

AlphaTCP®
User's Guide

DSM-00181-09

© 2008 Alpha Microsystems

REVISIONS INCORPORATED	
REVISION	DATE
00	April 1993
01	January 1994
02	February 1995
03	July 1995
04	September 1996
05	October 1997
06	June 1998
07	March 1999
08	September 1999
09	April 2008

AlphaTCP User's Guide
To re-order this document, request part number DSO-00181-00.
This document applies to AlphaTCP version 1.5B and later.

The information contained in this manual is believed to be accurate and reliable. However, no responsibility for the accuracy, completeness or use of this information is assumed by Alpha Microsystems.

The following are registered trademarks of Alpha Microsystems, Irvine, CA 92614:

AMOS	Alpha Micro	Alpha Micro	AlphaACCOUNTING
AlphaBASIC	AlphaCALC	AlphaFORTRAN 77	AlphaLAN
AlphaWRITE	*in*Sight/am	AlphaRJE	

The following are trademarks of Alpha Microsystems, Irvine, CA 92614:

AlphaBASIC PLUS	AlphaVUE	AMPC	AMTEC
AlphaDDE	AlphaConnect	DART	AlphaMAIL
AlphaNET	ESP	MULTI	

All other copyrights and trademarks are the property of their respective holders.

ALPHA MICROSYSTEMS
17534 Von Karman
Irvine, CA 92614

Table of Contents

PREFACE

Graphics Conventions iii

CHAPTER 1 - INTRODUCTION 1-1

WHAT IS ALPHATCP AND WHAT CAN IT DO FOR ME? 1-1
What Is TCP/IP? 1-1
What Is the Internet? 1-2
AlphaTCP FEATURES 1-2
Basic Concepts 1-2
Host Names and Addresses 1-3
Authorization/Permissions 1-4
A Note on Case 1-4
Protocols 1-4

CHAPTER 2 - TRANSFERRING FILES 2-1

FTP 2-1
Commands 2-2
Options 2-6
Sample 2-7
Transferring Random-Access Files 2-7
TFTP 2-8
Options 2-9
URL 2-9
Format of a *url* 2-9
Using *urls* With AMOS 2-10
Using the URL Command 2-10
CONVERTING UNIX FILE NAMES TO AMOS 2-11
Creating Unique File Names 2-13
CONVERTING AMOS FILE NAMES TO UNIX 2-13

CHAPTER 3 - THE VIRTUAL TERMINAL CONNECTION 3-1

USING TELNET 3-1
Ending a TELNET Session 3-2
TELNET Command Mode 3-2
Multiple TELNET Connections 3-4
REMOTE printing with MYLPR 3-4

CHAPTER 4 - INFORMATION COMMANDS 4-1

RUPTIM 4-1
The RUPTIM Display 4-1
PING 4-2
TELDMP 4-2
FINGER 4-3

CHAPTER 5 - UTILITIES 5-1

UUE 5-1
UUD 5-1
EMAIL 5-1
Blind Copy 5-2
Attachments 5-2

APPENDIX A -ERROR MESSAGES A-1

APPENDIX B -AMOS/UNIX COMPARISON B-1

FILE STRUCTURE B-1
File Names B-2
READ/WRITE SECURITY B-3
UNIX B-3
AMOS B-3

GLOSSARY 1

DOCUMENT HISTORY 1

INDEX

PREFACE

This guide describes the commands and utilities provided by AlphaTCP. These commands let you transfer files between hosts and carry out tasks on other computers. You can also display information about who is using the network and which hosts you can communicate with. If you are familiar with the variants of TCP/IP found on UNIX computers, you will feel quite at home with AlphaTCP since it originated on those computers and the user interfaces are very similar. The *AlphaTCP User's Guide* consists of:

- Chapter 1 defines what AlphaTCP is and what it can do for you. It also contains some background information.
- Chapter 2 describes the file transfer commands, FTP, TFTP, and URL.
- Chapter 3 discusses TELNET, which creates a virtual terminal connection. It also covers MYLPR, which you can use in some situations to print reports from a remote AMOS computer to your local printer.
- Chapter 4 describes the information AlphaTCP commands RUPTIM, PING, FINGER and TELDMP.
- Chapter 5 describes the AlphaTCP utilities UUE, UUD, and EMAIL.
- Appendix A lists AlphaTCP error messages and gives solutions to some common difficulties.
- Appendix B discusses and compares some of the basic concepts and features of the AMOS and UNIX operating systems.
- The glossary defines some terms relating to AlphaTCP which you may not be familiar with.

If you are the person in charge of AlphaTCP at your installation, you should also refer to the *AlphaTCP Administrator's Guide*. That manual describes how to install and configure AlphaTCP. Please read the current *AlphaTCP Release Notes* for late-breaking information on this product.

GRAPHICS CONVENTIONS

This manual conforms to other Alpha Micro publications in its use of a standard set of graphics conventions. We hope these conventions simplify our examples and make them easier for you to use. Unless stated otherwise, all sample commands are assumed to be entered at AMOS command level.

SYMBOL	MEANING
filespec	An AMOS file specification that identifies a specific file within an account. A complete filespec for the local computer is made up of the device name, the file name, the file extension, and the account number. For example: DSK0:SYSTEM.INI[1,4] A file specification may also consist of an ersatz name, which specifies a particular disk account, and a file name, like this: TCP:NETLOG.LST. A file specification for a remote computer includes the host name of the remote computer. The exact format depends on the network protocol. Appendix B describes AMOS and UNIX file structures.
{ }	Optional parts of a command appear in braces. You can enter exactly what is in the braces or substitute the correct value if it's a parameter. Do not include the braces. For example: **`DIR{/option}`** RETURN indicates ***/option*** is not required.
TEXT	This **`bold`** typeface represents characters you type. Since many AlphaTCP commands are case sensitive, the text may be either upper or lower case. Variable parts of the entry are in italics, as noted below.
Text	We use this ***`bold italic`*** type for variable parts of command examples. Replace the text shown with the appropriate entry.
TEXT	Text in this `typeface` is used for characters the computer displays on your screen, command syntax, program examples, and sample host names. In command syntax, variable parts of the command line are in italics, as noted below.
Text	This *`italic`* typeface indicates a variable part of a command line or program syntax.
Parameter	Following a command line or program format sample, the first reference to each parameter in the sample is in this ***bold italic*** text. Subsequent references are not.
COMMAND	AMOS command names and file names are given in all capitals. Most AMOS commands are not case sensitive. However, some AlphaTCP commands on AMOS, such as TELNET, must be entered in lower case.
command	UNIX commands, directory paths, and file names are given in **bold** text. All UNIX commands and file names are case sensitive, so they are given in the proper case (normally lower case). Since they are based on corresponding UNIX commands, the commands you use while "inside" AlphaTCP, such as the various FTP commands, are given in lower case bold.
RETURN	The key symbol indicates a reference to a key on your keyboard. The name of the key appears inside the key symbol.
CTRL-C	This indicates a control sequence you press on the keyboard. Press CTRL and hold it down while you press the indicated key.
^	When displayed in front of a capital letter, this means the letter is a control

SYMBOL	MEANING
	character. For example, when you press CTRL-C, it appears on your screen as ^C (^C is the control character that cancels most programs and returns you to AMOS command level).
	This Halt! symbol indicates an important note you should read carefully before going further in the documentation. Usually, text next to this symbol contains instructions for something you *must* or *must not* do, so read it carefully.
	This Hint symbol indicates a helpful bit of information, or a "short cut" that could save you time or trouble.
	This Remember symbol indicates something you should keep in mind while following a set of instructions.

Chapter 1 - Introduction

WHAT IS ALPHATCP AND WHAT CAN IT DO FOR ME?

What's the weather like in Paris? What's the exchange rate for the U.S. dollar in Tokyo? When's the next space shuttle flight at Kennedy Space Center? What's the current class schedule at Harvard?

Imagine being able to talk to and share information with a million different computers in over forty countries to ask questions like these—and doing it from your own terminal! Using AlphaTCP gives an Alpha Micro computer access to huge networks that span the globe.

AlphaTCP provides true, multi-user support to let you transfer files and mail messages between dissimilar computers and networks, and log onto a remote computer as a virtual terminal. AlphaTCP allows direct communication between networks that use different media (phone line, twisted pair wire cable, and coaxial cable) and different lower-level protocols (Ethernet® or serial link). It also allows communication between computers from different manufacturers (IBM, DEC, SUN, Apple, and Alpha Micro, to name a few) that use different operating systems such as AMOS®, UNIX®, Microsoft® Windows and MS®-DOS.

AlphaTCP is Alpha Micro's implementation of the TCP/IP protocol suite. TCP/IP is an open, non-proprietary standard—the most widely used non-proprietary standard in the U.S. One advantage of using TCP/IP is that many off-the-shelf TCP/IP hardware networking products are available, such as bridges and routers, that allow you to bridge your AlphaTCP network to other types of networks such as Novell® and token ring. AlphaTCP is your ticket to the open standards world of open communication.

What Is TCP/IP?

One of the most difficult problems system integrators and managers face is connecting different types of computers into a network and getting existing networks to talk to each other.

In an ideal world, each computer would talk to all others, no matter what operating system or processor was inside. In the real world, it seems as if every type of computer talks a little bit different language. The bright spot in this otherwise dismal picture is the TCP/IP protocol, the common denominator that bridges the communication gap among a wide variety of minicomputers, mainframes, and PCs.

TCP/IP (Transmission Control Protocol/Internet Protocol) was developed by the U.S. Department of Defense as a set of rules spelling out how dissimilar computers can exchange data. (That's what a protocol is—rules on how to transmit and receive data.) Because it is used on such a wide variety of networks and computers, in the U.S. TCP/IP has become the standard protocol for linking computers and networks that normally could not talk to each other.

The TCP layer of the protocol provides reliable service to applications; IP is the layer that handles transfer of information between networks.

