

**AT&T UNIX[®] PC
ENHANCED TCP/IP
WIN/3B LAN INTERFACE**

USER's GUIDE

July 1986

RESTRICTED RIGHTS

Use, duplication or disclosure by the Government is subject to restrictions as set forth in paragraph (b) (3) (B) of the Rights in Technical Data and Computer Software clause in DAR 7-104.9(a). The Wollongong Group, Inc., 1129 San Antonio Road, Palo Alto, California 94303.

Copyright © 1986 The Wollongong Group, Inc. All rights reserved. This document contains confidential trade secret information of The Wollongong Group, Inc. No part of this program or publication may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language or computer language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual or otherwise, except as provided in the license agreement governing the documentation or by prior written permission of The Wollongong Group, Inc., 1129 San Antonio Road, Palo Alto, California 94303. U.S.A.

PREFACE

INTENDED AUDIENCE

The User's Guide describes The Wollongong Group's implementation of TCP/IP for the AT&T UNIX¹ PC computer. This user's guide is intended to introduce end users to WIN²/3B LAN. It is assumed that the reader is familiar with the UNIX Operating System. You should also know the basics of working with a UNIX PC - how to open and close windows, select menu items, etc. See the AT&T UNIX PC Getting Started guide for information.

ASSOCIATED DOCUMENTATION AND APPLICATIONS

The UNIX PC Enhanced WIN/3B Programmer's Reference Manual contains a brief synopsis of each program connected with the use, administration, or installation of WIN/UNIX PC. Details on WIN/3B LAN system administration and installation can be found in the UNIX PC Enhanced WIN/3B Administrator's Guide.

Make sure the date and time zone are set properly on the UNIX PC's on your network. Otherwise, "ruptime", "finger", and "rwho" may not function properly.

If UNIX PC Voice Power is used on the same machine as WIN/3B LAN, it must be loaded first, or, if the STARLAN Network is used on the same machine as WIN/3B LAN, the STARLAN Network driver must be loaded first.

-
1. UNIX is a registered trademark of AT&T
 2. WIN is a trademark of The Wollongong Group, Inc.

CONVENTIONS USED IN THIS MANUAL

Information displayed on the screen is shown as below. For example:

Please login:

Information you type is shown in boldfaced type. For example:

Type meeting

Names of keys on the keyboard are enclosed in <>'s when used in procedures. For example:

< Exit>

The mouse buttons are represented as follows:

Left mouse button < B1>

Middle mouse button < B2>

Right mouse button < B3>

Screen keys are enclosed in []'s. For example:

[PASTE]

To use a screen key, press the corresponding function key, or point to the screen key with the mouse and press < B1> .

When asked to select a highlighted (reverse video) item from a menu, that item will be enclosed in |'s. For example:

Select | Filecabinet|

Key combinations are represented by the two keys separated by a plus mark (+). For example:

< Ctrl> +< D>

To type this combination, hold down <Ctrl>, press <D>, then release both keys.

The exit symbol in the lower left corner of a window is shown as:

[X]

The help symbol in the upper right corner of a window is shown as:

[?]

The symbol at the bottom of a form used to implement changes is shown as:

[OK]



**AT&T UNIX PC
ENHANCED TCP/IP
WIN/3B LAN INTERFACE**

USER's GUIDE

CONTENTS

1. INTRODUCTION TO COMPUTER NETWORKING	1
1.1 SHARING THE RESOURCES	2
1.2 THE ETHERNET CONNECTION	4
1.3 CALLING ON HOSTS	5
1.4 GATEWAYS TO THE WIDE WORLD	6
2. WIN/3B LAN NETWORKING COMMANDS	8
3. NETWORK VIRTUAL TERMINAL (telnet)	10
3.1 TELNET COMMAND MODE	12
4. TRANSFER PRIVATE OR PUBLIC FILES (ftp)	16
4.1 USING THE ftp COMMANDS	17
5. FTP AUTOLOGIN MANAGEMENT	24
5.1 THE ftp INITIALIZATION FILE (.netrc)	25
6. THE REMOTE COMMANDS	27
6.1 COPY A FILE BETWEEN UNIX HOSTS (rcp)	28
6.2 DISPLAY USERS ON THE NETWORK (rwho)	32
6.3 DISPLAY STATUS OF HOSTS ON THE NETWORK (ruptime)	34
6.4 DISPLAY INFORMATION ON REMOTE USERS (finger)	37
6.5 EXECUTE A COMMAND ON A REMOTE UNIX HOST (remsh)	41
6.6 LOGIN TO A REMOTE UNIX HOST (rlogin)	44
6.7 REMOTE COMMANDS' PERMISSION MANAGEMENT	51

7. TRANSFER PUBLIC FILES (tftp)	54
7.1 ftp COMMAND SUMMARY	55
8. SEND MAIL OVER ETHERNET (mailx)	57
9. HOST AND NETWORK INFORMATION	59
APPENDIX A. TELNET OPTIONS	60
APPENDIX B. FTP COMMAND SUMMARY	61
APPENDIX C. NETWORK ERRORS	66

WIN/3B LAN

USER's GUIDE

1. INTRODUCTION TO COMPUTER NETWORKING

A computer network can be defined as two or more computers that are connected by wires, satellite, or other means. The connection between the computers allows the individual computers to share and exchange information. The individual computers that comprise a computer network are called *hosts*.

A computer network gives you a choice of other computers on which to work. Rather than being confined to the computer to which your terminal is connected, you may log in to any computer on the network (providing you have a user account on that computer). Network software also provides utilities for moving or copying (transferring) files between hosts on the network. This allows greater information access between hosts, as information contained on one computer can be shared with other hosts on the network. Computer networks also provide the facility to send mail to users on other hosts, thereby providing easier and better communication between users.

With WIN/3B LAN Interface, Wollongong Integrated Networking gives you the capability to access computers across the hall, and across the globe. With WIN/3B LAN, you can easily share computing and information resources with colleagues across dissimilar computers, operating systems, and devices. WIN/3B LAN turns your computer into more than a standalone personal computer--it makes your UNIX PC a participating network host. Furthermore, WIN/3B LAN turns your UNIX PC into a virtual terminal to other hosts on the network. By using your UNIX PC as both a host and a terminal, you will gain more computing power, flexibility, and productivity at your UNIX PC.

This introduction will give you a general idea of how WIN/3B LAN operates in a network. A lot of networking ideas will be discussed here, although this introduction is not meant to be an indepth treatment of the subject. Nevertheless, if the only "hosts" you are acquainted with are those who throw parties, at the very least this introduction will serve to make you more familiar with networking terminology. This familiarity will assist you in understanding the terms that appear in the rest of this manual.

1.1 SHARING THE RESOURCES

Computers can be connected together in a variety of ways. When several computers that are located close together are connected on a network, they are said to comprise a Local Area Network (LAN). A LAN generally covers a relatively small geographical area, such as several computers in a company that are linked together via a network to share information and resources. One of the more popular local area networks is Ethernet which was developed by Xerox Corporation in the mid 1970's.

Networks are designed to share resources like:

- computers
- terminals
- files
- printers
- modems

Machines, like your UNIX PC, that can compute on their own and can share their files, computations, programming languages, and other services, are called network "hosts." "Terminals" are devices that transmit data to and from hosts, but are themselves incapable of computing. Some hosts are more powerful than others. Large timesharing computers have greater memory and data processing capabilities than a UNIX PC. By connecting your UNIX PC to larger computers in a network, you have access to those greater capabilities.

You can also share your files and programs with others who need them, and they can share their files with you. For instance, you can:

- employ a shared data base among co-workers in your company
- share a programming language with colleagues across the continent
- collaborate on a joint paper with associates from across the globe

At one time, this sort of intercommunication between dissimilar computers was impossible. The many different types of computers, incompatible operating systems, and dissimilar networks made the computing world a virtual Tower of Babel. People found their work was hindered because their computers simply could not understand each other.

Recognizing this problem, the Department of Defense evolved a set of networking protocols called TCP/IP for its Defense Data Network. TCP/IP stands for Transmission Control Protocol/Internet Protocol, and it has since been widely adopted as a networking standard.

The IP part of TCP/IP provides software routines to route your data throughout the network. Built on top of IP are two protocols for transporting the data: Transmission Control Protocol (TCP) and User Datagram Protocol (UDP). TCP has error checking built into it so that it is a highly reliable way to

send data. Its error-checking techniques examine data flow, acknowledge receipt of data, and retransmit bad or missing data.

UDP is a standard Department of Defense protocol that sends data without performing validation checks on the data. WIN/3B LAN makes use of the IP and Ethernet error control system to maintain the integrity of UDP transmission data.

"Internet Protocol" describes the standard well, because TCP/IP makes communication possible between such physically different networks as Arpanet, Ethernet, Pronet, and Satnet (the satellite network). As far as TCP/IP is concerned, it doesn't matter how the computers are physically linked—twisted wire, radio waves, coaxial cable, or whatever. Nor does TCP/IP care what kind of data is sent across the network—users are free to transmit ASCII text files (spreadsheet print files, word processor print files, and program source code) or binary files (data bases, spreadsheet data, and executable software programs).

The only requirement WIN/3B LAN makes of another host on the network is that the other host must support the Department of Defense standard TCP/IP networking protocols. The Wollongong Group distributes and supports TCP/IP software for many different computers through its line of Wollongong Integrated Networking Solutions (WINS) products. TCP/IP already exists on most supermini computers, on virtually all CAD/CAM (computer aided design/computer aided manufacturing) workstations, on all machines using the Berkeley 4.2 UNIX Operating System, and on most major European made machines.

1.2 THE ETHERNET CONNECTION

WIN/3B LAN uses the high speed Ethernet link to join your UNIX PC to other hosts in a local area network. A system that protects data transmission by sensing the proper time to

broadcast, by detecting transmission errors, and by rebroadcasting damaged data is built into Ethernet.

1.3 CALLING ON HOSTS

So that computers in a network can be reached by other computers, each computer, including your UNIX PC, must have its own unique Internet address. Depending on whether the address is specified in decimal (base 10) numbers or octal (base 8) numbers, each byte is separated by a period or a comma. For example, an address in decimal format might be 127.0.0.1, while its octal address would translate as 177,0,0,1.

Your System Administrator is likely to assign your UNIX PC its Internet address. But if it has been left to you to assign your UNIX PC an Internet address, be sure that you identify your network correctly (as well as your subnetwork and local area network, if your network includes them), and that you give your UNIX PC a host number that is not used by any other machine on your local area network.

