HAYDEN *Macintosh Library* BOOKS

48405

IBM® PC and Macintosh® Networking

Featuring TOPS™ and AppleShare™

Stephen L. Michel

IBM® PC and Macintosh® Networking

IBM® PC and Macintosh® Networking

Stephen L. Michel

HAYDEN BOOKS
A Division of Howard W. Sams & Company
4300 West 62nd Street
Indianapolis, Indiana 46268 USA

To Margaret and my family

FIRST EDITION
FIRST PRINTING—1988

International Standard Book Number: 0-672-48405-6
Library of Congress Catalog Card Number: 87-63172

Acquisitions Editor: Greg Michael
Manuscript Editor: Sara Bernhardt Black
Cover Art: Celeste Design
Indexer: Northwind Editorial Services
Technical Reviewers: Mike Rogers—TOPS; Cliff Guren, Robert Lenoil, and Michael Quinn—Apple Computer, Inc.
Compositor: Shepard Poorman Communications

Printed in the United States of America

Contents

Preface

Who Should Read This Book?

Those Who Have Both PCs and Macintoshes and Want to Network Them

This book discusses two software-hardware product combinations that allow the PC and the Macintosh to be connected in a network and to share files, storage, and printers over that network. These two products—TOPS from TOPS, in Berkeley, California, and AppleShare and AppleShare PC from Apple Computer, Inc.—provide two different approaches to the situation. If you are in one set of circumstances, you might choose one product; if you have another configuration, you might choose the other. This book can help you decide between the two solutions.

Those Who Have One or the Other of These Products

Despite the constraints of having to describe two very different solutions to the networking situation, I have tried to include a fair amount of useful information for users of either. Much of the material herein—such as chapter 4, dealing with "coexistence" or file transfer between the two environments, and chapter 5, dealing with managing the network—apply as much as possible to either situation.

Those Who Need to Transfer Files

Even if you are using only a serial connection to transfer information (and a product such as DataViz' MacLink to translate the files), much information is in here about the file formats used by various programs on each type of computer. Sometimes, I daresay, I go into more detail than is absolutely necessary about these file formats. But if you must choose among several ways to transfer, say, a database file from one machine to the other, this book should help you decide whether to use DIF, SYLK, or comma-delimited ASCII files to do it.

Those Who Are Curious

Clearly, we are going to be living in a computer world that has room for both these machines. How—or even—do you choose between them? What is the true nature of their differences? Besides just telling you how these machines are different, I have tried to explain why they are different and why it is necessary that so many of their internal mechanisms are different.

It's not possible to provide a book that is all things to all people; networking is a subject worthy of an encyclopedia. Such things as work habits and the structure of work groups are beyond the scope of this book. I have tried, though, to provide information that will benefit most of the people who seek to integrate these two major computer systems.

A Few Words About Terms

Much confusion about terminology exists in the computer world, especially in cases such as this book where the same terms can have many different meanings, depending on the context. I have included as complete a glossary as possible, given the changing nature of the industry. The glossary often features several definitions for terms—whether they are used in the PC context, the Macintosh context, or the TOPS context (or sometimes all three at once). When in doubt about a term, check the glossary and the context. Between the two, you should be able to figure it out. A short discussion of two common terms follows.

PC

When I say "PC" I don't mean to say only IBM's PC, to the exclusion of all the clones, the IBM XT, and the AT. Rather, I'm referring to the range of computers that are built around the IBM standard and that run some version of Microsoft's DOS. Insofar as they are able to use the hardware and software discussed in the book, PC also refers to the IBM PS/2 line of computers. PC, then, is a generic term (but not too generic—the Macintosh and Apple II are also personal computers).

DOS

Again, several versions of DOS are available. When IBM sells DOS, they call it PC DOS. Other companies call it MS DOS. In this book, I call it plain DOS. Because most, if not all, networking products require DOS 3.1 or later versions, take as a given that I am referring to one of these versions.

How Is This Book Organized?

Networking is a complex subject. Networking two such disparate computers as the Macintosh and the PC adds to the complexity. This book is divided into two parts, dealing with the two sides of networking the PC and the Macintosh. The first part deals with the two main solutions available today for hooking up the two machines: TOPS and AppleShare. Other solutions are available, but these products provide the most flexibility and are the two products likely to be most commonly used.

The second part of the book deals with more complex issues of file transfer across the network, such things as moving MultiMate files from the PC to the Macintosh and preserving the formatting of the original document. Beyond word processing, we discuss moving database, spreadsheet, and even graphics files between the two machines. This section should be useful even to those who are not using a network; there is nothing in any of the techniques discussed that is specific to any one networking system. You can use them successfully to translate files no matter what mechanism you are using to transfer the files from one computer's disk to another. Alternatives to a network include modem transfer and direct serial connection as well as using a PC-compatible disk drive on a Macintosh.

What is not discussed in any detail in this book is the physical set-up of the network. Many different strategies are available for cabling a network, and the strategy you use depends on a number of variables. You might cable the network one way for TOPS and a different way for AppleShare. Different cabling products—such as Apple's cables or Farallon Computing's PhoneNet—also give you different options for network topology. And the number of machines you wish to connect also requires that you set up the network or networks in different ways. There's enough variation in this topic for a book in itself; we can hope that such books are forthcoming.

What Should Be Done About Software Licensing?

When running on a network, it is very important to consider the legal, or licensing, consequences of using software. If you have just one computer, questions don't arise: You buy one copy of the program, and you use it on that computer. Only one computer, and only one user, are using that program at a time.

However, in a network situation, that changes. If you purchase just one copy of Microsoft Word 3.0, say, or WordPerfect, that program can be used by several people simultaneously on several computers.

Here is how the license agreement for Microsoft Word 3.0 on the Macintosh spells out networking:

> 1. GRANT OF LICENSE. In consideration of payment of the LICENSE fee, which is a part of the price you paid for this product, Microsoft, as Licensor, grants to you, the LICENSEE, a nonexclusive right to use and display this copy of a Microsoft software program (hereinafter the "SOFTWARE" on a single COMPUTER (i.e. with a single CPU) or on a LICENSED COMPUTER NETWORK. A computer network is any combination of two or more terminals that are electronically linked and capable of sharing the use of a single software program. A LICENSED COMPUTER NETWORK is a computer network for which the LICENSEE has purchased and dedicated at least (1) Microsoft SOFTWARE manual (which can include an instruction manual(s) for the single-user version of the SOFTWARE) for each concurrent user of the SOFTWARE on the network. Each concurrent user of the SOFTWARE must have exclusive

> access to a Microsoft SOFTWARE manual during his use. Microsoft reserves all rights not expressly granted to LICENSEE.

That passage is taken verbatim from the licensing agreement on the disk package from Microsoft Word 3.0. All capitalization is from the original.

What this passage says, essentially, is that you must purchase a separate copy of Word for each concurrent user on the network. If you have one copy of Word and only one person (on one computer) is using the package at a time, that's fine. Things are different if you have one copy of the software and have several people using it at the same time (concurrently). Then, according to this agreement, you must have a separate licensing agreement and manual (and hence a separate copy of the program) for each user.

It's easy to ignore this kind of thinking. I've talked to a lot of microcomputer managers at companies who feel this is unfair. If you buy a copy of the program, you should be able to use it however you please. I even know of one very large company that has produced very nicely bound photocopied versions of the Microsoft Word manual for each of their employees using a Mac.

The situation is at something of an impasse: Customers continue to abuse software licensing agreements by using multiple copies on a network or by making copies of the programs to pass around to their friends. The issue of duplicating software (bootlegging) is pretty clear: It's wrong and nothing is going to change that. Manufacturers have started, in the last several years, to drop copy protection from their products. Consumers should respond to this by not taking advantage of it and making illicit copies.

The networking question is much less clear. Software manufacturers like Microsoft are building networking support into their software, thus encouraging us to use it. At the same time, they are insisting that we purchase separate copies of the program for each simultaneous user. This puts the user in a tough situation. Say, for example, a company has purchased three copies of Microsoft Word for use on a network. What happens when a fourth user wants into Word? Does one person have to quit? Does that user have to wait? No, typically that fourth user just starts using the package. But that puts the company into a violation of their license agreement with the software publisher.

WordPerfect Corp. makes a networking package of WordPerfect available for $695. This supports three workstations; support for additional workstations is $150 each. If you were paying $495 per copy of WordPerfect (the list price), this would make sense and save money. However, WordPerfect is easily available for far less than the list price. I've seen it for as low as $200. Especially for smaller installations, it makes more sense to buy multiple copies of the program at a heavily discounted price. The networking package, which is usually stocked only by the higher end retail outlets, is not discounted often.

The solution, I think, is some form of site licensing. Site licensing would allow a company to purchase the program once without having to worry about multiple copies of it or how many users were in the program at once. Pricing is a problem: Obviously the price should be lower than buying multiple copies of the same program, but it should be higher than just buying one. And those with larger installations should pay more than those with smaller.

Until there is a general, industry-wide standard for dealing with network situations, most locations are in some sort of violation of their licenses with software manufacturers. It is difficult not to be. And anybody so in violation is at least theoretically liable for their actions. To users, I make two suggestions:

1. Follow the license agreements. If this means you have to purchase multiple copies of the program, do it. As one with close ties to the retail end of the computer business, I find it hard to recommend that you buy from mail order houses, but you shouldn't pay the same list price for additional copies of software at your installation. Any responsible dealer will realize this and give you a package price for multiple copies.
2. Until such time as software publishers adopt licensing agreements that are easier to work with, communicate your desires to the publishers. Tell them what you want to do, that you want to abide by their license, but that they are placing onerous demands on your compliance. If there are similar products that you could use but that aren't perhaps quite as good and have laxer networking arrangements, tell the publishers. Public demand accomplishes more than books like this.

For this book, I assume that network users are in compliance with their licensing agreements. Nothing said in this book is to be taken as condoning the violation of those agreements.

Acknowledgments

Any book that is about as many things as this one could not have been written without the extensive and gracious help of many companies and individuals. All the good stuff in this book is due to the help of those listed below and others; all the mistakes are mine.

Eric Alderman and Dale Coleman provided moral assistance, advice, encouragement, and distractions. There is no way this book would have been written without them.

At TOPS, Mike Rogers, Michael Stack, Ric Cummings, and Marty Jost were helpful and encouraging.

At Apple, Scott Schnell took time from his very busy schedule to show me AppleShare PC. Martha Steffens was also giving of her time. Robert Lenoil, Clifford Guren, and Michael Quinn read the manuscript and set me straight about a number of things, although I may still be wrong about some others.

Other software companies that helped include Adobe Systems, Ashton-Tate, Microsoft, DataViz, Think Technologies, PC Quik-Art Inc., Aldus, DesignSystems, and more. I'm sure I have forgotten some of them, and I apologize to those I have forgotten. They were left out not for any reason relating to their products, but to my own poor memory. Again, without their help, this book would not have been written. The editorial staff at Howard W. Sams & Company—including Jim Rounds and Greg Michael—have been very patient. Sara Black was a gentle and helpful copy editor.

Lyal, Richard, and Margaret Michel gave me time and reason to write this book.

Other individuals and companies who have given me moral assistance include: Bill Bowen of Whole Earth Access Computers in Berkeley, CADEX (Tom Mandel, Natasha Gordon, and Eric Webster), Arthur Naiman, Roger Strukhoff, Xavier Raya, Robert Kermish, James Kalin, John Williams, and Barbara Grieser. Again, my apologies to anybody inadvertently left out.

Trademarks

All terms mentioned in this book that are known to be trademarks or service marks are listed below. In addition, terms suspected of being trademarks or service marks have been appropriately capitalized. Howard W. Sams & Company cannot attest to the accuracy of this information. Use of a term in this book should not be regarded as affecting the validity of any trademark or service mark.

4th Dimension is a trademark of ACIUS, Inc.

AutoCAD is a registered trademark of AutoDesk, Inc.

CompuServe is a registered trademark of CompuServe Information Services, an H & R Block Company

CP/M is a registered trademark of Digital Research, Inc.

dBASE, dBASE Mac, and FrameWork are registered trademarks of Ashton-Tate

Diablo is a registered trademark of Diablo Systems, Inc.

DiskFit is a trademark of SuperMac Technologies

DiskTop is a trademark of CE Software

DiskTools is a trademark of Electronic Arts

DisplayWrite; Displaywriter; IBM PC, XT, and AT; PC DOS; and PC Network are trademarks of International Business Machines, Inc.

Double Helix and Helix are registered trademarks of Odesta Corporation

Dr. Halo is a trademark of Media Cybernetics, Inc.

Epson is a registered trademark of Epson Corporation

Excel, Microsoft Windows, and Word are trademarks and Microsoft, MS-DOS, MultiPlan, and QuickBasic are registered trademarks of Microsoft Corporation

FileMaker Plus is a registered trademark of Nashoba Systems, Inc.

Hewlett-Packard is a registered trademark of Hewlett-Packard Co.

HyperCard, ImageWriter, MacDraw, and Macintosh are trademarks and Apple, AppleShare, AppleTalk, LaserWrite, LaserWriter Plus, Mac Paint, and MacWrite are registered trademarks of Apple Computer, Inc.

InBox is a trademark of Think Technologies, Inc.

Intel is a registered trademark of Intel Corporation

MacLink and MacLink Plus are registered trademarks of DataViz, Inc.

Motorola is a registered trademark of Motorola Inc.

MultiMate is a registered trademark of MultiMate International Corporation, an Ashton-Tate Company

NetModem is a trademark of Shiva Corporation

Netware and Novell are registered trademarks of Novell, Inc.

Norton Utilities is a trademark of Peter Norton Computing

Omnis III, Omnis III Plus, and Quartz are registered trademarks of Blythe Software Limited

PageMaker is a registered trademark of Aldus Corporation

Paradox is a registered trademark of Ansa Software

PC PaintBrush is a trademark of Z-Soft Corporation

PC Quik-Art is a trademark of PC Quik-Art, Inc.

PhoneNet is a trademark of Farallon Computing

PostScript is a registered trademark of Adobe Systems

rBASE is a registered trademark of MicroRim, Inc.

Ready! and ThinkTank are trademarks of Living VideoText

ReadySetGo! is a trademark of Manhatten Graphics

Samna Word is a trademark of Samna Corporation

SideKick is a trademark and 1-2-3 and Lotus are registered trademarks of Lotus Development Corporation

Teletype is a registered trademark of AT&T Teletype Corporation

The Graphics Link is a trademark of TerraVision, Inc., licensed to PC Quick-Art, Inc.

TOPS Print is a trademark and TOPS and TOPSMENU are registered trademarks of TOPS, a Sun MicroSystem Company

Ventura Publisher is a trademark of Ventura Software, Inc.

VisiCalc is a registered trademark of VisiCorp

VP Planner is a trademark of Paperback Software, Inc.

WordPerfect is a registered trademark of WordPerfect Corporation

WordStar is a registered trademark of MicroPro International Corporation

XyWrite III is a trademark of XyQuest, Inc.

Why a Network?

Many computer books start out with anecdotes that illustrate their topics and the problems they will be solving. This book is no different. My anecdote illustrates what can happen when computers cannot network.

The Anecdote

Mary is the product manager for MakeBelieve Toy Company's line of "Fantasy Island" action adventure kit. She uses an IBM PC, working with Lotus 1-2-3 to manage the numbers—cost of manufacturing, sales forecasts, advertising budget, and the like. She also uses WordStar to create the reports, memos, and letters that she must type.

It is Tuesday afternoon. MakeBelieve Toy Company's top management has just asked Mary to give a presentation to the board of directors on Wednesday morning so that they can justify increasing the ad budget for the "Fantasy Island" line or consider dropping the line altogether. Mary knows the management team. She knows that they may drop the line if she makes a presentation that is in any way flat; it must reach out and grab them and illustrate the potential for growth of the "Fantasy Island" line. Tables of numbers will not do; she needs some sharp, eye-catching graphics. A report printed on a dot matrix printer or typewriter will not impress them. Aside from the printed reports and graphs, Mary would like to have clear overlays to use on an overhead projector, so she can better explain the trends she has found with the spreadsheet.

Mary calls Wally Beaver in MakeBelieve Toy Company's art department and explains her needs. Wally uses a Macintosh with an attached laser printer. If he had a way to bring Mary's numbers and report into his Mac, he could quickly create the graphics and high-impact report she wants.

However, Wally's Mac is not compatible with Mary's PC. What can they do? There is no time to have the numbers and report retyped into the Macintosh. In any case, the possibility for transcription errors is high. Neither Mary nor Wally have modems on their machines, so they cannot send them electronically. It is possible for Wally to drag his Mac to Mary's office and hook the two computers up directly to one another, but that is time-consuming. Wally has a hot dinner date, so he is reluctant to take that much time. Unable to get a nice presentation together, Mary has a difficult time with the board of directors the next morning.

The problem that Wally and Mary have is not uncommon. Over the past several years, while MakeBelieve Toy Company grew rapidly, they added computers in a helter-skelter manner that does not encourage their integration. Let's take a quick look at the history of computers and networking and see how we came to be here.

The History of Networks

In the late 1970s and early 1980s, the personal computer became a very popular tool for office use. In many companies, individuals and departments found they could accomplish more with small, personal computers than with the expensive, large mainframes and minis that their companies were using for such things as order processing, inventory, and order entry. Managers, using programs such as VisiCalc, could gain control of many of the numbers they were responsible for, allowing them to make projections, try alternate scenarios, and the like. Given the right supporting peripherals and software, the PC could serve a major role in data processing.

However, personal computer resources were limited, especially in the early days. One of the greatest limitations was in disk storage.

At first, personal computers were limited to floppy disk storage, and those early drives could hold only 100–200 kb of data. This capacity was totally insufficient for serious uses beyond small word-processing functions. Hard disks were very expensive: It was difficult for most users and companies to justify spending several thousand dollars for 5- or 10-Mb hard disks for just one user. Additionally, other peripherals such as printers and modems were expensive and slow. Perhaps most importantly for the evolution of computer use in an office, personal computer users found it difficult to share information with one another and access information on a company's mainframe.

The earliest local area networks were designed to make up for some of these problems. Instead of buying an expensive hard disk for each user, a central hard disk could share its data with others. The software that evolved in response to these needs was essentially limited to meeting them; that is, it relied on a centralized, dedicated file or print server. Since the prime reason for using a network was to share expensive peripherals, using networks that were more flexible in terms of peripheral sharing seemed unnecessary.

Hence came the idea of a dedicated network server—a centralized machine whose purpose was to share its resources with other linked computers. Networks such as Novell's Netware and IBM's PC Network followed this philosophy.

In some ways, the dedicated network philosophy has some advantages. With the need for a dedicated network server, it was important to have one person who was responsible for virtually all network activity—things such as deciding who should or should not have access to certain files or other resources and who is responsible for such necessary activities as backing up important data, maintaining software, or purging the disk of unnecessary files. Theoretically, at least, if one person has that responsibility, then it will always get done. (However, many networks can work in both ways: Designate one machine to be the server and don't use any of the other machines as servers.)

Early personal computer networks also shared another characteristic. These networks usually required that you divide a serving hard disk into several areas, one reserved for each network user. These areas are usually called volumes or user areas. They are available to only one network user at a time. Networks that work in this manner are called disk server networks. They make these volumes available to the network one user at a time. This makes the network easier to create and somewhat easier to maintain, but it takes away much of its functionality and flexibility. For example, with a disk serving network, two users cannot access the same file at a time. (You might want to do this if you are running a personnel or inventory database.)

The Evolution of the Personal Computer

The first personal computer to make inroads into the office environment was the Apple II. In the late 1970s, I saw Apples—usually acquired as office equipment—being used in many companies. It was rare that many of these machines were purchased and rarer still for them to be networked. Only with the advent of IBM's personal computer in 1981 did the PC really take off in offices. At that time, users began to feel the need for a local area network. The number of IBM PCs and compatibles in offices in the early 1980s mushroomed, and the need for a network grew. Users seemed to believe that a standard for personal computer use had developed and that it could not be challenged.

In 1984, Apple introduced the Macintosh. In its first incarnation, this machine, with only 128 kb of RAM and a 400 kb floppy disk, was seriously underpowered for business use. However, the Macintosh had many assets. Its new user interface was a major breakthrough for PC users. Instead of having to learn a new set of commands for every new program, all programs worked along the same lines. Learning times—for both the computer and for new programs—decreased markedly. The integrated graphics nature of the Macintosh held promise for new breeds of applications that were only dreamt of on PC-type machines. As the Macintosh grew in power—with memory expansion first to 512 kb and then to 1 Mb and the addition of fast, powerful hard disks—more and more users felt that the Macintosh offered a unique solution to some problems.

With the introduction of the LaserWriter, Apple caused two things to happen. At an introductory price of $7000, the LaserWriter bucked the trend of decreasing printer prices, making it desirable for several computers to share one printer. To facilitate this, Apple introduced AppleTalk, a low-priced network designed to allow printer sharing. Additionally, the powerful PostScript page description language resident in the LaserWriter brought new power to printing, power that users of other machines, such as the IBM PC, wanted to use.

By its very nature, the Macintosh was incompatible with the IBM standard of computing. It was incompatible in several ways.

1. It could not run the programs. The PC DOS standard machines were based on Intel's 8086 family of microprocessors, whereas the Macintosh was based on a totally incompatible line of processors manufactured by Motorola—the 68000. Thus, the large body of software written for DOS was inaccessible to Macintosh users. As the Mac grew in power, much of this software was translated for the Macintosh, or new applications performing the same functions were developed. However, some software is still inaccessible. This problem will only go away with time.
2. The disks were not the same. Instead of the 5-1/4-in. floppy drive used in most all computers before, the Macintosh used the new 3-1/2-in. disk. Thus, for those who had data stored on IBM disks, it was difficult and time-consuming to transfer their data into Macintosh format.

Even if the disks had been the same size, the operating systems were still incompatible in the way they stored files on disk. We'll examine these incompatibilities in chapter 2, but suffice it to say that they are severe enough that disk size and shape is not the whole story.

While the Macintosh was growing in power, these two problems remained and represented a stumbling block to many who wanted to access the unique virtues of the machine. Apple, along with other manufacturers, has addressed the first of these problems with the Macintosh SE and the Macintosh II. It is now possible to purchase an add-on board for these computers that allows them to run DOS-based programs and to purchase a 5-1/4-in. floppy disk for the Macintosh that can read MS DOS disks. The Macintosh SE and Macintosh II can now read 3-1/2-in. DOS disks directly. But in many ways and for many users, these are not optimal solutions and do not really address the second problem.

As we will see later, the second of these problems is more important than the first. Its solution follows a second development that was taking place at the same time that the Macintosh was growing in importance as an office computer.

As we have seen, in the early days of PCs, such peripherals as hard disks, modems, and printers were very expensive. Nevertheless, the clear trend in all things related to computers is price drop. The 10-Mb hard disk that cost $2500 in 1983 is a 20-Mb hard disk that is available for well under $1000 in 1988. Similar price drops have occurred with the computers themselves. In 1983 an IBM PC with two floppy drives, a monochrome monitor, and 256-kb RAM cost in the neighborhood of $3500—without, of course, a hard disk or printer. A similar machine is well under $1000 in 1988. The Macintosh, introduced in 1984 at $2495 in its 128-kb incarnation is now priced at $2195 for a 1-Mb model (the Plus), which includes a double-sided drive and SCSI interface (at the time of writing). These prices continue to drop.

These price decreases have caused several things to happen regarding networks. First, the number of PCs an office can afford to install has grown rapidly. It is much easier to justify putting a $1000 machine on nearly every desk than a $3000 machine. With the explosion in the number of machines comes a greater need to network those machines.

The decrease in price also caused the change in the justification of a network. Early networks were purchased to share expensive disks. Now, since disks cost much less, the reason for the network—and the demand made on its functionality—has changed. Instead of buying a network to share expensive disks, it is often installed to facilitate communication between computer users.

While computer and peripheral prices have dropped, the prices for many networking products have not dropped at the same rate. For example, IBM's network solution requires a network interface board be installed in every machine—and IBM charges in the neighborhood of $600 for each of these boards. Installing a board in a computer that costs almost as much as the computer itself is hard to justify.

Lastly, more sophisticated users demand greater functionality from their software. A company that is running a sales or inventory application, for example, will probably want to give two or more people access to that data at a time. One person might enter data into the program, while another prints a report from it. Such sophistication means that the old disk-serving networks have to evolve into file-serving networks, which make files available to users in a more flexible manner.

The expense of adding peripherals—aside from the still costly laser printers—is no longer always the primary motivation for purchasing a network. Networks are used now to facilitate the sharing of other resources—multiuser databases and desktop publishing programs for example. Networks are also used to facilitate communication between individuals in an office, through electronic mail, or to make certain files available to various individuals who perform different functions. For example, a newsletter designer or layout person must have access to documents created by the newsletter writer or to graphics from the art department. The network has come into its own as a conveyor of information.

The Introduction of Intermachine Networking

In 1986 and 1987, two new products were created to address these problems of interoperating system networking. Both products are important to those who wish to integrate the PC and Macintosh environments.

TOPS

TOPS was released in the summer of 1986. It was the first product to address directly the problems of integrating the Mac and the PC. TOPS also addressed the needs of those who want a distributed network, one that does not require that any hardware be dedicated to the file-serving function and that allows any computer on the network to work as either a client, a server, or both. Any machine on the network that has a hard disk can be made available to others on the network. And any machine on the network can access the resources that others have made available. With TOPS, any networked computer can do both simultaneously.

Second, TOPS provides a translation function that allows computers with different operating systems access to files residing on machines with other operating systems. For example, a Macintosh user can access files on an IBM PC, as long as the PC user has made those files accessible to the network. It works the other way, too. IBM users can access files on a Macintosh hard disk. Users of either machine can access shared printers or LaserWriters on the AppleTalk network.

The TOPS network is also file server based. Networked hard disks need not be divided into volumes that can be accessed one user at a time. Programs that are written in a manner consistent with network operation can let several users look at, and in some cases modify, the same file simultaneously.

AppleShare and AppleShare PC

AppleShare was introduced by Apple in January 1987. It initially provided only a solution for networking Macintoshes. AppleShare PC, which was announced in August 1987 for delivery in the first quarter of 1988, allows a PC to become a station on the AppleShare network, accessing files residing on a Macintosh AppleShare file server disk.

In many ways, AppleShare bucks the networking trends we have been discussing. AppleShare works in a dedicated file server mode: A Macintosh on the network that is acting as the file server can do nothing else while it is performing the function (aside from running corollary networking programs, such as those managing electronic mail). Only the dedicated Macintosh working as the file server can be a file server: Only it can share its disks with other computers on the network.