Because much of the research and development of TCP/IP was done on UNIX computers, TCP/IP has become known as the ticket to communicating with UNIX computers. However, many other types of computers, such as those using DOS, MVS or VMS, also use TCP/IP.

What Is the Internet?

Any network that consists of multiple networks is called an "internet." However, Internet with a capital "I" refers specifically to a huge, world-wide collection of networks that use TCP/IP.

The Internet consists of almost a million computers, all using TCP/IP. The computers that use Internet reside in universities, military installations, research institutions, and every day sorts of places like Alpha Micro.

Thousands of databases and bulletin boards exist on the Internet. Special interest groups exchange information on topics ranging from the latest music group, to vegetarian recipes, to world economic policy. You can pick up shareware, college class schedules, and reviews on the newest restaurants in Boston. Using AlphaTCP and an electronic mail package on your Alpha Micro computer, you can send a mail message to someone across the world.

Getting onto the Internet is as simple as connecting to a network that has access to it. The hard part is finding out which of the thousands of information servers you are interested in. Entire books are available containing information on what Internet services are available and how to get to them. The Internet has several information servers used just for searching for other information servers. For example, Internet programs **archie**, **gopher**, and WAIS allow you to catalog and explore the vast archives of information available on the Internet. WWW (World Wide Web) provides access to the vast wealth of the Internet using hypertext (much like using Windows on-line help).

ALPHATCP FEATURES

AlphaTCP gives you four basic network capabilities: remote login (also called a virtual terminal connection), which lets you log on to another host; file transfer connections (you can choose from three types of file transfer); electronic mail; and network printer access.

Other commands let you display information relevant to these operations. For example, you can see which hosts are available and how long they have been up.



This manual discusses AlphaTCP as a way to connect AMOS and UNIX computers. While TCP/IP is widely used on UNIX computers, AlphaTCP lets you connect an AMOS computer to any computer which supports TCP/IP, regardless of its operating system.

BASIC CONCEPTS

This section is a general introduction to some features of networking you should understand. For example, it is helpful to be clear about the use of host names, so you know which users are on the same computer. On the other hand, to use the networking facilities you don't need to know how the protocols work.



If you aren't familiar with either AMOS or UNIX, some of the basic concepts of these two operating systems are discussed in Appendix B.

Host Names and Addresses

The computers joined by the physical link are usually known as hosts. Each host has a unique number associated with it; which, together with the number of the network, makes up its address. Data on the network carries either the address of a specific host or a broadcast address which sends it to all the hosts on the network. A large number of hosts may be supported on one network. For example, the limit for an Ethernet network is 1024 stations.



For those of you familiar with the seven-layer model for networking software, this guide deals with the “Presentation” and “Application” layers.

When data is transported across a network, it is sent in packets. These packets are labeled with control information, such as the address of the host which sent the packet.

Ethernet addresses are 48-bit hexadecimal numbers, and are unique for each computer in the world. Apart from networks which send data out through Ethernet boards, different types of networks may be available, such as SLIP, X.25, or Token Ring. To cope with these networks, which may have many duplicate addresses worldwide, TCP/IP uses two other ways to identify a device on a network: by its internet address or by its host name.

An internet address consists of four decimal integers, separated by three periods. For example, `27.3.0.129` could be an internet address. The internet address finds a device in relation to the internet, which is the theoretical network consisting of all TCP/IP networks. Each computer has a unique internet address (computers which are attached to more than one network have different internet addresses for each connection). The AlphaTCP command ARP shows you how internet addresses are related to Ethernet addresses on your own network.

Host names are the easiest way to specify a device. For example, the computer with the internet address `27.3.0.129` might have the host name `admin`. AlphaTCP host names are local to your own host computer, and are assigned by the Network Administrator. All AlphaTCP commands understand host names. The relationship between the two kinds of addresses and host names is illustrated in this diagram:





On very large Internet installations, host names are translated by many “name server” computers. Various parts of the name are assigned by different Internet authorities.

Authorization/Permissions

In general, you can only access files you have permission to access. On UNIX, for example, you can set a file's permissions to “owner,” “group,” or “other.” In addition, a distinction is made between permission to read a file and permission to write it. Having write permission on a file lets you change its contents. A typical document file, for example, might have write permission for the owner and read permission for others—anyone is allowed to read or copy the file, but only the owner may alter it.

To access an AMOS file across the network, the file must be on a disk device which allows network access. If the device is set to “noaccess,” no network facilities can read from or write to it.

Most AlphaTCP commands require you to have at least read permission on a file you access. In addition, AlphaTCP commands need to be able to find your local computer and user name, but the Network Administrator usually sees to this.

See Appendix B for more information about file security under UNIX and AMOS.

A Note on Case

In general, most AMOS commands and file names are not case sensitive—you can enter them using either upper or lower case. However, since the commands in AlphaTCP are based on commands developed for UNIX, they *are* case sensitive, as are all commands and file names you enter while connected to a remote UNIX computer. Be careful to use the correct case while working with AlphaTCP!

Also, see Chapter 2 for a discussion of case translation when you’re copying files between UNIX and AMOS.

Protocols

AlphaTCP is based on the TCP/IP family of protocols. Within the software, network protocols provide rules for the structure of the packets transported across the network, including header and data format. Addresses and control information are included. The suite of TCP/IP protocols lets user programs establish, use, and end sequenced, error-free connections. A brief overview of these protocols is given in the *AlphaTCP Administrator’s Guide*. This is background information—you don’t need to know it to use AlphaTCP.

Chapter 2 - Transferring Files

This chapter describes the three ways you can transfer files between your computer and a remote host using AlphaTCP. The three file transfer commands are:

- FTP, the file transfer command. It includes several other options in addition to simple file transfers.
- TFTP, the trivial file transfer command. TFTP is easier to use than FTP, but it is less powerful and offers fewer options.
- URL. This command lets you retrieve a file from a remote computer by specifying its URL. You cannot use it to transfer files from your computer to a remote computer.

After the sections on the commands, this chapter describes how file names are translated when you copy files between UNIX and AMOS.



If you want to look at a file on another computer, the best solution may be to log in to the remote computer. See Chapter 3.

FTP

The FTP file transfer program lets you transfer a file or a number of files to or from a remote host. It is the user interface to the internet standard File Transfer Protocol. While you are using FTP, you can enter a variety of commands and options.



The Network Administrator might have to arrange for you to have a user name on the other computer before you can use FTP.

The format for FTP is:

```
FTP {options} {host}
```

The available ***options*** are discussed below, after the description of FTP command mode.

host is the host name or internet address of the computer you want to connect to. If you omit host, you enter FTP command mode without connecting to another computer, and can then use the FTP command **open** to make a connection.

If you included host, and the FTP connection is successful, you are prompted for a user name and password.

Once the connection is made, or if you did not specify a host to connect to, you see the FTP prompt:

```
ftp>
```

At this prompt, you can enter any of the commands described below. You can also perform the functions of some of these commands by including options on the FTP command line. The options are discussed following the commands.

Commands

All of the FTP commands follow these general rules:

- They must all be entered in lower case, as shown.
- If you omit the parameters for a command, FTP prompts you for them.
- You can abbreviate any command to its shortest unique form.



All file transfers use the current settings for file type. Be sure to set the file type correctly before transferring files! The default file type is ASCII. To transfer programs or data files, set the file type to binary.

- You can enter full file and path specifications for any file name. The default is the current working directory (on UNIX) or the account you are logged into (AMOS).
- If you are copying files between UNIX and AMOS and don't specify a new file name, the file name is converted as necessary, following the rules given at the end of this chapter.
- If any command argument has embedded spaces, you must enclose the argument in quotes. For example: **`! "dir /w"`** RETURN.
- If you use **-** as a file name, FTP uses **stdin** (usually the keyboard) for reading or **stdout** (usually the screen) for writing.



FTP follows all read and write restrictions of the operating system. On UNIX, you can only read and write files you are authorized for; on AMOS, you cannot write to other accounts unless you are logged to OPR:, and any device you want to access on the remote host must not have noaccess set. Appendix B discusses file security on UNIX and AMOS.

The FTP commands:

`! {command}`

Runs the specified ***command*** on the local computer.



You cannot use **!** to run any command (such as MONGEN) which loads a module in user memory and leaves it there when finished.

`append` *`local-file`* `{`*`remote-file`*`}`

Appends a local file to a file on the remote computer. If you don't specify ***remote-file***, the file is appended to the remote file of the same name.

`ascii`

Sets ASCII transfer mode. This performs end-of-line and end-of-file conversions between systems.

`bell`

Toggles bell mode. When bell mode is on, FTP rings a bell after each file transfer command is done. By default, the bell is off.

`binary`

Sets image transfer mode. Files are transferred without any conversions. If you are transferring an AMOS contiguous file between two AMOS hosts, the file's type is preserved.

`bye`

Ends the FTP session with the remote server and exits.

`cd` *`remote-dir`*

Changes the working directory on the remote computer to ***remote-dir***. For an AMOS remote host, you can specify an ersatz device name or a disk account in *`devn:[#,#]`* format.

`close`

Ends the FTP session with the remote server and returns to the command interpreter.

`delete` *`remote-file`*

Deletes the ***remote-file*** on the remote computer.

`debug`

Toggles debugging mode. When debugging is on, each command sent to the remote computer is preceded by the string `--->`. By default, debugging is off.

`dir {`*`remote-dir`*`} {`*`local-file`*`}`

Lists the directory of ***remote-dir*** and puts the output in ***local-file***. If you omit remote-dir, the current working directory on the remote host is used. If you omit local-file, output is displayed on the terminal.

`get` *`remote-file`* `{`*`local-file`*`}`

Copies ***remote-file*** to the local computer. If you don't include ***local-file***, the file is given the same name it has on the remote computer. If the remote file name is not legal for the local computer, the file name is converted as necessary. See the end of this chapter for details.

