulator using MUI, multiple emulators performing their own session management), you may freely use the multi-tasking capabilities of enhanced mode and have multiple terminal emulators active at any time. Some valid combinations include:

- A single copy of AdvanceLink for DOS using its own session management to connect to a single remote host.
- Kermit executing in a single DOS prompt window, using MUI to perform session management.
- Multiple copies of AdvanceLink for Windows/NewWave (which does its own session management) while concurrently executing a single terminal emulator using MUI.

Note The Kermit terminal emulator program, provided as a convenience and described earlier in this manual, requires the use of MUI for connection management. You CANNOT have multiple Kermit terminal emulators in different DOS prompt windows. This would violate the rules described earlier (only a single terminal emulator using the MUI at any given time).

In order to guarantee correct operation in Windows 3.*x* enhanced mode, your Windows 3.*x* SYSTEM.INI configuration files must be suitably modified. The networking installation program installs the WINFIX program in your NET_STRT.BAT file during installation and automatically adds appropriate entries to the Windows SYSTEM.INI file for correct networking operation. WINFIX is described in detail in the *Installation Guide – HP ARPA Services 2.1/MS-DOS*.

Using Standard and Real Modes

Provided that your terminal emulators meet the restrictions described earlier (single emulator using MUI, multiple emulators performing their own session management), you can either have a single DOS terminal emulator active in a DOS prompt window, or multiple Windows-based terminal emulators active in Windows.

While executing a DOS terminal emulator program in a DOS prompt window, you cannot press Alt-Tab, Ctrl-Esc, or any other key sequence that would cause another application to run. Windows 3*x* restricts DOS command prompt windows in either standard or real mode to be full-screen applications, and only one instance of the application can be executing at any time.

You can have multiple Windows-based terminal emulator applications active, if they perform their own session management. However, one additional limitation exists on Windows-based terminal emulators while executing in standard mode.

If you are using Windows 3*x* in standard mode with a protected mode terminal emulator (using a BAPI interface), you must load the BAPI dynamic link library, BAPIDLL.EXE. An example of this type of terminal emulator is AdvanceLink for Windows/NewWave. The BAPIDLL.EXE file is supplied with the HP ARPA 2.1 software, but is not configured to load. To load this DLL, add the following line to your WIN.INI file in the WINDOWS directory:

```
run=<network path>\bapidll.exe
```

If you are using AdvanceLink for NewWave, check the installation procedure for that product.

Note Real mode is supported in Windows 3.0 only. Later versions of Windows do not support real mode.

Running Terminal Emulators

Only one terminal emulator, Kermit, is provided with ARPA Services 2.1, and a PIF file (`telnet.pif`) is provided for your convenience. To run Kermit:

1. Go to the Program Manager "desktop."
2. In this window, select the HP ARPA Services 2.1 icon. The ARPA Services window will appear.
3. In the ARPA Services window, select the Telnet icon. The PIF file associated with Kermit will be invoked automatically.

If you are not planning to use the Kermit emulator, you may need to construct a PIF file for your own application. First, check with the vendor of your emulator to see if a PIF file has been provided. If not, here are some guidelines to use in constructing your own PIF file:

- Look at the PIF file for the Telnet application for suggestions on what parameters to include in your own PIF file.
- Use the Microsoft Windows 3*.x* PIF editor in real or standard mode. (This is described in the Microsoft Windows 3*.x* User's Guide).
- If you are using standard mode, you should set the `No Screen Exchange`, `Prevent Program Switch`, and `Reserve Shortcut Keys` check boxes. Setting the switches in this way will guarantee that real and standard mode operation will be limited appropriately, while still allowing full use in enhanced mode.

Ping and Netdiag

You may execute these utilities in any Windows mode, but you must invoke them via the PIF files provided in this product. These PIF files ensure that PING and NETDIAG will only be executed in full screen exclusive mode, which is required to prevent data corruption for these particular applications.

Note

If you run PING while in the DOS 5.0 Task Manager shell, you will get the error message, NET0131, telling you that NMTSR must be loaded before you start the Task Manager.

You should:

1. Exit the DOS 5.0 Task Manager.
2. Edit the NET_STRT.BAT file to make sure that NMTSR will be loaded.
3. Try running PING again within the DOS 5.0 shell.

Using Demand Loading

The ability to load and unload networking modules from DOS memory arbitrarily is referred to as demand loading. Demand loading is typically used directly from DOS, but may be used, with some limitations, in the Windows 3.*x* environment.

Using Enhanced Mode

Demand loading does not work in Windows 3.*x* enhanced mode. Windows 3.*x* enhanced mode uses 80386 hardware support to implement multiple virtual machines, and demand loading is not supported in this environment. If you try to use demand loading, you will get the following error message:

```
NET0232 Cannot demand load modules after Windows 3.0  is running.
```

Using Standard and Real Modes

Demand loading in either Windows 3.*x* real or standard mode works, although Windows does generate an erroneous error that should be disregarded. You can successfully load networking modules, execute your application, and unload the networking modules. When you exit the DOS command prompt window, Windows displays the following message:

```
Your pop-up program has been loaded and you may activate it as you would normally. When
you have finished using this pop-up program, exit it and press Ctrl-C to return to
Windows.
```

You can disregard this message. Simply press Ctrl-C to return to Windows.

Note Real mode is supported in Windows 3.0 only. Later versions of Windows do not support real mode.