AppleShare also requires that many of the network functions that can be divided among individuals on a TOPS network be provided by a special network administrator on the AppleShare network. Only this special user can assign passwords, group users, and the like.

Against these potential limitations of AppleShare, it delivers enormous power and file security. When a file or a folder is created, the creator of the file has much flexibility in deciding which users (or groups of users) will have rights to see, read, or change that file. By removing some of the democratic aspects of TOPS from the network, AppleShare gives back greater security and control over what is going on within the network.

The Anecdote Revised

Let's review the marketing story about product manager Mary and designer Wally.

It is Tuesday afternoon. Mary, you will recall, needs a high-impact report, complete with graphics, tables, and text, for her important presentation to MakeBelieve Toy Company's board of directors Wednesday morning. On her IBM PC, she has all her data in 1-2-3 format and all her text in WordStar. She wants Wally in the art department to prepare high-impact reports from her data with his Macintosh. Their machines are incompatible.

With TOPS, Mary first calls Wally and asks him if he has the time to do the job. Since Wally is a Mac wizard, he thinks he probably can. He just needs the data. From his Macintosh, Wally simply publishes a folder with TOPS, to allow Mary access to it. From her PC, Mary mounts that folder, calling it "Drive D:" on her PC. Using standard DOS commands, she copies her 1-2-3 spreadsheet and the WordStar report that refers to that spreadsheet over to Wally's hard disk.

Wally uses Microsoft's Excel spreadsheet to access Mary's 1-2-3 spreadsheet. All of Mary's numbers, formulas, and annotations are intact. It doesn't take him long to create high-quality line, pie, and bar charts to show the hidden trends in her numbers—trends that are not readily apparent when the numbers are viewed as a table. Wally also uses TOPS to translate Mary's WordStar file into MacWrite's format.

To put it all together, Wally uses a desktop publishing program on the Macintosh. With that program, he integrates Mary's tables and charts from the 1-2-3 spreadsheet, as well as her text from WordStar. He prints the report using Apple's LaserWriter and makes copies for each member of the board of directors. He also uses the LaserWriter to print clear overlays of the charts and tables for the overhead projector. Mary has all the tools she needs to make a bang-up presentation to the board of directors.

The Future

We have seen the evolution of the personal computer, networking, and networking products and how these elements work together in getting our work done more easily, more quickly, and with better results. In the rest of the book, we'll see how to put all these elements together.

At some point, the subject matter of this book will be obsolete. In the best of all possible computer worlds, we won't have to worry about where we are getting our information or what kind of computer or even program created it. We will summon information into our own local computers, and the computer will not care about its source. The machines and programs themselves will handle that transparently to us. It is not hard to envision a worldwide network, with almost instantaneous electronic publishing and sharing. Read Ted Nelson's *Literary Machines* for a glimpse of how this will be implemented and a suggestion as to how it will change the world.

However, all that is to come. Today, and for at least the next couple of years, we do have to worry about these matters. Our computer systems are balkanized. They do not talk the same languages; they do not use the same file formats. Sometimes they do not even use the same alphabets. The scope of this book is not broad enough to describe connectivity between all types of computers. Instead, we are dealing with the two most popular computers in use in home and office and how these machines can come to "coexist" peacefully, if a little fitfully. That's not enough, but it is the first step on the road.

CHAPTER 1

How the Macintosh and the PC Really Differ

Before we discuss specific network products, let's take a look at the reason special networking tools are needed to connect the Macintosh and the PC. As we shall discover, the differences are more than just skin deep.

There are many differences between the Macintosh and the PC, even though the machines might appear to be similar. They both store numbers, text, and pictures on disk. The Mac uses 3-1/2-in. disk drives, has a white-on-black monitor, can display different fonts, uses a mouse, and has a standard user interface for most of the programs that run on it.

The IBM PC is hardly standardized. Some models have 5-1/4-in. drives; some have 3-1/2-in. drives. Some have monochrome text-only monitors; some have color monitors. A mouse is still a rarity in the PC world. Most importantly, the user interface—the way we deal with computers—is not standard between programs.

But the differences between the two machines are more subtle than all these superficial differences. The manner in which this information is stored on disk can be quite different.

A discussion of how the Macintosh and the PC differ in the way they store information on disk follows. We also examine how the Macintosh is able to assign icons to files, store graphics, and more. While this discussion may seem to be somewhat technical or extraneous, it is important for a full understanding of the two machines and how they can work together.

Files are located in subdirectories or folders that reside on disks or volumes. In this chapter, we'll talk about how information is stored and managed both by the Macintosh and by DOS. We'll talk about differences between how the two machines do things and about how this will affect us when we want to network the machines and use files created by one machine on the other.

At first it might seem logical to present the information by discussing the largest items (disks and volumes) first, then move into them to discuss directories and files. However, the picture becomes clearer if we approach it from the other direction. First, we'll discuss files—how they are alike on the Mac and the PC and how they differ. Next we'll discuss folders and subdirectories. By that time, disks and volumes will almost have taken care of themselves.

Macintosh and PC Files

Overview

A file is a collection of information or a document that you have created on disk. We sometimes use the words "document" and "file" interchangeably. To a point this is correct; however, programs are files, but they are not documents. Usually context will tell. A file is anything that you save by itself on disk. It might be a letter to your grandmother, a spreadsheet, a painting, or a collection of names and addresses stored by a database. Any information that is retrievable as a unit is a file.

One of the first and most obvious incompatibilities between the Macintosh and the PC is in how their files are named, structured, and maintained. The differences are obvious. Simply by looking at a file name in the PC world, you usually cannot tell what program created it or what type of information the file is likely to hold. Of course, some programs force certain extensions on the files they create, which helps somewhat. On the Macintosh, things are very different. Every file has an icon, a graphic picture of the file that you can see on the Mac DeskTop. You can usually tell what program created it by looking at the icon. And you can open the file for editing or printing, without having to worry about what program created it or in what folder or directory that program is located on the disk. In the rest of this chapter, we'll take a closer look at how the Macintosh does this. We'll also discuss the effects this has when using files over the network and transferring them back and forth between the two operating systems.

PC Files

A brief discussion of DOS file naming and the information DOS stores about each file follows. While it might be basic knowledge to advanced PC users, it does have certain ramifications in the network environment.

File Names A DOS file name has the following components:

Name The name can be from one to eight letters long. It can contain any letter of the alphabet, the numbers 0 through 9, and these special characters: # & @ ! % ( ) – { ' ~ ^ } _. It cannot contain: spaces, * ? / \ | or a period (.).

Period The period (.) is used to separate the file name from the extension. Therefore, you cannot use the period as part of a file name or extension.

Extension The extension can be from one to three characters long. It can contain any of the same characters as the file name itself. The extension is usually used to indicate what type the file is; however, there are some conventions and limitations. The extension is optional—that is, for purposes of naming a file, DOS does not require that each file have an extension. With some exceptions imposed by the applications used, you can give a file almost any extension.

Reserved file names Certain file names have a special meaning to DOS. Do not use them as file names either on the PC or on the Mac. Those file names and an explanation of each follow:

.AUX. This file name refers to an input or output as an auxiliary device—a printer or a modem. In practice, it is rarely used; however, you should still avoid using it as a file name.

.CON. This is short for console. It refers to keyboard input or displayed output. Use this when creating a batch file or text file from the keyboard or displaying a file on the screen. To create a text file with it, use it to copy a file from the console to a file.

.NUL. This file name is used when you need to direct output from a DOS program but you do not want to direct it to the screen, a file, or the printer.

.PRN. This file name refers to the printer.

For details about how each of these names is used, refer to your DOS manual.

Reserved extensions As already mentioned, there are some constraints on what extensions you can use for a file name. The DOS conventions follow:

.BAS. This file name refers to a file that contains a program written in BASIC.

.BAT. A file that has this extension is merely an ASCII text file that contains a list of DOS commands, allowing you to execute those commands simply by typing the name of the batch file at the DOS prompt. See chapter 2C for a discussion of batch files and how to create and use them in the network environment.

.COM. Short for command file, these files are usually called com files. A com file contains a program that can be executed from the keyboard. DOS imposes certain conventions in how a com file is structured.

.EXE. This file name is short for executable file. Like a com file, it identifies a program that is executed by the computer and so must have a certain structure.

Note These file names, though important to DOS, are still mainly conventions, and nothing can stop you from using these extensions as part of any file you create. Many programs, such as WordPerfect, allow you to assign any extension you like to your files. Other programs, such as 1-2-3, force certain extensions on their file names and will not open files that do not have those extensions.

Even if you are using a program that does not impose conventions for extensions, it is important that you not use any of these extensions for your files. If you type a program name from your

Book Mark

HOWARD W. SAMS & COMPANY

DEAR VALUED CUSTOMER:

Howard W. Sams & Company is dedicated to bringing you timely and authoritative books for your personal and professional library. Our goal is to provide you with excellent technical books written by the most qualified authors. You can assist us in this endeavor by checking the box next to your particular areas of interest.

We appreciate your comments and will use the information to provide you with a more comprehensive selection of titles.

Thank you,

Vice President, Book Publishing
Howard W. Sams & Company

COMPUTER TITLES:

Hardware
- ☐ Apple I40
- ☐ Macintosh I01
- ☐ Commodore I10
- ☐ IBM & Compatibles I14

Business Applications
- ☐ Word Processing J01
- ☐ DataBase J04
- ☐ Spreadsheets J02

Operating Systems
- ☐ MS-DOS K05
- ☐ OS/2 K10
- ☐ CP/M K01
- ☐ UNIX K03

Programming Languages
- ☐ C L03
- ☐ Pascal L05
- ☐ Prolog L12
- ☐ Assembly L01
- ☐ BASIC L02
- ☐ HyperTalk L14

Troubleshooting & Repair
- ☐ Computers S05
- ☐ Peripherals S10

Other
- ☐ Communications/Networking M03
- ☐ AI/Expert Systems T18

ELECTRONIC TITLES:
- ☐ Amateur Radio T01
- ☐ Audio T03
- ☐ Basic Electronics T20
- ☐ Basic Electricity T21
- ☐ Electronic Design T12
- ☐ Electronic Projects T04
- ☐ Satellites T09
- ☐ Instrumentation T05
- ☐ Digital Electronics T11

Troubleshooting & Repair
- ☐ Audio S11
- ☐ Television S04
- ☐ VCR S01
- ☐ Compact Disc S02
- ☐ Automotive S06
- ☐ Microwave Oven S03

Other interests or comments: ______________________________

____

____

Name ______________________________
Title ______________________________
Company ______________________________
Address ______________________________
City ______________________________
State/Zip ______________________________
Daytime Telephone No. ______________________________

A Division of Macmillan, Inc.
4300 West 62nd Street
Indianapolis, Indiana 46268

48405

HOWARD W. SAMS & COMPANY

Book Mark

HOWARD W. SAMS
& COMPANY

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD

FIRST CLASS PERMIT NO. 1076 INDIANAPOLIS, IND.

POSTAGE WILL BE PAID BY ADDRESSEE

HOWARD W. SAMS & CO.
ATTN: Public Relations Department
P.O. BOX 7092
Indianapolis, IN 46206

MPW and Assembly Language Programming
Scott Kronick
ISBN: 0-672-48409-9, $24.95

Introduction to Macintosh™ Pascal
Jonathon Simonoff
ISBN: 0-8104-6562-0, $19.95

The Macintosh™ Advisor (updated for MultiFinder)
Cynthia Harriman and Bencion Calica
ISBN: 0-8104-6569-8, $19.95

Object-Oriented Programming for the Macintosh™
Kurt J. Schmucker
ISBN: 0-8104-6565-5, $34.95

Programming the 68000
Edwin Rosenzweig and Harland Harrison
ISBN: 0-8104-6310-5, $24.95

The Excel™ Advanced User's Guide
Richard Loggins
ISBN: 0-672-46626-0, $19.95

dBASE® Mac Programmer's Reference Guide
Edward C. Jones
ISBN: 0-672-48416-1, $21.95

HyperTalk™ Programming
Dan Shafer
ISBN: 0-672-48426-9, $24.95

Supercharging HyperTalk™ with C and Pascal
Dan Shafer
(forthcoming)
ISBN: 0-672-48427-7, $21.95

IBM® PC and Macintosh™ Networking
Steve Michel
ISBN: 0-672-48405-6, $21.95

Macintosh™ Hard Disk Management
Bencion J. Calica and Charles Rubin
(forthcoming)
ISBN: 0-672-48403-X, $18.95

Nonlibrary Titles

C Programming Techniques for the Macintosh™
Zigurd Mednieks and Terry M. Schilke
ISBN: 0-672-22461-5, $18.95

Artificial Intelligence Programming on the Macintosh™
Dan Shafer
ISBN: 0-672-22447-X, $24.95

68000, 68010, 68020 Primer
Stan Kelly-Bootle and Bob Fowler
ISBN: 0-672-22405-4, $21.95

Place
Postage
Here

HOWARD W. SAMS & COMPANY

Dept. DM
4300 West 62nd Street
Indianapolis, IN 46268

console, DOS searches your disk for .COM, .EXE, or .BAT files that have that name. Problems will arise if you have used those extensions. For example, if you create in WordPerfect a file called "LETTER.COM" and later type in "LETTER" at the DOS prompt, DOS will attempt to execute your letter as if it were a program. Troubles can arise as DOS attempts to execute this nonexecutable file.

Aside from using the extensions provided by certain applications, your organization should develop a standard that defines what certain extensions mean. This is very important in a network situation, where various people may be using the same data. For example, affixing an extension of .LTR to all your letters makes sense. That way, when users look through your directories, they can see immediately what each file is. Of course, some programs make this difficult. Microsoft Word on the PC, for example, wants to append the extension .DOC to its files. When you go to open a file, it will list only files that have that extension. You can override this, however.

Other Information In addition to the file name, DOS also keeps track of some other information about each file. Some of this is available using the simple DOS directory or dir command; some of it is harder to examine or modify.

Date and time With the dir command (i.e., without the /W parameter on the command line), DOS stores and displays the date the file was last modified. Without special utilities, DOS provides no way for directories to be sorted or selected by this date. The DOS backup command, however, uses this date and matches it against the date the last time the disk was backed up to do an incremental backup.

File size In the directory display, DOS shows the size of the file in characters.

The following information is not available with the standard directory command. However, it is important to understand, since TOPS and AppleShare use it to translate file formats to and from the Macintosh. These items of information are called "attributes."

System file A system file contains special programming information that DOS needs to operate. DOS keeps track of whether a file is a system file. Usually there are only two files on a disk that have been formatted as a system disk (using the /S switch with the format command). These two files are IBMBIO.COM and IBMDOS.COM and are hidden files.

> ***Note*** These files have the same name even on non-IBM clones. These system files will not come up in our discussion of TOPS and networking.

Read only files A read only file is one that the operating system will not let you alter. (This is similar to the locked Macintosh file.) DOS will not let you save a file that has the same name as one that is read only. As part of its ability to display Macintosh files on the IBM, TOPS creates some read only files. If you publish a DOS directory with TOPS and make that directory read only, all the files in it will appear to be so tagged from a networked station.

> ***Note*** Some programs can bypass the DOS read only settings and modify them. Be very careful, therefore, about attempting to modify any files of whose contents you are not sure. In the chapter dealing with TOPS on the PC, we'll discuss these files.

Hidden files A hidden file is not displayed by the standard directory command and is not present in most lists of files offered by applications. TOPS uses hidden files to contain information about Mac files displayed on the PC.

Archive file DOS keeps track of whether a file has been backed up. The DOS backup command and other backup utilities change this setting for all files as they back them up. When you modify a file, DOS changes this attribute to show that it now needs to be backed up.

Your version of DOS may include a command, often called "CHMOD.EXE," that allows you to examine and alter these attributes. As mentioned, TOPS uses many of these attributes to help it keep track of Macintosh files displayed in the DOS format. *Unless you are absolutely certain of what you are doing—and the circumstances are quite dire—you should not modify any TOPS-created files using this command.* See chapter 2 for a discussion of these TOPS-created files.

Macintosh Files

From the user's point of view, the Macintosh system is far simpler than the PC system. File names can be much longer and more varied. You do not have to worry about reserved names (to a point) or extensions to indicate what kind of file any one is or what program created it. However, from a perspective of managing these files and how the Macintosh keeps track of things, many more complexities are apparent. We'll discuss these in much the same manner as we did for the PC.

The DeskTop File Before we discuss Macintosh files, a few words about the Macintosh DeskTop file are in order.

Every Macintosh disk—floppy disk, hard disk, or networked disk—contains an invisible file called "DeskTop." This file is created only by the Finder, and the Finder must be able to write to the disk in order to create this file. This file is used by the Finder to keep track of a number of things. All the icons used to represent files on the screen are stored in the DeskTop file. The DeskTop file also keeps track of where in the window or on the desktop a particular item is located, and it stores information about the file that you have typed into the Get Info window.

The DeskTop file—its existence and use by the Macintosh—is a major part of what constitutes the difference between the Mac and PC operating systems (aside from the obvious differences: user interface, graphics, fonts, etc.).

File Names A Macintosh file name can be up to thirty-one characters long. It can contain any character you can type from the keyboard, with one exception. The colon (:) is used by the operating system to separate a disk name from folder names and file names. (Try this: If you are renaming a file from the Macintosh Finder, you cannot even enter the colon as part of a file name. The Finder substitutes the dash [-] for the colon as you type it.)

Long file names on the Macintosh can be very handy. On a PC, a letter you wrote to your mother and sister might have a name like MOM&SIS.LTR—it is not exactly clear from the name what the letter actually holds. On the Mac, however, you can call the file Letter to Mom & Sis or you can type even more to help identify the file. In practice, do not create files that are thirty-one characters long. As we shall see when we examine the mechanisms TOPS uses to translate

Mac file names to those that are acceptable to the PC, this can create confusion. Additionally, most programs do not provide room to display thirty-one characters in their Open or Save as dialogue boxes. If you use names that are too long, you may not be able to tell them apart when you are opening them.

File Formats DOS enforces no structure on a file. The Macintosh is quite different. In the Mac world, each file can be thought of as two files bundled together (more about "bundle" in a moment). These two files are referred to as forks. They perform quite different functions and are important to an understanding of how networking between the Mac and the PC works. Both applications and data files usually have both forks, but we'll limit this discussion to the contents of data files and not applications.

The data fork is the part of the file that contains just what its name implies—the data. The data fork contains most of the contents of a file. For example, a file created by a word processor typically stores all the text you type at the keyboard as part of the data fork. Similarly, the numbers you enter in a spreadsheet or the characters you type in a database are stored as the data fork of the file. There are exceptions, of course, but they are beside the current point. When you are transferring files to a PC, it is usually the information stored in the data fork that you want to transfer.

The resource fork of a Macintosh file contains mostly stuff that is particular to the Macintosh or to the formatting of the information in the file. In our word-processing example, the resource fork might contain information that the program uses to format your document—where underlines go, what fonts are used, and more. Many programs make little or no use of the resource fork when creating their files, and thus the resource fork can be quite small. On the other hand, many Macintosh applications are composed almost entirely of resources.

In addition to the two forks that are part of a Macintosh file, each file also contains certain header information. This leads us to file types and creators.

As part of the header information of a file, the Macintosh stores a number of items that allow it to do many things that the PC cannot do. One of the things it does is keep track of what type of file it is and what application created it.

File Type The file type is what allows the Macintosh Finder to associate and display a certain icon with a file. For example, a file created by Microsoft Word 3 on the Macintosh will have a file type of WDBN. When the Finder is creating the DeskTop file, it looks at the header part of the file, reads the file type, then assigns the appropriate icon to that file.

Applications use file type to tell the Finder what icon to display on the DeskTop. This information is used in other ways, too. For example, if you are using Microsoft Word 3 to open a file, it will only display files that it can open. At first thought, you might think that means it will open only WDBN files. However, since Word will also open MacWrite or text files, these will also show in the list.

File Creator Like the file-type information, this is a four-character code that identifies the application that created the file. For Word 3 documents, the creator is MSWD. Again, all applications have a unique code that identifies them. The creator information is what allows you to open a file by double-clicking on it from the Finder. When you do that, the Finder checks the file creator information, then uses its own internal table to locate that application on a disk, open it, and tell it to load the file you have double-clicked.

When you use the Open dialogue box from within a program, that program will usually show you only those files that have the creator type of that program (or programs that create compatible files). Thus, Word 3 will usually list only files that have the creator MSWD (for exceptions to this rule, see above). All applications on the Macintosh have the creator type of APPL.

As we'll see, when transferring files to the Mac from the PC, these items may need to be altered. We'll discuss that in the appropriate places.

Dates Like the PC, the Macintosh also stores the date and time that the file was last modified. In addition, the Mac also stores the date and time that the file was actually created. This information is available from the Get Info box that is available from the Finder. You can sort files by the date they were created by selecting "By Date" from the Finder's view menu.

Various attributes Like the PC, the Mac also keeps a set of flags or attributes about each file. You won't need to worry about many of these in networking; you cannot normally get to them anyway. Nevertheless, a brief description of each bit follows.

Bozo. On early versions of the Macintosh Finder, the bozo bit was used to indicate whether the file was copy-protected. If this bit was set, then the Finder would not copy the file. This turned out to be a fairly useless method of copy-protection, as it was easy to circumvent. It is not used on versions of the Finder greater than 5.0. This is also called "no copy" in some places.

Bundle. This bit is very important. Since each file is composed of two parts, the bundle bit is used to tell the Finder that these two are part of the same file and to keep them together when moving or erasing the file. Essentially, it bundles the resource fork together with the data fork. If this bit is off, only a generic document icon will be displayed for the file.

Busy. This bit is set when the file is open or otherwise being used.

Changed. The changed bit simply means that the file has been changed since the very first time it was created. As you can imagine, most files have this set, unless you have worked on the file only once. It has no great impact on networking.

Inited. This signifies that a file's icon has been given a specific location on the DeskTop. For example, if you create and save a file from within a program, the file won't be inited until you return to the Finder.

Invisible. The Finder won't display the icon on the DeskTop nor will most applications show it when you use the Open or Save as dialogue boxes. This is similar to the PC's hidden file attribute. This DeskTop file is normally invisible.

Locked. The Finder and applications cannot delete, rename, or replace this file. It is similar to the read only attribute in DOS. Unlike most of the other file attributes managed by the Finder, this one can easily be changed by the user: select the file, choose "Get Info" from the Finder's file menu, and click in the little box with the word "locked" next to it. If there is an X in this box, the file is locked; an empty box means that it is not locked.

Shared. With the introduction of Finder version 5.3, Apple included this flag to allow programs to determine whether a file was currently being shared or could be shared among users. A shared file is one that can be opened by many people using the same program on a network. If this bit is not set, second and further users of a file will not be able to open it or will be able to open it as a read only file.

Cached. This item is not used often, but it is used to indicate whether the file has been loaded into a RAM cache. Always switch on launch. In chapter 2, we'll discuss switch launching. Briefly, this means that when a program is launched or started, this item tells the Finder that control of the computer should first pass to the version of the System and Finder that is nearest to (i.e., on the same disk as) the application itself. Many installer programs that insert new resources into existing System files use this bit. It makes sure that the installer program is getting the correct resources from the correct disk.

Just by looking at a directory entry in the PC world, you usually cannot tell what program created a typical file or what kind of file it is. The convention, and probable intention of the creators of DOS, is to use the extension part of the file name. For example, Lotus 1-2-3 2.0 will include .WKS as the extension, despite whether you specify it. When it lists files you can open, it will not list any files that do not have that extension. Many other programs work the same way. dBASE will not open a file that does not have the .DBF extension.

However, not all programs work this way. WordPerfect, for example, allows you to provide any extension you like to a file and will at least attempt to open files with any extension. (This includes files with the .COM and .EXE extensions, which are programs.)

In the Mac world, things are simpler. From the Mac Finder, you can tell what program created a file simply by looking at its icon. For example, a MacPaint file has a little paintbrush in its icon, and a MacWrite file shows a sheet of paper with lines of text on it. How does the Mac do this without using extensions in file names? Stored as part of the information about each file, there are two types of information. The first is the file type. The file type tells the Mac operating system what type of program it is (i.e., bitmap text).

The file type consists of four uppercase characters. For example, MacPaint files have the file type of PNTG, which, of course, stands for painting. Macintosh files also have information associated with them that tells what program created the file. This is the creator field, and it is what lets you double-click on a file and open the program with that file. When you double-click on a file, the Finder looks into the DeskTop file and locates the application that created the file, launches the application, and opens the file.

When using files that are converted from one computer to another, you will frequently find it necessary to change the file types and creators manually. For example, a MacPaint-type application will not open any file that does not have the PTNG file type, so you have to set any bitmap (see chapter 4E) that is moved to the Mac from the PC. I am aware of at least three programs that let you do this, and many others are available:

DiskTop. This program is available from CE Software and is one of the handiest Mac desk accessories I've used, both for network operations and normal Macintosh use. DiskTop is a program that provides many features that are part of the Macintosh Finder. It is easy and intuitive to use. To change a file's type and creator with DiskTop, first use its Mac-like interface to locate the file you wish to change. Select the file, then choose "File Info" from the DiskTop menu.

DiskTools. This shareware program provides many of the same functions of DiskTop. Both are very handy for a number of network uses.

ResEdit. This program is Apple's standard resource editor and allows you to modify many characteristics of files and even programs. Although more powerful for modifying files than either DiskTop or DiskTools, ResEdit operates as a stand-alone program and is somewhat more inconvenient and dangerous to use than the others.

Mac files can be quite a bit different than PC files. In their internal structure, they sometimes are quite similar. In how they are dealt with by the computer they are created on, however, they are quite different. This affects how we manage them in an environment that features both machines and leads us to our next topic.

Subdirectories and Folders

As we have seen in our discussion of files, there are many differences between the PC and the Mac, specifically how information is stored, managed, and viewed on disk at the file level. The same is true at the directory level.

First, let's look at the history of hard disks on both the PC and the Mac. In the early days of the PC, a computer rarely had a hard disk attached to it. In the early days, computers were expensive, and adding a hard disk made them much more expensive. Most computers ran from floppy disks. Now, a floppy disk has a limited storage capacity (IBM's first single-sided disks held about 180 kb) and so could hold a very limited number of files. The first versions of DOS (through version 2.0) worked well with floppy disks.

In an environment where there are only a small number of files being stored, there does not need to be a complex structure for storing those files. They can be stored in a flat system. That is, for each disk, there is one directory that holds all the files. However, on a hard disk, which can hold from ten to several hundred megabytes, the number of files stored can grow to the thousands. Obviously, something better is needed to store and manage all those files. Many early manufacturers of hard disks for the PC created software that divided the disk into partitions, which appeared to the user and to DOS to be separate disks, each of which was small enough to be manageable.