`hash`

Toggles whether a hash-sign, #, is printed with each data block transferred. When you turn hash on, it displays the size of a data block. By default, hash mark printing is off.

`help {`*`command`*`}`

Prints an informative message about the meaning of ***command.***. If you don't include a command, FTP lists all known commands.

`lcd {`*`directory`*`}`

Changes the working directory on the local computer. If you don't specify a ***directory***, the home directory is used. On an AMOS computer, you must include directory, which can be a user name, an ersatz device, or an account in the format *`devn:[#,#]`*

`literal` *`arg1 arg2 ...`*

The specified arguments (***arg1***, etc.) are sent, verbatim, to the remote FTP server. A single FTP reply code is expected in return. You can use the **remotehelp** command to see the commands supported on the remote computer.

`ls {`*`remote-dir`*`} {`*`local-file`*`}`

Prints an abbreviated list of a directory from the remote computer. If you don't include ***remote-dir***, the current working directory is used. If you omit ***local-file***, the output displays on the terminal. Also, you may use the standard switches available for the UNIX **ls** command.

If you want to use both switches and a remote-dir, you must enclose them in quotes so they are both passed as arguments to the remote computer. For example: `ls "-la subdir1"`. If you don't do this, remote-dir is interpreted as local-file.

`mdelete` *`remote-files`*

Deletes the specified ***remote-files*** on the remote computer. If prompting is on, you must respond to a prompt before each file is deleted.

`mget` *`remote-files`*

Copies ***remote-files*** from the remote computer to the current local directory. If prompting is on, you must respond to a prompt before each file is transferred. File names are converted as necessary for the local operating system, as described at the end of this chapter.

When you use **mget**, FTP creates a file called FTP.LOG in your current account. This file contains all the file name conversions performed. You can look at this file to see what the transferred files are called on the local computer. If you do not want this file created, or you have trouble running out of memory transferring large numbers of files, you may disable this feature using the **-l** switch.

`mkdir directory-name`

Creates a directory named ***directory-name*** on the remote computer. If the remote host is an AMOS computer, this command has no effect.

`mput` *`local-files`*

Transfers ***local-files*** from the current local directory to the current working directory on the remote computer. If prompting is on, you must respond to a prompt before each file is transferred. File names are converted as necessary for the remote operating system, as described at the end of this chapter.

`open host {`*`port-number`*`}`

Connects you to the specified ***host*** FTP server. If you include ***port-number***, FTP tries to contact an FTP server at that port. If the auto-login option is on (this is the default), FTP also tries to log you on to the FTP server (see the **user** command, below).

`prompt`

Toggles interactive prompting. Interactive prompting lets you selectively send or retrieve files during multiple file transfers. If prompting is off, any **mget** or **mput** transfers all files. By default, prompting is on.

`put` *`local-file`* {*`remote-file`*}

Copies ***local-file*** to the remote computer. If you don't include ***remote-file***, the local file name is used in naming the remote file. See the end of this chapter for how file names are converted between UNIX and AMOS.

`pwd`

Displays the name of the current working directory on the remote computer.



Some remote servers may not support **pwd**. If not, use the **xpwd** command.

`quit`

A synonym for **bye**.

`recv` *`remote-file`* {*`local-file`*}

A synonym for **get**.

`remotehelp` {*`command`*}

Requests a help display about ***command*** from the remote FTP server. If you don't include a command, it displays the commands supported through FTP on the remote computer.

`rename` *`remote-file new-name`*

Renames the ***remote-file*** to the ***new-name***.

`rmdir` *`directory`*

Deletes the specified ***directory*** on the remote host. If the remote host is an AMOS computer, this command has no effect.

`runique`

Specifies files received will have unique names. See the section "Creating Unique File Names," later in this chapter, for how the unique names are created on AMOS computers.

`send` *`local-file`* {*`remote-file`*}

A synonym for **put**.

`status`

Displays the current status of FTP. This example shows the default settings:

```
Connected to Admin.
Type: ascii; Verbose: on; Bell:off; Prompting: on;
Debugging: off; Hash mark printing: off.
Store unique: off; Receive unique: off.
```

`sunique`

Specifies files sent will have unique names. Not all remote hosts support this feature. If it isn't supported, you will see an error message when you invoke FTP with **sunique** on. In this case, turn this option off and be sure to specify a destination file name when sending files to the remote host.

`type type-name`

Sets file transfer type to ***type-name***, which may be ASCII, binary, tenex, or image. If you omit the type-name, it displays the current type. The default is ASCII.

When transferring ASCII files, it is important to have the type set to ASCII so carriage return/line feeds are handled correctly. The binary and image types are identical, and merely transfer the raw data between computers. The tenex type is for transferring files to or from a host which uses the Tenex operating system.

`user` *`user-name`* {*`password`*} {*`account`*}

Identifies you to the remote FTP server as ***user-name***. If you don't include ***password*** and the server needs it, FTP prompts you for it (after disabling local echo). If you don't include ***account***, and the FTP server needs it, it prompts for it.

When the auto-login option is on (the default condition), FTP prompts you for a user name when you make the remote connection, and you don't have to use this command. You need it only when you've turned auto-login off by using the -n option (described below).



You can sometimes use **user** to change user names in the middle of a session. However, not all hosts allow this. If you try to change user names and it isn't allowed, you may receive an error message, the FTP connection may be closed, or the FTP connection may hang.

`verbose`

Toggles verbose mode. In verbose mode, all responses from the FTP server are displayed. In addition, when a file transfer completes, statistics about the efficiency of the transfer are displayed. By default, verbose is on for an interactive session and off otherwise.

`?` {*`command`*}

A synonym for **help**.

Options

You can include the following options on the FTP command line. There are also FTP commands which let you set all of these options.

`-b` Forces the session to begin in binary mode.

`-d` Turns on debugging.

`-h` Turns on hash mark printing.

`-i` Turns off interactive prompting during multiple file transfers.

`-l` Turns off the local transfer log file.

`-n` Suppresses auto-login on initial connection.

`-u` Enables both sunique and runique.

`-v` Turns off verbose mode.

Sample

For example, a typical FTP session may look like this:

```
ftp yoursys RETURN
Trying [192.0.2.1]...
220 yoursys FTP server ready.
User (yoursys:): myname RETURN
331 Password required for myname.
Password: ****** RETURN
230 User myname logged in.
ftp> type image RETURN
200 type set to I
ftp> get oldunix olduni.xxx RETURN
file : oldunix received as olduni.xxx
200 PORT command okay.
150 Opening data connection for oldunix (192.45.18.1,1060) (1050247 bytes)
226 Transfer complete
1050247 bytes received in 31 seconds (33.08 Kbytes/sec)
ftp> quit RETURN
221 Goodbye
```



If your computer is using the Domain Name Service (DNS) protocol, FTP will try all addresses returned for the specified name until it finds one that responds.

Transferring Random-Access Files

On AMOS, files may either be arranged on the disk so that all blocks are next to one another (these are called contiguous or random-access files), or may be arranged so that blocks are scattered on the disk but tied together by “links” (sequential-access files).

FTP may be used to transfer contiguous files. How they are handled depends on each side of the connection. The contiguous status of the file will only be preserved if transferring between two AMOS computers, both of which are using AlphaTCP version 1.2 or later. If transferring from AMOS to a non-AlphaTCP computer, the file will be translated into whatever standard, native format exists on the destination computer. If transferring between two AMOS AlphaTCP hosts, if one of the computers is using a version of AlphaTCP earlier than 1.2, the contiguous file will be converted to a sequential-access file.

TFTP

The TFTP command is the user interface to the internet standard Trivial File Transfer Protocol. It lets you transfer one file at a time between hosts without providing a password. It doesn't verify you are a valid user, and only transfers files readable by anyone to accounts or directories writeable by anyone. Its syntax is:

```
tftp {-i}{-s} host put local-file {remote-file}
```

or:

```
tftp {-i}{-s} host get remote-file {local-file}
```

host is the name or internet address of the host you want to copy from or to. If you are copying from the local computer to the remote host, specify ***put***. Specify ***get*** to copy a file from the remote host to your local computer. ***local-file*** and ***remote-file*** are the file specifications on the local computer and remote host, respectively.



For security reasons, many computers with TCP/IP choose not to enable TFTP, so it may not be available for all remote hosts. See the *AlphaTCP Administrator's Guide* for a discussion of the security concerns of TFTP.

UNIX files you are copying to or from must be writeable (or readable) by anyone in order to be used with TFTP. See Appendix B for a discussion of security features.

Here is an example:

`tftp admin get /users/mine/sales.txt sell.it` RETURN

This copies the file **/users/mine/sales.txt** stored on the host `admin` to the file SELL.IT in the current account on the local computer (assuming the local computer is running AMOS). After a successful transfer, TFTP displays how much data was transferred.

If you don't specify a new file name, the file is copied with its original name. An AMOS device name or ersatz name can be specified on UNIX computers as a path prefix and the server on AMOS translates it to a device or ersatz reference. UNIX does not understand a device name if you end it with a colon.

For example, if you are on a UNIX computer and want to transfer the file **x.c** from the current directory to the SYS: account on an AMOS host called `admin`, the command to use is:

`tftp admin put x.c sys/x.c` RETURN



Generally, you should specify the full path names of the files you are transferring. If the file will be created on AMOS and you don't specify a name, the name will be truncated to fit AMOS conventions (see the next section for the conversion algorithm).

You can specify a dash (-) instead of a local file name when using get. This copies to your standard output (which usually means it prints on your terminal) instead of to a file.

In the default ASCII mode, any necessary character conversions to and from the network standard character set are done. This mode should be used when transferring text files.

Options

If the file being transferred contains binary data, such as compiled program code, include the ***-i*** option.

The ***-s*** option suppresses screen output. This is useful when the command is to be run in a command file.

URL

The URL command lets you retrieve a file from a remote system by specifying its *Universal Resource Locator*, or *url*. The *url* is the familiar notation used to specify World Wide Web pages, such as *www.alphamicro.com/oldocs*.