NewWave Office Application Support

ARPA Services 2.1 incorporates support for standard ARPA networking services as well as NewWave Office NetIPC applications. Previously, this support was only available by purchasing two separate products (ARPA Services 2.0 and NS 2.0).

You should follow the recommendations provided with the particular NewWave Office application regarding support of the various Windows 3*x* modes. As part of installing the networking needed by NewWave Office applications, the WINFIX utility correctly sets Windows configuration values for NewWave Office NetIPC application support. For more information, see *Installation Guide – HP ARPA Services 2.1/MS-DOS*.

Glossary

AdvanceLink

An HP software package that allows PC workstations to emulate a terminal on the following computers: HP 1000, HP 3000, HP 9000, and DEC VAX computers. AdvanceLink also allows file transfer with an HP 3000 computer.

ARPA domain

A portion of the computername required for computers on the LAN that are using ARPA Services. The domain name can contain as many fields as will fit within 240 characters. See also **ARPA Services**.

ARPA Services

A group of software programs that allows terminal access and file transfer between a PC and an HP 9000, an HP 1000, or a DEC VAX computer. ARPA Services include `FTP`, `TELNET`, `RSH`, `RCP`, and `PING`.

Batch file

An executable file which contains a group of commands that are performed whenever the file is run. See your DOS or OS/2 manual for details.

Command

A word or phrase that you type at the system prompt to carry out an action when you press the ENTER key.

Computername

The name of a PC workstation. Computernames can be up to 15 characters in length and cannot be duplicated on the LAN.

Configuration

The way your PC workstation, server, or LAN is set up. This includes both hardware and software. LAN Manager software configuration is stored in the `LANMAN.INI` and `PROTOCOL.INI` files.

Domain Name Resolution (DNR)

A service under ARPA that provides a way to identify IP addresses of host computers without using a host table.

FTP (File Transfer Program)

An ARPA Service that allows you to transfer files between a PC workstation and a minicomputer such as the HP 9000, HP 1000, or DEC VAX computer.

Host computer

A minicomputer on the network that provides networking services, such as ARPA Services and Network Services to PC workstations.

HP AdvanceLink

See **AdvanceLink**.

HP ARPA Services

See **ARPA Services**.

HP-UX

HP's version of the UNIX operating system that runs on HP 9000 computers.

InterProcess Communication (IPC)

The ability of local and remote applications to transfer data and messages among themselves; used to offer services to and receive services from other programs on the network. See also **Sockets**.

IP (Internet Protocol) address

The number that identifies the computer to other computers on the network. Any computer using the TCP/IP protocol on the network must have a unique IP address.

LAN

See **Local Area Network.**

Local

Describes any server, PC workstation, or network resource connected to a computer and physically located where the user is working.

Local Area Network (LAN)

One or more PC workstations and one or more servers or hosts connected together so they can share files, printers, and plotters.

NetSetup utility

An HP LAN Manager utility which allows a network administrator to install and initially configure an HP PC LAN.

Network administrator

An individual responsible for the LAN. This person typically sets up the network, assigns passwords and privileges, and helps users with problems they may have while using the LAN.

Network Services (NS)

Software programs that allow terminal access and file transfer between a PC workstation and HP 1000, HP 3000, HP 9000, or DEC VAX computers.

Node

A computer or network component, such as a hub or bridge, connected to the network.

NS domain

A two-field name that is required for a computer running Network Services (NS) on the network. The NS domain must be the same for all computers running NS on a given network. Each field within the domain name can be up to 16 characters. See also **Network Services**.

Path

The course that a message takes from the source computer to the destination computer.

Pathname

A unique sequence of names (the device letter, directory name(s), and file name) that locate a directory or file on a local or shared disc. An example of a pathname format is `B:\work\memo`. This pathname points to the file named `memo` in the `work` directory on the `B:` drive.

PC workstation

A personal computer running network software that allows the PC to connect to and use network resources.

PING

An HP ARPA Service that tests whether two computers running ARPA Services on a LAN can communicate with each other. See also **ARPA Services**.

Primary domain name

Name of the server or host on the LAN that is running the Domain Name Service (DNS) under ARPA. See also **Domain Name Resolution**.

RCP (Remote Copy Protocol)

An HP ARPA Service that copies files between PC workstations and HP-UX or UNIX computers on the LAN.

RSH (Remote Shell)

An HP ARPA Service that allows PC workstations to connect to HP-UX or other UNIX hosts or servers and to execute commands. RSH is useful if you need to execute one command or a series of commands in a command script on the host or server.

Service

Any software program that allows applications to run on the network. ARPA Services include FTP, TELNET, RSH, and RCP.

Sockets

An interface to interprocess communication between PC workstations and various remote computers. See also **InterProcess Communication**.

TELNET

An ARPA Service that allows terminal access between PCs and HP 9000, HP 1000, or DEC VAX computers. See also **ARPA Services**.

Terminal emulator

A program running on a computer which makes it function like a terminal. See also **AdvanceLink**.

Troubleshooting

Tracing and correcting configuration errors or hardware malfunctions in a system.

Wildcard character

A character that can be included in a filename to indicate any character or group of characters. With LAN Manager and DOS, you can use an asterisk (*) or a question mark (?) as wildcards. For example, the filename MARKET.* refers to all files named market with any filename extension. The filename ANNEX?.LST would be equated with filenames like ANNEX1.LST and ANNEX2.LST.

Workstation

See **PC workstation**.

Index

E

F

S

T

U

V

W