Since remembering the long Internet addresses of all the hosts in your network would prove an unpleasant chore, WIN/3B LAN has simplified host addressing for you. A "name server" is built into WIN/3B LAN; by entering the Internet addresses of your frequently used hosts into a host file, and by giving these hosts easily remembered names, the WIN/3B LAN name server will serve up the Internet addresses for you by name. For example, you can log into a VAX³ by calling it VAX instead of 82.34.12.74. You can swap files with your officemate's UNIX PC by calling it tom's-pc instead of 82.43.12.69.

Whenever you transmit data to an Internet address, the data travels in "packets." A packet is a unit of bits that includes your UNIX PC's address, the destination address for the

3. VAX is a trademark of Digital Equipment Corporation.

packet, an identification number, and error control information. Packets are variable in length, but they are generally defined to be between 200 and 1000 bits long.

As your packet travels down the Ethernet, every host along the cable will check to see to whom the packet is addressed. Your UNIX PC, too, must be given an Internet address. If a host finds that the packet is not addressed to it, the host will pass the packet down the line to the next host.

Internet addresses are designed to operate on many types of networks. To translate Internet addresses into Ethernet addresses, a standard called Address Resolution Protocol (ARP) is used by hosts on the Ethernet. Since Ethernet does not understand Internet 32-bit addresses, it is up to the ARP to resolve matters. As long as the host you are addressing has ARP, it can recognize its own 32-bit Internet address and translate it to its 48-bit Ethernet address.

If you are unsure as to whether the hosts on your network have ARP, ask your System Administrator.

1.4 GATEWAYS TO THE WIDE WORLD

The physical medium of Ethernet cables precludes your local area network from extending beyond a building or close set of buildings. To extend the physical limits of the Ethernet, a machine acting as a "gateway" can connect your UNIX PC to a wide area network that extends far beyond your local network. Gateways take data from one network and retransmit it to another, which can be another nearby local area network like Ethernet, or a different, long-haul network, like Arpanet or Satnet.

Your network System Administrator can tell you if your network includes a gateway. If it does, and if you try to access a host that exists outside of your local area network, the gateway will route your transmission to an outside network. Gateways will continue to route your transmission across

networks until your host is finally reached. You do not even have to know how the other networks operate. All you have to do is customize your gateway's Internet address to the WIN/3B LAN Interface and let TCP/IP protocols handle the rest.

2. WIN/3B LAN NETWORKING COMMANDS

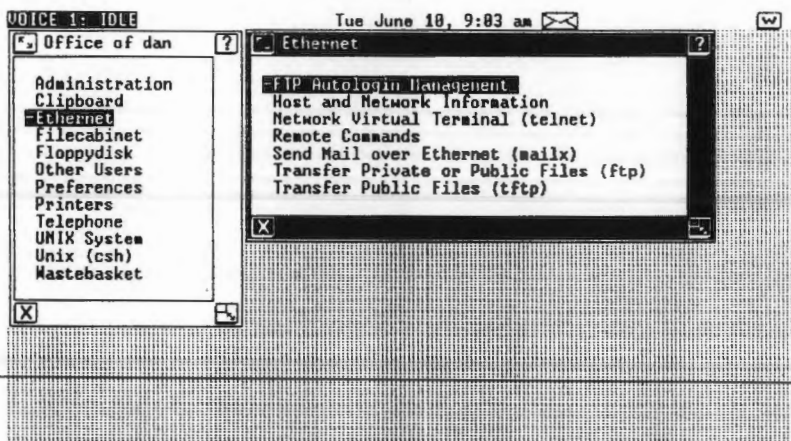
WIN/3B LAN networking commands are found under the Ethernet object in your Office window.

To access them:

1. Select | Ethernet|

and press <Enter>

You see the Ethernet windows:



Rather than explain the commands alphabetically, as they are listed in the Ethernet window, this guide will explain them in

the order that you will most likely be using them.

- **Network Virtual Terminal (telnet)**
- **Transfer Private or Public Files (ftp)**
- **FTP Autologin Management**
- **Transfer Public Files (tftp)**
- **Remote Commands**
- **Send Mail over Ethernet (mailx)**
- **Host and Network Information**

3. NETWORK VIRTUAL TERMINAL (telnet)

telnet uses the Department of Defense (DoD) Transmission Control Protocol (TCP) for reliable, virtual terminal communication. With *telnet* you can connect to any host that supports DoD standard TCP/IP on your network, regardless of the host's operating system. Using a remote host requires knowledge of its operating system, because *telnet* does not translate the commands of different operating systems. For example, the UNIX commands that operate on the UNIX PC will not operate on any hosts in your network that might be running the VMS⁴ or TOPS-20⁵ operating systems. However, if you use *telnet* to log into a host running a different operating system, you can type a command for that operating system from your keyboard, and the remote host will faithfully carry out your command.

telnet also offers a helpful, prompt-driven command mode from your UNIX PC to assist you in connecting to remote hosts. Because of this feature, and because *telnet* allows you to connect with a wide variety of computers and operating systems, *telnet* is probably the preferred method of remote login.

When you use *telnet*, you will always be prompted for your login name and password by the remote host, which then checks your login profile for authentication before granting you access. Your login profile on the remote host should include information concerning your terminal type, because *telnet* does not automatically pass this information along to the remote host. You will, however, be prompted for the terminal type when you log in to the UNIX PC. Consult your Network Administrator for assistance in setting up accounts and login profiles on remote hosts.

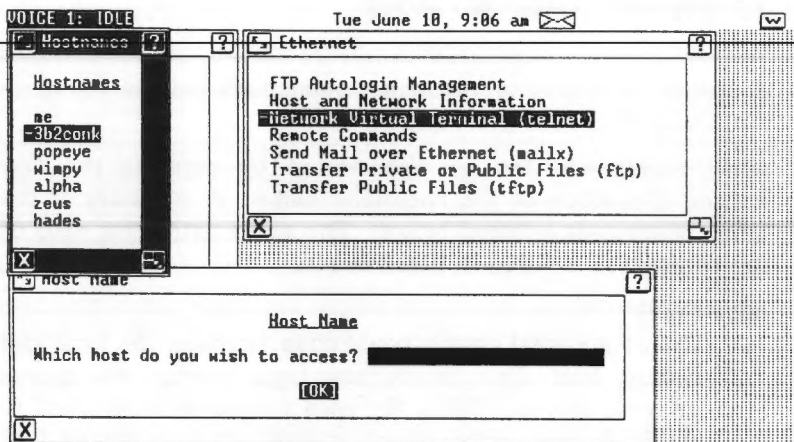
4. VMS is a trademark of Digital Equipment Corporation.

5. TOPS-20 is a trademark of Digital Equipment Corporation.

1. To access *telnet*:

Select | Network Virtual Terminal (telnet)| and press <Enter> .

You see the Host Name window:



2. Type *telnet* or press <Cmd> , highlight *telnet* and press <Enter> .

NOTE: You can also use <Slect/Mark> to step through the list of known hosts, stopping when *telnet* appears.

3. Press <Enter> to invoke *telnet*.

*teln*et will search its files, find the necessary Internet address, and make the connection. If that remote host is functioning, you will receive a login prompt when the connection is made.

When you clear the connection by logging off the remote host, *teln*et will close the Host Name window and return you to the Ethernet Command window.

NOTE: To invoke the *teln*et command mode, press <Enter> without choosing a host.

3.1 TELNET COMMAND MODE

A quick reference to the *teln*et commands can be found in Appendix A of this user's guide.

*teln*et commands can be abbreviated by entering the first unique characters of the command name. A summary of the *teln*et commands is listed below. The abbreviation for each of the commands is noted in parentheses.

open (ope)

Opens a virtual circuit connection between the local and remote host. The remote host login prompt will appear on your screen. Using the *open* command is identical to establishing a virtual circuit connection to a remote host by invoking *teln*et in input mode. This *teln*et command can only be used if a connection to a remote host has not been established.

close (cl)

Closes the current *teln*et connection. The *close* command severs the connection between the remote and local host. Once the connection has been closed, any processes associated with that connection will eventually die. This *teln*et command can only be used if a connection to a remote host has been established.

quit (q)

Severs any open connection, exits from the *teln*et

program, and returns you to the working environment of the local host. The *quit* command can be used with or without a connection to a remote host.

z (z)

Temporarily suspends the *telnet* program. The *z* command will return you to the working environment of the local host. If you are connected to a remote host, the connection will remain open though it is temporarily not in use. To reactivate the suspended *telnet* program, exit from the *shell* (<CTRL> +<D>). The *z* command can be used with or without a connection to a remote host.

escape (e)

Changes the default escape character. The default *telnet* escape character is a *tilde* (~). To change the default escape character, simply enter the *escape* command. You will be prompted to input the desired escape character. Once the new escape character has been entered *telnet* will reset the escape character for the remainder of the session (or until you change it again). Before returning the "telnet>" prompt to you, *telnet* will print the new escape character. The *escape* command can be used with or without a connection to a remote host.

status (s)

Prints *telnet* status information. This command prints the following information on your screen: the name of the remote host (if applicable) and the current escape character. The *status* command can be used with or without a connection to a remote host.

negotiate (n)

Negotiates various *telnet* options. This command is not used by most *telnet* users. For complete information on the negotiate command, refer to the **UNIX PC Enhanced WIN/3B Programmer's Reference Manual**. The *negotiate* command can only be used if a connection to a remote host has been established.

options (opt)

Displays the options that you negotiate using the *negotiate* command. The *options* command can be used with or without a connection to a remote host.

? (?)

Displays help information. When used without an argument, the *?* command prints a help summary. The *?* command can take a *telnet* command as an argument. If an argument is specified, *telnet* will print the help information about the specified *telnet* command. The *help* command can be used with or without a connection to a remote host.

3.1.1 Invoking Telnet's Command Mode from the UNIX Shell**Format:**

\$ telnet

Invokes the *telnet* program in command mode. *Telnet* will respond with the "telnet>" prompt. This prompt indicates that you are in *telnet's* command mode.

Example:

In the example below, *telnet* is invoked in command mode, and a connection is opened with *twghost2*.

\$ telnet

telnet> open

(to) twgvax2

Trying...

Connected to twghost2

Escape character is '^'.

UNIX System V (TWG-TCP/IP) (3b2comk)

login:

3.1.2 Invoking Telnet's Command Mode from Telnet's Input Mode

Telnet's command mode can be invoked from input mode by preceding one of the *telnet* commands with the *telnet* escape character. The *telnet* escape character is set to a tilde ("~") by default. It is possible to change the *telnet* default escape character.

Format:

\$ escape command

Issues *command* to *telnet*, where *escape* is the *telnet* escape character and *command* is one of the valid *telnet* commands.