When IBM introduced their own hard disk computer—the PC XT—they released a new version of DOS, which was more able to deal with hard disks. With DOS 2.0 for the PC, IBM and Microsoft introduced a new structure for holding disk information. This structure makes use of subdirectories. As its name implies, a subdirectory is a directory that is located under the main directory, which became known as the root directory. This is called a tree-structured directory system. The root directory, or top level, contains all the other directories. Other directories, which branch off that root directory, hold different types of files. For example, your word processor, and sometimes all the files it holds, is located in a directory called, maybe, "Word" below the root directory. Directories can be nested several layers deep. This is a much more efficient way of managing a disk. It makes working with the disk much faster and easier. You, and your programs, do not have to scroll through a long list of files to find the one you want to work with.

The same course of events happened in the Macintosh world. In fact, the history is so similar that many wondered then why Apple hadn't learned the lesson at the start. As with the PC, the first release of the Macintosh was designed to work in an environment with floppy disks. Since only a relatively few files can be stored on a floppy disk, a complex structure is not needed for managing those disks. However, even with the early software, the Macintosh was laying the groundwork for dealing with larger devices—the Finder's folders.

The first Macintosh System software (called, today not then, MFS for Macintosh Filing System) featured folders that could be addressed from the Finder. That is, when working at the Finder DeskTop, you could create folders that had the recognizable icon of a file folder and tuck files of different types into it. You might have a MacWrite data disk that contained different files created by MacWrite. You could put all your letters into one folder, all your reports into another, or whatever. However, there was one big drawback to the way this was handled: The folders were illusions created by the Finder. That is, at the DeskTop you would see all your files organized nicely by folder. However, when you wanted to open a file from within MacWrite, it listed all the MacWrite files on the disk, with no regard to what folders you might have put them into. MacWrite, like any other program, simply did not "see" the folders.

When the first hard disks for the Macintosh appeared, it quickly became evident what a drawback that was. Even if you could use folders to organize everything visually on the desktop, your applications would show you all the files on the disk. And the Mac operating system also ran against a built-in limitation on the number of files that a disk could hold. Many manufactures got around this situation by creating software that allowed you to partition the hard disk or divide it into sections, each of which was small enough to be easily managed by the user and by the operating system.

When Apple released their own hard disk for the Mac, and later when they released the Mac Plus, they designed a new operating system—or filing system—that was more able to deal with hard disks. This is called the hierarchical filing system (HFS). It is built into the ROM of all current Macintoshes.

Apple implemented HFS in a manner that was very natural to the development of the Mac. They simply (or not so simply) made folders real. Instead of the Finder being the only program that could see files, all programs that used standard Macintosh ROM calls to open a file would now let you move through the folders on a disk to find the file you wanted.

For Apple to do this, and still maintain much of what the Macintosh was all about, they did some things that made networking the Mac with PCs even more difficult than it would first appear.

As mentioned in the section dealing with files, the Macintosh keeps a DeskTop file on every disk it has worked with. As we have seen, this file contains information such as what icons to display and where to display them. Also, in this file, the Mac keeps a list of all applications and where they are located on the disk. (You can see this list by looking at the APP resource in the DeskTop with something like ResEdit.) Actually, it keeps a list of the file creators and what folder they are in. Thus, when you double-click on a MacPaint file, the Finder looks into this resource, finds the location of the creator MPNT, and launches that application from the folder. For the Macintosh to do this, it must, of course, keep a list of all the folders on a disk. Each folder is numbered. When you create a new folder, the Finder assigns a number to that folder. If you move an application into that folder, the Finder then must modify the DeskTop file to indicate that the application with that creator has moved and update that information in the DeskTop file. What this means is that the Finder must constantly monitor any files you move, create, delete, or whatever and must always be updated as to folder names, numbers, and the like. That's no big deal—you almost always use the Finder to move files.

However, given the way Apple implemented HFS, it has been difficult to implement this structure in the DOS world. Keep in mind the distinction between directories on a PC (even those that contain all Macintosh files) and folders on a Macintosh. When we discuss implementing TOPS on the two machines, we'll talk more about this.

Disks and Volumes

All disks are volumes, but not all volumes are disks. At any rate, not all volumes represent entire disks. The distinction is an important one. When networking, we use volumes as our means of accessing parts of the hard disks that are on other computers, so it makes sense to take a moment and examine the distinctions.

To repeat, all disks are volumes. In DOS, every volume is accessed by referring to it with a different letter of the alphabet: A: is your floppy disk; C: is your hard disk. If you have mounted a TOPS volume it is probably disk D: or E:. On the Macintosh, a volume has its own name and own icon at the right-hand side of the desktop. Your hard disk is one volume, your floppy disk is another (the Mac refers to items by name rather than by where they are, as is done on the PC.)

Your computer calls something a volume based on how it sees it, not on how the computer that owns it sees it. For example, if I make available a subdirectory called "Pictures" on the PC, that is, and will ever be, just a subdirectory on my disk. However, if I mount that directory on another machine, it is now a volume. What was a subdirectory to the computer that owns it is the root directory of a mounted volume to another computer that uses it. If I use it on a PC, it has its own separate drive designation. If I use it on a Macintosh, then it has its own disk-type icon. A volume, then, is really the top level of file organization for the machine that is presently using the file. We are more concerned with how it is perceived by the user than by the owner. This distinction will become clearer as we discuss in more detail the ins and outs of running the network between machines.

CHAPTER 2

TOPS

Part A INTRODUCTION

In the introduction, we discussed several trends in networking, namely the trend toward inexpensive, decentralized networks designed for file sharing and communications between users. TOPS is such a network. It requires no file server or network administrator. On a TOPS network, any computer can be a server, and any computer can use network resources—often at the same time.

> ***Note*** As this book was being written, the second major release of TOPS—TOPS 2—was in preparation. By the time this book is released, TOPS 2 should be in general use. Unless specified otherwise, all discussions of TOPS in this book apply to TOPS 2, as it adds major functionality to the product. If you have not already upgraded TOPS on your system to TOPS 2, I heartily recommend that you do so as soon as possible. The upgrade is inexpensive (costs had not yet been set at the time of this writing), and the added functions make it well worth the cost.

TOPS was released in the summer of 1986. It was the first product to allow easy networking of PCs and Macintoshes. Simply by plugging a special interface card into a PC, TOPS adds AppleTalk capabilities to that machine. Using the TOPS software, PCs can then make their hard disks available to Macintoshes, use information that is on Macintosh hard disks, and use Apple's LaserWriter. TOPS can also be used to connect TOPS networks of Macintoshes and PCs to other networks of PCs—even from other manufacturers, such as IBM and Novell.

The TOPS philosophy is similar to that of its parent company, Sun MicroSystems. Heterogenous networks should not just be possible between dissimilar computers, but it should also be transparent to the users of those computers. All network resources should be available to all users on the network without increasing the complexity of working with any one computer. To that end, TOPS is being extended upward to work with many minicomputer operating systems, including UNIX.

Decentralized Network Functions

One key to TOPS is the way it works as a decentralized network. Many networks (such as AppleShare, which we'll discuss in part C) require that one or more computers on the network be dedicated to the function of file serving. Only information on the hard disks connected directly to the file server may be made available to the network. The machine that is acting as the dedicated file server can perform no other functions (or only limited other functions). Though you can operate TOPS in this manner, it does not require this sort of setup.

Any computer on the TOPS network can make parts of its hard disk available to the network. Any computer on the network can use resources published by other computers on the network. This provides a large measure of flexibility in configuring and using TOPS. It also adds some responsibilities to every user on the network.

This distributed file-serving architecture gives TOPS a great deal of flexibility in setting up networks.

Networks of PCs Only

Since TOPS works as a DOS 3.1 compatible network, it can be used in a network containing only PCs as well as one that contains only

Macintoshes or a mixture of machines. Programs running on it can automatically take advantage of DOS file- and record-locking schemes. Since TOPS uses a distributed file server, it does not necessarily require that new computers, hard disks, or printers be purchased to build a useful network. Many times, existing resources can be used more effectively with TOPS, perhaps even eliminating the need to purchase new hardware. Instead of buying a new hard disk for one user with a limited need for it, perhaps that user can share, over TOPS, an existing hard disk with another user. The same applies to printers.

Networks of Mixed PCs and Macintoshes

Since TOPS was designed for, and excels at, interoperating system networking, it is adept at handling cases where there is a variable mix between Macintoshes and PCs. The network functions as well in cases with many PCs and one Macintosh as it does in cases where there are many Macintoshes and one PC. Files—and programs—used on one type of computer do not need to be stored on that type of computer; TOPS does not make any requirements as to what type of hardware is connected to what type of computer. A PC can share its hard disk with a Macintosh as easily as it can share it with other PCs (and vice versa).

TOPS can "coexist" in a PC and the Macintosh with the software and hardware that is used by many other networking solutions—both for the PC and the Macintosh. Installations with existing networks of PCs can use TOPS to bridge those networks to TOPS networks.

Networks of Macintoshes Only

TOPS works equally well in networks consisting of only Macintoshes. TOPS supports the AppleTalk Filing Protocol—a method of sharing files simultaneously among several users.

Parts B and C discuss both versions of TOPS: the Macintosh and the PC, respectively. We'll talk about putting them together in part D.

TOPS Basics

Two concepts are key to understanding and using TOPS. They are publishing and its flip-side, mounting.

On TOPS, publishing is the process of making a resource available to the network. To share a subdirectory, folder, entire hard disk, or printer with the rest of the network, you first publish that directory, which makes it available to others. Any computer that has published volumes or printers becomes a server.

Mounting is simply a matter of finding a network resource (e.g., hard disk or folder) and making it appear to be another disk connected directly to your computer. TOPS calls any computer that has mounted volumes or printers a client.

> ***Note*** Any computer can either publish volumes, mount volumes, or both. Any computer on the network can be either a client, a server, or both at the same time.

For example, User A might have a folder on her hard disk, let's call it "Public Files." She wants that folder to be available to the network, so she publishes it. On the Mac, this is as easy as choosing the TOPS Desk Accessory (DA), clicking on the folder, and clicking in the Publish button. On the PC, she can use the TOPSMENU program or work directly from the DOS command line. The result, in either case, is the same. User A is now a server.

Now, User B wants to access this information. On the Macintosh, he chooses the TOPS DA, finds this networked volume, and clicks on mount. When he closes the TOPS DA, this networked volume appears as if it were just another disk drive. On the PC, again, he would use TOPSMENU or work from the command line. If he has a hard disk on his PC, the networked volume might become drive D:, and he can access it as if it were just another hard disk on his machine User B is now a client of User A.

So, the key TOPS concepts—and we'll refer to them often—are publishing and mounting. For any disk drive (or folder or subdirectory) to be available to others on the network, it must first be published. For any user to be able to access that published volume, it must first be mounted to that user's machine.

Publishing

Here are some general tips that should help you understand publishing volumes in TOPS.

Publishing Is General When you publish a volume, you do not publish it to any specific computer or user on the network. All others on the network can see that you have published that volume. And, unless you assign a password to the volume, all others can use that volume.

Publishing Is Inherited When you publish a directory or a folder, all the directories and folders it contains are published as well. These directories inherit the state of being published from their parents. It is important to keep this in mind if you have items on your hard disk that you do not want others to see. Make sure that you keep them in directories that will never be published and that their parent directories will never be published.

> ***Note*** By putting frequently published directories together in one directory that you publish often, you can save some time. Instead of having to publish several directories in several steps, all can be published with one command. This also saves some memory. If you have many directories to publish, it keeps you from running up against the limits of the number of directories that can be published directly.

> ***Note*** Any directories or folders that are in a published directory or folder can be mounted individually. You do not have to mount them all. (There are some differences in mounting DOS directories on the PC and the Mac.)

Mounting

Before a published volume can actually be used by another user, that volume must first be mounted. (This is not strictly true in the case of the Macintosh version, but the Mac's departure from that standard is discussed in part B.) When a volume is mounted, TOPS makes that volume appear to your computer (and to you) just as if it were a disk connected directly to your machine. Virtually all the commands, programs, and the like that you use to work with your own disks work with networked disks over TOPS. On a Macintosh, a mounted volume takes on its own icon on the DeskTop and can be accessed with the Drive button in most file dialogue boxes. On the PC, a mounted volume will become drive D: or E: and can be accessed in the same manner as other disk drives.

Here are some things to keep in mind about mounting.

Mounting Is Specific Just as publishing is general—an offer to share is made to the entire network—mounting is specific. One shared volume is mounted at a time.

Mounting Is (Almost Always) Inherited When you mount a volume, all the directories or folders within that volume are mounted as well. For example, User A publishes a folder on the Macintosh called "Data," which includes another folder called "Sales." On other computers on the network, "Sales" will be accessible as well as "Data" when "Data" is mounted. The same holds true for PC directories: Subdirectories are automatically available whenever a parent directory is mounted. (There are, however, some subtleties that are exceptions to this rule when mounting volumes on a Mac that were published by a PC—see part D.)

Part B *TOPS ON THE MACINTOSH*

TOPS for the Macintosh provides a complete environment for networking between Macintoshes. When it is used along with TOPS for the PC, it provides a complete environment for interoperating system networking.

The TOPS Desk Accessory

On the Macintosh, all TOPS functions are performed through the TOPS Desk Accessory (DA) shown in figure 2.1. (Disks that are connected directly to the Macintosh are shown in the left-hand window; file servers are visible in the right-hand window.) This means that TOPS functions can be accessed by the user even while other programs are running. If you need to access a remote hard disk that you have not already mounted, you do not need to quit your current program. Simply call up the TOPS DA, mount the volume, and close the DA. The remote hard disk will be available to you.

All Macintosh TOPS functions are performed through this Desk Accessory. Its elements follow.

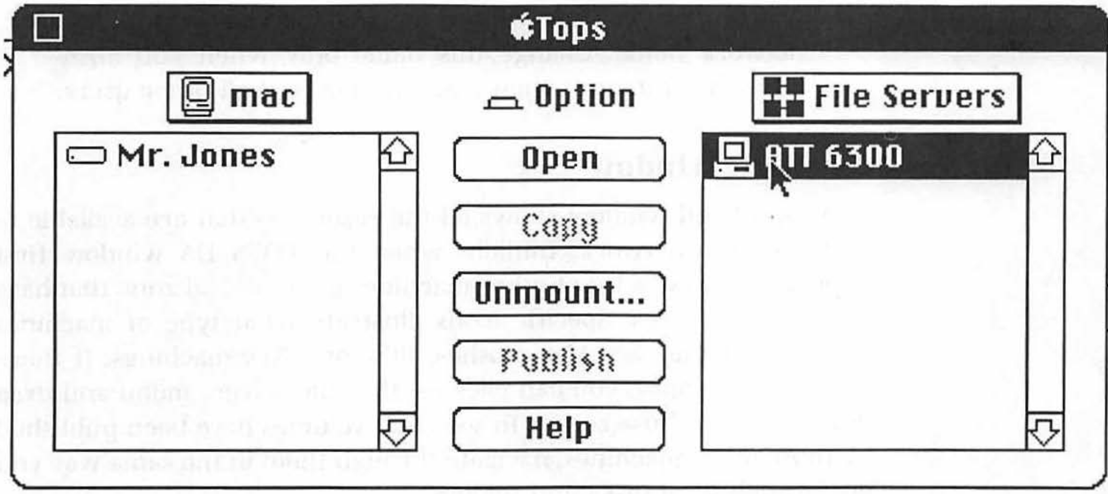


Figure 2.1 The TOPS Desk Accessory.

Local Window

The left-hand window of the TOPS DA shows the network name of your Macintosh, as well as the local volumes currently connected to your Macintosh. This will include hard disks and floppy disks. You can open any of these volumes using standard Macintosh techniques for opening folders—double-clicking on the folder or volume name or selecting it and clicking on the Open button. If you are currently in a folder, the name of that folder shows at the top of the left-hand window. To move up to see the folder that contains the current folder, click on the name of the folder at the top of the window, hold down the mouse button, and drag down until you are at the folder you wish to see. The Macintosh Owner's Manual discusses moving through hierarchies of folders in greater detail.

To change the network name of your Macintosh, select the network name (at the top of the left-hand window). With the name selected, hold down the Option key and click on the Open button. TOPS will present you with a dialogue box that allows you to type in a new network name. TOPS will remember this network name; the next time you start the machine, this network name will be registered.

Note When you change your network name, any automatic mounting of your volumes by others on the network will not work: They will be looking for you by your old network name. Change this name only when you absolutely need to and when you have notified all other users.

Network Window

The right-hand window shows all the resources that are available to you over the network. Initially, when the TOPS DA window first appears, it shows a list of other machines in your local zone that have published volumes. Specific icons illustrate what type of machines they are—if they are Macintoshes, PCs, or UNIX machines. If there are multiple zones, you can click on the File Servers menu and drag down to show those zones. To see what volumes have been published by those other machines, navigate through them in the same way you do through local disks and folders.

Buttons

The portion of the TOPS Desk Accessory in the middle of the window contains six buttons that actually perform the various TOPS functions.

Option The first button in the TOPS window is the Option button. Most of the buttons in the TOPS window function in different ways depending on whether the option key on the keyboard is held down. We'll discuss these in greater detail later. When you click on the Option button, the graphic display of this changes: If it is the first time you click on it, the button goes down. In other words, if you click on any of the other buttons, they will perform as if you had held down the option key on the keyboard. Clicking on this button again reverses its state: If it was down, it comes up. Buttons whose functions are altered by the option key will change in their appearance when this button is down.

Open This button opens volumes in either the local or network windows. It merely opens the selected item so you can see the names of the other items contained therein. It is basically the same as double-clicking on a volume name. Opening a file server or zone name in the network window shows the names of the volumes or file servers that are contained in that item. If you open a folder or volume in either window, you will be able to see the names of volumes, folders, or files that are contained in that item. Using Open or double-clicking on a volume allows you to see what files are on that volume before mounting it.

Copy This is one of the handiest of the TOPS buttons; it allows you to copy files from local volumes to network volumes (or from network to local volumes) without actually mounting the networked volumes. To copy a file with this button, open your local volume so that the folder into which you want to copy the file is opened. Next, find the file you wish to copy on the networked volume. Once you have selected the file, the Copy button will be enabled (i.e., the text will be black instead of grayed out), and arrows will appear in the Copy button indicating whether you are copying to or from the local volume. In this case, it will point to the local volume. Click on the Copy button to copy the file. To copy a file to a remote volume, reverse the process.

Copying files with the TOPS Desk Accessory is a great shortcut and a boon to speeding up network performance. The TOPS DA is always available. You do not have to mount the volume and work from the Macintosh Finder; you do not need to quit your application.

> ***Note*** Copying a file to a local volume to edit it is not always desirable. This creates two versions of the file in two different places; later it can cause confusion over version numbers. If the file is one that will be edited by several people, you should leave it in its original location while working on it.

Option-copy Holding down the option key and clicking on the Copy button (or clicking on the Copy button with the Option button on) changes the appearance of the Copy button to "Copy . . . ," indicating that it will perform a slightly different function. It brings up the dialogue box in figure 2.2 and allows you to perform a very useful function: removing nontext characters from a document. As discussed in chapter 4, the Macintosh and the PC do very different things with their files, particularly the text files. The PC, for example, adds a linefeed character (ASCII character 10) along with a return character at the end of a line or a paragraph. Option-copy removes most of these linefeeds when copying a file. Other programs will also insert control characters into text when formatting it. Option-copy removes these characters as well. To remove the nontext characters from the file, click on the button labeled "Copy Text Only." To copy the entire file, click in the Copy All Data button. To cancel, click on the Cancel button.

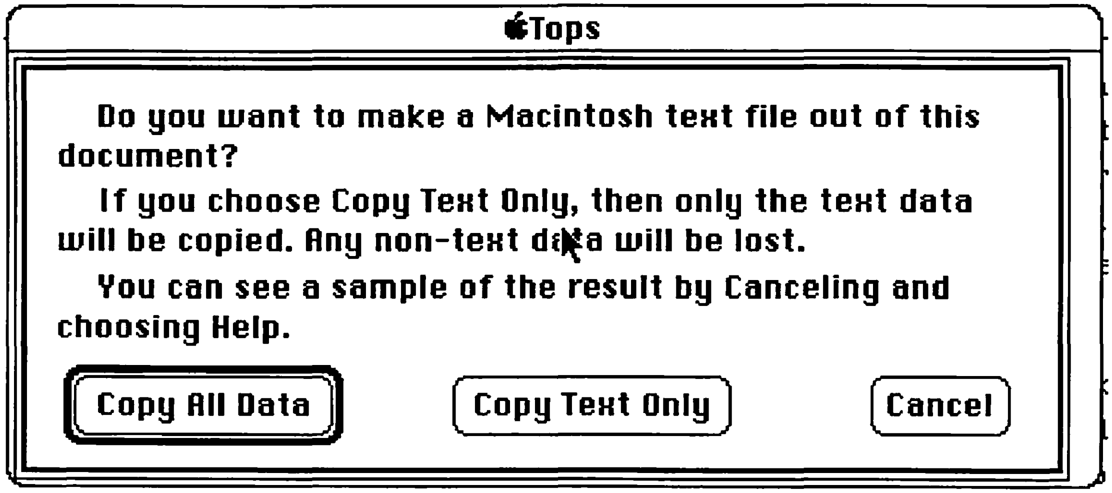


Figure 2.2 The Option-copy dialogue box.

> ***Note*** You can lose some data when making a Macintosh text file out of a PC file. If you have a program on the Macintosh that can read a PC file directly, do not use this option. Likewise, if you have a conversion utility that will convert files between different formats, use that utility to perform the conversion, since some of the nontext data may contain codes that are necessary for the file. Use the Help function described later to help determine if there will be loss of data with an option-copy.

It is also possible to copy a Macintosh file to a PC and remove the nontext characters. While this function is useful, some Macintosh files can lose much data with it. As discussed in chapter 1, Macintosh files are very different from PC files, which often contain information in the resource fork of a file. Option-copying deals with the data fork of a file only, so the chances of loss of data are much greater. As with networked files, use the Help function described later to examine the file and help determine whether data will be lost.

Mount The next button in the TOPS DA window is the Mount button. This button mounts a networked volume and is enabled only if you have selected a networked volume (or server). That is, it makes that volume appear on your DeskTop (and to your programs) as if it were a local disk. To mount a volume, select the volume in the list of networked volumes and click on this button. When you close the TOPS DA, the volumes you mounted will appear on the DeskTop (if you are in the Finder) and will be accessible to you. If any of the volumes that you are mounting have been given a password, you will be asked for that password.

Unmounting If you have selected a volume that is already mounted, the Mount button will change to "Unmount" allowing you to close your access to that volume. If a server is selected in the right-hand window, you can unmount all volumes mounted from that server by clicking in this button. If you have a selected zone in the right-hand window, you may unmount volumes from all servers in that zone.

Unmounting, by the way, can also be performed directly from the Finder, without using the TOPS Desk Accessory, simply by throwing the volume away. Just as you eject a floppy disk and tell the Finder to "forget" the disk by dragging it to the trash, the Finder will unmount a TOPS volume in the same manner. In fact, if you are in the Finder and want to unmount a volume, this method is preferable to using the Desk Accessory.

Option-mounting With the option key held down (or the Option button in its down state), the Mount button changes to "Mount . . . ," indicating that the function of the button will be enhanced. The window that appears (shown in figure 2.3) asks if you want to mount the volume as a volume you can write to as well as read from or if you want it to be read only. If the owner of the volume has published it as a read only volume, the Read/write button will be disabled, and it will appear on your DeskTop as a locked disk.

The Remember button in this dialogue box allows you to tell TOPS that you always want to mount this volume. If you click on this button, the next time you start your computer, TOPS will "remember" that you wanted to mount this volume and, if the volume has been published, will mount that volume upon startup. Generally, as is discussed later in more detail in "Strategies for TOPS," you should

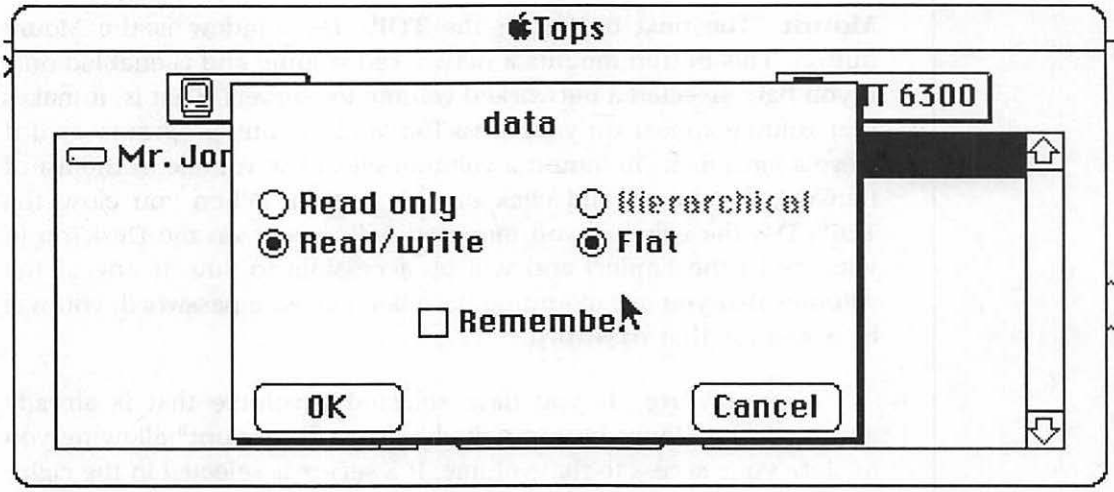


Figure 2.3 A window appears when you hold down the Option key when mounting a volume with the TOPS DA.

only remember a networked volume if it is absolutely necessary that you always have access to it.

The Hierarchical and Flat buttons refer to the two types of Macintosh file structures. Since the Macintosh cannot currently mount a PC volume as a hierarchical volume, these functions are not yet available. All PC volumes are automatically flat and Macintosh volumes are either flat or hierarchical depending on how they were originally formatted.

Publish The next button is the Publish button. It is enabled when you select a volume or folder in the list of your local volumes. If you have selected a volume already published, then this button changes to "Unpublish." If the option key is down, then this button changes to "Publish . . . ."

Publishing a volume makes it available to others on the network. Before anyone else on the network can access a volume, it must first be published. To publish a volume, simply select the volume in the list of your local volumes and click on this button.

Option-publish Holding down the option key while publishing a volume brings up the dialogue box shown in figure 2.4. This dialogue box allows you to set various parameters for publishing a volume to the network including assigning a password to the volume and setting the options as to how many others will be allowed to write (i.e., change or save files) to that volume. The options follow.