To avoid confusion, when talking about the URL command, we'll use all capital letters; when talking about the Universal Resource Locator of a file on the Internet, we'll use lower-case italics, *url*.

*url*s can refer not only to web pages, but also to text or binary files at an FTP site. AlphaTCP's URL feature can retrieve any of these file types, using either HTTP (HyperText Transfer Protocol) or FTP protocol as appropriate. When using FTP, URL supports wildcards, and can transfer contiguous files from another AMOS computer.



To use URL, the TAMED server must be running. See the *AlphaTCP Administrator's Guide* for information about starting TAMED.

In addition to the URL command, AlphaTCP also provides an AlphaBASIC subroutine, URL.SBR, so you can retrieve files from your AlphaBASIC or AlphaBASIC PLUS programs. See the *AlphaBASIC XCALL Subroutine User's Manual*, Rev. 04 or later, for more information.

Format of a *url*

A *url* consists of three parts separated by slashes: a *method*, *system*, and *path*:

- Though *method* could be many different things, URL supports either *http://* or *ftp://*, with *http://* being the default if no method is specified.
- The *system* portion may be either a host name or IP address.
- *Path* is interpreted by the remote system. It often specifies a subdirectory and a filename, though it might invoke a program or perform a search instead.

Here are two example *urls*, one using the FTP protocol and the other using HTTP:

ftp://ds.internic.net/rfc/rfc1123.txt
http://smtpgw.alphamicro.com/anon/files.txt

Using *urls* With AMOS

The AlphaTCP web server and FTP server handle the path portion of a *url* in the same way. It is scanned for a filename, and either an ersatz name or device and account. If no ersatz, or device and account, is specified, the file is assumed to exist in the default login directory. With HTTP, a question mark indicates a query, and causes the server to scan the web applications list. When using FTP, question marks and asterisks are used as wildcards in a file name.

The first example below uses HTTP to reference the default page (home.htm) in the ersatz WEBPPN: on the local system; the second one refers to a specific binary file:

http://localhost/webppn/
http://localhost/webppn/demo.lit

The following *urls* each refer to all the system commands on another system with the FTP protocol. The first uses the SYS: ersatz name, the second contains the DSK0:[1,4] disk account:

ftp://backup.widget.com/sys/*.lit
ftp://backup.widget.com/dsk0/[1,4]/*.lit



URL follows the same security rules as other AlphaTCP features. These security rules are described in the *AlphaTCP Administrator's Guide*.

Using the URL Command

The URL command retrieves one or more files from a remote computer. Its syntax is:

```
URL {flags} {outputfile=}{proto://}hostname{:port}/path/filename
```

flags is optional and consists of one or more of the following:

/A	Transfer files in ASCII mode (FTP only)
/D	Output debug information to the screen
/K	Kill remote files after transfer (FTP only)
/L:*name*	Log in using login *name* (FTP only)
/N	Nodelete, don't delete existing local files
/P:*pass*	Log in using password *pass* (FTP only)
/Q	Query for each file
/S	Suppress screen output
/T	Enable twinkle progress indicator
/U	Generate unique local filenames

Case does not matter for the flags, but might be important for ***name*** and ***pass***.

Outputfile lets you save a remote file to a specific name. Wildcards are not supported for the output file. You can send the file to the screen by using TRM: as the outputfile, or to a printer by using TRM0:pname or TLP0:pname.

Proto:// is optional and may be either *http://* or *ftp://*. The default is *http://*.

Some systems run their web or FTP servers on non-standard port numbers. The optional ***:port*** specification is the standard method used in a *url* to override the port number.

Assuming the login "orders" exists with the password "opwd" and the proper privileges, this example retrieves all .ORD files from the login account on the remote system "sales.widget.com." Each file transferred is deleted on the remote host.

```
URL /k /l:orders /p:opwd ftp://sales.widget.com/*.ord RETURN
```

Since the files are in this login's default account, the login does not need to be able to change directories.

You could use this example to retrieve all of the terminal drivers from another AMOS system:

```
URL /l:admin /p:admpwd ftp://template.widget.com/dvr/*.tdv RETURN
```

This example retrieves a particular file from a well-known World Wide Web page:

```
URL support.microsoft.com/support/kb/articles/q677.htm RETURN
```

This example shows an HTTP query *url* (query submitted using GET method):

```
URL smtpgw.alphamicro.com/basic-demo/optional.file?userquery RETURN
```

This example shows an HTTP query *url* (query submitted using POST method) :

```
URL smtpgw.alphamicro.com/basic-demo/optional.file??userquery RETURN
```

CONVERTING UNIX FILE NAMES TO AMOS

When transferring UNIX files to AMOS, file name conversion may be a problem. The maximum length of a UNIX file name varies, but is always at least 14 characters in almost any combination; AMOS names are less flexible. A full AMOS file specification contains four parts:

- An optional four character logical device name (the last character must be a number) or up to six character ersatz device name. This part, if it exists, is followed by a colon.
- The file name of one to six letters and/or numbers, followed by a period.
- An optional extension of one to three letters and/or numbers.
- An optional account number within brackets. An account number consists of two octal numbers between 0 and 377, separated by a comma.

Also, AMOS file specifications are not case-sensitive.

Here is an example of a complete AMOS file specification: DSK0:SYSTAT.LIT[1,4].

AlphaTCP uses the following procedure to convert a UNIX file specification to an AMOS one. It is complex, but is designed to create the most meaningful AMOS file name possible.



This algorithm is used only when you do not specify a destination file name when transferring a file from UNIX to AMOS. It is also used by a number of AlphaTCP servers The program "FNU2A" described in the Administration Guide, performs the conversions on the screen for you.

The steps below outline the general procedure, and describe the conversion of an example UNIX file specification: `unix/dbase_1_old.index`.

1. All slashes are converted to colons. In our example, `unix:dbase_1_old.index`.
2. Everything through the last colon is stripped and used as the AMOS device specification. Our example is now the ersatz device name `unix:` and the file name `dbase_1_old.index`. When writing a file in FTP, the ersatz device name is discarded. Note that if there is no slash in the original UNIX specification there is no device name in the AMOS specification.
3. If there is a left bracket (`[`), it and everything after it up to a right bracket (`]`), if any, is stripped and used as an account number. When writing a file in FTP, this part is discarded.
4. If there are one or more periods in the remaining name, the first three letters and/or numbers after the last period are used as the file extension. Any further characters are discarded. The file name part of our example is now `dbase_1_old` with the extension `.ind`.
5. Any non-alphanumeric characters in the remaining name are used to fragment the name into up to six parts. Our sample is now `dbase 1 old`.
6. The AMOS file name can be only six characters long. Starting from the right fragment, one character at a time is taken from each part until six characters have been taken. In our example, three characters are taken from `old`, one from `1`, and two come from `dbase`.
7. The characters from each fragment are taken from left to right to create the file name. Our example yields the name `db1old`.

The complete AMOS file specification from the example is UNIX:DB1OLD.IND. (Remember, AMOS file names are not case sensitive.) If you are writing this file with FTP, the UNIX: is ignored and the file DB1OLD.IND is written to the current disk account.



In FTP, if the device and account are omitted, the default is the account you are currently logged into. With TFTP, you are not logged to any account, so you must always give a destination account.

Here are some more samples of file name conversion. These examples do not include device names or account numbers:

UNIX FILE NAME	AMOS FILE NAME
big_filename	BIGFIL.
Big_file_name	BIFINA.
File_one.exten	FILONE.EXT
file.a_b_c_d_e	FILE.ABC

Creating Unique File Names

When copying a file to AMOS using the FTP server, you can specify **sunique** on (or **runique** on if receiving a file from a remote computer). This causes the FTP server to look up the destination file name, and, if it already exists, modify the incoming file's extension to create a unique file specification so no file is overwritten.



These options affect FTP only, not TFTP.

This procedure is used to create a unique file extension:

1. The first blank character in the extension is changed to 0. If no blank character exists, go to step 2.
2. The last character in the extension is changed to the next higher character in this list:

   0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ

   If the last character is a Z, it is changed to a 0, and the second to last character is moved up one.
3. These steps are repeated as necessary until a unique file specification is created.

For example, the extension FF becomes FF0, 009 becomes 00A, and ACZ becomes AD0.

CONVERTING AMOS FILE NAMES TO UNIX

All AMOS file names are also legal file names under UNIX. However, AMOS file names are not case-sensitive, and UNIX names are. Since most UNIX file names are in lower case, AMOS file names are converted to lower case when you transfer them to UNIX without specifying a destination file name. For example, the AMOS file SAMPLE becomes the UNIX file **sample.**

Chapter 3 - The Virtual Terminal Connection

This chapter covers TELNET, the command you use to make a virtual terminal connection to a remote computer. TELNET lets you log on to another computer from your terminal. This is the only way you can use the remote computer's full screen editors, such as **vi** (UNIX) or AlphaXED (AMOS), or application programs. You specify your terminal type to the remote host, and you can use all the commands available to any user on the remote computer.

At the end of this chapter, there's a section on the MYLPR command, which you can use while you're connected via TELNET to an AMOS computer using AlphaTCP. MYLPR lets you print reports to your local printer from the AMOS computer you're connected to.

USING TELNET

The format for TELNET is:

```
telnet {host} {port}
```

Host is the host name or internet address of the remote computer you want to connect to. If you don't include a host, you are put into TELNET command mode (described below) without being connected to another computer.

Port is an optional port number to connect to. If you omit it, you are connected to the standard TELNET port for the remote host.

