A/UX Networking Essentials

Release 3.0

Limited Warranty on Media and Replacement

If you discover physical defects in the manuals distributed with an Apple product or in the media on which a software product is distributed, Apple will replace the media or manuals at no charge to you, provided you return the item to be replaced with proof of purchase to Apple or an authorized Apple dealer during the 90-day period after you purchased the software. In addition, Apple will replace damaged software media and manuals for as long as the software product is included in Apple's Media Exchange Program. While not an upgrade or update method, this program offers additional protection for up to two years or more from the date of your original purchase. See your authorized Apple dealer for program coverage and details. In some countries the replacement period may be different; check with your authorized Apple dealer.

ALL IMPLIED WARRANTIES ON THE MEDIA AND MANUALS, INCLUDING IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, ARE LIMITED IN DURATION TO NINETY (90) DAYS FROM THE DATE OF THE ORIGINAL RETAIL PURCHASE OF THIS PRODUCT.

Even though Apple has tested the software and reviewed the documentation, **APPLE MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS, OR IMPLIED, WITH RESPECT TO SOFTWARE, ITS QUALITY, PERFORMANCE, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS SOFTWARE IS SOLD "AS IS," AND YOU, THE PURCHASER, ARE ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND PERFORMANCE.**

IN NO EVENT WILL APPLE BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY DEFECT IN THE SOFTWARE OR ITS DOCUMENTATION, even if advised of the possibility of such damages. In particular, Apple shall have no liability for any programs or data stored in or used with Apple products, including the costs of recovering such programs or data.

THE WARRANTY AND REMEDIES SET FORTH ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL OTHERS, ORAL OR WRITTEN, EXPRESS, OR IMPLIED. No Apple dealer, agent, or employee is authorized to make any modification, extension, or addition to this warranty.

Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.

Apple Computer, Inc.

© 1992, Massachusetts Institute of Technology and Apple Computer, Inc. All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, mechanical, electronic, photocopying, recording, or otherwise, without prior written permission of Apple Computer, Inc. Printed in the United States of America.

The Apple logo is a registered trademark of Apple Computer, Inc. Use of the "keyboard" logo (Option-Shift-K) for commercial purposes without the prior written consent of Apple may constitute trademark infringement and unfair competition in violation of federal and state laws.

Apple Computer, Inc.
20525 Mariani Ave.
Cupertino, CA 95014-6299
408-996-1010

APDA, Apple, the Apple logo, AppleShare, AppleTalk, A/UX, EtherTalk, ImageWriter, LaserWriter, LocalTalk, Macintosh, and StyleWriter are trademarks of Apple Computer, Inc., registered in the United States and other countries.

Finder and Quadra are trademarks of Apple Computer, Inc.

Adobe Illustrator and Postscript are trademarks of Adobe Systems Incorporated, registered in the United States.

ITC Garamond and ITC Zapf Dingbats are registered trademarks of International Typeface Corporation.

Linotronic is a registered trademark of Linotype Co.

Microsoft is a registered trademark of Microsoft Corporation.

NFS is a registered trademark of Sun Microsystems, Inc.

QuarkXPress is a registered trademark of Quark, Inc.

UNIX is a registered trademark of UNIX Laboratories.

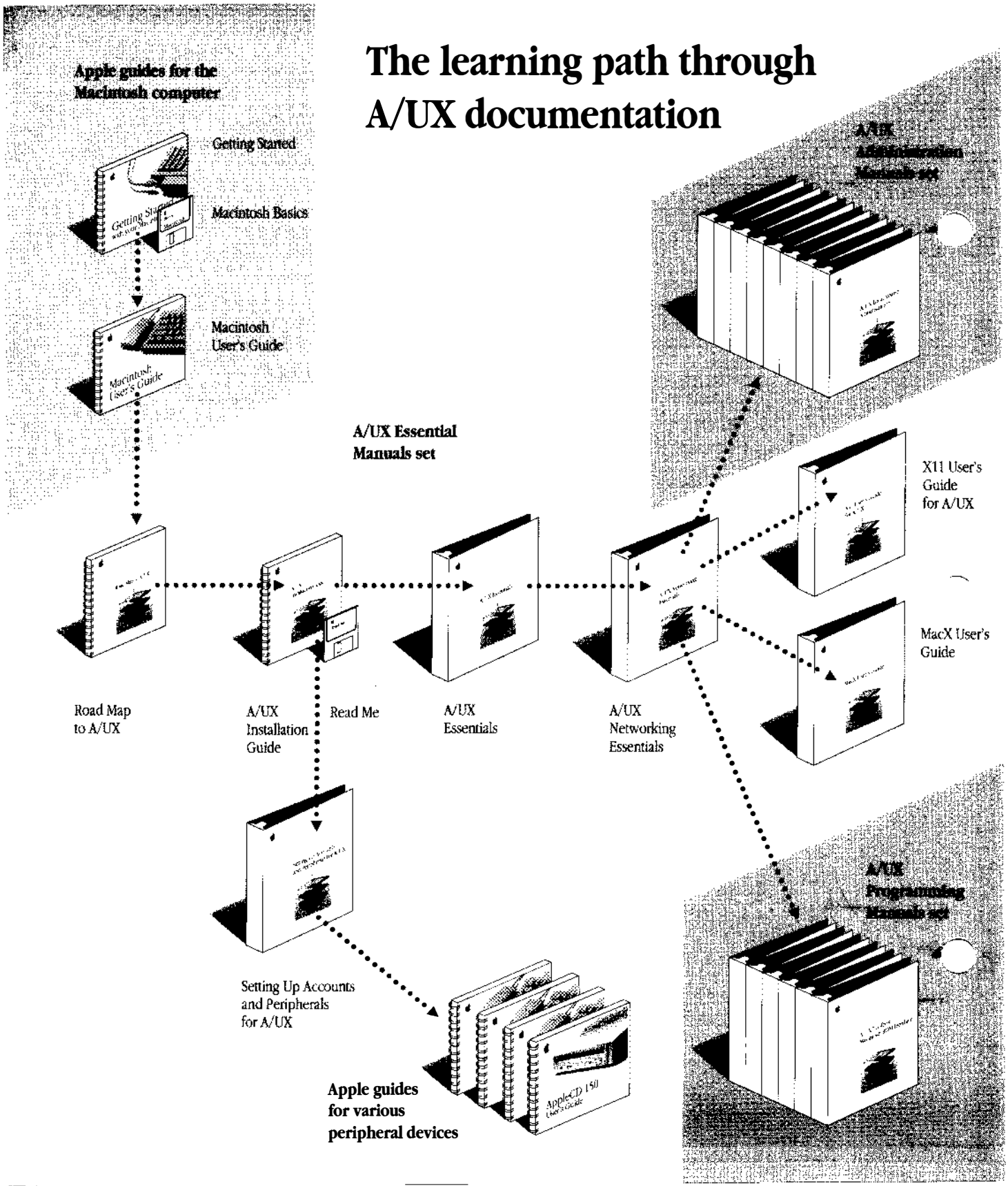
VAX is a trademark of Digital Equipment Corporation.

X Window System and X11 are trademarks of the Massachusetts Institute of Technology.

Simultaneously published in the United States and Canada.

Mention of third-party products is for informational purposes only and constitutes neither an endorsement nor a recommendation. Apple assumes no responsibility with regard to the performance or use of these products.

The learning path through A/UX documentation



Contents

About This Guide / xv

What you should already know / xv
How to use this guide / xvi
What's covered in this guide / xvi
What's covered elsewhere / xviii
Conventions used in this guide / xviii
- Keys and key combinations / xviii
- Terminology / xix
- The `Courier` font / xx
- Font styles / xx
- A/UX command syntax / xxi
- Manual page reference notation / xxii

For more information / xxiii

1 Introduction to A/UX Networking / 1-1

A/UX network connections / 1-3
Network features of your A/UX system / 1-4
Choosing a network file-sharing option / 1-5

2 Connecting to an Existing Network / 2-1

Types of networks you can connect to / 2-3
Identifying your network / 2-4
Connecting to a TCP/IP network / 2-6
Before you connect to a network / 2-6
Setting up your computer to use the TCP/IP network / 2-9
Testing the network connection / 2-12
Switching AppleTalk networks / 2-13
Switching from LocalTalk to EtherTalk / 2-13
Switching from EtherTalk to LocalTalk / 2-16
What's available now / 2-18

3 Printing Over the Network / 3-1

Understanding network printing / 3-2
Printer connections / 3-2
Choosing a printer / 3-4
Printing from the Finder / 3-5
Printing with `lpr` / 3-6
Setting up access to a printer on a TCP/IP network / 3-6
Sending a file to a remote UNIX printer / 3-8
Finding out how many files are waiting in the printing queue / 3-9
Canceling a printing job / 3-10
Overview of available print facilities / 3-11

4 Sending and Receiving Mail / 4-1

Creating a mail alias / 4-3
Sending mail / 4-3
Sending mail to users on your local system / 4-4
Sending mail to one person / 4-5
Sending mail to more than one person / 4-6
Using an alias to send mail to more than one person / 4-6
Sending mail to users on other systems / 4-8
Sending mail to a user on the TCP/IP network / 4-8
Sending mail to more than one person on the TCP/IP network / 4-9

Sending mail to a user through UUCP / 4-10
Sending mail to more than one person through UUCP / 4-11
Canceling a message / 4-11
Receiving and reading mail / 4-12
Reading mail / 4-12
Deleting a message / 4-13
Saving a message / 4-14
Exiting `mail` / 4-14

5 Using TCP/IP / 5-1

What TCP/IP adds to a network / 5-2
TCP/IP networking commands / 5-4
Finding out which computers can be accessed / 5-5
Working with network permissions / 5-7
Setting up the `.rhosts` file / 5-8
Connecting to other computers / 5-10
Using `rlogin` to log in to a remote computer / 5-10
Using the remote system / 5-12
Ending the `rlogin` session / 5-14
Using `telnet` to log in to a remote computer / 5-14
Logging in / 5-15
Learning the current status / 5-15
Logging out / 5-16
Using `telnet` commands / 5-16
Running commands on other computers / 5-17
Using `remsh` 5-17
Ending the `remsh` session / 5-19
Copying files between systems / 5-19
Using `rcp` 5-19
Using `rcp` to copy files / 5-19
Using `rcp` to copy directories / 5-21
Shorthand notations / 5-21
Ending the `rcp` session / 5-22
Using File Transfer Protocol / 5-22
Using `ftp` to copy files / 5-22
Using `ftp` to transfer files / 5-23
Using directories on the remote computer / 5-24
Logging out / 5-24

6 Sharing Files Over the AppleTalk Network / 6-1

About file sharing / 6-3
What you should know before you begin / 6-4
Starting file sharing / 6-4
Sharing a folder or disk / 6-6
Naming specific people to share folders and disks with / 6-7
Naming a user / 6-8
Naming a group of users / 6-11
Reviewing group membership / 6-12
Sharing a folder or disk with specific people / 6-13
Setting up your computer for remote accessing / 6-15
Changing your password / 6-17
Giving folder ownership to someone else / 6-18
Changing a user's password / 6-19
Monitoring file-sharing activity / 6-20
Disconnecting a user / 6-21
Reducing access to your shared files / 6-22
Making a shared folder or disk private again / 6-22
Denying access to a specific user / 6-22
Denying access to guest users / 6-23
Denying access to yourself / 6-24
Removing a user's membership in a group / 6-24
Removing a user or group altogether / 6-24
Turning off file sharing / 6-25
About accessing information on other computers / 6-26
What to do before you begin / 6-27
Connecting to a shared disk / 6-28
Connecting automatically / 6-32
Using aliases to connect quickly / 6-32
Connecting automatically when you start up / 6-33
Using and storing files on a shared disk / 6-35
Disconnecting from a shared disk / 6-35
Changing your password to access a shared disk / 6-36
Affecting access to folders / 6-37
Giving folder ownership to someone else / 6-37
Accessing your own computer remotely / 6-38

7 Sharing Files With NFS / 7-1

About NFS file sharing / 7-3
When to use NFS / 7-4
What you need to run NFS / 7-4
Preparing your system to run NFS / 7-5
Making your files available (Being a server) / 7-6
Using files on other systems (Being a client) / 7-7
Modifying the `/etc/fstab` file / 7-10
Verifying the mounting / 7-11

8 Using UUCP to Connect to a Remote System / 8-1

Introduction to UUCP / 8-2
What you need to run UUCP / 8-4
Finding names of connected UUCP systems / 8-4
Checking UUCP network status / 8-4
Copying files / 8-5
Identifying remote files / 8-5
Copying a file directly to another system / 8-7
Copying a file to another system through an intermediate system / 8-7
Copying a file from another system through an intermediate system / 8-8
Using your system as an intermediate (switching) / 8-8
Copying multiple files / 8-9
Using `uuto` to copy files / 8-10
Running commands on a remote computer: `uux` / 8-12
Tracking remote commands and determining status / 8-12
Receiving notification / 8-12
Checking job status: `uustat` / 8-13
Job control / 8-13
Terminating jobs / 8-13
Requeuing a job / 8-14
Other utilities that use UUCP / 8-14

9 Using `cu` to Connect to a Remote System / 9-1

Selecting a remote session option / 9-2
Connecting to a remote system / 9-5
Switching between local and remote commands / 9-6
Sending local output to a remote system / 9-7
Copying files / 9-8
Using `cu` with non-A/UX computers / 9-8
Closing the `cu` connection / 9-9

10 Using MacX on A/UX / 10-1

Special terms used in this chapter / 10-3
About MacX / 10-4
Before you begin / 10-7
Starting a MacX session / 10-7
Opening client applications / 10-8
Giving client applications access to your computer / 10-9
Opening a local client application from the A/UX Finder / 10-10
Opening a local client application using the Remote Command window / 10-11
Opening a local client application from CommandShell or `xterm` command line / 10-15
Using a CommandShell window / 10-15
Using an `xterm` window / 10-16
Disconnecting a client application / 10-18
Ending a MacX session / 10-19
Quitting the MacX application / 10-19
Closing a settings document without quitting MacX / 10-21
Additional reading / 10-23

Appendix A Troubleshooting / A-1

`rlogin` messages / A-2
`telnet` messages / A-3
`remsh` messages / A-4
`rcp` messages / A-5
Correcting `newconfig` errors / A-6

Appendix B More About Mail / B-1

Using the tilde escape / B-2
Displaying a summary of tilde escapes / B-2
Displaying the message you are currently entering / B-2
Canceling a tilde escape / B-3
Editing a mail message / B-3
Including a file with a mail message / B-4
Saving a current message / B-4
Including a message with a message / B-5
Adding recipients to a message / B-5
Changing the subject line of a message / B-6
Sending message copies / B-6
Editing mail message fields / B-7
Escaping to the shell / B-7
Accessing the shell from within mail / B-8
Escaping to the `mail` command mode / B-9
Displaying the message to which you're replying / B-9
Using the tilde character with the tilde escape / B-9
Changing the escape character / B-10
Working with mail messages / B-10
Displaying several messages at once / B-11
Replying to a message / B-12
Mailing from within mail / B-13
Forwarding a message / B-14
Reading mail from a different file / B-14
Customizing mail / B-14
Using the `set` command / B-15
Setting up a permanent mail environment / B-16
Sending mail to files / B-17
Sending a file through mail / B-17

Appendix C More About `remsh` / C-1

Using remote devices / C-2
Ending the `remsh` session using the `kill` command / C-3
Local and remote shell topics / C-3
Quoting mechanisms / C-4

`remsh` command line considerations / C-6
Multiple commands / C-6
Pipelines / C-6
Filename expansion / C-7
Output redirection / C-8
History substitution / C-9
Aliases / C-9
The `grep` command / C-10
Shell variables / C-10
Shell scripts / C-11
`remsh` options / C-12

Appendix D More About `ftp` / D-1

Setting conditions / D-2
Using `ftp` options / D-2
`ftp` commands / D-3
Setting up the `.netrc` file / D-10
Abbreviated command formats / D-11
Toggle commands / D-12
File transfer parameters / D-14

Glossary / Gl-1
Index / In-1

Figures and Tables

Chapter 1 Introduction to A/UX Networking

Figure 1-1 Network connections you can make as an A/UX user / 1-2

Table 1-1 Networking features of your A/UX system / 1-4

Chapter 2 Connecting to an Existing Network

Figure 2-1 LocalTalk cables and connectors / 2-4
Figure 2-2 Ethernet cables and connectors / 2-4
Figure 2-3 Apple Ethernet media adapters / 2-5
Figure 2-4 LocalTalk printer ports / 2-14

Chapter 3 Printing Over the Network

Figure 3-1 Two types of printer connections / 3-3

Table 3-1 Print commands / 3-11

Chapter 5 Using TCP/IP

Figure 5-1 A/UX connected to TCP/IP services / 5-3
Figure 5-2 Wide area and local area networks / 5-4
Figure 5-3 Network computer status lines / 5-6
Figure 5-4 Logging in to a remote computer / 5-11
Figure 5-5 Printing a local file on a remote printer / 5-18

Chapter 6 Sharing Files Over the AppleTalk Network

Figure 6-1 Sharing files on the network / 6-3

Chapter 7 Sharing Files With NFS

Figure 7-1 Remotely mounted manual pages / 7-3

Chapter 8 Using UUCP to Connect to a Remote System

Figure 8-1 A/UX connected to serial services / 8-3
Figure 8-2 Sending the `prospects` file from `yosemite` to `pinnacle` / 8-7
Figure 8-3 Copying the `prospects` file from `shasta` to `tahoe` / 8-9

Chapter 9 Using `cu` to Connect to a Remote System

Figure 9-1 A simple `cu` configuration / 9-3
Figure 9-2 Using `cu` remotely / 9-4

Chapter 10 Using MacX on A/UX

Figure 10-1 MacX running under the Macintosh OS / 10-5
Figure 10-2 MacX running under A/UX / 10-6

About This Guide

Your Apple Macintosh computer comes with a number of built-in networking capabilities, which the A/UX system extends by adding the networking features of the UNIX® operating system. A/UX offers a variety of programs that allow you to communicate with other users and gain access to information and services from a wide range of outside sources. Indeed, given the myriad of network services available with A/UX, the process of selecting from those services can be one of the most challenging aspects of using the network effectively. *A/UX Networking Essentials* is intended to introduce you to the advantages that networking offers and to quickly guide you to the means by which you can make best use of the network to suit your specific purposes.

What you should already know

This guide assumes that you're reasonably comfortable getting around on the Macintosh and using A/UX. The procedures in *A/UX Networking Essentials* assume that you've learned the basic skills in *A/UX Essentials.* Also, because A/UX includes all the standard Macintosh networking features, you may find useful procedures in *Macintosh Networking Reference.*

How to use this guide

This manual addresses two audiences: the network novice, someone who knows the Macintosh but has little experience working with a network, and the more experienced network user, someone who may be familiar with the network environment and its terminology. To serve both audiences, we've arranged the chapters to group the most-used procedures first (Chapters 2–7) and the environment-specific chapters later (Chapters 8–10). The appendixes cover the more in-depth (and less often used) information about the `mailx`, `remsh`, and `ftp` commands.

What's covered in this guide

A/UX Networking Essentials is divided into these chapters:

- Chapter 1, "Introduction to A/UX Networking," introduces the networking capabilities of A/UX and tells you how to use this manual.
- Chapter 2, "Connecting to an Existing Network," describes how to set up a Macintosh computer running A/UX so that you can use A/UX on a network. It covers the available networking options and discusses how to physically connect your system to a network. This chapter also discusses procedures for obtaining a network address.
- Chapter 3, "Printing Over the Network," discusses printing with a printer that is connected to a network. It tells you how to print with the Macintosh interface and how to set up access to a printer in order to print using the `lpr` command.
- Chapter 4, "Sending and Receiving Mail," discusses the `mailx` program, the facility you use to send and receive electronic mail. This chapter covers the most-often used commands that allow you to send and receive messages, save them, and delete them.
- Chapter 5, "Using TCP/IP," describes how to work with files, programs, and information that's stored on computers other than your own. It also gives you a brief overview of network permissions. When running A/UX you can use Macintosh file sharing in addition to UNIX methods, such as the remote copy program (`rcp`) to copy files between computers, and File Transfer Protocol (`ftp`), which also copies files between computers. This chapter also describes how to log in to different computers (`rlogin` and `telnet`) and how to run commands on different computers on the network (`remsh`).

- Chapter 6, "Sharing Files Over the AppleTalk Network," describes Macintosh file sharing in the A/UX environment. Just as you can make files, disks, and other portions of a standard Macintosh available to other users, you can do the same with a Macintosh that's running A/UX.
- Chapter 7, "Sharing Files With NFS," describes sharing files with other computers running A/UX or another UNIX system; this includes mounting files from remote systems and enabling others to do the same to your directories.
- Chapter 8, "Using UUCP to Connect to a Remote System," describes how to use the UUCP utilities to connect to a remote computer in order to copy files, run commands on remote computers, and determine job status.
- Chapter 9, "Using `cu` to Connect to a Remote System," describes how to use the `cu` program to connect to a remote computer in order to copy files and run commands on remote computers.
- Chapter 10, "Using MacX on A/UX," introduces you to the advantages of the X Window System on A/UX. It also tells you how to start and end a MacX session.
- Appendix A, "Troubleshooting," describes some potential network problems and gives steps for diagnosing and fixing them.
- Appendix B, "More About Mail," is a reference section that discusses those less-used commands that allow messages to be edited, included in other messages, and "carbon copied" to other users. This appendix also covers mail routing, the means by which you can use your local mail program to convey messages across other networks and receive mail routed from other systems.
- Appendix C, "More About `remsh`," is a reference section that covers `remsh` options, gives directions for using remote devices, and describes how various A/UX shell capabilities are used on the network.
- Appendix D, "More About `ftp`," is a reference section that gives you more information about `ftp`, such as how to set conditions and use `ftp` options and commands.

What's covered elsewhere

Here are some related topics that are covered in other manuals about the Macintosh and about A/UX:

- Setting up and configuring various types of networks is covered in *A/UX Network System Administration.*
- The basics of A/UX, such as logging in and manipulating files and directories, are tasks covered in *A/UX Essentials.* You should be thoroughly acquainted with these tasks before using this manual.
- Information on connecting and managing printers is located in *Setting Up Accounts and Peripherals for A/UX.*
- Basic information on UNIX permissions is located in *A/UX Essentials,* and more in-depth information on permissions is located in *A/UX Local System Administration.*
- Your A/UX system (Release 3.0 and later) also offers you the ability to link programs to programs on other computers and to update files automatically (the System 7 "publish and subscribe" feature). These features work exactly the same on A/UX as they do on the Macintosh System 7 software. To learn how to use these programs, see the *Macintosh Networking Reference* and the documentation that came with your Macintosh.

Conventions used in this guide

A/UX guides follow specific conventions. For example, words that require special emphasis appear in specific fonts or font styles. The following sections describe the conventions used in all A/UX guides.

Keys and key combinations

Certain keys on the keyboard have special names. These modifier and character keys, often used in combination with other keys, perform various functions. In this guide, the names of these keys are in Initial Capital letters followed by SMALL CAPITAL letters.

The key names are

CAPS LOCK	DOWN ARROW (↓)	OPTION	SPACE BAR
COMMAND (⌘)	ENTER	RETURN	TAB
CONTROL	ESCAPE	RIGHT ARROW (→)	UP ARROW (↑)
DELETE	LEFT ARROW (←)	SHIFT	

Sometimes you will see two or more names joined by hyphens. The hyphens indicate that you use two or more keys together to perform a specific function. For example,

Press COMMAND-K

means "Hold down the COMMAND key and then press the K key."

Terminology

In A/UX guides, a certain term can represent a specific set of actions. For example, the word *enter* indicates that you type a series of characters on the command line and press the RETURN key. The instruction

Enter `ls`

means "Type `ls` and press the RETURN key."

Here is a list of common terms and the corresponding actions you take.

Term	*Action*
Click	Press and then immediately release the mouse button.
Drag	Position the mouse pointer, press and hold down the mouse button while moving the mouse, and then release the mouse button.
Choose	Activate a command in a menu. To choose a command from a pull-down menu, position the pointer on the menu title and hold down the mouse button. While holding down the mouse button, drag down through the menu until the command you want is highlighted. Then release the mouse button.

Select	Highlight a selectable object by positioning the mouse pointer on the object and clicking.
Type	Type a series of characters *without* pressing the RETURN key.
Enter	Type the series of characters indicated and press the RETURN key.

The `Courier` *font*

Throughout A/UX guides, words that appear on the screen or that you must type exactly as shown are in the `Courier` font.

For example, suppose you see this instruction:

Type `date` on the command line and press RETURN.

The word `date` is in the `Courier` font to indicate that you must type it.

Suppose you then read this explanation:

After you press RETURN, information such as this appears on the screen:

```
Tues Oct 17 17:04:00 PDT 1989
```

In this case, `Courier` is used to represent the text that appears on the screen.

All A/UX manual page names are also shown in the `Courier` font. For example, the entry `ls`(1) indicates that `ls` is the name of a manual page in an A/UX reference manual. See "Manual Page Reference Notation" later in this preface for more information on the A/UX command reference manuals.

Font styles

Italics are used to indicate that a word or set of words is a placeholder for part of a command. For example,

`cat` *filename*

tells you that *filename* is a placeholder for the name of a file you want to display. For example, if you wanted to display the contents of a file named `Elvis`, you would type the word `Elvis` in place of *filename*. In other words, you would enter

```
cat Elvis
```

Terms that appear in **boldface** are defined in the glossary. Boldface is also used for steps in a series of instructions.

A/UX command syntax

A/UX commands follow a specific command syntax. A typical A/UX command gives the command name first, followed by options and arguments. For example, here is the syntax for the `wc` command:

`wc [-l] [-w] [-c]` [*filename*]...

In this example, `wc` is the command, `-l`, `-w`, and `-c` are options and *filename* is an argument. Brackets ([ ]) enclose elements that are not necessary for the command to execute. The ellipsis (...) indicates that you can specify more than one argument. Brackets and ellipses are *not* to be typed. Also, note that each command element is separated from the next element by a space.

The following table gives more information about the elements of an A/UX command.

Element	*Description*
command	The command name.
option	A character or group of characters that modifies the command. Most options have the form *–option,* where *option* is a letter representing an option. Most commands have one or more options.
argument	A modification or specification of a command, usually a filename or symbols representing one or more filenames.
[ ]	Brackets used to enclose an optional item—that is, an item that is not essential for execution of the command.
...	Ellipses are used to indicate that you can enter more than one argument.

For example, the `wc` command is used to count lines, words, and characters in a file. Thus, you can enter

```
wc -w Priscilla
```

In this command line, `-w` is the option that instructs the command to count all of the words in the file, and the argument `Priscilla` is the file to be searched.

Manual page reference notation

A/UX Command Reference, A/UX Programmer's Reference, A/UX System Administrator's Reference, X11 Command Reference for A/UX, and *X11 Programmer's Reference for A/UX* contain descriptions of commands, subroutines, and other related information. Such descriptions are known as *manual pages* (often shortened to *man pages*). Manual pages are organized within these references by section numbers. The standard A/UX cross-reference notation is

command (section)

where *command* is the name of the command, file, or other facility; and *section* is the number of the section in which the item resides.

- Items followed by section numbers (1M) and (8) are described in *A/UX System Administrator's Reference.*
- Items followed by section numbers (1) and (6) are described in *A/UX Command Reference.*
- Items followed by section numbers (2), (3), (4), and (5) are described in *A/UX Programmer's Reference.*
- Items followed by section number (1X) are described in *X11 Command Reference for A/UX.*
- Items followed by section numbers (3X) and (3Xt) are described in *X11 Programmer's Reference for A/UX.*

For example

`cat(1)`

refers to the command `cat`, which is described in Section 1 of the *A/UX Command Reference*.

You can display manual pages on the screen by using the `man` command. For example, you could enter the command

```
man cat
```

to display the manual page for the `cat` command, including its description, syntax, options, and other pertinent information. To exit a manual page, press the SPACE BAR until you see a command prompt, or type `q` at any time to return immediately to your command prompt.

For more information

To find out where you need to go for more information about how to use A/UX, see *Road Map to A/UX*. This guide contains descriptions of each A/UX guide and ordering information for all the guides in the A/UX documentation suite.

Appendix A: Troubleshooting

This appendix describes some potential network problems you might encounter and gives steps for diagnosing and fixing them. The information in this appendix is arranged by the task. For example, if you receive an error message while you're using `rcp` to copy files, you would look in the `rcp` section. This chapter also tells you how to correct entries if you make a mistake when you're using `newconfig` to connect to a network.

`rlogin` messages

This section lists some common problems you may encounter when using `rlogin` and suggests solutions to them.

`Login incorrect`

First check to make sure you entered the information correctly. If the information is correct, check with your network administrator about your login name and password on the remote host or in the NIS database.

`Connection timed out`

The remote computer is either down or very busy. If the remote computer is down, you'll have to wait until it's up again, or try another computer. The `ruptime` command doesn't report that a computer is down unless there has been no response for five minutes.

rhost`: unknown host`

The local computer doesn't know the remote system's address. You may have entered the remote host name incorrectly on the command line. If the host name is correct, your network administrator needs to add the address to a local system file.

rhost`: host name for your address unknown`

The remote computer does not know your local system's name and address. You may have entered the remote host name incorrectly on the command line. If the host name is correct, your network administrator needs to correct this situation by modifying system files on the remote computer.

When you've logged in to the remote computer through `rlogin`, the shell environment of your remote account may be different from your local environment (the prompt may be different). This can cause some unexpected responses; for example, `vi` could treat your terminal as a dumb terminal. (See the Note on page 5-11.)

If certain files are missing that you think should be on the remote computer, contact your network administrator.

If you think the problem is with your personal files rather than the file system, open another CommandShell window and run `rcp`. For example, if you are using the Bourne or Korn shells, copy your `.profile` file to the remote system using the command

`rcp .profile` *rhost*`:`

After you have done this, try `rlogin` again.

`telnet` messages

The error messages described for the `rlogin` command also apply to `telnet`. Your local system administrator can probably correct the situation if you see the message

rhost`: unknown host`

The local computer does not know the remote computer's address. You may have entered the remote host name incorrectly on the command line. If the host name is correct, your network administrator needs to add the address to a local system file.

In the following cases, you may need to contact the administrator of the remote system.

`Connection refused`

The remote system detects a permissions violation or that the remote system is up but the remote server is not running.

`Connection timed out`

The remote system is either down or very busy.

rhost`: host name for your address unknown`

The remote computer does not know the local computer's address. You may have entered the remote host name incorrectly on the command line. If the host name is correct, your network administrator needs to correct this situation by modifying system files on the remote computer.

`remsh` messages

The remote system must be able to determine that your local and remote accounts are equivalent, or it won't allow you to use `remsh`. For further information, see "Working With Network Permissions" in Chapter 5.

The most common mistake on the `remsh` command line is incorrect quoting of command arguments. Quoting refers to the use of quotation marks or other characters to indicate that a name or character should be taken literally rather than interpreted. See "Quoting Mechanisms" and "`remsh` Command Line Considerations" in Appendix C, "More About `remsh`," and the appropriate chapter in *A/UX Shells and Shell Programming* for details.

`No match`

The local shell may be trying to intrepret shell metacharacters such as the asterisk (*). (A metacharacter is a character interpreted by the shell in a special way.) Try enclosing the entire command in single quotation marks.

`Command not found`

This message occurs if you use a specialized command (such as a system shell script) that exists on one system and not others. This can also occur if the file system is set up differently on the remote system and the path to the command is interpreted locally. In this case, try quoting the argument or typing out the full path.

`No such file or directory`

The permissions set by the system administrator are not as they should be. See "Working With Network Permissions" in Chapter 5 for more information.

rcp messages

The `rcp` command does not prompt for passwords, so the remote system must be able to determine that your local and remote accounts are equivalent. Otherwise, it won't allow you to use `rcp`. See "Working With Network Permissions" in Chapter 5 for more information.

In some cases, more than one user on a computer may share the same user ID. If that's the case on either system, the *path* argument may be interpreted relative to a directory that's not your home directory. If the command

`rcp` *rhost*`:` *source-file dest-file*

prints an error message such as

```
rcp: source.dir: No such file or directory
```

check first for typos. If the entry is correct, try the same command again, specifying your remote login name with the format

`rcp` *rhost*`:~`*username*`/` *source-file dest-file*

replacing *username* with your login name on the remote computer. If this command solves the problem, you should specify your login directory in every `rcp` command.

If you have repeated problems, check with your network administrator.

When you're using the `-r` option to `rcp`, remember that it creates a directory *below* your current directory. It's a common mistake to create a destination directory and move into it before using the `rcp -r` command.

Correcting `newconfig` errors

When you're setting up your system to use the TCP/IP network, the system prompts you to enter pieces of information. If you make a mistake, but you don't discover the mistake until after `newconfig` is finished, remove the files indicated below and run `newconfig` again.

- **If you entered the wrong host name or domain name, enter this command:**

  ```
  rm /etc/HOSTNAME
  ```

- **If you entered the wrong IP address or netmask, enter this command:**

  ```
  rm /etc/NETADDRS
  ```

- **Run the `newconfig` command again.**

  See the procedure in Chapter 2.

Appendix B: More About Mail

This appendix gives you more information about the `mailx` command, such as using the tilde escape, working with mail messages, customizing your mail environment, and sending mail to files. These features are more advanced than the features covered in Chapter 4, and they are useful when you're comfortable with the standard mail capabilities and you'd like to learn more.

Using the tilde escape

While you're typing a `mail` message, you can change to command mode by typing a tilde (~). This is called a **tilde escape**. It allows you to edit a message or a mail message field, append a file to a message, go to the shell, or manipulate messages in a variety of ways. A tilde escape consists of a tilde (~) at the beginning of a line followed by a single-character command.

- **To change to command mode when you're typing a mail message, enter this command at the beginning of a new line:**

  ~command

Displaying a summary of tilde escapes

It is sometimes useful to display a summary of tilde escapes as a reminder.

- **To display a summary of tilde escapes, enter this command:**

  `~?`

Displaying the message you are currently entering

If you want to review the message you're currently entering and remind yourself who the recipients are, you can display the message on the screen.

- **To display the message, enter this command:**

  `~p`

  A line of dashes appears followed by the login names of the recipients of your message and the text of the message.

◆ **Note** Remember, the tilde command must appear at the beginning of a new line ◆

Canceling a tilde escape

You can cancel any tilde escape command.

■ **To stop any tilde escape, do the following:**

Press CONTROL-C.

▲ **Warning** Do not press CONTROL-C twice or you'll cancel the letter. ▲

Editing a mail message

You can edit a mail message that you are writing by calling up an editor.

1 **To edit your message, enter this command at the beginning of a line:**

```
~v
```

2 **Edit the message, save it, and quit the editor. `mail` displays this line:**

```
(continue)
```

3 **To see the modified message and continue typing at the end of the modified message, enter this command:**

```
~p
```

4 **Continue typing your message, or end the message by entering this:**

CONTROL-D.

To use an editor other than the default `ed` editor, set the `EDITOR` variable. For example, to use the `vi` editor, include the following line in your `.mailrc` file:

```
set EDITOR=/usr/bin/vi
```

Including a file with a mail message

In some cases it may be convenient for you to send an existing file along with your mail message.

- **To append a file to the message you are currently entering, enter this command:**

  `~r` *filename*

  The filename may contain shell metacharacters like `*` and `?`. The system returns a message if the file doesn't exist or can't be read. When `mail` appends the file, it displays the number of lines and characters appended. You will see a message that looks something like this:

```
~r filename

"filename" 13/484
```

  The number before the slash indicates the number of lines and the number after the slash indicates the number of characters.

  After this, you may continue adding text to your message.

Saving a current message

You can save a message into a file as you enter it.

- **To save the message you are currently entering in a file, enter this command:**

  `~w` *filename*

  You may use shell metacharacters in the filename. As in `~r`, `mail` prints the number of lines and characters written to the file, after which you may continue appending text to your message.

Including a message with a message

You can add a message you received to the message you are currently entering.

- **To append the message that's marked as the current message to the message you are entering, enter this command:**

  `~m`

  The whole message appears, including the header.

- **To append a specific message to the message you are entering, enter this command:**

  `~m` *message-number*

  You may specify any undeleted message or list of messages. For example,

  ```
  ~m 4
  ```

  adds message 4 to the message you are currently entering, shifted to the right by one tab stop. If you don't want the message shifted to the right by a tab stop, use the `~f` command, which is the usual way to forward a message.

Adding recipients to a message

After you've started entering a message, you can add recipients to the message.

- **To add recipients to the message you are currently entering, enter this command:**

  `~t` *name1 name2*

  The recipients you originally addressed will still receive the message.

  ◆ **Note** You cannot remove someone from the recipient list with `~t` ◆

Changing the subject line of a message

You can change the wording of the subject line so that it no longer reads "Subject."

- **To change the wording of the subject line, enter this command:**

  `~s` *arbitrary-string-of-text*

  This creates a new subject line with whatever text you type (which can include spaces).

- **To see the new subject line, enter this command:**

  `~p`

Sending message copies

You can list certain people as carbon copy recipients (`Cc:`) or as blind carbon copy recipients (`Bcc:`) rather than as direct recipients. `Cc:` recipients are all listed on the message, and all other recipients can read who the `Cc:` recipients are; `Bcc:` recipients are listed only on the message they receive and not on the other messages sent to `Cc:` recipients.

- **To list certain people as carbon copy recipients, enter this command:**

  `~c` *name1 name2*

  Those named will be added to the `Cc:` list.

- **To list certain people as blind carbon copy recipients, enter this command:**

  `~b` *name1 name2*

  Those named will be added to the `Bcc:` list.

Editing mail message fields

You can edit these four fields (as opposed to only adding to them): `To:`, `Subject:`, `Cc:`, and `Bcc:`. You may delete characters or add characters to any field. To edit the message fields, follow these steps:

1 **To display the first field, enter this command:**

```
~h
```

The first field and the current list of recipients appears and leaves the cursor at the end of the line.

2 **Edit the `To:` field.**

Characters you type are appended to the end of the current list of recipients. Now you're ready to advance to the next field.

3 **Press RETURN**

The cursor is in the `Subject:` field.

4 **Edit the `Subject:` field the same way you edited the `To:` field.**

5 **Press RETURN again.**

Now you can edit the `Cc:` field or press **RETURN** again to edit the `Bcc:` field.

6 **Press RETURN again and continue appending text to the end of your message.**

Escaping to the shell

Sometimes you may want to temporarily escape to the shell.

- **To escape temporarily to the shell, enter this command:**

  `~!` *command*

  This executes *command* and returns you to `mail` without changing your message.

- **To filter your message through a shell command, enter this command on a new line:**

  `~|` *command*

  This pipes your message through the command and uses the output as the new text of your message. If the command produces no output (for example, due to an error), `mail` retains the old version of your message. A frequently used filter is the command `fmt`, which adjusts the length of the line.

- **To see the results of the above procedure, enter this command on a new line:**

  `~p`

Accessing the shell from within mail

To execute a shell command without leaving `mail`, type the command preceded by an exclamation point (just as in the text editor), as in the example below:

- **To display the current date without leaving `mail`, enter this command:**

  ```
  !date
  ```

Escaping to the `mail` command mode

You can escape temporarily to the `mail` command mode.

- **To escape temporarily to the `mail` command mode, enter this command on a new line:**

  ~ : *mail-command*

Displaying the message to which you're replying

You can display the message you're replying to and its entire header.