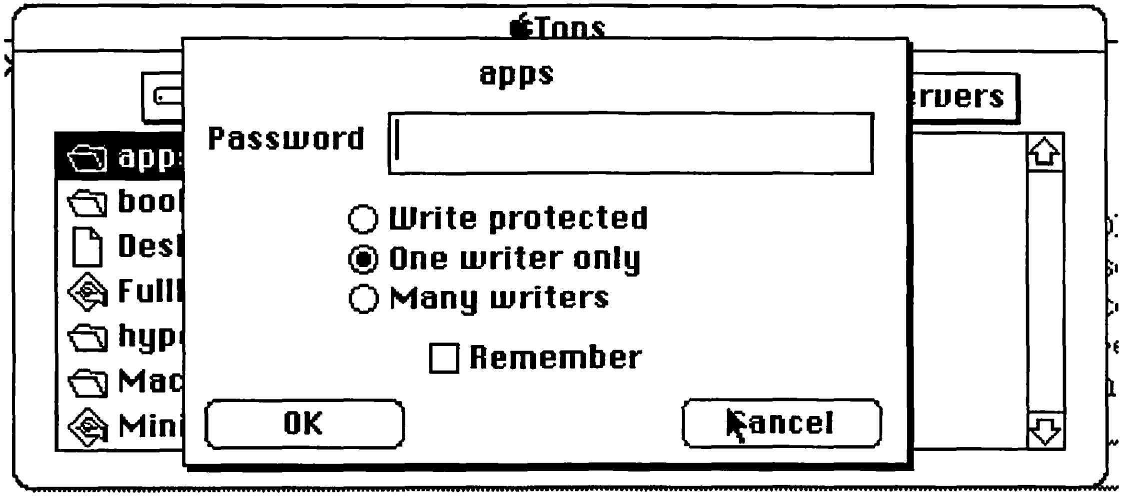


Figure 2.4 "Option-publish" brings up this dialogue box.

Password. Before others can use the volume, TOPS will ask them for the password.

Note When you type in the password in this dialogue box, it is not scrambled as you type it. Make sure nobody is watching over your shoulder.

Write protected. This locks the volume to all network users of this volume. To others on Macintoshes, it will appear as a locked disk, and the TOPS DA will not allow them to mount it as a read /write volume. Nor will they be allowed to save to that volume or to make changes to files on it. If you wish to publish data to share with others but do not want to allow them to alter the data, use this option.

One writer only. Only the first person to mount a volume will be allowed to write to that volume. Successive users of the volume will see it as a locked or read only volume. If the first user of the volume unmounts it, then its status will change for the next person who mounted it, and it will become a read/write volume to that user. This is the TOPS default. That is, unless you use the Option-publish function to change the status of this volume, it will be published in this manner and without a password. This is an excellent safeguard to use; it protects the volume against corruption by programs that are not network-aware.

Many writers. Publishing a volume in this manner allows all those who mount the volume to be able to make changes to that volume. All will see it as an unlocked volume. Publish volumes that contain applications or data for all to use, see, and change in this manner. This is not an inherently dangerous procedure. The Macintosh operating system still protects files from being opened simultaneously by more than one user.

Remember. As with the Mount button, the Remember button allows you to instruct TOPS to always publish this volume when you start the System. This is a good way to handle volumes that are necessary for others on the network to use. We all forget things, and telling TOPS to remember to publish a volume is part of good network citizenship. Others on the network won't have to call you to ask you to publish a volume. If you have a System crash while others are using a volume that you have published, "remembering" to publish that volume also increases the likelihood that others will not lose any of their work: The volume will be published as soon as possible when your Mac is restarted.

It is important to remember that publishing a folder or volume also publishes all folders contained in that folder. You might have a folder called "Data" that you want to make accessible to the network. But in that folder, you might also have another folder called "Private" that you do not want to share with the network. If you publish "Data," "Private" will also be on the network. There are several ways to get around this.

The best way around this is to move the "Private" folder out of the "Data" folder and not publish the "Private" folder. That is the simplest way: If you do not publish a folder (or a folder that contains that folder), then others on the network cannot access it.

Help The final TOPS button is the Help button. It performs several functions, depending upon what is selected in the other windows.

Getting help If you have nothing selected in either the list of your local volumes or in the list of networked volumes, then the Help button does what its name suggests: It brings up information about how to use the TOPS Desk Accessory. Specifically, it brings up six small windows, each containing information about the various TOPS buttons. If you have forgotten how to use some of the buttons, this is a good reminder.

> ***Note*** The text for the TOPS help is contained in a file called, appropriately, "TOPS help." It is located in the System folder of your startup disk. This file uses 44 kb of disk space. On a System using a floppy startup disk, where space is at a premium, you can remove this file. When you click this button and there is no Help file, TOPS acts as if you had option-clicked the button.

Option-help Holding down the option key and clicking on the Help button displays the version number of the TOPS you are running, and the date TOPS was last modified (see figure 2.5). It also shows your TOPS serial number. All copies of TOPS that are in use on the network should be of the same version and date, or conflicts can occur. When debugging a TOPS network, especially when unusual or unexpected things are happening, always make sure that all copies have the same version number. Do not update one copy of TOPS on the network without updating all copies. And, very importantly, before calling for technical support on TOPS, check the version number.

File help If you click on the Help button with a file selected in either the list of your local volumes or the list of your network volumes, TOPS will provide you with some information about the file. For files on your local Macintosh disks, two windows will appear. One will tell you whether the file is a local file, and the other will give you information about the file, including the size of both the resource and data forks (see chapter 1) and the dates it was created and last modified. A third window will show the filtered file.

For files on networked volumes, the help function will show you three windows. The first will tell you whether the file is on a remote or networked volume and give you some information about what you can do with that volume. The second window will show the length (in both the resource and data forks) of the file and the dates it was

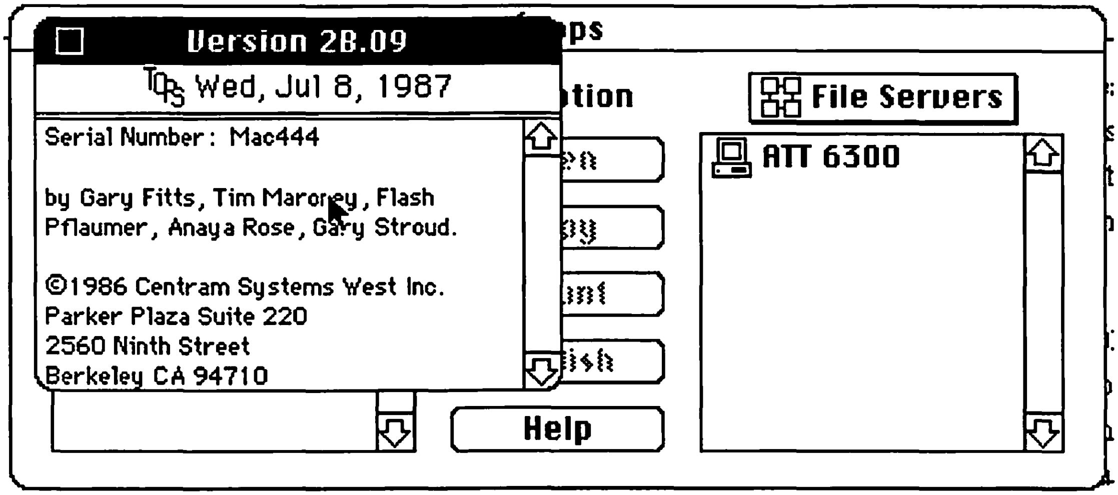


Figure 2.5 Window displaying the version, date last updated, and serial numbers of your copy of TOPS.

created and last modified. A third window will show a filtered portion of the file.

Filtered files As already discussed in dealing with the Copy button, TOPS can filter nontext characters from a file when copying it, thus creating a file that contains only the text characters in the file. The filtered window shows the first 3000 text characters of a file. This window is handy for several reasons. First, you might not remember the contents of a file by the name. Rather than having to start up an application to tell you what the file contains, you can use this button to examine it (something like using the type command in DOS to display a file on the screen). Second, if the file is a PC file and you want to move it to the Macintosh (or move a Macintosh file to a PC), you can use the filtered view to help you determine whether you will need to use some other translation utility to convert the file before it can be used on the other computer. Frequently, filtering a file allows it to be edited on the other computer without too much loss of data. (You will almost certainly lose most formatting information, and going from a Mac to a PC will definitely cause loss of paragraph markers—see chapter 4C).

That wraps up our discussion of the TOPS Desk Accessory. For details in installing TOPS on your Macintosh, refer to the TOPS Owner's Manual.

Strategies for TOPS

Publish Early, Mount Late

This is one of the key rules to follow when being a good network citizen. It increases the efficiency of the network not only as you perceive it, but as others on the network perceive it as well.

Publish Early If others rely on having certain parts of your hard disk available to them over the network, it is very important that you make them available as soon as possible. For those folders that others absolutely need, make sure you check "Remember" in the dialogue box that appears when you hold down the option key upon publishing a volume. Clicking in "Remember" means that your Macintosh will publish this volume as soon as it boots.

Mount Late By the same token, unless you really need a remote volume, do not click in the "Remember" box when mounting it. You might, for example, really need a remote volume that contains a System folder you always use. On the other hand, you probably do not always need many remote volumes that contain only documents or certain applications.

As already mentioned, remote volumes can be mounted while you are working in most Macintosh applications. In fact, this is many times preferable. Mounting a volume from the Macintosh Finder can take some time. The Finder must update the DeskTop on the remote volume—assigning icons, locations for icons, and the like. This can slow down operations both on your machine and on the machine that has published the volume. The Finder must do something like this every time you return to it. If the network is busy, if there are many files in the mounted volume, or if you have several mounted volumes, this can slow things down considerably. One way to speed things up, decreasing the time the Finder has to spend dealing with disks, is to mount only volumes from within applications.

For example, you might be working in Microsoft Word and need to load in some text created in Word on a PC. Assuming that the PC hard disk has been published, you do not need to quit Word on the Mac. You simply must summon the TOPS DA, mount the volume, close the TOPS DA, and use Word's standard "Open . . . "

dialogue box to access the file. After you are done working with the file in Word, save it, close it, and use the TOPS DA to unmount the volume. By keeping the volume mounted only while you are actually using it, you have helped yourself (things happen faster the next time you return to the DeskTop), and you have done a public service: There is less traffic over the network so you have not slowed down others.

Some caveats to this procedure follow. Before you can unmount a volume, you must first make sure you have closed all the open files on that volume. The TOPS DA will notify you when you have open files. In that case, it is best to return to your application and make sure the file is closed before unmounting the volume. With some programs, it is possible that, even though you might have used the Close menu to close the file, it is actually keeping it open. Microsoft Word 3 is a case in point. Word keeps temporary files on disk to hold portions of text that it cannot keep in memory. If it has created these files on a remote disk, it might not actually close them when you close the document. In such cases, it is wiser to return to the DeskTop before unmounting the volume. If you are running under MultiFinder, mounting a volume from within an application still mounts it with the Finder—since the Finder is always running in this environment. In this case, you should probably switch quickly to the Finder and use the standard technique for "forgetting" the volume.

A variation on this theme uses the TOPS DA not to mount a volume but to copy a file to a local drive from which it is opened. When you are done with working on the file, use the TOPS DA to copy it back to the remote drive. The problem with this is that you have to make sure that no one else has done the same thing and that you do not destroy the changes they have made to the file.

Cache Wisely

One of the Macintosh Control Panel settings is for a RAM cache. Using the cache wisely can speed up most operations and increase the performance of the network.

A cache is a special area of memory that the computer sets aside to hold material that is frequently requested from a disk drive. For example, if you are working with a large database or word-processing file, that file cannot always be held in its entirety within the Macintosh RAM. A program will only call for it when it is needed for a particular purpose. The RAM cache holds data that is frequently requested from disk, thus speeding up accessing of that data.

By setting the RAM cache of a machine that is using network resources to a fairly high level (say, 128 kb), many of the functions of the computer will go much faster. This is especially true over a network. Since the Mac views a networked hard disk the same way it views a local hard disk, information coming over the network will be cached. Additional requests for it might come only from RAM. This not only speeds up local operations but also lessens network traffic, thus speeding things for everyone.

It is almost impossible to make hard and fast recommendations about RAM cache size. Different application programs have different memory requirements, and they can actually be slowed down (or not work at all) if the RAM cache is set too high. And, of course, if you have a large amount of RAM in your machine (more than 1 Mb), you can set much larger cache sizes than you might otherwise set. You will have to experiment to find the optimum cache value for your System and the programs you run.

You cannot change the size of the RAM cache from within a program. Rather, you can, but the change will not take effect. Changes in RAM cache take effect only when the next program is run.

Run Applications Locally

While it is not always possible or desirable, given your configuration, to do this, network performance can be increased greatly if the networked resources are used primarily to transfer data instead of application programs. If you keep the programs local—either on a local hard disk or floppy—performance for both the local user and others on the network is enhanced.

When you double-click on a program icon on the Macintosh, not all parts of that program are called into memory at one time. Rather, the program only loads what is necessary for it to run. As you access parts of the program, those are called into memory as they are needed. Many programs—especially the larger ones— do this to a considerable degree. With these larger programs, it makes sense to store and run them from local devices, minimizing network traffic.

A few programs—Microsoft Word 3 is the only one that comes to mind—allow you some flexibility in this matter. In the Word 3 Preferences . . . dialogue box, there are check boxes that allow you to instruct Word whether it should keep the program, the file, or both in memory. If you are running the program from a networked device, instruct it to keep the program in memory. If you are loading files across the network, tell it to keep the file in memory.

Using a Diskless Workstation or Local System Disk

There is much written in the computer press about the diskless workstation. With the Macintosh, it is not currently possible for the computer to boot from the network. (Many PC networks have special cards that allow this.) It is possible, however, to use a floppy disk to start the computer and then switch Systems to a networked drive.

The Macintosh System software is in an almost constant state of change—as Apple adds new capabilities or problems are found, new versions of the System file, the Finder, and other utilities appear. It is important that all users on the network have the same version of these programs. By storing System files in only one or two locations on the network, you can more easily update to new versions and make sure that all users are working with the same version.

Additionally, with the large numbers of fonts, desk accessories, printer drivers, and other utilities that are available now for the Macintosh, all the needed resources may not fit on a single disk. Keeping all these elements on a remote hard disk reduces this possibility.

Cautions Some problems associated with working with System files that are on a network follow.

Network traffic increased As already discussed, the Macintosh frequently needs to call from such disk-specific resources as desk accessories, fonts, and information about dialogue boxes. Many of these resources are called from the System file. If the System file is located on a remote disk, the amount of traffic on the network is drastically increased. This practice degrades performance for all.

Possible conflicts with other Systems With the current Macintosh System files, multiple users cannot share the same System file or Finder. Therefore, separate copies of these files must be kept on any disk that you want to use as a System file server.

Only one System on a disk With Macintosh hard disks, the rule is that there should only be one System on a hard disk. When the Macintosh starts, it loads from the first System it finds on the disk (usually in a folder called "System Folder"). This folder—the one the Mac boots from—is called the "blessed folder." TOPS handles this conflict well. If you mount a volume that contains the "blessed" System file, TOPS renames this System file to reduce conflicts. That prevents Finders on other disks from attempting to use that System.

The Macintosh should not find other versions of the System and Finder on that disk and attempt to use them. In a situation where there is only one user per hard disk, or where all Systems are run locally for all users, this is no problem.

However, if you want to keep several copies of the System and Finder on a particular hard disk, you must take steps to ensure that the machine does not get confused as to which System to use. One way to do this is to assign different names to different Systems. You might have several System folders on the disk with different names: "John's System Folder," for example, or "Marketing System Folder." Inside these folders, the actual System and Finder files should be renamed, in a manner consistent with the name of the folder. You should also keep these folders a few levels below the level of the main System folder on the hard disk, as a secondary guard against confusion. On networks where the System files are kept on a PC disk, the possibilities of confusion are decreased.

You can switch-launch Systems in two different ways. Both require that the new System disk first be a mounted (and thus published) volume. Assuming that you have mounted the volume, you can open the folder containing that System, locate the Finder (which might not be named "Finder"), then hold down the command and option keys on the Mac keyboard, and double-click on the Finder icon. This instructs the Macintosh to execute the Finder as if it were a program. The Finder will automatically use the closest System.

> ***Note*** With some programs, such as Affinity System's Tempo, it is possible to construct macros that perform this automatically at startup. Programs such as Tempo can be useful if you are setting up a system for users who do not wish to learn all the details about networks, switch launching, and the like.

As already mentioned, if you wish to store System files on a PC disk, there is no worry about the server machine getting confused about which System to use. There are some other problems, though. If you mount a volume on the PC that contains other subdirectories, they won't be visible as folders on the Macintosh DeskTop. In this case, after mounting the volume, you may have to mount others on the disk separately. With more volumes opened, there is more traffic over the network, and things slow down.

While switch launching is certainly possible and might have certain benefits, you must carefully weigh them against the side effects of possible System confusion, more network traffic, and loss of a certain amount of ease of use.

Part C *TOPS ON THE PC*

TOPS for the PC is a complete environment—hardware and software—for connecting PCs in a network and for connecting PCs and Macintoshes together in a network. TOPS for the PC is compatible with network standards set up by Microsoft and IBM for file and record locking within the DOS environment. Programs written to work with standard DOS network products will generally work as well with TOPS for the PC. TOPS also provides a mechanism for PCs to share nonnetworked printers with other PCs (not Macintoshes) on the network.

There are three components to TOPS PC.

1. TOPS PC is the software that provides TOPS disk and printer sharing for computers on the TOPS network.
2. The TOPS FlashCard contains hardware that allows PCs to be present on AppleTalk networks.
3. TOPS NetPrint is software that allows PC applications programs to access the Apple LaserWriter and other PostScript printers.

For TOPS PC to work, an AppleTalk card (e.g., the TOPS FlashCard) must be present to allow the PC to connect to the AppleTalk network. Aside from the FlashCard, TOPS PC can work with AppleTalk cards from Apple, Hercules, and others. By the same token, the TOPS FlashCard allows other software programs (e.g., AppleShare PC) to communicate over the AppleTalk network.

As discussed in part A, the basic mechanisms that TOPS uses for sharing resources—whether those resources are disks or printers—are publishing and mounting. To make a resource available to the network, you must publish it first. For any workstation to use that resource, you mount it first. An overview of these TOPS functions is found in part A; specific instructions for publishing and mounting volumes are detailed here.

Flexibility

Network Design

TOPS is a flexible network. Any computer on the network can be a server (one that publishes resources) or a client (one that uses network resources) or both. You do not have to set up the network in a special way or do things such as install special software on a server. Dedicated machines for file or printer serving are not necessary, although they can be used with TOPS.

User Options

TOPS can also be used in several different ways. TOPSMENU is a menu-driven program that allows easy access to all functions of the network. The relatively unsophisticated DOS user can be using TOPSMENU soon after installing TOPS. Once you understand the concepts of publishing and mounting, the rest comes easy.

For the more sophisticated user, TOPS includes a full range of commands that can be executed from the DOS command line, giving a more interactive feel to network operations. As with any standard DOS commands these TOPS commands can be placed into batch files for easy execution of routine operations. Batch files provide a way for a network administrator or sophisticated user to shield the novices from the intricacies of TOPS.

Components of TOPS

TOPS for the PC is composed of both hardware and software.

Hardware

The TOPS hardware is simply an interface card or board that adds to the PC circuitry similar to that which is built into the Macintosh and allows it to communicate over AppleTalk networks.

Software

The TOPS software has three layers of components.

Device Driver This device driver—called "ATALK.SYS"—is loaded when the system starts, before DOS is loaded, and merely provides a low-level interface to allow DOS software (and the other TOPS modules) to communicate with the TOPS hardware.

Memory-Resident Network Programs These three programs—TOPSTALK.EXE, TOPSKRNL.EXE, and TOPSPRTR.EXE—are required for all other functions. These programs remain in memory once they have been executed and are required for all functions of TOPS to work. (The exception to this is TOPSPRTR.EXE—it is only needed if you are going to share printers with other PCs on the network.)

TOPS Commands These include TOPS.EXE, TOPSMENU.EXE, and the other programs that come on the TOPS disk. Essentially, they provide the user interface for TOPS—the mechanism by which users control TOPS functions.

Installing TOPS

Hard Disk

Installing TOPS is a straightforward process, covered in the TOPS Owner's Manual. The on-screen prompts in the Install program are straightforward and easy to follow. However, here are some concepts that you should be aware of.

The TOPS Directory The TOPS Install program will ask you in what directory to place your TOPS files. The default value is C:\TOPS, but you can override this and place the files in another directory if you like. No matter where you place your TOPS files, there are a couple of things to keep in mind.

1. Keep your TOPS files together. It is a good idea to keep all your TOPS files in one directory. The Install program will do this for you, but you should not move them around afterward. This makes for easier updating of TOPS and for easier operation of the computer. However, if you usually keep all your batch files in a directory of their own, which is a good idea not often implemented, then you might keep your TOPS batch files there as well.
2. Put your TOPS directory in the search path. The search path is the list of directories that DOS searches to locate a command you type at the DOS prompt. This means you can type TOPS commands from the keyboard, no matter what directory you are logged into. The TOPS Install program will add the TOPS directory to your path statement in your AUTOEXEC.BAT file if you wish. If you do not understand the search path, see your DOS Owner's Manual, which will talk about paths in general and the path command in particular.

The CONFIG.SYS File As a PC is started—when it is booting—one of the first things it does is read some information from a file on disk called "CONFIG.SYS." This file contains some information regarding certain operating parameters of the computer, such as how many memory buffers to set up to speed disk access and how many files the operating system should permit programs to open at one time. The CONFIG.SYS file also includes statements about what other nonstandard devices it can use. These device drivers may include such things as mice, networks, and CD-ROM players. The CONFIG.SYS file usually just tells the computer what device drivers to use and where they are located on the disk. The reference to a device driver in the CONFIG.SYS file will usually look something like this:

```
DEVICE=C:\MOUSE.SYS
```

The TOPS interface card is a nonstandard device that requires a device driver. Any reference to that device driver must be made in the CONFIG.SYS file, and the TOPS device driver (ATALK.SYS) must be in a location readily accessible to DOS. The TOPS Install program will ask you whether you want it to so modify your existing CON-

FIG.SYS file and will also place the TOPS driver ATALK.SYS in the root directory of your PC hard disk.

The ATALK.SYS driver must be active for TOPS to work. If you do not allow the TOPS Install program to do this for you, then you must do it yourself.

> ***Note*** The ATALK.SYS file does not necessarily have to be in any particular location on the disk for the driver to work; it must only be in the location specified in the CONFIG.SYS file. For example, the line "DEVICE = ATALK.SYS" will cause DOS to look in the root directory of the disk for the driver. If you place the driver in another directory, the line can read:
>
> ```
> DEVICE = C:\TOPS\ATALK.SYS
> ```
>
> DOS will look in the TOPS directory for the driver. Since DOS reads the contents of the CONFIG.SYS file before doing anything else, the path command will not work to direct DOS to the location of the device driver.

Once the TOPS Install program has copied the TOPS files onto your disk, and your CONFIG.SYS file has been modified to allow it to load ATALK.SYS, you are ready to run TOPS. As the TOPS Owner's Manual specifies, however, it is necessary for you to first reboot the machine to use TOPS. Since the drivers that are part of CONFIG.SYS can only be run before DOS is loaded, it is necessary that the machine be restarted for the other functions of TOPS to work.

Floppy Disk

Installing TOPS on a floppy disk is very much like installing it on a hard disk. The TOPS Install program does everything for you. Certain things are the same: You need to have a CONFIG.SYS file that summons the ATALK.SYS file, and you may have subdirectories for the TOPS files. However, since the available space on a floppy disk is much less than on a hard disk, some differences should be noted.

1. You must install onto a DOS System disk. You can create a System disk by using the format command with the /S parameter. (The /S signifies System and instructs the format command to place the two hidden DOS files and COMMAND.COM onto

the disk.) To create a System disk on the blank floppy in drive B:, type "FORMAT B: /S" at the DOS prompt.

2. Instruct the TOPS Install program to put TOPS onto your floppy disk. If you have two disk drives, you probably have the TOPS installer disk in drive A:. Put your new disk in drive B: and instruct TOPS to install to that disk. If you only have one floppy, tell TOPS to install on drive B: anyway, and you will be prompted to switch disks.
3. TOPS will install all the TOPS files on the floppy. If, however, you need to put other software on the floppy disk, you can then delete some of the TOPS files to make room. Here are the absolutely essential TOPS files that you must have on disk:

   TOPSPRTR.EXE must be on your disk if you are going to share your printer with other PCs on the network or use other shared printers (non-AppleTalk).

   TOPS.EXE contains the command-line TOPS interface and must be present if you are going to use TOPS from the DOS command line or batch files.

   TOPSMENU.EXE is needed only if you are going to use the menu-driven TOPS interface. If not, and all you are doing is using TOPS from the command line or batch files, this file can be deleted.

Starting TOPS

Once you have installed TOPS correctly, starting TOPS is very easy. Assuming the computer has been started from a disk that has the correct entries in the CONFIG.SYS file, the following two commands can be entered from the DOS prompt:

```
C:>TOPSTALK
C:>TOPSKRNL
```

These two programs actually activate TOPS in your computer and are necessary to access the functions of the network. These programs will execute and display their version numbers on the screen. Since they are TSR (Terminate and Stay Resident) programs, they will remain in memory. The only difference you see in your computer will be the ability to access TOPS functions. When TOPSKRNL.EXE executes, it looks into your TOPS directory to find TOPSKRNL.DAT, which contains information about a number of variables that control how TOPS operates on your computer (including such things as how many volumes you can publish or mount and your station name). If this file is not found, TOPSKRNL.EXE will not execute.

If you are going to be using TOPS to share non-AppleTalk printers with other PCs on the network, you must also execute the TSR program that provides the necessary support for this function. The command is TOPSPRTR and is entered at the command line in the same way as the other TSR components of TOPS.

> ***Note*** If you are not going to be sharing printers with other PCs, do not execute this program. It uses memory that will be put to better use by the programs you run.

After you have started TOPS you can use any of the TOPS functions. You can run TOPSMENU or TOPS.EXE to publish or mount volumes or printers.

If you will need to access other printers on the network or to share printers, then you will need to type "TOPSPRTR" on the command line or include it in your AUTOEXEC file.

If you always want to start your computer with TOPS available, you can insert these commands into an AUTOEXEC.BAT file, which can also be used to publish a standard set of volumes. A sample AUTOEXEC.BAT file is detailed in "Batch Files Are Your Friends" later in this chapter, along with some other batch files for use with TOPS.