Example:

In the example below, the user has logged into *twghost1* via *telnet* in the input mode. The tilde (~) is the escape character. It is used to invoke the *telnet* command mode momentarily so that the user can determine the current status of the connection.

```
$ ~  
telnet> status  
Connected to twghost1  
Escape character is '~'.  
$ _
```

4. TRANSFER PRIVATE OR PUBLIC FILES (ftp)

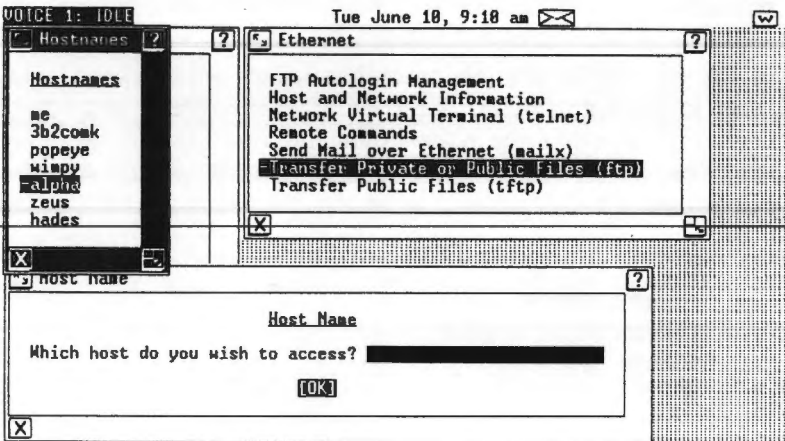
ftp (file transfer protocol) is the most powerful file transfer application available among the Department of Defense (DoD) standard TCP/IP protocols. It allows you to transfer files between hosts on your network that support TCP/IP, regardless of the host's operating system.

To invoke *ftp*:

1. Select | Transfer Private or Public Files (ftp)|

in the Ethernet window and press <Enter>.

You see the Host Name window:



2. Type *ftp* or press <Cmd>, highlight |ftp| and press <Enter>.

NOTE: You can also press <Slect/Mark> to step through a list of known hosts, stopping when *ftp* appears.

3. Press <Enter> to invoke *ftp*.

ftp will search its files, find the necessary Internet address, and attempt to make the connection. If the remote host is not on line, you will be placed in command mode. If the remote host is on line and the connection is made, your UNIX PC will open the FTP Command window.

The remote host will prompt you for your user name and password, and will check your login profile for authentication before granting you access to its files.

When you clear the connection by logging off the remote host, *ftp* will close the FTP Command window and return you to the Ethernet Command window.

NOTE: You can go directly to *ftp* command mode by pressing <Enter> without choosing a host.

The prompt-driven command mode provides you with many file manipulation tools, including moving, deleting, and listing files, and showing file transfer statistics.

4.1 USING THE *ftp* COMMANDS

The file transfer protocol (*ftp*) is used to copy (transfer) files between your UNIX PC and a remote host.

To transfer files *ftp* must know which host systems are available on your network. The remote host to which you would like to connect must be specified to the *ftp* program by using the actual Internet address of the remote host or one of the aliases that has been assigned to that host by the Network

Administrator. Your UNIX PC does this automatically when you specify the remote host in the Host Name window.

The Internet address and valid aliases for each host on the network are contained in the hosts file located in */etc/hosts*. To transfer files via *ftp*, you must have a login account on the remote host.

When *ftp* is in command mode, the program directly interprets a variety of commands. Once *ftp* has been invoked in command mode, you must exit from *ftp* by using the *bye* or *quit* command.

4.1.1 *ftp* Command Listing

A summary of the *ftp* commands is listed below. *ftp* commands can be abbreviated by entering the first unique characters of the command name. A quick reference table of the *ftp* commands can be found in Appendix B of this user guide.

!

If your System Administrator has given you "expert" status, this command will return you to the shell. To terminate the shell and return to *ftp*, press **<CTRL> + <D>**. This command is only available from *ftp's* command mode.

bye

Closes the connection to the remote host and exits from the *ftp* program returning you to the Ethernet Command window. The *quit* command is an alias for the *bye* command.

close

Closes the connection to the remote host but does not exit the *ftp* program. Once disconnected, it is possible to open a connection to a new host using the *open* command. For more complete details, refer to the UNIX

PC Enhanced WIN/3B Programmer's Reference Manual.**delete *rfile***

Deletes *rfile*, where *rfile* is the name of a file on the remote host.

dls [*rdir*] [*lfile*]

Retrieves a listing of the names of all the files in directory *rdir*, where *rdir* is a valid working directory on the remote host, and *lfile* is the name of a file on the local host. The listing is put into file *lfile* on the local host. If *rdir* is not specified, the current working directory on the remote machine is used. If *lfile* is not specified, the listing is displayed on the standard output. The listing that this command produces is in long format and includes the following information: file name(s), length, owner, permissions, and the date and time of last modification.

get *rfile* [*lfile*]

Retrieves a copy of *rfile* and copies it into *lfile*, where *rfile* is the name of a file on the remote host and *lfile* is the name of a file on the local host. The transfer is done in *ascii* mode unless another mode has been specified (i.e., binary mode). If *lfile* is omitted, *rfile* is copied into a file called *rfile* on the local host. If *verbose* is on, various statistics including the number of bytes transferred, the elapsed time in seconds, and the effective baud rate are printed to the standard output upon completion of the file transfer.

help [*command*]

Prints help information on *command*, where *command* is a valid *ftp* command. If *command* is not specified, *ftp* prints a list of its known commands. The ? command is an alias for *help*.

ls [*rdir*] [*lfile*]

Retrieves a listing of the names of all the files in directory *rdir*, where *rdir* is a valid working directory on

the remote host, and *lfile* is the name of a file on the local host. The listing is put into file *lfile* on the local host. If *rdir* is not specified, the current working directory is used. If *lfile* is not specified, the listing is displayed on the standard output. The listing that this command produces is in short format, resembling that of the UNIX *ls* command.

open host

Opens a connection to *host*, where *host* is a valid name of a remote host.

put lfile [rfile]

Stores a copy of *lfile* in *rfile*, where *lfile* is the name of a file on the local host and *rfile* is the name of a file on the remote host. The transfer is done in *ascii* mode unless another mode has been specified (i.e., binary). If *rfile* is omitted, *lfile* is copied into a file called *lfile* on the remote host. If *verbose* is on, various statistics including the number of bytes transferred, the elapsed time in seconds, and the effective baud rate are printed to the standard output upon completion of the file transfer. The *send* command is an alias for the *put* command.

pwd

Prints the name of the current working directory on the remote host.

status

Shows the current connection status, transfer mode, messages, hash, bell, and other options of the current *ftp* session.

user name

Initiates the login process on the remote host, where *name* is a valid login name on a remote host. The *ftp* program will prompt you for your password to complete the login process.

verbose

Switches the status of *verbose* (from off to on or on to

off). When *verbose* is turned on, all responses from the *ftp* server are displayed to the user. Verbose is turned off by default.

? [command]

Prints help information on *command*, where *command* is a valid *ftp* command. If *command* is not specified, *ftp* prints a list of its known commands. The *help* command is an alias for *?*.

4.1.2 *ftp* Options Summary

There are several options to the *ftp* command which may be specified on the command line.

- *-v* turns on verbose mode, showing all responses from the remote server and data transfer statistics.
- *-n* restrains *ftp* from attempting autologin upon initial connection.
- *-i* turns off interactive prompting during multiple file transfers.
- *-d* enables debugging.
- *-g* disables globbing.

For complete information on these options, refer to the **UNIX PC Enhanced WIN/3B Programmer's Reference Manual**.

4.1.3 Invoking the *ftp* Command from the UNIX Shell

Format:

\$ *ftp* [options]

Invokes the *ftp* program in command mode. When the *ftp* program responds with its "*ftp*>" prompt, you can issue any *ftp* command not relating to file transfer, such as *help*, *status*, etc., but it is usually a good idea to log into the remote host

using the *open* command before proceeding with the *ftp* session. Once a connection has been opened, you can use any of the *ftp* commands. To terminate the *ftp* session, you must issue the *bye* command (or one of its aliases).

\$ ftp [options] [hostname]

Invokes the *ftp* program in command mode with a connection to *hostname*, where *options* are valid *ftp* options, and *hostname* is the valid name of a remote host. The specified *options* will be used. The *ftp* program will respond with the login sequence. To terminate the *ftp* session, you must issue the *bye* command (or one of its aliases).

Example:

c

The following example shows how a user invokes *ftp* in command mode. The user then opens a connection with *twghost1*, transfers the file *personnel* from *twghost1* to the local host, quits *ftp* with the *bye* command, and returns to the command prompt of the local host.

\$ ftp

FTP User Process (Version 2.0)

ftp> open

(to) twghost2

Name (host. logname):

Password (host. logname):

ftp> get

(remote filename) personnel

(local filename) personnel

Transfer complete.

ftp> bye

\$ _

In the example below, the user starts *ftp* and opens a connection with *twghost2* in the same command line. The user

then gets file *personnel* and quits *ftp*.

```
$ ftp host2
```

```
FTP User Process (Version 2.0)
```

```
Name (host. logname):
```

```
Password (host. logname):
```

```
ftp> get
```

```
(remote filename) personnel
```

```
(local filename) personnel
```

```
Transfer complete.
```

```
ftp> bye
```

```
$ _
```

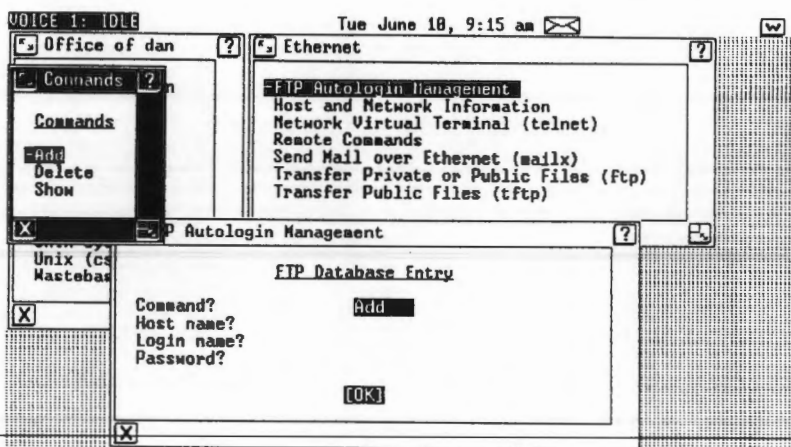
5. FTP AUTOLOGIN MANAGEMENT

To manage your *ftp* autologin privileges on remote hosts:

1. Select | FTP Autologin Management|

in the Ethernet window and press <Enter> .