- **To display the entire current message, including one that you just started, enter this command:**

  ```
  ~:t
  ```

Using the tilde character with the tilde escape

If you're typing a message containing a line beginning with a tilde, you must type the tilde character three times to avoid its being interpreted as an escape. For example,

```
~~~This line begins with a tilde.
```

incorporates the line in your message

```
~This line begins with a tilde.
```

Changing the escape character

If your terminal does not have a tilde (~) key, you need to change the escape character. `mail` allows you to change the escape character with the `escape` option. For example, you could change the escape character to a right bracket:

- **To change the tilde escape to a right bracket, enter this command:**

```
set escape=]
```

Now when you want to escape to command mode, you would enter the following:

`]` *command*

Changing the escape character means that the tilde (~) no longer has any special meaning to `mail`. If you want this to be true every time you use mail, put this command in your `.mailrc` file. See `mailx`(1).

◆ **Note** The `.mailrc` file is the place where you can customize the way mail is presented to you. You can put many commands in it, such as your alias lists. For more information about the `.mailrc` file, see "Using the `set` Command" later in this appendix. ◆

Working with mail messages

You may want to do more with your mail messages than just read them. This section tells you how to display several at once, how to reply to a message, send mail when you're reading a message, read mail from a different file, and access the shell from within mail.

Displaying several messages at once

Many `mail` commands accept a message list as an argument. A message list is a list of message numbers, ranges, and names, separated by spaces or tabs. A message list allows you to address a command to many messages at once. Commands that use message lists ignore deleted messages, except for the `undelete` command, which applies only to deleted messages.

A **message number** may be either the number of the message (listed in the second field) or one of the following special characters:

`^` (first message)

`.` (current message; when you enter `mail`, the current message is the first message)

`$` (last message)

A **message range** is two message numbers separated by a hyphen (`-`). The examples below explain the various ways to display messages.

- **To display the first four messages, enter this command:**

```
type 1-4
```

- **To display all the messages from the current to the last, enter this command:**

```
type .-$
```

A **message name** is the login name of a user who sent a message. If you specify a message list with only user names, `mail` selects every message sent by one of those users. If you specify both a range and a message name, `mail` selects every message sent by that user *within that message range.*

- **To display every message sent to you by `root`, enter this command:**

```
type root
```

- **To display all undeleted messages, enter this command:**

```
type *
```

The asterisk character (*) specifies every message.

- **To delete all messages, enter this command:**

```
delete *
```

- **To undelete all deleted messages, enter this command:**

```
undelete *
```

The slash character (/) searches for a word in the subject line. For example, if you want to display the headers of all messages containing the word PASCAL, enter

```
from /pascal
```

Note that subject searching ignores uppercase/lowercase differences.

Replying to a message

The `reply` command allows you to reply to a message. You may abbreviate this command to `r`. When you use `reply`, `mail` begins a message addressed to the sender. When you finish your message, press CONTROL-D at the beginning of a line to end it. The message `EOT` appears followed by the `mail` prompt. It's then ready to accept another command.

If you had just read a message, you could reply by typing:

```
reply
```

`mail` responds with the following:

```
To: root
Subject: Re: Tuition fees
```

`mail` then waits for you to enter your reply. Note that it copies the subject header from the original message so that correspondence about a particular matter retains the same subject heading.

If the original message was mailed to several people (listed in the `To:` header), `mail` sends your reply to the same people. Similarly, if the original message contains a `Cc:` field or a `Bcc:` field described under "Sending Message Copies," `mail` sends your reply to those users, too. `mail` will not send the message to you, even if you appear in the `To:` or `Cc:` field, unless you ask to be included by setting the `metoo` option in your `.mailrc` file.

The `reply` command is useful for sustaining extended conversations over the message system, with other "listening" users receiving copies of the conversation.

Sometimes you receive a message sent to several people and want to reply only to the person who sent it. The `Reply` command (with a capital `R`) sends your reply only to the message sender.

Mailing from within mail

You can send mail while you are reading your mail by using the `mail` command. The form of this command is

`mail` *username*

The examples below explain how to send mail while reading mail.

1 **To send a message to `jeanne`, enter this command:**

```
mail jeanne
```

You can abbreviate the `mail` command to `m`.

2 **Then, after entering a subject (if you wish), type your message:**

```
This is to confirm next Friday's 4 pm meeting.
```

3 **To send the message, start a new line and press CONTROL-D.**

Forwarding a message

You can forward the current message or forward a specific message. When you display the list of messages, the current message appears in the message list preceded by the right angle bracket (>)

```
>current message
```

- **To forward the current message, enter this command:**

  ```
  ~f
  ```

- **To forward a specific message, enter this command:**

  ```
  ~f message number
  ```

Reading mail from a different file

To read mail from a file other than your system mailbox, use the `-f` option when you start `mail`, as in the example below:

- **To read the mail in your `mbox` file in your home directory, enter this command:**

  ```
  mail -f mbox
  ```

  By default, `mail -f` by itself reads your `mbox` file.

Customizing mail

This section describes how to customize your `mail` environment by changing the variables. You can change the variables to be active during an individual `mail` session or you can change them in the `.mailrc` file so that they're set every time you use `mail`.

Using the `set` command

You may tailor many features of `mailx` with the `set` command. See `mailx`(1) for a list of all the `mailx` variables you may set or unset. A variable alters the behavior of the `mailx` program. For example, when `askcc` is set, `mailx` prompts you for a `Cc:` If you change the `askcc` variable, mail will no longer prompt you for a `Cc:`.

- **To see what variables are set, enter this command:**

```
set
```

Toggle variables are either on or off. The `set` command turns a toggle variable on. The `unset` command turns it off. For example, the `askcc` variable tells `mail` to prompt you for a `Cc:` for your messages.

- **To change the `askcc` variable, enter this command:**

```
set askcc
```

Now you will not be prompted for a `Cc:` list.

Another toggle variable is `hold`. The `hold` variable keeps messages in your system mailbox after you read them instead of moving them to the `mbox` file in your home directory.

Some `mail` variables, however, are on by default.

- **To turn off some `mail` variables, enter this command:**

```
set novariable
```

For example, the `save` variable, which saves canceled messages in the `dead.letter` file in your home directory, is on by default. Entering

```
set nosave
```

would turn it off.

Not all `mail` variables can be toggled on or off. These variables must be given a value. For example, the shell **variable** tells `mail` which shell you want to use. To tell `mail` to use the C shell, enter

```
set SHELL=/bin/csh
```

No spaces are allowed in `SHELL=/bin/csh`. For more information about shells, refer to *A/UX Shells and Shell Programming.*

One helpful `mail` variable is `crt`. The `crt` variable paginates messages that are too long to fit on your screen. To set the `crt` variable, tell `mail` how many lines your terminal has. For example, usually you would enter

```
set crt=24
```

This sends messages longer than 24 lines through the `pg` program. `pg` displays a screenful of information, then displays a colon prompt (:).

Press RETURN to see the next screenful. See `pg`(1) in *A/UX Command Reference* for more information. If you prefer, you could use the `more` command to display a screenful of information. To use `more` instead of `pg`, you would enter

```
set PAGER=/bin/more
```

Setting up a permanent mail environment

The `set` and `alias` commands let you make changes in how you use `mail` during an individual session. Each time you leave `mail`, however, these changes are lost. To make an `alias` or `set` command part of your permanent `mail` environment, use any A/UX text editor to include it in the `.mailrc` file in your home directory.

Your `.mailrc` file might look like this:

```
set asksub nosave SHELL=/bin/csh

alias project john kathy allen vicki
```

In the example, `nosave` turns off the `save` variable, which is on by default. The example also demonstrates that you can set many `mail` variables in the same `set` command.

You can continue long lists in the `alias` command onto the next line by starting the next line with a space or tab.

You can also put comments in `.mailrc` by starting them with a `#` character.

Your system administrator can also set `mail` variables and aliases for everyone on your system. The system-wide `mail` environment instructions are in the file `/usr/lib/mailx/mailx.rc`

Sending mail to files

You might want to send mail to a file to maintain a journal or keep a record of mail sent to a group of users. A recipient name with a slash (/) in it or beginning with a plus sign (+) is assumed to be the pathname of a file. If the file already exists, the message is appended to the end of the file. To refer to a file in your current directory, precede the name with a period followed by a slash (./). The examples below show how the commands would look:

- **To send mail to a file called `memo` in the current directory, enter this command:**

```
mail ./memo
```

A name beginning with a `+` is assumed to be a mail folder name. When you create an alias for members of a team, you could add the name of a folder so that all mail sent to the members of a team called `project` would also be saved in a folder called `project`. Following the `alias` example shown in Chapter 4, you would use this command:

```
alias project lori paul mike linda +project
```

Then all mail sent to the alias called `project` would be saved in the folder called `project` in your home directory and be sent to the members of the project.

- **To read the contents of the special folder you created, enter this command:**

```
mail -f +project
```

Sending a file through mail

Sometimes you may want to send a file as a mail message.

- **To send a file as a mail message, enter this command:**

`mail` *username* < *filename*

▲ **Warning** In the command above, you must type a left angle bracket (<). If you accidently type a right angle bracket (>), the file you're trying to send will be overwritten. ▲

Appendix C: More About `remsh`

This appendix tells you how to use `remsh` to connect to remote devices, describes the implications of using shell commands on the network, and offers some `remsh` options.

Using remote devices

Many networks have printers attached to a system that may be accessed by all users across the network. The following examples show how to send a file to a printer attached to a remote system named `sierra`. Note that pipes (`|`) are interpreted by the local computer unless they're quoted.

- **To send a file called `feet` across the network to be printed on the remote line printer, enter this command:**

```
cat feet | remsh sierra lpr
```

The next example shows how to set up the command to format the file with `nroff` before printing the file. You can format the file on the local or remote system.

- **To format `feet` locally and print it remotely, enter this command:**

```
nroff feet | remsh sierra lpr
```

- **To format and print `feet` on the remote system, enter this command:**

```
cat feet | remsh sierra nroff \| lpr
```

Because the second pipe character in the command is preceded by a backslash (`\`), that character (and the command that follows it) is interpreted on the remote computer rather than the local shell. Thus, the file called `feet` is printed on the line printer, which is attached to the remote computer named `sierra`.

If the second pipe character is not preceded by a backslash

```
cat feet | remsh sierra nroff | lpr
```

it's interpreted on the local computer. In that case, the `nroff` process runs on the remote computer, but the output is sent to a local spooler.

The `remsh` command also allows you to run pipelined processes on a number of remote computers. For example,

```
cat feet | remsh sierra nroff | remsh yosemite lpr
```

In a pipeline, you can quote the pipes between commands or not, with no noticeable effect. In the command line in the previous example, the second pipe is not quoted, so it is interpreted on the local system. Thus, the output of the remote `nroff` command is returned to the local system where it is again sent across the network to another system. A similar command

```
cat feet | remsh sierra nroff \| remsh yosemite lpr
```

uses a backslash to prevent the shell from interpreting the second pipe character, causing it to be interpreted on the remote system. Thus, the output of the `nroff` command is sent to another remote computer without first returning across the network to the local system. See "Quoting Mechanisms" later in this appendix for more information.

Ending the `remsh` session using the `kill` command

You can usually terminate a remote background command using the `kill` command with the **process ID** on the local computer. If the `kill` command works only on the local `remsh` command (allowing the remote process to continue running), find out the process ID of the remote process by running this command:

`remsh` *rhost* `ps -aef`

Then run this command:

`remsh` *rhost* `kill -9` *process-ID*

Local and remote shell topics

This section discusses the implications of using shell commands across the network. This is particularly important to the `remsh` command when you're specifying which computer runs a command, which devices are used, where output files are created, and so on.

Quoting mechanisms

Various shells use symbols for special purposes, so if you want an interpretation of the symbol different from the one dictated by the shell you are using, you must quote the argument; that is, you must use quotation marks or the backslash to prevent the shell from misinterpreting the metacharacter. There are three ways to quote special characters across the network:

- Precede each metacharacter with a backslash (`\`). A backslash preceding a metacharacter prevents the local shell (the shell on the local computer) from interpreting the metacharacter.
- Enclose the metacharacters in single quotation marks (`' '`). Single quotes prevent the local shell from interpreting any metacharacters in the enclosed string.
- Enclose the metacharacters in double quotation marks (`" "`). Within double quotes, variable substitution and command substitution occur in the local shell, but filename expansion does not.

The difference between using single and double quotation marks in a network command applies to variable substitution (for example, `$j`) and command substitution (`'command'`). When you use double quotes, both of these substitutions occur locally. (See *A/UX Shells and Shell Programming* for a description of these shell capabilities.)

A backslash before each metacharacter causes the local shell to strip off the backslash and pass the character to the remote shell. For example, in the command

```
remsh sierra cat jelly \> jam
```

the **redirection** (>) symbol is interpreted by the remote shell, and the file `jam` is created on the remote computer `sierra`. The backgrounding is done locally.

In the command

```
remsh sierra cat jelly > bread
```

the redirection symbol is interpreted by the local shell, and the file `bread` is created on the local computer. This has the same effect as an `rcp` command.

Single quotation marks can be placed around the metacharacters, command arguments, or parts of the command arguments. All information within the single quotes is passed intact to the remote shell and interpreted there. All of the following commands have the same effect:

```
remsh sierra cat jelly '>' jam

remsh sierra cat 'jelly > jam'

remsh sierra 'cat jelly > jam'
```

The remote shell interprets the filenames and redirection symbol, creating a new file on the remote host.

Double quotes around the command arguments (or any part of them) are usually used when you want to interpret a shell variable on the local system, or when you want command substitution to occur on the local system. For example, if you have a shell variable set in your local shell

```
j=22
```

and a variable set in the remote shell

```
j=15
```

you can print the remote value using single quotes as follows:

```
remsh sierra 'echo $j'

15
```

To print the local value on the remote system, use double quotes:

```
remsh sierra "echo $j"

22
```

In the latter case, the remote system prints the value of the variable, which has already been interpreted by the local shell. See "Shell Variables" later in this appendix and *A/UX Shells and Shell Programming* for more information.

`remsh` command line considerations

The following subsections discuss the various `remsh` command line considerations: multiple commands, pipelines, filename expansion, output redirection, history substitution, aliases, the `grep` command, and shell variables.

Multiple commands

You can use the `remsh` command to run multiple commands by quoting the semicolon (`;`) that separates the different commands on the same command line. To run the `date` command followed by the `hostname` command (which returns the name of the system) on a remote host named `sierra`, enter

```
remsh sierra date \; hostname
```

You get a response such as

```
Mon Aug 12 10:55:37 PDT 1985
sierra
```

If you don't quote the semicolon by preceding it with a backslash, the `remsh` command will simply terminate after executing a single command. The semicolon character (and the command that follows it) will then be interpreted and executed on the local computer.

If the name of your current system is `lcl`, the command

```
remsh sierra date ; hostname
```

gives a response such as

```
Mon Aug 12 10:57:06 PDT 1985
lcl
```

Pipelines

You may use the `remsh` command to run pipelined commands on a remote system by quoting the pipe character (`|`) that separates the multiple processes on the same command line. If you use the backslash to quote the pipe character, you may run different parts of the command line on the local or remote computer, as you see fit.

Suppose you have a remote file `jelly` that needs to be formatted using `neqn` and `nroff` before it is printed. The command

```
remsh sierra neqn jelly | nroff | lpr
```

causes the `neqn` process to run on the remote computer. The output of that process is passed through the pipe. Because the pipe is not quoted, it's interpreted by the *local* shell, so `nroff` runs on the local computer, and the output is passed to the local line printer. If the command were in either of the following formats,

```
remsh sierra neqn jelly \| nroff \| lpr
```

or

```
remsh sierra 'neqn jelly | nroff | lpr'
```

all processes would run on the remote computer, and the remote line printer would be used. Quoting the pipes individually allows a range of flexibility in using different resources, but in this case the same effect is obtained when you quote the whole command argument or the individual metacharacters.

For example, if you need to format a local file `bread` on the remote computer and direct the output to a file `frmt.bread` on the local computer, use the command

```
cat bread | remsh sierra troff > frmt.bread
```

Neither of the metacharacters needs to be interpreted on the remote shell in this format.

The `remsh` command may also be used in pipelines. For example,

```
cat bread | remsh sierra nroff | remsh yosemite lpr
```

In this case, the pipes may be quoted or not quoted, with the same effect.

Filename expansion

The shell metacharacters for filename expansion are interpreted by your local shell unless they are quoted. For example, the command

```
remsh sierra ls -l 1*
```

causes the local shell to search your current directory for any filename beginning with `1`. If it finds one or more files beginning with `1` in your local directory, it expands the

filenames and sends these across the network. If the local shell doesn't find a filename beginning with `1`, it prints the message

```
No match.
```

If the remote shell received an expanded name beginning with `1` and doesn't have a file by that name, it prints the same message.

Unless you have two identical files (one local and one remote) that begin with the character `1`, the above command will not produce the effect you want. To run `ls -l` on *remote* files beginning with `1`, you must quote the filename expansion metacharacter with a backslash, single quotes, or double quotes. For example,

```
remsh sierra ls -l '1*'
```

or

```
remsh sierra ls -l "1*"
```

or

```
remsh sierra ls -l 1\*
```

causes the filename expansion to take place in the login directory of your remote account. Assuming that you have one or more filenames beginning with `1` there, the output of the remote process appears on your screen. For example,

```
-rw------- 1 joe    239 Aug 23 11:33 10.check
-rw-r----- 1 joe    555 Sep 3  12:21 15.ftp
-rw-r----- 1 joe  36467 Nov 15 12:36 1234
```

Output redirection

The shell metacharacters for redirecting standard output are interpreted by your local shell unless they are quoted. The command

```
remsh sierra cat '1*' >> 1.all
```

concatenates the files beginning with `1` in your login directory on the remote computer into a file named `1.all` on the *local* computer. That is, the command creates a file called `1.all` that contains a copy of every file that begins with `1`.

If the output redirection symbols are quoted, for example,

```
remsh sierra cat '1* >> 1.all'
```

all files beginning with `1` in your login directory on the remote computer are concatenated into a file named `1.all` on the *remote* host.

History substitution

History substitution, a way of correcting errors at the command line or repeating a long line, does not work across the network in the C shell. The `remsh` command does not use the shell in which you have a history of your commands but invokes a new shell to run its command. The command

```
remsh sierra '\!c'
```

does not perform history substitution on either computer. The system prints the message

```
Event not found.
```

For the same reason, history substitution in the Korn shell does not work across the network.

Aliases

Aliases that you have set on either system may be used with the `remsh` command if they are set using the format

`alias` *alias-name command(s)*

If, however, you have aliases set only in interactive shells, for example, your `.cshrc` contains

```
if ($?prompt) then
        alias ls ls -f
        .
        .
        .
endif
```

they cannot be used in a `remsh` command. The remote shell is not interactive, so it doesn't set the alias. Also, anything that is done in your `.login` file (if you are using the C shell) or your `.profile` file (if you are using the Korn shell) will not be done, because the shell that's invoked by the `remsh` command is not a login shell.

The `grep` command

When you use `grep` with the `remsh` command to search files on the remote system for a regular expression, the regular expression has to be quoted *twice*. The argument to `grep` is always quoted once to prevent the local shell from interpreting the metacharacters contained in the regular expression; the second quotes are required to prevent their interpretation by the remote shell. You may use single quotes and a backslash:

`remsh` *rhost* `grep 'abc\[0-9\]*'` *remote-file*

or single and double quotes:

`remsh` *rhost grep* `'"abc[0-9]*"' remote-file`

(Note that the latter method does not work reliably for embedded dollar signs.) Either command causes the `grep` command to receive the argument and search *remote-file* for the regular expression you intended.

Shell variables

If you create a shell variable `i` and set its value on the local system to the pathname

```
set i =/v2/sys/dist/sys
```

and on the remote computer you have a variable `i` set to another value,

```
set i =/v/test/sys
```

the following command uses the local value:

`remsh` *rhost* `cat "$i | remote-program"`

The shell variable `$i` will be substituted locally within double quotation marks as `/v/sys/dist/sys`. That value is then passed with the rest of the command arguments to the remote shell.

If you want the value substituted on the remote system instead, use single quotation marks instead of double quotation marks and set the variable in the `.profile` file on the remote system to force the `.profile` file on the remote system to be executed first:

```
remsh rhost '. ./.profile; $i | remote-program'
```

Because `remsh` does not automatically run the `.profile` file on the remote system, you have to force its execution. Then the string `$i` is passed to the remote shell, which substitutes the value it has recorded for this variable.

Of course, the pathname that is substituted for `$i` has to be accessible on the remote computer for either of these `remsh` commands to have any usable effect.

Shell scripts

A **shell script** contains the same commands that you might run from a CommandShell window plus shell commands that you can use to control the flow of command execution, which makes the shell script an interpreted programming language. A shell script saves you time when you have a long series of commands you need to run. You enter those commands into a text file so that you can run the file instead of the commands.

If the shell script on the remote computer expects to receive arguments, these arguments can be passed across the network through `remsh`. The command

```
remsh rhost myscript A B C
```

provides arguments `A`, `B`, and `C` as values to the variables `$1`, `$2`, and `$3` (or `$argv[1]`, `$argv[2]`, and `$argv[3]`) to the shell script `myscript` on the remote computer.

There are three things to consider when you pass arguments to a remote shell script in this way:

- You must quote any metacharacters in the arguments.
- You may define variables within the arguments locally, and in that case they may not be available, or may differ, on the remote computer.
- Environmental variables (which could affect the way the shell script runs) may differ on the remote computer.

`remsh` options

You can specify the following options on the `remsh` command line:

`-n` Redirect standard input from `/dev/null`.

If you're using the `remsh` command in the background, the `-n` option redirects the standard input of the remote command from `/dev/null` (instead of your terminal) as follows:

`remsh` *rhost* `-n` *command* `&`

If a remote command tries to read from (remote) standard input while it's executing, it could "hang" indefinitely without terminating. If this happens to a remote background command, you can try using the `-n` option on the `remsh` command line the next time you use that command.

Appendix D: More About `ftp`

This appendix gives you more information about `ftp`, such as how to set conditions and how to use `ftp` options and commands.

Setting conditions

There are several `ftp` toggle commands that turn a condition on or off, depending on the previous state of the condition. See the section "Toggle Commands" later in this appendix for a complete list. These commands can only be entered from the `ftp>` prompt.

For example, the command

```
ftp>bell
```

causes a bell to sound or not to sound after each file transfer, depending on the previous state.

The `verbose` command is on by default. The command

```
ftp>verbose
```

turns explanatory messages off.

To find the current state of all toggle conditions and other status information, use the command

```
ftp>status
```

Using `ftp` options

You may specify the following options on the command line when you invoke `ftp`.

`-v` Force `ftp` to show all responses from the remote server, as well as report on data transfer statistics.

`-n` Restrain `ftp` from autologin so that you are connected to a host without being prompted to log in (see `user` in the following section, "`ftp` Commands").

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

`-g` Disable filename expansion; that is, treat all filenames and pathnames literally, without interpreting metacharacters as such.

ftp commands

The following commands are recognized by `ftp` and are entered at the prompt

```
ftp>
```

Command arguments that have embedded spaces may be quoted with double quotation marks (`""`).

`!` [*command* [*args*]]
Invokes an interactive shell on the local computer (when a shell is interactive, when you enter commands, you see a response). If there are arguments, the first is taken to be a command to run directly, with the rest of the arguments as the command's arguments.

`$` *macro-name* [*args*]
Runs the macro *macro-name* that was defined with the `macdef` command. Arguments are passed to the macro without metacharacter expansion.

`account` [*passwd*]
Supplies a supplemental password required by a remote system for access to resources once a login has been completed successfully. If no argument is included, the user will be prompted for an account password in a non-echoing input mode.

`append` *local-file* [*remote-file*]
Appends a local file to a file on the remote computer. If *remote-file* is left unspecified, the local filename is used in naming the remote file, after being altered by any `ntrans` or `nmap` setting. File transfer uses the current settings for `type`, `form` (format), `mode`, and `struct` (structure).

`bye`
Terminates the FTP session with the remote server and exits `ftp`. An end-of-file (CONTROL-D in the A/UX standard distribution) will also terminate the session and exit.

`cd` *remote-directory*
Changes the working directory on the remote computer to *remote-directory*.

`cdup`
Changes the remote computer working directory to the parent of the current remote computer working directory.

`close`
Terminates the FTP session with the remote server, and returns to the command interpreter. Any defined macros are erased.

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

`dir` [*remote-directory*] [*local-file*]
Prints a list of the directory contents in the directory, *remote-directory*, and, optionally, places the output in *local-file*. If no directory is specified, the current working directory on the remote computer is used. If no local file is specified, or *local-file* is `-`, output comes to the terminal.

`disconnect`
A synonym for `close`.

`get` *remote-file* [*local-file*]
Retrieves the *remote-file* and stores it on the local computer. If the local filename is not specified, it is given the same name it has on the remote computer, subject to alteration by the current `case`, `ntrans`, and `nmap` settings. The current settings for `type`, `form`, `mode`, and `struct` (structure) are used while transferring the file.

`help` [*command*]
Prints an informative message about the meaning of *command*. If no argument is given, `ftp` prints a list of the known commands.

`lcd` [*directory*]
Changes the working directory on the local computer. If no *directory* is specified, the user's home directory is used.

`ls` [*remote-directory*] [*local-file*]
Prints an abbreviated list of the contents of a directory on the remote computer. If *remote-directory* is left unspecified, the current working directory is used. If no local file is specified, or if *local-file* is `-`, the output is sent to the terminal.

`macdef` *macro-name*
Defines a macro. Subsequent lines are stored as the macro *macro-name*. A null line (consecutive newline characters in a file or carriage returns from the terminal) terminates macro input mode. There is a limit of 16 macros and 4096 total characters in all defined macros. Macros remain defined until a `close` command is executed. The macro processor interprets `$` and `\` as special characters. A `$` followed by a number (or numbers) is replaced by the corresponding argument on the macro invocation command line. A `$` followed by an `i` signals to the macro processor that the executing macro is to be looped. On the first pass, `$i` is replaced by the first argument on the macro invocation command line; on the second pass, it is replaced by the second argument, and so on. A `\` followed by any character is replaced by that character. Use the `\` to prevent special treatment of the `$`.

`mdelete` [*remote-files*]
Deletes the *remote-files* on the remote computer.

`mdir` *remote-files local-file*
Like `dir`, except multiple remote files may be specified. If interactive prompting is on, `ftp` will prompt the user to verify that the last argument is indeed the target local file for receiving `mdir` output.

`mget` *remote-files*
Expands the *remote-files* on the remote computer and does a `get` for each filename thus produced. See `glob` for details on the filename expansion. Resulting filenames will then be processed according to `case`, `ntrans`, and `nmap` settings. Files are transferred into the local working directory, which can be changed with

`lcd` *directory*

New local directories can be created with

`! mkdir` *directory*

`mkdir` *directory-name*
Makes a directory on the remote computer.

`mls` *remote-files local-file*
Like `ls`, except multiple remote files may be specified. If interactive prompting is on, `ftp` will prompt the user to verify that the last argument is indeed the target local file for receiving `mls` output.

`mput` *local-files*
Interprets metacharacters in the filenames in the list of local files given as arguments and does a `put` for each file in the resulting list. See `glob` for details of filename expansion. Resulting filenames will then be processed according to `ntrans` and `nmap` settings.

`nmap` [*inpattern outpattern*]
Sets or unsets the filename mapping mechanism. If no arguments are specified, the filename mapping mechanism is unset. If arguments are specified, remote filenames are mapped during `mput` commands and `put` commands issued without a specified remote target filename. If arguments are specified, local filenames are mapped during `mget` commands and `get` commands issued without a specified local target filename.

This command is useful when connecting to a non-UNIX remote computer with different file naming conventions or practices. The mapping follows the pattern set by *inpattern* and *outpattern*. *inpattern* is a template for incoming filenames (which may have already been processed according to the `ntrans` and `case` settings). Variable templating is accomplished by including the sequences

`$1, $2,..., $9`

in *inpattern*. Use the `\` character to prevent special treatment of the `$`, `[`, `]`, and `,` characters. All other characters are treated literally, and are used to determine the `nmap` *inpattern* variable values. For example, if the *inpattern* is

`$1.$2`

and the remote filename is `mydata.data`, `$1` would have the value `mydata`, and `$2` would have the value `data`. The *outpattern* determines the resulting mapped filename. The sequences

```
$1, $2,..., $9
```

are replaced by any value resulting from the *inpattern* template. The sequence `$0` is replaced by the original filename. Additionally, the sequence `[`*seq1,seq2*`]` is replaced by *seq1* if *seq1* is not a null string; otherwise, it is replaced by *seq2*.

For example, for input filenames `myfile.data` and `myfile.data.old`, the command

```
nmap $1.$2.$3 \[$1\,$2\].\[$2\,file\]
```

would yield the output filename `myfile.data`. For the input filename `myfile`, the same command would yield the output filename `myfile.file`, and for the input filename `.myfile`, the same command would yield `myfile.myfile`.

Spaces may be included in *outpattern*, as in the example:

```
nmap $1 |sed "s/ *$// "> $1
```

`ntrans` [*inchars* [*outchars*]]

Sets or unsets the filename character translation mechanism. If no arguments are specified, the filename character translation mechanism is unset. If arguments are specified, characters in remote filenames are translated during `mput` commands and `put` commands issued without a specified remote target filename. If arguments are specified, characters in local filenames are translated during `mget` commands and `get` commands issued without a specified local target filename.

This command is useful when connecting to a non-UNIX remote computer with different file naming conventions or practices. Characters in a filename matching a character in *inchars* are replaced with the corresponding character in *outchars*. If the character's position in *inchars* is longer than the length of *outchars*, the character is deleted from the filename.

`open` *host* [*port*]

Establishes a connection to the specified *host* FTP server. An optional port number may be supplied, in which case, `ftp` will attempt to contact an FTP server at that port. If autologin is enabled (default), `ftp` will also attempt to log the user in to the FTP server.

`proxy` *ftp-command*

Runs an `ftp` command on a secondary control connection. This command allows simultaneous connection to two remote FTP servers for transferring files between the two servers. The first `proxy` command should be an `open`, to establish the secondary control connection. Enter the command