Using TOPS

Once you have started TOPS, you are ready to use it to publish and mount volumes or printers. TOPS provides two mechanisms for accessing its functions. We'll talk about them separately. Before we discuss using TOPS commands, though, let's review some TOPS basics and discuss some strategies.

As discussed in the opening of part B, the basic mechanism for sharing devices over TOPS and for accessing devices that others have shared is publishing and mounting. Publishing takes a local resource—typically a hard disk or a subdirectory of a hard disk—and makes it available to others on the network. Mounting accesses a hard disk or subdirectory that others have published and makes it appear as if it were a local disk. Thus, to share your disk, you must publish it; to use a shared disk, you must mount it. The same is true for printers.

Publishing

When you publish a disk, nothing happens on your local machine, from your point of view. Except for occasional times when some operations might slow as others access your disk, you won't really notice anything different. You do, though, have some added responsibilities to others on the network.

Foremost of these new responsibilities is that you do nothing on your machine that could seriously interrupt the work of others who might be using your hard disk. You definitely do not want to shut off your machine or reset it; others who are using your published hard disk could have the rude and shocking experience of no longer having the disk available and could lose much work. Other operations that slow down or seriously impair TOPS operations are intensive operations that use much the energy of your computer—formatting a disk is such an example. Using a large database and performing a complex operation such as indexing it is another operation that can impair network operations. In general, when you are using a computer that is acting as a server, keep your local operations to a minimum. That is not to say that you should not run the software you need. Instead, you should take the needs of others into consideration and run complex operations only at a time when others are not trying to do their own complex operations involving your computer.

Publishing also entails that you take some responsibility for others who might rely on having certain of your resources always available. If, for example, you have a resource on your computer that is often (or always) needed by other users, you must make sure that a resource—disk or printer—is always available. Put the commands to publish those resources in your AUTOEXEC.BAT file, so that you do not have to remember to publish them manually every time you start your machine.

Mounting

When you mount a networked volume to your computer, the only thing that changes is that it appears that you have another disk connected to your computer. If, before mounting, you have drives A: (floppy disk) and C: (hard disk) available, after you mount a networked drive, you will probably have a drive D: that you can access in the same manner as you accessed A: or C:. The same commands that you use from DOS will work with this networked disk, and the same commands that you use in programs to open or save files on different disks will work with the networked disk. The main difference is that it will take a little longer to get information from your networked disk than it will take to get it from your local hard disk.

Just as publishing volumes entails responsibilities toward the users of those volumes, mounting volumes adds responsibilities to those who have published those volumes. Using someone else's hard disk is something like being a guest in somebody else's house: You do not want to leave your dirty underwear in the bathroom, and you want to make sure you pick up after yourself. When you are working on a networked hard disk, there are several things you can do to make sure that you are a welcome guest.

1. Delete unnecessary files. Almost every computer user I know has many old files laying around on their hard disks. That's fine if you are the only person using the disk, but when there are several users who are doing the same thing, the disk can fill up pretty quickly. Always back up your outdated or completed files to one of your floppies and delete them from the server you are using.
2. Do not keep unnecessary backups of files on the hard disk. One of the better techniques for saving yourself from disaster is to save a file under a different name before you perform some drastic action (e.g., deleting many records of a database or performing a search and replace on a large text file). That's always a good thing to do, especially on a network where files might be more fragile than otherwise. However, when you are a guest on somebody else's hard disk, be doubly sure to delete those files as soon as you know you won't need them any more.
3. Unmount volumes when you do not need them any more. Even though you are not actually doing things with a mounted volume,

it can still bog down operations on a server PC. Cut down on the likelihood of this happening by unmounting volumes that you are not actually using. Mount volumes when you need them and unmount them when you do not. Since a server can have a maximum number of clients, this will also help to make the server available to others.

4. Naming files in a network environment is also critical. As we discuss in the section on managing the network, standard schemes for naming files must be adhered to. This helps all users to know what a file contains, what program created it, and who might own it.
5. In the PC world, you must be careful not to open files twice. Many modern DOS programs, such as MicroSoft Word, are network aware. They can determine if a file is already open and will then open it only in read only manner, not allowing subsequent users of the file to change it. However, many older programs are not network aware. Lotus 1-2-3 is one example. There is no good way to enforce this on a network, it is something that users must be careful of.

The preceding tips do not constitute a comprehensive catalog of the responsibilities of those who are acting as either servers or clients on the network. They simply provide guidelines: you should consider how your actions affect others on the network and try to structure those actions around the needs of others.

Publishing and Mounting Printers

Just as you can share disk drives with TOPS PC, you can share printers with other PCs on the TOPS network. As of this writing, it is not, however, possible to share PC printers with Macintoshes. Since most printers for the Macintosh can automatically be shared over AppleTalk, specific TOPS support for sharing those printers is not needed. More details can be found in the section "Using Networked AppleTalk Printers."

To understand sharing printers with TOPS, you must understand how DOS deals with printers and the difference between the physical connection of the printer to the computer and the logical connection—how DOS and programs refer to the printer.

Most standard PCs come equipped with a parallel printer connection and a serial printer (or modem) connection. Most standard printers are hooked up to the parallel connection, which is usually referred to by DOS as "LPT1:" (for line printer 1) or as "PRN" (for printer). If there are multiple printer ports on the PC, these are usually referred to as "LPT2," "LPT3" and so on. Usually the serial interface port will be called by the name "COM1."

However, as with drives, the names do not necessarily have to refer to specific physical connections. They can, instead be mapped or redirected so that they refer to different physical devices. For example, if you have a printer hooked up to a serial port (COM1), it is possible to use the DOS mode command to redirect this port or to make DOS send any output that you send to the standard printer port (LPT1 or PRN) to the serial port instead. This gives you more flexibility for dealing with printers.

The mode command for redirecting output from the serial port to the parallel port is:

```
MODE LPT1 =COM1
```

This command tells DOS that it should take all output that is sent to LPT1 and redirect it to the serial interface port.

> ***Note*** Before you invoke the mode command, you must also use "MODE" to set up the parameters of the serial port: its baud rate and so on.

TOPS printer redirecting works in the same manner. When you use TOPS to redirect the printer output, you are telling it not to change the physical location of your printer so much as you are changing the logical location of the printer. After you have mounted a printer with TOPS, TOPS intercepts all the output you send to the logical port you have designated and, instead, sends it over the network to the computer that published the printer, which in turn sends it to the physical address of the printer.

Before you can publish a printer, you must first execute TOPSPRTR.EXE, a TOPS module that handles all printer-sharing functions. Once this is loaded, printer sharing can work.

When you publish a printer, you refer to it in the only way that DOS knows: by its logical description. You may, for example, have a printer connected to the first serial port on your machine, but for some reason have told DOS to redirect output sent to LPT1 (normally the parallel port) to COM1. In this case, you should probably tell TOPS that you are sharing the printer on LPT1 as this is how DOS normally deals with the printer.

When you publish a printer, TOPS will also need to know where it should keep its spool files. Spool files are temporary files that contain the information to be printed. When the print server is working, it receives, as quickly as it can, all the data to be printed from the client. It then saves this information in temporary files and prints it as it can. This cuts down on traffic over the network. Since the printer is usually slower than the computer that is sending the information, the data is sent all at once at network speeds. TOPS then can send the data to the printer leisurely. This also keeps the client from having to wait for the printer. As soon as the printer server receives all the printer data, the client machine is able to do other things. The spool files TOPS creates have a DOS extension of .TPQ.

When TOPS is done printing, the spool files are deleted, reclaiming the space on the disk.

Mounting a printer involves telling TOPS to redirect any output that is sent to a specific logical device—typically LPT1—to a printer server on the network. After a printer has been mounted, any output that DOS directs to that device goes instead to the printer server to be printed there.

For example, you may have a dot matrix printer attached directly to your PC and refer to it by the name LPT1. You use this printer to print rough drafts of your letters and reports. On the network, there may be, say, an HP LaserJet 2 that you use for printing the final versions of your reports. To access this network printer, use TOPSMENU to mount the printer and indicate that you wish to refer to it as LPT2. Then, when you want to print to that printer, tell the program you are using to print to send its output to LPT2. TOPS will take care of the rest. It will take a few moments (depending, of course, on the size of the print job) for the program to send the output to the server printer, then you will have control of your program back.

Here are some things to note about sharing printers:

1. As with publishing volumes, if others rely on having your printer available over the network, publish the printer as soon as possi-

ble. Including the TOPS instructions for printer publishing in your AUTOEXEC.BAT file is the best way of ensuring this.

2. Again, as with mounting volumes, unmount printers when you do not need them any longer.
3. On machines that are heavily used as printer servers, make sure that there is sufficient room on the hard disk to hold spool files. Even though these files are deleted automatically when the print job is finished, the print jobs can back up seriously if many users are printing to the server.
4. If a computer is being heavily used as both a file server and a print server, consider dedicating that computer to server use. At the very least, keep the work that is done locally on that computer to a minimum.
5. When using a remote printer, the software must be configured to work with that printer correctly. This goes beyond letting the software know that the printer is on LPT2. The software must also be aware of what type of printer you are using. For example, if the network printer is an Epson printer, your software must be using the correct printer driver to print to the Epson printer.
6. DOS (and TOPS) cannot determine which printers are actually physically available. Therefore, users could publish (and mount) printers that do not really exist! This is not a very likely occurrence (who would publish a nonexistent printer?), but it is possible.
7. When publishing a printer, make sure that the printer is actually set up to work correctly. For example, it must be on line and have a sufficient supply of paper.

Using Networked AppleTalk Printers

In the previous section, we discussed printing to printers that have been published by other PCs on the network. But there are other types of printers that you might need to share on the network, namely AppleTalk ImageWriters and LaserWriters and compatibles. The Apple LaserWriter and LaserWriter Plus have, built in to them, their own network circuitry, and they exist on the network as their own stations. The ImageWriter and ImageWriter LQ printers have an option board that allows them, too, to act as separate stations on an AppleTalk network. Macintoshes can print to these printers simply by selecting them in the Chooser Desk Accessory.

But how about printing to these printers from PCs on TOPS? The LaserWriter is, of course, a PostScript printer and requires that programs printing to it do so in that printer language. Many of the newer programs for the PC (and those that have been updated recently) include support for PostScript. For those programs, TOPS includes a few utilities that allow PCs to use a networked LaserWriter. We'll talk about those later. For software that does not directly support PostScript, TOPS offers a separate utility—TOPS NetPrint—that, in most cases, can translate their printer formats and commands into PostScript. A new version of that program—TOPS NetPrint—is in preparation as this book is being written. It is discussed in appendix C.

The mechanism that TOPS uses to print to a networked LaserWriter or ImageWriter is the program TPRINT.EXE, which is on the standard TOPS distribution disk. This utility is designed to be a replacement for the standard DOS print command, and it works much the same way. Tprint provides spooled printing. That is, when you issue a command to have Tprint print a file, it saves a copy of that file and sends it to the printer as the printer needs it. You can still do other work.

Tprint works from the DOS command line only. It does not support the redirecting we talked about in part B. To print files to a networked LaserWriter or ImageWriter, you must first use your application program to print the file to disk. Many standard DOS programs allow you to do this. Generally, the process merely saves the text that you are printing on disk as an ASCII text only file, so that it may in turn be printed. These files might also include commands that direct the printer to perform special formatting, such as underlining or boldface. Programs that support PostScript will generate a PostScript program that describes the entire page or pages in the print job so that PostScript then may construct the page and print it.

This is all much clearer by example. Say you want to print from WordPerfect to a LaserWriter on the TOPS network. From within WordPerfect, you must first use the standard WordPerfect printer control command (shift-F7, then press the 4 key). Next select the printer you want to use.

Note WordPerfect refers to different fonts on a Laser-Writer with different printer drives. In our example, let's choose "Helvetica Portrait."

We want to print in the standard page orientation, in the Helvetica font. After we choose the printer, WordPerfect wants to know to which interface port the printer is connected (e.g., LPT1, LPT2, or COM1). The last option on the list is the one that interests us: "Device or File Path Name." When we select this option (which is number 8 on the WordPerfect 4.2 Printer Port screen), we can then type in the name of the file to which we want WordPerfect to print the file. Let's call this "OUTPUT.PSC." It's the material we are sending out, so OUTPUT is appropriate; it's in PostScript, so the .PSC reminds us of that. If we want to save this in a particular subdirectory, we can put the entire path name in as well—C:\OUTPUT.PSC.

Next, we issue the standard WordPerfect print command—hold down the shift key and press the F7. WordPerfect shows a menu that asks us if we want to print the entire document. We do, so we press the 1 key. Instead of WordPerfect sending the text to the printer, it will now create a file named "OUTPUT.PSC" in the root directory of our hard disk. This might take a few moments; you can see the activity light go on and off on the hard disk as it saves the file.

When WordPerfect is finished, we can exit the program and use Tprint to print the file. We'll talk about the techniques for doing that shortly. The techniques for printing from most programs will vary somewhat from what I have just described. However, many programs—including Microsoft Word, PageMaker, Ventura Publisher, and others that support PostScript—will work in a similar manner. By the way, the file that WordPerfect creates in this manner is much larger than the original file was. In addition to including all the formatting commands and text that were part of the original file, the print file will contain a header that WordPerfect uses. This header or dictionary will contain PostScript command that WordPerfect uses to print the rest of the text. Make sure you have ample space on your disk before you use it. If all you have on your computer is floppy disks, make sure that WordPerfect saves this file to another hard disk. If it is a long document, chances are it won't fit on a floppy.

Now, we need to use Tprint to print the file that WordPerfect created. Suppose we want to send this file to a LaserWriter named marketing. To do so, we would issue this command:

```
TPRINT LASERWRITER/Y MARKETING/N OUTPUT.PSC/X
```

To explain this line, we'll discuss each of the arguments that can be sent to Tprint. Tprint can take up to four arguments, in this manner:

```
TPRINT DEVICETYPE/Y DEVICENAME/N ZONENAME/Z FILENAME/E
```

Here's what each of these arguments means:

DEVICETYPE/Y is the type of printer to which you are printing (i.e., either an ImageWriter or LaserWriter). The "/Y" must be included on the line, and no spaces can separate it from the device type.

DEVICENAME/N is the name of the particular device to which you are printing. Besides computers on the network, printers on the network have names as well, so you can choose the printer to which you wish to print. Again, the /N must be included in the line, and no spaces can separate it from the device name.

ZONENAME/Z should be used if the printer is not on your local zone. It specifies the zone the printer is on. If the printer is in your local zone, you do not need this argument.

FILENAME/E is the name of the file you are printing. If the file is not in the current DOS subdirectory, you will need to include the complete path name to the file. If you want to print more than one document, you can list them on the same line, with spaces separating them.

The "/E" in the file name must be included, with no spaces separating the two. This sets the emulation mode that Tprint will use when printing. There are two values it can have.

/X means that the file preceding the argument will be printed in the native mode of the printer. If you are printing to a LaserWriter, the file will be sent as PostScript code. Of course, the text file that you are printing must have been prepared by a program that supports PostScript and must, in essence, contain PostScript executable code. Tprint provides no translation of standard text files to PostScript.

/L means that Tprint will print files that have this argument in the Diablo 630 emulation mode of the LaserWriter, using the Courier font. This is good if you are using a program that does not support PostScript and you merely need to print the text. Depending on how well your software supports the Diablo 630, many formatting features, such as underlining and boldface, will work.

Note The Apple LaserWriter (and many other PostScript printers) has a built-in Diablo 630 emulation. That is, each of these printers can appear, from the point of view of the software, to be a Diablo 630. This provides them with some flexibility for accepting formatted material from programs that do not (yet) support PostScript. However, as already mentioned, this precludes the use of fonts other than Courier and any graphics. This situation should get better as time passes and more programs are upgraded to add PostScript support.

Note When you are listing several files to be printed on the Tprint command line, not all the files need to be printed in the same manner (i.e., in LaserWriter or Diablo 630 mode). Each file will be printed in the manner of the argument that is associated with that file name on the command line. For each file, the /X or /L command must be included, with no spaces separating the file name from the argument.

Finding the Name and Status of Networked Printers

Tprint also includes two arguments that are useful for discovering which printers are available. The /V argument causes Tprint to list the names of all the printers or the specified type on the network. For example, to get the names of all the LaserWriters on the network, type this command:

```
TPRINT LASERWRITER/Y /V
```

Those names will be sent to your screen. To discover the status of a particular printer, type this command:

```
TPRINT LASERWRITER/Y MARKETING/N /S
```

Tprint will tell you if the printer is busy, which file it is printing, and more. You can get the names and status of all the LaserWriters on the network with this command:

```
TPRINT LASERWRITER/Y /V /S
```

A list of the printers, along with their status, will be sent to your screen. The status command can be very useful. Suppose you have a file that you need to print right away. Use this command to find out which printers are free, and then use Tprint to send the file to that printer.

Saving Some Time

It's always worthwhile to do whatever you can to save some keystrokes as you are typing. The fewer keys you have to press to do something, the fewer chances you have of pressing the wrong key. Tprint gives us one device that saves us from having to type in the type of printer and its name every time we want to print.

Whenever Tprint is executed, it reads the contents of a file called "TPRINT.DAT," if there is one in the directory that Tprint is in. You can create a TPRINT.DAT file that contains the name of the printer you will be using. You can use any standard method of creating DOS text files to create this file, the simplest is probably to copy it from the keyboard, with the COPY CON TPRINT.DAT method.

Type the following into your TPRINT.DAT file:

```
LASERWRITER/Y PRINTERNAME/N /L
```

where instead of "PRINTERNAME" you put the name of the printer you will be using. Now, to print a file, you won't need to include all the Tprint arguments on the command. To print our OUTPUT.PSC file that we created with WordPerfect, the command is much simpler:

```
TPRINT OUTPUT.PSC
```

Since Tprint reads from the TPRINT.DAT file the information about where and how it is to print the file, we do not need to include that information on the command line. This is much simpler!

A Final Note About Printing

For those programs that do not support PostScript printing, TOPS also makes TOPS NetPrint. This program provides two features that are handy for network use: redirection of the output and translation into PostScript. With the first feature, a networked PostScript printer can appear to software to be connected to a standard printer interface, such as LPT1: or COM1:. This allows you to print to the PostScript printer from within your program without having to return to DOS. With the last feature, you can print to PostScript printers from programs that do not support PostScript at all. TOPS NetPrint can translate output that was intended for the popular Epson line of dot matrix printers and IBM Proprinters and translate it into PostScript. This program was going through considerable modification as this book was being written. It is discussed in appendix C.

> ***Note*** Some programs that support PostScript need to be modified somewhat to work with both Tprint and TOPS NetPrint. This is because they are designed to work in situations where the LaserWriter is connected directly to the computer and is not used on the network. This, too, will be discussed in appendix C.

Using TOPSMENU

TOPSMENU is an easy-to-use program that allows you to use all the features of TOPS, without requiring that you know the internal workings of TOPS or DOS. It has a point-and-click user interface that, along with clear prompts, makes it always clear where you are in the hierarchy of menus and what functions you can perform.

Selecting Menu Items

The TOPSMENU screen is divided into two windows. (See figure 2.6. This opening screen shows the version number and serial number of the copy of TOPS you are running.) The top window contains menus and shows the functions that are available in the program. It also contains descriptions of the functions that are performed by the various menu items. The bottom window lists instructions for using the program. It tells how to select menu items and lists certain keystrokes or combinations that perform special functions.

```
compaq                 Welcome to TOPS                Main

TOPS for the PC Copyright (C)
  Centram Systems West 1985, 1986, 1987.
    All Rights Reserved.

    Serial Number P100000079
    PC-TOPS Kernel Version 2.00
    TOPSMENU Version 2.00.30    9/03/87

MENU INSTRUCTIONS
  This window will tell you which keys will give you access
  to the available features of this application.
  Press any key to continue.
```

Figure 2.6 The TOPSMENU opening screen.

After you select one item in the menu list, the help text in the left-hand portion of the window describes the function performed by that menu item. If you are not sure what each menu item does, simply move the highlight to the menu item and read what it does.

There are several ways to select the various menu items. Usually there will be fewer than six items listed in the menu window. You can select any item by using the PC's cursor keys (either on the numeric keypad or on a separate cursor keypad) to move the highlighting bar to your selection. Pressing the enter key selects the highlighted item. Another way to select an item is by pressing the first letter of the command. For instance, to perform server functions from the opening screen, you could do it by pressing s. In the remainder of this discussion of TOPSMENU, we'll just tell you to "point at an item."

As you can see from the screen shown in figure 2.7, Control-Q (entered, of course, by holding down the control key and pressing the

Q) quits TOPSMENU, and the escape key backs you up to the previous menu.

```
  compaq                      Welcome to TOPS                Main

 Main Menu
   Lists all Servers on the network and                 Client Utilities
   the Volumes and Printers available on                Server Utilities
   each.  You may Mount remote Volumes and              Remember
   remote Printers for 'local' use.                     Quit
```

```
MENU INSTRUCTIONS
  Select a Command using the space bar, up arrow, down arrow or first
    letter of the Command.
  <ENTER>- Invoke a selected command.

  <ESCAPE>- Return to the previous window.     <CNTRL-Q>- Exit to DOS.
```

Figure 2.7 The main menu of TOPSMENU.

The remember command on the main TOPSMENU command list is one that is very useful. You can use TOPSMENU to set up TOPS the way you want it to run on your machine, then save this configuration to a DOS batch file, called TOPSTART.BAT. Included in this batch file will be all the necessary TOPS commands to publish and mount your drives or printers. This is very handy: You do not have to run TOPSMENU over and over every time you start the computer. Rather, just type "TOPSTART" at the DOS prompt. TOPS will be started up, and the volumes or printers will be published or mounted in the same manner as with TOPSMENU. Even though we discuss using TOPS from the command line in detail later and describe some useful batch files, you do not really need to understand all the TOPS commands to streamline things. If you know how to publish and mount with TOPSMENU, it can create the batch file for you.

Aside from the two main TOPSMENU windows, additional windows may pop up in the middle of the screen, as you access their functions. We'll talk about these other windows as the need arises.

> ***Note*** When you first run TOPSMENU, if you have not already published your station name to the network, the program will first ask you to enter the name of your station. Station names are detailed in the discussion of the TOPS station command in the next section. Your station name appears in the extreme upper left-hand corner of the TOPSMENU screen.

Client Utilities

As a TOPS client, you are a user of network resources: published printers or volumes. TOPSMENU includes facilities to mount volumes or printers and to determine what resources you currently have mounted. Figure 2.8 shows this screen, which you reach by pointing to "Client Utilities" from the main menu and pressing return. This menu allows you to mount file and printer servers and determine what servers are available on the network. It shows the volumes and printers you currently have mounted.

```
compaq               TOPS : Client Utilities       Client Utilities

Client Utilities Menu                              File Servers
  You can see all the File Servers on the          Volumes Mounted
  net, all the Volumes available on a Server,      Printer Servers
  all the Volumes Mounted from a Server,           Mounted Printers
  and you can Mount an available Volume.           Change Zone
```

```
MENU INSTRUCTIONS
  Select a Command using the space bar, up arrow, down arrow or first
    letter of the Command.
  <ENTER>- Invoke a selected command.

  <ESCAPE>- Return to the previous window.     <CNTRL-Q>- Exit to DOS.
```

Figure 2.8 TOPSMENU client utilities menu.

File Servers The first item on the client utilities menu allows you to determine what file servers are currently available and to mount volumes published by those servers. Point to this item and press the return key. TOPSMENU will look to the network to find all the file servers. When the list of servers appears, you can point to one of the server names and press the return key; a new window will appear showing that server's published volumes. Figure 2.9 shows the display of file servers available, with one of the servers selected and its volumes shown. The window at the left shows the file servers currently available; the one at the right shows the volumes published by that server. The window at the bottom right shows the volumes you have currently mounted from that machine.

```
  compaq                  TOPS : Client Utilities      File Servers

 File Servers                              Volumes Available on: mac plus

                                           Marketing Data
    mac plus                               Word Processing Data

 MENU INSTRUCTIONS
   Select a Volume using the space bar, up arrow, down arrow or first
     letter of the Volume.
   <F1>- See a DOS 'DIR' of the Volume.
   <ENTER>- Mount a Volume.
   <ESCAPE>- Return to the previous window.     <CNTRL-Q>- Exit to DOS.
```

Figure 2.9 The TOPSMENU file server.

To mount a volume, point to that volume and press the return key. A pop-up window (shown in figure 2.10) will appear and ask to which drive letter you wish to mount this networked volume. The choices available depend on your current configuration: what local volumes are available and the maximum number of drives you have specified in TOPSKRNL.DAT. If you are not sure to which letter you should mount this drive letter, press any key, and it will be assigned to the next available volume. The next pop-up window (figure 2.11) will ask whether you want this mounted as a volume you can only look at (read) or one you can change as well (write and read). If the volume you are mounting has had a password assigned to it, TOPSMENU will ask you for the password before mounting the volume.

```
compaq                TOPS : Client Utilities       File Servers

File Servers                          Volumes Available on: mac plus

                                         Marketing Data
   mac plus

                    Choose drive for: Marketing Data

                    TO CHOOSE THE DRIVE TO MOUNT TO:
                    TYPE A LETTER FROM 'A' TO 'J'

                    OR PRESS ANY OTHER KEY FOR THE
                    NEXT AVAILABLE DRIVE.
MENU INSTRUCTIONS



  <ESCAPE>- Return to the previous window.     <CNTRL-Q>- Exit to DOS.
```

Figure 2.10 This dialogue box asks you to which logical volume you should mount the selected networked volume.