You see the FTP Autologin Management window:



2. Press <Cmd> to select one of the options described below.

You must already have login privileges on the remote host

before you can use these options.

Add enables your automatic login privilege on a remote host.

Delete disables your automatic login privilege on a remote host.

Show displays a list of the remote hosts on which you have automatic login privileges.

3. Answer the rest of the questions about the remote host. Press < Return > after answering each question.
4. Press < Enter > when all questions are answered.

5.1 THE ftp INITIALIZATION FILE (.netrc)

The *ftp* program requires that you log into the remote host in one of two ways. If the *ftp* program has been invoked without a hostname as an argument, you can log into the remote host using the *user* command.

Another way to log into the remote host is to use a *.netrc* file. Your UNIX PC uses the *.netrc* file to automatically handle your login when you choose a remote host from the Host Name window.

The *.netrc* file is created and modified when you choose *Add* or *Delete* from the FTP Autologin Management window, and the data is displayed when you choose *Show*.

Your *.netrc* file is kept in your home directory.

Format:

The *.netrc* file can be also be created using a file editor, such as *vi*. The format of an entry in the *.netrc* file is

```
machine xxx login yyy [password zzz]
```

where *xxx* is the official name (as opposed to an Internet address or an alias) of a remote host, *yyy* is a your valid login name on *xxx*, and *zzz* is your valid password for login name *yyy*. This file automatically logs login name *yyy* into host *xxx* when an *ftp* connection has been opened. Each entry on a line is separated by blank space, whether it be single or multiple spaces or tabs.

NOTE: The *official* name of the host must be listed in the *netrc* file.

Each line in the *netrc* file contains information on a different host on the network. There is no limit on the number of lines in a *netrc* file.

Since the *netrc* file can list the passwords for accounts that are listed in the file, your *netrc* file should be protected so that it is readable only by you. When you use the FTP Autologin Management window functions, this is done for you automatically.

If you do not want to list your password in the *netrc* file, the *password* parameter may be omitted, and *ftp* will prompt you for your password during the login process.

Example:

The sample *netrc* file shown below provides an automatic login for user *pattie* on *host1*. User *pattie's* password on *host1* has been omitted, which will cause *ftp* to prompt her for her password during the login process.

```
machine host1 login pattie
```

6. THE REMOTE COMMANDS

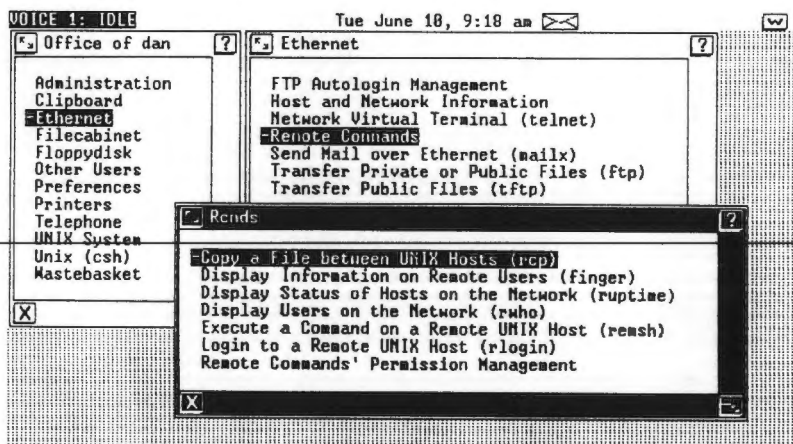
WIN/3B LAN contains a set of extra utilities for 4.2 BSD UNIX and other UNIX versions running a version of WIN software. These commands, called Remote Commands, provide status and other information about remote hosts on your network, as well as other utilities.

To access the Remote Commands:

1. Select Remote Commands

in the Ethernet window and press <Enter> .

You see the Rcmds window:



6.1 COPY A FILE BETWEEN UNIX HOSTS (rcp)

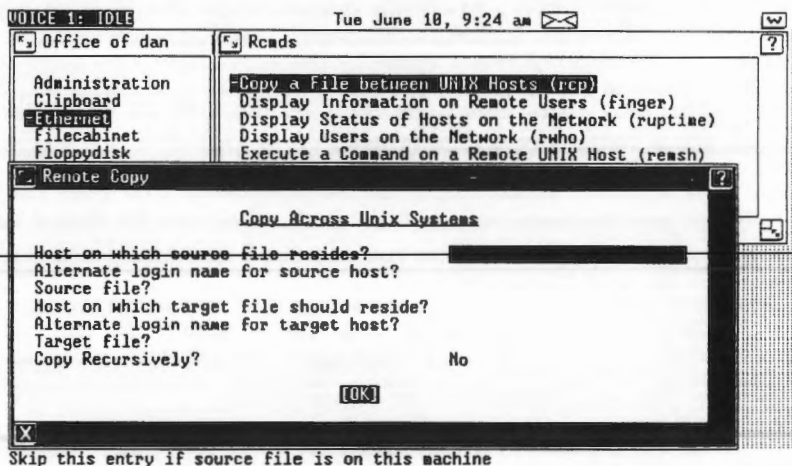
This function allows you to copy files from host to host. The file on your UNIX PC can be copied to a remote host (or vice versa), or a file can be copied from one remote host to another. The only stipulations are that you must have log in authorization on all systems involved. The remote host **MUST** be running 4.2 BSD or later versions of UNIX, or a version of the WIN software, such as WIN/VX or WIN/3B. In addition, you must have log in authorization on all systems involved.

To access the remote copy function:

1. Select | Copy a File between UNIX Hosts (rcp)|

in the Rcnds window press <Enter> .

You see the Remote Copy window:



2. Answer each question on the Copy Across Unix Systems form. Press <Return> after answering each question.

The first three questions in the window refer to the host from which the file is to be taken (the source).

- **Host on which source file resides?** The program needs the official hostname of the host from which you are copying the file. If you are copying a file from your UNIX PC to another host, DO NOT enter your hostname. Press <Return>.
- **Alternate login name for source host?** What is the user name to be used by the program when logging into that remote host. If you are using the same user name for login from the source (or you are copying the file from

your own UNIX PC host) you only need to press <Return>. Otherwise enter the user name for the source host.

- **Source File?** What is the name of the file to be copied? If the file is not in your home directory on the source host, then you must also enter the path name.

The next three questions refer to the host on which the file is to be copied (the target).

- **Host on which target file should reside?** To what host are you copying the file. If the file is to be copied to your UNIX PC, then you can leave this field blank and press <Return>.
- **Alternate login name for target host?** If you are copying the file to a host under a different user name, that user name is to be entered here.
- **Target file?** The file name to be given to the copy on the target host.

The last question is meaningful only if the source file is a directory. It means, "Do you want to copy all files and any subtrees rooted in that directory?" If the answer is "no", you are assumed to have made an error since it makes no sense to copy a directory only and not its contents. So no files are transferred.

- **Copy Recursively?**

3. Press <Enter> when all questions are answered.

6.1.1 Calling the Remote Copy Function from UNIX

If your System Administrator has given you "expert" status on the system, you can use the remote copy function by invoking *rcp* at the UNIX prompt.

It is possible to copy files in a directory recursively via the *rcp* *-r* option. *rcp* is also able to handle third party copies, where neither the source nor target files are on the local host. For example, suppose that you are logged into *host1* on the network. By using the *rcp* command, it is possible to copy files between two additional hosts on the network, i.e., *host2* and *host3*. Refer to the **UNIX PC Enhanced WIN/3B Programmer's Reference Manual** for details.

6.1.1.1 Specifying Filenames to the *rcp* Command

The following conventions must be used to specify filenames to the *rcp* command.

Specifying Remote Filenames to *rcp*:

- Remote filenames are specified in the form *rhost:path*, where *rhost* is the name of the remote host, and *path* is the pathname of the file on the remote host. If *path* is not an absolute pathname, it is interpreted relative to your home directory on *rhost*.
- It is possible to use UNIX metacharacters to specify a remote filename. For complete details, refer to the **UNIX PC Enhanced WIN/3B Programmer's Reference Manual**.

Specifying Local Filenames to *rcp*:

- Local filenames are specified by using the actual filename on the local host.
- Local filenames cannot contain *colons* (:).

6.1.1.2 Invoking the *rcp* Command

Format:

```
$ rcp file1 file2
```

Copies *file1* to *file2*, where *file1* is the name of either a remote or local file and *file2* is the name of either a remote or local file.

Example:

```
$ rcp -r file ... directory
```

Copies the list of files, where *file...* may be files and/or directories on the local or remote host, to a directory on a local or remote host.

Example:

This copies file *proposal* on the local host into file *report* on host *twghost2*.

```
$ rcp proposal twghost2:report
$_
```

Example:

The command below copies file *proposal* on *host1* into file *report* on *host2*.

```
$ rcp host1:proposal host2:report
$_
```

6.2 DISPLAY USERS ON THE NETWORK (*rwwho*)

This function will print the hostname, login name, terminal number, and login time for each user on your local area network. The information is gained from a report that comes from each host on the local area network every two minutes. If no report is received from a host for five minutes, it is assumed that the host is down, and it is omitted.

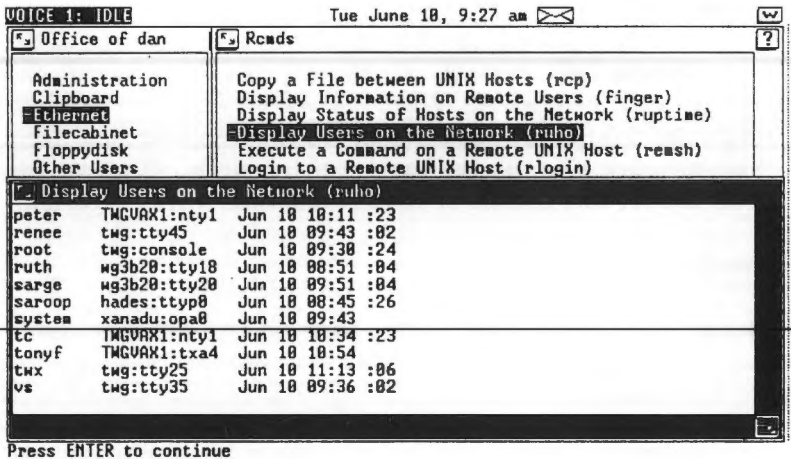
If a user has not typed into the system for one minute or more, then the report states the idle time for that user. If a user has not typed into the system for an hour or more, the user will be omitted from the listing.