For example, this command begins a TELNET session and connects you to the remote computer with the host name `herts`:

```
telnet herts RETURN
```

When you make a TELNET connection, you see a message telling you which host you are connected to. For example:

```
Trying [192.0.2.1]...
Connected to hertfordshire [192.0.2.1]
Escape character is '^]'.
```

If you are connecting to an AMOS host with the server setup in its default mode, TELNET now prompts you for the terminal type and memory size you want. Here is an example:

```
*********  Welcome to AMOS *********

Enter terminal type and memory required (default 64k)
or a question mark (?) for a list of supported terminals
Example: >am62 100k
>? RETURN
AM62 AM62A TELTYP AM75
Example: >am62 100k
>am62a 200k RETURN
```

Several server options exist which may change this process. The login may have different memory defaults, enforce a valid range rather than a default, or welcome you with a different message. The server may even be set up with a pre-defined terminal type, or directly enter an application, bypassing the system login completely. If this is the case you may want to ask your system operator for help. This section assumes the server is set up for its default mode of operation.

Once you are connected, you see the remote computer's normal login prompt. You are now in TELNET input mode. In input mode, whatever you type is sent directly to the remote host. You can now log on and use the computer as if you were a local user. Some computers let you log off and log on again later, without breaking the TELNET session. Other computers (including AMOS hosts) are set up to close the TELNET session when you log off.

If TELNET automatically starts an application when you connect (as mentioned above), it will close the session as soon as you leave the application.

While you are in input mode, you can enter TELNET command mode by pressing the currently defined escape character. In the example above, this is ^] (CTRL-]).

Ending a TELNET Session

As mentioned above, some computers break the TELNET connection when you log off or leave the application. On others, you may end the TELNET session by entering TELNET command mode and using the **quit** or **bye** command.



The easiest way to disconnect from an AMOS system is to type "logoff."

TELNET Command Mode

If you use TELNET without arguments, you enter command mode. You can also enter command mode by pressing the TELNET escape character while you are in input mode. Command mode is indicated by the prompt: `telnet>`.

In command mode, you can use the commands listed below. You can use the shortest unique form of each command. The TELNET commands are:

`! command`

Runs the specified ***command*** on the local computer.



You cannot use **!** to run any command (such as MONGEN) which loads a module in user memory and leaves it there when finished.

`bye`

Synonym for **quit**.

`close`

Closes a TELNET session and returns to command mode (unless TELNET was executed with a host specified on the command line, in which case it exits).

`connect`

Synonym for **open**.

`crmod`

Toggles carriage return mode. When this mode is on, any carriage return characters received from the remote host are mapped into a carriage return and a line feed. This mode does not affect characters you type, only those received. This mode is not very useful, but is required for some hosts that ask the user to do local echoing.

`escape`

Prompts for and sets a new TELNET escape character. See the section on multiple TELNET connections, below.

`help {`*`command`*`}`

Displays helpful information about ***command***, or a help summary if command is not specified.

`open` *`host {port}`*

Opens a connection to ***host***. If you don't include a ***port*** number, TELNET tries to contact a TELNET server at the default port. The host specification may be either a host name or an internet address.

`options`

Toggles viewing of TELNET options processing. When option viewing is on, all TELNET option negotiations are displayed. Options sent by TELNET are displayed as `SENT` while options received from the TELNET server are displayed as `RCVD`.

`quit`

Closes any open TELNET session and exits TELNET.

`status`

Displays the current status of TELNET, including the host you are connected to and the state of debugging.

`?`

Synonym for **help**. You can leave command mode and return to input mode in your current session by pressing RETURN without entering a command at the `telnet>` prompt.

Multiple TELNET Connections

If you want, you can "chain" TELNET connections, connecting from host A to host B, then from host B to host C, and so on. When you do this, if you want to be able to break the connections one at a time, you must reset the TELNET escape character for all connections except the final one.

For example, you connect from host A to host B. Before connecting to host C, use the **escape** command to change the escape character from ^] to ^!. Then, connect to host C. Now, if you press the default escape character, ^], it breaks the connection from host B to host C and you return to host B; if, however, you press ^!, it breaks both connections and returns you directly from host C to host A.

You must redefine the escape key in this way, using the default character for the last connection you make. In our example, if you tried to change the escape character to ^! for the host B to host C connection instead of the A to B connection, either ^] or ^! would return you from host C directly to host A.

REMOTE PRINTING WITH MYLPR

As discussed in the *AlphaTCP Administrator's Guide*, AMOS computers using AlphaTCP can define a "dynamic" printer. Basically, this means there is a printer name defined to be used by AlphaTCP, but there is no actual printer with that name. Instead, anyone attaching to the AMOS computer via TELNET can use the MYLPR command to temporarily define his or her local printer using that name. That way, you can attach to an AMOS computer, print a report from AMOS to your local printer, and log off. Then someone else, anywhere in the world, can do the same thing, using the same printer name. MYLPR even works with printers connected via dynamic IP dialup accounts which have temporary addresses.

To assign an AMOS dynamic printer to your local printer, first make a TELNET connection to the AMOS host. Then, use the MYLPR command from AMOS command level. The format is:

```
MYLPR printer
```

Printer is the device name assigned to the printer when it was defined on the AMOS host.



You must use the *device* name from the printer or spooler initialization file, not the *printer* name. These are often, but not always, the same. You may need to look at the printer's initialization file to determine the device name, if the system operator of the remote computer hasn't given it to you. Setting up printers under AMOS is described in the *AMOS System Operator's Guide*.

Once you've assigned the dynamic printer name to your local printer, you can print to it as you would any other AMOS printer. When you're done printing, be sure to release the printer name so others can use it. To do so, add the /U option to MYLPR:

```
MYLPR printer/U
```

If you don't "unassign" the printer name using /U, it's possible someone else will print a report to the dynamic printer name, and it will print on your printer! Even worse, if you are connected via a dynamic IP dialup account someone else will be assigned your previous address when you hang up. In this case, it's possible reports could be printed to a total stranger!



MYLPR works only if the remote AMOS host has a dynamic printer name assigned. You may not be able to use MYLPR on all AMOS hosts. In addition, the system with the printer attached must be running an LPR server. For instance, if you are using a PC, standard Microsoft printing does not support LPR directly and requires additional software.

Chapter 4 - Information Commands

This chapter describes four of the information and network status commands included with AlphaTCP. The commands covered are:

- RUPTIM, which shows the current status of all hosts on the local network.
- PING, which lets you see whether one or more specific hosts are operational.
- TELDMP, which displays the source of incoming TELNET sessions.
- FINGER, which allows you to query remote host and user information.

Commands which give additional information about the setup and current state of the network are described in the *AlphaTCP Administrator's Guide*.

RUPTIM

RUPTIM tells you which hosts are available on the local network, whether they are up or down, and how long they have been in that state. For UNIX computers, you also see some load information. The format for RUPTIM is:

```
RUPTIM {-option}
```



On UNIX computers, the command name is **ruptime**

The available ***options*** are `-l` to sort the list by load average and `-t` to sort the list by uptime. If you include both options, only the last one is used.

The RUPTIM Display

Here is an example RUPTIM display:

```
herts   up 14+08:06    load 1.28, 0.50, 0.06
beds    up 10+16:53    load 0.51, 0.34, 1.16
lan     up 48+04:01
glos    up 24+18:54    load 1.06, 1.44, 0.50
fred    up 17+02:19
```

The first number gives the length of time the computer has been up, in days, hours and minutes.
The three numbers following `load` are the average number of jobs in the run queue over the last minute, last five minutes, and last fifteen minutes. This information may help you decide which host to use.



The UNIX **ruptime** command also shows the number of users logged into each computer. This user information is not shown on AMOS. Also, **ruptime** will not show user information for any AMOS hosts on the network. The number of users is not significant under AMOS in the same way it is on UNIX.

PING

PING checks whether a host is operational. It sends ICMP ECHO packets to the host, listens for ECHO REPLY packets, and prints the number of packets transmitted and received. Each received packet is validated against the transmitted message. Four ECHO packets containing 64 bytes of data (a periodic sequence of upper case letters) are transmitted. PING's syntax is:

```
PING host {host...}
```

Host is the internet address or host name of the computer you are checking. For example:

PING ADMIN RETURN

sends echo packets to the host `admin`. PING's display looks like this:

```
Pinging host admin: 79.0.0.5
Host admin replied to all 4 of the 4 Pings
```

The message may show the host replied to 3 out of 4 pings, for example. If the host is down, it won't reply to any of the pings.

One PING command can check several hosts. For example:

PING ADMIN SALES SUPPORT RETURN

checks the hosts `ADMIN`, `SALES` and `SUPPORT`.

Only one PING may run at a time (IP permits only one such opened ICMP device).



If PING seems to hang, press CTRL -C. This will display the number of properly echoed packets, if any.

TELDMP

TELDMP displays information about incoming TELNET connections, including the process id (pid), the job name, and the remote IP address and port number. If there are multiple servers, connections are listed for each server. TELDMP takes no arguments, so the syntax is simply:

```
TELDMP
```

TELDMP's display looks like this:

```
                ALPHATCP VIRTUAL TERMINAL SESSIONS
================================================================
  pid    jobnam   ipaddress:port running      username
----------------------------------------------------------------
Virtual Terminal Service
    26   TSKAAA   127.0.0.1:1052 STAT          System Service
```

FINGER

The FINGER command queries remote systems about logged in users and other information. Users on UNIX systems may choose to publish information about themselves, which is then available via FINGER, including their PGP public encryption keys. FINGER has even been used to query soft drink machines to determine if they are out of the user's favorite selection!

FINGER is used as follows:

```
FINGER {-s} {user}@host
```

The ***-s*** option requests the 'short form report'. This may be interpreted differently between FINGER servers. In short, it sends the query to the server without using /W, which may be required for some specialized services.

The ***user*** argument is optional. Providing a user login name will query for that user only. If you do not specify a user name, the query should return login or general host information instead.

Chapter 5 - Utilities

This chapter describes the utility commands included with AlphaTCP.

UUE

UUE is used to "uuencode" binary files. The encoded file may then be included into an electronic mail message or transferred by any other means requiring readable ASCII text. The format for UUE is as follows:

```
.UUE {filename.ext}
```

UUE will read the specified binary file and create an output file with the same name except the extension UUE will be used.