```
proxy ?
```

to see other `ftp` commands executable on the secondary connection.

The following commands behave differently when prefaced by `proxy`:

`open`	Does not define new macros during the autologin process.
`close`	Does not erase existing macro definitions.
`get` and `mget`	Transfer files from the host on the primary control connection to the host on the secondary control connection.
`put`, `mput`, and `append`	Transfer files from the host on the secondary control connection to the host on the primary control connection. Third party file transfers depend upon support of the FTP protocol PASV command by the server on the secondary control connection.

`put` *local-file* [*remote-file*]

Stores a local file on the remote computer. If *remote-file* is left unspecified, the local filename is used, after processing according to any `ntrans` or `nmap` settings in naming the remote file. File transfer uses the current settings for `type`, `form` (format), `mode`, and `struct` (structure).

`pwd`

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

`quit`

A synonym for `bye`.

`quote` *arg1 arg2 . . .*

The arguments specified are sent, verbatim, to the remote FTP server.

`recv` *remote-file* [*local-file*]
A synonym for `get`.

`remotehelp` [*command-name*]
Requests help from the remote FTP server. If a *command-name* is specified, it is supplied to the server as well.

`rename` [*from*] [*to*]
Renames the file *from* on the remote computer, to the file *to.*

`reset`
Clears the reply queue. This command resynchronizes command/reply sequencing with the remote FTP server. Resynchronization may be necessary following a violation of the FTP protocol by the remote server.

`rmdir` *directory-name*
Deletes a directory on the remote computer.

`send` *local-file* [*remote-file*]
A synonym for `put`.

`status`
Shows the current status of `ftp`.

`user` *username* [*password*] [*account*]
Identifies yourself to the remote FTP server. If the password is not specified, and the server requires it, `ftp` prompts you for it (after disabling local echo). If an account field is not specified, and the FTP server requires it, you will be prompted for it. If an account field is specified, an account command will be relayed to the remote server after the login sequence is completed if the remote server did not require it for logging in. Unless `ftp` is invoked with autologin disabled, this process is done automatically on initial connection to the FTP server.

`?` [*command*]
A synonym for `help`.

Setting up the `.netrc` file

You may set up a `.netrc` file in your login directory to provide information to the `ftp` autologin process. This file is used to contain login and initialization information for the remote FTP server.

◆ **Note** For security reasons, this file is often not used because a user logged in at `root` on the local system is able to obtain your password for the remote FTP site by looking at this file. Note that if you supply your remote password in this file, `ftp` aborts the autologin process unless the `.netrc` file is unreadable to anyone but yourself. Thus, after you create a `.netrc` file, you should change the mode to read-write permissions for yourself only. ◆

The `.netrc` file uses the following keywords; they may be separated by spaces, tabs, or newline characters:

`machine` *rhost* (where *rhost* is the remote computer name)
The autologin process searches the `.netrc` file for a `computer` token that matches the remote computer specified on the `ftp` command line or as an `open` command argument. If it finds the right name, it processes the rest of the tokens in your `.netrc` file, stopping when the end-of-file condition is reached or another `machine` token is encountered.

`login` *name* (where *name* is your login name on the remote computer)
If this token is present, the autologin process initiates a login using the specified name.

`password` *string* (where *string* is your password on the remote computer)
If this token is present, the autologin process supplies the specified string if the remote server requires a password as part of the login process. Note that if this token is present in the `.netrc` file, `ftp` aborts the autologin process if the `.netrc` is readable by anyone other than the user.

`account` *string* (where *string* is an additional account password)
If this token is present, the autologin process supplies the specified string if the remote server requires an additional account password, or the autologin process initiates an ACCT command, if it does not.

`macdef` *name* (where *name* is a macro definition)
This token works like the `ftp macdef` command. A macro is defined with the specified name; its contents begin with the next `.netrc` line and continue until a null line (consecutive newline characters) is encountered. If you define a macro named `init`, it is automatically run as the last step in the autologin process.

Abbreviated command formats

Your system administrator can set up a `/usr/hosts` file on each system. If this file exists, include the path `/usr/hosts` in the `PATH` variable in your local `.login` file. (Actually, this file can have any pathname as long as you include it in your `PATH` variable.)

When you have done this, you may omit the `rlogin` and `remsh` names from the command line. In this case, the `rlogin` command format is

rhost

or

rhost `-l` *username*

The `remsh` command format is

rhost command

or

rhost `-l` *username command*

When used in this format, *rhost* must be the official name of the system (that is, the name shown in `ruptime` output).

Toggle commands

The following `ftp` commands toggle environment and file transfer conditions (that is, turn a condition on or off, depending on the previous setting). Arguments enclosed in brackets are optional; the brackets aren't part of the command syntax. By default, `bell` is turned off, `case` is turned off, `cr` is turned on, `prompt` is turned on, `runique` is turned off, and `sendport` is turned on.

`bell`
Arranges that a bell be sounded after each file transfer command is completed. The default is off.

`case`
Toggles remote computer filename case mapping during `mget` commands. When `case` is on (default is off), remote computer filenames with all letters in uppercase are written in the local directory with the letters mapped to lowercase.

`cr`
Toggles carriage return stripping during ASCII-type file retrieval. Records are denoted by a carriage return–line feed sequence during ASCII-type file transfer. When `cr` is on (the default), carriage returns are stripped from this sequence to conform with the UNIX convention (a single line feed as the record delimiter). Records on non-UNIX remote systems may contain single line feeds; when an ASCII-type transfer is made, these line feeds may be distinguished from a record delimiter only when `cr` is off.

`debug` [*debug-value*]
Toggles debugging mode. If an optional *debug-value* is specified, it is used to set the debugging level. When debugging is on, `ftp` prints each command sent to the remote computer, preceded by an arrow (the string `-->`).

`glob`
Toggles filename expansion for `mdelete`, `mget`, and `mput`. If "globbing" is turned off with `glob`, the filename arguments are taken literally and not expanded. Globbing for `mput` is done as in the C shell (see `csh`(1) in *A/UX Command Reference*). For `mdelete`

and `mget`, each remote filename is expanded separately on the remote computer and the lists are not merged. Expansion of a directory name is likely to be different from expansion of the name of an ordinary file: the exact result depends on the foreign operating system and FTP server and can be previewed by issuing

`mls` *remote-files* -

◆ **Note** `mget` and `mput` are not meant to transfer entire directory subtrees of files. That can be done by transferring a `tar` archive of the subtree (in binary mode). (See `tar`(1) in *A/UX Command Reference* for more information.) ◆

`hash`
Toggles number-sign (`#`) printing for each data block transferred. The size of a data block is 1024 bytes.

`prompt`
Toggles interactive prompting. Interactive prompting occurs during multiple file transfers to allow the user to retrieve or store files selectively. If prompting is turned off (default is on), any `mget` or `mput` will transfer all files, and any `mdelete` will delete all files.

`runique`
Toggles storing of files on the local system with unique filenames. If a file already exists with a name equal to the target local filename for a `get` or `mget` command, a `.1` is appended to the name. If the resulting name matches another existing file, a `.2` is appended to the original name. If this process continues up to `.99`, an error message is printed, and the transfer does not take place. The generated unique filename will be reported. Note that `runique` will not affect local files generated from a shell command. The default value is off.

`sendport`
Toggles the use of PORT commands. By default, `ftp` will attempt to use a PORT command when establishing a connection for each data transfer. The use of PORT commands can prevent delays when performing multiple file transfers. If the PORT command fails,

`ftp` will use the default data port. When the use of PORT commands is disabled, no attempt will be made to use PORT commands for each data transfer. This is useful for certain FTP implementations which do ignore PORT commands but, incorrectly, indicate they've been accepted.

`sunique`
Toggles storing of files on remote computer under unique filenames. The remote FTP server must support FTP protocol STOU command for successful completion. The remote server will report the unique name. Default value is off.

`trace`
Toggles packet tracing.

`verbose`
Toggles verbose mode. In verbose mode, all responses from the FTP server are displayed. In addition, if verbose is on and a file transfer is completed, statistics regarding the efficiency of the transfer are reported. By default, verbose is on.

File transfer parameters

The following `ftp` commands set file transfer parameters. By default, `ftp` uses `ascii` type, `file` format, `stream` mode, and `stream` structure.

`ascii`
Sets the file transfer type to network ASCII. This is the default.

`binary`
Sets the file transfer `type` to support binary image transfer.

`form` *format*
Sets the file transfer `form` to *format*. The default format is `file`.

`mode` [*mode-name*]
Sets the file transfer mode to *mode-name*. The default mode is `stream` mode.

`struct` [*struct-name*]
Sets the file transfer structure to *struct-name*. By default, `stream` structure is used.

`tenex`
Sets the file transfer `type` to that needed to talk to TENEX computers.

`type` [*type-name*]
Sets the file transfer type to *type-name*. If no type is specified, the current type is printed. The default type is network ASCII.

1 Introduction to A/UX Networking

A/UX expands the reach of your Apple Macintosh computer into UNIX **network** environments, providing you the ability to obtain information and use services from remote computers, both Macintosh and non-Macintosh. Under A/UX, you have all of the same capabilities you have with the Macintosh, plus the ability to communicate with users on a variety of other computer systems. Figure 1-1 shows the various connections you can make as an A/UX user.

This chapter provides an introduction to some concepts that may be helpful for you to understand and use your network effectively. Review this chapter so that you'll be familiar with A/UX networking terms. This chapter contains the following sections:

- A/UX network connections
- network features of your A/UX system
- choosing a network file-sharing option

To know where to turn in the manual for the information you most need, see the section before this chapter called "About This Guide."

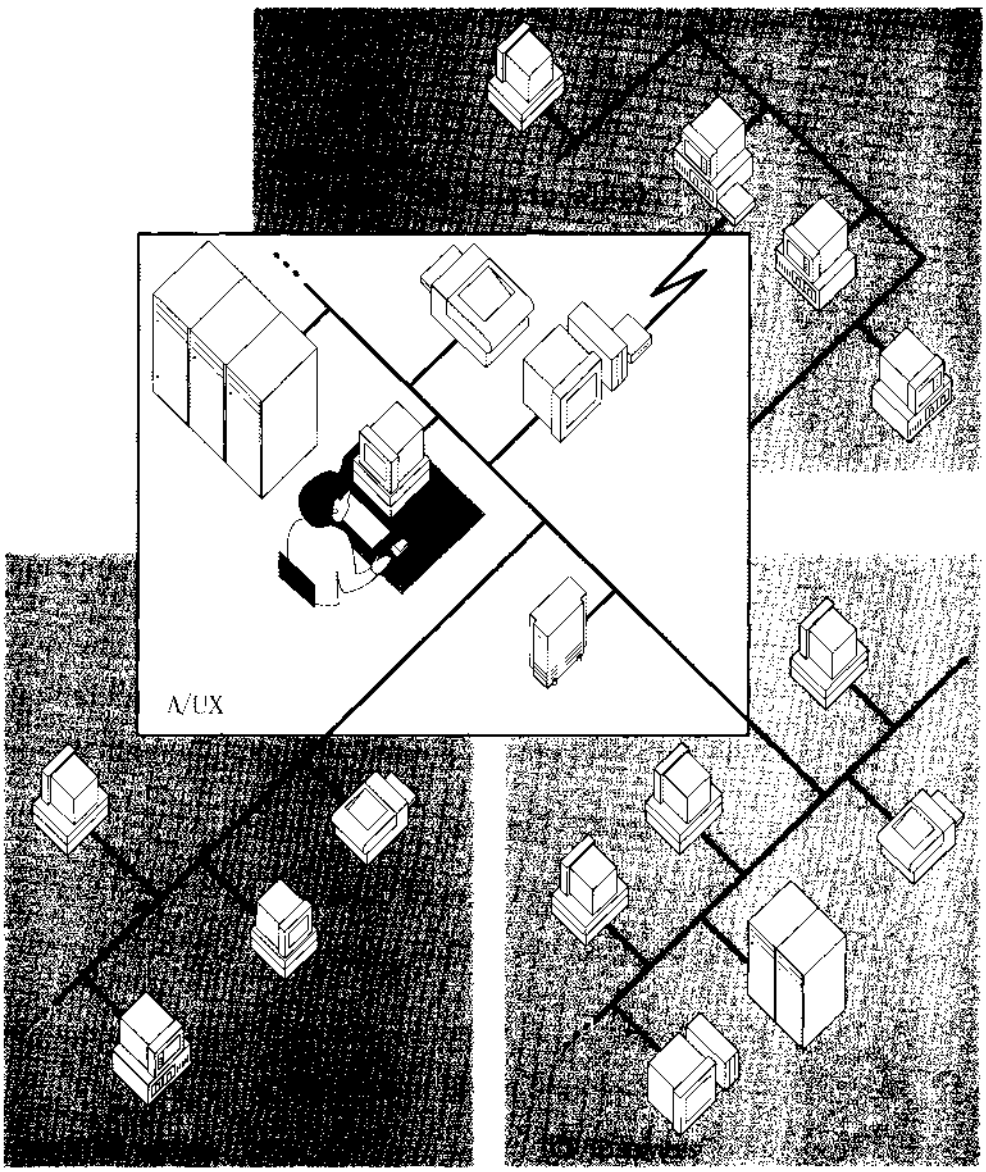


Figure 1-1 Network connections you can make as an A/UX user

A/UX network connections

Various networking services are available to you as a user of A/UX 3.0. You'll be able to share computers, files, printers, software, and other resources. You'll also be able to send and receive mail. Figure 1-1 shows some of the network connections you can make. They are described briefly below:

- **AppleTalk services**

  A/UX includes AppleTalk network software. You use **AppleTalk** services to print over the network and to share files with other computers that support AppleTalk. AppleTalk is a set of **protocols** that supports two types of networks: LocalTalk and EtherTalk. You can change network types with the Network control panel. You can connect to either network, but you cannot connect to both at the same time.

  - LocalTalk provides low-cost communication between computers and devices such as printers. It is the default network, which means that it's selected when you log in to A/UX.
  - EtherTalk provides high-speed communication between computers over an Ethernet network. Your computer must be configured to use the TCP/IP networking software in order to use EtherTalk. See Chapter 2 for information about installing an Ethernet card and the procedure to configure your network.

- **TCP/IP services**

  A/UX includes TCP/IP network software. With TCP/IP you can communicate with other TCP/IP-supported computers on your network (regardless of their **operating systems**) to log in to remote computers, copy files, receive a status report on computers on the network, and send and receive mail. Both EtherTalk networks and TCP/IP networks run over Ethernet hardware, so you can be connected to both an AppleTalk network and a TCP/IP network at the same time. See Chapter 5 for more information about the capabilities of TCP/IP.

- **Serial services**

  A/UX comes with standard UNIX commands that use the Macintosh serial port to connect to other UNIX and non-UNIX systems. By connecting a serial line or modem from a Macintosh serial port to another computer running a standard version of UNIX, you

can use a variety of communications utilities that allow you to log in to remote computers and transfer files. The standard UNIX utilities include the UUCP set of utilities (`uucp`, `uux`, and `cu`), as well as `tip` and `kermit`. See Chapter 8 and Chapter 9 in this manual for more information.

These networking options allow you access to the variety of network features provided by your A/UX system.

Network features of your A/UX system

Your A/UX system software (Release 3.0 and later) provides a number of networking capabilities. Table 1-1 shows different ways of doing the same thing. For example, you can share files with other users on Macintosh computers or on UNIX computers; you can copy and transfer files, log in to remote computers, and print on the network. The table gives you a quick review of the capabilities available to you depending upon who you're trying to communicate with, and it indicates the kind of connection you'll use.

Table 1-1 Networking features of your A/UX system

A/UX networking connections	Share and access files	Copy and transfer files	Connect to remote computers	Send and receive mail	Print documents
to Macintosh	AppleTalk: file sharing			Third-party mail programs	AppleTalk: Chooser `lpr`

to UNIX	TCP/IP: Network File System (NFS)	TCP/IP:
		`rcp`
		`ftp`
			TCP/IP:
			`rlogin`
			`telnet`
				`mailx`
				`(sendmail)`
					TCP/IP:
					`lpr`

to non-UNIX		`tip`			

- *Sharing files from your own computer and accessing information on other computers.* You can share files from your A/UX computer or system so that other people can access them over the network from either a Macintosh or another UNIX-based system. You can also control who can access this data and whether they can open, read, or modify the information that you've made available. (See Chapters 6 and 7.)
- *Copying and transferring files.* You can copy and transfer files and directories between your A/UX computer and another UNIX-based computer on the network. (See Chapter 5 and Appendix D.)
- *Logging in to a remote computer.* Working at your own computer, you can connect to another UNIX-based computer and run commands to access information you want to use. (See Chapters 5, 8, and 9.)
- *Sending and receiving mail.* You can send and receive mail between your A/UX computer and other UNIX-based systems, or you can use a third-party mail program to send mail to other Macintosh computers, even if they aren't running A/UX. (See Chapter 4 and Appendix B.)
- *Printing documents on network printers.* As soon as you connect to the network, you can print documents on network printers by selecting printers listed in the Chooser, and you use them just as you use printers that are connected directly to your computer. You can also print to a printer that is directly connected to a remote computer if such printing has been enabled on your host by your **network administrator**. (See Chapter 3.)

Choosing a network file-sharing option

A/UX offers these means by which you can share files between your computer and another computer on the network:

- AppleShare, which is available through the Chooser
- File sharing, which is available through the File menu
- Network File System (NFS), which you set up through the A/UX CommandShell

The method you choose depends on the kind of computers with which you are sharing files and the types of files you are sharing (see Table 1-1). AppleShare and the Macintosh file-sharing capabilities allow you to share files with other Macintosh computers, whether those other systems are running A/UX or not. Network File System (NFS) software, a standard networking service that comes with A/UX, allows you to share files with any kind of computer running the UNIX operating system and NFS. You can use NFS to share files with other Macintosh computers running A/UX and with other computers running NFS.

If all the users with whom you'd like to share files are working on Macintosh computers, Macintosh file sharing, described in Chapter 6, is recommended. If you want to share files with users on non-Macintosh UNIX systems, then the NFS procedures described in Chapter 7 are appropriate.

2 Connecting to an Existing Network

This chapter describes how to connect your A/UX system to an existing network. It doesn't describe how to design and set up a network; this job belongs to your network administrator.

This chapter contains the following sections:

- types of networks you can connect to
- identifying your network
- connecting to a TCP/IP network
- testing the network connection
- switching AppleTalk networks
- what's available now

Typically a network administrator connects the system to a network, but with the proper information from your network administrator, you can connect your computer to an existing network. If you're new to networking, read the sections "Types of Networks You Can Connect To" and "Identifying Your Network" before you do the procedures. If you're comfortable that you know enough about networks, go straight to the section "Connecting to a TCP/IP Network."

For an in-depth look at network setup, see *A/UX Network System Administration.* For more information on network basics, such as what kinds of networks can be-set up and how networks operate, see the Apple publication, *Understanding Computer Networks* (Addison-Wesley, 1989).

Types of networks you can connect to

Your Macintosh computer comes with networking software built in, which allows your computer to operate on an AppleTalk network quickly and easily. Here are the network options available to you through your A/UX system:

- **Ready-to-use LocalTalk**

  LocalTalk connection hardware and software are built into your computer, allowing you to connect to an AppleTalk network by simply plugging LocalTalk-compatible cables into the back of your computer. LocalTalk is also built into LaserWriter printers, making them equally simple to connect to a LocalTalk network.

  If this is the only network you will be using, first connect your system as described in *LocalTalk Cable System Owner's Guide,* which is in the LocalTalk connector kit. Next, see Chapter 3 of this manual for information on how to select an AppleTalk printer.

- **Ethernet**

  Ethernet is a standard network connection medium. You can use it to transmit AppleTalk packets, and you can use it to transmit TCP/IP packets. An Ethernet cable can be used to transmit both kinds of packets at the same time. As an A/UX user, you might wish to connect to the Ethernet-based **TCP/IP** (**Transmission Control Protocol/Internet Protocol**) network, which allows you access to the **Network Information Service (NIS)** (formerly Yellow Pages) and the **Network File System** (**NFS**) facility, giving you the ability to share network resources with a wide variety of machines. Also, if the Ethernet network is connected to another network that has LaserWriter printers, you can set up your system to use printers on that AppleTalk network, too.

  The Macintosh Quadra 700 and Quadra 900 come equipped with built-in Ethernet. If you have one of the other A/UX systems, you can add an expansion card to your computer to connect to an Ethernet network. To install the card, follow the instructions in the manual that comes with the card and physically connect your system to the Ethernet network. Then proceed to the section in this chapter called "Connecting to a TCP/IP Network."

  An Ethernet network can be connected to a LocalTalk network using a router, which allows you to print on printers connected to the LocalTalk network from a system on the Ethernet network. There is also a printer that allows you to print directly on the EtherTalk network. For more information, see the manual that came with your printer.

Identifying your network

Because A/UX supports both LocalTalk and Ethernet, the computers on your network may be connected by either LocalTalk or Ethernet cables, or both.

- If your cable looks like those in Figure 2-1, you are connected to a LocalTalk network. Hardware support for LocalTalk is built into all Macintosh computers.

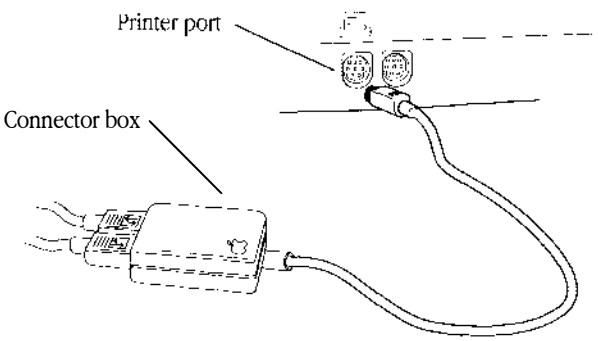


Figure 2-1 LocalTalk cables and connectors

- If your cable looks like those in Figure 2-2, you are connected to an Ethernet network. To connect with Ethernet, your system must have Ethernet hardware.

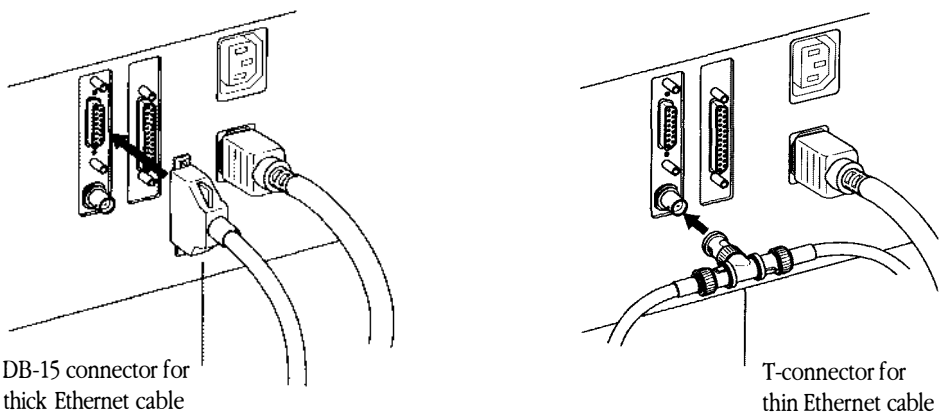


Figure 2-2 Ethernet cables and connectors

▲ **Warning** If you are using thick Ethernet, be sure that you connect the DB-15 thick Ethernet cable to the Ethernet card and not to the video card or you will cause serious damage to your computer. The Ethernet card is easy to identify: it has connectors for both the thick Ethernet cable and the thin Ethernet cable, as you can see in Figure 2-2. ▲

- If your cable looks like one in Figure 2-3, you are connected to an Ethernet network with an Apple Ethernet media adapter. The Macintosh Quadra computers and computers that have the Apple Ethernet NB Card require an Apple Ethernet media adapter.

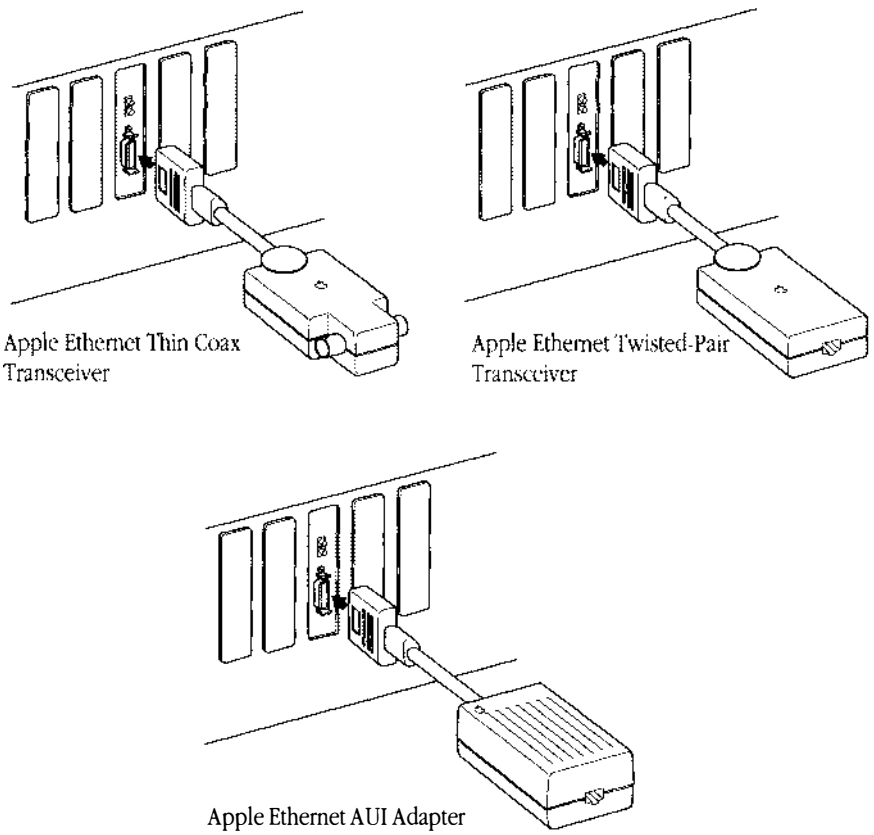


Figure 2-3 Apple Ethernet media adapters

If you are confident that you know what your network options are and you've determined the type of network cable you have, continue to the next section for a description of how to connect to a TCP/IP network. If not, check with your network administrator before proceeding.

Connecting to a TCP/IP network

After you install hardware and connect the Ethernet cables, you can set up A/UX files to communicate with other computers and share resources on a TCP/IP network. You'll be able to use EtherTalk to connect to other Macintosh computers and printers, share files using NFS, log in to remote systems and copy files, and run commands remotely. Each of these capabilities is described in a chapter in this manual.

In general, **local system** refers to the computer on which you initially log in using the standard A/UX login procedure. **Remote system** refers to other computers on the network that you access using the TCP/IP networking commands.

Before you connect to a network

Before you can set up your computer to use the services of a TCP/IP network, you need to know the Internet address and the netmask number of your system. When you make a new kernel, you'll be prompted to enter this information, so you need to collect it before you start. It's possible that your computer has been connected to a network at one time. If it has, you won't be prompted to enter this information, but when you run `newconfig bnet`, you will see the current settings.

Before you connect your computer to a network, follow these steps:

1 **Collect the necessary network addressing and services information from your network administrator. You need the following information:**

- A host name, which is a unique name that you give your system to identify it. You can enter any host name you choose, as long as it meets these conditions:

- The name must be no longer than 31 characters. If you plan to use UUCP, you may want to limit your host name to 14 characters.
- The name must be unique on the network.
- The name may be made up of lowercase letters, numbers, or hyphens. The first character must be a letter; the last character may be a number or a letter. For example, you might call your system "sierra" or "pacific1" or "mono-lake."

- An NIS domain name, if the network you are connected to uses NIS. If it does, ask your network administrator for your NIS domain name and whether you should use NIS for password, group, and `sendmail` alias lookup.
- Your Internet (IP) address (a number such as `192.33.20.1`).
  The IP address is made up of two parts—the network part and the host part. The network part (`192.33.20` in the example) is shared by everyone on the network. The host part (`1` in the example) must be unique—any number from 1 to 254. Each host on the network has its own number, so it's important that you do not make up a number. Each IP address conveys important information that can affect a large number of network users, so be sure to use legitimate numbers. Your computer must be the only one on the network that has this number. This discussion of IP addresses is by no means complete. For a full discussion, see *A/UX Network System Administration.*
- A **netmask**
  The netmask enables the system to identify local computers, thereby keeping local traffic off the general network. An example netmask is `0xffffff00` (zero followed by eight hex digits). Networks that don't use subnets use a default netmask. This default is entered automatically by the A/UX system if you do not enter a mask.
- The **Internet address** of another computer that is running on the network
  You'll use this address to test that your computer is working on the network.

◆ **Note** You also need the broadcast address for the system you're setting up, but you don't need to collect this information from the network administrator because the system calculates it for you from the IP address and the netmask. ◆

2 **Log in as** `root`.

If you need help with this, see Chapter 1 of *A/UX Essentials*.

3 **Open a CommandShell window if one isn't already open.**

In the Finder, choose CommandShell from the Applications menu.

4 **To see if the IP address and netmask information have been previously entered, enter these commands:**

```
cat /etc/HOSTNAME
cat /etc/NETADDRS
```

If the IP address and netmask have not been previously entered, `cat` will display this message: "`cat: cannot open` *file*." If this information has been previously entered and you don't change it, when you run `newconfig bnet`, you won't be prompted to enter this information, but you will see the current settings.

5 **If the IP address and netmask information have been previously entered and you would like to change them to new values when you run** `newconfig bnet`**, rename the old files first by entering this command:**

```
mv /etc/NETADDRS.OLD
```

Because you've renamed the existing files, the system will prompt you to enter the information again when you run the `newconfig bnet` command.

Next, edit the `/etc/hosts` file and remove all references to the old host name and IP address.

Setting up your computer to use the TCP/IP network

If you've collected all the information you need, you're ready to set up your computer to use the services of the TCP/IP network. To do this, make sure you're logged in as `root`. After you enter the `newconfig bnet` command to make a new kernel, follow the prompts and then restart the system so the changes take place. For more information about making a kernel, see *A/UX Local System Administration.*

▲ **Warning** If you make a mistake typing the information and *you haven't pressed* RETURN *yet*, use the DELETE key to correct errors. *Do not* use the arrow keys to try to edit incorrect information. After you've entered the information and pressed RETURN, the system will prompt you to check the information to make sure it's correct. If the IP address or the netmask isn't correct, the system will prompt you to enter the numbers again. ▲

1 **If you haven't already logged in, open a CommandShell window and log in as `root`.**

2 **To make a new kernel, enter this command:**

```
newconfig bnet
```

Messages appear indicating that it may take several minutes to prepare to build a new kernel. Then this prompt appears:

```
Do you want this machine to be an NIS client?  (y or n)
[default: n]
```

If you enter `y`, you'll be prompted to answer `y` or `n` for NIS password, group, and `sendmail` alias lookup.

3 **Press RETURN to choose the default (n).**

This reminder appears:

```
Remember to list the file systems you wish to export in
/etc/exports.
```

Messages appear telling you it will take a few minutes to build a new kernel. Then, if a host name hasn't been previously entered, the system prompts you for one:

```
Please enter a hostname (it must be unique):
```

Remember, if this information has been previously entered, the prompt will not appear.

4 **Enter your host name.**

The host name must meet the conditions described in the previous subsection, "Before You Connect to a Network." You are not prevented, however, from entering an invalid host name.

5 **Enter additional information as prompted.**

If the information hasn't been previously entered, the system prompts you to enter the information you collected from your network administrator: your IP address and netmask. (If this information has been previously entered, the prompts do not appear.) After you enter the information, you'll see a confirmatory message that looks something like this:

```
ae0:  IP address 192.33.20.1
      netmask 0xffffff00
      broadcast address 192.33.20.255
OK?   [y/n, default=y]>>>
```

Remember, the broadcast address was calculated for you, so you don't have to correct it.

If you enter the wrong IP address or netmask, you may see a message indicating the nature of the error and prompting you to enter the number again.

6 **Check the numbers to make sure they're correct.**

If the IP address or the netmask is incorrect, enter `n`. When you enter `n`, you'll be prompted to enter the Internet address and the netmask again, and another prompt will appear asking you if the numbers are ok.

If the IP address and the netmask are correct, enter `y`. When you enter `y`, you'll be prompted to restart.

The changes won't be effective until you restart the system. When `newconfig` is done, it displays the current settings for your host name, IP address, IP broadcast address, and netmask.

7 **Return to the Finder by selecting Finder in the Applications menu.**

8 **Choose Restart from the Special menu.**

The system restarts using the new kernel, and the changes take effect.

◆ **Note** After you restart, if you discover you've entered the wrong host name, domain name, IP address, or netmask, you can correct the error and run `newconfig` again. See Appendix A for information. ◆

Testing the network connection

Follow these steps to check that your system has a working connection to the network:

1 **Log in and open a CommandShell window.**

2 **Enter this command:**

```
telnet loop
```

After a short time, the login prompt appears

```
Apple Computer, Inc. A/UX (hostname)

login:
```

3 **Press CONTROL-D to stop the test.**

If the login prompt doesn't appear in several seconds, see your network administrator. For details on network tests, see the section called "Testing the Networking Software" in Chapter 2, "Configuring and Managing a TCP/IP Network," of *A/UX Network System Administration.*

4 **Test communications across the network.**

To check whether your system can communicate across the network, you need to know the address of another system on the network. Get this information from another user or from your network administrator. Use the command

`ping` *address*

Replace *address* with the address of another network computer. For example, if `192.33.20.2` is the address of another system already on the network, you would enter

```
ping 192.33.20.2
```

If the network and both hosts are functional, a display like this appears:

```
64 bytes from 192.33.20.2: icmp_seq=0. time=16 ms
64 bytes from 192.33.20.2: icmp_seq=1. time=16 ms
64 bytes from 192.33.20.2: icmp_seq=2. time=16 ms
64 bytes from 192.33.20.2: icmp_seq=3. time=16 ms
```

Stop the display by pressing CONTROL-C. Some statistics appear, then the system prompt, as shown here:

```
----address PING Statistics----
6 packets transmitted, 6 packets received, 0% packet loss
round-trip (ms) min/avg/max = 16/16/16
```

If you don't see a display like this one, first make sure all your cables are connected. Then see your network administrator.

Switching AppleTalk networks

When you set up your A/UX system, LocalTalk is the default AppleTalk network. The following subsections tell you how to switch from LocalTalk to an Ethernet connection, and, if the need arises, how to switch from Ethernet to LocalTalk.

Switching from LocalTalk to EtherTalk

Any user who is logged in to the A/UX Finder environment can switch from a LocalTalk to an Ethernet connection. Once EtherTalk is selected in the Network control panel, it remains selected even if you restart the system.

▲ **Warning** Before you switch from LocalTalk to EtherTalk, make sure no LocalTalk cables are connected. Disconnecting the LocalTalk cable prevents the degraded system performance that occurs if you inadvertently start a `getty` process on a printer **port** to which a LocalTalk cable is connected.

Figure 2-4 shows the location of the printer port on your A/UX computer. The printer icon may be located directly above or below the printer port or to one side of the printer port.

It's unlikely that you would need to switch from EtherTalk to LocalTalk, but if such an event should occur, see the following section. ▲

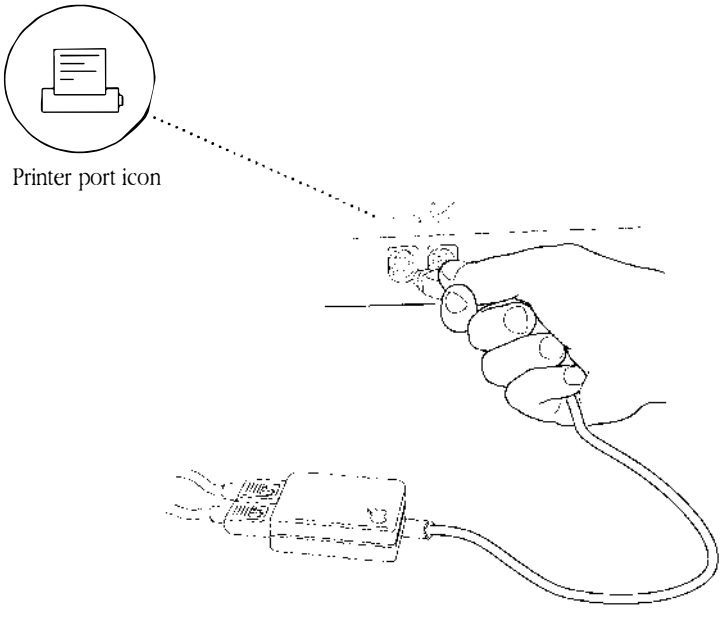


Figure 2-4 LocalTalk printer ports

Follow these steps to switch from LocalTalk to EtherTalk:

1 **Log in to your A/UX system.**

2 **Select Control Panels from the Apple (🍎) menu.**

3 **Scroll, if necessary, to locate the Network icon; then double-click it to open the Network control panel.**

The Network control panel appears, with the icon of the network currently in use selected. The icons represent the available choices for your network connection, including LocalTalk Built-in and EtherTalk. The LocalTalk Built-in icon represents the LocalTalk network connection. If your Macintosh has more than one Ethernet interface card, you will see a separate connection icon for each card.

4 **Click the EtherTalk icon.**

When you select the icon, it becomes highlighted.

5 **Answer the alert box query.**

The query tells you that access to current network services will have to be reestablished and asks you if you are sure that you want to change the connection. If your computer is currently running a Macintosh or a UNIX application that cannot gracefully handle a transition between LocalTalk and EtherTalk, close these applications. Otherwise, click the OK button.

When you click the OK button, the selected network connection becomes your computer's AppleTalk connection until you or another user selects LocalTalk or until the Ethernet interface card that corresponds to the selected icon is removed. In the latter case, your Macintosh computer automatically changes the network connection to its built-in LocalTalk connection.

6 **Close the Network control panel.**

You can close the Network control panel by clicking its close box. Then click the Control Panels close box.

◆ **Note** When you run AppleTalk using an EtherTalk connection, you can still use an ImageWriter printer or other serial device connected directly to your computer's printer port. ◆

Switching from EtherTalk to LocalTalk

Once you've switched from LocalTalk to EtherTalk, it's unlikely you'll want to switch back, but if, for example, your Ethernet card is down and you want to use LocalTalk, you can switch networks. Once LocalTalk is selected, it remains selected even if you restart the system. Follow these steps to switch from EtherTalk to LocalTalk:

1 **Log in to your A/UX system in the Finder environment.**

2 **Select CommandShell from the Applications menu and disable any** `getty` **process that may be running on the printer port.**

```
setport -o tty1
```

The entry in the `/etc/inittab` file for the printer port (`tty1`) should now be as follows:

```
01:2:off:/etc/getty tty1 at_9600 #port...
```

Note that the `setport` command always defaults to 9600 baud. If you want to retain a specific baud rate, check the `/etc/inittab` file for the current baud rate and run `setport` with the `-s` option. Here is an example:

```
setport -o -s 19200 tty1
```

The entry for `tty1` is now as follows:

```
01:2:off:/etc/getty tty1 at_19200 #port...
```

The `setport` commmand automatically stops the `getty` process by running `init q`.

3 **Connect the LocalTalk cable to the printer port at the back of your computer.**

4 **Choose Control Panels from the Apple (🍎) menu.**

5 **Scroll, if necessary, to locate the Network icon; then double-click it to open the Network control panel.**

The Network control panel appears, with the icon of the network currently in use selected. The icons represent the available choices for your network connection, including Built-in and EtherTalk. The Built-in icon represents the LocalTalk network connection. If your Macintosh has more than one Ethernet interface card, you will see a separate connection icon for each card.

6 **Click the Built-in icon.**

When you select the icon, it becomes highlighted.

7 **Answer the alert box query.**

The query tells you that access to current network services will have to be reestablished and asks you if you are sure that you want to change the AppleTalk connection. If the system is running a Macintosh or a UNIX application that cannot gracefully handle a transition between EtherTalk and LocalTalk, click the Cancel button and close the application before making the change. Otherwise, click the OK button.

When you click the OK button, your computer's printer port is used as a LocalTalk connection to an AppleTalk network and remains the connection until you or another user switch to an EtherTalk connection or disable AppleTalk.

8 **Close the Network control panel.**

You can close the Network control panel by clicking its close box. Then click the Control Panels close box.

◆ **Note** If any users on your system have already created personal System Folders (as described in *A/UX Essentials*), they may need to update their personal System Folders to use a different default printer. ◆

What's available now

Now that you've completed the basic steps for connecting to the network and setting up your computer on the network, you can gain access to other computers on the network and use remote printers. See the appropriate chapter in this manual for more information.

Several other network services are available; for instance, you can make your files available to other network users and access other network services. To implement these options, see your network administrator for specifics on your network configuration.

See *A/UX Network System Administration* for complete information about network setup and management, as well as information on adding services such as NFS.

3 Printing Over the Network

This chapter discusses printing with a printer that is connected to a network. It describes selecting the printer from the Chooser and printing with `lpr`. It also tells you how to set up access to a printer on a TCP/IP network.

For more information about printing, see Chapter 7 in *A/UX Essentials*. For a quick look at the printer language and printer hardware available on the network, see "Overview of Available Print Facilities" at the end of this chapter.

This chapter contains these sections:

- understanding network printing
- choosing a printer
- printing from the finder
- printing with `lpr`
- overview of available print facilities

You can read this chapter in its entirety or go directly to those sections that are of use to you.

Understanding network printing

A/UX provides several printing utilities, which print files created by a text editor (such as `vi` or TextEditor), by a word processor, or by other Macintosh or UNIX applications.

Before you can print a document over the network, your system must be connected to the network. Chapter 2 describes the steps for connecting to a network. This chapter focuses on how the printer is connected to the network, how you select the printer, and how you set up access to a printer on a TCP/IP network. Consult *Setting Up Accounts and Peripherals for A/UX* to connect a printer on an AppleTalk network.

Printer connections

A printer can be used by network users when it is connected to the network in one of the ways below:

- **LocalTalk connection**

  A/UX systems that are connected to a LocalTalk network can send files created with a text editor (including `troff` output) or a Macintosh application to the printer. The files are printed in the order they are received. If the LocalTalk network is connected to an EtherTalk network by a Macintosh computer running the Apple Internet Router software, A/UX systems on the LocalTalk network can also print on any AppleTalk printer that may be connected to the EtherTalk network.

- **EtherTalk connection**

  A/UX systems that are connected to an EtherTalk network can send files created with a text editor (including `troff` output) or a Macintosh application to the printer. The files are printed in the order they are received. If the EtherTalk network is connected to a LocalTalk network by a Macintosh computer running the Apple Internet Router software, A/UX systems on the EtherTalk network can also print on any AppleTalk printer that may be connected to the LocalTalk network.

- **Direct connection to a remote computer**

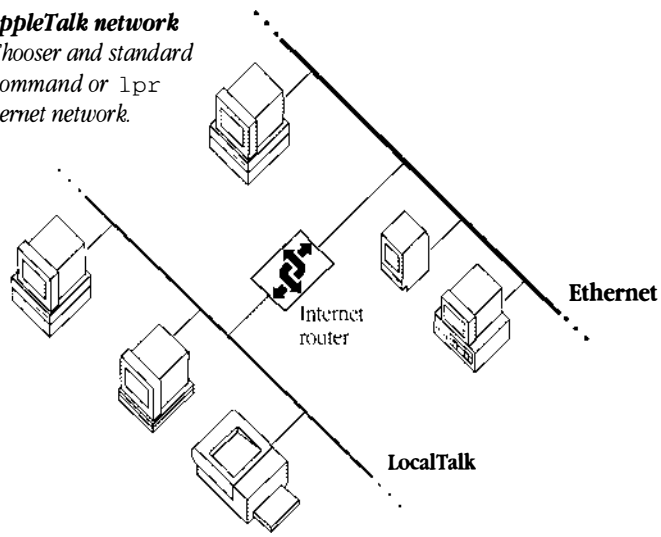
  A/UX systems that are connected to a TCP/IP network can send text files (but not the output of Macintosh applications) to a printer that is directly connected to a remote computer that, like A/UX, supports the `lpr` spooler. The files are printed in the order in which they are received.

To learn about connecting a printer directly to an A/UX computer, see *Setting Up Accounts and Peripherals for A/UX.*

◆ **Note** A single Ethernet cable supports both EtherTalk and TCP/IP networks at the same time. ◆

Figure 3-1 shows the different printer configurations.

Printing on an AppleTalk network
You can use the Chooser and standard Macintosh Print command or `lpr` *to print on an Ethernet network.*



Printing on a TCP/IP network
You can use the `lpr` *and* `lp` *commands to print to a spooler on a TCP/IP network.*

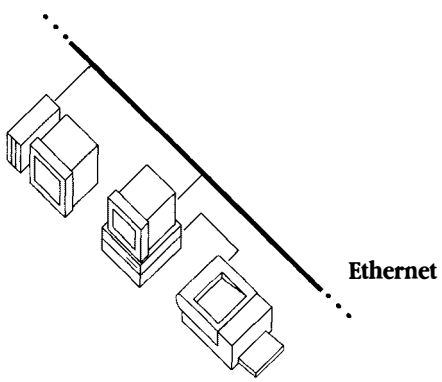


Figure 3-1 Two types of printer connections

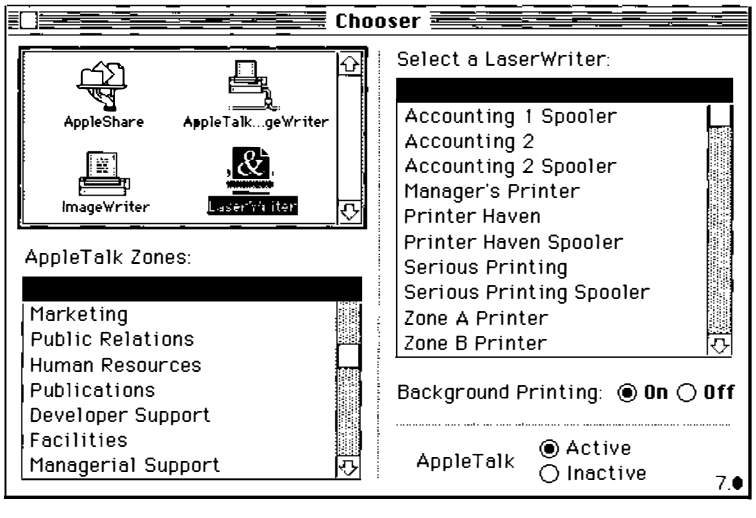
Choosing a printer

Before you can print a file, A/UX must know which printer to use. You choose a printer for A/UX just as you do for the Macintosh Operating System (OS). Use the Chooser (in the Apple menu) to select a printer that has been connected to your network through LocalTalk (or LocalTalk-compatible) or EtherTalk cables. Some applications direct you to choose Page Setup after using the Chooser. For further information, see the guide that came with the application you are using.

Follow these steps to choose a printer:

1 **Choose Chooser from the Apple menu.**

The Chooser dialog box appears. You see an icon representing each of the printer types for which you have printer resources installed in the currently active system file.



2 **Click the icon representing the type of printer you want to use.**

3 **Select the appropriate zone from the AppleTalk Zones list.**

The zone name is highlighted. (This step may be optional, depending on your network configuration.)

4 **Select the appropriate printer name in the AppleTalk printer list.**

The printer name is highlighted.

5 **Click the close box.**

The Chooser dialog box disappears, and you are ready to print using `lpr` or any other print utility.

Printing from the Finder

Printing a document from the Finder in A/UX is the same as printing a document from the Finder in the Macintosh OS. You can also consult the manual that came with the application for more information on printing.

Follow these steps to print a Macintosh document from the Finder:

1 **Choose a printer.**

See the previous section.

2 **Select the document you want to print by clicking its icon.**

3 **Choose Print from the File menu.**

4 **Click OK in the Print dialog box.**

Printing with `lpr`

You can use the A/UX `lpr` ("line printer") command to print files created with text editors (such as `vi` or TextEditor). A/UX **spools** the files to be printed; that is, it stores them on disk until the printer is ready for them. The spooler prints them in the order in which they are received. When you print with `lpr`, you're using a default printer selected by the Chooser.

Before you can print to a remote printer, you must set up the `/etc/printcap` file, as described below. By default, `lpr` prints to a printer called "lp." `/etc/printcap` is shipped to the printer selected by the Chooser. If you intend to print to an AppleTalk printer using `lpr`, you do not need to change the `/etc/printcap` file.

Setting up access to a printer on a TCP/IP network

If your network includes other UNIX hosts that support `lpr`, you can set up access to a printer on a TCP/IP network by completing the steps below. If your printer is connected to a LocalTalk network, see Chapter 7 of *A/UX Essentials*.

1 **Log in as `root`.**

2 **Make a backup copy of the `/etc/printcap` file.**

3 **Open the file `/etc/printcap`.**

Use TextEditor, `vi`, or the text editor of your choice.

4 **Find the `RemoteHost` line.**

The `RemoteHost` line is in the section that refers to the remote UNIX line printer, as in the last line of this example:

```
#Remote UNIX line printer
#Change 'RemoteHost' to actual name of remote UNIX host
remote|remote line printer:\
        :lp=:rm=RemoteHost:sd=usr/spool/lpd/Remote
```

The remote host is the name of the computer that is the connected to the printer you want to access.

5 **Replace `RemoteHost` with the name of the remote UNIX system connected to the printer you intend to use.**

For example, if the name of the remote host is `tahoe`, edit the line to look like this:

```
:lp=:rm=tahoe:sd=usr/spool/lpd/Remote
```

Editing the line as shown above causes `lpr` commands sent to the remote host to be automatically printed out on the default printer of the computer named `tahoe`.

◆ **Note** If you don't know the name of the remote UNIX system connected to the printer you intend to use, see your network administrator. ◆

6 **If you wish, specify a printer other than the remote host's default printer.**

For example, to print files on the printer named `shasta`, which is directly connected to the remote host `tahoe`, insert the name of the printer on the `RemoteHost` line, as shown here:

```
:lp=:rp=shasta:rm=tahoe:sd=usr/spool/lpd/Remote
```

7 **Save the `/etc/printcap` file.**

8 **Choose Restart from the Special menu.**

The changes are put into effect.

Sending a file to a remote UNIX printer

To use the `lpr` command, follow these steps:

1 **Be sure that your printer is connected to AppleTalk.**

2 **Follow the directions in the previous section for choosing a printer from the Chooser.**

3 **Choose CommandShell from the Applications menu.**

4 **Enter this command:**

`lpr` *documentname*

This command sends the file *documentname* (the name of the file you want to print) to the printer.

To find out how many files are waiting in the print queue, use the `lpq` command (described in the next section).

◆ **Note** If the document is a `troff` document, you must run `troff` and then convert the output of `troff` to PostScript format by using a command such as `troff` *documentname*`|psdit|lpr`. ◆

Finding out how many files are waiting in the printing queue

A/UX spools printing jobs, placing them in a queue and then printing them on a first-come, first-served basis. If you share your printer and there are other jobs ahead of yours, those jobs are printed first. Your file is printed when your turn comes.

- **To get a list of the print jobs in the printing queue, enter this command:**

```
lpq
```

A list appears that looks something like this:

```
Rank      Owner     Job      Files        Total size
active    jms       3        filename     720 bytes
1st       pzp       4        filename     1500 bytes
2nd       alp       5        filename     2451 bytes
```

The first column shows the rank, the second shows the owner of the file sent to the print job, the third shows the identification number of the print job, the fourth column shows the filename (including its path), and the last column shows the size of the file.

Canceling a printing job

If you want to cancel a printing job before it's completed but after you've sent it to the printer, you can do so by using the `lprm`, which removes the most recent print job before it's finished printing. You can also cancel specific printing jobs by including the ID number of the job with the `lprm` command.

- **To cancel a printing job that hasn't finished printing, enter this command:**

  `lprm`

- **To cancel all printing jobs that haven't finished printing, enter this command:**

  `lprm-`

  To cancel a specific printing job before it's completed, you must know the ID number. To find out the ID number, use the `lpq` command described in the previous section.

- **To cancel a specific printing job, enter this command:**

  `lprm` *ID-number*

  You'll substitute the ID number with the number of the printing job you want to cancel. The following message appears as A/UX cancels the job:

  ID-number `dequeued`

  For more information about `lprm`, see `lprm`(1) in *A/UX Command Reference.*

Overview of available print facilities

You can use the Chooser and standard Macintosh print commands to print over an AppleTalk network, or you can print over a TCP/IP network using the `lpr` command. The method you use depends upon your computer network connection and the kind of printer you want to use. Table 3-1 shows the print commands available to you from your A/UX system.

Table 3-1 Print commands

Printer type	Printer language	Printing from Macintosh applications	Printing from UNIX applications	Notes
LaserWriter Plus LaserWriter II family* Personal LaserWriter family**	Serial PostScript	No	`lp` not `lpr`	
LaserWriter Plus LaserWriter II family* Personal LaserWriter family**	AppleTalk PostScript	Yes	`lp` and `lpr`	`lpr` printer selectable with Chooser
ImageWriter II ImageWriter LQ StyleWriter Personal LaserWriter LS	Serial QuickDraw	Yes	`lp` and `lpr` no `troff` support	
ImageWriter II ImageWriter LQ *(both require LocalTalk Interface Card)*	AppleTalk QuickDraw	Yes	`lp` and `lpr` no `troff` support	`lpr` printer selectable with Chooser

* Except LaserWriter II SC

** Except Personal LaserWriter LS

4 Sending and Receiving Mail

This chapter describes how to use an A/UX mail program to send **messages** to and receive messages from people using other computers running UNIX, how to save messages, and how to delete them. You can send a message to another user or group of users with accounts on your computer or to an account on a remote computer connected by a network.

The standard A/UX distribution is handled by `sendmail`—which works like a post office to route your mail—and must be set up by your network administrator. For details, see Chapter 9, "Setting Up Network Mail," in *A/UX Network System Administration.*

You can use two programs for reading and writing mail: `mail` and `mailx`. This chapter describes the use of `mailx`. To use the information in this chapter, you must have an alias set up so that every time you type `mail`, `mailx` runs. If this hasn't been done, you can set up the environment yourself, following the directions in the section "Creating a Mail Alias." For more information about mail, see Appendix B, "More About Mail." A/UX also supports third-party local-area mail programs as well as UNIX mail.

This chapter contains these sections:

- creating a mail alias
- sending mail
- receiving and reading mail
- exiting `mail`

You can read this chapter in its entirety or go directly to those sections of use to you.

Creating a mail alias

To use the information in this chapter, you must have an alias set up so that every time you type `mail`, `mailx` runs. `mailx` offers you more features and an easy-to-use interface. Check with your network administrator to see if this has been done. If this hasn't been done, you can set up the **environment** yourself following the directions below:

- **To create a `mail` alias in the C shell, enter this command in the `.cshrc` file in your home directory:**

```
alias mail mailx
```

- **To create a `mail` alias in the Korn shell, enter this command in the `.kshrc` file:**

```
alias mail=mailx
```

- **To create a `mail` alias in the Bourne shell, enter this command in the `.profile` file:**