To get a list of the users on the network:

1. Select | Display Users on the Network (rwho)

in the Rcnds window and press <Enter> .

You see a Display Users on the Network (rwho) window like this one:



6.2.1 The *rwho* Command

This function uses the *rwho* command to collect and report the status of the users on the network.

6.2.2 *rwho* Options Summary

The omission of users with more than one hour of idle time can be overridden by using the *rwho -a* option. The output from this command is similar to the UNIX *who* command, except that *rwho* includes every user on the local area network.

6.2.3 Invoking the *rwho* Command from UNIX

Format:

```
$ rwho [-a]
```

prints the hostname, login name, terminal number, and login time for each user on the local area network. If a user has been idle for one minute or longer, idle time will be reported.

Example:

The example below reports three users on two different hosts.

\$ rwho	
mercedes	twghost1:ntty6Dec 1811:05
mpd	twghost2:tty4Dec 1917:24
tonyf	twghost1:pty2Dec 1916:37

6.3 DISPLAY STATUS OF HOSTS ON THE NETWORK (*ruptime*)

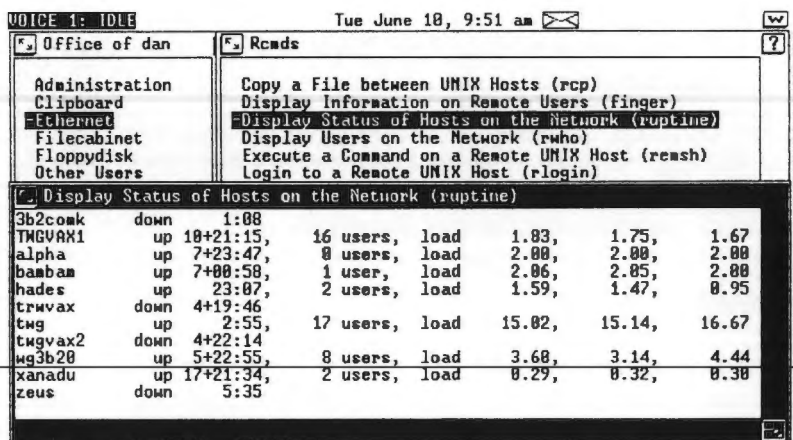
This function shows the status of the hosts on your network.

It tells which remote hosts are up, how long they've been up, how many users are using that host, plus other information useful to the System Administrator. To access this information:

1. Select | Display Status of Hosts on the Network (ruptime)|

in the Rcmds window and press <Enter> .

You see the Display Status of Hosts on the Network (ruptime) window:



Press ENTER to continue

6.3.1 Getting Status Information from UNIX

The *ruptime* command will show the host status of all machines

on the local area network. The command prints a table containing the following information for each host on the local area network: hostname, current host status (up or down), amount of time that the host has been up or down, the number of users on the host, and the load average. *runtime* normally sorts the list it produces by hostname.

6.3.1.1 *runtime* Options Summary

Several options enable the list to be sorted in a different way:

- *-l* sorts by load average.
- *-t* sorts by uptime.
- *-u* sorts by the number of users.
- *-a* includes users that have been idle for more than one hour in its count.

For complete information on *runtime* options, refer to the UNIX PC Enhanced WIN/3B Programmer's Reference Manual.

6.3.1.2 Invoking the *runtime* Command

Format:

\$ runtime [option]

Prints the status of all machines on the local area network.

Example:

\$ runtime

twg	up	8:20	1 user	load	0.08	0.11	0.12
host1	up	13+00	2 users	load	1.78	1.89	1.92
host2	down	1+01:30					

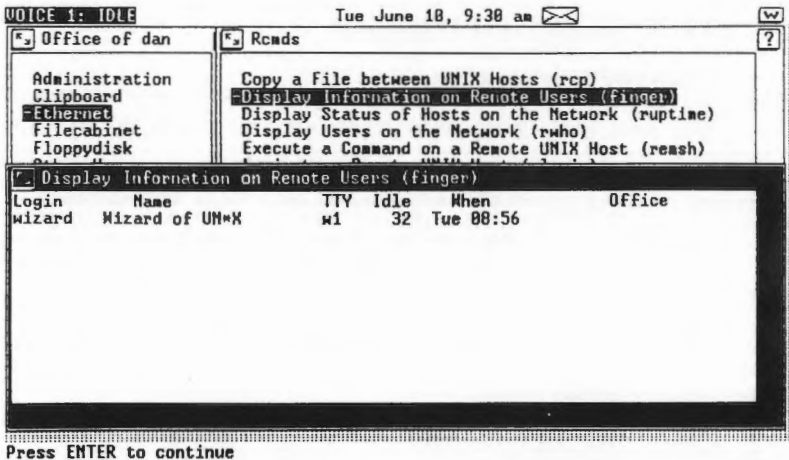
6.4 DISPLAY INFORMATION ON REMOTE USERS (finger)

This function tells you who is logged into your network. The information given is

- **Login**--The user name being used
- **Name**--The user's real name
- **TTY**--The port being used to access the UNIX PC by that user
- **Idle**--How long it has been since the user has made any access to data on the host (how long the user has been idle)
- **When**--When the user logged onto the machine
- **Net Site**--Where the user is physically located

To access this function:

1. **Select | Display Information on Remote Users (finger)|**
in the Rcnds window and press <Enter> .
2. When the Host Name window opens, type the name of the host you wish to access, or press <Cmd> , select the host, and press <Enter> .
3. **Press <Enter>**
4. You see a Display Information on Remote Users (finger) window like this one:



6.4.1 The finger Command

The host information display utilities use the *finger* command. *finger* is an information lookup program, which, if invoked without options, lists the login name, full name, terminal name, write status, idle time, login time, office location, and phone number (if known), for each current UNIX user on the local host. The *finger* command can also be given an argument, causing it to print information about specific user(s) on the local or remote hosts.

6.4.1.1 Finger Options Summary

The options to the *finger* command are

- *-m* matches arguments for login name only
- *-l* forces long output format
- *-p* suppresses printing of the *.plan* files
- *-s* forces short output format

For complete information on the options to the *finger* command, refer to the **UNIX PC Enhanced WIN/3B Programmer's Reference Manual**.

6.4.1.2 Invoking The finger Command

Format:

\$ finger

Lists the login name, full name, terminal name, write status, idle time, login time, office location and phone number, for each current UNIX user on the local host.

\$ finger [options] name ...

Lists the login name, full name, terminal name, write status, idle time, login time, office location and phone number for each current UNIX user on the local host, where *options* is a valid *finger* command option, and *name* is the login name or full name of a valid user on the local host. More than one *name* may be specified on the command line. *Names* should be separated by blanks. The output is displayed in the format specified by *option*. The listing is displayed in long format by default.

\$ finger name@host

Lists the login name, full name, terminal name, write status, idle time, login time, office location and phone number and net site for each current UNIX user on the local host, where *host* is the name of a valid host on the network, and *name* is the login name or full name of a valid user on the local host. No more than one *name* and/or *host* may be specified on the command line. The listing is displayed in long format by default. Note that the @ metacharacter must be escaped by a backslash unless you have redefined the *kill* character by *stty*. Otherwise the @ metacharacter will be interpreted by UNIX as a line kill.

Example:

The example below lists the login name, full name, terminal name, write status, idle time, login time, and office location for each current UNIX user.

\$ finger

Login	Name	tty	Idle	When	Office
mpd	MelDio	po.3b	41	Thu 17:24	14
karen	K.Jone	p1.3b	2:05	Thu 16:24	2
mpd	MelDio	p2.3b	4:35	Thu 13:03	15

Example:

The example below lists the login name, full name, terminal name, write status, idle time, login time, office location, and phone number for user *mpd* on the local host. The listing is displayed in long format by default.

\$ finger mpd

Login name: mpd	In real life: MelDio
Directory: /u1/mpd	43 minutes idle time
Office: 14	Phone: x3772

Plan: to make a plan by Tuesday

On since Dec 19 17:24:42 on po3b

Example:

The example below lists the login name, full name, terminal name, write status, idle time, login time, office location, and phone number for user *dave* on host *host1*. The listing is displayed in long format.

\$ finger dave@host1

Login name: dave	In real life: Dave Long
Directory: /u1/dave	14 minutes idle time
Office: 2	Phone: x5667

Plan: give everybody a huge raise

On since Jan 19 17:24:42 on p4.cu

6.5 EXECUTE A COMMAND ON A REMOTE UNIX HOST (remsh)

This function allows you to execute one command from the shell on a remote host. The remote host must be using UNIX and you must be authorized to log into that system.

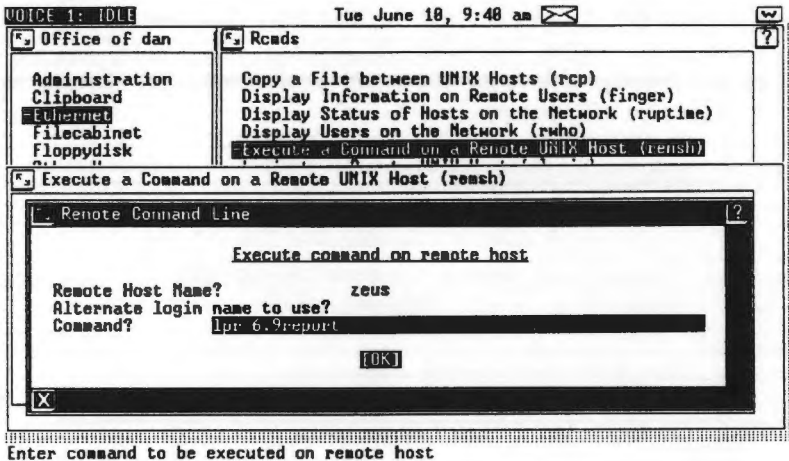
Control will return to you on the local host after execution of the command. For this reason, you cannot execute interactive commands such as *vi*.

To execute a command on a remote host:

1. Select | Execute a Command on a Remote UNIX Host
(remsh)|

in the Rcnds window and press <Enter> .

You see a Remote Command Line window like this one:



2. Enter the name of the remote host, or press <Cmd>, select the desired host, and press <Enter>.
3. Enter an Alternate login name only if your login on the remote host differs from that of the local host.
4. Enter the command you wish to execute on the remote host.
5. Press <Enter> to execute the command.