UUE only handles sequential files, not contiguous ones

UUD

UUD is used to decode "uuencoded" files back into binary form. The format for UUD is as follows:

```
.UUD {filenam.ext} {filenam.ext} {...}
```

If the source is split into multiple parts, such as when mailing large binary files, you may list all the parts in order on the command line. UUD creates the file listed on the "begin" line of the uuencoded file. If there are multiple files listed within all of the parts, UUD will decode each one. If the filename is not a valid AMOS filename, UUD will transform the name using the same method described in chapter 2.

EMAIL

EMAIL can be used to send a text file with optional attachments from the AMOS command line. The format for EMAIL is as follows:

```
.EMAIL{/D/M/T} messagefile {resultfile}
```

/D	Turns on debugging, displays command sequence
/M	Enables MIME support
/R	Used to add to the header the line *"Disposition-Notification-To: "Sender"sender@location.com"*.
/T	Turns on trace mode, displays message being sent

resultfile is an optional file which error messages will be written to. This may be useful if you need to handle error recovery from a program. Upon success, this file will contain the phrase **"Message sent"**

messagefile must begin with the following header. The remainder of the file may contain the body of the message:

```
Sender's Address
Reply-To Address (usually same as Sender's Address)
Subject
Recipient 1
Recipient 2
Recipient 3...
(blank line)
Body of Message
```

The following example would be typical of *messagefile*:

```
engineering@widget.com
engineering@widget.com
The new widget prototypes are ready
tester@widgetworld.org
testers@worldowidgets.com

The new batch of vivid blue widgets is ready to be tested.
Please do not mix with your prototype olive drab widgets as it
could look quite tacky.
```

Blind Copy

Any of the recipients may be sent a copy of the message without appearing to the others. This done by appending /B to the end of their address. For example, if one of the recipients is:

```
myboss@widget.com/b
```

the To: lost viewed by other recipients will not contain "myboss", even though a copy was sent to "myboss".

Attachments

Attachments may be included in the body of a message using the following line:

```
%attach filespec{/R/U}
```

/R	ROT13, a simple text-file encoding
/U	UUENCODE, an older binary file encoding

Text files may be included without any encoding. Binary files must always have some form of encoding. When MIME support is *not* specified, you *must* encode all binary files with /U (uuencode).

If you specify MIME support, the file extension will be searched for in LIB:MIME.TYP. Files listed as *7bit* will be included without encoding. All other files will be encoded with MIME *Base64* encoding. You may override this behavior by specifying one of the above encoding switches instead.

Appendix A - Error Messages

This section explains the error messages that may appear when you are using AlphaTCP. Suggested solutions are given where possible. When there is no suggested solution, or you cannot solve the problem, talk to your Network Administrator.

In addition to the messages listed here, you may see messages listing an AlphaTCP utility name and a reason the utility failed. See your Network Administrator for help with these messages.

`?Already connected to [host]`
: You tried to connect to somebody while you were already connected to the listed host. Disconnect and try again.

`?Ambiguous command`
: You used a short version of a command with more than one interpretation. Try again with a more complete command.

`ARP table alloc error`
: There is insufficient memory to dump the ARP table. Wait a few seconds and try again.

`Bad directory components`
: The specified directory does not exist on the remote host.

`[number]:bad port number`
: You've specified a bad port number when trying to make a connection. Either check the port number and try again or use the default port.

`Bad Sequence of commands`
: You are not using the FTP **rename** command properly. Check your syntax and try again.

`Cannot build data connection: [reason]`
: The remote computer could not establish a data connection to the local computer for the reason given. The console of the remote host should give further diagnostics. Based on the information, try to correct the condition and try again.

`Cannot determine current directory`
: The remote host cannot determine what the current directory is during FTP. Use the **cd** command to set the remote directory.

`Can't create data socket ([address], [port])`
: The remote computer cannot create a socket with the above address and port number. The console of the remote host should give further diagnostics.

`Can't find list of remote files, oops.`
: For some reason, FTP could not get a directory listing of the requested files from the remote host to the local computer.

`Can't set guest privileges`

You tried to login as guest, and the remote host is unable to set up the required environment because the default directory for the guest account does not exist.

`Child process closed`

The remote FTP server timed out the connection because it failed to read an FTP command. The connection should be closed with an FTP CLOSE command.

`Command not implemented or Command not understood`

Enter **`help`** RETURN or **`?`** RETURN for a list of available commands. If you have **sunique** on, you may see this message if the remote host doesn't support **sunique**.

`Connection Aborted`

The connection (TELNET, FTP) timed out while idle because the remote host is not responding to keepalive packets (the connection or computer is down).

`Connection closed by foreign host`

During a session, you either logged off or problems with the remote host caused it to close the connection.

`Connection Refused`

This message may appear when you are transferring files or running TELNET. It indicates there is no server on the remote computer to process your command. Arrange for the server on the remote computer to be started by the System Administrator of that computer or switch to another utility.

`Connection Reset`

The connection was reset by the remote host.

`Connection Timed Out`

The command you used can't establish a connection to the target host. The connection timed out because no response was received from the remote host. Usually, this is because the remote host is not running networking software or is down. You can check the hosts available by using NETSTA or PING.

`Error: Returned buffer [number] differs at char [number]`

The data transmitted in the numbered ICMP ECHO packet was not correctly echoed.

`Error in server: [reason] Closing Connection due to server error`

The remote host encountered the error described. Re-start the session. You may need to wait until a remote site problem is corrected.

`Exceeds licensed user count`

The new connection would exceed your AMOS license. Increase your AMOS license, or reduce the pre-allocated TRMDEF statements in your system initialization file. This can be done by removing TRMDEF statements for hardware ports or reducing the VTSER pre-allocated terminal definitions (TRMDEF #n...).

`Failed to unlink workfile`
: For some reason, FTP could not get a directory listing of the requested files from the remote host to the local computer.

`GATE Table alloc error`
: Not enough memory was available to print out the routing table.

`GATE Table read error`
: A problem occurred with the IP driver between requesting to read the routing table and reading it.

`ICMP Protocol already in use`
: PING is being executed by another user. Try again in a few seconds.

`IDV not set up - system error`
: You should not see this error message. Contact your VAR with information about this error.

`?Invalid command`
: Check the documentation or use **help** or **remotehelp** to see a list of commands.

`Login failed or Login Incorrect`
: The name and password information entered for the account on the remote computer is invalid. If you're sure you typed the name correctly, check with your Network Administrator to see why the name isn't valid.

`Memory request exceeds upper limit`
: The amount of memory you requested exceeds the amount specified as the upper limit for the server. Request less memory or increase the upper limit specified on the server startup line.

`The network is down`
: The networking software is not running on the local computer.

`Network Unreachable`
: There is no route to the network containing the host you wish to contact.

`No control connection for command`
: There is no control connection to pass the command to the remote host. Use the **open** command to establish a connection with the host.

`No memory available for connection`
: There is no more memory available for connections. Increase the SMEM statement in the system initialization command file.

`No more job allocations available`
: There are insufficient job table entries defined. Increase the JOBS statement number in the system initialization command file.

`[file]: not a plain file`
: You tried to retrieve a file which is either a directory or a special device. Use a real file name. You also receive this message if you try to **get** a random file from AMOS via TFTP, or via FTP if you are not in binary mode.

`Not Connected`
: There is no connection to the remote host. Use the **open** command to establish a connection.

`Out of memory`
: There is insufficient memory for the data structures required to do the transfer. Try again in a few seconds. If you continue to see this, contact your System Administrator about acquiring more memory.

`ping:getmsg [reason]`
`ping:putmsg [reason]`
: PING was unable to access the IP driver, for the reason stated.

`ping:Invalid Protocol`
: PING was unable to queue an ICMP Echo packet to the IP driver owing to an internal error in the driver.

`ping Unknown IP Error [number]`
: PING couldn't queue an ICMP Echo packet to the IP driver because of an internal error in the driver.

`Please login with [user] and PASS`
: The connection has been established to the remote host but you have not logged in. Try again using a permitted login on the remote host.

`Poll failed: [reason]`
: The data connection was not properly closed down owing to the described failure in the poll computer call.

`Ran out of memory`
: The remote host cannot allocate sufficient memory to do the requested command. Try again in a few seconds. If you continue to see this, contact your System Administrator to acquire more memory.

`Requested terminal type not supported`
: The terminal driver you selected is not available. Verify the name was entered properly. You may need to add a terminal driver to the system and load it using TDVDEF in the system initialization file.

`Ret ICMP type [number] (not ECHO_REPLY!)`
: A different ICMP type than ECHO_REPLY was received. If [number] is 3, an ICMP Network Unreachable packet was received because one of the gateways on the route to the host was unable to forward the ICMP Echo packet.

`Return msg type [number] (not ICMP_RX!)`
: An internal error in the IP driver occurred on reception of an incoming packet.

`Timeout [number] seconds: closing control connection.`
: The remote FTP server timed out the connection because an FTP command was not received in the time it was expected.

`Timeout occurred`
: The data connection was not properly closed down.

`t_look returned [error]`
: The data connection was not properly closed down owing to the event described.

`Try again later`
: A new process could not be created.

`[host]: unknown host`
: The host doesn’t have an entry in the TCP:HOSTS. file. Check the host name you entered for errors; if it looks correct, contact your Network Administrator.

`[MODE]: unknown mode`
: The data transfer mode specified is invalid or not supported.

`Unknown startup failure encountered`
: There is a mismatch between the TCPEMU and TELNED programs. Verify both are from the same release of AlphaTCP.

`User [user]: can't change directory to [directory]`
: The default directory for the specified user does not exist.

`User [user] cannot login`
: The specified user is forbidden from using FTP on the remote host by the System Administrator. Try again using a permitted login on the remote host.

`User [user] unknown`
: The user does not have an account on the remote host. Make sure you typed the user name correctly or try again using a different login.