If you are not sure about what the networked volume contains, you can easily get a directory of that volume's contents with the F1 key. Point to the volume, press F1; a DOS directory of the volume appears in the top left-hand window (see figures 2.12 and 2.13). If this is a Macintosh volume, press F3 when in this window. A directory showing the Macintosh long file names will appear. Pressing F3 again toggles back to the DOS directory. If there are subdirectories within the directory or volume, you may point to any of those subdirectories, and press F1 again to see their contents.

```
compaq                     TOPS : Client Utilities      File Servers

File Servers                                    Volumes Available on: mac plus

                                                   Marketing Data
    mac plus

                         Choose Access Mode for: Marketing Da

                             Read
                             Write and Read

MENU INSTRUCTIONS
  Select a Mode using the space bar, up arrow, down arrow or first
    letter of the Mode.
  <ENTER>- choose a Mount mode.

  <ESCAPE>- Return to the previous window, aborting the Mount operation.
```

Figure 2.11 This dialogue box asks if you want to mount the volume as one that you can only read from, or one you can change.

```
compaq                     TOPS : Client Utilities      File Servers

DESKTOP                0  9-09-87  7:30p     Volumes Available on: mac plus
DOCUMENT      <DIR>       8-13-87  3:56p
SYSTEM_F      <DIR>       9-14-87 10:50a        Marketing Data
UPDATE_F      <DIR>       8-13-87  3:56p
4 files: Untitled

MENU INSTRUCTIONS
  Select a Directory using the space bar, up arrow, down arrow or first
    letter of the Directory.
  <F1>- See a DOS 'DIR' of the directory.
  <F3>- See the long names of remote files.
  <ESCAPE>- Return to the previous window.     <CNTRL-Q>- Exit to DOS.
```

Figure 2.12 Pressing the F1 with a volume selected causes TOPS to show a DOS directory of that volume.

```
  compaq                     TOPS : Client Utilities      File Servers

DESKTOP      DeskTop                 Volumes Available on: mac plus
DOCUMENT <D> Documents
SYSTEM_F <D> System Folder              Marketing Data
UPDATE_F <D> Update Folder
4 files: Untitled

MENU INSTRUCTIONS
  Select a Directory using the space bar, up arrow, down arrow or first
    letter of the Directory.
  <F1>- See a DOS 'DIR' of the directory.
  <F3>- See the DOS information for remote files.
  <ESCAPE>- Return to the previous window.    <CNTRL-Q>- Exit to DOS.
```

Figure 2.13 Pressing F3 shows the Macintosh long names of a volume.

Note Even while you are using TOPSMENU, others on the network may be publishing volumes. To make sure that you are seeing all the available servers, you can strike the home key on the numeric keypad. TOPSMENU will again query the network to make sure that it is showing you all the servers. Due to the dynamic design of AppleTalk, it is possible that you may sometimes miss a file server that has signed on since you started TOPSMENU. If this appears to be the case, use the home key to search again for the server.

Mounted Volumes To work with any volumes you already have mounted, choose this option from the client utilities menu. TOPSMENU will present you with a screen showing your currently mounted volumes, which logical drives you have mounted them to, which servers have published them, and whether you have mounted them as read only or read/write volumes. Figure 2.14 shows such a display.

To unmount any volume, point to that volume, and strike the delete key.

```
   compaq                 TOPS : Client Utilities      Volumes Mounted

Mounted Volumes are:
Drive Remote Server                       Remote Volume Name                  Mode
  D:  mac plus                            Marketing Data                       R



MENU INSTRUCTIONS
   Select a Volume using the space bar, up arrow, down arrow or first
     letter of the Volume.
   <DELETE>- UnMount a Volume.

   <ESCAPE>- Return to the previous window.     <CNTRL-Q>- Exit to DOS.
```

Figure 2.14 TOPSMENU displays the volumes currently mounted and lets you unmount them.

Printer Servers and Mounted Printers Accessing printer servers works in the same manner as accessing file servers. When you select "Printer Servers" from the client utilities menu, TOPS will display a list of all the print servers on the network. Pointing to one of the server's published printers will mount that printer. TOPSMENU will ask to which logical print device you wish to attach that printer. After mounting a printer, TOPS will redirect any output sent to that logical print device and send it instead to the print server over the network. See "Sharing Printers with TOPS" in part D for more details.

Change Zone This item allows you to choose which AppleTalk zone you wish to work with. A zone is actually a separate AppleTalk network or networks, joined with a bridge device. When you select this option, TOPS will display all the zones currently on the network and allow you to select which zone you wish to make your current zone. The next time you use any of the server options to display file or printer servers, TOPSMENU will display the servers that are part of that zone. You may use resources from several zones by changing zones, then accessing servers from that zone, changing zones again, and repeating the process. For the purposes of publishing from other machines, you will appear to be in your local zone.

Server Utilities

The TOPSMENU server utilities menu allows you to perform functions associated with making your local resources (volumes and printers) available to the network. Figure 2.15 shows the server utilities menu.

```
 compaq                    TOPS : Server Utilities      Server Utilities

Server Utilities Menu                                  Publish a Volume
  Publish a Volume to allow network access.            Volumes Published
  You will be prompted for the full                    File Clients
  pathname.  You must choose an                        Publish a Printer
  alias, mode and password.                            List Published Printer
                                                       My Printer Clients
                                                       Show Name
```

```
MENU INSTRUCTIONS
  Select a Command using the space bar, up arrow, down arrow or first
    letter of the Command.
  <ENTER>- Invoke a selected command.

  <ESCAPE>- Return to the previous window.     <CNTRL-Q>- Exit to DOS.
```

Figure 2.15 TOPSMENU server utilities menu.

Publish a Volume The first item on the menu allows you to make one or more of your subdirectories available to the network. There are several ways to publish a volume using this option. When you select this menu, TOPSMENU will clear the screen and ask you which subdirectory to publish. if you press the letter of the disk (i.e., C), TOPSMENU will supply the necessary :\ portion of the volume name and the full path name of the directory you wish to publish.

Alternatively, if you press the F1 key, TOPSMENU will display a list of all the subdirectories in your root directory (see figure 2.16). If there are more directories than will fit in the window, pressing the "PgDn" displays the remainder of the list. Press F1 if you wish to view the subdirectories that are located within a particular directory. These subdirectories are displayed in the right-hand portion of the top window. You can point to one of them and publish it.

```
  compaq                 TOPS : Server Utilities      Publish a Volume

 Publish a Volume for Network Access   TOPS          <DIR>      8-29-87  2:54p
                                       DATA          <DIR>      8-30-87 10:28a
      Path:                            BAT           <DIR>      8-30-87 10:34a
                                       WP            <DIR>      8-30-87 10:40a
                                       TOPS2         <DIR>      9-07-87  8:45p
                                       DOS           <DIR>      9-01-87  9:18a
                                       LIB           <DIR>      9-08-87  8:36a
                                       CDROM         <DIR>      9-11-87 10:09a
                                       BKSHLF        <DIR>      9-11-87 10:12a
                                       LOGO          <DIR>      9-11-87  6:55p
                                       HOTSHOT       <DIR>      9-13-87  1:05p
                                       C:                                 MORE

 MENU INSTRUCTIONS
   Select a Directory using the space bar, up arrow, down arrow or first
     letter of the Directory.
   <F1>- List the subdirectories of a directory.
   <ENTER>- Publish a directory.
   <ESCAPE>- Return to the previous window.     <CNTRL-Q>- Exit to DOS.
   <PGDN> - next page
```

Figure 2.16 The "Publish a Volume" option.

Publishing a directory requires that you also specify the alias by which the volume is to be known over the network. Since this is how others will access the volume choose a name that describes its contents.

> ***Note*** TOPS alias names are independent of the DOS file or directory name restrictions—they can be as long as sixteen characters—so you have some flexibility in naming them.

You will also need to optionally specify a password for the volume and whether you want to allow others to write to the volume or be able to only read from it. If you do not enter an "alias" for the subdirectory, the name of that subdirectory will automatically become its TOPS alias. To make it one they can write to, set its mode to "RW," otherwise set it to "R." Figure 2.17 displays these options for a volume about to be published.

```
  compaq                  TOPS : Server Utilities      Publish a Volume

 Publish a Volume for Network Access

     Path:     c:\data
     Alias:    sales data
     Password:
     Mode:     RW

 MENU INSTRUCTIONS
   Type each item as requested and press <ENTER>:
     1-16 character alias by which you want this Volume known on the network.
     0-8 character password with which you wish to secure access.
     1 or 2 character mode: R- Read Only; RW- Read/Write.
   <ESCAPE>- Start over.                       <CNTRL-Q>- Exit to DOS.
```

Figure 2.17 The directory C:\DATA is about to be published with the network name of "Sales Data" in a read/write mode, with no password.

Volumes Published This item shows you a list of all items published and allows you to work with them. Choosing it summons the screen shown in figure 2.18.

Pointing to a volume and pressing the enter key displays the network names of all who are currently using that volume. That is, those who have that volume mounted on their machine. If there are no active clients, you may alter the manner in which the volume is published. You can change the network name or alias of the volume (with F2), its password (with F10), and its mode (i.e., read only or read /write—with F6), or unpublish it using the delete key. If you attempt to perform any of these operations while a volume is in use by other clients, TOPSMENU will give you an error message in the bottom window and will not permit you to continue the operation.

```
   compaq                  TOPS : Server Utilities      Volumes Published

Volumes Published from this Server:
Alias             Path                                            Password Mode
 data             C:\DATA                                             No    RW

 MENU INSTRUCTIONS
   Select a Volume using the space bar, up arrow, down arrow or first
     letter of the Volume.
   <ENTER>- List Clients using this Volume.      <DELETE>- UnPublish a Volume.
   <F10>- Change the password.                   <F2>- Change the Alias.
   <ESCAPE>- Return to the previous window.      <F6>- Change the mode.
```

Figure 2.18 The "Volumes Published" option.

File Clients Whereas the "Volumes Published" item shows all the volumes you have published and allows you to see which clients are using those volumes, this option works the other way around. It shows your active clients and allows you to see what volumes they are using. Figure 2.19 illustrates this display. The users are shown in the upper left-hand window, and the volume in use by the selected user is shown in the lower right-hand corner. One of the primary uses of this option is to log out a crashed client. A crashed client is one whose machine has been turned off, or for whatever other reason is no longer actually using the volume, but who has not officially notified your server of this fact. Sometimes, software can freeze up a computer and cause it to need to be reset or shut off. In this case, your TOPS probably still thinks the client is using the volume and will not allow you to unpublish it. Using the delete key when you have selected such a client logs them off, and you can then unpublish the volume or change its mode of publication.

> ***Note*** It is possible to log out a client that is actually still active. Do not use this option unless you are certain the client is off the network.

```
compaq                 TOPS : Server Utilities      File Clients

Clients using this File Server are:

    mac plus

                                          Volumes used by:   mac plus

                                           data

MENU INSTRUCTIONS

  <ESCAPE>- Return to the previous window.    <CNTRL-Q>- Exit to DOS.
```

Figure 2.19 A listing of the current users of your published volumes.

Publish a Printer Publishing a printer works in the same manner as publishing a volume. You choose which printer you are to publish, provide a network name for it, and tell TOPSMENU where to keep the temporary spool files it creates. See figure 2.20. TOPS publishes printers according to which of the logical ports they are connected to. The options are the parallel ports: LPT1, LPT2, and LPT3. It might seem that you cannot, then, publish printers connected to a serial port in this manner. However, printers connected to a serial port can be referred to by the LPT1 names by using the DOS mode command to redirect serial output to a logical LPT device. In this manner, serial printers can be shared. As with published volumes, you should choose a descriptive network alias for your printer. All printer names on the network should be unique and should leave no doubt in the client's mind as to which printer they are using. The spool directory item refers to the directory on your disk where TOPS should keep the temporary files it creates when it receives output from a client and before it sends the information to the printer. See "Sharing Printers with TOPS" in part D for more details.

```
  compaq                  TOPS : Server Utilities      Publish a Printer

Publish a Printer

     Printer:         LPT1 (PRN)
     Alias:           epson fx80
     Password:
     Spool directory: c:\tops
```

```
MENU INSTRUCTIONS
  Type each item as requested and press <ENTER>:
    Full path including drive specification.

  <F1>- List directories on a local drive.
  <ESCAPE>- Return to the previous window.     <CNTRL-Q>- Exit to DOS.
```

Figure 2.20 The printer connected to LPT1 is about to be published with the name "epson fx80." The spool files will be kept in the TOPS directory.

Note You will not be able to publish a printer if you have not loaded TOPSPRTR into memory. In fact, if TOPSPRTR has not been loaded, TOPSMENU will not even display printer-related options in the server menu.

Note All clients of each printer must have their software configured correctly to work with your printer.

List Published Printer This shows you the printer(s) you have published and allows you to examine the print queue (list of files waiting to be printed) and the current users of that printer. If there are no active users of the printer or if there are no files waiting to be printed, you may unpublish the printer.

My Printer Clients This allows you to find out who is using your printer and log out crashed clients in the same manner crashed file clients are logged out.

Note It is possible to remove active users of your printer in this manner. Do not do this when they are using the printer. It could result in loss of printing or, even worse, loss of work for that user.

Hide Name/Show Name The last item in the server utilities menu allows you to hide your name so that others on the network will not see your published resources. You may, for example, need to perform some intensive operation on your computer and do not want to inhibit the performance of others who are using your resources or do not want to have your performance inhibited by others using your machine. This option takes your server off the network, so that others will not see it. Selecting it again re-registers your server on the network.

> ***Note*** If other users have previously mounted one of your published resources, they will still be able to use it. This option only prevents new users from accessing your resources.

You can tell visually whether your name is shown to others on the network from anywhere within TOPSMENU. In the upper left-hand corner of the screen is your station name. (See any of the previous screens in this chapter.) If your name is currently shown on the network, your station name will be highlighted with bold letters. If it is not currently shown on the network, it will appear as standard text.

Remember

This option, accessed on the main TOPSMENU screen, allows you to save your current TOPS configuration in a DOS batch file so that you can recreate the configuration without having to start from scratch with TOPSMENU. It creates a batch file called "TOPSTART.BAT" in the root directory and includes all the necessary commands to load TOPS and publish and mount volumes. As we will discuss later, you can create your batch files or modify this one to speed up the process of starting TOPS on your machine, and make sure standard volumes are published or mounted.

That concludes our discussion of TOPSMENU; you can perform all TOPS functions from this program. In the next section, we'll discuss using all the TOPS functions from the DOS command line.

Working from the DOS Command Line

While TOPSMENU provides a menu-driven, easy-to-use interface to the TOPS functions, all the TOPS functions can also be accessed in a more traditional DOS manner—by typing commands from the DOS prompt. For those of us who are very familiar with DOS, this is almost as easy—and usually quicker—than running TOPSMENU. Additionally, as we will see later, this facility of TOPS allows us to automate many TOPS functions by placing commonly used commands in batch files.

The TOPS program that we use to type in these commands is TOPS.EXE. This program is located in the TOPS directory (or wherever you installed TOPS). As already discussed, if you put this directory into your DOS search path, the command will be available to you no matter which directory is your currently active directory.

You type TOPS commands by typing "TOPS" followed by the name of the command. TOPS.EXE executes, then interprets the next word you have typed on the command line. If you make a typing mistake or type a command incorrectly, TOPS then displays a message and tries to give you help about the particular command. If you type "TOPS" alone on the line, it displays the TOPS version and serial numbers. Typing "TOPS HELP" displays a list of the valid commands.

Most of the TOPS commands are explained very well in the TOPS PC Owner's Manual. Here we'll discuss some particular applications of those commands, with some tips on their uses, leading to using them in DOS batch files.

TOPS Station

This command is used to display, set, or change the status of your station name. Station names are used by TOPS to identify computers on the network (other networking systems use numbers). The station name is normally specified in the file TOPSKRNL.DAT, which TOPSKRNL.EXE reads when it is executing. If you have not modified TOPSKRNL.DAT to include your station name, you must use this command (or TOPSMENU) to set the name of your station. To set or change the name of your station, type "TOPS STATION" followed by a space and the name. You can include spaces in the station name, as long as you then enclose the entire name in double quotes (").

Note If you change the name of your station, you should tell everyone else on the network that you have changed it. Others might be using batch files to mount volumes that you publish. If you change your station name, they will need to change their batch files. Since TOPS looks for network volumes by station name, it will not find them if the name has changed.

Typing "TOPS STATION OFF" turns off the display of your network names to others. Those who have already mounted any disks or printers you have published will still be able to access those volumes, but others will no longer be able to see your station name on the network. Typing "TOPS STATION ON" activates the station name when it has been turned off. This command is equivalent to "hide /show names" in TOPSMENU.

TOPS Mount

This is the command you use to mount networked drives or printers. (For more on sharing printers among PCs, see part D. In this section, we'll talk only about sharing disks.) The syntax of the mount command follows:

```
TOPS MOUNT D: TO SERVER VOLUME [/Z ZONE] [/R | /RW] [/P [PASSWORD]]
```

In this line, "D:" refers to the drive designation by which you wish to refer to the networked volume. For example, if you wish to refer to this mounted drive as drive E: then you would include this designation in this place. "SERVER" refers to the name of the server, and "VOLUME," to the name of the volume that server has published. One cause for confusion is for servers that have spaces in their names. Since TOPS figures out the meaning of this line based on the spaces in it, you need to surround server names that have spaces with double quotes.

The "/R" and "/RW" parameters control how TOPS is to try to mount the volume. "/R" causes TOPS to make the volume a read only volume: You will not be able to save files to that volume. "/RW" causes TOPS to attempt to mount the volume as one you can write to as well as read from. However, if the volume has been published as a read only volume, you will not be able to mount it as a read/write volume.

When you try to save a file to a volume that has been mounted as read only, DOS will return a file creation error message, exactly as if the volume was a floppy disk with the write protect notch covered.

"/Z" sets the zone for the server, and is only necessary if the server is not in the local zone or the zone that has been set with the "TOPS ZONE" command.

"/P[PASSWORD]" allows you to specify, from the command line, the password to get access to the volume. If you do not specify the password from the command line, TOPS will prompt you for it.

Automated forms of this command are implemented in batch files, which are discussed later.

> ***Note*** Do not include the password for any volume in a batch file. There is no simple way to encrypt batch files so that others cannot use them. The password can be saved in an encrypted form in a batch file with the TOPS remember command. This includes the encrypted password in the batch file, with a "/PE" preceding it.

To mount a networked printer, you need to tell TOPS several things: how you will be referring to the printer, the network name of the printer, and the name of the server. The syntax of this command is:

```
TOPS MOUNT LPT#: TO SERVER PRINTER [/Z ZONE] [/P [PASSWORD]]
```

As already discussed, you need to tell TOPS what output to redirect. If, for example, you want TOPS to send any output that normally goes to your first parallel printer to the network instead, you would put "LPT1:" in place of "LPT#:." All other TOPS MOUNT commands work in the same manner.

TOPS Publish

The syntax for the publish command is;

```
TOPS PUBLISH D:PATH AS VOLUME [/R | /RW] [/P [PASSWORD]] [/X]
```

"D:PATH" refers to the exact description of the directory on your hard disk. The full path name (including the drive designator) must be included for TOPS to publish the directory correctly. If you are not familiar with DOS path names, consult your DOS Owner's Manual. You must understand paths and the DOS directory structure to use your computer and TOPS efficiently.

"VOLUME" refers to the name or alias by which you wish other TOPS users to use to refer to the directory. This is the name that TOPS will display when the volume is listed in other TOPS programs (i.e., when other PC users use TOPSMENU or TOPS DIR to discover what volumes are available on the network). The network is easiest to use when published volumes have names that are clearly descriptive of their purpose or their contents: "John's Data" as a name might mean something to John, but it does not necessarily tell other users what is in the directory, but "Sales Figures" does.

> ***Note*** When assigning a network name, if you want to include spaces in the name (as in the examples above), you must surround the name with spaces. Otherwise, TOPS cannot correctly decipher what you mean by the line and will return an error message.

The "/R" and "/RW" parameters instruct TOPS as to how to publish the directories and whether you want to allow others to be able to read or read and write to the volume. If you publish a volume with the /R command, they will be able to look only at files on the volume or copy them to other volumes. The disk will appear to DOS as a locked disk, and others will not be able to save files onto that disk. The effect is exactly the same as placing a write protect tab over the notch of a 5-1/4-in. floppy disk. Using the /RW command allows others to change files on the disk or save files onto that disk. If you do not specify one of these parameters, TOPS publishes the volume as a read/write volume.

The /P command allows you to set a password that others must type to mount the volume. If you do not specify the password on the command line, TOPS will prompt you for it. While you can—and should—put most of these items into a batch file to help you access them easily, you should not put your password into the batch file. There is no practical way to protect batch files from prying eyes: The DOS type command can display it to anybody who can get to your keyboard. (Unless you create the batch file with "TOPS remember.")

The /X command is used to stop TOPS from updating the contents of the directory. As we'll discuss later, when you publish a directory, TOPS creates a hidden or invisible file in the folder, called

"XDIRSTAT.TPS." This file contains information that is necessary for TOPS to allow the volume to be shared. Normally, TOPS updates that file whenever you publish a volume. In normal practice, you should not publish with the "/X" option, unless you are sure that there have been no changes to the files in the volume since it was last published. Otherwise, it is possible that you could lose data in the directory.

The syntax for publishing a printer from the command line is not too different from publishing a volume. In the publish line, specify the logical address of the printer, the printer's network name, and the directory to use for spool files. For example, if you want to publish the Epson LQ800 that is attached to your parallel port, your publish command might look like this:

```
C:> TOPS PUBLISH LPT1: AS "EPSON LQ800" USING C:\TOPS /E /P
```

In this line, "USING C:\TOPS" tells TOPS to use this directory to hold the temporary spool files. The "/E" tells it to delete any old spool files that might have remained from prior sessions (otherwise they will be printed). And the "/P" causes TOPS to pause and ask for the password that users need to enter before they can use the printer.

> ***Note*** Do not include the password in the batch file. That file can be displayed by anyone who can get to your machine (unless you use the "TOPS remember" option to do so). Enter it every time yourself, to prevent unauthorized access to your printer.

TOPS Pstat and TOPS Cstat

These two TOPS commands allow you to determine the status of your computer on the network regarding server or client activities.

TOPS Pstat This command stands for publisher status, and shows information about the volumes and printers you have published. There are several parameters you can use with this command that control the type of information that is returned to you.

/V With this parameter, TOPS shows the status of the volumes you have published. Adding space, followed by a volume name, displays the information about that particular volume.

/PR This parameter displays information about printers. As with the /V option, you can ask for information about a specific printer by naming the printer on the same line.

/C This parameter shows all the clients now using your published resources. Adding a client's name after this parameter shows the status of the specified client.

/D and /F These two commands show the open volumes or directories (/D) and the open files (/F) on the volumes you have published.

Before shutting TOPS down, check your server status. If there are active users, you should wait until they have logged off your published resources or notify them that you need to shut down and ask them to log off.

TOPS Cstat This command stands for client status. It displays the list of network resources that you currently have mounted. Always check this list before shutting off your machine; unmount any of these resources to make sure that the servers know that you are no longer using their resources. By following the command with a drive designation, you can ask for information about your open files on a specific drive. By supplying the logical printer address (LPT#) on the command line, you will be informed about your use of the network printer, including which printer jobs you currently have pending.

TOPS Unload

As already discussed, most of the functions of TOPS are handled by several programs that remain in your computer's memory while TOPS is active. The unload command allows you to remove TOPS memory-resident modules from memory. This allows you to run programs that require more memory than TOPS leaves in the computer.

The unload command allows you to remove from memory TOPSPRTR, TOPSKRNL, or TOPSTALK, or all the installed TOPS modules. To use it, enter at the command line TOPS UNLOAD followed by a space and the name of the module you wish to unload. If you want to unload all the TOPS modules, type "TOPS UNLOAD/A ." The /A parameter signifies that all the TOPS modules should be removed in memory. Using this assures that the modules will be removed in the correct order.

Using this command means that TOPS will no longer be executing on your machine, of course, so some care must be taken before executing it. Here are some warnings.

1. Unmount all mounted volumes and printers. TOPS will return an error if you try to unload any of the TOPS modules while they are actually in use.
2. Unpublish all volumes and printers. TOPS will return an error if you try to unload it when you have volumes published. It is very rude to unpublish volumes or printers while others are using them; it can result in loss of work on the part of others on the network. If you must unpublish TOPS volumes or printers, notify those who are using them, so they can act accordingly.
3. Unload the modules in the reverse order they were loaded. Typically, TOPS memory-resident modules are loaded in this order: TOPSTALK, TOPSKRNL, and then TOPSPRTR. If you are going to unload all these modules, then use the /A parameter to this command, which unloads them all. If, however, you have loaded the files in this order, remove them in the reverse order.

   Say you want to unload TOPSKRNL from memory. TOPS will allow you to do this, but you won't be able to reap any benefits from it: TOPSPRTR is in the way of other programs accessing that memory. What you should do, rather, is first unload TOPSPRTR, then TOPSKRNL. Then reexecute TOPSPRTR, so that it takes up the same memory addresses that TOPSKRNL occupied. This allows other programs to use the memory.
4. Be careful using the unload command when other memory-resident programs are active in the computer. Programs, such as Ready or SideKick, work in much the same manner as the TOPS modules. When you execute them, they remain in the machine's memory and can be called up at any time. If you have loaded any of these programs after you have loaded any of the TOPS modules, you won't gain any memory by unloading TOPS. Before you unload TOPS, do whatever is necessary in these programs to unload them too. Then unload the TOPS modules. It should free up the memory.

Other TOPS Commands

Tcopy Tcopy performs the same functions as the DOS copy command, with added functions that are suitable for the TOPS environment. This command copies, along with the actual file as perceived by the DOS, the hidden portions of the file that are part of Macintosh files, such as the resource forks. It will also update the XDIRSTAT.TPS file, to make sure that TOPS is aware of the new files in the directory.

The syntax of this command is the same as for the DOS copy command, with some significant enhancements.

1. You do not need to specify a file name, or use the *.* designator to copy all the files in a subdirectory. Typing "TCOPY" followed by a source directory name and a destination directory will copy all the files in the source directory. For example, to copy all the files in the directory "DATA" to a network disk (mounted as drive D:), the command would look like this: "TCOPY C:\DATA D:."
2. There is an additional parameter—/S—that allows you to copy all the files that are in directories contained in the directory you are copying. In this example, if we wanted to copy the data that is contained in directories which are in turn contained in the directory C:\DATA, the command would look like "TCOPY C:\DATA D: /S." If the same directories do not exist in drive D: as in drive C:, Tcopy will ask you if you want to create them.
3. Tcopy can create subdirectories on the destination disk to match the directories in the source disk. To use the previous example, if the same directories did not exist on drive D: as exist in "C:\DATA", we would add the /C parameter.
4. Tcopy can prompt before copying each file. This can be handy when you do not want to copy all the files in a certain directory. Instead of having to execute the copy command many times, once for each file, execute Tcopy, and add the parameter /P after telling Tcopy what to copy. Tcopy will pause and ask you whether it should copy each file.
5. Tcopy can work with files that contain the DOS reserved characters ? and *. Even though these characters do not normally appear as part of file names in files created in DOS, they can, of course, be part of Macintosh file names. The DOS copy command cannot work with them. By including the /W parameter as part of the copy command, Tcopy will treat these characters as part

of the file name and will not interpret them according to DOS conventions.