6.5.1 Using Remote Execute from UNIX

WIN/3B LAN uses the *remsh* command to execute the UNIX

command on a remote host on the local area network. This command can only be used when the remote host is running UNIX. By default, *remsh* allows the user to execute a command in an account with the identical login name on the remote host. Equivalent login names (in the sense of *rlogin*) may be specified by using the *-l* option to *remsh*. The *remsh* command can be run in the background.

When executing a command on a remote host using *remsh*, interrupt, quit and terminate signals are passed to the remote host. Shell metacharacters (&, ?, *, |, \, and ;) that are not quoted are interpreted on the local machine, while quoted metacharacters are interpreted on the remote machine.

NOTE: Interactive commands, such as *vi*, cannot be run using *remsh*. These commands should be invoked with *rlogin* or *telnet*.

6.5.1.1 Remsh Options Summary

There are two options to the *remsh* command.

- *-l* is used to specify a different remote login name to *rlogin*.
- *-n* is used to redirect the input of *remsh* to */dev/null*.

For complete information on these options, refer to the **UNIX PC Enhanced WIN/3B Programmer's Reference Manual**.

6.5.1.2 Invoking the remsh Command

Format:

\$ remsh rhost [-l] [-n] command

Initiates a remote login to *rhost* and executes *command*, where *rhost* is the name of a remote host, and *command* is a valid UNIX command. *Remsh* will terminate when the command has

finished processing, and you will be returned to the working environment of the local host. If *command* is omitted, *remsh* will initiate a remote login.

Example:

In the example below, a remote login to *twghost2* is performed with *remsh*. The file *proposal* on *twghost2* is sent to that host's line printer as a background process. Once the command is executed on *twghost2*, the user is returned to the command prompt of the local host.

```
$ remsh twghost2 lpr proposal &
```

6.6 LOGIN TO A REMOTE UNIX HOST (rlogin)

This function provides you the means to easily log into a remote host on your network. The remote host **MUST** be running 4.2 BSD or later UNIX.

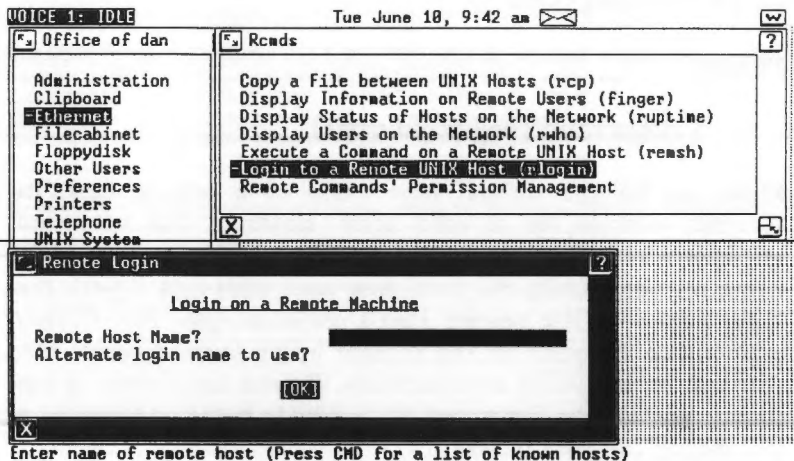
Once the connection has been established, you will have the same privileges and capabilities as a user who is directly connected to that host. You must have a login account on that remote host.

To log into a remote host:

1. Select | Login to a Remote UNIX Host (rlogin) |

in the Rcnds window and press <Enter> .

You see the Remote Login window:



2. Type the name of the host you wish to access, or press <Cmd>, select the desired host, and press <Enter>.
3. Enter an alternate login name only if your login on the remote host is different from that of the local host.
4. Press <Enter> to invoke *rlogin*.

6.6.1 The *rlogin* Command

The *rlogin* command provides login to remote hosts on your network that run 4.2 BSD UNIX or a derivative of the 4.2 BSD networking package. Once a connection has been established, you will have full privileges and all of the capabilities of a user

that is directly connected to the remote host. However, you must have a login account on the remote host in order to use that machine.

6.6.1.1 Invoking rlogin

Format:

```
$ rlogin rhost [-ec][-llogin name]
```

Rhost can be the official host name of a remote host, its Internet address, or a valid alias. Entering this command initiates a remote login to *rhost*. *Rlogin* looks up your login name on your UNIX PC local host and compares it with the login names in the remote host's */etc/hosts.equiv* file. If your login name is found in the remote host's */etc/hosts.equiv* file, you will be logged in automatically. If your login name is not found, you will be prompted for the login name and password you use on the remote host.

Example:

In this example, a remote login to is initiated to *twghost1*. Since the local host's host name is in *twghost1*'s */etc/hosts.equiv* file, and since the user's login name is the same on both hosts, the user is logged into *twghost1* automatically.

```
$ rlogin twghost1
```

```
.....  
TWGHOST1  
.....
```

```
$ _
```

Example:

In the example below, the user's login name on *twghost1* is not

the same as the login name used on the local host. Therefore the user is prompted for a login name on *twghost1*.

```
$ rlogin twghost1
```

```
login:
```

```
pete
```

```
password:
```

```
*****
```

```
TWGHOST1
```

```
*****
```

```
$ _
```

Once the *rlogin* connection has been established and you are logged in, it will appear as if you are hardwired to the remote host. To freeze and release the screen display after you have logged in, use **<CTRL>+<S>** and **<CTRL>+<Q>**. To disconnect from the remote host, do not issue the UNIX *logout* command. Instead, issue the *tilde period* command (**~.**), where *tilde* is the default escape character. This will terminate the *rlogin* connection and return you to the working environment of your local host.

The **-e** option allows you to change the *rlogin* escape character from the tilde character (**~**) to whatever character you type after **-e** (represented as the variable *c* in the command format above).

Example:

This initiates a remote login to host *twghost1* and changes the escape character from **~** to **&**.

```
$ rlogin twghost1 -\&
```

```
$ _
```

For more information about this option, refer to the **UNIX PC Enhanced WIN/3B Programmer's Reference Manual**.

The **-l** option allows you to log in automatically even if your login name on the remote host does not match your username on your local UNIX PC host. To use this option, you must replace the variable *login name* shown in the command format above with the username of your account on the remote host. But you must first include the login name of your UNIX PC account in a file called *~/.rhosts* in your home directory on the remote host.

6.6.1.2 Autologin Check

The *rlogin* utility passes on your local terminal type to the remote host automatically. *Rlogin* also supports automatic login. If automatic login is not provided when you use the *rlogin* command, then you will be prompted for your login name and password. Otherwise, autologin allows you to log in to your user account on a remote host without having to type in your login name or password. There are two files on the remote host that manage autologin: a system file called */etc/hosts.equiv*, and a user file called *~/.rhosts*.

The */etc/hosts.equiv* file is set up on each host by the Network Administrator. Contained within the file are the names of other network hosts that are granted access to the host. If your local host's hostname is in the remote host's */etc/hosts.equiv* file, and if your login name is the same on both hosts, then you will be logged in to the remote host automatically. But if your local host's host name is not in the remote host's */etc/hosts.equiv* file, you will not be allowed access at all. (If you are denied access to a remote host, consult your Network Administrator.)

For example, if your local host is named "eng," you are logging in to a remote host named "dandy," and your login name on both hosts is "kmd," when you issue the command

```
$ rlogin dandy
```


The remote host "dandy" checks its */etc/hosts.equiv* file to see if host "eng" is listed. Once it finds "eng," it checks to see if the login name "me" is in its password file. If it is, you will be logged in automatically. You will not be asked to provide a password, because you already authenticated your account by entering your password when you logged in to the UNIX PC local host.

If the login name of your account on your local UNIX PC host differs from the login name of your account on a remote host, to log in automatically you must create a *~/rhosts* file in your working directory on the remote host, and then use the *-l* option whenever you issue the *rlogin* command.

To accomplish all this, begin by logging into the remote host, and then use a text editor to create a *~/rhost* file. At the top of the file, type the official host name for the UNIX PC machine to which your terminal is directly connected.

The next time you try to log in to the remote host, type *rlogin*, a blank space, *-l*, then the login name of your account on the remote host. After you enter this command, *rlogin* will be flagged by the *-l* option to check into the *~/rhosts* file of your working directory on the remote host. If your login name for the UNIX PC is found there, you will be logged in without being prompted for your password.

For example, if your login name on your UNIX PC is "pete," but your username on a remote host is "kmd," you would include "pete" as a login name in a *~/rhosts* file on the remote host and afterward log in to the remote host with the following command:

rlogin -lkmd

This will check "me"'s home directory for the file *~/rhosts*, and when it finds the host name for your UNIX PC followed by "pete" within the file, you will be logged in automatically.

The `~/rhosts` file that you create in your home directory can also contain the names of remote hosts and users on those hosts for whom you wish to grant access to your account. Those users would then enter the `-l` flag followed by your login name on the remote host when using the `rlogin` command. Again, `rlogin` checks into the `~/rhosts` file in your home directory, and if the host name and login name for the person invoking `rlogin` is found, then that person is automatically logged in to your account. Thus, you can create an `~/rhosts` file on your local host to allow and even disallow remote users automatic login access to your directory.

When creating or revising your `~/rhosts` file, keep the following rules in mind.

- Each host name must begin on a separate line.
- A login name must follow the host name, separated from it by one space.

Example:

The example shown below illustrates the use of both the minus sign and the wildcard character in a sample `~/rhosts` file.

```
# rhosts file
eng pete
3B20 brian
market
3B2 cs-eng
```

Example:

This initiates a remote login to host `twghost1`. The `rlogin` utility is flagged to check the `~/rhosts` file in the home directory of user "me" for the host name and login name of the host and the user initiating the remote login request. If they are found, the user is allowed access without being prompted for a password.

```
$ rlogin twghost1 -lme
```

```
$ _
```

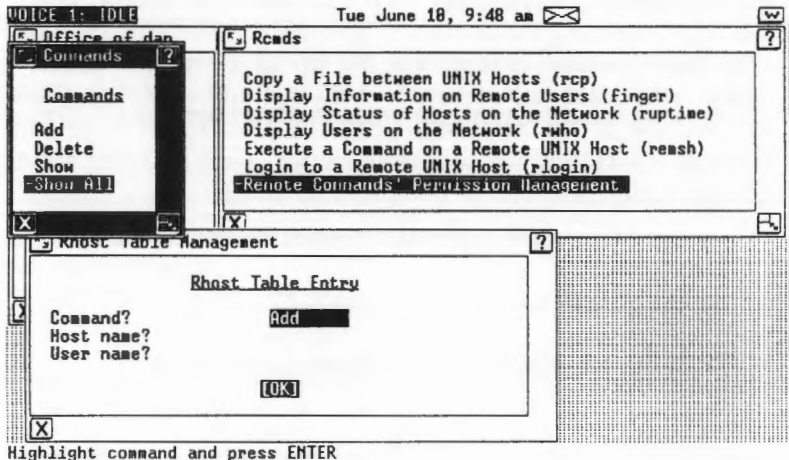
6.7 REMOTE COMMANDS' PERMISSION MANAGEMENT

Use this command to maintain the list of users who can have access your local host account. You can add or delete users, or see a list of the users who currently have access to to your account. Executing this command causes *~/rhosts* to be edited automatically. To look at the list or update it:

1. Select | Remote Commands' Permission Management!

in the Rcnds window and press <Enter> .