Appendix B - AMOS/UNIX Comparison

AlphaTCP lets AMOS computers and computers running UNIX talk to each other and share information. Since it has not been possible to do this easily before, we assume that at least some AlphaTCP users are either AMOS users who are unfamiliar with UNIX or UNIX users who have never worked with AMOS before.

This appendix, therefore, is an introduction to AMOS and/or UNIX. It is written from a practical point of view, not a theoretical one. Anything going on behind the scenes which doesn't affect the way you use the computer (such as internal memory management) is not considered. The areas covered here include:

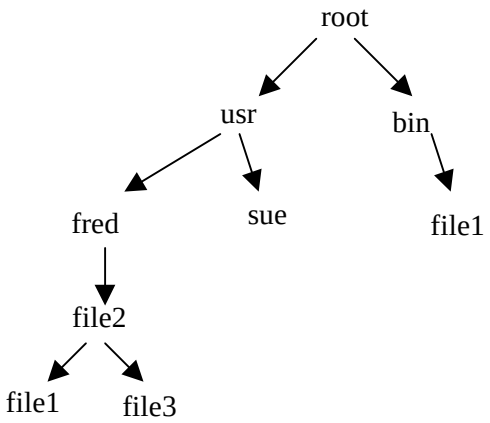
- A comparison of UNIX and AMOS file structures.
- Read/write security.
- A list of common, equivalent commands for the two operating systems.

While we assume you are a novice on UNIX or AMOS, we also assume you are an experienced computer user, and are familiar with terms such as disk, files, logging in, and so on.

FILE STRUCTURE

By file structure, we mean how files are logically stored on the disk and how you access them, not the internal data structure of a file, which is normally hidden from the user and is not discussed here. AMOS and UNIX arrange their files on the disk quite differently.

Each computer has a store of files. In UNIX, these files are arranged as the leaves on a tree of directories. Each directory is a branch, and may have both leaves (files) and smaller branches (subdirectories) coming out from it. Here's how the structure looks:





Those of you familiar with MS-DOS file structure probably recognize this. About the only difference between UNIX and DOS in this area is that UNIX uses slashes (/) in path names and DOS uses backslashes (\).

Here, **root, usr, bin, fred**, and **sue** are directories. **file1**, **file2**, and **file3** are files. Each file is uniquely identified by its pathname. A pathname is the path you follow to get from the root directory to a particular directory. For example, the pathname for **file2** is **/usr/fred/file2**.

There are two files with the name **file1**. They are differentiated by their pathnames. **file1** in the directory **fred** has the pathname **/usr/fred/file1**. The other **file1** has the pathname **/bin/file1**.

When Fred logs on, he sees the files in the directory **/usr/fred**. This is called his home directory or working directory, and is his default directory. All UNIX users have directories with their user names.

UNIX assumes that when Fred asks for **file2** while he's in this directory, he means **/usr/fred/file2**, so doesn't need to use the pathname. However, if Fred moves to another directory, he needs to use the full pathname to reach **file2**.

Under AMOS, files are arranged on disks in numbered accounts instead of named directory paths. Each account has a two-part number, and account numbers are normally shown in brackets, like this: [10,7]. Each part of the account number is an octal number between 0 and 377 (the first part cannot be 0 and the second cannot be 377). Each user can be assigned a default account they always go to when they log on.

Two accounts with the same first number are said to be in the same project. For example, the accounts [60,0], [60,3] and [60,376] are all in the [60,*] project.

Rather than making you remember account numbers, AMOS offers another way to identify accounts. For any account, you can define one or more ersatz names. For example, if Fred always uses account [100,70] when he's on AMOS, it could be assigned the ersatz name FRED:. Ersatz names contain from one to six letters and/or numbers, and are always followed by a colon.

Files in different AMOS accounts can have the same name, like files in different directories on UNIX. A file is uniquely identified by its disk device, account number and name, or its ersatz name and file name. For example, DSK2:SALES.LST[42,4] is a complete file specification (on a network, a file specification may also include a host name or address). Note that the account number follows the file name. When you use an ersatz name, it replaces the disk device part of the specification (it comes first), and there is no account number: FRED:LETTER.TXT.

FILE NAMES

The maximum length of a file name on the various implementations of UNIX varies from 14 to 255 characters. UNIX file names can include any character except a slash (/), though certain other characters are not recommended. Upper and lower case letters are treated differently, so **test** and **TEST** identify different files.

On AMOS, a file name consists of a one- to six-character name, a period, and an optional one- to three-character extension. Only letters and numbers are allowed, and upper and lower case letters are considered the same.

For example, **test.steve** is a legal UNIX file name, but would not be allowed under AMOS.



For the rules AlphaTCP uses when converting UNIX file names to AMOS, see Chapter 2.

READ/WRITE SECURITY

Since AlphaTCP has the potential of making your computer available to a large number of users (if any host on your network has a modem attached, a virtually unlimited number of users), data security is an important consideration. The security provided by the various AlphaTCP servers is described in the *AlphaTCP Administrator's Guide*. This section deals in general terms with AMOS and UNIX file security features.



This section deals only with security provided by the operating system itself. On both UNIX and AMOS, application software may include further security measures.

UNIX

The most basic security measure on UNIX is the same as on most multi-user computers—you must know a user name and password to log on to the computer. In addition, there is a file security system.

UNIX file security depends on the concept of file ownership. In UNIX, the user who creates a file "owns" it. This means he or she can set the file permission flags for that file. There are three types of permission:

- Write protection: you can change or add to the file.
- Read protection: you can read or copy the file.
- Execution permission: if the file is executable, you can execute it.

The owner can allow or deny these three types of permission for three different groups of users: the owner, members of the owner's group of users (user groups are set up by the system operator when the users are established), and everyone else.

For example, a file could be set so only its owner can write to it, only the owner and users in the same group can read it, and anyone on the computer can execute it. These permissions and protections are set and changed using the **chmod** command.

AMOS

Like UNIX, AMOS requires a user name and (optionally) a password to gain access to the computer. But, AMOS can include another level of password security. Accounts can also be given passwords. Thus, even if someone finds out or guesses a user's password, they won't be able to log to a specific account unless they also know the password for the account. Sensitive accounts should always be given passwords, especially in networked installations where it is difficult to control computer access.

Another AMOS security feature applies only to networked computers. You can set any logical disk device to "noaccess." Setting a device to noaccess doesn't affect local users; they can still access files on the disk. But no one can access any information on the disk over a network connection. Note that this stops FTP or TFTP file copies, but does not prevent a TELNET user from accessing the device. Since TELNET creates a virtual terminal connection, a TELNET user looks just like a local user.

In the extended directory format, AMOS also offers file level security. Read, write, execute, and delete/rename permission can be set separately for users in this directory, users in other directories in this project, local users in other projects, users in the same network group, and all other users. The permissions/protections are set according to the user's defaults when the file is created. User defaults are established with the MUSER command. See the *AMOS Monitor Calls Manual* for more information.

EQUIVALENT COMMAND LIST

The following list shows some of the more common commands in both UNIX and AMOS. It gives only the command names, not their formats or any options. Some of the commands listed have more than one function, depending on the format you use.

AMOS COMMAND	UNIX COMMAND	DESCRIPTION
DIR	**ls**	Lists files in the current account (AMOS) or directory (UNIX).
VUE	**vi**	Invokes standard full screen text editor.
STAT	**who**	Shows the users currently logged on the computer. STAT also shows the current activity of each job (user).
LOG	**cd**	Lets you log to a different account or directory. LOG also lets you log onto the computer when you are logged off.
LOG	**pwd**	Displays the account or directory path you are currently logged into. LOG displays the current account if you enter it without an account to log to.
COPY	**cp**	Copies a file.
RENAME	**mv**	Changes a file's name. **mv** also lets you move a file to another directory.
ERASE	**rm**	Erases a file.

Glossary

ARP	Address Resolution Protocol, RFC 826. A broadcast protocol used to resolve Internet Addresses into physical network addresses.
ARPA	The U.S. Defense Department Advanced Research Projects Agency.
DDN	Defense Data Network, the long-haul network for the U.S. Defense Department.
DNS	The Domain Name Service is a distributed database of host names and addresses. It replaces the HOSTS. file in large installations. AlphaTCP supports the Resolver portion of DNS.
Ersatz	On AMOS, a one- to six-character name, followed by a colon, which precedes a file name in a file specification. The ersatz name defines a specific disk account. For example, SYS: is the ersatz name for the DSK0:[1,4] account.
Ethernet	Ethernet, or IEEE 802.3, is a network based on an access method called CSMA/CD. Ethernet was originally developed by the Xerox Corporation in 1972. A specification was published in 1980 for others to use. This specification, known as the "Blue Book," was the basis for CSMA/CD LANs published in 1983.
Ethernet address	A 48-bit station identifier. Each network node has a unique address. The high order bytes are usually manufacturer specific. Certain reserved addresses have special meaning. An all-ones address is a broadcast, received by all stations.
Gateway	A device which interconnects two networks at the IP level.
ICMP	Internet Control Message Protocol, RFC 792, used by gateways and hosts.
Internet Address	A 32-bit address used to identify nodes on the network which use the Internet Protocol.
IP	Internet Protocol, RFC 791. The lower level of the ARPA protocols.
Keepalive	An on-going handshaking that maintains a connection between hosts.
LAN	A Local Area Network, in which several hosts and devices are located near each other are connected physically by cables. Typically, these cables are Ethernet coaxial cables.
LP	The Line Printer protocol is used to transfer print data to a host with a printer attached. AlphaTCP supports the client side of the Line Printer protocol.
Packet	A unit of information containing a header with addresses and handling informa-

	tion and a message. Longer messages can be split up into several different packets and sent separately.
RFC	A “Request for Comment” discussion document produced by interested parties, which may be adopted as a standard.
RWHO/RUPTIME	Berkeley protocol used to transmit UPTIME over UDP.
SLIP	Serial Line Internet Protocol. Allows use of TCP/IP protocols over serial connections such as phone lines.
SMTP	The Simple Mail Transfer Protocol is used to pass electronic mail messages between hosts.
TCP	Transmission Control Protocol, RFC 793. Used to provide a transport service in the ARPA protocol set.
TELNET	The ARPA application level protocol, RFC 854, RFC 855. A bi-directional, byte-oriented communications protocol.
TFTP	Trivial File Transfer Protocol, RFC 783. This is used to receive or send an ASCII or binary code image.
UDP	User Datagram Protocol, RFC 768. This provides a datagram service over the Internet Protocol.