6. Tcopy can filter unwanted, usually nontext, characters from a file. As we discussed in chapter 1, text files on the Macintosh and the PC follow some different conventions. Using the /F option causes Tcopy to remove most control characters from a file, as well as the linefeed characters DOS appends (along with a carriage return) to the end of every line.
7. Tcopy also features two commands that can be very useful to back up files from a networked disk. The /U command copies only files that have been updated or are newly created. It uses the archive bit that is part of DOS files to determine which files to copy. This is very handy for backing up your files from a networked disk to a local floppy: Use the /U command to add to your backup floppy files that have been changed since you last used the command. You can also use the /M flag, followed by a date (in the form month/day/year) to copy only files that have been created or modified since a specific date. This, too, is very handy for backing up your files.
8. Always use Tcopy if Macintosh files are involved. It will see to the dirty work of copying the hidden resource forks of Macintosh files and will see to it that the DeskTop file is changed.
9. Finally, Tcopy can work in the same way as the DOS copy command. Adding the parameter /D causes TOPS to copy only information recognized by DOS. It does not execute XSYNC, nor does TOPSKRNL need to be loaded.

For Tcopy to work, TOPSKRNL must reside in memory (unless you use the /D parameter). Before executing, Tcopy also executes XSYNC on a disk, to make sure the TOPS extended directory has been updated on the disk.

XDIR and XSYNC

To supply the extended functionality of the networking environment, TOPS must record some information about each file that is not normally part of the DOS file itself. This might include the Macintosh resource forks of files (see part D), the long Macintosh file name, and more. To do this, TOPS creates a file on disk called "XDIRSTAT.TPS." This is a hidden file that contains all the information TOPS needs to manipulate non-DOS files and to allow sharing of directories between machines. When a directory is published, TOPS creates this file if one does not already exist and updates it if it does.

Even if a directory is not currently published, you can force TOPS to update this file with XSYNC.

XDIR allows you to view the TOPS extended directory. This utility displays the NetName, file Type, and RealName of files in a specified directory. "NetName" refers to how the file is known over the network. If the network files are DOS files, this is actually the name of the file. For Macintosh files, it represents a DOS-legal version of the Macintosh file name. For DOS files, all will have a type of data (with the exception of directories, which have a type of dir). Macintosh files, though, can have two types (and usually do): data and resource. As discussed in part D, these are actually part of the same file, the hidden resource file containing information that is inappropriate in the DOS environment. XDIR displays both the visible and hidden portions of the file. The RealName of the file will typically be the Macintosh file name, which can be much longer than the DOS version and can contain characters illegal in DOS. In cases where you are viewing directories that are only shared in the DOS environment or are not shared at all, RealName will be the same as the NetName.

Before you do any work with files in a directory that has been published or is mounted from another station (particularly if it is mounted from a Mac), use XDIR to examine the contents of the directory.

Batch Files Are Your Friends

One of the nicest features of DOS is its batch-processing ability. Simply put, a batch file is an ASCII file that contains a series of DOS commands. Any command that can be executed from the keyboard can be placed in a batch file. Instead of having to enter the command, or commands, separately from the keyboard, you simply type the name of the batch file, which must have a .BAT extension. DOS will execute those commands exactly as if you had typed them. Batch files are the perfect tools for those of us with lazy or inaccurate fingers. They can also shield us from some of the complexities of DOS and allow power users to tailor DOS to be more friendly to neophytes.

While TOPSMENU is a handy and well-written program, many of the day-to-day functions of TOPS can be handled from the DOS command line and, hence, through batch files. Certain TOPS functions can be automated, through the use of the AUTOEXEC.BAT file, to, say, publish a certain set of directories automatically on startup.

A Set of TOPS Batch Files

In the next few pages, we'll discuss some things that can be done with TOPS batch files and present a set of files that are handy in the day-to-day use of TOPS. Use these sample batch files as launching points for developing your own batches more suited to the way you work.

Creating Batch Files Creating a batch file is simple. From the DOS command line, type:

```
COPY CON: FILENAME.BAT
```

That is, you are copying from the console (keyboard) to a file. Replace FILENAME with the name of the batch file you are creating. Type each line individually, and press the enter key at the end of the line. When you have typed the last line, hold down the control key and press Z. The file will be created.

You can also use a word processor or other text editor (i.e., the DOS utility Edlin or a program such as SideKick) to create the file. The only restriction is that the editor must be able to create an ASCII text file. Most word processors can do this; it might be called a "non-document," "ASCII," or "text-only" mode.

Finally, TOPS itself—from either the command line or from within TOPSMENU—can create a batch file to start TOPS and publish or mount a set of volumes. After you have published and mounted the standard set of volumes, from within TOPSMENU, select "Remember" from the main menu. TOPS will create a batch file called "TOPSTART.BAT." This batch file will include the commands to start TOPS and to mount and publish the volumes you currently have mounted or published and will be placed in your root directory.

From the DOS command line, you can also tell TOPS to create such a batch file. This gives you a little more control over where TOPS will save your batch file. Enter this at the command line:

```
TOPS REMEMBER /F FILENAME
```

Instead of typing "FILENAME," type the name of your batch file. The file name can consist of a path to a directory in which you want to save the batch file. For example, to save a batch file called "TOPSTART.BAT" in your directory BAT, which holds your batch files, type:

```
TOPS REMEMBER /F C:\BAT\TOPSTART.BAT
```

The resulting batch file, whether created from TOPSMENU or from the command line, can be edited with any standard editor (as long as it can read and write plain ASCII files) and can be used the same way as any standard batch file.

An Autoexec File Even if you are not overly familiar with DOS, you probably are aware of its autoexec facility. Put simply, when DOS starts, it looks on its boot disk for the presence of a batch file called "AUTOEXEC.BAT." This is a batch file that is *exec*uted *auto*matically at boot.

The first lines of the batch file in listing 2.1 are pretty standard. "ECHO OFF" stops the commands in the batch file from being displayed on the screen. It makes the file proceed a little faster and allows you more control over what the screen looks like. "PROMPT PG" changes the standard DOS prompt from one that shows what drive you are logged into one that also shows the current directory.

Listing 2.1 Sample AUTOEXEC.BAT file for starting TOPS automatically when starting the computer.

AUTOEXEC.BAT

```
ECHO OFF
PROMPT $P$
PATH = C:\;C:\DOS;C:\BAT;C:\TOPS
REM =============
REM start TOPS?
CLS
BEEP
ASK Start TOPS?
IF ERRORLEVEL 1 GOTO TOPSPRINT
:STARTOPS
```

continued

Listing 2.1 (cont.)

```
TOPSTALK
TOPSKRNL
REM ==============
REM publish TOPS
ASK Publish C:\DATA?
IF ERRORLEVEL 1 GOTO TOPSPRINT
TOPS /Q PUBLISH C:\DATA AS C:\DATA /RW
PROMPT DIRECTORIES PUBLISHED!$_$P$Q
REM ==============
:TOPSPRINT
REM topsprint?
CLS
BEEP
ASK Start TopsPrint?
IF ERRORLEVEL 1 GOTO NOTOPS
TOPSPRINT
ECHO Print the WordPerfect PostScript header file?
ASK
IF ERRORLEVEL 1 GOTO NOTOPS
PRINT C:\WP\LASERWRT.PS
REM ==============
REM startup without TOPS
:NOTOPS
CLS
```

The path statement tells DOS where it should look for commands; instead of having to be logged into, say, the TOPS directory, its inclusion in the path statement tells DOS to look there automatically. On my hard disk, and probably yours, "C:\DOS" contains the various files and commands that are part of DOS itself. I keep all my batch files in a separate subdirectory to help keep the disk uncluttered. So, this line tells DOS to first look for any commands in the root directory of the hard disk, then in the DOS directory, followed by the BAT directory and the TOPS directory. The path command can contain many directories. The limit is actually on the length of the string itself: It can contain 115 characters. So, if you have long directory names you can insert fewer of them into the path than if you have short names. Note, however, that too many directories in the path statement can make DOS sluggish. There are other consequences as well.

The rem statement can be used to insert remarks that explain what different portions of a batch file do, or it can be used, as below, to set off different portions, making it easier to read. The "CLS" line simply clears the screen to present a clearer focus for the question. Beep is a program available with the excellent Norton Utilities. As its name implies, it simply beeps the PC's speaker. Ask is a public domain program (©Copyright T.A. Davis, 1983). It allows a batch file to stop and ask the user a question—something not provided in standard DOS. It is available on many bulletin boards or with the excellent book *The Fully Powered PC*. A similar program is also part of the Advanced Norton Utilities. If the user presses an N, Ask sets a DOS variable called "ERRORLEVEL" to 1.

In this case, if the user presses N, the batch file branches to a labeled area called "TOPSPRINT" to find out whether the user wants to use TOPS to print to the LaserWriter.

"TOPSTALK" provides the low-level support for TOPS functions. The user has no interaction with this program, but it is necessary for other TOPS programs to run. Additionally, other network-related software may access TOPS functions without the necessity of the full TOPS packages, if this module has been loaded.

"TOPSKRNL" actually starts TOPS. If you execute this program from a directory other than your TOPS directory, it will not find the TOPSKRNL.DAT file, and the standard parameters in that file will not be applied. This file must be in the directory to which you are currently logged. Such things as your station name and the number of volumes you have published are contained in this file. To remedy this situation, you might insert a line "CD \ TOPS" into your batch file before executing TOPSKRNL. You will then need to insert a line after this to change back to your original directory after executing TOPSKRNL. Another, messier, way of remedying this situation is to place TOPSKRNL.DAT in your root directory.

This next part of the batch file queries the user as to whether a standard directory or set of directories should be published. In the "TOPS /Q PUBLISH" statement, a directory is published. The "/Q" in the statement prevents the copyright banner from being printed to the screen. The /RW parameter allows others to write to the directory as well as read from it. Another parameter, /P, allows protection from unauthorized access.

Note If you insert the password into your batch file, it will be possible for others to read the password by looking at your batch file (although they cannot look at it over TOPS if you do not publish the directory holding the batch file).

The prompt line again alters the standard DOS prompt to remind you that you have published volumes. After executing this command, your DOS prompt will look like this:

```
DIRECTORIES PUBLISHED!
C:\>
```

Note Since the C:\TOPS directory is now in the DOS search path, the TOPS and TOPSKRNL commands can be executed without having to specify their subdirectories. In this case, I am only publishing one directory, the one I normally use to hold my data files. Subsequent lines could publish other directories.

Again, if the user answers N, flow of the batch file is routed to the TOPSPRINT area.

This part of the batch file queries the user as to whether TOPS NetPrint should be started.

After starting TOPS NetPrint, the batch file asks if it should print to the laser printer a header file needed by WordPerfect. WordPerfect needs to have this PostScript file present in the laser printer if it is to work correctly. By printing it just once to the printer, WordPerfect will not have to download the file with each print job. This simple operation saves time. The file usually needs only to be downloaded once to the printer after it has been turned on. While the procedure as shown here is specific to WordPerfect, the same can be made to work with the header files used by other programs. Since this header file is written in PostScript, TOPS NetPrint does not need to translate it into PostScript. (This would print the contents of the header file instead of causing the laser printer to interpret them.)

If you are using TOPS NetPrint and are using the downloadable font of the IBM character set, which is included with TOPS NetPrint, insert the commands for downloading that font into the AUTOEXEC.BAT file along with any other header files. This assures that the font will be available to TOPS NetPrint for printing screen dumps. The same applies to any other downloadable fonts that you might want to use. See appendix C for more details on downloadable fonts and IBM character set fonts.

"NOTOPS" simply presents an exit point for the batch file. You can put here calls to any other routines or programs that you normally use.

More Handy Batch Files

Many people, especially those who are more accustomed to using a Macintosh than a PC, have complained about the complexity of TOPS as executed from the DOS command line. This is part of the very nature of DOS and is one of the reasons for the success of the Macintosh interface. Much of the syntax of TOPS commands is hard to remember, and the TOPS help facility does not offer much assistance to the neophyte. It is easy to mistype one parameter in a long TOPS statement, only to have that statement rejected. Batch files can help a great deal in simplifying using TOPS from the command line.

TOPSTART.BAT TOPSMENU can be used to create a standard batch file. To do this, use TOPSMENU to publish and mount a standard working set. Then select "Remember" on the TOPSMENU main menu; TOPSMENU will create a batch file called "TOPSTART.BAT." This is a painless way to go if you do not want to worry about the intricacies of creating batch files or testing them to make sure they work. TOPSMENU creates a batch file and places it in the root directory. So, if you want to create a batch file with TOPSMENU, make sure you are either in the TOPS directory or in your batch directory when you invoke the program.

PUBLISH.BAT The TOPS publish statement is used frequently and can be especially tricky. Here is the syntax for the statement, in traditional DOS notation:

```
TOPS PUBLISH D:PATH AS VOLUME [/R | /RW] [/P [PASSWORD]]
```

That's not hard to understand once you know DOS and TOPS. However, it is not easy to type it correctly all the time. A simple batch file called "PUBLISH.BAT" can help. It simply lists the previous statement and will publish DOS subdirectories to TOPS. To invoke it, simply type at the DOS command line:

```
PUBLISH C:\DATA C:\DATA
```

This batch file works because DOS can pass command line parameters to a batch file. The "%1" in listing 2.2 refers to the first item typed on the command line after the batch file name. It is a place holder in the batch file for a value that will be inserted when the batch file passes the command to TOPS.EXE. In this case, the first parameter is the name of the file to pass. The second parameter is the alias by which the directory will be known over TOPS.

Listing 2.2 Modified PUBLISH.BAT for publishing multiple directories with one command.

```
PUBLISH2.BAT

ECHO OFF
CLS
:PUBLISH
TOPS /Q PUBLISH %1 AS %1 /RW
SHIFT
IF NOT TEMP == TEMP%1 GOTO PUBLISH
```

If this batch file is placed in your root directory, batch file directory, or any other directory that is searched automatically, then you do not have to worry about what directory you are logged into to execute it. See the previous discussion of path statement in "Autoexec File." It is important, however, to make sure you insert the full path name of the directory to be published on the command line.

> ***Note*** There is no ECHO OFF statement at the beginning of this batch file. It is handy, with certain batch files, to be able to see the actual command that is being executed.

A More Complex PUBLISH.BAT If you want to publish several directories with one command, PUBLISH.BAT can be modified, as in listing 2.2.

To call up this batch file, type:

```
PUBLISH2 DIRECTORY1 DIRECTORY2 . . .
```

DIRECTORY1 and DIRECTORY2 are directories to be published. You can type as many as ten directory names on this line.

How does this batch file work? Note that we have a labeled area called "PUBLISH." When the batch file is executing, it first passes through the file and executes the TOPS publish command, using parameter %1. This publishes the directory. After this line is the line "SHIFT." This command moves the parameters you entered on the command line; it shifts them to the lower numbers. For purposes of executing the batch file, what was the second item on the command line becomes the first item. After the next line, which tests to see if there are any more variables, the batch file loops back to the publish line, and the second item on the command line is published.

The "IF NOT . . . " line refers to a variable, called "TEMP." The DOS Owner's Manual (and most books I have read about DOS) are cryptic about using this kind of looping. Basically, TEMP is a dummy variable that contains nothing. If the value of TEMP appended to %1 is not equal to TEMP, the flow of control to the batch file is sent back to ":PUBLISH." If TEMP, which contains nothing, is equal to TEMP appended to %1, which happens when all the variables have been shifted through, then the batch file executes. Is that clear? I thought not.

One problem with using this batch file is that all the published volumes have the same aliases as their DOS path names. For example, if I publish C:\DATA and C:\STUFF with this batch file, that is the name they will be called by the other computers on the network.

> ***Note*** It is possible that there can be certain conflicts in publishing a file with a colon (:) in the name. On the PC, the colon is part of the designation for a disk drive (i.e., A: and C:). In Macintosh file names, the colon is reserved to separate the disk name from the file/folder name. As a matter of fact, the Finder will not even let you create a file name with a colon as part of it. (Try to type a file name with a colon on the Macintosh. When you type the colon, Finder substitutes a dash [–] for the colon. This works in Finder 5.3 and higher and may be the same in earlier versions.) I have not yet encountered any problems with this kind of naming, but it is possible they could occur. Likely symptoms would be the inability of programs to access a drive with this designation.

MOUNT.BAT The MOUNT.BAT batch file in listing 2.3 is much the same as PUBLISH.BAT.

Listing 2.3 Batch file for mounting TOPS volumes.

```
MOUNT.BAT

TOPS MOUNT %1 TO %2 %3 /RW
```

In this case you must enter three parameters at the command line: the driver letter to call the TOPS volume by, the name of the server that has published the volume, and the volume's alias. You do not gain much. The batch file simply saves some typing, which is not a bad reason for using a batch file. They are blessings for those of us with lazy or inaccurate fingers.

SHUTDOWN.BAT A problem with running a computer as a network server is the danger that it will be turned off while others are using its disk. This can have results that range from annoying to disastrous. In the Macintosh environment, users are conditioned not to just reach back and turn off the machine. Since the Finder presents a Shutdown option that ejects an internal disk and present a black screen with "You may now turn off the computer safely" screen, we are accustomed to using this feature. By creating a SHUTDOWN.BAT file and making sure it is used before turning off the computer, this Mac facility can be simulated and help to guard against shutting off a server being used.

Unfortunately, even with a SHUTDOWN.BAT file, such as the one shown in listing 2.4, it is too easy to just turn the computer off.

Listing 2.4 Batch file that will remove the computer from the TOPS network first checking that it will interrupt no one else's work.

```
SHUTDOWN.BAT

ECHO OFF
CLS
ECHO Please check the next screen to make sure you
ECHO don't have any active users.
PAUSE
CLS
```

continued

Listing 2.4 (cont.)

```
TOPS /Q PSTAT
ECHO If you have no files currently open or if you
ECHO have no directories currently open, go ahead and shutdown.
ECHO Otherwise, you risk damaging others' work.
ASK Shutdown?
IF ERRORLEVEL 1 GOTO NO
TOPS /Q SHUTDOWN
TOPS /Q UNLOAD /A
CLS
ECHO You have shutdown TOPS. You can turn off your computer.
:NO
ECHO You have not shutdown TOPS
PAUSE
CLS
```

Since there is no way to make a batch file check on its own to see if there are any active directories or files being used by other machines, this is second best. Basically, the batch file simply presents a message warning the user about shutting down and clears the screen. It then calls the command "TOPS PSTAT," which displays a screen listing all the published directories on the machine and whether there are any remote clients active. The batch file then asks whether to continue the shutdown process.

The "TOPS UNLOAD /A" line in the batch file actually removes the TOPS memory-resident modules from RAM, freeing up the memory so you can run other programs.

There are enhancements that could be made to this file. For example, you might call a "shipdisk" or "park" type of program that moves the hard disk heads to a safe location, in case the machine itself has to be moved. That is one function that is performed by the Macintosh Shutdown routine with many hard disks.

Unloading TOPS from Your Machine

Many DOS programs—such as Paradox or Ventura Publisher—require that virtually all the memory in your PC be free so that it can run. As software evolves and manufacturers try to add more and more power to programs, this will continue to be the case. It is quite likely that memory requirements for new programs will continue to grow. Until the release and wide acceptance of OS/2, which will probably take several years, there is, today, no satisfactory way to get around the 640-kb limitation of DOS. That memory space is getting more and more crowded. Unfortunately, TOPS must exist in that memory space along with the programs you need to run. Sometimes, there won't be enough room for both TOPS and your programs.

TOPS 2 allows you to remove most of the memory-resident (TSR) portions of TOPS from memory, thus allowing you to make room for large programs without having to restart your computer.

The BIGPROG.BAT batch file in listing 2.5 is a modified version of Shutdown. The difference is that after this batch file removes TOPS from memory, it includes statements to run a large program—one that cannot execute while TOPS is resident. After executing this program, control returns to the batch file, and TOPS is restarted. Remember that if you have other memory-resident programs (TSRs) loaded after the TOPS modules, you will not realize any gain in memory.

Listing 2.5 Batch file that removes TOPS from memory, runs a program, and restarts TOPS after the large program has finished executing.

BIGPROG.BAT

```
ECHO OFF
CLS
ECHO Please check the next screen to make sure you
ECHO don't have any active users.
PAUSE
CLS
TOPS /Q PSTAT
ECHO If you have no files currently open or if you
ECHO have no directories currently open, go ahead and shutdown.
ECHO Otherwise, you risk damaging others' work.
ASK Shutdown?
IF ERRORLEVEL 1 GOTO NO
TOPS /Q SHUTDOWN
TOPS UNLOAD /A
CLS
ECHO You have shutdown TOPS.
```

continued

Listing 2.5 (cont.)

```
ECHO Now executing . . .
REM —Insert commands here to start a large program, such as
REM Ventura Publisher or Paradox.
REM The next line will call your batch file "TOPSTART.BAT"
TOPSTART.BAT
:NO
ECHO You have not shutdown TOPS
PAUSE
CLS
```

Part D *PUTTING IT TOGETHER: IMPLEMENTING TOPS NETWORKING*

In the first three parts of this chapter, we discussed working with TOPS on the PC and on the Macintosh. Here, we'll talk about how to provide networking between the PC and the Macintosh. We'll also look at some tips on setting up a TOPS network.

One of the benefits of using a networking system such as TOPS is the flexibility it gives you in configuring the network: You do not have to dedicate a computer to the file server function, Macintoshes can store their files on PC hard disks (and vice versa), security is easy to arrange, and more. This means that there are few hard and fast rules for TOPS networking. If you are doing something that works, keep doing it. If the way you have things set up now is not working right, experiment to find solutions that meet your needs. How you set up your network will depend upon such things as how many computers are on the network, how many of each kind (PC or Macintosh) are being used, how many hard disks you are willing to purchase, and what kind of control you want or need to give to individual users on the network.

How TOPS Handles the Major Differences Between the Macintosh and the PC

As already discussed, the Macintosh and the PC have very different mechanisms for keeping track of files and directories.

Folders and Directories

At first glance, Macintosh folders are the same as PC directories. This is generally true. However, there are some subtle differences in the ways these structures are implemented on the two machines that affect the way we use them when networking. The technical background reasons for this are not particularly important here. What is important is that we have to be aware of these differences when publishing volumes to be used on other machines.

As discussed in part A, publishing is inherited. If you publish a directory or folder, all directories or folders that are contained in that published item are in turn published. This is the same on the Macintosh as on the PC.

When you mount a Macintosh volume to the PC, the folders contained in that volume are visible as if they were standard DOS subdirectories. All the standard DOS mechanisms for dealing with subdirectories work with them.

However, when you mount a PC volume to the Macintosh, this is not the case. The subdirectories contained in that volume are not visible to the Macintosh Finder or to other programs that access that volume. As we discussed in chapter 1, this is due to the manner in which Apple implemented folders in the revised HFS system of disk storage. All PC TOPS volumes become MFS volumes on the Macintosh. In the MFS or flat system of disk organization, folders are illusions created by the Finder, and folder information is kept in the hidden DeskTop file that is part of every disk. TOPS implemented PC volumes in this manner for good reasons: There is less overhead for keeping track of items in folders, and the performance of the network is improved. Future versions of TOPS may implement the HFS file structure on PC volumes, but TOPS 2 does not.

What does this mean to us as we use TOPS? Since publishing is inherited, when you publish a directory, every directory it contains is published in turn. Generally, too, mounting is inherited. Nevertheless, mounting a PC volume to a Macintosh violates this generalization. The solution lies in the TOPS Desk Accessory, which can "see" directories as folders and can be used to mount these folders separately. Mounting a volume published by a PC on another PC does not present this problem: All subdirectories are usable.

For example, suppose that a volume called "Data" has been published on a PC. This volume, in turn contains a subdirectory called "Sales." If "Data" is mounted on a Macintosh, then "Sales" will not be visible to the Macintosh Finder. However, it is visible to the TOPS DA. Use it to select the "Sales" folder that appears in the right-hand window and click on the Mount button. "Sales" will be mounted as a separate volume.

Files

There are major differences between DOS files and Macintosh files. Macintosh files may have long names that may contain characters that are illegal in DOS. Macintosh files have resource forks that contain information essential to the Macintosh; however, this information is not necessary for DOS to consider. TOPS must handle these differences for DOS to "see" and work with the files correctly.

As discussed in part B, TOPS/DOS creates a file called "XDIR-STAT.TPS" that contains file information. Most of this information is of a sort that DOS provides no standard mechanism for dealing with it.

File Names File names are one of the problems TOPS has to deal with: It must remove illegal characters and truncate or shorten the name to conform with the rules of DOS. As it does this, the first thing it must do is remove any illegal characters: spaces, characters such as * and $, and all the other characters that are illegal to DOS. If after doing this, the file name is legal, then TOPS does no more. However, the file might still be too long. If there is a period (.) in the file name, TOPS truncates the text to the right of the period to three characters and presents it as the DOS file extension. It then uses the first eight characters of the text to the left of the period as the file name.

It is possible, however, that with this method there might be two files on disk that have the same name: The Macintosh file names "Sales Reports January" and "Sales Reports February" are an example. If this is the case, TOPS will create one file in the manner already described. The second file will use the first seven characters of the file name, followed by a number, to name the file differently. To a PC, the two files mentioned above would be: "SALES_RE" and "SALES_R0". (Since there were no periods in these examples, TOPS will provide no extensions.) Of course, neither of these names are as descriptive of the files' contents as their Macintosh names. TOPS cannot know which of the characters in the file name are important and which are not. When you are creating files on the Macintosh that are to be used on a PC, try to give them names that will be meaningful in the DOS environment.

The mechanism that accomplishes this file name translation is the interbase file that is part of TOPS for the Macintosh. InterBase, like TOPS itself, executes as the system is started. If you are running a Macintosh-only network, then, that InterBase is unnecessary.

The TOPS PC XDIR command displays the files, showing both the DOS name of the file and the real or Macintosh name.

Resource and Data Forks Macintosh files have two components: the resource forks and the data forks. Typically, the data fork contains information such as the text typed in a word processor or spreadsheet. The resource fork often contains formatting instructions. There is no hard-and-fast rule about what these two forks contain. (Some programs, such as Odesta's Helix, keep all their data in resource forks.) In any case, the data in a resource fork is virtually always unusable to DOS programs. If DOS tried to read or display this information, only "garbage" would come out. TOPS must handle this discrepancy between the two types of files.

When displaying a Macintosh file on a PC, TOPS presents the file to DOS as two files: one containing the data fork and one, the resource fork. The resource fork has the same file name as the data fork, with the characters "R–" put at the front of it, to mark it as a resource. The file is then given the DOS hidden file attribute to prevent users or programs from trying to manipulate this inappropriate information. You can use the TOPS PC XDIR command to display the names of resource files.

You should be aware of two side effects to this. First, the DOS dir command does not tell the full story of a file's contents: It will show only the data fork, so that the file size information will be inaccurate. Some files appear to DOS to have no characters, when they could be quite large files. Macintosh files can have long resource forks.