You see the Rhost Table Management window:



2. Press <Cmd> and select one of the following commands:

Add adds a host and login name entry to your *.rhosts* file, adding access to the specified user on the specified host.

Delete deletes an entry from your *.rhosts* file, deleting access to the specified user on the specified host.

Show displays the entries in your *.rhosts* file. These are the only entries you can delete with the Delete command.

Show All displays the contents of your *.rhosts* and */etc/hosts.equiv* file.

3. To add or delete a user, Enter a host name, or press <Cmd> to select the host from the list of known hosts, then enter the login name for the user.
4. Press <Enter> to show the *rhost* table or to add or delete an entry in the table.

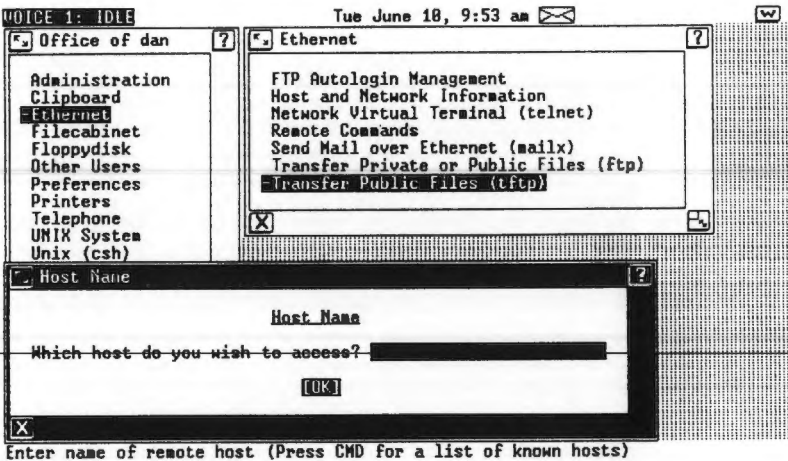
7. TRANSFER PUBLIC FILES (tftp)

To transfer unprotected files to a remote host:

1. Select Transfer Public Files (tftp)

in the Ethernet window and press <Enter>.

You see the Host Name window:



2. Type the name of the host you wish to access, or press <Cmd> to select the desired host and press <Enter>.

3. Press <Enter> to be connected to the remote host.

tftp uses a file transfer protocol (UDP) that is not as reliable as *ftp*.

Since *tftp* does not require a password, you can only access files that are "public". That means you can copy only files that are open to be read by everyone who has access to the remote host. Any files that you send to the remote host will also be "public"

7.1 *tftp* COMMAND SUMMARY

tftp [hostname]

Start local *tftp*. Specifying a hostname will connect that remote host to your local host.

connect [hostname]

Connect to a remote host.

mode [type]

Sets the mode of transfer, where type is binary or ASCII. ASCII is the default mode.

put [lfile] [rfile]

Send lfile from your host to rfile on the remote host.

get [rfile] [lfile]

Get the rfile from the remote host and place it in your directory as lfile.

quit

Quit *tftp* and return your own operating system.

verbose

Toggle on and off extra information about *tftp* activity.

status

Display status of the connection:

- mode of file transfer
- status of verbose
- status of tracing

8. SEND MAIL OVER ETHERNET (mailx)

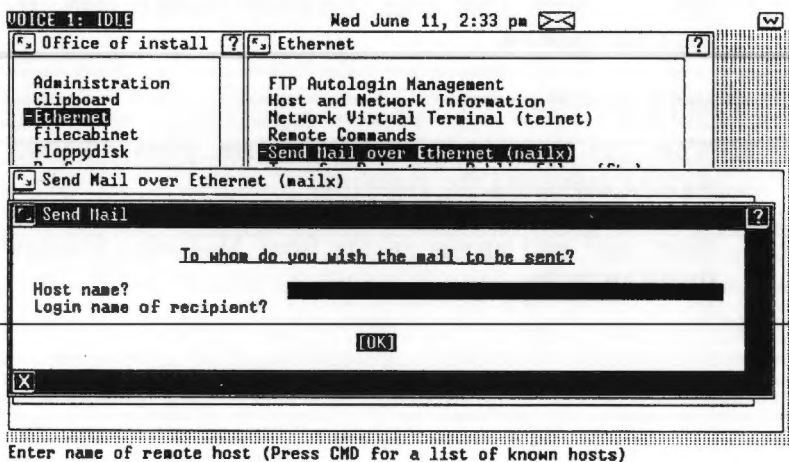
This function allows you to send mail to and receive mail from local and remote hosts on your network and remote networks.

To send mail:

1. Select | Send Mail over Ethernet (mailx)| from the Ethernet window

and press <Enter> .

You see the Send Mail window:



2. Type the name of the host to which you would like to send mail or press <Cmd>, select the desired host and press <Enter> .
3. Type the login name of the recipient of the mail.
4. Enter a subject
5. Press <Return>
6. Enter a message
7. Press <Return>
8. Press <CTRL> + <D>
9. Press <Enter> to send the mail.

To receive mail:

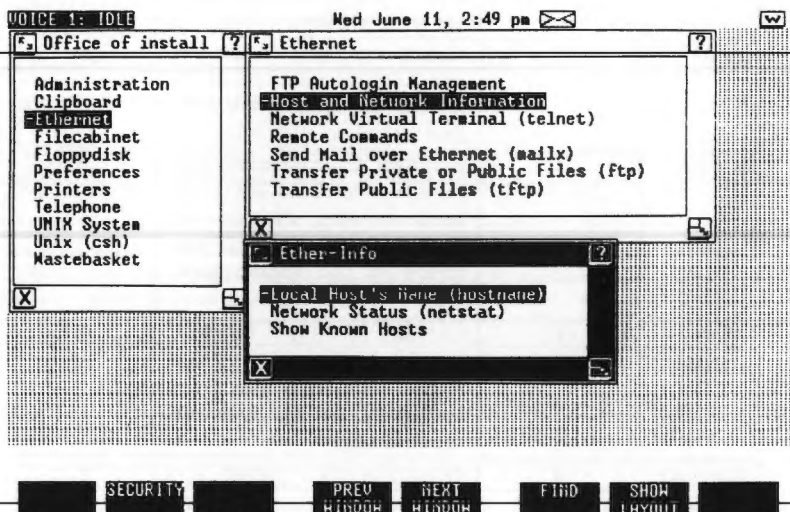
1. Select | Send Mail over Ethernet (mailx)| from the Ethernet window

and press <Enter> .
2. When the Send Mail window appears, press <Enter> without answering the questions.

Your mail will appear in the Send Mail over Ethernet (mailx) window.

9. HOST AND NETWORK INFORMATION

The Host and Network Information utilities in the Ether-Info window shown below provides information about the local host's name, the status of various network-related data structures, and the other hosts on the network.



These utilities are described fully in the **UNIX PC Enhanced WIN/3B Administrator's Guide**.

APPENDIX A. TELNET OPTIONS

Option	Abbreviation	Description	Connection Required
open	ope	Opens a virtual connection between a local or remote host.	n
close	c	Severs the connection from a remote host still leaving the user in the <i>telnet</i> program at the <i>telnet</i> prompt.	y
quit	q	Closes any open connection and exits the <i>telnet</i> program, leaving the users in the local host's working environment.	n
z	z	Suspends the connection and creates a shell in the local hosts working environment. To exit from this shell and return to the remote host, type <CTRL+<D>.	n
escape	e	Changes the escape character.	n
status	s	Prints the current <i>telnet</i> status: name of remote host, current escape character, and other settings.	n
negotiate	n	Negotiates <i>telnet</i> options.	y
options	opt	Displays options set by negotiate command.	n
?	?	Displays help information.	n

APPENDIX B. FTP COMMAND SUMMARY

Command	Format	Description	Remote or Local Host	Connect Req
append	append lfile rfile	Appends file <i>lfile</i> on local host to <i>rfile</i> file on remote host.	both	yes
ascii	ascii	Sets file transfer type to ascii .	neither	yes
bell	bell	Sets end of transfer bell on or off.	neither	no
binary	binary	Sets file transfer type to binary .	neither	yes
bye	bye	Closes any open connection to remote host and exits <i>ftp</i> program.	both	no
cd	cd rdir	Changes the working directory on the remote host to <i>rdir</i> .	remote	yes
close	c	Severs any connection, leaving the user in the <i>ftp</i> program.	both	yes
debug	debug	Changes the status of <i>debug</i> mode.	both	yes
delete	delete rfile	Deletes <i>rfile</i> on the remote host	both	yes

dir	dir [rdir] [lfile]	Produces long listing of <i>rdir</i> working directory on remote host. If <i>lfile</i> filename supplied, puts this listing in that file on local host.	both	yes
form	form format	Sets file transfer form to <i>format</i> .	neither	no
get	get rfile lfile	Transfers <i>rfile</i> from remote host and puts it in <i>lfile</i> on the local host.	both	yes
glob	glob	Sets file globbing on or off so file or pathnames can be processed for the metacharacters (*,?,[,],{,}).	neither	no
hash	hash	Sets <i>hash</i> status on or off.	neither	no
help	help command	Prints help information for <i>command</i> .	neither	no
lcd	lcd [ldir]	Changes the local host's working directory to <i>ldir</i> . If no directory is specified it changes by default to the home directory.	local	no
ls	ls [rdir] [lfile]	Produces a short listing of the contents of the <i>rdir</i> on the remote host. If <i>lfile</i> is specified it puts the listing in this file on the local host.	remote	yes
mdelete	mdelete rfile1 rfile2...	Deletes file <i>rfile1</i> and <i>rfile2</i> on the remote host	remote	yes