```
mail () {
/usr/bin/mailx $@
}
```

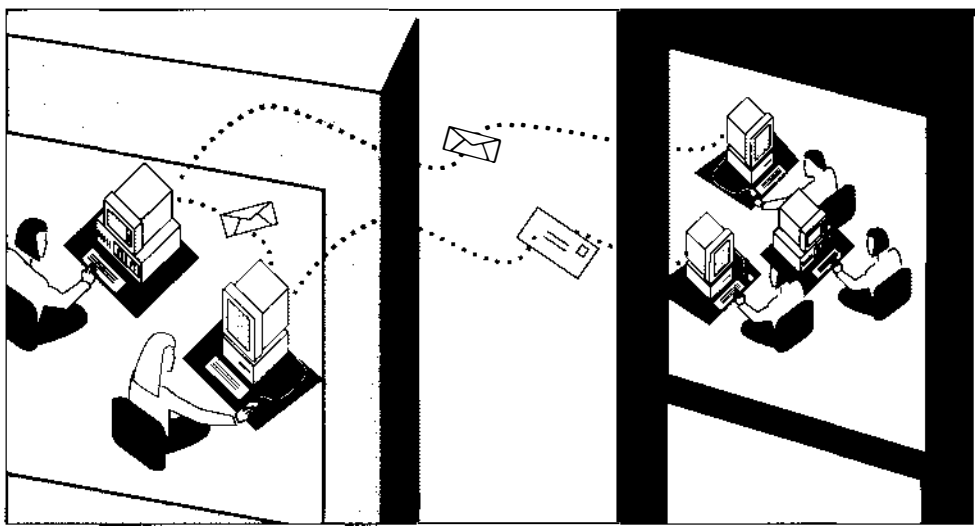
Once the mail alias is set up, all you need to use the mail program are the login names of the people you'll send mail to.

Sending mail

You can send electronic messages to and receive messages from other user accounts on your network. The procedure is the same for sending mail to accounts on both the local and the remote computers; the only difference is in the form of address.

Here are a few of the ways you can use `mail`:

- If you share your computer, you can send messages to the other users' accounts on your computer and they can send them to you.
- If you want to test mail, you can send mail to yourself.
- If your computer is connected to a network, you can send messages to and receive messages from the other people on the network who have accounts, even at sites in other countries.



Sending mail to users on your local system

To send a message to someone, you start up the `mail` program, type the message, and then send the message. The following subsections tell you how to send mail to one person and to more than one person on your local system.

Sending mail to one person

The following sequence of steps explains how to send mail to a single user on your local system. The steps work only if you are connected to other mail users on a network. If you're not already logged in to A/UX, log in under your own account name, which is your **login name**. If you're not sure if your system is set up for network mail, see your network administrator.

1 **Choose CommandShell from the Applications menu.**

A CommandShell window appears with the shell **prompt**.

2 **To send mail to the user at *address*, enter this command:**

`mail` *address*

Substitute the login name of someone on your computer for *address*. After you press RETURN, a subject line appears. This line is like the line in a memo that indicates the contents of the message.

3 **Enter a subject.**

4 **Type the first line of your message.**

5 **Press RETURN.**

Doing this starts a new line.

6 **Type the remaining line of your message, and after the last line, press RETURN.**

7 **Press CONTROL-D.**

The `mail` program prints `EOT` (for *end of text*).

When the user to whom you sent the message logs in, this message appears in a CommandShell window:

```
you have mail
```

The mail is the message you sent, as well as any other messages sent to that user account, preceded by a line identifying you (your login name) as the sender and the date and time the message was sent.

◆ **Note** The message `you have mail` appears only if the instruction has been set in the `.cshrc` file. ◆

Sending mail to more than one person

You can send mail to more than one login name on your system. Repeat the steps presented for sending mail to one person on your local system, but instead of listing only one user, list the login names of all the users you want to send mail to:

1 **To send mail to more than one login name, enter this command:**

`mail` *address1 address2 address3*

Replace the italicized words with the login names of the people you're addressing.

If you type several login names as part of the mail command, the message goes to all those users. For example, the command

```
mail lori jeff laura linda
```

will send the subsequently typed message to those four login names.

2 **Proceed as with any mail message.**

The rest of the process is the same as the procedure for sending mail to a single user on your local system.

Using an alias to send mail to more than one person

An **alias** is a name that stands for one or more login names. For example, if you were working on a project with several other people, you could define an alias for all the members of the project and then send mail to the alias. The definitions of the aliases are located in these places:

- `.mailrc`
- `/usr/lib/aliases`
- `/usr/lib/mailx/mailx.rc`
- `/etc/netgroup` (for NIS users only)

To use an alias on your local system, you must enter the following alias commands in your `.mailrc` file:

- **To tell `mail` that the alias stands for four members of your project, enter this command:**

  `alias` *aliasname address1 address2 address3 address4*

  Replace *aliasname* with the name that you want to use that will send mail to the people you specify.

  For example, if Lori, Jeff, Laura, and Linda are members of a project and you want to give them an alias, you could name the alias `project`. The command would look like this:

  ```
  alias project lori jeff laura linda
  ```

- **To send mail to the four people you named in the alias, enter this command:**

  `mail` *aliasname*

  So, to send mail to the members of `project`, you would type this command:

  ```
  mail project
  ```

  Mail is sent to Lori, Jeff, Laura, and Linda.

  This command also provides a convenient alias for someone whose user name is hard to type. For example, if a user named Throckmorton Quixote had the login name `quixote`, you might use

  ```
  alias tq quixote
  ```

  This alias allows you to send mail using the shorter name, `tq`.

Sending mail to users on other systems

In addition to sending mail to users on your own computer, you can use the mail command to send mail to users on other computers. This section tells you how to form network mail addresses, but the ability to send network mail depends greatly on the extent to which your system has been customized for network access. See your nework administrator for details.

The procedure for sending mail to a user on another computer is the same as for sending mail to a user on your local system. The only difference is in the value of *address*.

Sending mail to a user on the TCP/IP network

If your system is connected to a TCP/IP network, you can send mail to users whose systems are also connected to your network. If your network is also connected to other networks by, for example, a gateway, you can also send mail to users on other networks.

Before you can use this facility, be sure that your network administrator has connected your system to the network and configured it properly. For further information, see Chapter 9, "Setting Up Network Mail," in *A/UX Network System Administration*.

1 **To send mail to a user on a TCP/IP computer over the network, enter this command:**

`mail` *address@rhost*

Replace *address* with the login name of the person you want to send mail to, and replace *rhost* with the name of the remote host. For example, if you wanted to send mail to John whose computer is named `tahoe`, you would enter this command:

```
mail john@tahoe
```

2 **Proceed as with any mail message.**

The rest of the process works the same way as the procedure for sending mail to users on your local system. See those steps in "Sending Mail to One Person" earlier in this chapter.

If you log in to a computer that is on another TCP/IP network that is connected to your local area network, your network is probably running the name **server** to look up host names. The name server does this by dividing the networks into domains of computers that organize the addresses, much like street names organize the houses in a city. When the name server is being used, include the domain name as part of the address. If you wanted to send a message to John on a computer whose host name is `tahoe` in the domain `alpha.com`, you would enter this command:

```
mail john@tahoe.alpha.com
```

Sending mail to more than one person on the TCP/IP network

If you want to send mail to more than one person on the same remote computer over the TCP/IP network, you must type each person's address separately. The example below demonstrates how to send mail to more than one person.

1 **To send mail to users with login names** `linda`, `tony`, `sharon`, **and** `laura`, **enter this command:**

`mail linda@`*rhost* `tony@`*rhost* `sharon@`*rhost* `laura@`*rhost*

Replace *rhost* with the name of the remote host.

2 **Proceed as with any mail message.**

Sending mail to a user through UUCP

Before you can use this facility, your system administrator must set up your system for UUCP. For further information, see Chapter 8, "Setting Up the UUCP System," in *A/UX Network System Administration.*

Sending mail to someone over UUCP is just like sending mail to anyone else, except that the form of the address is different.

1 **To send mail to a user through UUCP, enter this command:**

`mail` *rhost*`\!` *address*

Replace *address* with the login name of the person you want to send a message to, and replace *rhost* with the name of the remote host.

For example, if you wanted to send mail to John on `tahoe` through a UUCP connection, you would enter this command:

```
mail tahoe\!john
```

You need the backslash (`\`) only if you're working in the C shell.

2 **Proceed as with any mail message.**

The rest of the process works as in the procedure for sending mail to users on your local system.

UUCP can store and forward mail from computer to computer, so you can send messages all over the world. In this case, you need to use a path that specifies each computer along the way as shown here:

`mail` *rhost1*`\!` *rhost2*`\!` *rhost3*`\!john`

If your computer uses an intermediate system (any system between your own and the destination), the UUCP system sends the mail to the intermediate systems and then to the user on the remote system. So if you were sending mail to John on `tahoe` through an intermediate system called `placer`, you would enter this command:

```
mail placer\!tahoe\!john
```

Sending mail to more than one person through UUCP

If you want to send mail to more than one person on the same remote computer over UUCP, you must type each person's address separately. The example below demonstrates how to send mail to more than one person.

1 **To send mail to users with login names `don`, `vicki`, `bob`, and `jackie`, enter this command:**

`mail` *rhost*`\!don` *rhost*`\!vicki` *rhost*`\!bob` *rhost*`\!jackie`

Replace *rhost* with the name of the remote host.

Use the backslash (\) before the exclamation mark if you are working in the C shell.

2 **Proceed as with any mail message.**

Canceling a message

If you decide not to send a message while you're writing it, you can cancel the message by doing this:

1 **Press CONTROL-C.**

This sends an interrupt signal to the system. When you press CONTROL-C, `mail` displays the message

```
(Interrupt--one more to kill letter)
```

2 **Press CONTROL-C again.**

This action cancels the message. A/UX stores the canceled message in the file `dead.letter` in your home directory. If you want to keep a canceled message, save it. Each message that is sent to the `dead.letter` file overwrites the previous message.

You can delete these files by dragging them to the Trash and using the Empty Trash command in the Special menu. To learn about using the `dead.letter` file in your home directory, see `mailx`(1).

Receiving and reading mail

Your incoming messages are stored in a file called `/usr/mail/`*login-name*. If a message is received while you are not logged in, the system waits until you log in to post the message in your CommandShell window. If you receive a message while you're working, you'll see this message displayed in the CommandShell window:

```
you have mail
```

The A/UX system is automatically set to check every 60 seconds to see if new mail has arrived for you. For Korn and Bourne shell users, this check is set in the `/etc/profile` file; for C shell users, this check is set in `/etc/cshrc`.

Reading mail

When you want to read your mail, just open a CommandShell window and type `mail`.

1 **Choose CommandShell from the Applications menu.**

This displays a CommandShell window with a **shell prompt**.

2 **Enter this command:**

```
mail
```

If you don't have any mail, this message appears:

```
no mail for name
```

The name in italics is your login name.

When you have mail, each message is summarized in a numbered list that shows the author's login name, the subject, and the date it was sent.

3 **Press RETURN or enter the number of the message you want to read.**

Pressing RETURN causes `mail` to display the first message. If you enter the number of a message, A/UX displays that message. For example, if you have 11 messages and you want to read message 7, enter `7`. The header at the top of a message tells you who sent the message, when it was sent, and what it's about. A question mark prompt (`?`) appears at the end of the message. After you read the message, press RETURN, and the next message is displayed.

Deleting a message

You can delete one or more messages when you finish reading them. You can also restore deleted messages before you quit `mail`.

- **To delete a specific message, enter this command:**

  `d` *n*

  For example, to delete message 3, you would enter this command:

  ```
  d 3
  ```

  The prompt returns, indicating that the message is deleted.

- **To delete a specific range of messages, enter this command:**

  `d` *n-m*

  n stands for the first number and *m* stands for the last number in the range you want to delete.

- **To delete all messages, enter this command:**

  ```
  d^ - $
  ```

  `^` means begin and `$` means end.

- **To restore ("undelete") a specific deleted message, enter this command:**

  `u` *n*

- **To "undelete" the last message you deleted, enter this command:**

  ```
  u
  ```

- **To verify that a message was deleted or undeleted, enter this command:**

  ```
  h
  ```

  The list will show the messages that are still available.

Saving a message

If you want to save a message in a file, you can tell `mail` to save it and give it a **filename**. If the filename you give doesn't exist, the system creates the file and puts the message in it:

- **Enter the following command:**

  `s` *messagenumber filename*

  For example, if you enter `s 4 memo.lori`, the `mail` program saves the fourth message on the list in a file named `memo.lori`. If you save another message to the same file, it is appended at the end of the first message you saved.

Exiting `mail`

There are two ways to exit the `mail` program. The first way deletes any files you have marked for deletion. The second way ignores any delete commands and restores mail to the way it was when you started up the `mail` program.

- **To quit `mail`, enter this command:**

  `q`

  The `q` command (for *quit*) removes messages you marked for deletion, places any unread messages on the unread list, and ends the `mail` program. If you exit `mail` using `q`, you cannot restore a message you have deleted.

- **To quit `mail` and keep all your messages, enter this command:**

  `x`

  The `x` command (for *exit*) does not remove the messages marked for deletion, places any messages you read back on the new **message list**, and ends the `mail` program.

To learn more about mail, see Appendix B in this manual.

5 Using TCP/IP

If your computer is connected to a TCP/IP network, you can use other computers as if they were your own. This chapter tells you how to find out what computers you can access, how to log in to those computers, how to use their services, and how to copy and transfer files. You'll also learn about network permissions. If you're not sure what capabilities you have or what computers are available to you, see your network administrator.

This chapter contains these sections:

- what TCP/IP adds to a network
- finding out which computers can be accessed
- working with network permissions
- connecting to other computers
- running commands on other computers
- copying files between systems

You can read this chapter in its entirety or go directly to those sections that are of use to you.

What TCP/IP adds to a network

TCP/IP (Transmission Control Protocol/Internet Protocol) allows computers to communicate with each other over an Ethernet cable. TCP/IP allows you to use standard UNIX commands to gain access to remote computers. Figure 5-1 illustrates an A/UX system connected to a TCP/IP network.

Figure 5-1 shows that a user logged in to the A/UX system can print files on remote printers, copy files to and from remote computers, and execute commands on remote computers.

For the purposes of this chapter, a **local area network (LAN)** is a group of computers that are connected for the purpose of sharing resources, are typically joined by a single transmission cable, and are located within a small area such as a single building; a **wide area network (WAN)** is a system of interconnected local area networks that spans a wide geographical area and may include different types of computers and operating systems. Typically, computers on a wide area network communicate through high-speed data lines, while computers on a local area network are connected through Ethernet or another type of cable. Figure 5-2 shows the two kinds of networks.

Before you can use TCP/IP networking, you need a login account on at least one other computer on the network, or you can log in as Guest. Your system administrator usually sets this up for you on the remote computer. If the network is running NIS and your name is in the NIS database, you may be able to log in to any computer on the network. When you do, you'll see a message that tells you there is no home directory and you're being logged in with / (slash) as your directory.

The computer that you log in to using the standard A/UX login procedure is your local system, and any computers that you access via the network commands are remote systems. The commands listed in the section "TCP/IP Networking Commands" are those you'll use most often when you're working with TCP/IP. To use these commands, you must have an account on a remote computer.

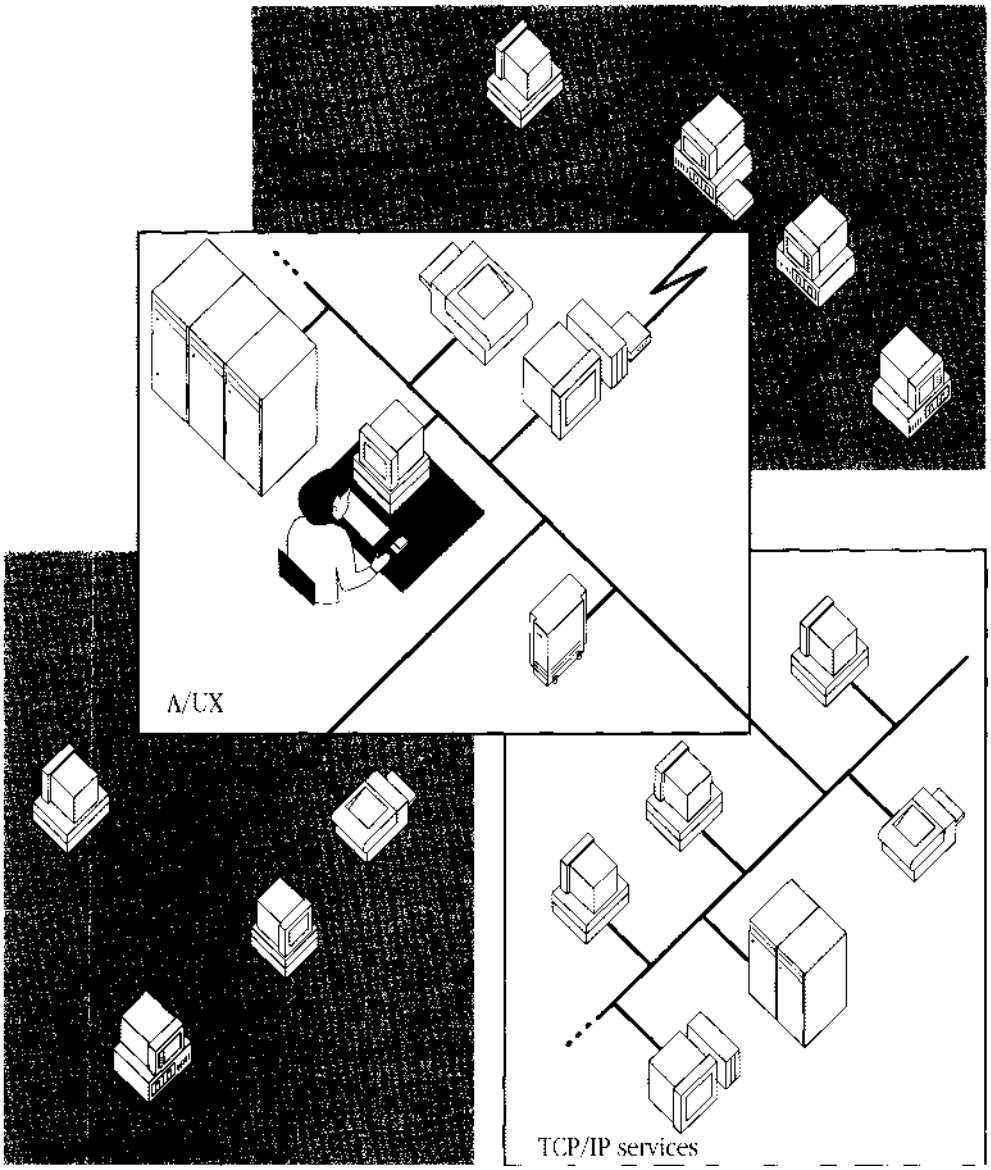


Figure 5-1 A/UX connected to TCP/IP services

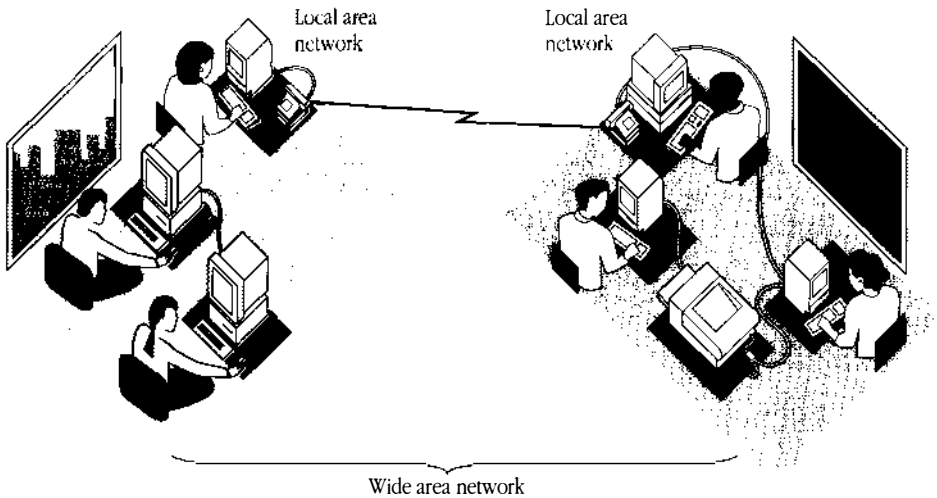


Figure 5-2 Wide area and local area networks

TCP/IP networking commands

The TCP/IP networking commands explained in this chapter enable you to do the following:

- Log in to different systems:

  `rlogin` connects you to a remote system that is also running a version of the TCP/IP networking software. Then you can use applications on the remote system as if they were on your local computer. (See "Using `rlogin` to Log In to a Remote Computer" in this chapter.)

  `telnet` connects you to computers on the same local area network as your computer or to computers on a wide area network. The computer may be running a UNIX-based operating system or an operating system that is not based on UNIX. (See "Using `telnet` to Log In to a Remote Computer" in this chapter.)

- Copy files between different systems:

  `rcp` allows you to copy files between UNIX computers on the local TCP/IP network. (See "Using `rcp`" in this chapter.)

`ftp`, more widely available than `rcp`, allows you to transfer files to and from a remote computer, regardless of its operating system. The remote computer may be on the same local area network as your computer or it may be on a wide area network. (See "Using File Transfer Protocol" in this chapter.)

- Print on printers located on remote computers using the `lpr` command. (See "Running Commands on Other Computers" in this chapter.)
- Run commands remotely:

  `remsh` connects to the computer you specify, runs the command you specify, and ends the session when the command has ended. (See "Running Commands on Other Computers" in this chapter.)
- Send and receive electronic mail (See Chapter 3).

Finding out which computers can be accessed

The following two commands check the status of remote computers on the local network:

`ruptime`	Give a status line for each computer on the local network. `ruptime` stands for "remote uptime."
`rwho`	Give the login names of users who are logged in at each computer on the local network.

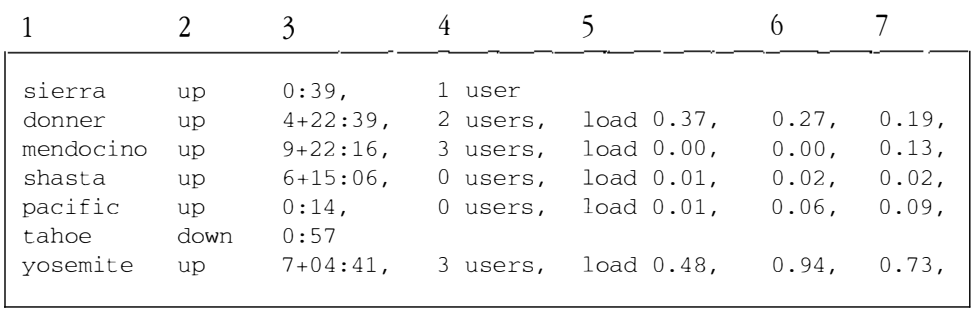
◆ **Note** When network traffic is heavy, some system administrators may decide not to run the **programs** that `ruptime` and `rwho` depend on. This means that `ruptime` and `rwho` may help you determine if a system is up or down but won't necessarily give you a definitive answer. You can tell a computer isn't running if you get this message: `no host !?!` ◆

■ **To check the status of computers connected to the network, enter this command at the shell prompt:**

```
ruptime -a
```

The `-a` counts all logged-in users, no matter how long they've been idle.

A status line shows each computer connected to the network. It looks like this:



1 Name of each system answering `ruptime` requests

2 Tells you whether or not the system is up

3 Indicates how long the system has been up or down in days, hours, and minutes, respectively. (For example, 4+22:39 means 4 days, 22 hours, and 30 minutes.)

4 Represents how many users are currently logged in to the system

5, 6, 7 Shows the load average (the average number of processes on that system) over the past 1, 5, and 15 minutes, respectively.

Figure 5-3 Network computer status lines

■ **To see the listing for a specific computer, enter this command:**

```
ruptime | grep computer name
```

For example, if you wanted to see the listing for a computer named `donner`, you would use the command

```
ruptime | grep donner
```

This returns the status line

```
donner up 4+22:39, 2 users, load 0.37, 0.27, 0.19
```

For more information about `ruptime`, see `ruptime`(1N).

- **To list all the users logged in on any computer on the local network and the name of the computer they're logged in to, enter this command:**

```
rwho -a
```

If a user hasn't used the system for a minute or more, `rwho` reports this as idle time.

You may also use the `rwho` command with `grep` to see if a particular user is logged in or who is using a particular remote system. For example, the command

```
rwho | grep sierra
```

returns the login names of every user currently logged in on the remote system `sierra`.

◆ **Note** Not all hosts answer these requests. A computer must be running the appropriate software, such as `rwhod`, in order to answer a request. ◆

Working with network permissions

Any system on the network may have an `/etc/hosts.equiv` file that network administrators can set up. When set up, this file contains lines that specify the names of systems whose users are allowed to access the local system without a password. If you're remotely accessing this system from a "friendly" host that is listed in the local file, and if you have an entry in `/etc/passwd` on the local system, you are allowed to access the system using the `rcp`, `rlogin`, and `remsh` commands.

For security reasons, some network administrators don't set up the `/etc/hosts.equiv` file.

- **To see if the network administrator has set up the `/etc/hosts.equiv` file, enter this command:**

```
cat /etc/hosts
```

If the only output is `loop`, the `/etc/hosts.equiv` file is not set up.

Setting up the `.rhosts` file

If the `/etc/hosts.equiv` file is not set up, you can set up a `.rhosts` file in your login directory to allow explicit access to your login account. This file is most often used when your user name is different on the local and remote systems (as in the example on page 5-9 where Bill has two login names, `bill` and `wrn`). You could also include login names of other users, which would allow them unrestricted access to your account if they logged in using the `-l` option to the `remsh` or `rlogin` commands.

◆ **Note** When you include a line in your `.rhosts` file that allows a certain user (or all users from a certain system) access to your account, in effect you're giving that person your password. Normally you should restrict this file to your own login names. ◆

■ **To set up the `.rhosts` file, enter a line such as this in a file called `.rhosts` in your home directory:**

hostname [*username*]

The parts are interpreted as follows:

hostname	The name of a remote system followed by any number of spaces or tab characters.
username	The name of a user on *hostname*. This argument is optional. If the *username* parameter is not stated, anyone from *hostname* is allowed access to your account.

Each line of this format indicates that *username* from *hostname* has permission to access this particular account.

◆ **Note** You can set up the `.rhosts` as shown above if your network is not running the name server. If your network is running the name server, see your network administrator for the correct way to specify the *hostname*. ◆

The `.rhosts` file is most commonly used by users with login names that differ across systems. If Bill has the login name `bill` on `mendocino` and the login name `wrn` on `sierra`, his `.rhosts` file on `mendocino` should include this line:

```
sierra wrn
```

His `.rhosts` file on `sierra` should include this line:

```
mendocino bill
```

From his `mendocino` account, he can then use the command format

```
rlogin sierra -l wrn
```

or

```
remsh sierra -l wrn command
```

If he is logged in to `mendocino`, he can use the format

```
rlogin sierra -l bill
```

or

```
remsh mendocino -l bill command
```

Network users can also share accounts. For example, Bill has the login name `bill` on the local system `redwood`. Another user, Ramona, has the login name `ramona` on remote system `sierra`. Ramona is logged in to her account on `sierra`. She wants to run the `date` command remotely on `redwood`, but she doesn't have an account on that computer.

If Bill has a `.rhosts` file containing the line

```
sierra ramona
```

Ramona can use the command

```
remsh redwood -l bill date
```

Connecting to other computers

There are two approaches to logging in to a remote computer: using the `rlogin` command or using the `telnet` command. Both allow you to use applications on the remote system as if it were your local computer. When you're connecting to another computer that is on your TCP/IP network, you can use `rlogin` or `telnet`. If you want to connect to a computer that runs an operating system that is not based on the **UNIX operating system** but has `telnet` available, use `telnet`. Both work on wide area as well as local area networks. See the section below that meets your needs.

Using `rlogin` to log in to a remote computer

You use the `rlogin` (remote login) command to log in to a remote computer. The `rlogin` command logs you in to the remote system so that you can work from your local system just as if you were using a terminal attached directly to the remote system. When you use this function from your local computer, it is said to become a **virtual terminal.** Figure 5-4 shows a remote login.

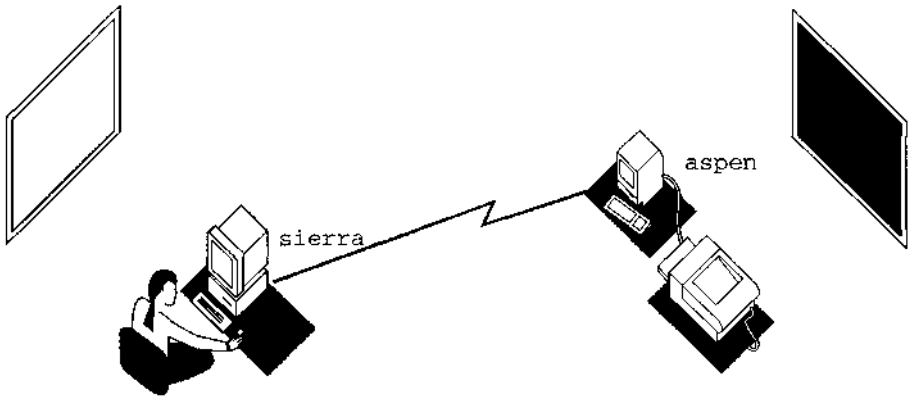
◆ **Note** To see a list of computers you can log in to, you can enter the command `cat /etc/hosts` ◆

■ **To log in to a remote computer, enter this command:**

`rlogin` *rhost*

rhost (for "remote host") is the name of the remote computer.

The remote system reads your remote login file before displaying a shell prompt. When the shell prompt appears, you are connected to the remote computer and are working in its **shell**.



Remote log in command: `rlogin aspen`

Figure 5-4 Logging in to a remote computer

◆ **Note** The login file that is read depends on the shell being used. Depending upon the shell you're using, you may want to customize the remote environment with these files so that it looks like your normal environment: If you're using the Korn shell, you may need to modify the `.profile` file or the `.kshrc` file; in the Bourne shell, you may need to modify `.profile`; in the C-shell, you may need to modify `.cshrc`. You can change the default shell with the `chsh` command if you're logged in to a system that supports it. If that command is not available, contact your network administrator in order to change the default. ◆

If you can't log in to the remote system, it may be that you don't have an account on that system or that your network administration files are not set up properly. In either case, see your network administrator.

Once you have logged in, the command

```
hostname
```

returns the name of the remote computer.

- **If your login name is not the same on the local and remote systems, or if you have access to another user's account, enter this command:**

  `rlogin` *rhost* `-l` *username*

  Replace *username* with your login name on the remote system. You should also create a `.rhosts` file in your remote login **directory** to establish equivalence between the accounts on the two systems. See "Working With Network Permissions" earlier in this chapter for more information. Some remote systems are set up to automatically establish an equivalence, and you don't need a `.rhosts` file. The benefit of having a `.rhosts` file is that you don't need a password and you can run `rcp`. You can tell that you need a `.rhosts` file if you are asked for your password when you run `rlogin`.

Using the remote system

When you're connected to a remote A/UX system through `rlogin`, everything you do in the CommandShell window is executed on the remote computer. The example below demonstrates how to create a directory on the remote computer and move a file into it.

1 **To create a directory called `Test` on the remote computer, enter this command:**

```
mkdir Test
```

2 **To change to the directory you just created, enter this command:**

```
cd Test
```

3 **To create a file called `jelly`, enter this command:**

```
vi jelly
```

4 **Add text to the file.**

For the purposes of the tutorial, it doesn't matter what you type here.

5 **To leave the file, enter this command:**

```
:wq
```

6 **After you write and quit the file, to display it on your screen, enter this command:**

```
cat jelly
```

7 **To direct the output of the `cat` command into a new file, enter this command:**

```
cat jelly > jam
```

8 **To echo the host name of your current host, enter this command:**

```
hostname
```

The output is directed into `jam` on the remote computer. To make a copy of `jelly` on your local computer, use the `rcp` network command. (See "Using `rcp` to Copy Files" later in this chapter.)

Ending the `rlogin` *session*

On most systems, the commands `exit` and `logout` end the `rlogin` session. If this approach does not work, do the following steps:

1 **Press RETURN**

2 **Enter this command:**

```
~.
```

The command is a tilde followed by a period.

This sequence disconnects you from the remote system and returns you to your local account. This sequence also halts or stops the `rlogin` process if you enter it before you are logged in to the remote account. You may type the tilde-period sequence at any time on the remote system (even if you are running an application there), and it will return you to the local computer. You can also close the CommandShell window to stop `rlogin`.

◆ **Note** If you use `rlogin` to log in to a remote system and start some jobs in the background, the jobs are canceled if you log out before the job is complete. To prevent the cancelation, use `nohup` when you start the jobs. For example, use

`nohup` *command* `&`

rather than just

command `&` ◆

See Appendix A for solutions to some problems you could encounter using `rlogin`.

Using `telnet` to log in to a remote computer

The `telnet` command is a user interface to the TCP/IP standard TELNET Protocol. (Most computers implementing the TCP/IP protocols recognize FTP and TELNET.) The `telnet` command is used to establish a connection with another host. It sets up a remote login connection similar to `rlogin`, simulating a terminal with a physical connection to the

remote host. The `telnet` command is useful for connecting to computers that run a non-UNIX operating system and do not support `rlogin`.

Once you establish a connection with the remote system, `telnet` enters input mode, and the text you type is sent to the remote host. When you have logged in to the remote system, you see the regular prompt for that computer, and you can carry out a normal login session.

Logging in

To connect to a remote system using `telnet`, you specify the remote system on the command line:

- **To connect to a remote system using `telnet`, enter this command:**

  `telnet` *rhost*

  Replace *rhost* with the name of the remote computer you want to use.

Learning the current status

After you have logged in to a remote host with `telnet`, you can learn the current `telnet` status by doing the following:

1. **Press CONTROL-] (right bracket) to enter command mode.**

2. **Press RETURN.**

3. **Enter this command:**

   ```
   status
   ```

   This message appears:

   ```
   Connected to rhost
   Escape character is ^]
   ```

4. **To exit status mode, press RETURN.**

Logging out

To close your current `telnet` connection but remain in `telnet` **command mode**, do the following:

1 **Press CONTROL-] (right bracket) to enter command mode.**

2 **Press RETURN**

3 **At the prompt, enter this command:**

```
close
```

Now you may use the commands in the `telnet` commands section below to open a connection to a different host computer or to close the current session and quit the `telnet` program. See Appendix A for solutions to some problems you could encounter using `telnet`.

Using `telnet` *commands*

The following `telnet` commands are available to you if you are in command mode or if you precede them with the `telnet` **escape character** (by default, CONTROL-]). You may abbreviate each command by entering enough of the command to distinguish it from the others. (For example, to abbreviate the `close` command, you would enter `cl` to distinguish it from the `crmod` command.)

`close`	Close this session and return to command mode.
`quit`	Close any session that is currently open and exit `telnet`.
`set escape [`*char*`]`	Set the telnet escape character to the specified character. **Control characters** are specified as ^ followed by a single letter (e.g., CONTROL-X is ^X).
`status`	Show the current status of `telnet`.
`options`	Toggle viewing of `telnet` options. When this mode is enabled, all `telnet` options are displayed. Options sent by `telnet` are displayed as sent, and options received from the `telnet` server are displayed as received.

`crmod`	Toggle carriage return mode. When this mode is enabled, any return characters received from the remote host are mapped into a return and a line feed. This mode does not affect those characters you type, only those received. This mode is required for some hosts that like to ask the user to do local echoing.
`?` [*command*]	The help command. When you don't include a command name, `telnet` prints a help summary.

Running commands on other computers

The `remsh` command connects to a computer, runs a command, and ends the connection when the command has ended.

Using `remsh`

You would use the `remsh` command when you want to execute a command on a remote UNIX computer without actually logging in to the remote computer.

◆ **Note** To use `remsh`, your network administrator must have set up the `/etc/hosts.equiv` file on your computer and on the remote computer, or you can configure `.rhosts` on the remote computer. (See "Working With Network Permissions" earlier in this chapter.) This section assumes that the login names are identical across the systems. If you need help, see your network administrator. ◆

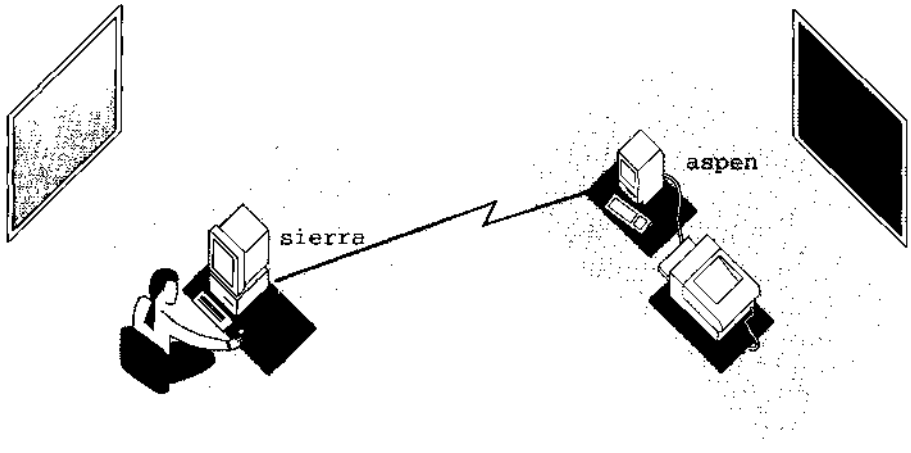
■ **The format of the `remsh` command is**

`remsh` *rhost command*

Replace *rhost* with the name of the remote computer, and replace *command* with a single command or a series of commands.

Figure 5-5 shows how you would print a local file (called `report`) on a printer connected to a remote computer called `aspen`. The local `cat` process **pipes** the input file (`report`) to the `remsh` command, which in turn sends the file to `aspen` and executes the remote `lpr` **process**. This is how the command looks:

```
cat report | remsh aspen lpr
```



Print command: `cat report | remsh aspen lpr`

Figure 5-5 Printing a local file on a remote printer

Note that the `lpr` command is executed on the remote CPU `aspen`. It is important to understand the two separate execution environments; for example, the following command pipes the remote file `report` on `aspen` to a local printer:

```
remsh aspen cat report | lpr
```

You must have a login account (or access to another user's account) on the remote system. (See "Working With Network Permissions" earlier in this chapter for more information.)

You cannot use the `remsh` command successfully to run an **interactive program** like `vi`. For that, you must use `rlogin` or `telnet`.

- **If your login name differs across systems, or if you're logging in to someone else's account on the remote computer, enter this command:**

  `remsh` *rhost* `-l` *username command*

Ending the `remsh` session

The `remsh` command normally ends when the remote command ends. If a problem arises and the command doesn't end, simply close the CommandShell window and open another one. See Appendix A for solutions to some problems you could encounter using `remsh`.

Copying files between systems

There are two commands you can use to copy files between systems: `rcp` and `ftp`. `ftp` is more generally available on more kinds of computers than `rcp`, but once `rcp` is set up, it's easier to use.

Using `rcp`

The following subsections tell you how to use `rcp` to copy files and to copy directories. You can also find information on shorthand notations and how to end the `rcp` session.

Using `rcp` to copy files

With `rcp`, the first argument is always the file to be copied and the second argument is always the file where the copy will reside. Either or both of these files may be remote or local. The format is similar to that of the A/UX `cp` command. The `rcp` command uses this format:

`rcp` *source-file dest-file*

source-file is the file being copied and *dest-file* is the destination file (into which a copy of the source file is placed).

- **To copy files from a local system to a remote system, enter this command:**

  `rcp` *source-file rhost*`:`*path*

path is the path to the file (or directory) on the remote computer. The *path* is interpreted relative to your home directory on the remote computer, unless you enter the **absolute pathname** (that is, the path relative to the root directory).

- **To copy and rename a file from your remote account to your local directory, enter this command:**

  `rcp` *rhost*`:`*path dest-file*

  The *path* should end in a filename.

  For example, to copy the remote file `pacific` from the system `sierra` to your **current directory** and rename it `atlantic`, you'd enter the command

  ```
  rcp sierra:pacific atlantic
  ```

  `pacific` is copied from your login directory on `sierra` and renamed `atlantic` in your current directory on the local system.

▲ **Warning** If you copy a file from another host to a file of the same name in your local system, the local system file will be overwritten by the contents of the remote file. You will not receive a warning that this is going to happen. ▲

- **To copy a file from one remote computer to another, enter this command:**

  `rcp` *rhost1*`:`*path1 rhost2*`:`*path2*

  For example, to copy the `boulder` file in Brian's directory on `sierra` to Lynn's directory on `canyon`, you'd enter this command:

  ```
  rcp sierra:/users/brian/boulder canyon:/users/lynn/boulder
  ```

- **To copy a file to another user (or another login name of yours on *rhost*), enter this command:**

  `rcp` *rhost*`.~`*username*

  username is the name of another user (or another login name of yours) on *rhost*.

Using `rcp` *to copy directories*

When using `rcp`, the `-r` option performs a recursive copy, which means that the directory structure is copied. Both the source and destination arguments must end in a directory name or shorthand notation. See the next section for more information about shorthand notation.

- **To create a new directory below the local pathname and copy the contents of a remote directory into it, enter this command:**

  `rcp -r` *rhost*`:`*spath dpath*

  spath is the source path and *dpath* is the destination path.

  The name of the new local directory is the same as its remote name. For example,