Document History

REVISION	RELEASE	DATE	DESCRIPTION
00	1.0	4/93	New document for AlphaTCP 1.0; adapted from Spider TCP documentation.
01	1.1	1/94	Revised for AlphaTCP 1.1. Added TELDMP command, ! option for TELNET. Various other small changes.
02	1.2	2/95	Revised for AlphaTCP 1.2. Added information on random file transfer.
03	1.3	7/95	Revised for AlphaTCP 1.3. Added new features and enhancements.
04	1.3B	9/96	Revised for AlphaTCP 1.3B. Added new FTP switch and FINGER command.
05	1.4	8/97	Revised for AlphaTCP 1.4. Added MYLPR command to Chapter 3.
06	1.4	6/98	Added URL command to Chapter 2.
07	1.4A	3/99	Added EMAIL utility command and Attachments
08	1.5	9/99	Added EMAIL blind copy feature
09	1.5B	4/08	Added EMAIL /R switch

INDEX

!

! command in TELNET · 3-3

A

Address
  Ethernet · 1-3
  internet · 1-3
AMOS
  command list · B-4
  compared to UNIX · B-1
  file structure · B-1
  security · B-3
append command in FTP · 2-3
ARP command · 1-3
ASCII file in FTP · 2-7

B

Beginning an FTP session · 2-5
bell command in FTP · 2-3
Binary files
  in FTP · 2-7
  in TFTP · 2-10
Breaking a TELNET connection · 3-3
Breaking an FTP connection · 2-3
bye command in FTP · 2-3
bye command in TELNET · 3-3

C

Case sensitivity of AlphaTCP commands · 1-4
cd command in FTP · 2-3
Checking remote host status · 4-2
close command in FTP · 2-3
close command in TELNET · 3-3
connect command in TELNET · 3-3
Connections, listing · 4-2
Contiguous file, transfer · 2-8
Conventions used in this book · iii
Converting UNIX/AMOS file names · 2-12, 2-14
Copying files · 2-1, 2-2
  contiguous · 2-8
  name conversion · 2-12, 2-14
  with FTP · 2-4, 2-5, 2-6
  with TFTP · 2-9
Copying files with FTP · 2-6
crmod command in TELNET · 3-3

D

debug command in FTP · 2-3
Default settings for FTP · 2-6
delete command in FTP · 2-3
Deleting a directory with FTP · 2-6
Deleting files with FTP · 2-4
dir command in FTP · 2-3
Directory
  changing during FTP · 2-3, 2-4
  creating with FTP · 2-5
  displaying remote directory · 2-5
  listing with FTP · 2-3, *2-4*
  removing with FTP · 2-6
Displaying help during FTP · 2-6
Displaying help in FTP · 2-4, 2-7
Displaying help in TELNET · 3-3
Displaying help information in TELNET · 3-4
Displaying TELNET status · 3-4
Documentation, other · iii
Dynamic IP printing · 3-4

E

EMAIL
  attachments · 5-2
  blind copy · 5-2
  utility · 5-1
Ending a session in FTP · 2-6
Ending a TELNET session · 3-2, 3-3, 3-4
Ending an FTP session · 2-3
Error messages · A-1
escape command in TELNET · 3-3
Establishing a TELNET connection · 3-3
Ethernet addresses · 1-3
Executing an AMOS command during TELNET · 3-3

F

File names
  UNIX/AMOS conversion · 2-12, 2-14
File security · 1-4
File structure
  AMOS and UNIX · B-1

File transfer · 2-1, 2-2
File type
  during FTP command · 2-2
Files
  copying with FTP · 2-4, 2-5, 2-6
  copying with TFTP · 2-9
  deleting with FTP · 2-3, 2-4
  renaming with FTP · 2-6
Filespec · iv
FTP command · 2-1, 2-2
  appending files · 2-3
  argument format · 2-2
  bell option · 2-3
  bye command · 2-3
  cd command · 2-3
  changing the local directory · 2-4
  changing the remote directory · 2-3
  close command · 2-3
  command line options · 2-7
  commands · 2-2
  copying file · 2-5
  copying files · 2-4, 2-5, 2-6
  creating a directory · 2-5
  debug command · 2-3
  default directory and account · 2-2
  default settings · 2-6
  delete command · 2-3
  deleting files · 2-3, 2-4
  dir command · 2-3
  displaying a directory · 2-3, 2-4, 2-5
  displaying help · 2-6, 2-7
  ending a session · 2-3, 2-6
  file name conversion · 2-12, 2-14
  file type · 2-2
  format · 2-1
  get command · 2-4
  hash command · 2-4
  lcd command · 2-4
  literal command · 2-4
  ls command · 2-4
  mdelete command · 2-4
  mget command · 2-5
  mkdir command · 2-5
  mput command · 2-5
  open command · 2-5
  prompt · 2-1
  prompt command · 2-5
  put command · 2-5
  pwd command · 2-5
  quit command · 2-6
  read and write restrictions · 2-2
  recv command · 2-6
  remotehelp · 2-6
  removing a directory · 2-6
  rename command · 2-6
  renaming files · 2-6
  rmdir command · 2-6
  runique command · 2-6
  runqiue command · 2-14
  sample · 2-8
  send command · 2-6
  setting file types · 2-7
  specifying unique file names · 2-6, 2-7, 2-14
  status command · 2-6
  sunique command · 2-7, 2-14
  type command · 2-7
  user command · 2-7
  user names · 2-7
  verbose · 2-7
  viewing settings · 2-6
FTP.LOG file · 2-5

G

get command in FTP · 2-4
Graphics conventions · iii

H

hash command in FTP · 2-4
help command in FTP · 2-4
help command in TELNET · 3-3
Help information in FTP · 2-4, 2-6, 2-7
Help information in TELNET · 3-3, 3-4
Host · 1-3
  address · 1-3
  names · 1-2, 1-3
Host status, checking · 4-2

I

Image files in FTP · 2-7
Internet
  addresses · 1-3

L

lcd command in FTP · 2-4
Listing incoming connections · 4-2
literal command in FTP · 2-4
Local printing · 3-4
ls command in FTP · 2-4

M

Making a virtual terminal connection · 3-1
mdelete command in FTP · 2-4
Messages · A-1
mget command in FTP · 2-5
mkdir command in FTP · 2-5
mput command in FTP · 2-5
MYLPR command · 3-4

N

Network access to AMOS files · 1-4
Network information, viewing · 4-1
Noaccess, on disk device · 1-4

O

open command in FTP · 2-5
open command in TELNET · 3-3
Opening an FTP session · 2-5
Option settings for FTP · 2-6
options command in TELNET · 3-3
Options of FTP command · 2-7
Options of TFTP command · 2-10
Other documentation · iii

P

Permission to read and write files · 1-4
PING command · 4-2
Printing to a local printer · 3-4
prompt command in FTP · 2-5
Prompt for FTP commands · 2-1
Protocol definitions · 1-4
put command in FTP · 2-5
pwd command in FTP · 2-5

Q

quit command in FTP · 2-6
quit command in TELNET · 3-4

R

Random file, transfer · 2-8
recv command in FTP · 2-6
Remote printing · 3-4
remotehelp command in FTP · 2-6
Removing a directory with FTP · 2-6
rename command in FTP · 2-6
Restrictions on FTP command · 2-2
rmdir command in FTP · 2-6
runique command in FTP · 2-6, 2-14
RUPTIM command
    options · 4-1
    viewing network information · 4-1

S

Security, AMOS and UNIX · B-3
send command in FTP · 2-6
Setting terminal type in TELNET · 3-1
Starting an FTP session · 2-5
status command in FTP · 2-6
status command in TELNET · 3-4
sunique command in FTP · 2-7, 2-14

T

TCP · 1-1
TELDMP command · 4-2
TELNET command · 3-1
    ! command · 3-3
    breaking a connection · 3-3
    bye command · 3-3
    close command · 3-3
    command mode · 3-4
    connect command · 3-3
    crmod command · 3-3
    displaying help · 3-3
    displaying help information · 3-4
    displaying status · 3-4
    ending a session · 3-2, 3-3, 3-4
    escape command · 3-3
    establishing a connection · 3-3
    help command · 3-3, 3-4
    input mode · 3-4
    open command · 3-3
    options command · 3-3
    quit command · 3-4
    setting terminal type · 3-1
    setting the escape character · 3-3
    status command · 3-4
    switching between modes · 3-4
    viewing options · 3-3
Tenex files in FTP · 2-7
Terminal type
    setting in TELNET · 3-1
TFTP command · 2-9
    file name conversion · 2-12, 2-14
    options · 2-10
Transferring files · 2-1, 2-2
type command in FTP · 2-7
Type of files
    setting in FTP · 2-7

U

Unique file names in FTP · 2-6, 2-7, 2-14
UNIX
    command list · B-4
    compared to AMOS · B-1
    file structure · B-1
    security · B-3
UNIX/AMOS file name conversion · 2-14
URL command · 2-10
    format · 2-11
    url format · 2-10
url format in AMOS · 2-11
user command in FTP · 2-7

User name in FTP · 2-7
UUD utility · 5-1
UUE utility · 5-1

V

verbose mode in FTP · 2-7
Viewing network information · 4-1
Virtual terminal connection · 3-1