mdir	mdir rfile1 rfile2... lfile	Obtains a long listing of the files <i>rfile1</i> and <i>rfile2</i> on the remote host and puts this listing in <i>lfile</i> on the local host.	both	yes
mget	mget rfile1 rfile2 ...	Retrieves a copy of <i>rfile1</i> and <i>rfile2</i> and places them in files of the same name in the current working directory on the local host.	both	yes
mls	mls rdir1 rdir2 ... lfile	Obtains a listing of directories <i>rdir1</i> and <i>rdir2</i> and places it in <i>lfile</i> .	both	yes
mode	mode name	Changes the file transfer mode to <i>name</i> .	neither	yes
mput	mput lfile1 lfile2...	Puts a copy of the files <i>lfile1</i> <i>lfile2</i> in the working directory on remote host.	both	yes
open	open host [port]	Opens a connection to <i>host</i> .	both	no
prompt	prompt	Sets <i>prompt</i> to on or off.	neither	no
put	put lfile rfile	Copies <i>lfile</i> from the local host into <i>rfile</i> in the current working directory of the remote host.	neither	yes
pwd	pwd	Prints the name of the current working directory on the remote host.	remote	yes
quit	quit	Alias of <i>bye</i> .	both	no

quote	quote command	Passes <i>ftp</i> command for execution directly to the remote host, bypassing the local host.	remote	yes
recv	recv	Alias of <i>get</i> .	both	yes
remotehelp	remotehelp [command]	Displays the list of <i>ftp</i> commands recognized by the remote host. Where <i>command</i> is a valid <i>ftp</i> command, displays help information.	remote	yes
rmdir	rmdir rdir	Deletes directory <i>rdir</i> on the remote host.	both	yes
send	send file [rfile]	Alias of <i>put</i>	both	yes
sendport	sendport	Changes use of <i>port</i> commands		
status	status	Shows current connection status, transfer mode, messages, hash, bell, other options of the current <i>ftp</i> session.	neither	no
struct	struct name	Set file transfer structure to <i>name</i> .	neither	no
tenex	tenex	Sets file transfer type to <i>tenex</i> enabling communication with TENEX machines.	neither	no
type	type name	Sets the file transfer type to <i>name</i> .	neither	yes
user	user name	Initiates login sequence onto remote host.	remote	yes

verbose	verbose	Sets <i>verbose</i> on or off. All responses from <i>ftp</i> server are displayed.	neither	no
?	?command	Prints help on <i>ftp</i> <i>command</i> .	neither	no
!	!command	Executes <i>command</i> in local host's UNIX Shell.	neither	no
<INTERRUPT>		Kills any <i>ftp</i> operation in progress.	neither	no

APPENDIX C. NETWORK ERRORS

This appendix contains tables of most of the error messages associated with WIN/3B LAN. Next to them are possible causes and solutions. It is organized into five sections based on the probable message source. Within each section the errors are organized alphabetically according to the text of the error.

SYSTEM ERRORS

These occur when an improper operation occurs at the kernel level. They are produced by the lower level protocols. Many of these will be seen only by those in the process of developing new network applications with the SCI or TLI libraries.

Address already in use

Addresses must be unique to each host and are permitted to be used only once.

Address family not supported by protocol family

An address incompatible with the requested protocol was used.

Bad protocol option

A bad option was specified in a *getsockopt* or *setsockopt* call.

Can't assign requested address

This normally results from an attempt to *bind* a socket to an illegal address or to one already in use.

Can't send after socket shutdown

A request to send data was disallowed because the socket had already been shut down with a previous *shutdown* call.

Connection reset by peer

A connection was forcibly closed by a peer. This normally results from the peer executing a *shutdown* call.

Connection timed out

A *connect* request failed because the connected party did not properly respond after a specified period of time. This could indicate that the remote host is down.

Connection refused

No connection could be made because the remote host actively refused it. This usually results from trying to connect to a service that is inactive on the foreign host.

control block: symbol not in namelist

This error indicates a problem with the kernel symbol table. Contact AT&T Technical Support.

Destination address required

A required address was omitted from an operation on a socket.

Message too long

A message sent on a socket was larger than the internal message buffer.

Network dropped connection on reset

The host you were connected to crashed and rebooted.

Network is down

A socket operation encountered a dead network.

Network is unreachable

A socket operation was attempted on an unreachable network.

No buffer space available

An operation on a socket or pipe was not performed because the system lacked sufficient buffer space.

Operation already in progress

An operation was attempted on a non-blocking object that already had an operation in progress.

Operation now in progress

An operation was attempted that takes a long time to complete (such as a *connect*).

Operation would block

A blocking operation was attempted upon an object in the non-blocking mode.

Socket is already connected

A *connect* request was made on an already connected socket or a *sendto* or *sendmsg* request made on a connected socket specified a destination other than the connected party.

Socket is not connected

A request to send data was disallowed because the socket had already been shut down with a previous *shutdown* call.

Socket operation on non-socket

A socket operation was attempted on a non-socket device.

Socket type not supported

Support for the specified socket type has not been configured into the system or no implementation for it exists.

Software caused connection abort

A connection was aborted by the local host.

FTP AND TFTP ERRORS**Already connected to *hostname*, use *disconnect* first**

You are currently connected to remote host *hostname*, even though you may not be logged on. Either log in or *disconnect* and try again.

bad port number

A zero or negative port number was specified by the user with the *port* option on the *open* command.

can't find list of remote files, oops

When invoking a multiple command, a temporary file is created in the */tmp* directory. The system cannot find this file, or it was accidentally removed. Try the procedure again.

filename not a plain file

A directory or some special device was specified for a transfer operation when only a file was appropriate.

ftp/tcp: unknown service

The */etc/services* file does not contain an entry for the specified *ftp* service. An entry for this service should be created in this file. See *services(5)* in the **UNIX PC Enhanced WIN/3B Programmer's Reference Manual**.

Login failed

The user gave an invalid username or password. Try again.

Lost connection

The server on the remote host closed the connection. This could be caused by many different things, including the remote system crashing, automatic logout, etc.

No port available for data connection

This is an unlikely error. It could indicate software problems. Contact AT&T Technical Support.

No target machine specified

This error is associated with *iftp* when a file transfer is invoked without specifying a remote machine. Do an *open* before invoking the transfer command.

Not connected

You attempted to use an *ftp* command requiring a connection before you established the connection. Do a *connect* before invoking the transfer command.

Unknown host

The hostname given is not in the */etc/hosts* file. Verify the hostname and try again.

User already logged in

A user who is already logged into a remote host attempted to login again.

We only support non-print format, sorry

Currently *fip* only supports the "non-print" format.

We only support file structure, sorry

Currently *fip* only supports the "file" structure.

unknown mode

Attempted to set a mode that is not supported. Currently, the supported modes are binary, ASCII and tenex.

?Ambiguous command

Invoked an abbreviated *fip* command that was not unique. Be sure to specify enough of the command to make it unique.

?Ambiguous help command *command*

The *command* specified for the help command was not unique. Be sure to specify enough to make it unique.

?Invalid command

Invoked an unknown command. For a complete list of valid commands, type *help*.

?Invalid help command *command*

Invoked the help command with an invalid *command* name. For a complete list of valid commands, type *help*.

TELNET ERRORS**bad port number**

A zero or negative port number specified by the user with the port option on the *open* command.

Connection closed by foreign host

A connection was forcibly closed by a peer. This normally results from the peer executing a *shutdown* call,

although it may also indicate that the remote host has crashed.

no open connection to negotiate options

You must be connected to a remote host before you can negotiate options.

telnet: tcp/telnet: unknown service

The */etc/services* file does not contain an entry for the *telnet* service. An entry for this service should be created in this file. See *services(5)* in the **UNIX PC Enhanced WIN/3B Programmer's Reference Manual**.

unknown host

The hostname given is not in the */etc/hosts* table or a hostname was specified incorrectly.

?Already connected to *hostname*

You attempted to establish a connection with a remote host to which you are already connected.

?Ambiguous command

You invoked an abbreviated *telnet* command that was not unique. Be sure to specify enough of the command to make it unique.

?Ambiguous help command *command*

The *command* specified for the help command was not unique. Be sure to specify enough to make it unique.

?Invalid command

Invoked an unknown command. For a complete list of valid commands, type *help*.

?Invalid help command *command*

Invoked the help command with an invalid *command* name. For a complete list of valid commands, type *help*.

REMOTE COMMAND ERRORS**Bad .rhosts ownership**

~/.rhosts must be owned by the current user. Its ownership is determined to be the owner of the home directory in which the file resides.

Host name for your address unknown

The name of your host is not in */etc/hosts* on the remote host.

Login incorrect

An invalid username or password was specified. Try again.

Login timed out after *x* seconds

The user did not login within the required time, so the login procedure exited.

.netrc file not correct mode. Remove password or correct mode

The *.netrc* file should only be readable by the owner. No other protection mode is allowed due to the sensitivity of the information this file contains.

No directory!

There is no home directory specified in the */etc/passwd* file for the user logging into a remote host.

No shell

There is no shell specified in */etc/passwd* file for a user logging into a remote host.

rcp: who are you?

rcp attempted to ascertain the username of the user but failed.

rcp: invalid user name

There is no such user on the remote host.

rcp: lost connection

During a remote copy, the remote host closed the connection. Possibly the remote host crashed.

too many hosts

The *ruptime* command can print information on a limited number of hosts, (about 100). If the number of hosts on a network exceeds this amount, *ruptime* produces this error.

no hosts!?!

This error occurs when the *ruptime* command is invoked and there is no information on any hosts in the */usr/spool/rwho* directory. Most likely *rwhod* is not running.

rwho too many users

The *rwho* command can print information on a maximum of about 1000 users. If the number of users on a network exceeds this amount, the *rwho* produces this error.

.rhosts is a soft link

/.rhosts file cannot be a link.

Unknown .netrc option

/.netrc contains illegal information. See *netrc(5)* of the UNIX PC Enhanced WIN/3B Programmer's Reference Manual.

unknown service

The */etc/services* file does not contain an entry for this command. An entry for this service should be created in this file. See *services(5)* in the UNIX PC Enhanced WIN/3B Programmer's Reference Manual.

You have too many processes running

There are too many processes running on the remote host for one user.

ADMINISTRATIVE COMMAND ERRORS**arphypass must have SYSPRV**

You must be a superuser in order to run the *arphypass* command.

no interface for internet address

No interface has been defined yet for the specified Internet address. See *ifconfig(1M)* in the UNIX PC Enhanced WIN/3B Programmer's Reference Manual.

No room in ARP table, try later

Attempting an *arpbypass* command, the ARP table currently is full. Delete unwanted ARP entries with the *arpbypass(1M)* command.

Warning does not match utsnames

This message is produced when the you attempt to set the hostname to something other than the current UNIX system name.