  ```
  rcp -r sierra:Test .
  ```

  creates a `Test` directory below your current (local) directory. The local `Test` directory contains copies of every file and subdirectory contained in the remote `Test` directory on `sierra`.

- **To copy the contents of a remote directory to a directory with a different name on the local system, enter this command:**

  `rcp -r` *rhost*`:`*path newname*

◆ **Note** In the situation described above, `rcp` creates the directory for you, so you don't have to create a directory before copying a file into it. ◆

Shorthand notations

When *rhost*`:`*path* is the second (destination) argument with `rcp`, it may end in an existing directory name. In that case, the local file is copied into that directory and retains its local name.

Ending the `rcp` *session*

The `rcp` command ends when the file copy is complete. If a problem arises and the command doesn't end, simply close the CommandShell window and open another one or press CONTROL C. See Appendix A for solutions to some problems you could encounter using `rcp`.

Using File Transfer Protocol

The `ftp` command is a user interface to the TCP/IP standard File Transfer Protocol (FTP). (Most computers implementing the TCP/IP protocols implement FTP and TELNET.) The `ftp` command was developed to allow the reliable transfer of files to and from a remote network site, regardless of the hosts' operating systems, in contrast to `rcp`, which usually requires that both hosts are running a UNIX-based operating system.

Using `ftp` *to copy files*

Unlike the `rcp` command, `ftp` is interactive; it keeps track of your current directory on both local and remote computers, and it interprets **pathnames** relative to your current directory on either computer. It does not interpret pathnames relative to your login directory, unless that is your current directory.

1 **To invoke `ftp`, enter this command:**

`ftp` *rhost*

The `ftp` program immediately attempts to establish a connection to an `ftp` server on *rhost* and prompts for your login name and password on the remote computer. You'll see something like this:

```
Connected to rhost.
220 rhost FTP server (version 4.1 Sun Jan 13 05:35:29 PST
1991) ready.
Name (rhost:theresa):
331 Password required for theresa.
```

```
Password:
230 User theresa logged in.
ftp>
```

In this example, `theresa` is your login name on the local host.

2 **If you have a different login name on the remote system, enter the remote login name in response to the prompt**

`Name (`*rhost*`:theresa):`

If you do not have a different login name on the remote system, skip to Step 3.

3 **Press RETURN.**

If the remote account has a password, you will be asked to supply it.

Using `ftp` *to transfer files*

Files may be transferred across the network using the `get` and `put` commands. The `get` command copies a file from the remote system to your local system, while `put` copies in the reverse direction. Both commands take filename arguments that may be either complete pathnames or pathnames relative to your *current* directory.

- **To copy a file from the remote system to your local system, enter this command:**

  `get` *remote-source local-dest*

- **To copy a file from your local system to the remote system, enter this command:**

  `put` *local-source remote-dest*

  If you specify the filename as - (the minus sign), `ftp` uses the **standard input** for reading or **standard output** for writing.

  For a list of commands to use in transferring files, see the sections "`ftp` Commands" and "File Transfer Parameters" in Appendix D, "More About `ftp`."

Using directories on the remote computer

To learn what files are in the current directory on the remote computer, use the `dir` command. `ftp` responds with a listing identical to that given by the A/UX command, `ls -al`. You'll see something like this:

```
drwxr-xr-x 5   vz   16655   Jul   26   10:11   forms
-rw-rw-r-- 1   vz    7389   Aug   23   13:34   program
-rw-rw-r-- 1   vz      57   Sep   27   12:03   new.01
```

- **To change your current directory on the local computer, enter this command:**

  `lcd` *directory name*

- **To change your current directory on the remote computer, enter this command:**

  `cd` *directory name*

  In the example directory listing above, the command

  `cd forms`

  changes the current remote directory to the `forms` directory.

  For a list of `ftp` commands for using directories, see the section "`ftp` Commands" in Appendix D, "More About `ftp`."

Logging out

- **If you want to close a session with a particular computer but remain in `ftp`, enter this command:**

  `close`

- **To close the current connection and exit `ftp` altogether, enter this command:**

  `quit`

6 Sharing Files Over the AppleTalk Network

This chapter tells you how to share files and access them over the AppleTalk network—that is, how to share files with other Macintosh users who may or may not be running A/UX. This chapter also tells you how to control who has access to the files you want to share. The process of sharing files over the AppleTalk network in A/UX is similar to sharing files from your Macintosh OS. You can make files available to coworkers without having to print documents or copy files to a floppy disk. Files that you put into a shared **folder** on your computer can be retrieved by people you designate to have access to that folder. Much of this information also appears in your *Macintosh Networking Reference*. If you want to share UNIX files with other UNIX users, see Chapter 7, "Sharing Files With NFS."

This chapter includes these sections:

- about file sharing
- starting file sharing
- sharing a folder or disk
- naming specific people to share folders and disks with
- sharing a folder or disk with specific people

- setting up your computer for remote accessing
- changing your password
- giving folder ownership to someone else
- changing a user's password
- monitoring file-sharing activity
- reducing access to your shared files
- about accessing information on other computers
- connecting to a shared disk
- connecting automatically
- using and storing files on a shared disk
- disconnecting from a shared disk
- changing your password to access a shared disk
- affecting access to folders
- accessing your own computer remotely

Although you may find it useful to go through this entire chapter in sequence, you may also use it as a reference work by going directly to the section that contains the material you need.

About file sharing

Sharing files using A/UX is the same as sharing files in the Macintosh environment. All of the extensive capabilities available to you as a Macintosh user are available to you as a user of A/UX. You can set the file-sharing permissions you want, define users and groups, share folders and disks, and grant and deny access, all using the Macintosh interface. For your convenience, this manual offers you the same information on sharing and accessing files as you will find in your *Macintosh Networking Reference,* with additional information that affects the A/UX user. For basic information about setting UNIX permissions, see Chapter 5 in *A/UX Essentials.* For more in-depth information about permissions, see *A/UX Local System Administration.*

This section provides an overview of how you can use Macintosh **file sharing** in A/UX. Subsequent sections in this chapter explain the specific steps you need to take to start and use file sharing.

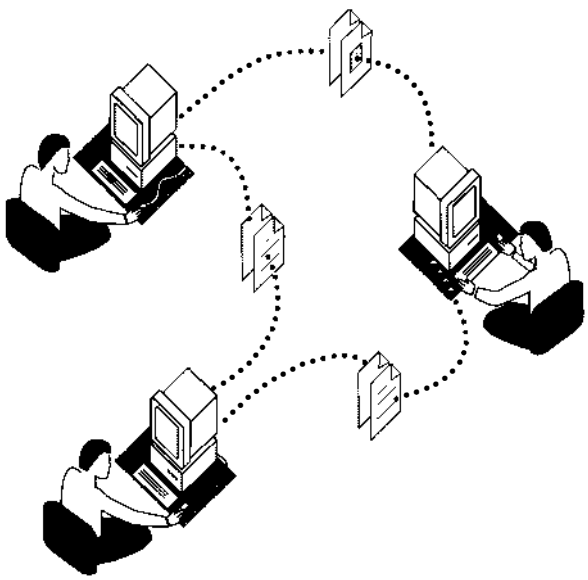


Figure 6-1 Sharing files on the network

To share files on your computer with other people over the network, you set aside an area of your computer, such as a folder or an entire hard disk, as a public space.

What you should know before you begin

You can share information on your computer in ways that are comfortable and convenient for you:

- You can share as much—or as little—as you like. In fact, you don't have to share anything at all. However, you can share only those files that you have UNIX permission to share, and you can further limit access to files with file-sharing permissions.
- You can share folders or disks with everyone on the network, or you can share items with only one person or a **group** of people.
- You can set up your A/UX system so that you can access the entire contents of your disks from another computer on the network.
- If you share an A/UX system with other people and you use the `root` system folder, the file-sharing permissions set by the last user can govern who you can share your files with. So if you share a system with other people, it's a good idea to create a personal System Folder. With a personal System Folder, you can create the file-sharing permissions you want, and you can store permissions and users and groups so that only you can determine who can access your files through file sharing.

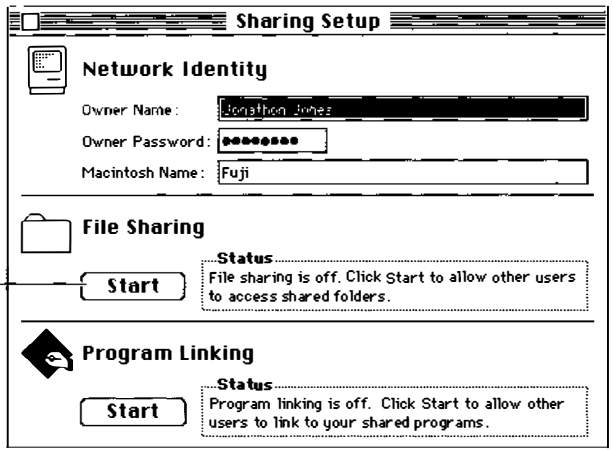
Starting file sharing

Before you can share folders or disks using A/UX, you must be logged in and the file-sharing capability must be turned on. Once on, it stays on until you turn it off. This means that even when you log out of A/UX, the next time you log in, file sharing will still be active. Before you begin to use the network, you must give your Macintosh a name that will appear in the Chooser of other computers on the network. You need to enter this information only once.

1 **Log in to A/UX.**

2 **Choose Control Panels from the Apple menu and double-click the Sharing Setup control panel icon.**

The Sharing Setup control panel opens.



3 **Type a name in the Owner Name text box and press TAB.**

Type the name of the primary user of this computer—very likely, this is you.

4 **Type your password in the Owner Password text box and press TAB.**

After you type your password and press TAB, the password is replaced by bullets (•) to help keep your password private.

Your password can be up to eight characters long. When you enter your password to connect to your computer, you will have to match the use of uppercase and lowercase letters that you enter here.

5 **Type a name for your Macintosh in the Macintosh Name text box.**

If a name already appears in the Macintosh Name text box, you can use that name or type in a new one. When people on your network want to access your computer, they will select this name in their Chooser.

6 **Click the Start button in the File Sharing section in the middle of the window.**

After you click the Start button, its label changes to "Cancel" and the status line describes what is happening while the file-sharing capability is starting up. It may take a few seconds to complete the startup. When the button is labeled "Stop" and the status line says "File sharing is on," your computer is ready to share files.

7 **Close the Sharing Setup control panel and the Control Panels folder.**

Sharing a folder or disk

You can share the contents of any disk—including hard disks and CD-ROM discs—connected to your computer. You cannot share anything on floppy disks, however. You can share up to ten folders or disks at one time. If you try to share more than the allowed number of folders or disks, you'll see an error message. (This number does not include the folders that are within the folders or disks that you share.) To share a folder or disk, do the following:

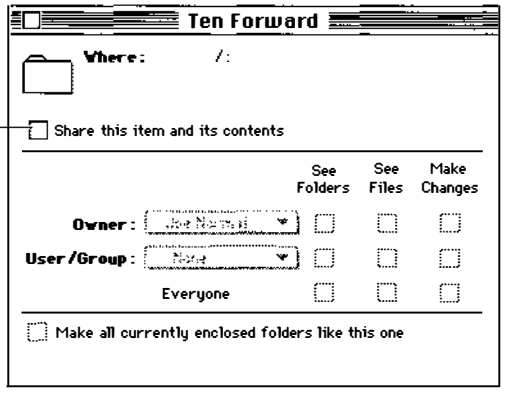
1 **Make sure file sharing is turned on.**

For directions, see the previous section, "Starting File Sharing."

2 **Select a folder, hard disk, or CD-ROM disc to share.**

3 **Choose Sharing from the File menu.**

A window appears.



Click here to share the folder.

4 **Click the box labeled "Share this item and its contents."**

This checkbox is used to share the folder or disk. When it is checked, and after you close the window, anyone on the network can access the folder or disk and its contents over the network.

To find out more about sharing a folder or a disk, see "Sharing a Folder or Disk With Specific People" later in this chapter.

5 **Close the window. To save your changes, click Save in the dialog box that appears.**

If you shared a folder and you are using the By Icon view or the By Small Icon view, the folder icon appears with network cables to indicate that it is shared.

If you are using other views, the shared folder icons look the same as other folder icons.

6 **Drag any files and folders that you want to make available to the network into your new shared folder or disk.**

The files and folders that you place in the shared folder or disk will be available to the user who connects to your computer. When a user does connect, a shared icon will appear on the user's desktop that can be opened just like the folders on a local disk.

When people on your network are connected to the shared folder, the icon appears with faces in the center to indicate that it is in use. To learn more about accessing a shared folder, see "About Accessing Information on Other Computers" later in this chapter.

△ **Important** Before you put application programs into shared folders, remember that copyright laws apply to sharing application programs over a network as well as to copying application programs from a floppy disk. Check with any applicable licensing agreements and follow their specifications. △

Naming specific people to share folders and disks with

If you've shared a folder as described in the previous section, you shared it with everyone on your network. You can also share folders only with specific people whom you've named on your computer. This section describes how to name those people.

Naming a user

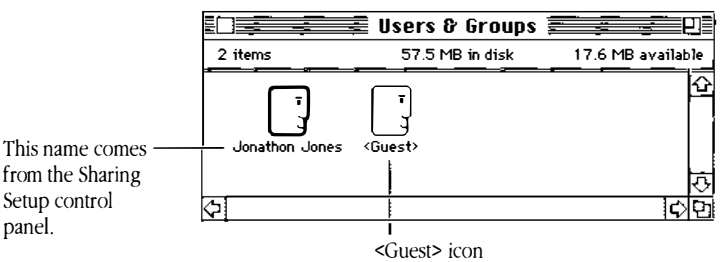
Once you name a user, that user is considered a **registered user**. You can name as many as 100 users and groups of users combined, but for optimal performance, it is recommended that you name no more than 50. If you try to name more than 100, an error message appears.

△ **Important** Macintosh file-sharing Users & Groups have no relationship to UNIX users and groups. For the sake of simplicity and consistency, it's a good idea to give users the same name for file sharing as they have in UNIX. So if a user's login name in UNIX is `maxwell`, name that user "Maxwell" when you name users to share files. △

To name a user and assign a password, follow these steps:

1 Choose Control Panels from the Apple menu and open the Users & Groups control panel.

The Users & Groups control panel is used to name the specific people you'll allow to access your computer for both file sharing and program linking. When you open the control panel for the first time, it displays two icons. The first icon has a bold outline and represents you, the user who is logged in to A/UX using the Finder.



The second icon in the window is the **<Guest>** icon. You use this icon to let anyone on the network access your computer or to prevent them from doing so. Guest access is automatically allowed when file sharing is first started.

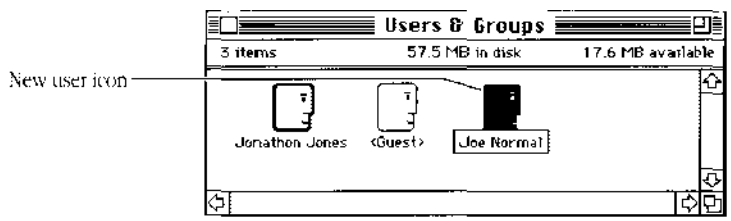
2 Choose New User from the File menu.

The New User command in the File menu is present only when the Users & Groups control panel is open and is the active window. After you choose the command, an icon labeled "New User" is created.

◆ **Note** You can also use COMMAND-N to create a new user. ◆

3 Replace the name "New User" with the name of a user for your computer.

Type a name for the user. Later, make sure that you notify the user of the exact name you type here since he or she must type the exact name to be able to use the folders and disks you share. (Since this is a function of the Macintosh OS, which, unlike UNIX, is not **case sensitive**, uppercase and lowercase letters are interchangeable.)

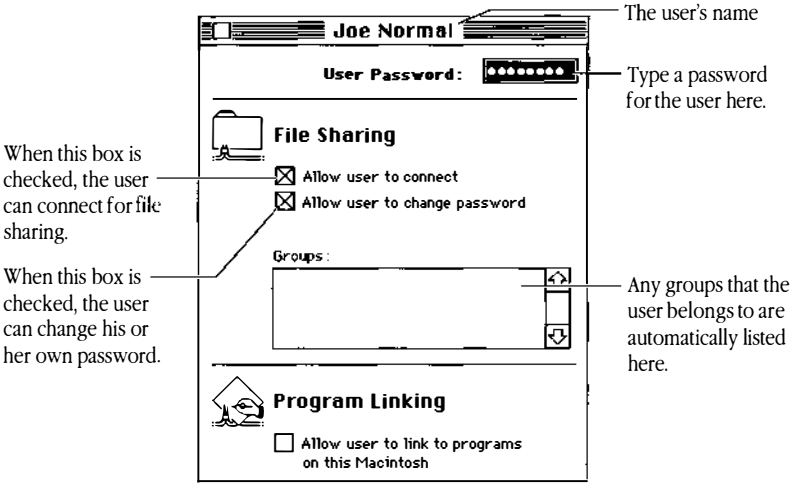


The user is now named on your computer allowing that person to share your folders or disks.

4 To give the user a password or view other user preferences, open the user icon.

If you don't want to assign a password, you have finished naming the user on your computer. To name more users, repeat Steps 2 and 3 of this section.

To give the user a password, continue with this step. When you open the user icon, the following window appears:



5 **To assign a password, type a password in the User Password text box.**

When you finish typing the password and press TAB, each character of the password is replaced by a bullet to help keep that password private.

The password can contain as many as eight characters. The user must type the password exactly as you type it here, including uppercase and lowercase letters. The password is case sensitive.

6 **Check the remaining checkboxes as desired.**

When the checkbox labeled "Allow user to connect" is checked, the user can connect to your computer from his or her computer. When the checkbox labeled "Allow user to change password" is checked, the user can change his or her password from his or her own computer.

The checkbox labeled "Allow user to link to programs on this Macintosh" is described in Chapter 6 of your *Macintosh Networking Reference.*

7 **Close the window. To save your changes, click Save in the dialog box that appears.**

The user information is saved and takes effect immediately. To name more users, repeat Steps 2 through 7 of this section.

8 **When you're finished naming users, close the Users & Groups control panel.**

To share folders or disks with the users you have now named, go on to the section titled "Sharing a Folder or Disk With Specific People." To name a group of users, continue with the next section, "Naming a Group of Users."

To learn more about sharing files with specific people, see "Recommendations for File Sharing" in Appendix A of your *Macintosh Networking Reference.*

Naming a group of users

If your office or work environment consists of groups of people, such as project teams, who work together and need to share similar information, you can simplify your work by naming a group of users on your computer and later sharing a folder with that group.

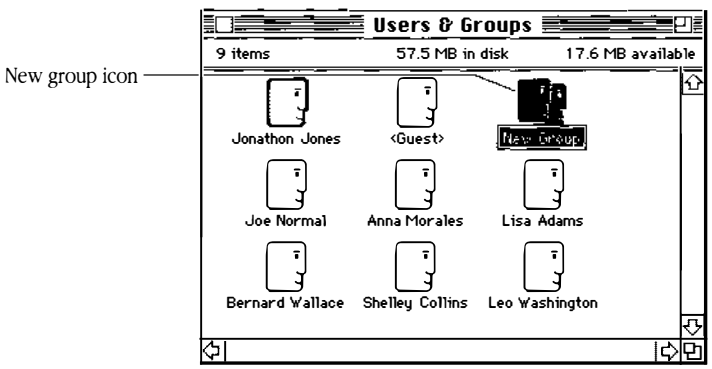
Once you name a group, it is considered a **registered group.**

To name a group of users, follow these steps:

1 **Choose Control Panels from the Apple menu and open the Users & Groups control panel.**

2 **Choose New Group from the File menu.**

The New Group command is present only when the Users & Groups control panel is active. After you choose the command, a new icon labeled "New Group" appears.



3 **Replace the name "New Group" with a name for the group.**

4 **Drag the user icon of each person that you want to be a member of the group to the new group icon.**

Alternatively, you can drag the group icon to the user icon for the same result.

It's not necessary to include your owner icon in groups because you always have access to the entire contents of your computer, unless you specify otherwise. Also, you don't need to include the <Guest> icon in any group.

◆ **Note** To select more than one user, hold down the SHIFT key while clicking the names of all the users you want to add. ◆

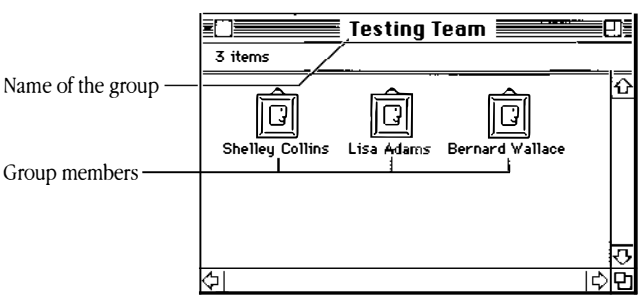
Reviewing group membership

To review what users belong to a group or to review what groups a user belongs to, do the following:

■ **To see the names of all of the members of the group, open the group icon.**

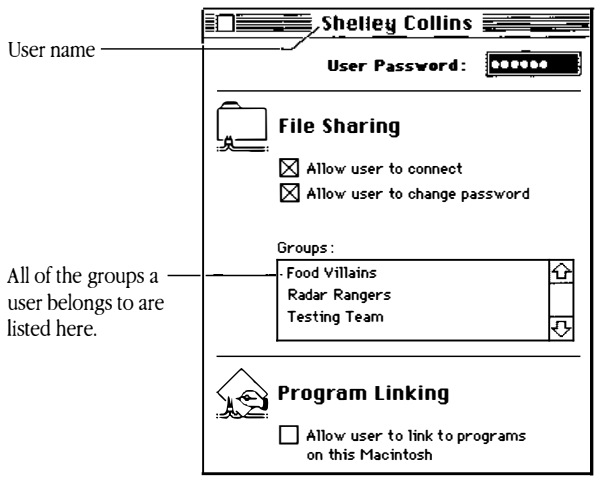
A window opens and shows a **member** icon for every user that you added to the group. If you want to see or change information in a user icon, you can open either the user icon itself or its member icon in any of the groups in which it is a member.

If you use the By Icon or By Small Icon view, the contents of a registered group will look like the following:



- **To see the names of all of the groups that a user belongs to, open the member icon or user icon.**

  You can double-click either the member icon that appears in the group window or the original user icon that appears in the Users & Groups control panel. In both cases, a window opens and displays a Groups list that names the groups to which the user belongs.



To share folders or disks with the groups you have now named, go on to the next section titled "Sharing a Folder or Disk with Specific People."

Sharing a folder or disk with specific people

If you named the users and groups with whom you want to share information as described in the previous section, you can now share folders and disks with those people. Follow these steps to do so:

1 **Make sure file sharing is turned on.**

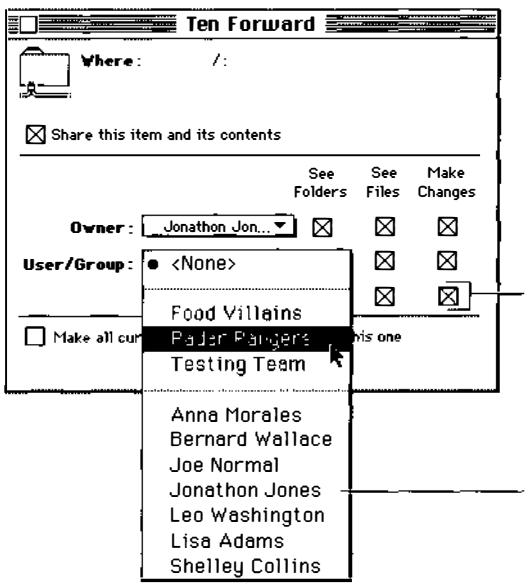
2 **Select a folder, hard disk, or CD-ROM disc to share.**

3 **Choose Sharing from the File menu.**

A window appears.

4 **Check the box labeled "Share this item and its contents," if it is not already checked.**

5 **Select a user or a group from the User/Group pop-up menu.**



The **User/Group** pop-up menu contains the names of all of the users and groups that you registered. You can choose only one of the items in the list. If you make a mistake, simply change the selection. The name you choose is displayed when you finish making a selection.

6 **To share this folder or disk with only the user or group that you've chosen, click the bottom row of three checkboxes to remove the X from each box.**

This action removes privileges to anyone else on your network aside from you and the user or group you chose in Step 5. For more information about privileges and strategies for using them, see Chapter 5 in *A/UX Essentials*.

7 **Close the window. To save your changes, click Save in the dialog box that appears.**

The folder or disk can now be accessed by the people you chose in the window.

Setting up your computer for remote accessing

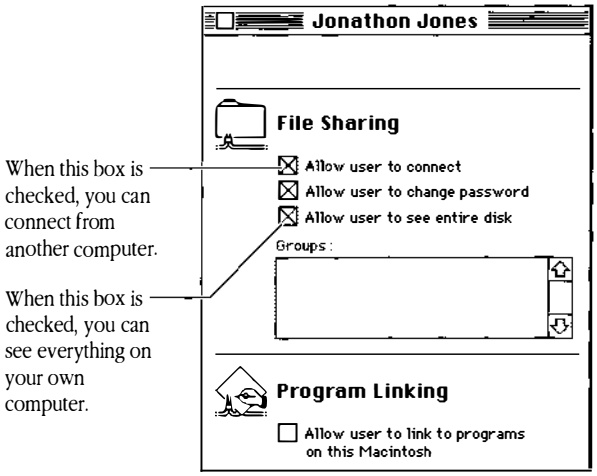
To access your Macintosh computer from another Macintosh computer on the network, prepare it in the following way:

- Log in to your computer.
- Make sure that file sharing is turned on.
- Then do the following:

1 **Choose Control Panels from the Apple menu and open the Users & Groups control panel.**

2 **Open the Owner icon—the one with the bold outline.**

The name of the Owner icon comes from the name entered in the Sharing Setup control panel. The Owner icon opens.



3 **Check the boxes labeled "Allow user to connect" and "Allow user to see entire disk," if they are not already checked.**

When these boxes are checked, you'll be able to see and use everything on your hard disks or CD-ROM discs from another computer on the network. You can access everything on your disks, regardless of which folders are shared or what privileges have been assigned to them.

△ **Important** The owner password that is entered in the Sharing Setup control panel is what helps to ensure that only you—the owner of this computer—can gain entry to the entire contents of the disks on your computer. When you connect to your computer from another computer, you use that password to identify yourself. Be sure that you keep your password private to prevent anyone else from gaining such access to your computer. △

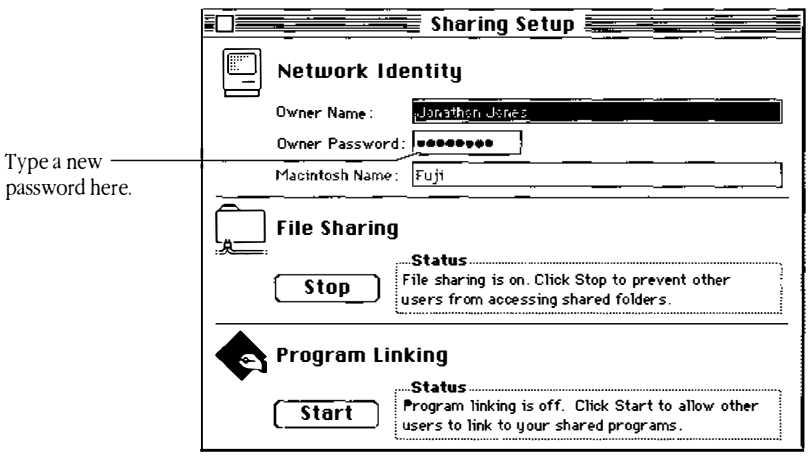
4 **Close the window and the Users & Groups control panel.**

Changing your password

You use a password to be able to connect to your own computer from another computer on the network. This password is your file-sharing password only, not your UNIX login account password. To change your password, follow these steps:

1 **Choose Control Panels from the Apple menu and open the Sharing Setup control panel.**

You see the Sharing Setup window.



Type a new password here.

2 **Type a new password in the Owner Password text box.**

The password is case sensitive; you must type uppercase and lowercase exactly as you enter them.

3 **Press TAB.**

Each character of your password is replaced by a bullet to help keep it private.

4 **Close the Sharing Setup control panel.**

Your new password is now in effect.

Giving folder ownership to someone else

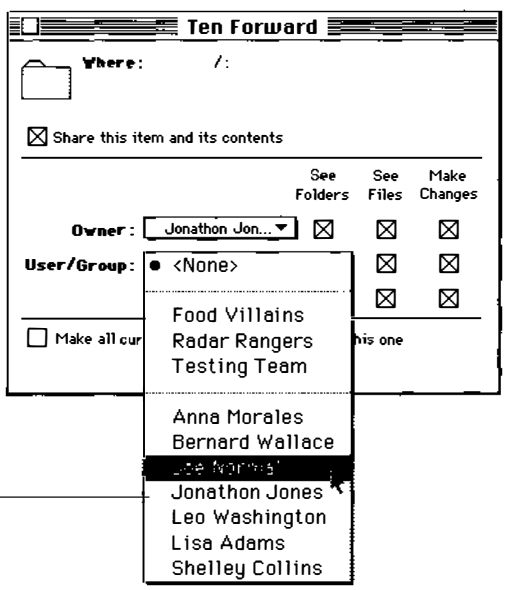
You can give away ownership of a shared folder or disk on your computer to another registered user or registered group, or even allow general ownership by anyone on the network. Folder ownership gives a person the ability to change privileges to a shared folder, so if you name another person as the owner of a shared folder on your computer, you share this ability with that person. (As the computer owner, you can always open and use everything on your own computer and you can change ownership of the folder again at any time.) If you give away ownership of a folder or disk, make sure that you move any important folders or documents out of it first so that you can maintain complete control of the privacy of those documents.

1 **Select a shared folder or disk.**

2 **Choose Sharing from the File menu.**

3 **Select the name of a registered user, group, or <Any User> in the Owner pop-up menu.**

If you select <Any User>, then anyone on the network can change the access privileges of the folder or disk or rename the owner.



Choose a name from this pop-up menu to change ownership of this folder.

4 **Close the window.**

A dialog box appears, asking you to confirm that you want to change ownership.

5 **Click OK.**

Changing a user's password

To reduce the chances of a person on the network connecting to your computer with someone else's name and password, you can change the passwords of the users you've named on your computer. Follow these steps to do so:

1 **Choose Control Panels from the Apple menu and open the Users & Groups control panel.**

2 **Open the user icon of the user whose password you want to change.**

A window opens with the password—hidden by bullets—already selected.

3 **Type a new password to replace the bulleted password.**

The password can be up to eight characters in length. The user must type the password exactly as you type it here, including uppercase and lowercase letters.

4 **Close the window. To save your changes, click Save in the dialog box that appears.**

The new password is saved and takes effect immediately. Remember to tell the user the new password.

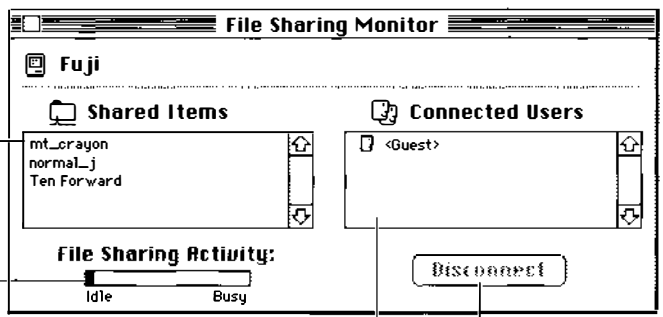
For more information about password protection, see Appendix A, "Privacy and Data Protection," in your *Macintosh Networking Reference*.

Monitoring file-sharing activity

When file sharing is on, you can review who is connected to your computer and what folders or disks you've shared.

- **Choose Control Panels from the Apple menu and open the File Sharing Monitor control panel.**

  The File Sharing Monitor opens.



All of the folders and disks you are currently sharing are listed here.

Level of file-sharing activity on your computer.

Any people currently connected to your computer for file sharing are listed here.

To disconnect a user, first select his or her name; then click here.

A scrolling list of your shared folders and disks appears on the left side of the window. Though only the folders and disks that you shared using the Sharing command are displayed in this list, all of the files and folders contained within them are accessible to network users, unless you have modified their access privileges. Use the scroll bar if necessary to see more of the list.

A scrolling list of all the network users currently connected to your computer appears on the right side of the window.

Disconnecting a user

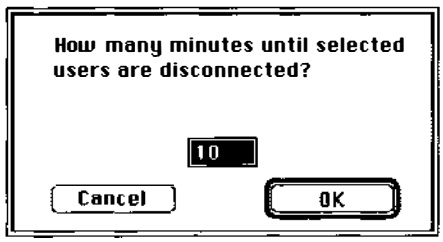
You can disconnect users who are currently connected to your computer.

1 **Select the user or users that you want to disconnect from the list of connected users in the File Sharing Monitor.**

To select more than one user, hold down the SHIFT key while clicking the names of all the users you want to disconnect.

2 **Click Disconnect.**

The Disconnect dialog box appears.



3 **Type the number of minutes that you want to elapse before disconnection occurs.**

To disconnect a user immediately, type the number 0. Otherwise, type in the amount of time you wish to elapse before the user is disconnected. It's a good idea to give people enough time to save any changes they have made to the files. A minute or two is usually enough. The next time you disconnect a user, the number of minutes you set this time will appear as the default.

4 **Click OK.**

A message appears telling you that users will be disconnected in the number of minutes you specified. This message stays on the screen for the number of minutes you specified.

The users will be disconnected in the amount of time you requested.

5 **Close the File Sharing Monitor.**

Reducing access to your shared files

This section describes the ways you can reduce access to your shared files by defining what information can be accessed by which people.

Making a shared folder or disk private again

You can make a shared folder or disk private again after you have shared it. When you do, any files or folders it contains will become private again as well.

1 **Select a shared folder or disk.**

2 **Choose Sharing from the File menu.**

The Sharing window appears.

3 **Click the box labeled "Share this item and its contents" to remove the X.**

4 **Close the window. To save your changes, click Save in the dialog box that appears.**

Now no one can access this folder from the network, except yourself.

Denying access to a specific user

To prevent a registered user from being able to access the shared folders and disks on your computer, follow the steps given here.

If you simply want to disconnect a user who is connected to your computer, see "Disconnecting a User" earlier in this chapter.

1 **Choose Control Panels from the Apple menu and open the Users & Groups control panel.**

2 **Open the user icon representing the person you want to prevent from accessing your computer.**

3 **Click the box labeled "Allow user to connect" to remove the X.**

4 **Close the user window. To save your changes, click Save in the dialog box that appears.**

The user can no longer access your computer from the network as a registered user. If the user is currently connected, he or she will be disconnected immediately.

Denying access to guest users

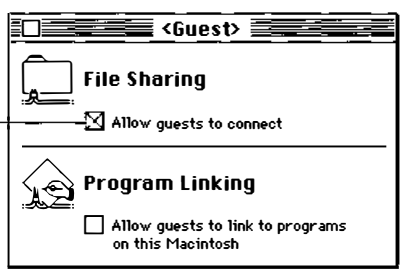
Your computer is set up to allow guest access when you first start using it. However, you can deny or reinstate this capability at any time. If you deny guest access, you can only share the information in your shared folders and disks with registered users.

1 **Choose Control Panels from the Apple menu and open the Users & Groups control panel.**

The Users & Groups control panel displays icons for all the users and groups that you have registered, an icon for you, and a <Guest> icon.

2 **Double-click the <Guest> icon.**

The <Guest> icon opens.



When this box is checked, guests can connect for file sharing.

3 **Click the box labeled "Allow guests to connect" to remove the X.**

4 **Close the window. To save your changes, click Save in the dialog box that appears.**

Now guest users cannot access any shared folders or disks.

Denying access to yourself

To prevent yourself from being able to access anything on your own computer from another computer, follow these steps:

1. **Choose Control Panels from the Apple menu and open the Users & Groups control panel.**
2. **Open the owner icon—the one with the bold outline.**
3. **Click the box labeled "Allow user to connect" to remove the X.**
4. **Close the window. To save your changes, click Save in the dialog box that appears.**

Removing a user's membership in a group

To revoke a user's membership in a group, follow these steps:

1. **In the Users & Groups control panel, open the group icon that the user belongs to.**
2. **Drag the member icon for that user to the Trash.**
3. **Select Empty Trash from the Special menu.**

The user is no longer a member of that group.

Removing a user or group altogether

You can remove a registered user or group at any time. When you remove a user, all of the memberships to groups for that person are removed automatically.

1. **In the Users & Groups control panel, drag the user or group icon to the Trash.**
2. **Select Empty Trash from the Special menu.**

You cannot throw away the <Guest> icon or the owner icon. To prevent guest access, see the section "Denying Access to Guest Users," earlier in this chapter.

Turning off file sharing

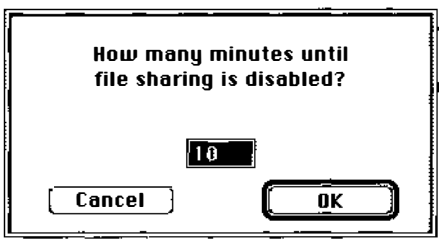
You can turn off file sharing at any time. Once you do so, network users cannot access any folders or disks on your computer, and you cannot access your computer from another computer on the network.

If you subsequently turn file sharing back on, all folders or disks that you shared before you turned off file sharing will again be available to people on the network.

1 **Choose Control Panels from the Apple menu and open the Sharing Setup control panel.**

2 **Click the Stop button in the File Sharing section of the window.**

After you click the button, its label changes to "Start." If users are currently connected to your computer, a dialog box appears in which you can specify a delay before file sharing is turned off. This gives current users some time to save their work and to disconnect from your computer themselves.



3 **Type the number of minutes you want to elapse before file sharing is turned off.**

It's a good idea to give people a few minutes to save their files before you turn off file sharing.

4 **Click OK.**

After the designated period of time, the file-sharing capability is turned off. If network users are currently accessing your computer and you specified a delay before file sharing is turned off, they will receive warning messages so they can save their documents. If someone is using files on your computer when file sharing is turned off, any changes he or she has made and has not saved will be lost.

◆ **Note** If you log out while files are being shared, the system will respond as it does if you turn off file sharing. ◆

About accessing information on other computers

When you connect computers on a network, you can take advantage of the extra storage space that multiple computers provide. For example, you're no longer confined to the amount of space on your own hard disk, but you can store and retrieve information on **shared disks**—-hard disks or CD-ROM discs that contain files and folders that can be retrieved over the network.

Your Macintosh can access and store information on shared disks that are attached to two types of computers:

- AppleShare and AppleShare-compatible **file servers**
- other Macintosh computers running system software version 7.0 or later that have shared disks

The steps you take to connect to a shared disk—whether it's attached to a file server or another Macintosh computer—are exactly the same. However, when you use shared disks, be aware that file servers are usually turned on and available at all times, whereas personal computers may be turned off at the end of the day or at other times, as determined by their owners.

What to do before you begin

Before you attempt to retrieve or store information on the network, do the following tasks:

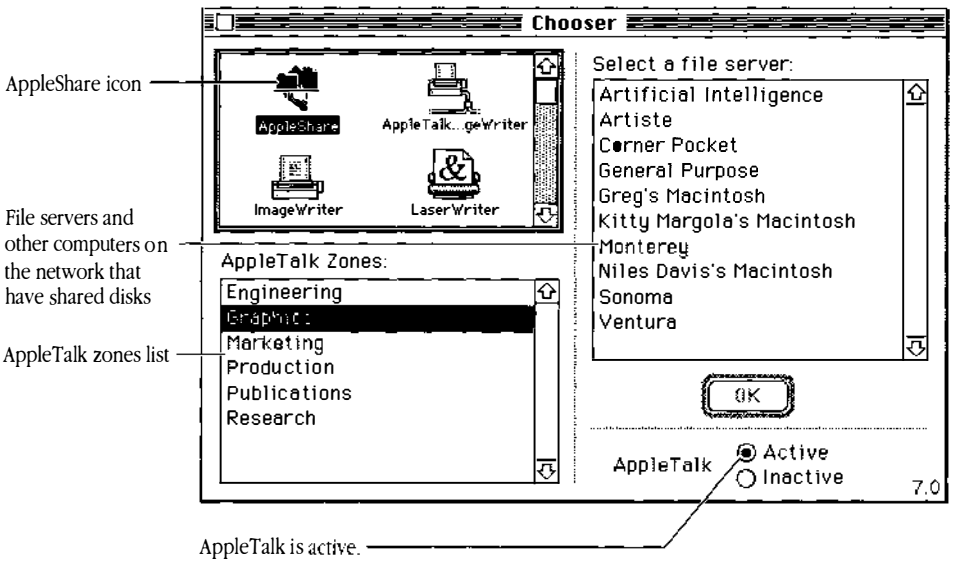
- Obtain the name of the file server or the name of the computer that has the shared disk you want to access.
  Ask your network administrator for the name of the file server, or the computer owner for the name of the computer.
- If your network has **zones**, find out the name of the zone that file server or computer is in.
- Find out if you are registered on that file server or computer or if it allows guest access.
  The network administrator or computer owner can name specific users on the file server or computer, thereby enabling users like yourself to access it. Alternatively, administrators and computer owners can allow **guest access**, which enables any network user to access the computer. You can access files from another computer on the network only if you are a registered user or if guest access is allowed.
- If you are a registered user, find out your user name and password to that server or computer.
  If you are registered, you will need to type your name and password to verify that you are entitled to have access. Find out exactly how both are spelled and make sure you use uppercase and lowercase letters exactly because your password is case sensitive.

Connecting to a shared disk

Follow these steps to connect to a shared disk:

1 **Select Chooser from the Apple menu.**

The Chooser opens.



2 **Make sure that AppleTalk is active.**

The AppleTalk Active button must be selected in order to connect to a shared disk.

3 **Click the AppleShare icon in the upper-left section of the Chooser window.**

4 **If your network has zones, click the zone in which the computer with the shared disk you want resides.**

If you do not see a box in your Chooser labeled AppleTalk Zones, skip this step.

You may have to scroll to locate the zone you want. When you select a zone, the available computers in that zone are listed in the upper-right portion of the window.

◆ **Note** A quick way to find a zone name in the AppleTalk Zones list is to make that section of the Chooser active by clicking inside it or by pressing TAB until that section is active. Then, type the first letter or the first few letters of the name of the zone you want to select. The list will automatically scroll to and select the zone name starting with those letters. This shortcut can be used in all three panels in the Chooser. ◆

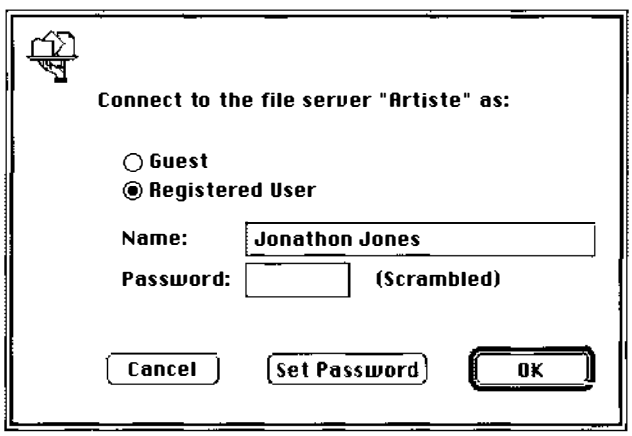
Zones are set up by a network administrator and are used to distribute the devices on the network among groups for easier access. For example, although your entire network may contain a large number of file servers or Macintosh computers, you can quickly find one by knowing what zone it's in.

5 **Click the name of the computer with the shared disk that you want to use.**

You may have to scroll to find the one you want. If the name you want isn't listed, make sure that you have selected the correct zone, or ask your network administrator or the Macintosh owner if the computer is on and you have the correct name.

6 **Click OK.**

A dialog box appears for you to identify yourself as a registered user or guest.



7 **Click Guest or Registered User.**

If you are a guest or if you are not certain that you are registered, click Guest and skip to Step 10. If you are a registered user, click Registered User and continue to Step 8.

If the Guest option is dimmed, this indicates that guests are not permitted access to this computer.

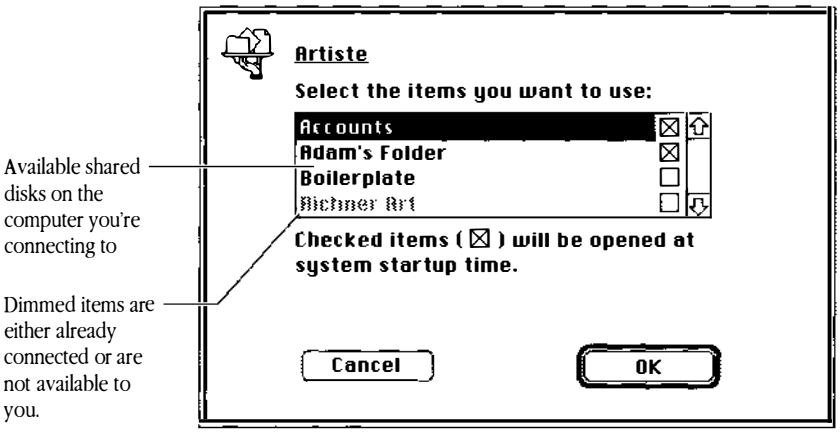
8 **If you are a registered user, make sure that the displayed name is the exact name that is registered. If it is not, correct it.**

Type your registered user name with the exact spelling as it is registered. Uppercase and lowercase letters do not have to match. If a name is already entered, it is taken from the Owner Name in the Sharing Setup control panel on your computer. If that is not your registered name then type your registered name now. (Or, if you use this registered name frequently, you may want to change your owner name. See the section, "Naming Your Macintosh and Its Owner" in Chapter 2 of *Macintosh Networking Reference*.)

◆ **Note** For your information, the type of security measure that is being used to connect your computer to the shared disks is displayed next to your password. The possible methods are *Clear Text, Scrambled,* and *Two-Way-Scrambled.* The method used is determined for you. ◆

9 **Type your password and click OK.**

Type your password exactly as it is registered, including uppercase and lowercase letters. After you click OK, a dialog box appears in which you select the shared disks that you want to use.



10 Select the name of the shared disks you want to use.

You may have to scroll to find the name you want. You can also type the first few letters of the name of a shared disk to find it. To select more than one, hold down the SHIFT key or the COMMAND key and click the name of each shared disk that you want.

A dimmed name indicates either that you are already connected to that shared disk, or that you do not have access privileges to it.

11 Click OK.

If the OK button is dimmed, you have not selected a shared disk. Go back and select one.

When you click OK, the dialog box closes and a shared disk icon appears on your desktop in the area where hard disk and floppy disk icons appear.

12 Close the Chooser.

Connecting automatically

There are two methods by which you can connect to another computer automatically. Both methods are especially convenient if you use certain shared disks frequently. These are described below.

Using aliases to connect quickly

You can connect to a shared disk almost instantly by using aliases. Follow these steps to do so:

1 **Connect to a shared disk.**

Follow the steps in the previous section to connect to a shared disk.

2 **Select the shared disk that you'll want to connect to later.**

3 **Choose Make Alias from the File menu.**

An alias icon appears that has the name of the shared disk with the word *alias* appended to it.

4 **If desired, rename the alias.**

5 **Drag the new alias to a convenient place on your desktop or disk.**

6 **When you're finished using the shared disk, disconnect from it by dragging its icon to the Trash.**

7 **The next time you want to use the shared disk, double-click the alias you created in Step 3.**

8 **If you originally connected to the shared disk as a registered user, a password dialog box appears. Type your password.**

If you originally connected as a guest, the dialog box will not appear, so skip this step.

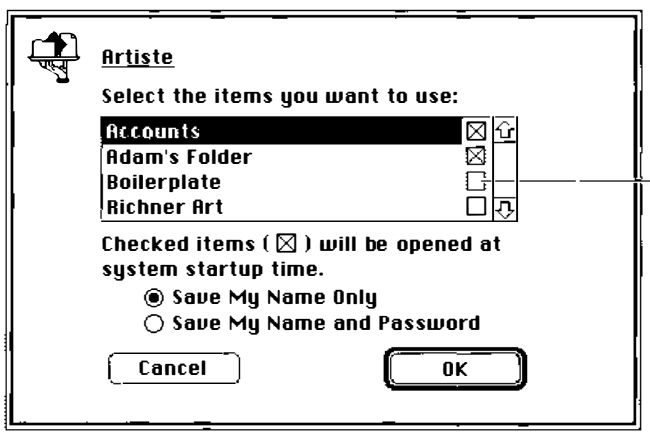
The shared-disk icon appears on your screen.

For more information, see the user's guide that came with your Macintosh computer.

Connecting automatically when you start up

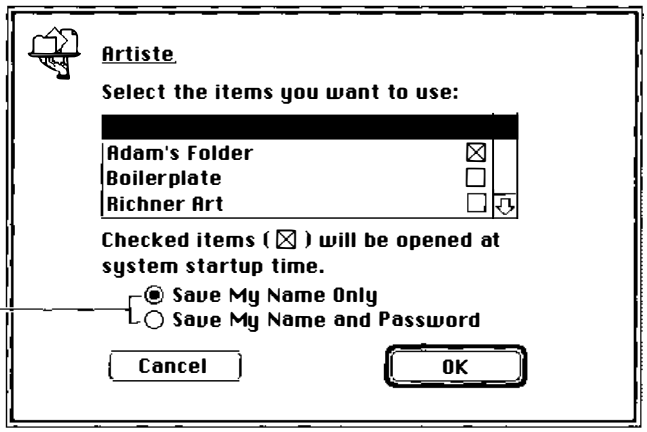
You can set up your computer to connect automatically to shared disks whenever you log in. Follow these steps to do so:

1 **Connect to a computer as described in Steps 1 through 9 of "Connecting to a Shared Disk."**



2 **Click the checkbox next to the name of a shared disk that you want to connect to automatically when you log in.**

If you are a registered user, two buttons appear when a checkbox of a shared disk is checked.



Click one of these buttons to indicate whether you'll have to enter your password at startup.

3 **If you're connecting as a registered user, click one of the buttons to indicate whether you want to save your name and password or just your name.**

If you are connecting as a guest, skip this step.

If you select Save My Name Only, when you turn on your computer, you'll have to enter your password in order to connect to and access the shared disk. This option is recommended to prevent unauthorized people from accessing the shared disk from your computer. If you select Save My Name and Password, your Macintosh will automatically connect to the server without any input from you. This is a less secure method than the Save My Name Only option because anyone walking by your computer could connect to the shared disk without even knowing your registered name or password.

4 **Click OK.**

5 **Close the Chooser.**

Using and storing files on a shared disk

Once you've connected to a shared disk, you can use the files stored on it in a similar manner to how you use stored files on your own computer. You can create new documents on a shared disk but in some cases your ability to do so may be limited. This is because the owner of a shared disk can control access to it and can prevent anyone else from changing its contents in any way.

You can also use application programs stored on the shared disk. Be aware, however, that programs generally run faster when they are stored on your own computer.

Whenever you send or receive information across the network using AppleShare, the double-arrow icon flashes in the upper-left corner of the menu bar:

Disconnecting from a shared disk

When you're finished using a shared disk, you can disconnect from it in three different ways. First close any open files or programs and then use any one of the following methods:

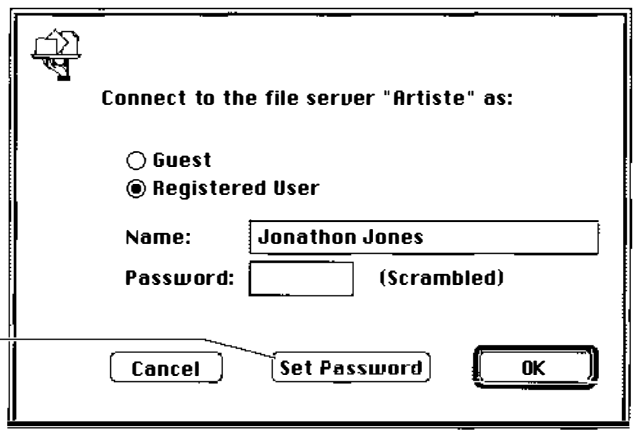
- Select the icon of the shared disk you want to disconnect from and choose Put Away from the File menu.
- Drag the icon to the Trash.
- If you are done working for the day, choose Logout from the Special menu as usual. You will be disconnected automatically when you log out.

Changing your password to access a shared disk

If you are a registered user, you may be able to change the password you use to access a shared disk. The file server administrator may have restricted this capability. In that case, a warning box will inform you that you cannot change your password to that computer.

1 **Follow steps 1 through 6 of "Connecting to a Shared Disk."**

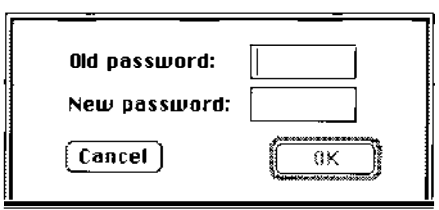
The following dialog box is displayed and the Registered User button is already selected.



Click here to change your password.

2 **Click the Set Password button.**

A password dialog box appears.



3 **Type your current password.**

Bullets appear for every letter you type to protect your privacy.

4 **Press TAB and type a new password.**

For best security, type a password that will not be easily guessed by someone else.

5 **Click OK.**

A small dialog box informs you that you'll need to retype your new password to verify its accuracy. Click OK.

6 **Retype your password.**

7 **Click OK.**

Your new password is now in effect.

Affecting access to folders

When you create a folder on another computer, you become its owner. Folders that you own appear with a tabbed folder icon.

Ownership allows you to set privileges to that folder in order to determine who can access it and in what way. See *A/UX Essentials* for information about setting privileges.

Giving folder ownership to someone else

Another way to affect access to a folder is to give its ownership to someone else. However, once you do so, the new owner can decide to restrict your access to that folder and the folders within it at any time. Make sure that you have moved anything you may need to a folder that you own before you give ownership to someone else.

1 **After connecting to a shared disk, select a folder on it that you own.**

2 **Choose Sharing from the File menu.**

3 **Type the name of a registered user or registered group in the Owner text box.**

Type the name exactly as your network administrator or the Macintosh owner registered it.

4 **Close the dialog box. Then click OK to save the changes to the access privileges.**

A dialog box asks you to confirm that you want to change the owner of this disk. Click OK. The user or group that you named assumes ownership immediately.

Accessing your own computer remotely

You can connect to your own Macintosh running System 7 from another computer on the network and access everything on all of the hard disks or CD-ROM discs attached to your computer. As the owner of your computer, only you have this privilege. This can be useful, for example, if you've forgotten an important file or if you need to refer to some information that you have on your computer, but you're no longer in your own office.

1 **To be able to access everything on your computer, set up your computer as described in the section "Setting Up Your Computer for Remote Accessing" earlier in this chapter.**

2 **Connect to your computer as a registered user in the normal manner. Enter the owner name and password as they are named in the Sharing Setup control panel on your computer.**

3 **Continue the connection process as you normally would.**

7 Sharing Files With NFS

This chapter discusses Network File System (NFS) file sharing—how and when to use it and how to make your files available to other users. NFS is networking software that works with a variety of operating systems and allows you to share files with other computers that are running NFS. For information on how to manage and administer NFS, see *A/UX Network System Administration.* For information about sharing files with other Macintosh users not running A/UX, see Chapter 6 of this manual, "Sharing Files Over the AppleTalk Network."

This chapter covers the following topics:

- about NFS file sharing
- when to use NFS
- what you need to run NFS
- preparing your system to run NFS
- making your files available (being a server)
- using files on other systems (being a client)

You can read this chapter in its entirety or go directly to those sections that are of use to you. Before using the information in this chapter, you must be familiar with file permissions (see *A/UX Essentials*).

About NFS file sharing

When you set up your system to run NFS, you make your system capable of sharing files with other systems on the network. Once you have access to files with NFS, it is as if those files actually reside on your system; you can work with them just as if they were physically stored on the computer at your desk. The process of making files available to others on the network is called **exporting.** When you make files available by exporting them, you become an **NFS server.** When you want to use files that have been exported, you **mount** a file system, which means that those files are as available to you as any files on your local system. When you mount a file system, you are being an **NFS client.** Often a system is both a server and a client at the same time. In effect, NFS can greatly increase the amount of disk storage space on your local system by giving you access to files on remote systems, rather than duplicating those files on the local system. As you can see in Figure 7-1, large databases, such as the manual (man) pages, can be stored on one hard disk, and once that hard disk is set up as a network server, all the users on the network can refer to them as easily as if they were stored on each user's system.

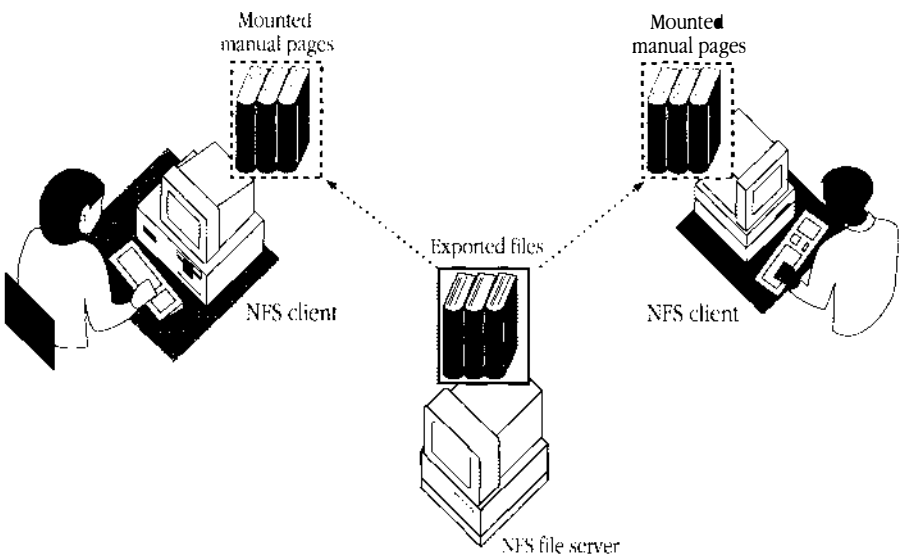


Figure 7-1 Remotely mounted manual pages

When to use NFS

If all the users with whom you'd like to share files are working on Macintosh computers, you can use Macintosh file sharing, described in Chapter 6. If you'd like to share files with users on UNIX or non-UNIX systems that support NFS, then the NFS procedures described in this chapter are appropriate. First, check the section "What You Need to Run NFS" to make sure that your system can run NFS. Next, if you are interested in setting up some or all of your files for use by others, go to the section "Making Your Files Available (Being a Server)" for information on how to set up your UNIX files for access by others. For the steps that enable you to connect with files that are stored on other systems, go to the section "Preparing Your System to Run NFS."

◆ **Note** Server and client systems can each affect the performance of the other in a variety of ways. For example, in certain configurations, the client system becomes unusable when the server system is down. Because of these issues, it's a good idea to check with your network administrator before setting up your system as an NFS server or client. ◆

What you need to run NFS

To make use of NFS file sharing, you must have these elements:

- Built-in Ethernet hardware or an Ethernet card.
- Your system must be connected to a network that is configured for TCP/IP networking services. These setup procedures are covered in Chapter 2, "Connecting to an Existing Network."
- The password for `root`.

Once you have all the necessary elements, go to the next section to configure your system for NFS.

Preparing your system to run NFS

When you configure your system to use NFS, you give it the ability to be a server, a client, or both.

Before you perform the steps to configure your system for NFS, you should log in to the Finder environment as `root` and select CommandShell from the Applications menu, or you can log in as `root` in the Console Emulator session.

To set up your system to use NFS, follow these steps:

1 **To make a new kernel that supports NFS, enter this command:**

```
newconfig nfs
```

Messages appear indicating that it may take several minutes to prepare to build a new A/UX kernel. Then this prompt appears:

```
Do you want this machine to be an NIS client?  (y or n)
[default: n]
```

2 **Enter** `n`

Then a message appears indicating that it may take several minutes to build a new A/UX kernel, followed by a message telling you to restart.

◆ **Note** It is assumed that you've already connected to an existing network by running `newconfig bnet` (as explained in Chapter 2). If you haven't, you will be prompted to enter more information here. See Chapter 2, the section called "Connecting to a TCP/IP Network." Collect the information listed in the section called "Before You Connect to a Network" so that you can respond to the prompts with the correct information. If you make a mistake entering this information, see Appendix A. ◆

3 **To restart, pull down the Special menu in the Finder and select Restart.**

When A/UX is restarted, the computer should be capable of using NFS, assuming that the computer is connected correctly to a TCP/IP network.

Making your files available (Being a server)

When the computer is running a kernel that supports NFS, you can make your files available to other systems on the network by exporting them. Giving access to the files on your system as a server is somewhat like handing out keys to locked file cabinets in your office: you control who gets access, and you also determine which files they can see. With NFS, you also determine what remote users can do with the files: read them, write (modify) them, or in the case of programs, run (execute) them. Follow these steps to export files:

1 **Using the text editor of your choice, edit the `/etc/exports` file.**

The `/etc/exports` file specifies the files and directories that you want to share and the names of hosts that are allowed to use the shared files. In NFS terminology, sharing files is accomplished by exporting them.

For example, this entry enables all users on the network to read the manual pages:

```
/usr/catman -ro
```

The `-ro` option (for read-only) is required because the file permissions on `/usr/catman` do not allow users to write to the files in that directory. If the file permissions on the file system, directory, or file you are exporting allow "others" write permission, you do not need to specify any option. If the file permissions allow writing for "others," and you want to prevent writing by NFS clients, you can use the `-ro` option. If you want to allow some NFS clients write permission and deny write permission to others, use the `-rw` option as shown here:

pathname `-rw=`*hostname*

The `-rw` option allows the host specified by *hostname* to read and write, but all other hosts are limited to read only.

2 **Check permissions on the files you plan to share.**

Make certain that "others" have permission to read the files or view the directory you want to share. When the client mounts the data, it must do so with permissions that match or are a subset of the permissions of the data on the NFS server.

3 **To make the files available, enter this command:**

```
exportfs -a
```

This is the same command that appears in `/etc/rc` and is executed each time A/UX is restarted.

4 **To check that the pathnames specified in `/etc/exports` can be mounted, enter this command:**

```
showmount -e
```

For example, if the entry in `/etc/exports` on *hostname* is

```
/usr/catman  -ro
```

the response would be

```
export list for hostname:
/usr/catman  (everyone)
```

For more information about being an NFS server, see Chapter 6, "Configuring and Managing the Network File System," in *A/UX Network System Administration.*

Using files on other systems (Being a client)

If you've run `newconfig` (as explained earlier), you have client capabilities. You can access files located on an NFS server when the following conditions are met:

- The NFS server has exported the file system, directory, or file.
- The NFS client has mounted the remote file system, directory, or file.
- The user has permission to perform read, write, or execute operations on the remote data.

To test that NFS is working properly, you can mount the online manual pages from the NFS server.

When you mount a remote file system, you need to specify a mount point. You can use an existing directory that is empty, or you can create a new directory by using the `mkdir` command. In either case, the permissions on the directory should be 755 to allow users permission to read and traverse the directory hierarchy.

Before you can mount the `/usr/catman` hierarchy from the NFS server, verify that the server has exported it.

1 **To verify that your system recognizes the files exported by the NFS server, enter this command:**

`showmount -e` *name-of-NFS-server*

For example, if `hostname1` is exporting

`/usr/catman`

as shown in the server example, the response would be

```
export list for hostname1:
/usr/catman (everyone)
```

Whenever you mount files from a server, you should mount them on an empty directory, so you need to know if the directory is empty.

2 **To check that the directory is empty, enter this command:**

```
ls /usr/catman
```

3 **If the directory isn't empty, back it up; then remove its subdirectories by entering this command:**

```
rm -r /usr/catman/*
```

4 **To check permissions on the client `/usr/catman` mount point, enter this command:**

```
ls -ld /usr/catman
```

You should see something like this:

```
drwxr-xr-x 1 bin    bin    512 Sep 13 05:06  /usr/catman
```

As with local mounts, the mount point directory must have access permissions that are at least as open as those of the remote pathname being mounted. In addition, users' UIDs (user IDs) and GIDs (group IDs) must agree across systems. If the permissions don't match the example, change them using these commands:

```
chmod 755 /usr/catman
chown bin /usr/catman
chgrp bin /usr/catman
```

5 **To temporarily mount `/usr/catman`, enter this command:**

`mount -o bg,soft,ro` *name-of-NFS-server*`:/usr/catman /usr/catman`

This command mounts the remote `/usr/catman` hierarchy with the `bg`, `soft`, and `ro` (read-only) options. The `bg` option allows the NFS client to restart, even if the NFS server is down. The `soft` option allows NFS client programs to continue running even if the NFS server goes down. The `ro` option is used when the permissions of the pathname you are mounting do not allow writing for "others," as in this case. For more information about `mount` options, see Table 6-1 in *A/UX Network System Administration* and `mount`(1M).

6 **To check that the remote file system is mounted correctly, enter this command:**

```
mount
```

This command displays the currently mounted file systems. You should see something like this:

```
/dev/dsk/c0d0s0 on / type 4.2 (rw,noquota)
```

name-of-NFS-server`:/usr/catman on /usr/catman type nfs (bg,soft,ro)`

You can also use the `df` command to display the currently mounted file systems.

7 **To test that the manual pages are available, enter this command:**

```
man ls
```

If the test succeeds, the text of the `ls` manual page should appear on your screen. If the test fails, see `mount`(1M) for `mount` options and Chapter 6, "Configuring and Managing the Network File System," in *A/UX Network System Administration.*

8 **To unmount `/usr/catman`, enter this command:**

```
umount /usr/catman
```

Modifying the `/etc/fstab` file

When you have verified that NFS is working and you have tailored the `mount` commands to mount the remote data in the way that you want, you should modify the `/etc/fstab` file so that it contains the information required to mount the remote data each time the system is restarted. Here is a sample `/etc/fstab` file:

```
/dev/dsk/c0d0s0 /           ignore rw          1 1
name-of-NFS-server:/usr/catman  /usr/catman nfs   bg,ro,soft   0 0
```

The first entry describes the root file system, and the second entry describes a remote mount. The fields in the `/etc/fstab` file are separated by blanks or tabs. A number sign (`#`) at the beginning of an entry is treated as a comment. Here is an explanation of each field:

- The first field is the name of the host, followed by a colon, and the full pathname of the file system, directory, or file that has been exported.
- The second field is the name of the mount point that you want to mount the exported data on.
- The third field is the type of file system. For remote data, the value is always `nfs`.
- The fourth field is a list of `mount` options. See `fstab`(4) and `mount`(1M) for more information.
- The fifth and sixth fields are always `0` for remote mounts.

You can make a permanent entry in `/etc/fstab` and mount the file system with this command:

```
fsentry -o ro,bg,soft  host:/usr/catman  /usr/catman
```

Verifying the mounting

Once you have modified `/etc/fstab`, verify that the entry is correct.

- **If you used `fsentry`, enter this command:**

  ```
  mount
  ```

  You should see something like this:

  ```
  /dev/dsk/c0d0s0 on / type 4.2 (rw,noquota)
  name-of-NFS-server:/usr/catman on /usr/catman type nfs
  (bg,soft,ro)
  ```

There is more you can do as an NFS client. For example, you can run the `automount` daemon, which mounts files when they are needed and unmounts them when they are not in use. For details, see Chapter 6, "Configuring and Managing the Network File System," in *A/UX Network System Administration*.

▲ **Warning** If your home directory contains a personal System Folder (that is, a directory called `System Folder`) and you export your home directory from an NFS server and mount it on more than one NFS client, be careful when using the Finder to log in. If you use the Finder to log in on more than one NFS client at a time, the Finder running on each client will attempt to update the same files in the personal System Folder. When this happens, the files in the personal System Folder are corrupted, which causes each login session to fail. You will be unable to log in using the Finder until the files in the personal System Folder are rebuilt. ▲

8 Using UUCP to Connect to a Remote System

This chapter discusses the UUCP (UNIX-to-UNIX copy) network. Using UUCP you can connect to a remote computer in order to transfer files between systems and run commands on remote computers. This chapter tells you how to use these UUCP functions. For more information about configuring and managing UUCP, see *A/UX Network System Admininstration.* To use UUCP to send and receive mail, see Chapter 4. This chapter contains these sections:

- introduction to UUCP
- what you need to run UUCP
- finding names of connected UUCP systems
- checking UUCP network status
- copying files
- using `uuto` to copy files
- running commands on a remote computer: `uux`
- tracking remote commands and determining status
- other utilities that use UUCP

You can use this chapter in its entirety or go directly to those sections of use to you.

Introduction to UUCP

The UUCP (UNIX-to-UNIX copy) network was originally developed to communicate between UNIX systems connected by telephone lines or direct serial lines. Now you can also run UUCP over Ethernet using TCP/IP. Using UUCP, you can transfer files between systems, run commands on remote computers, and send and receive mail. These are some of the situations in which you might find UUCP particularly useful:

- Obtaining files from or copying files to other systems that are beyond the reach of your local area network. UUCP works through modems and phone lines, allowing you to communicate with distant computers.
- Obtaining files from or copying files to non-A/UX UNIX systems. The systems involved don't need to be Macintosh computers, nor do they need to have TCP/IP networking. Because UUCP doesn't require special protocols or network configurations, you can use it to communicate with virtually any other UNIX system.
- Running UNIX commands on remote systems with `uux`. Using `uux`, you can gain access to functions not available on your system and can off-load time-consuming tasks to other computers.

When you copy files between systems with UUCP, the system actually "spools," or saves up, your request until the remote system is available to respond. Because of this spooling, there may be a noticeable time lag between the time that you enter the UUCP command and the time at which your request is completed. Later sections in this chapter describe how to check the status of spooled jobs. Figure 8-1 shows A/UX connected to various serial services.

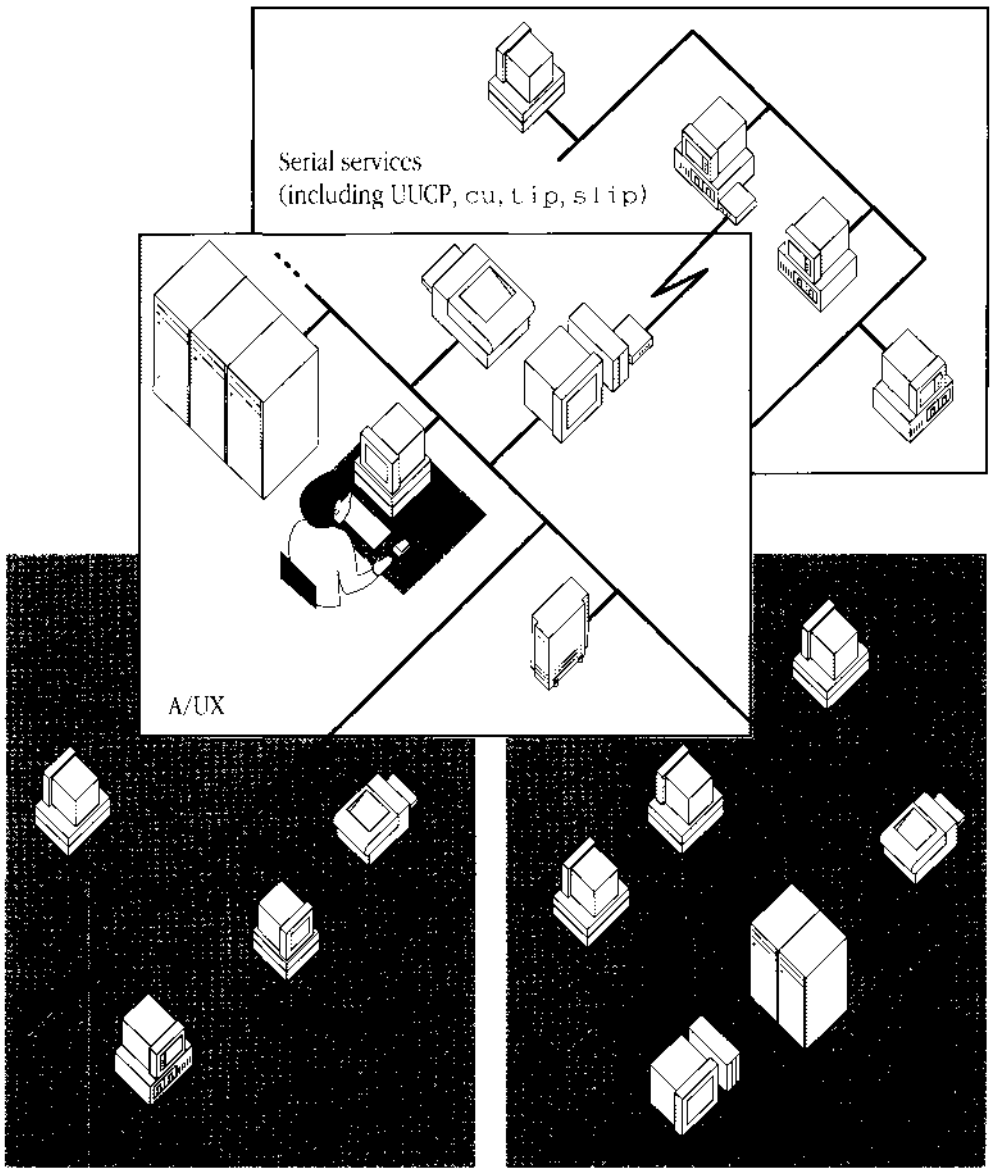


Figure 8-1 A/UX connected to serial services

What you need to run UUCP

The UUCP network is neither a commercial network that you have to "join" nor an open system that anyone can access. The system administrator sets permissions that control who is allowed to use UUCP computers. Before you can use UUCP, you must receive access permission from the remote system, and your computer must be set up correctly. The network administrator determines your access to his or her system and sets permissions accordingly. Many systems restrict UUCP access to a public area or public directory (usually `/usr/spool/uucppublic`). The next section describes how to tell which systems you can access with UUCP. For information about setting up your computer to connect to other systems, see *A/UX Network System Administration* or consult your network administrator.

Finding names of connected UUCP systems

An easy way to see if your system is configured for UUCP is to enter this command in a CommandShell window:

```
uuname
```

If you are set up for UUCP, the system displays the names of systems to which you have UUCP access. If your system isn't set up for UUCP, the command prompt appears; no system names are listed.

Checking UUCP network status

You get the status of the last transfer to each system on the network by entering the `uustat` command in CommandShell. For example:

```
uustat -m
```

Your A/UX system displays the status of the most recent transfer your computer has made to each UUCP-connected system. The result looks something like this:

```
mhb5c       08/10-10:35  CONVERSATION SUCCEEDED
resear      08/20-17:01  CONVERSATION SUCCEEDED
minimo      07/22-16:31  DIAL FAILED
austra      08/20-18:36  WRONG TIME TO CALL
ucbvax      08/20-20:37  LOGIN FAILED
```

This example shows only the most important information. The first column gives the name of the other system involved in the UUCP process, and the next two columns indicate the time and state of the last transfer to each system.

When you're sending files to a system that has not been contacted recently, it's a good idea to use `uustat` to see when the last access occurred because the remote system may be down or out of service.

Copying files

This section tells you the various ways you can copy files to or from a remote system and how to identify remote files.

Identifying remote files

When using UUCP to copy files to or from remote systems, you must use specific naming conventions (syntax) to accurately specify the remote file. There are also some restrictions on pathnames to ensure security. For example, pathnames may not include "`..`" (two periods, which normally refer to the directory above the current directory) because the "double-dot" notation might lead to a restricted directory.

- **If you are using the C shell, specify remote files with this syntax:**

  systemname\!*pathname*

  The *systemname* is the name of a system that UUCP recognizes. All the commands in this chapter are based on the C shell, which is the A/UX default when a new account is created. If you are using the Korn shell or Bourne shell, you don't put a backslash (\) before the exclamation point (!). For information on the various UNIX command shells, see Chapter 8, "Customizing Your Work Environment," in *A/UX Essentials*.

  Naming conventions can apply either to the source filename (where the information is coming from) or to the destination filename (where the information is going). You can enter a pathname in any of these forms:

  - A file's full pathname, such as

    ```
    tahoe\!/usr/sid/stuff
    ```

    In this case, `tahoe` is the system and `/usr/sid/stuff` is the file's full pathname (the file `stuff` in the `/usr/sid` directory).

  - A directory name, as in

    ```
    tahoe\!/usr/sid
    ```

  - The login directory on a remote system, indicated with a tilde (~). The combination `~fred` refers to the login directory of user `fred` on the remote system. For example, if the login directory for user `fred` on the remote system is `/usr/sys/fred`, then the following pathname notation is the same as entering `tahoe\!/usr/sys/fred/names`:

    ```
    tahoe\!~fred/names
    ```

  - The public directory, referenced by a similar use of the prefix `~user` preceding the pathname. For example, if `/usr/spool/uucppublic` is the public directory, then the following pathname notation is the same as `tahoe!/usr/spool/uucppublic/sally/audit`:

    ```
    tahoe\!~/sally/audit
    ```

  - The system uses your current directory (or the login directory on the remote system) by default if the pathnames you enter don't use any other special notation. For example, `tahoe\!stuff` expands to `tahoe\!/usr/spool/uucppublic/stuff`.

Copying a file directly to another system

This command transfers the file `prospects` from your computer to the user with the account `fred`, who is on the system named `tahoe`:

```
uucp prospects tahoe\!~/fred/prospects
```

When copying a file directly from one system to another, you can also use `uuto`, a command that uses `uucp` to copy files between systems but adds some features that make it easier to locate those files on the recipient's system. For details on using `uuto`, see the section "Using `uuto` to Copy Files" later in this chapter.

Copying a file to another system through an intermediate system

You can use UUCP to pass files to systems that your computer does not communicate with directly. To do this, you send the files through intermediate computers.

For example, consider the situation where you are sending the file `prospects` from your system, `yosemite`, to a system named `pinnacle`. There is no direct connection between `yosemite` and `pinnacle`; however, `yosemite` connects to `tahoe`, `tahoe` connects to `donner`, and `donner` connects to `pinnacle`. This situation is illustrated in Figure 8-2.

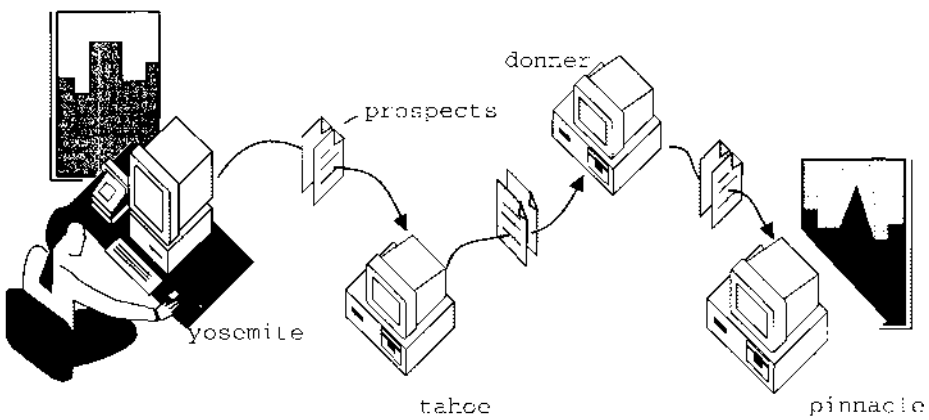


Figure 8-2 Sending the `prospects` file from `yosemite` to `pinnacle`

To accomplish the copy process, enter the `uucp` command, specifying the path from `yosemite`, through `tahoe`, and on to `pinnacle`, as in this command:

```
uucp prospects tahoe!donner!pinnacle!~fred/prospects
```

Copying a file from another system through an intermediate system

The procedure for copying a file from a remote system through an intermediate computer and then on to your system is the same as copying in the other direction. Note that the pathname is still the path from your computer, `yosemite` (also known as the local system), to the remote system, `pinnacle` (even though the file is traveling in the reverse direction, from the remote system to your system). This command copies the file `prospects` from Fred's account on `pinnacle`, through the intermediate systems `donner` and `tahoe`, then on to your system `yosemite`, where the filename is changed to `accounts`:

```
uucp tahoe!donner!pinnacle!~fred/prospects accounts
```

Using your system as an intermediate (switching)

You may arrange file copying in such a way that your local system is the intermediate between two systems, acting as a bridge or switch to create a connection between the two computers. The `uucp` command in this example (illustrated in Figure 8-3) copies the `prospects` file from `shasta`, renames it `accounts`, and places it in the login directory on `tahoe`:

```
uucp shasta!prospects tahoe!accounts
```

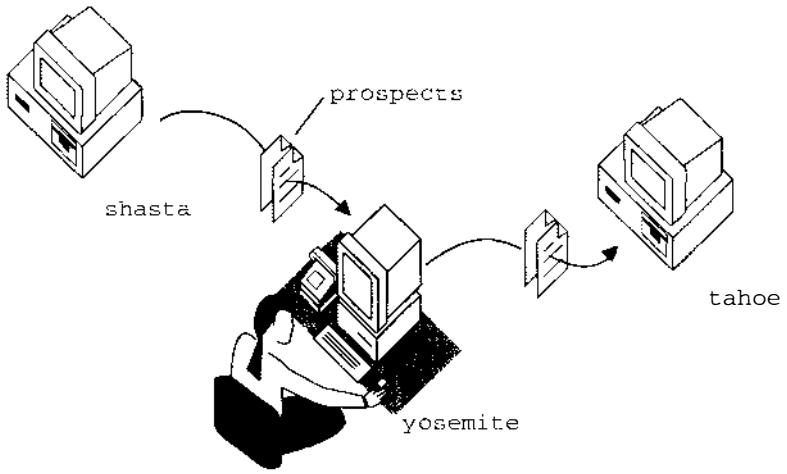


Figure 8-3 Copying the `prospects` file from `shasta` to `tahoe`

Copying multiple files

Using special symbols, known as **metacharacters,** you can create `uucp` commands that transfer any number of files between remote systems and your own. Here is a brief overview of the available metacharacters (for further information see *A/UX Shells and Shell Programming*):

~	A tilde serves as the first part of directory name. It signifies your home directory (if used alone or followed by a pathname below your home directory, such as `~/project/phase1`) or the home directory of another user (if followed by the login name of that user, such as `~lori`).
*	An asterisk matches any character
[ ]	Brackets around a sequence of characters causes the system to match each character one at a time.

Here is a sample command that copies from your current directory all files whose names end in `.c` or `.h` and puts those file copies in the `dir` directory of your login directory on `tahoe`:

```
uucp *.[ch] tahoe!dir
```

You may copy files from a remote system to your local system in a similar manner. For example, the following command copies all files from your login directory on `tahoe` and places those files in the `dir` directory of your local system:

```
uucp tahoe!\* dir
```

If you specify that the files be placed in a directory that doesn't exist, `uucp` creates that directory.

Using `uuto` to copy files

The `uuto` command, which uses `uucp` to copy files, provides automatic mail notification to the recipient and places the files in a hierarchy in the public directory, organized by login and system name. The `uuto` command also provides `uupick`, a command that enables recipients to easily pick up the files sent with `uuto`.

Before you send a file using `uuto`, make sure the file has read permission for others and that the directory containing the file has both read and execute permission for others (see `chmod`(1) in *A/UX Command Reference* for more information).

- **To send a file to someone on a local or a remote computer, enter this command:**

  `uuto` *filename rhost*`!` *username*

  Replace *filename* with the name of the file you're sending; replace *rhost* with the name of the computer you're sending the file to; replace *username* with the login name of the person you're sending the file to. If you're using the C shell, you must precede the `!` with a backslash or quotes.

  Suppose your login name is `ansel` and you're on system `yosemite`. You have a friend on system `tahoe` with a login name of `fred`. You could send the file `prospects` using `uuto` by entering this command:

  ```
  uuto prospects tahoe!fred
  ```

The file is sent to a public directory defined by UUCP. In this example, account `fred` receives this mail:

```
>From nuucp Tue Oct 22 11:09:55 1991
/usr/spool/uucppublic/receive/fred/tahoe\
//prospects from yosemite!ansel arrived
```

For more details, see `uuto`(1) in *A/UX Command Reference.*

- **To send a file to a different account on your own computer, enter this command:**

  `uuto` *filename username*

  When your file arrives at its destination, `uuto` informs the recipient.

- **To send a file and receive a message that your file has been delivered, enter this command:**

  `uuto -m` *filename rhost*`!` *username*

- **To check on the status of a file you sent, enter this command:**

  `uustat`

- **To retrieve a file sent with `uuto`, enter this command:**

  `uupick`

  The `uupick` command searches the public directory for any file sent to you. If a file is found, you're first prompted with `?`, the host name, and the filename. `uupick` then waits for your response. You can respond with one of these commands:

`m`	Save the file in the current directory.
`m` *directory*	Save the file in the specified directory.
`d`	Delete the file.
`q`	Quit `uupick`. Any unmoved or undeleted files remain in the public directory until the next time you use the `uupick` command.

Running commands on a remote computer: uux

You may run commands on a remote system (remote execution) using uux. The uux command is useful in situations where a remote system offers commands not available on your local computer, or when you want to run time-consuming tasks on a different system so that your computer remains available.

The following command runs diff on the password files on tahoe and shasta and places the result in pw.diff:

```
uux "diff tahoe\!/etc/passwd shasta\!/etc/passwd > pw.diff"
```

You need to use double quotes with uux when your command includes special shell characters such as <, >, ;, or |.

uux always notifies you when the remote execution is complete unless you suppress notification with the -n option. See uux(1) in *A/UX Command Reference* for more information.

Tracking remote commands and determining status

The following sections describe how you can determine the status and transmission results of uucp and uux commands.

Receiving notification

The uucp command reports transmission results through the mail command if you choose the -m option, which causes the system to send you mail when the copy process is complete. This option is useful when you're distributing files to other systems because you don't have to do a remote log in to the destination system to determine if the file arrived.

Checking job status: `uustat`

The `uustat` command lets you check on one or all jobs that have been queued. You may find your jobs' numbers and get the status for all jobs in progress by using the `uustat` command:

```
uustat
```

Then the system prints the job numbers for the current jobs:

```
macauxZ6eb1 11/22-12:47 S macaux lori 56940 /users/lori/alpha
macauxZ6eb2 11/22-14:49 S macaux ruth 56940 /users/ruth/list
```

In this example, the two jobs are queued.

You may use the job number to get the status of a particular job. For example, this command displays status information on job 0711:

```
uustat -j0711
```

The system responds:

```
0711 tom mhtsb 07/30-02:18 07/30-02:18 JOB IS QUEUED
```

There are several possible status messages for a given job; the most frequent ones are `JOB IS QUEUED` and `JOB COMPLETED`. For more information, see `uustat`(1) in *A/UX Command Reference*.

Job control

Using the unique job ID generated for each `uucp` or `uux` command, you can control jobs in several ways.

Terminating jobs

You can terminate a job that consists of transferring many files from several different systems using the `-k` ("kill") option of `uustat`:

```
uustat -kmacauxZ2f8d
```

If any part of the job has left the local system on its way to the destination system, only the *remaining* parts of the job on the local system are terminated.

Requeuing a job

`uucp` clears out its working area of jobs on a regular basis (usually every 72 hours) to prevent the buildup of jobs that cannot be delivered. You may use the `-r` option to force a job's date to become the current date, thereby lengthening the time that `uucp` attempts to transmit the job. Note that the `-r` option only postpones the job's deletion until the next cleanup. You may use the `-r` option over again, however, extending the job's lifetime even further.

Other utilities that use UUCP

Several utilities rely on `uucp` or `uux` to transfer files to other systems. The most common is USENET. A large number of UNIX sites worldwide participate in USENET (or `netnews`), a distributed electronic bulletin board with a wide range of topics. To participate, your system administrator must get the USENET software from a USENET site (`netnews` is not available through Apple Computer). Check with your system administrator to see whether your site is connected with the USENET.

The Office Automation System (OAS) uses `uux` to transmit electronic mail between systems in a manner similar to the standard `mail` command. Some sites have replaced utilities, such as `lp`, with shell **scripts** that invoke `uux` or `uucp`. Other sites use the UUCP network as a backup for higher speed networks.

9 Using `cu` to Connect to a Remote System

This chapter discusses how to use the `cu` program to connect to a variety of other systems, both UNIX and non-UNIX. A/UX has other programs such as `tip` and `kermit` that provide functions similar to the `cu` program. For information, see `tip`(1C) and `kermit`(1C). This chapter includes these sections:

- selecting a remote session option
- connecting to a remote system
- switching between local and remote commands
- sending local output to a remote system
- copying files
- using `cu` with non-A/UX computers
- closing the `cu` connection

You can read this chapter in its entirety or go directly to those sections of use to you.

Selecting a remote session option

A/UX offers a variety of options for using your computer to gain access to a remote system and working on that remote system as though you were actually in front of it. Once you've connected, you can transfer files back and forth between computers and do things such as send the output of a local command to the remote system or capture the output of a remote system command on your local computer. Practical examples of these uses are included later in this chapter.

The `cu` program allows you to connect with a variety of other systems, both UNIX and non-UNIX. Connection is usually through a serial line, either through a direct connection or with a modem. Because `cu` does no error checking on data transfers, it's not the best choice for large file transfers over (possibly noisy) phone lines because there's no guarantee that one computer receives exactly what the other computer sends.

To understand a typical use for `cu`, imagine yourself as the user in Figure 9-1, logged directly into a computer named `rio`. On `rio`, you run `cu` to connect through a modem to a remote computer, `paris`. After logging in to `paris`, you can work on `paris` exactly as if your terminal were connected directly to it. In addition, if `paris` is a UNIX system, you can copy files in either direction between the two systems.

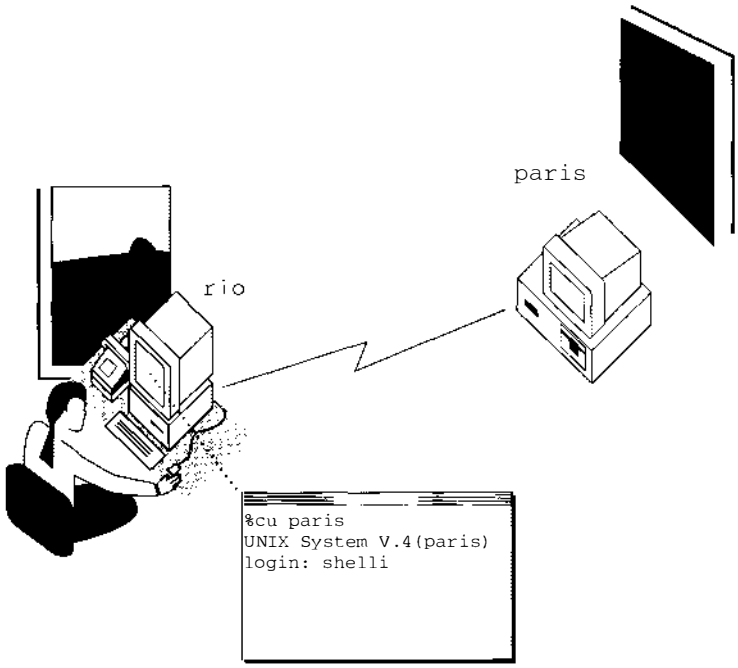


Figure 9-1 A simple `cu` configuration

In Figure 9-2 you run `cu` once again, this time on the remote computer, `paris`. This time, you also connect to a remote mainframe, `berlin`, and interact with that system as if you were actually there, logged in in person. Once again, you can copy files between `paris` and `berlin`; you can also copy files between `rio` and `berlin`, or (as before) between `rio` and `paris`.

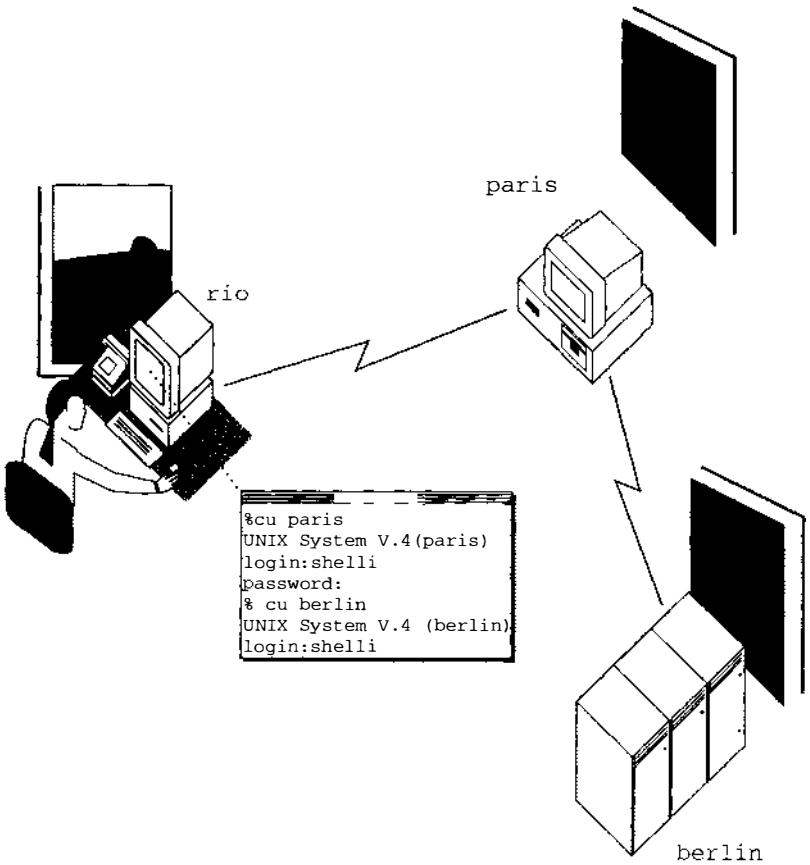


Figure 9-2 Using `cu` remotely

Connecting to a remote system

In order to connect to a remote system with `cu`, you need to have this information:

- the name of the port (such as the serial port `/dev/tty0`) through which your system is connected to the remote computer
- the speed at which the connection runs (such as 1200 baud)
- if you're going through a modem connection, procedures for getting the modem to dial the phone number

The `cu` command typically looks like this:

```
cu -l/dev/tty0 -s1200
```

`-l` specifies the line (the serial port `/dev/tty0`), and `-s` specifies the speed for the connection to the remote system (in this case, 1200 baud).

Once your system establishes a `cu` connection with the remote system, the login message of the remote computer appears on your screen. For example, this might appear if you connect to a remote computer named `paris`:

```
UNIX System V.2 (paris)
login:
```

At this point, you can log in to `paris` and use it just as if you were actually at a terminal connected directly to it. Additionally, `cu` allows ASCII file transfer between computers.

You should note that you may not have any choice in the serial line used by `cu` because this is configured by your system administrator. If the line you choose is not available, `cu` responds with an error message. For instance, if the line `/dev/tty0` (as requested above) is not configured to allow `cu` access, you get this message:

```
Connect failed: Requested device not known
```

Ask your system administrator for the name of the line you should use and try again.

To guard against multiple users connecting to remote computers through the same line, `cu` opens modems and terminal lines with exclusive access and honors UUCP's locking protocol (see Chapter 8, "Setting Up the UUCP System," in *A/UX Network System Administration,* and `uucp`(1C) in *A/UX Command Reference*).

You may also specify a telephone number as a command line argument. If the telephone number is the *only* argument, `cu` will attempt to connect to an available line and dial the specified number. Or you can specify a remote system name (as defined in the system file `/usr/lib/uucp/Systems`) on the command line. (See Chapter 8, "Setting Up the UUCP System," in *A/UX Network System Administration* for more information.) For example, if your system is appropriately configured, these may be acceptable `cu` command lines:

```
cu 8452583
cu paris
```

Switching between local and remote commands

Once you have entered the `cu` command, everything you type goes to the remote computer as if you were actually typing on the remote system. At some point during your `cu` session, you may want to run a command on the local computer. One way to do this is to click another CommandShell window on your A/UX system, making that local window active. There are two other ways: by escaping to an interactive shell, or by instructing the local computer to run a single command.

- **To escape `cu` to an interactive shell on the local computer, enter this command:**

  ```
  ~!
  ```

  When you exit that shell, you return to `cu`.

  To run a single command on the local system, use `~!` at the beginning of the command. For example, the following command gives a listing of the current directory on the local system sequence:

  ```
  ~!ls
  ```

As a further example, you might choose to sort a file called `staff.91` and store the sorted file as `newstaff.91` on the local system prior to copying it to the remote computer. Here is an example of how you could do that sort on the local system:

```
~!sort staff.91 > newstaff.86
```

The command is run in a subshell through `sh -c`. Accordingly, the command should conform to the syntactic conventions of the Bourne shell. When the command is completed, you return to `cu`.

Sending local output to a remote system

With `cu` you can run a command on the local system and send any output to the remote system. For example, this command sends the contents of the local file `test.sh` to the remote system:

```
~$cat test.sh
```

△ **Important** The contents of the file sent with `~$cat` are *not* put into a file but are simply sent to the other computer. The remote system will attempt to interpret the file's contents; therefore, a file that you send this way should contain commands or data meaningful to the remote system. To copy files, see the following section. △

There are two commands recognized by the `cu` receive process. If the remote computer sends a line beginning with the sequence

~>:*filename*

all the subsequent output, up to the next line beginning with the sequence `~>`, is diverted into a file having the name *filename*. Similarly, output is appended to *filename* if the sequence

~>>:*filename*

is encountered.

Copying files

To send a file from the local computer to the remote system, use the `~%put` command. For example, this command copies the file `chap.5` from the current directory on the local system to the current directory on the remote system:

```
~%put chap.5
```

If a file with that name already exists on the remote computer, the version you are copying from the local system replaces the one on the remote system. If there is no file `chap.5` in the current directory on the local system, `cu` gives an error message. If you encounter this message, you may want to change to a different directory before retrying. For example:

```
~%cd /users/doc/rachel/book
```

You can store the file you are copying under a different name on the remote system. For example, this command copies `chap.5` from your local system and stores the copy on the remote system in a file named `sec.5`:

```
~%put chap.5 sec.5
```

To transfer a remote file to the local computer, use the `~%take` command. For example,

```
~%take appendix.3
```

This works exactly like the `~%put` sequence, but in reverse.

Using `cu` with non-A/UX computers

You can use `cu` to communicate through a serial line with computers that do not run the A/UX operating system. However, in cases where the remote computer is not running A/UX (or any similar derivative of the UNIX operating system), certain `cu` commands won't work properly. In particular, the commands `~%take` and `~%put` probably will not

function correctly, since they require that the remote computer understand at least the three commands `stty`, `echo`, and `cat`. In such cases, it may be possible to achieve almost the same effect in a different way. For example, if you want merely to send to the remote device a string of characters stored in the file `test.ps` on the local computer, you may invoke the following command:

```
~$cat test.ps
```

This runs the A/UX command `cat test.ps` and sends the file contents to the remote computer. Presumably, `test.ps` contains commands or data that are meaningful to the recipient.

Generally, there is no easy way to get the remote computer to simulate the effect of the `~%take` command. The closest you can come to such an operation may be to have the remote computer echo an appropriate command (**~>:** *filename*) to cause the local computer to capture the subsequent output and save it in an A/UX file. This may or may not work for your purposes.

Closing the `cu` connection

Once you have finished the `cu` session, log out, and then enter this command:

```
~.
```

The tilde-period sequence drops the existing connection and exits `cu`, returning you to the command shell on the local computer.

10 Using MacX on A/UX

A/UX comes with two Apple X Window system products: MacX 1.1.7 and X11 for A/UX. You'll want to use MacX if you're new to the X Window System and you're most comfortable with the Macintosh interface, or if you want your favorite Macintosh applications to share the desktop with X applications. You'll want to use X11 if you're an experienced programmer who prefers the traditional X Window System interface. The *A/UX Installation Guide* tells you how to install either or both of these products. The purpose of this chapter is to give you guidelines for using MacX on A/UX. For details about how to use the many MacX features and capabilities on A/UX, see *MacX User's Guide*. For information about X11 on A/UX, see *X11 User's Guide for A/UX*.

This chapter describes the features of MacX that are specific to its use under A/UX. It contains these sections:

- special terms used in this chapter
- about MacX
- before you begin
- starting a MacX session

- opening client applications
- disconnecting a client application
- ending a MacX session
- additional reading

You can read this chapter in its entirety or just refer to the sections of use to you.

Special terms used in this chapter

Here are definitions of several terms unique to this chapter. These terms are standard in the Apple documentation that describes the X Window System for A/UX:

- **display server:** The X Window System program that runs on your computer and determines the graphics that you see on your computer screen, including the windows that appear on your screen when you display client applications. The display server acts as a traffic cop, controlling the flow of many levels of communication between you (using the keyboard and mouse), your computer, other computers, client applications, and other X programs.
- **client application:** An application developed for the X Window System that performs a specific task, such as the `xclock` client application that displays a clock. A client application may also provide a certain tool, such as a word processor. When you receive MacX as a part of A/UX, you receive the `xcalc` client application, a calculator that looks like a Macintosh desk accessory and allows you to perform arithmetic using the keyboard and mouse.
- **local client application:** A client application that resides and runs on your own computer. For example, when you install X11 for A/UX, you install a number of client applications, such as `xclock`. When `xclock` runs on your computer, you refer to it as a local client application.
- **remote client application:** A client application that resides and runs on a computer on a network. For example, suppose the `xcalc` client application is running on a VAX computer to which you have access over a network. If you use MacX to display that `xcalc` on your computer's screen, you would refer to `xcalc` as a remote client application.
- **MacX settings document:** The name for a file that MacX creates each time you start the MacX server. You can store any preferences you specify or commands you create by saving them in a settings document; in effect, you are saving your MacX environment. To return to a MacX environment that you saved in a settings document, you open the document in the A/UX Finder.

For additional terms and definitions, see the glossary in the *MacX User's Guide.*

About MacX

Apple Computer originally developed MacX to provide Macintosh users, working under the Macintosh OS, with access to X Window System applications. The X Window System, nicknamed X, was designed by a team of researchers from industry and from the Massachusetts Institute of Technology (MIT). X allows UNIX users to display multiple applications, called *client applications,* in separate windows on their computer screens. In addition, the structure of X makes these applications ideal for use over a network. MacX provides the same capabilities as X but with an added dimension: the Macintosh user interface. For your convenience, MacX is now shipped with A/UX.

MacX allows a user working under the Macintosh OS to connect to computers on a network running X client applications. A user runs MacX on a Macintosh computer and uses it to display on the screen a client application that is running on another computer. MacX displays the remote client application on the user's screen in an easy-to-use Macintosh-style window. The user can then manipulate the client application by means of standard Macintosh features such as pull-down menus, dialog boxes, and windows with size boxes.

These same capabilities are available to you when you use MacX under A/UX. You also have an additional advantage as an A/UX user: the ability to connect to client applications running on your own computer, called *local client applications.* You can use MacX to display client applications running on your local computer and on remote computers on a network, all at the same time.

Figures 10-1 and 10-2 illustrate the differences between using MacX under the Macintosh OS and using MacX under A/UX.

Figure 10-1 shows how you can use MacX on a Macintosh computer running the Macintosh OS in order to connect to remote client applications on other computers on a network.

Figure 10-2 shows how you can use MacX on a Macintosh computer running A/UX to connect to local client applications as well as remote client applications.

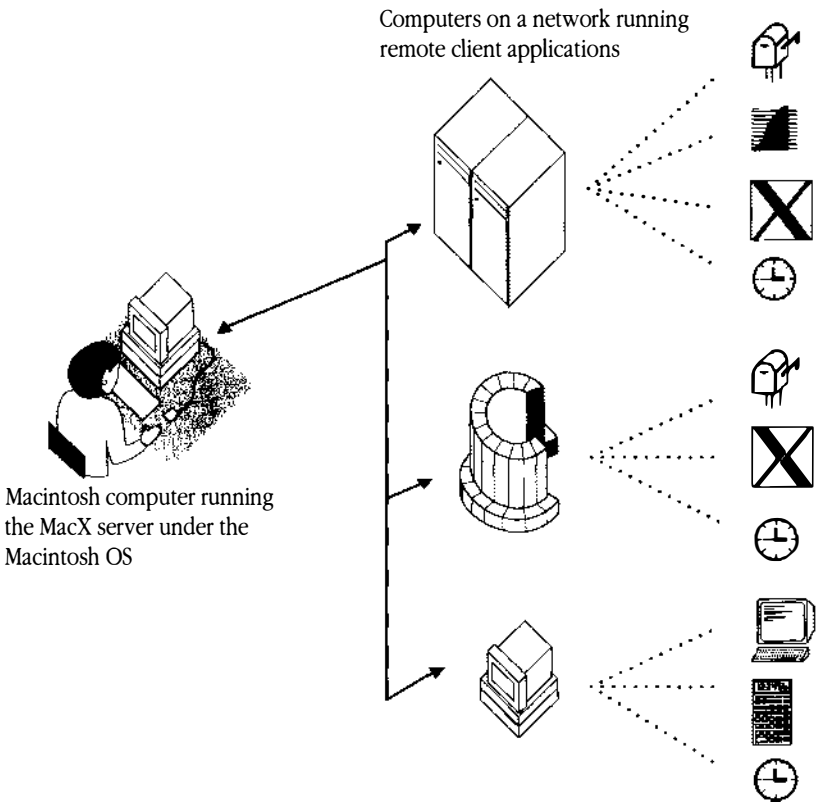


Figure 10-1 MacX running under the Macintosh OS

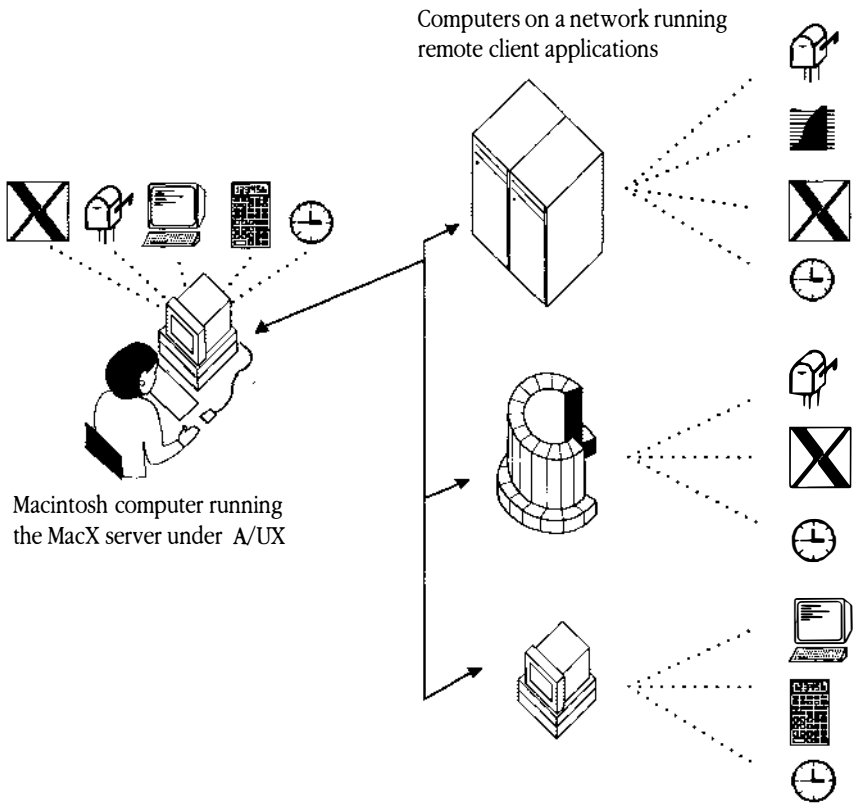


Figure 10-2 MacX running under A/UX

Before you begin

Before you can use MacX on your computer, you need to install it on A/UX. The *A/UX Installation Guide* contains installation instructions for A/UX, MacX, and X11 for A/UX.

◆ **Note** MacX includes the `xcalc` client application so that you can test MacX immediately after installation. If you want additional client applications, you can install X11 for A/UX. X11 comes with A/UX. ◆

Starting a MacX session

To start the MacX server in A/UX, follow these steps:

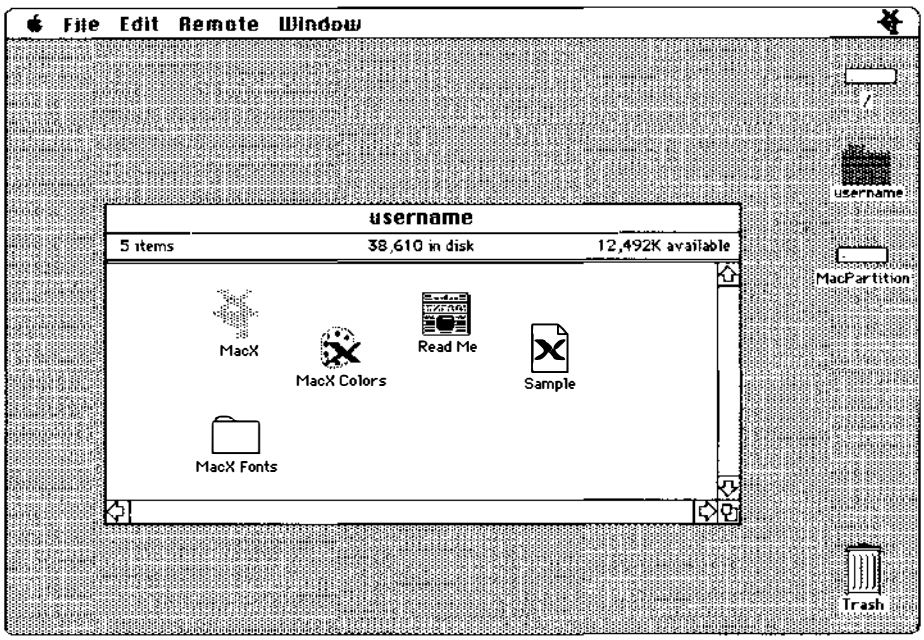
1 **Log in to your personal A/UX account.**

2 **Open the folder that contains the MacX application.**

If MacX was preinstalled, open the folder called `/mac/MacX`.

3 **Open the MacX application icon to start the MacX server.**

The About MacX window appears. After the About MacX window disappears, your desktop should look something like this:



The titles in the menu bar are the names of the menus you have available in MacX. You also see the MacX icon at the right edge of the menu bar. This icon indicates that the MacX server is now running.

See the next section, "Opening Client Applications," for step-by-step instructions on how to open client applications. See the section "Ending a MacX Session" for information about quitting MacX.

Opening client applications

You can open client applications in three ways when you are using the MacX server in A/UX:

- Open a client application from the A/UX Finder.
- Use the Remote Command window in MacX.
- Enter the name of the client application at the A/UX command line. You can do this from two places:

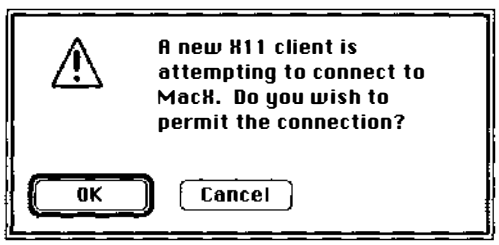
- a CommandShell window
- an `xterm` window

These three methods use `xcalc` as the example of how to open a client application. They are described in the sections that follow. Before you use one of these methods, read about how you give an application access to your computer screen.

▲ **Warning** You must have MacX running before you try to open an X client application from the A/UX Finder. If you don't, the X client application displays the error message `Can't Open Display . . .` in a CommandShell window. See the previous section, "Starting a MacX Session," for information about opening MacX. ▲

Giving client applications access to your computer

Each time you open a client application, MacX, by default, displays an Access Control alert box, as shown below. This alert box warns you that a new client application is trying to connect to MacX. You are required to grant or deny the new client application access to your screen. You can disable this alert box, thereby granting all client applications immediate access to your screen.



◆ **Note** You should disable Access Control before going on to the following sections that describe the different ways to open client applications. ◆

■ **To disable the alert box, choose Access Control from the Remote Menu.**

The check mark next to Access Control disappears. Access Control is now disabled.

Opening a local client application from the A/UX Finder

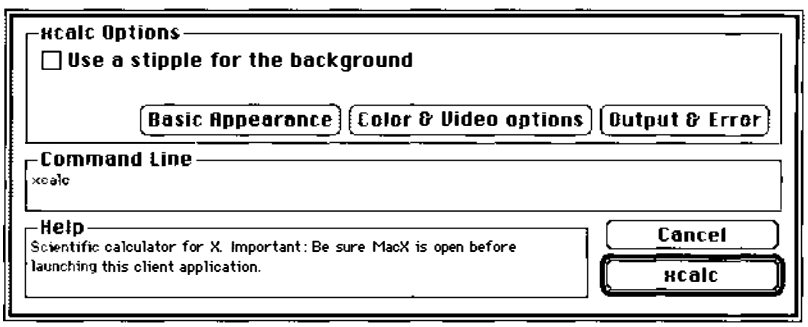
Follow these steps to open the `xcalc` local client application by opening the `xcalc` icon from the A/UX Finder. Once you have installed other client applications, such as those that come with X11, you can follow these same steps to open those client applications from the A/UX Finder.

1 **From the A/UX Finder, open the folder that contains `xcalc`.**

If you installed the MacX package during your A/UX installation, `xcalc` is in the `/mac/MacX` directory. If you installed additional X products, `xcalc` will probably be in the `/usr/bin/X11` directory.

2 **Open the `xcalc` application icon.**

You see a CommandShell window, and then the `xcalc` Commando dialog box appears with information about `xcalc` and with the available options. You can select options in the dialog box that determine how `xcalc` looks and behaves. For this example, you don't need to select any options.



3 **Click `xcalc` to open the calculator.**

You see a CommandShell window as the `xcalc` client application starts. To see the `xcalc` calculator, you must make the MacX application active.

If you did not disable Access Control, a warning icon flashes in the top left corner of the menu bar.

4 **Choose MacX from the Applications menu at the far right of the menu bar.**

◆ **Note** The Access Control alert box may appear at this point. Click OK to continue. ◆

The MacX application is now active. The `xcalc` calculator appears in a few moments. To open another client application while MacX is running, from the Finder just double-click the client application you want to open and follow Steps 3 and 4 described above.

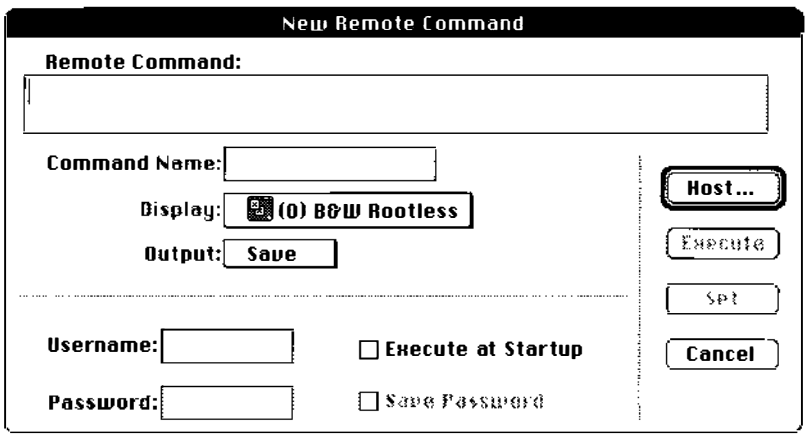
Opening a local client application using the Remote Command window

The `xcalc` client application is included as part of the MacX distribution. Follow these steps to open the `xcalc` client application by using the Remote Command window in MacX. Once you have installed other client applications, such as those that come with X11, you can follow these same steps to open those client applications by using the Remote Command window.

◆ **Note** Detailed descriptions of each element in the Remote Command window are not included in this chapter. See Chapter 2, "Getting Started," in the *MacX User's Guide* for a complete discussion of the Remote Command window. ◆

1 **Choose New Command from the Remote menu or press COMMAND-R.**

The New Remote Command window appears. In the New Remote Command window, you can move easily from one field to the next by pressing the TAB key. Or you can click in a field to select an insertion point.



2 **Type the following command line in the Remote Command field to open the `xcalc` client application:**

pathname`/xcalc -display "®display"`

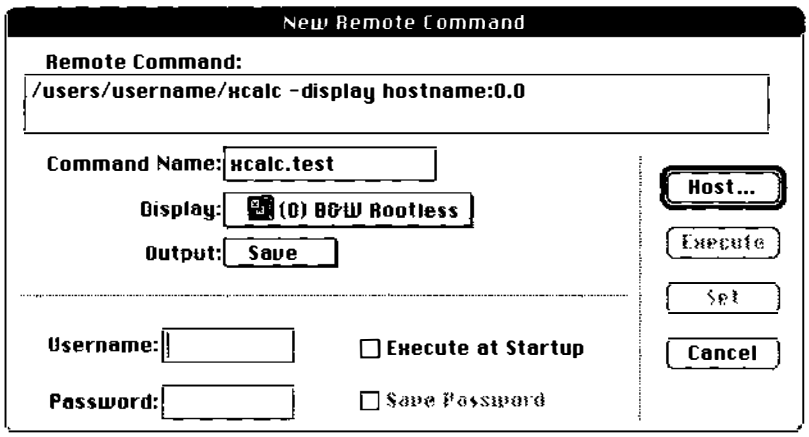
As you type, replace *pathname* with the pathname that specifies the location of the client application. For example, if you installed X11, the pathname would be `/usr/bin/X11`.

The `®display` option is substituted with the display station name or network address used by the host computer to identify your Macintosh. (To type ® on U.S. keyboards, hold down the OPTION key and type R.)

Do not press RETURN at the end of the line.

3 **Press TAB and type `xcalc.test` in the Command Name field.**

You can name the command anything you want.



4 **Press TAB and type your A/UX user name in the Username field.**

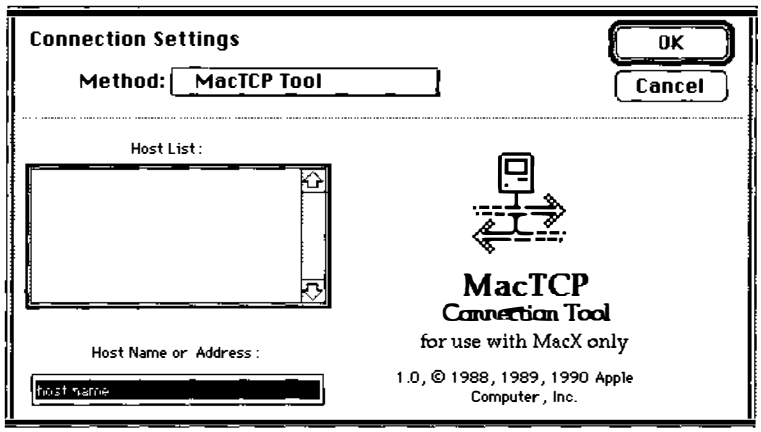
Since you are connecting to client applications on your local computer, you enter the same name you use to log in to A/UX. For example, if your login name is Lisa, type `lisa`.

5 **Press TAB and type your A/UX password in the Password field.**

If you do not have a password, just press RETURN.

6 **Click Host.**

The Connection Settings dialog box appears.



7 **Type the name or IP address of the host in the Host Name or Address field.**

The host computer is the computer on which MacX resides. In this example, type either the name of your computer or `localhost`.

8 **Click OK or press RETURN.**

You return to the New Remote Command window.

To run the command now, go on to Step 9.

If you don't want to run the command at this point, you can click Set and your command will be saved. The name that you gave the command appears in the Remote menu. You can run the command later by selecting its name from the Remote menu.

9 **Click Execute or press RETURN.**

In a few moments, the `xcalc` calculator appears on your screen.

◆ **Note** The Access Control alert box may appear at this point. Click OK to continue. ◆

You can use these same steps to open other client applications that you have on your computer. Just replace `xcalc` with the name of the client application you want to open and follow the steps described above.

Opening a local client application from CommandShell or `xterm` command line

To open a client application, you can type a command line similar to the one that you typed in the Remote Command window in either a CommandShell window or an `xterm` window.

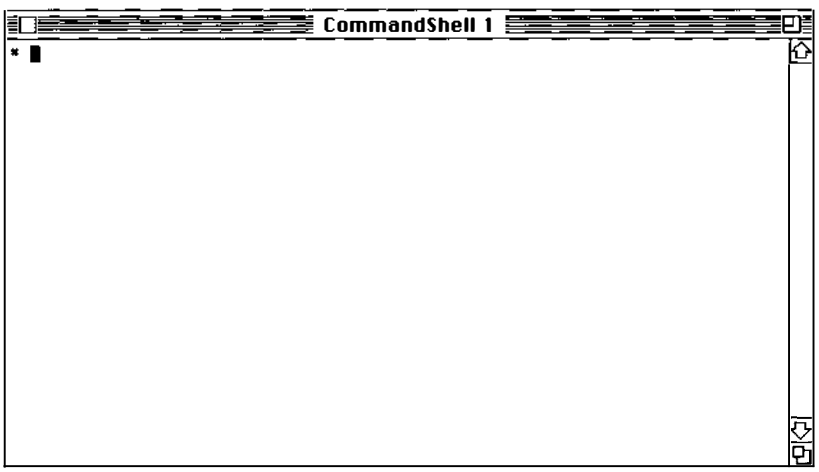
Using a CommandShell window

Follow these steps to open the `xcalc` application from a CommandShell window. Once you have installed other client applications, such as those that come with X11, you can follow these same steps to open those client applications by typing command lines in a CommandShell window.

These steps assume that you have already logged in to A/UX, opened MacX, and are in the A/UX Finder.

1 **Choose CommandShell from the Applications menu.**

A CommandShell window appears with a command prompt.



2 **Enter the following command in the CommandShell window to open the `xcalc` client application:**

pathname`/xcalc &`

As you type, replace *pathname* with the pathname that specifies the location of the client application.

Typing the ampersand (`&`) after the command puts the process running the client application in the background and returns a command prompt immediately. This allows you to enter additional commands while the client application is still running.

If you did not disable Access Control, a warning icon flashes in the top left corner of the menu bar.

3 **Choose MacX from the Applications menu.**

◆ **Note** The Access Control alert box may appear at this point. Click OK to continue. ◆

The MacX application is now active. The `xcalc` calculator appears in a few moments.

You can use these steps to open other client applications that you may have installed on your computer. Just replace `xcalc` with the name of the client application you want to open and follow the steps described above.

Using an `xterm` window

Follow these steps to open the `xcalc` client application from an `xterm` window. Once you have installed other client applications, such as those that come with X11, you can follow these same steps to open those client applications by typing command lines in an `xterm` window. You can get `xterm` by installing X11. See *X11 User's Guide for A/UX* for more information.

1 **Open the `xterm` client application.**

To open `xterm`, use any of the methods described previously. Remember that MacX must be running in order for you to be able to open a client application such as `xterm`.

An `xterm` window appears with a command prompt.

2 **Enter the following command in the `xterm` window to open the `xcalc` client application:**

pathname`/xcalc &`

As you type, replace *pathname* with the pathname that specifies the location of the client application.

Typing the ampersand (`&`) after the command puts the process running the client application in the background and returns a command prompt immediately. This allows you to enter additional commands while the client application is still running.

3 **Choose MacX from the Applications menu.**

◆ **Note** The Access Control alert box may appear at this point. Click OK to continue. ◆

The `xcalc` client application appears on your screen in a few moments.

You can use these steps to open other client applications that you have on your computer. Just replace `xcalc` with the name of the client application you want to open and follow the steps described above.

Disconnecting a client application

The Kill Client command disconnects client applications. You can quit `xcalc` by positioning the mouse over the button labeled "AC" and pressing the right arrow key (right mouse button equivalent).

▲ **Warning** Be sure to save any work you've done in a client application before disconnecting it. Client applications disconnected by using the Kill Client command may not clean up and save work. ▲

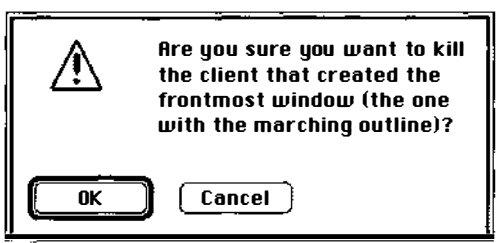
Follow these steps to disconnect a client application:

1 **Click inside the client application window to make it active.**

Horizontal stripes appear in the title bar of the client application window to indicate that the window is active.

2 **Choose Kill Client from the Window menu or click the close box in the window.**

You can also press COMMAND-W. A marching dashed line outlines the window you've selected and indicates you are disconnecting this client application. MacX displays the alert box to confirm that you really want to disconnect the client application.



3 **Click OK to disconnect the client.**

Click Cancel if you don't want to disconnect the client.

After you click OK, the client application is disconnected and its window disappears.

Ending a MacX session

You can end a MacX session in one of two ways:

- Quit the MacX application.
- Close a settings document without quitting MacX.

The sections that follow describe these two methods; use the one best suited to your task.

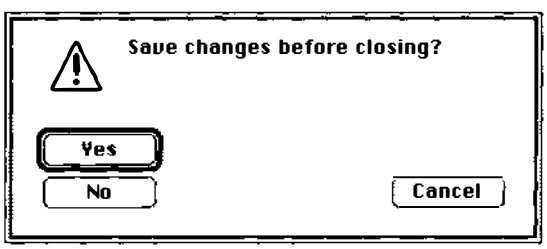
▲ **Warning** Before quitting MacX, save your work in each client application that is still running. Each client application may have a different way of saving information. See the documentation that came with a particular client application to find out how to save your work. ▲

Quitting the MacX application

To quit the MacX application, follow these steps.

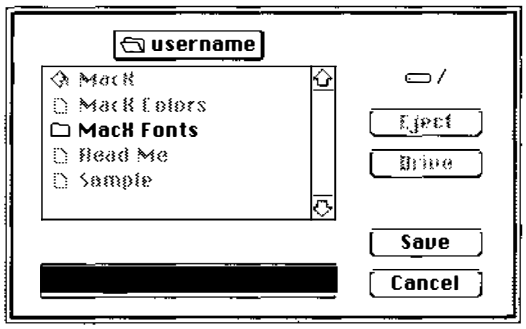
1 **Choose Quit from the File menu or press COMMAND-Q.**

If you've made changes to the current MacX setting document, such as creating a command, an alert box appears.



2 **Click Yes to save your changes to MacX in the current session.**

MacX presents a Save As dialog box.

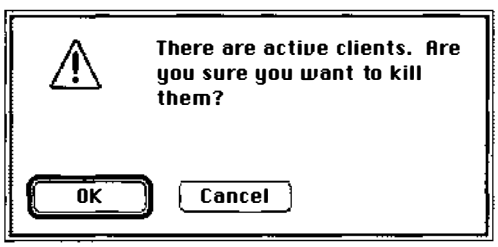


If you don't want to save the changes you've made to MacX in the current session, click No. Click Cancel to return to your current MacX session.

3 Type a name for your MacX session, and click Save.

The name you type automatically replaces "Untitled," the default name for the settings file in which your current MacX session is saved. MacX creates a document icon with the name you've chosen. Later, you can return to the same environment that you created in that particular MacX session by opening the MacX settings document icon from the A/UX Finder.

If you have client applications running, an alert box appears.



▲ **Warning** If you have a client application in which work is underway, such as a word processor, spreadsheet, or terminal emulator, don't quit this way. Any unsaved work you may have done with the client application will be lost. For more information, see "Closing Clients" in Chapter 2 of the *MacX User's Guide.* ▲

4 Click OK to disconnect all active client applications.

All active client applications are disconnected and disappear from your screen.

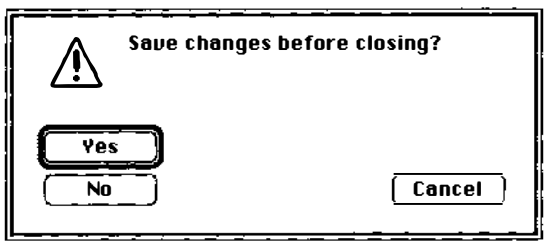
Closing a settings document without quitting MacX

You can close your current MacX session without quitting the MacX server. Any client applications you have running are disconnected when you close a MacX session.

Follow these steps to close a MacX session:

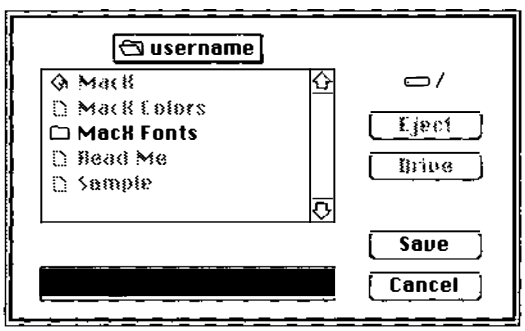
1 **Choose Close "Untitled" from the File menu.**

If you've made changes to the MacX environment, such as creating a command, an alert box appears.



2 **Click Yes to save your changes to MacX in the current session.**

MacX presents a Save As dialog box.



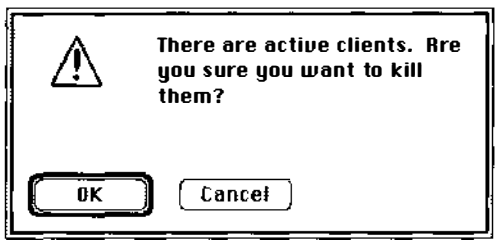
If you don't want to save the changes you've made to MacX in the current session, click No. Click Cancel to return your current MacX session.

3 **Type a name for your MacX session and click Save.**

The name you type automatically replaces "Untitled," the default name for the settings

file in which your current MacX session is saved. MacX creates a document icon labeled with the name you've chosen.

If you have client applications running, you see an alert box.



▲ **Warning** If you have a client application in which work is underway, such as a word processor, spreadsheet, or terminal emulator, don't disconnect it this way. Any unsaved work you may have done with the client application will be lost. For more information, see "Closing Clients" in Chapter 2 of *MacX User's Guide.* ▲

4 **Click OK to disconnect all active client applications.**

All active client applications are disconnected and disappear from your screen.

The MacX server continues to run. To continue working in MacX, you must choose New from the File menu to create a new MacX settings document. Any new commands that you create will be stored in this new settings document.

Additional reading

For more information about MacX and X11 for A/UX, consult the following guides:

- *MacX User's Guide* provides detailed descriptions of the MacX server. It includes instructions on using MacX in the Macintosh OS. Most of the information in the guide is applicable to MacX users.
- *A/UX Installation Guide* presents instructions on installing MacX and X11 on A/UX.
- *X11 User's Guide for A/UX* describes X11, Apple's standard implementation of MIT's X Window System. X11 offers users an X development environment complete with X libraries, toolkit, client applications, and a new Version 11, Release 4 server with dramatically increased speed.
- *X11 Command Reference for A/UX* and *X11 Programmer's Reference for A/UX* are primarily references for X developers and programmers. You can order these manuals separately.

You receive on-line copies of the manual (man) pages with A/UX 3.0. Manual pages for the X commands include descriptions, command syntax, options, examples, and references to related information. You can order manual pages separately if you'd also like paper copies.

If you need information about A/UX, see *Road Map to A/UX*. It presents an overview of A/UX features and provides descriptions of the available A/UX guides, including ordering information.

The Apple Publishing System

A/UX Networking Essentials was written, edited, and composed on a desktop publishing system using Apple Macintosh computers, an AppleTalk network system, Microsoft Word, and QuarkXPress. Line art was created with Adobe Illustrator. Proof pages were printed on Apple LaserWriter printers. Final pages were output directly to separated film on a Linotronic L300. PostScript, the LaserWriter page-description language, was developed by Adobe Systems Incorporated.

Text and display type are Apple's corporate font, a condensed version of ITC Garamond. Bullets are ITC Zapf Dingbats. Some elements, such as program listings, are set in Apple Courier, a fixed-width font.

Writer: Verna Watterson

Contributing Writers: Kristi Fredrickson, Diane Goulder, Linda Kinnier, Paul Panish, Kathy Wallace, and Laura Wirth

Developmental Editor: Paul Dreyfus
Design Director: Lisa Mirski
Art Director: Joyce Zavarro
Production Editor: Michael Oates

Special thanks to Vicki Brown, John Caldwell, Allen Parsons, and Lisa Rose for their technical expertise and continued interest, to Kathy Wallace for her thorough reviews and seasoned point of view, and to Chris Wozniak for her management.

Glossary

absolute pathname The complete name of a file, given by listing all the directories leading down to that file, starting from root (/) and concluding with the filename itself. The directories leading to the file are separated from each other and from the filename by slashes. For example, `/etc/passwd` is the absolute pathname of the system password file, `passwd`, located in the `etc` directory beneath the root (/) directory.

alias An alternative name used in UNIX to invoke or identify a command, a network host, a list of users, or some other applicable entity. Used in `csh` or inside `mailx`.

AppleTalk A set of networking protocols that works across LocalTalk (serial) and EtherTalk (Ethernet) networks.

argument A piece of information included on the command line in addition to the command; the shell passes this information to the command, which then modifies its execution in some particular way. Filenames, for example, are often supplied as arguments to commands so that a command operates on the named file.

argument list All the arguments passed to a program.

bang The exclamation point (`!`) used as a syntactic element by the C shell, by `uucp`, and by other utilities.

Bourne shell The standard UNIX System V command interpreter. See also **shell.**

BSD See **Berkeley Software Distribution.**

Berkeley Software Distribution (BSD) A version of the UNIX operating system developed at the University of California at Berkeley. The A/UX operating system incorporates many of the features of BSD versions 4.2 and 4.3.

case sensitive Able to distinguish between uppercase characters and lowercase characters. Programming languages are case sensitive when they require all uppercase letters, all lowercase letters, or proper use of uppercase and lowercase letters. For example, AppleSoft BASIC recognizes only uppercase letters. Instant Pascal, however, is not case sensitive; you can use any combination of uppercase and lowercase letters.

client (1) A computer that has access to services on a network. The computers that provide services are called **servers.** A user at a client can request file access, remote login, file transfer, printing, or other available services from servers. (2) An X Window System application.

command line The entire input string that you enter in response to the shell prompt to issue a command or to start a program. The command line includes the command itself and any **arguments** and options.

command mode The operating state in which a program (such as a text editor) interprets the characters you type as commands, rather than as data to be entered into a file.

control character A nonprinting character that orders an action to be performed. For example, the **interrupt character** (by default, entered by holding down the CONTROL key and pressing C) interrupts program execution and returns you to the shell prompt.

C shell The standard BSD and A/UX command interpreter. See also **shell.**

current directory The directory in which you are currently working. This directory is the starting reference point for all relative pathnames you enter. Also called the *working directory.*

daemon A process that, when started, runs continuously in the background; daemon processes typically provide a service, the need for which is unpredictable and intermittent.

demon See **daemon.**

device A part of the computer, or a piece of external equipment, that can transfer information.

directory A file that contains a list of other files; similar to a folder in the Macintosh hierarchical file system.

environment A list of characteristics that identifies you to the system and influences and constrains your access to it. You can modify many of these characteristics.

escape character A character that causes a program to interpret the following character or characters in a special way. For the `nroff` and `troff` utilities, for example, the escape character is a backslash (\ ), a nonprinting character that allows you to insert a command in a line of text.

Ethernet A hardware medium designed to provide high-speed interconnections between computers. The original Ethernet cable was a 1/2-inch coaxial cable with heavy shielding. Ethernet cable is now available in thinner coaxial cable and shielded twisted-pair cable.

EtherTalk Apple's data-link product that allows an **AppleTalk** network to be connected by **Ethernet** cables.

file mode See **permissions.**

filename On UNIX System V operating systems, the name of a file, consisting of up to 14 characters and specified without listing the directory under which the file is located. For example, `passwd` is the filename for the system password file. See also **pathname.**

file server A computer with special software that allows many network users to store and retrieve files on the hard disks or other storage devices attached to it.

file sharing The capability of sharing files among computers on a network.

file system A logical device that contains the data structures that implement all or part of the directory hierarchy.

filter A utility that transforms its input in some way and writes this transformed data to the standard output. Lines submitted as input to the `sort` command, for example, are reordered so that the lines in the output are arranged alphabetically or numerically.

folder (1) A holder of documents and applications on the Macintosh desktop. Macintosh folders, like UNIX file system directories, allow you to organize information in a hierarchical fashion. (2) For the BSD `mailx` program, a file that you create for saving similar mail messages.

group A convenient way for a person who is sharing files to control access privileges for a number of users at once. You can assign registered users to groups on your Macintosh and then share a folder or disk with that group.

guest A person who connects to another computer on a network without having to give a name or password.

guest access A type of access to a computer that permits any network user to connect to the computer for the purpose of file sharing. The owner of the computer can control whether guest access is allowed on that computer.

interactive program A program that allows you to enter additional commands and data during its execution instead of making you enter all your commands and data as options and arguments on the command line. The `vi` and `mail` utilities are examples of interactive programs.

internet (1) A group of networks interconnected by bridges or gateways. (2) The Internet, used as a proper noun, usually refers to the TCP/IP-based Defense Data Network (DDN), descendent of the DARPA (Defense Advanced Research Projects Agency) Internet (also called the ARPANET). (3) When the proper noun is used as an adjective (for example, Internet domain), this refers to a networking standard used by the DDN.

Internet address An address for a computer on a network. The internet address consists of a network number and a host number that is unique for that network.

interrupt character The keyboard character that, when pressed, interrupts execution of a program and returns you to the shell prompt. By default, CONTROL-C is the A/UX interrupt character (issued by holding down the CONTROL key while pressing C).

Korn shell A command interpreter that combines many of the best features found in the standard System V shell (the Bourne shell) and the standard BSD shell (the C shell). See also **shell.**

LAN See **local area network.**

line editor A utility for entering and manipulating text. The commands to add or change text are entered from a command prompt, they only operate on the lines you specify, and you cannot always see the results of your changes right away. The `ed` and `ex` utilities are line editors.

list To display on a monitor, or print on a printer, the contents of memory or of a file.

local area network (LAN) A group of computers connected for the purpose of sharing resources. The computers on a LAN are typically joined by a single transmission cable and are located within a small area such as a single building or section of a building.

local system The computer from which a user originates a network command. See also **remote system.**

LocalTalk Apple's low-cost connectivity product consisting of cables, connector modules, and other equipment for connecting computers and other devices to an **AppleTalk** network.

login name The name of a user's account. Used for identification purposes.

MacX An X Window System server that runs under both the Macintosh and A/UX operating systems.

member A registered user who belongs to a group.

message list An argument that allows you to specify a group of mail messages by number or name to various `mail` commands.

message name The login name of a user who sent a message.

message number Either the number of the mail message or one of the following special characters: `^` (first message); `.` (current message); `$` (last message).

message range Two message numbers separated by a hyphen.

messages A group of system calls that allow processes to communicate by sending formatted data streams to each other.

metacharacter A character interpreted by a program as standing for other characters or as designating a special function. For example, the ampersand (`&`) metacharacter at the end of a command line causes the shell to run the command as a background job.

mode See **permissions.**

mount To install a file system onto the directory hierarchy.

netmask A 32-bit value. The bits in the netmask that are set to 1 define the portion of the **Internet address** that is the network number, and the bits that are set to 0 define the portion of the Internet address that is the host number.

network A collection of interconnected, individually controlled computers, along with the hardware and software used to connect them. A network allows users to share data and peripheral devices and to exchange electronic mail.

network administrator A person responsible for setting up, maintaining, or troubleshooting your network.

Network File System (NFS) A protocol suite developed and licensed by Sun Microsystems that allows different makes of computers running different operating systems to easily share files and disk storage.

NFS See **Network File System.**

NFS client A user who gains access to (mounts) files made available by an NFS server through the **Network File System.** A client can be an **NFS server** at the same time.

NFS server A computer that makes its files available to UNIX users through the **Network File System.** A server can be an **NFS client** at the same time.

NIS See **Network Information Services.**

Network Information Services (NIS) (Formerly Yellow Pages) A network database facility for sharing a common database of user information across a local area network.

operating system Low-level software that controls a computer by performing such basic tasks as I/O, memory management, and interrupt handling.

pathname A filename preceded by its containing directory. A pathname can contain a list of directories, separated from the filename and from each other by slashes. Each item in a pathname is located in the directory named to its left. For example, `/etc/passwd` is a pathname for the system password file, `passwd`, located in the `etc` directory beneath the root (`/`) directory. See also **absolute pathname.**

peripheral device A piece of hardware, such as a disk drive, modem, printer, or terminal, that is connected to a computer and used for reading or writing data.

permissions Authorization to read, write, or execute a file or directory. Under UNIX operating systems, each capability is assigned on an individual, group, and system-wide basis. Also called the *file mode.*

pipe (noun) (1) A command line that connects two or more commands in a series so that the output of one command becomes the input to the next. (2) An intermediate file in which data is passed from one process to another. (verb) To connect two or more commands in a series so that the output of one command becomes the input to the next.

pipeline A **command line** that contains one or more **pipes.**

port (noun) (1) A socket on the back panel of a computer where you plug in a cable for connection to a network or a peripheral device. (2) A connection between the central processing unit and main memory or a device (such as a terminal) for transferring data. (verb) To move software from one computer environment to another.

print spooler A utility that writes a representation of a printed image of a document to disk or to memory, schedules it to print in a queue of other jobs, and then prints it.

process An instance of a program in execution. Usually one copy of a program is stored on a UNIX system like A/UX, but multiple instances of the program—each having its own address space—can be executed simultaneously as separate processes.

process ID (PID) A unique number assigned to each process being executed on the system. The PID is listed with its associated command when you enter the `ps` command. The PID is sometimes called the *process number.*

program A file containing coded instructions to the computer. A compiled program is a file created first in source code, then transformed by the compiler into object code. A **shell script** is a program that does not need to be compiled because it is interpreted by the shell.

prompt A character or string of characters displayed on the terminal when a program is waiting for input from you. The Bourne and Korn shells, for example, are set by default to display the dollar sign (`$`) as their prompt; the C shell is set by default to display the percentage sign (`%`) as its prompt.

protocol A set of defined communications rules.

quoting mechanism Special syntax in the command line that tells the shell to interpret metacharacters literally or to control the type of substitution allowed in the command.

redirection A feature of the shell that allows you to pass the output of a command to a file or device instead of to the terminal screen and to supply a command with input from a file or device instead of from the keyboard.

registered group A group of registered users who are listed on a computer on a network.

registered user A network user whose name and password are listed on a computer on the network. Usually, registered users have greater access privileges to shared folders and disks than do guest users.

remote system On a network, any computer other than the **local system.**

script A file containing commands. See also **shell script.**

server A computer that provides a particular service across a network. The service can be file access, login access, file transfer, printing, and so on. Computers from which users initiate the service are called **clients.**

shared disk A hard disk, CD-ROM disc, or other medium whose contents can be retrieved over the network. A disk can be shared by a file server or a computer that has file sharing turned on.

shell A utility that accepts your commands, interprets them, and passes them on to the appropriate programs for execution. A/UX provides three shells: Bourne, C, and Korn. Each can be used as an interpreted programming language. Through **shell variables** and environment variables, you can tailor the environment of your shell for your own needs.

shell prompt A character or string of characters displayed on the terminal to show that the shell is waiting for input from the user. The Bourne and Korn shells, for example, are set by default to display the dollar sign (`$`) as their prompt; the C shell is set by default to display the percentage sign (`%`) as its prompt.

shell script A shell program contained in a text file. Entering the name of the shell script from the command line executes the commands listed in the shell script.

shell variable A variable local to the shell. A shell variable is available only to the current invocation of the shell, not to any of its subshells or spawned processes.

spooler See **print spooler.**

standard input The data stream used for input to a command. By default, the shell accepts as input the characters you type from your keyboard. The left angle bracket (<) directs the shell to accept input from a file or device.

standard output The data stream used for output from a command. By default, the shell directs this to the terminal screen. The right angle bracket (>) directs the shell to write the output to a file or device.

stream editor A utility for manipulating text. Rather than allowing you to move back and forth within a file interactively, a stream editor processes the text in a single pass. The `sed` utility, for example, is a stream editor.

string option A setting specified by a set of characters.

TCP/IP See **Transmission Control Protocol/Internet Protocol.**

tilde escape The tilde character (~) used as an escape character to signal that the next input string is a command.

toggle option See **toggle variable.**

toggle variable A setting for the shell environment that can be turned on or off with the `set` or `unset` command. For example, the `set noclobber` command entered from the C shell turns on a toggle variable that helps ensure that existing files are not accidentally overwritten.

Transmission Control Protocol/Internet Protocol (TCP/IP) A suite of networking protocols developed initially for the U.S. Department of Defense that specifies the details of how computers communicate.

User/Group A category of users for whom you can set permissions to shared folders and disks. It can be any user or group that is registered on the computer containing the shared folder or disk.

UNIX operating system A general-purpose time-sharing system and related set of utilities, originally developed at AT&T Bell Laboratories. A/UX is an enhanced version of the UNIX operating system for certain Macintosh computers.

user ID A number that identifies a user at the time of login. Often called `uid`.

user name See **login name.**

virtual terminal A remote terminal that you have logged in to and can then use as if the data were on your computer.

WAN See **wide area network.**

wide area network (WAN) A system of interconnected local area networks that spans a wide geographical area.

Yellow Pages See **Network Information Services.**

zone Zones are set up by a network administrator and are used to group the devices on the network for easier access.

Index

* (asterisk) character 8-9, B-4, B-12
* (asterisk) command 8-10
\ (backslash) C-2, C-3
^ (caret) character B-11
^ (caret) command 4-13
$ (dollar sign) character B-11
$ (dollar sign) command D-3
! (exclamation point) command D-3
!date command B-8
< (left angle bracket) character B-17
. (period) character B-11
./ (period slash) character B-17
| (pipe) character C-2, C-3, C-6, C-7
+ (plus sign) B-17
? (question mark) command 5-17, D-9
> (right angle bracket) character B-14
/ (slash) character B-12, B-17
[ ] (square brackets) 8-9
~ (tilde) command 5-14, 8-6, 8-9, B-2, B-3
~$ (tilde dollar sign) command 9-7
~ (tilde) escape B-2
~! (tilde exclamation point) command 9-6, B-8
~% (tilde percent) command 9-8, 9-9
~. (tilde period) command 9-9
~| (tilde pipe) command B-8
~? (tilde question mark) command B-2
~> (tilde right angle bracket) command 9-7
~>> command 9-7
-f option in mail B-14
-g option D-2
-i option D-2
-n option D-2
-v option D-2
.mailrc file 4-7, B-10, B-13, B-14, B-16
.netrc file D-10
.profile command A-3
.rhosts file 5-8
/etc/exports file in NFS 7-6
/etc/fstab file, modifying 7-10
/etc/HOSTNAME file 2-8, A-6
/etc/hosts.equiv file 5-7, 5-17
/etc/NETADDRS file 2-8, A-6
/etc/printcap command 3-6, 3-7
/usr/catman command 7-6
~b command B-6
~c command B-6
~f command B-14
~h command B-7
~m command B-5
~p command B-2, B-3, B-6, B-8
~r command B-4
~s command B-6
~t command B-5
~v command B-3
~w command B-4

A

abbreviated command formats D-11
accessing. *See also* file sharing
  connecting
    on start-up 6-33
    to shared disk 6-28 to 6-35
  files
    allowing guest access 6-8, 6-27
    changing password 6-36
    connecting automatically 6-32
    disconnecting from shared disk 6-35
    using an alias 6-32
  folders 6-37
  information on other computers 6-26
  shared disk 6-28
  your own computer remotely 6-38
account command D-3
alias
  creating in mail 4-3
  setting up in file sharing 6-26
  using in mail 4-6
  using to connect in file sharing 6-32
  with remsh command C-9
alias command in mail 4-3, 4-7, B-16, B-17
append command D-3
appending messages in mail B-5
Apple Ethernet, connecting with 2-5

AppleTalk
services 1-3
switching networks 2-13 to 2-17
ascii command D-14
askcc variable in mail B-15
asterisk (*) character B-4, B-12
asterisk (*) command 8-10
A/UX text files, printing 3-2

B

backslash (\) C-2, C-3
Bcc: in mail B-6, B-7
bell command D-2, D-12
blind carbon copy (Bcc:) in mail B-6, B-7
broadcast address 2-7
bye command D-3

C

cables and connectors
Ethernet 2-3
LocalTalk 2-3
canceling
mail message 4-11
printing job 3-10
tilde escape in mail B-2
carbon copy (Cc:) in mail B-6, B-7
caret (^) command 4-13
case command D-12
cat /etc/hosts command 5-12
cat command 2-8, C-2
Cc: (carbon copy) in mail B-6, B-7
cd command 5-24, D-3
cdup command D-4
changing
domain name A-6
escape character in mail B-10
host name A-6
IP address A-6
netmask A-6
password in file sharing 6-9, 6-10
subject line of mail message B-6
Chooser, using to print 3-4
client application in X Window System 10-3
client on NFS 7-3, 7-7 to 7-11
close command 5-16, 5-24, D-4
commands. *See also* ftp commands; mail commands
* 8-10
? 5-17
~! 9-6
~% 9-8, 9-9
~$ 9-7
~. 5-14, 9-9
~ 8-6
~> 9-7
~>> 9-7
.profile A-3
/etc/printcap 3-6, 3-7
/usr/catman -ro 7-6
cat /etc/hosts 5-12
cat 2-8, C-2
close 5-16, 5-24, D-4
cr D-12
crmod 5-17
df 7-9
exportfs -a 7-7
ftp 5-5, 5-22
grep 5-7, C-10
kill C-3
lpq 3-9, 3-10
lpr 3-6, 3-8, 5-5, 5-18
lprm 3-10
mail 4-4, 4-5, 4-6
newconfig 2-9, 2-11
newconfig nfs 7-5
nohup 5-14
options 5-16
ping 2-12, 2-13
pipeline C-2, C-3, C-6, C-7
quit 5-16
rcp 5-4, 5-19 to 5-22
remsh 5-7 to 5-9, 5-17 to 5-19, C-4 to C-6
rlogin 5-4, 5-10 to 5-12
running on other computers 5-17
running on remote computer 8-12
ruptime 5-6
rwho 5-7
set escape 5-16
showmount-e 7-7, 7-8
status 5-15, 5-16
telnet 5-4, 5-14 to 5-16
telnet loop 2-12
toggle D-12 to D-14
umount 7-10
uuname 8-4
uupick 8-11
uustat 8-11, 8-13
uuto 8-10 to 8-11
uux 8-12
CommandShell
in MacX 10-9
printing from 3-6 to 3-8
configuring your system for NFS 7-4 to 7-5
connecting
printers 2-3
to a network 2-6 to 2-11
to remote computers 5-10 to 5-17
to remote system 9-7
to a shared disk 6-28 to 6-35
with Apple Ethernet 2-5
connections, serial 1-3
copying directories using rcp 5-21
copying files
between systems 5-4, 5-19 to 5-24
using cu 9-8
using ftp 5-22
using rcp 5-19 to 5-22
using UUCP 8-5 to 8-10
using uuto command 8-10 to 8-11
copyright laws, in file sharing 6-7
cr command D-12
creating alias in mail 4-3
crmod command 5-17
cu
closing the connection 9-9
copying files 9-8

requirements for connecting to remote system 9-4
selecting a remote session option 9-2 to 9-4
sending local output to remote system 9-7
switching between local and remote commands 9-6
using with non-A/UX computers 9-8
customizing mail B-14 to B-16

D

`d` command 4-13
date, displaying in mail B-8
`debug` command D-12
`delete` command D-4
deleting mail messages 4-13
`df` command 7-9
`dir` command D-4
directories
copying 5-21
on remote computer 5-24
`disconnect` command D-4
disconnecting
a file-sharing user 6-21
from shared disk 6-35
display server in MacX 10-3
displaying
mail message B-2, B-9, B-11
summary of tilde escapes B-2
dollar sign ($) character B-11
dollar sign (`$`) command D-3
domain name
changing A-5
NIS 2-3

E

editing
mail message B-3
mail message fields B-7
EDITOR variable B-3
escape character
changing in mail B-10
`telnet` 5-15
`escape` command 5-16
escaping mail B-9
Ethernet 2-3
cables and connectors 2-4
EtherTalk 1-3
connecting printers 3-2
switching from LocalTalk 2-13 to 2-15
switching to LocalTalk 2-16 to 2-17
with Quadra family 2-3
exclamation point (`!`) command B-8, D-3
exiting
from `cu` 9-9
mail program 4-14
`exportfs -a` command 7-7
exporting, in NFS 7-6 to 7-7

F

file sharing. *See also* accessing
accessing shared information 6-26
before you begin 6-4
changing password 6-17, 6-19, 6-36
choosing an option 1-6
connecting to a shared disk 6-23 to 6-31
copyright laws 6-7
denying access
to guest users 6-23
to specific user 6-22
to yourself 6-24
disconnecting a user 6-21
folders
giving ownership away 6-18, 6-37
making private again 6-22
monitoring activity 6-20
naming
a user 6-8 to 6-11
group of users 6-11 to 6-13
Users & Groups 6-8
naming specific people 6-7 to 6-13
overview 6-3
reducing access 6-22 to 6-26
removing
user or group 6-24
user or group membership 6-24
reviewing group membership 6-12
security measures 6-30
setting up
alias 6-32
remote access 6-15
sharing
folders and disks 6-6 to 6-7, 6-26
folders and disks with specific people 6-13
turning off 6-25
using and storing files 6-35
zones 6-29
file system
`mount` options in `/etc/fstab` 7-10
`umount` command 7-10
file transfer parameters D-14
filename expansion C-7
Finder, printing from 3-5
folders
accessing remotely 6-38
giving ownership away 6-18, 6-37
making private again 6-22
`form` command D-14
forwarding a mail message B-14
`ftp`
abbreviated command formats D-11
commands D-3 to D-9
copying files 5-22 to 5-23
ending the session 5-24
file transfer parameters D-14
options D-2
setting conditions D-2
setting up the `.netrc` file D-10
starting the session 5-22
toggle commands D-2, D-12
transferring files 5-23
`ftp` command 5-5, 5-22
`ftp` commands. *See also* commands; mail commands
`$` D-3
`!` D-3

ftp commands *(continued)*
? D-9
append D-3
ascii D-14
bell D-12
binary D-14
bye D-3
case D-12
cd 5-24, D-3
cdup D-4
close D-4
cr D-12
debug D-12
delete D-4
dir D-4
disconnect D-4
form D-14
ftp 5-5, 5-22
get 5-23, D-4
glob D-12
hash D-13
help D-4
lcd 5-24, D-4
ls D-5
macdef D-5
mdelete D-5
mdir D-5
mget D-5
mkdir D-6
mls D-6
mode D-14
mput D-6
nmap D-6
ntrans D-7
open D-7
prompt D-13
proxy D-8
put 5-23, D-8
pwd D-8
quit 5-24, D-8
quote D-8
recv D-9
remotehelp D-9
rename D-9
reset D-9
rmdir D-9
runique D-13
send D-9
sendport D-13
status D-9
struct D-15
sunique D-14
tenex D-15
trace D-14
type D-15
user D-9
verbose D-14

G

get command 5-23, D-4
getty process, disabling 2-13
glob command D-12
grep command 5-7, C-10
guest access, in file sharing 6-28, 6-27

H

h command 4-13
hash command D-13
help command D-4
history substitution C-9
host name
changing A-5
requirements 2-6 to 2-7

I

Internet address. *See* IP address
IP address 2-7
changing A-6

J

job control 8-13
jobs
requeueing 8-14
terminating with uustat 8-13

K

kill command C-3

L

LAN. *See* local area network
lcd command 5-24, D-4
left-angle bracket (<) character B-17
local area network 5-2
local client application in MacX 10-3
local shell topics C-3 to C-10
local system 2-6
LocalTalk 2-3
cables and connectors 2-4
connecting printers 3-2, 3-3
description of 1-3
switching from EtherTalk 2-13 to 2-15
switching to EtherTalk 2-16 to 2-17
logging in
to remote computer 5-10
using telnet 5-14
logging out, using telnet 5-16
lpq command 3-9, 3-10
lpr command 3-6, 3-8, 5-5, 5-18
lprm command 3-10
ls command D-5

M

macdef command D-5
MacX
client application, defined 10-3
closing a settings document 10-21 to 10-22
disconnecting a client 10-18
display server, defined 10-3
ending a session 10-19 to 10-22
giving client applications access to your computer 10-9
local client application, defined 10-3
MacX settings document, defined 10-3
opening applications
client 10-8 to 10-17
local client 10-11

local client remotely 10-11
xcalc 10-10, 10-11
overview 10-4
quitting the application 10-19
remote client application, defined 10-3
starting a session 10-7
terminology 10-3
using CommandShell 10-15
using xterm window 10-16
MacX settings document in X Window System 10-3
mail command 4-5, 4-6, B-13
mail command mode B-14
mail commands. *See also* commands and ftp commands
^ 4-13
!date B-8
~! B-8
~| B-8
~? B-2
~b B-6
~c B-6
~f B-14
~h B-7
~m B-5
~p B-2, B-3, B-6, B-8
~r B-4
~s B-6
~t B-5
~v B-3
~w B-4
alias 4-3, 4-7, B-16, B-17
d 4-13
h 4-13
mail B-13
q 4-14
reply B-12
s 4-14
set B-15 to B-16
type B-11 to B-12
u 4-13
x 4-14
mail message B-10 to B-14. *See also* mail program
/ (slash) character B-12, B-17
adding recipients to B-5
appending B-5
canceling 4-11
deleting 4-13, B-12
displaying B-9
current message B-2
several messages B-11 to B-12
editing
fields B-7
message B-3
forwarding B-14
list B-11
name B-11
number B-11
range B-11
reading
from a different file B-14
message 4-12
receiving 4-12 to 4-14
replying to B-12
saving 4-14, B-4
sending 4-3 to 4-11
and receiving 4-1
copies B-6
file with B-4
on TCP/IP network 4-8
to local system 4-4
to more than one person 4-6, 4-9
to one person 4-5
to other systems 4-8
subject line, changing B-6
type command B-11, B-12
undeleting (restoring) 4-13, B-12
mail program. *See also* mail message
.mailrc file 4-7, B-10
< (left angle bracket) character B-17
accessing the shell B-8
alias, using 4-6
appending a file B-4
askcc variable B-15
Bcc: (blind carbon copy) B-6, B-7
Cc: (carbon copy) B-6, B-7
command mode, changing to B-2
creating an alias 4-3
customizing B-14 to B-16
displaying summary of tilde escapes B-2
escape character, changing B-10
escaping
to command mode B-9
to the shell B-8
exiting 4-14
mailing from within mail B-13
sending files B-17
setting EDITOR variable B-3
setting up permanent environment B-16
tilde escape (~)
canceling B-3
displaying a summary B-2
using B-2 to B-10
tilde, using as a character B-9
toggle variables B-15
UUCP 4-10, 4-11
mailx 4-1
man pages
in NFS 7-7
reading 7-9
mdelete command D-5
mdir command D-5
message. *See* mail message
message list in mail B-11
message name in mail B-11
message number in mail B-11
message range in mail B-11
metacharacters C-4, C-7
mget command D-5
mkdir command D-6
mls command D-6
mode command D-14
monitoring file-sharing activity 6-20
mounting files 7-7 to 7-9
mput command D-6
multiple commands C-6

N

naming conventions in UUCP 8-6
netmask 2-7
  changing A-6
network
  administration 2-6 to 2-8
  checking status for UUCP 8-4 to 8-5
  connecting printer to 3-2 to 3-3
  connecting to 2-6 to 2-11
  features of A/UX system 1-4 to 1-5
  identifying 2-4 to 2-6
  LocalTalk 2-3
  overview of printing 3-6 to 3-10
  overview of TCP/IP 5-2 to 5-5
  permissions 5-7 to 5-9
  requirements for connecting 2-6
  software setup 2-6 to 2-8
  switching 2-13 to 2-17
  TCP/IP 2-3
  TCP/IP commands 5-4 to 5-5
  testing the connection 2-12
  types you can connect to 2-3
Network control panel 2-14, 2-15
Network File System. *See* NFS
Network Information Service. *See* NIS
`newconfig`
  correcting errors A-6
  warning about setting up computer 2-9
`newconfig` command 2-9, 2-11
`newconfig nfs` command 7-5
NFS (Network File System) 2-3
  `/etc/exports` file 7-6
  being a client 7-7 to 7-11
  being a server 7-6 to 7-7
  checking permissions 7-8
  client 7-3
  configuring your system for 7-5
  mount, verifying 7-8
  mounting files 7-3
  requirements for running 7-4
  server 7-3
  testing 7-7
  when to use 7-4
NIS 2-3
  client in NFS 7-5
  domain name 2-3
`nmap` command D-6
`nohup` command 5-14
`ntrans` command D-7

O

OAS 8-14
`open` command D-7
`options` command 5-16
output redirection C-6, C-8

P

password, changing in file sharing 6-17, 6-19, 6-36
period (.) character B-11
period slash (./) character B-17
permanent mail environment, setting up B-16
permissions
  checking in NFS 7-8
  network 5-7 to 5-9
`ping` command 2-12, 2-13
pipe (|) character C-2, C-6
`pipeline` command C-2, C-3, C-6, C-7
plus sign (+) B-17
print spooler 3-6
printers
  choosing with the Chooser 3-4
  connecting to network 3-2
  direct connection to remote computer 3-2
  EtherTalk connection 3-2
  LocalTalk connection 3-2
  selecting on a network 3-6
  setting up access on TCP/IP network 3-6
printing
  A/UX text files 3-6
  canceling 3-10
  CommandShell window contents 3-8
  destination 3-3, 3-6
  from the Finder 3-5
  overview 3-2 to 3-4, 3-11
  to a remote UNIX printer 3-6
  using printing queue 3-9
  with `lpr` 3-6 to 3-10
printing queue 3-9, 3-10
`prompt` command D-13
`proxy` command D-8
`put` command 5-23, 9-8, D-8
`pwd` command D-8

Q

`q` command 4-14
Quadra family 2-3
question mark (?) command 5-17, B-2, D-9
`quit` command 5-16, 5-24, D-8
quotation marks C-4
  with `ftp` commands D-3
`quote` command D-8
quoting A-4
quoting mechanisms C-4 to C-5

R

`rcp`
  copying directories 5-21
  copying files 5-19 to 5-22
  ending the session 5-22
  messages A-5
`rcp` command 5-4, 5-19 to 5-22
reading mail 4-12
receiving mail 4-12 to 4-14
`recv` command D-9
redirection symbol C-4
remote client application in MacX 10-3
remote commands, tracking 8-12
remote computers, using 5-10 to 5-17
remote devices C-2 to C-3
remote environment, customizing 5-11
`remotehelp` command D-9
Remote Host line 3-6

remote session option, selecting with cu 9-2 to 9-4
remote shell topics C-3, C-11
remote system
  defined 2-6
  using 5-12
removing user or group in file sharing 6-24
remsh
  alias in C-9
  command line considerations C-6 to C-10
  commands C-3 to C-9
  ending the session 5-19, C-3
  kill command C-3
  messages A-4
  options C-12
  using 5-17 to 5-19
remsh command 5-5, 5-17
rename command D-9
reply command in mail B-12
reset command D-9
right angle bracket (>) character B-19
rlogin
  ending the session 5-14
  messages A-2
  using 5-10 to 5-14
rlogin command 5-4, 5-10 to 5-12
rmdir command D-9
runique command D-13
ruptime command 5-6
rwho command 5-7

S

s command 4-14
saving
  current message in mail B-4
  mail messages 4-14
security
  .netrc file D-10
  in file sharing 6-30
send command D-9
sending
  copies of mail messages B-6
  files through mail B-17
  mail 4-3 to 4-11
  mail to files B-17
sendmail 4-1
sendport command D-13
serial connections 1-3
server on NFS 7-4, 7-6 to 7-7
set command B-15 to B-16
set escape command 5-16
shared disk
  accessing 6-26
  disconnecting from 6-35
  using and storing files 6-35
sharing files. *See* file sharing
shell
  accessing from within mail B-8
  escaping to in mail B-8
  scripts C-11
  topics C-3
  variables C-10
shorthand notations 5-21
showmount -e command 7-7, 7-8
slash (/) character B-12, B-17
square brackets ([]) 8-9
status
  checking with ruptime 5-6
  checking with rwho 5-7
  checking with uustat 8-4
status command 5-15, 5-16, D-9
struct command D-15
subject line in mail B-6
sunique command D-14
switching networks 2-13 to 2-17

T

TCP/IP network 2-3
  commands 5-4 to 5-5
  connecting to 2-6 to 2-11
  overview 5-2 to 5-5
  printing from 3-6, 3-11
  sending mail on 4-8
  services 1-3
  setting up your computer to use 2-9
telnet
  ending the session 5-16
  escape character 5-16
  messages A-3
  starting the session 5-15
  using 5-14 to 5-17
telnet command 5-4, 5-14 to 5-16
telnet commands 5-14 to 5-16
telnet loop command 2-12
tenex command D-15
testing
  network 2-12
  NFS file mount 7-7
text files, printing 3-6
tilde (~)
  using as a character in mail B-9
  using to escape to mail command mode B-9
tilde (~) command 5-14, 8-6
tilde dollar sign (~$) command 9-7
tilde escape
  canceling B-3
  displaying a summary B-2
  using B-2 to B-10
tilde exclamation point (~!) command 9-6, B-8
tilde percent (~%) command 9-8, 9-9
tilde period (~.) command 9-9
tilde pipe (~|) command B-8
tilde right angle bracket (~>) command 9-7
toggle
  commands D-12 to D-15
  variables in mail B-15
trace command D-14
transferring files
  between systems 5-5
  with ftp 5-23
troubleshooting
  correcting newconfig errors A-6
  rcp messages A-5
  remsh messages A-4
  rlogin messages A-2
  telnet messages A-3

type command D-15
in mail B-11, B-12

U

u command 4-13
umount command 7-10
undeleting (restoring) a mail message 4-13, B-12
USENET 8-14
user command D-9
Users & Groups in file sharing 6-8
UUCP
checking configuration 8-4
checking network status 8-4
copying files 8-5 to 8-10
multiple files 8-9
to another system 8-7
identifying remote files 8-5 to 8-6
in mail 4-10
introduction to 8-2
naming conventions 8-6
other utilities that use UUCP 8-14
switching 8-8
using an intermediate system 8-7
using uuto to copy files 8-10 to 8-11
what you need to run 8-4
uuname command 8-4
uupick command 8-11
uustat command 8-11, 8-13
uuto command 8-10 to 8-11
uux command 8-12

V

variables in mail B-15
verbose command D-14
virtual terminal 5-10

W

WAN. *See* wide area network
wide area network 5-2

X

xcalc, opening in MacX 10-10 to 10-11
x command 4-14
x-term window 10-16 to 10-17

Y

Yellow Pages. *See* Network Information Service

Z

zones, finding in file sharing 6-28 to 6-29