Second, you should not use the DOS del or erase commands to remove files from TOPS directories, unless you are sure that it is not part of a Macintosh file. Remember that the DOS del command will not remove the resource portion of the file. Instead, use the TOPS Tdel command to remove Macintosh files from a PC directory or delete the file from a Macintosh.

Strategies for Working Between the Two Computers

Every network and every organization is, in many regards, unique. How you set up your network will depend on how you work. The strategies you employ for setting up the network will vary greatly depending on your organization, which persons or groups are using which computers, and more. There can be no rules set in concrete that dictate how you will install the network and work with it after installation. There are, of course, some generalizations.

Dedicated Server

One of the outstanding features of TOPS is its flexibility. A TOPS computer can act as a server without having to be dedicated to that function. There are times, however, when you will want to create a dedicated TOPS server.

It's fairly simple to make TOPS behave as if it were a dedicated server. You can dedicate as many computers—both PCs and Macintoshes—on the network as you can afford. It is sometimes a good idea to have at least one server computer in each work group.

The trick is to set up the computer so that it automatically publishes its hard disk in whatever manner you deem appropriate. It might publish the entire disk, or it might publish separate directories or folders on the disk individually, perhaps with different passwords. No matter how it publishes the hard disk, you will want to leave the machine alone while it is working.

If you are using a computer as a dedicated server and it will be operating almost constantly, be sure to turn the brightness on the screen down low or use a screen-blanking program to clear the screen. If an image is left on a monitor long enough, that image can "burn" into the monitor.

Ideally, a computer that is acting as a dedicated server should be a high-performance computer with a high-performance hard disk. In the PC arena, it is easy to buy inexpensive clones of IBM's AT; often they will be higher performance computers than the AT itself. There are many fast hard disks for the PC, too.

Since Apple currently only makes two levels of the Macintosh (the SE or Mac Plus and the Macintosh II), your choices are limited. It might not be cost-effective to dedicate the higher-performance, but it is much more expensive for Mac II to serve as a network server.

There are, however, many accelerator boards on the market for the Mac Plus and SE that can offer performance that is as good as, if not better than, that of the Mac II. A combined package of a Mac SE or Plus and an accelerator board can be less expensive than the Mac II.

Macintosh Use the Remember option in the TOPS Desk Accessory to tell TOPS how to publish the various folders on the hard disk. They will be published every time the machine starts from that disk. You can unplug the keyboard and perhaps the mouse, if you want to prevent others from using the machine.

PC Create batch files—perhaps with TOPSMENU or the TOPS Remember command—to publish volumes automatically. Place this batch file within your AUTOEXEC.BAT file or have the autoexec file call another batch file when it executes.

> ***Note*** Unlike the Macintosh, the PC will not boot if the keyboard is not attached. It will return an error message and wait for you to attach the keyboard before it will start. So you'll have to remove the keyboard, if you wish, after the machine starts up. You can, however, use the PC without a monitor, and if you need to change the configuration of the server only rarely, this can save some of the expense of using a dedicated server.

Work Groups

In most organizations, not all employees work with all other employees. Those who create the newsletters or promotional materials may have limited contact with those in accounting. People tend to work together in groups associated with their functions. Marketing personnel work with marketing personnel; accounting personnel work with accounting personnel. Your network will function best if you organize its functions around the groups of persons who will work together. Typically, individuals that work together will share information with each other more often than they will share it with those in other departments.

Here are some tips about organizing your TOPS network to facilitate this.

Zones A zone is a self-contained network that is joined to others with a device called a "bridge." In large networks, work groups can be divided up into zones. This is a great boon to performance. If most communication is between persons in the same zone, there is no cross chatter between the various networks. Traffic on the network is reduced, and performance is increased.

File Servers Within every work group, it is a good idea to install at least one computer with a hard disk that serves as primary file server for that work group. The network will perform more efficiently, and file security on the server will be easier to implement. (Generally only accounting personnel will access information on the accounting department's server.) If this file server is located in close proximity to those who are using the files on that disk, it will be quicker for them to retrieve their data. A dedicated server can be used for each work group, if desired. And if many of the computers in the work group are PCs, the dedicated server can serve as a print server for others in the work group as well.

Printers Just as file servers should be located near those who are going to be using their data most, the same should be said for printers. By giving each work group their own printers to use, conflict between the groups is reduced, and security is increased. (If there is a special printer for accounting, you can guarantee more readily that personnel in other departments will not read the reports from that printer without authorization.)

Files on Hard Disks There are probably as many ways to organize files on hard disks as there are users. In the PC world, it has been traditional to organize files according to the program that created them. WordPerfect files go into a WordPerfect data subdirectory, where they are further organized by project; the same is true for most PC programs. There is some logic to this in the PC world: Often there is no way to tell by looking at a file name what program created that file. And some programs that were developed before the release of DOS 2.0 did not allow the user to save files into separate subdirectories. And some users just never did get the hang of saving files into different subdirectories.

That style of file organization never really caught on with the Macintosh. Its method of working with folders is easier to learn and use. You can usually tell, by looking at a file's icon, what program created it. You merely double-click on a file to open it: The application program is invoked automatically.

In joining the two machines, a common method of organizing files must be employed, at least for those directories and files that are to be shared among users. Individual users may, of course, continue to use their own methods of file organization for those files they won't be sharing with others.

In general, it is best to organize shared files on a project basis. There are a number of reasons for this. Users can get at the individual files that comprise a project more easily, if they are all in one folder. When moving from one file to another, users do not need to mount new volumes and search in other folders for files.

This type of organization makes the important task of backing up easier to do, too. If all the files pertaining to a project can fit on one floppy, then it's easy to archive those files to floppy and erase them from a hard disk, without having to search through a multitude of volumes and folders for them. (Freeing up disk space, remember, is a good way to keep the network performance high.)

Any volumes that contain information to be used by several members of the work group should be published always. Use the AUTOEXEC.BAT file on a PC to publish these volumes at startup; use the Remember option in Macintosh TOPS to ensure that these volumes are always available.

Network Security

Security is very important in any computer operation, and it gets more important as that computer is integrated into a network. There will always be private information on computers that the information creators do not want others to see. And there will always be information that personnel in certain departments will want to keep secure from those in other departments, or the outside world. Networking makes security a little more complicated.

The simplest rule for keeping your files secure in the TOPS environment is: Don't publish it. Keep the data you want secure in directories or folders that are never published, perhaps on your local hard disk. If it's never published, then others can never mount it. Of course, they might still be able to get at the data by accessing it directly from your machine, but that's not a network issue. There are also many programs available that allow you to encrypt your files.

If the data you want to keep secure is on a remote hard disk, the situation is a little more difficult, since only the owner of that computer can assign passwords, which must in turn be entered whenever the volume is published. You are out of luck if you do not want the owner of the hard disk to be able to access your data.

Probably the best way to keep data secure is never to put it on a hard disk at all. If it's really important to you, keep the data on a floppy that you lock up in a safe each night (perhaps even in encrypted form).

Setting Up Temporary Volumes While we are working, we often create temporary files that will only be used for a short time. In chapter 3, we talk about a pivot file, which is used to translate data from one program to another. These pivot files are usually only used briefly to translate the data, then they are discarded. Sometimes a problem arises when you forget to delete these old, unneeded files. Setting up a special volume that contains these temporary files makes this a little easier to deal with.

For example, you may have a file on dBASE III on the PC that you want to transfer to HyperCard on the Macintosh. HyperCard won't read dBASE files directly, so you use dBASE's copy command to save the data in a format that HyperCard can read. Put this data into the temporary volume, read it into HyperCard, then save the HyperCard data into a separate volume. Later, if you forget to delete the pivot file, you can be assured that it doesn't matter since it's in the temporary volume. All data that goes into such a volume as this should be fair game for deletion. This helps to keep a hard disk unclogged and free of unnecessary files.

Like a simple electronic mail system, message folders can provide a drop box for individuals or work groups. A message folder is always published, with nearly free access by anyone else on the network. When network users want to send a message or a file, they simply mount that message folder, copy the file into it, and unmount it. To read or use the file, the owner can open it from that folder, read it, and delete it. If they want to keep it around, they can copy it into a more permanent folder. Again, the idea is to have a directory or folder to hold temporary files that are fair game for deletion.

Naming Resources In choosing TOPS names, clearly identify the computer or printer. If there are many of them on the network, a name such as "Mac SE", "PC AT", or "LaserWriter Plus" is not going to help anybody when they are accessing networked resources.

Names such as "Marketing AT File Server" (though long), "Eric's Compaq", or "LaserWriter Room 216" are much better at identifying the machine. Ideally, a name should include identifiers that indicate what type of computer or printer it is (e.g., AT, Mac Plus or LaserWriter Plus), as well as who or what owns it (e.g., Marketing, or Eric) or where it is located.

Using TOPS to Bridge with Other Networks

TOPS can be used simultaneously with network products from many other manufacturers, including Apple, Novell, and IBM. The actual techniques and strategies you use for bridging networks with TOPS may have slight variations depending on which of the other networks you are using. With all of them, the strategy has two parts: achieving "coexistence" with the other network hardware and achieving "coexistence" with the other network software.

When you are using the techniques that follow, one PC on the network is acting as a bridge between the two networks. It mounts volumes from a network server, then through TOPS publishes those volumes to other machines on the TOPS network. This is a good way to use TOPS for Macintosh networks in installations where there are existing networks of PCs. In this manner, the Macintoshes can fit into the PC networks.

Making the TOPS Hardware Compatible with Other Network Boards

Making the TOPS hardware compatible with other networking hardware is a matter of modifying the TOPS and/or other interface card inside the PC so it is not using the same DMA (Direct Memory Access) channel or the same interrupt. The TOPS Owner's Manual includes specific instructions on how to adjust the DMA channels and interrupts. Before experimenting, you should determine which of the DMA channels and interrupts the other network board (and any other boards in the computer) is using. In chapter 5, we discuss a system log that should be kept for every network installation. Always use this system log to document the various configurations for all interface boards installed in computers on the network. Configuring these boards is very complex, the more so as additional boards are added. The standard PC has a very limited number of channels by which boards can communicate with the central processor, and many manufacturers use common boards.

Table 2.1 lists the DMA and interrupt channels that allow TOPS to work with some common networks. It is possible, however, that these will not work for all installations. The other network board may have been reset to make room for some other interface boards. They are offered as a starting point for experimentation; in many cases, they will work. All settings are for the TOPS board only.

Table 2.1 DMA and Interrupt Channels.

Network Name	TOPS Board Changes
IBM PCNet	Board Interrupt = 3
Ungermann-Bass	Board Interrupt = 3
	DMA = 3 (IBM AT)
	DMA = none (IBM PC/XT)
Novell S-Net	Board Interrupt = 3
IBM Token Ring	Board Interrupt = 3
Standard Microsystems ARCNet	Board Interrupt = 3

Note When any of these changes are made, jumpers on the interface board will need to be reset, and the line in the CONFIG.SYS file that refers to ATALK.SYS will also need to be changed. Contact TOPS support for more details.

At the time of this writing, TOPS had been tested by the TOPS support staff. By the time this book sees print, many other multiple-network configurations may have been tested. If you are trying to bridge TOPS to one of these networks, contact the TOPS Technical Support Department.

Making the TOPS Software Compatible

Once the board is operating, you must load the separate network programs. Typically, both networking systems can "coexist" in the computer. To bridge the two networks, mount the server volumes used by the non-TOPS network first. Then you must make TOPS think that the virtual disks available to that computer are physical disks connected directly to it.

Suppose, for example, you are using the IBM PC Network program, to access data on a remote file server. You have probably used its net use command to make a server volume available to the computer, say, as drive D:. With this software active and after you have executed all the PC Network commands for accessing that drive, you can load TOPS and publish that volume so that other machines on TOPS can use it. This procedure is straightforward. The only thing you really have to change is the last line of the TOPSKRNL.DAT file, which tells TOPS which physical disks are available to that computer. In this example, the line would read something like "FFHHUUUUUUU," where "F" indicates a floppy disk, and "H" indicates a hard disk. If this line contains a "U" for any volume (either physical or virtual from a network), then TOPS will not allow you to access, let alone publish, that volume.

Generally, when you are bridging two networks, the situation is somewhat more fragile than when one network is running by itself. The computer that is acting as a bridge should be dedicated to that service; if it is running other software, performance of that software will most likely degrade considerably, and its performance as a server will degrade as well. Since both networks will be using scarce DOS RAM, there probably won't be much left over for application programs.

Bridging to AppleShare

TOPS can act as a bridge to AppleShare in the same manner, though the problems with the boards do not occur, of course. On one of the Macintoshes that is acting as a workstation on the AppleShare network, load TOPS and publish the AppleShare volume to which that workstation Macintosh is connected. Other machines on the TOPS network—including PCs—can then access the AppleShare volumes.

When doing this, TOPS will not recognize the Access Privileges that are part of AppleShare. Instead, the volume that is published will have the same Access Privileges as the publishing Macintosh. For example, if the Macintosh that is serving as the bridge has only See Folders privileges for the AppleShare volume, or folders in the volume, those same privileges will be passed along to the TOPS users. A detailed discussion of AppleShare's Access Privileges can be found in chapter 3.

Again, any machine that is acting as a bridge should probably be a dedicated machine and should not be used to run applications programs (unless they are designed to "coexist" with a network server, such as an electronic mail program).

CHAPTER 3

AppleShare

AppleShare, introduced in January 1987, is Apple's offering to the market, which provides file serving for AppleTalk networks. In both philosophy and functionality, it is very different from TOPS.

AppleShare differs from TOPS in two key ways. First, it requires the use of a dedicated server, a Macintosh on the network that makes its attached hard disk or disks available to others. Second, AppleShare features a rich set of tools for assuring the privacy of folders and documents residing on volumes attached to the file server. AppleShare also allows the AppleShare administrator to create groups of users who can have the same Access Privileges to folders and the files they contain. This is a very powerful means for protecting sensitive information and for decreasing the glut of files that is visible to everyone.

AppleShare PC, announced in August 1987, for delivery in the first quarter of 1988, is a set of software tools that allows PCs to serve as a workstation on an AppleShare network. Aside from the software, it requires that the PC also have an AppleTalk interface board installed. AppleShare PC conforms both to Apple's standards for file and record locking (of course) and to those standards established by IBM and Microsoft for file and record locking among DOS programs. This allows Macintosh servers to be repositories for DOS programs and data files, even if they are to be used exclusively by, and shared among, PCs.

The AppleShare System

The Server

As already mentioned, AppleShare requires that at least one Macintosh on the network be dedicated to the purpose of file serving. Only volumes connected directly to the server may be shared among users. AppleShare further requires that the server Macintosh be dedicated to that purpose: It cannot be running other programs at the same time (although there are certain other applications—such as Apple's LaserShare print spooler and most electronic mail programs—that can be executing concurrently). Other workstations on the network may have hard disks attached to them, but these hard disks may not directly be shared on the network. There may be more than one server on the network, and multiple networks may be joined with a bridge. Networks themselves can be connected by bridges, and the different networks are called "zones." Zones can also include groups of networks.

The Appleshare Administrator

The AppleShare administrator is the individual designated for overseeing all network functions. Foremost among these functions is registering users and assigning passwords. Using the AppleShare Admin program, the administrator registers the names and passwords of all those who will be using the server. Only those who are registered users will be allowed to use all the features of an AppleShare server. Others, who do not have a network name and password, will be able to log onto the server as guests. They will be able to use many of the folders on the volume, but they will not be able to assign Access Privileges to the items they create.

On AppleShare, users are often organized into groups. A group usually consists of individuals who work together or regularly share information. Individuals may belong to several groups. With these groups, it is possible for users to limit access to information in folders to just members of a specific group. Only the AppleShare administrator may create groups.

Access Privileges

One of the keys to the power and flexibility of AppleShare is the rich set of options available for the protection of information on the server disk.

When a registered user creates a folder on an AppleShare server, that folder is automatically set up as a private folder. Only its creator may see or change its contents. However, the user may assign privileges for that folder to others on the network. To understand how these privileges are assigned, let's look at the three categories of users.

Users

Owner The owner of the folder is the registered user who created them.

Group A group of users that has been created by the administrator.

Everyone Everyone includes all users who access the server, including both registered users and guests.

A registered user is a user who has been given a registered user name and password by the AppleShare administrator. A guest is any other user. Guests may log onto the server, but they will have fewer privileges than registered users. The folders they create are public (i.e., everyone may make changes to them), and they will be able only to see or make changes to folders that have that privilege set for everyone. They may not be part of groups.

Privileges Access Privileges for a folder are typically assigned separately for each of these three entities. The privileges that can be assigned follow.

See Folders This privilege allows others to see the folders that exist in a folder or volume. If you have this privilege, the folders within that folder will be visible when you open a folder. Whether you will be able to manipulate those folders depends on the privileges you have for that folder. If this privilege is not assigned to the users, then folders will be invisible.

See Files This privilege allows you to see the icons of documents and applications within the folder. If you do not have this privilege, then the files within the folder are invisible. If neither this nor See Folders is set, then you will not be able to open the folder.

If you have this privilege, you may work with the files in the folders but may not be able to change their contents. That depends on the next privilege, Make Changes. However, files inside the folder may be copied out of the folder to another, where they may be changed.

Make Changes With this privilege, you may make changes to the folder's contents, including creating, editing, or deleting files. If Make Changes is not available to you, the folder acts as if it were locked, and the applications program will treat files as if they had been loaded from a locked disk. Even with the Make Changes privilege available, folders within the folder will have their own privileges.

All these privileges are set at the folder level and may be set separately for all the folders on a volume, or even all the folders within another folder. If the privileges of a particular folder are not specifically set, then privileges for any new folder are limited to the user who created that folder.

Using AppleShare on Macintosh Workstations

A Macintosh workstation is a Macintosh that is connected via AppleTalk to an AppleShare server. AppleShare on Macintosh workstations is closely integrated and consistent with the Macintosh environment.

To use a server volume or folder, the workstation Macintosh must be connected to that server first. The Chooser Desk Accessory is used to do this. Before AppleShare (and on systems without AppleShare), the Chooser selected the printers—either LaserWriters or ImageWriters—the machine would use. Its use for selecting servers is a natural extension of this function.

When working with AppleShare, the Chooser presents a new icon in its left-hand window (see figure 3.1). This icon represents the AppleShare servers on the network. If you click on the icon, the Chooser's right-hand window will display a list of servers on the network. (If additional zones are available, an additional window will appear, showing the various zones.) If you select one of those servers and click on the OK button, a new window (figure 3.2) will appear, asking how you want to connect to the server. If you are connecting as a registered user, you must type in your network name and the password. The password appears scrambled to prevent unauthorized access by people who may be looking over your shoulder. (Furthermore, the password is scrambled using DES encryption as it is sent over the AppleTalk network to prevent electronic eavesdropping.) Clicking on the OK button brings up the next window.

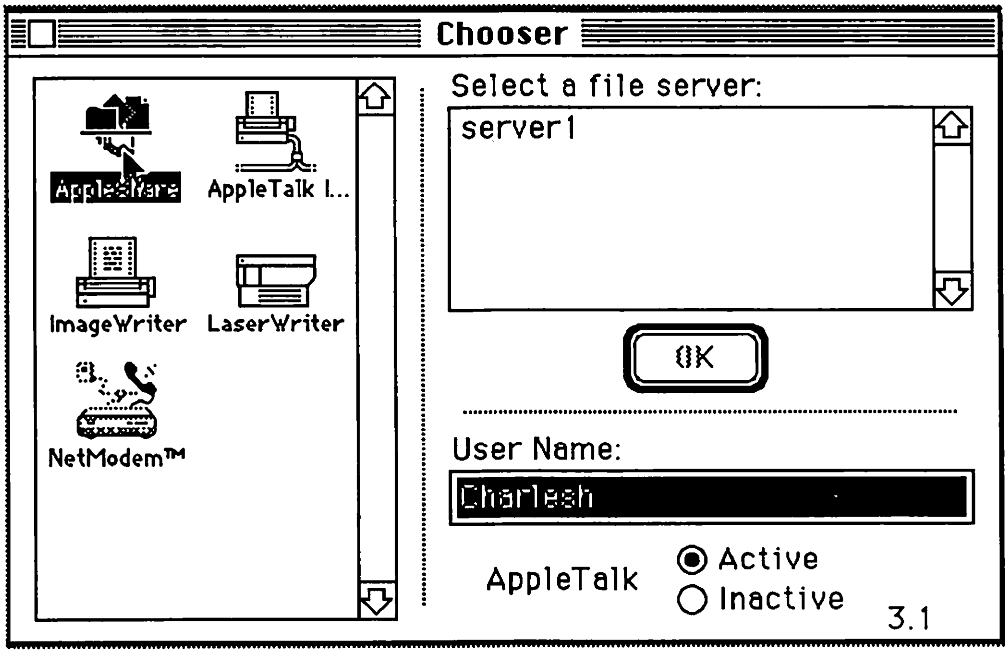


Figure 3.1 The Chooser Desk Accessory.

The window in figure 3.3 shows the volumes that the server has made available to the network and asks which volume you want. A box to the right of the volume name allows you to specify volumes that are mounted automatically when the workstation Macintosh is started. This feature—akin to the TOPS Remember function—allows the normal use of server volumes with the System, so that you do not need to log onto that volume every time the System is started. At startup, AppleShare will only request your password. Then it will connect to the server automatically (or the System can be made to remember the password and enter it automatically).

Once the workstation Macintosh is connected to the server, the

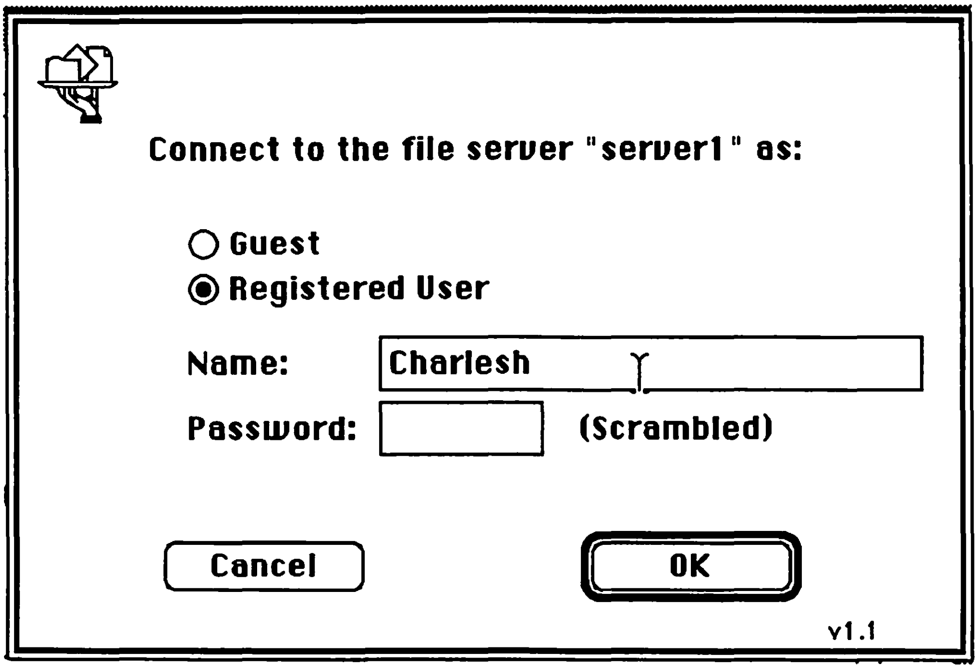


Figure 3.2 Selecting an AppleShare Server brings up this window.

Macintosh functions as it did before. The server volumes can be accessed using standard Macintosh methods—it appears to the users and to software as if it were simply another disk connected directly to the Macintosh. If you no longer want to use a server volume, then you must drag that volume's icon into the trash. The Finder will "forget" the volume, and the server will be aware that the volume is no longer in use.

Accessing Privileges for a Folder

AppleShare uses another desk accessory to allow users to review and change the Access Privileges for a folder. This desk accessory is called, logically enough, "Access Privileges," and its initial screen is shown in figure 3.4. The scrolling window shows all the folders at the current level of the server volume. You can scroll through this list of folders, open other folders, or use the standard Macintosh techniques for viewing other folders higher up in the folder hierarchy. Double-clicking on a folder opens it. When you have found the folder whose privileges you want to review or change, click on the Folder Info button.

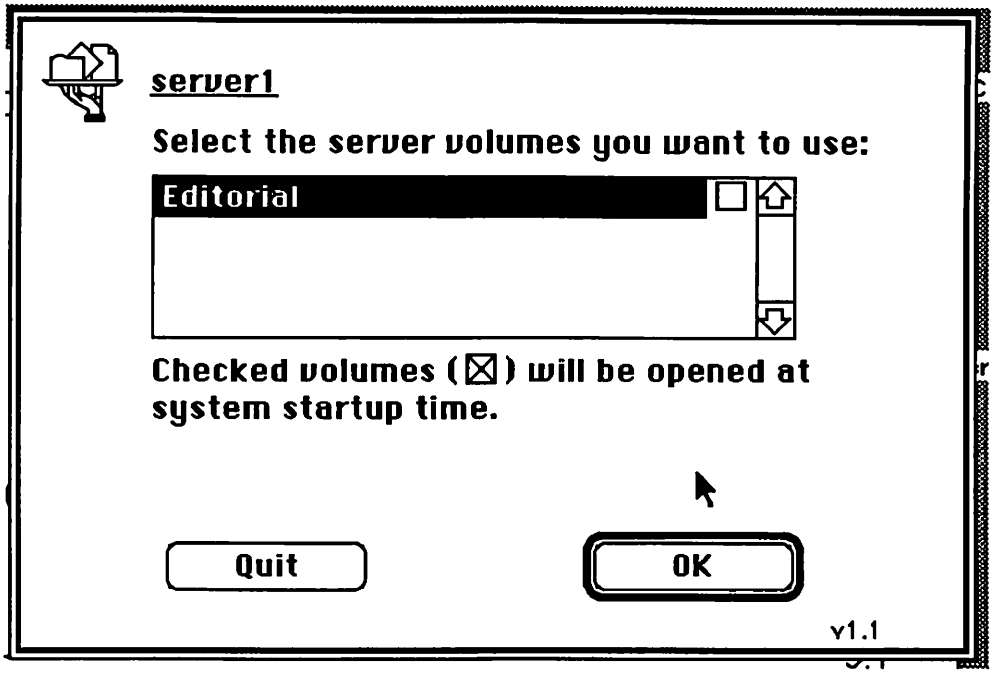


Figure 3.3 The Chooser shows the volumes connected to the selected server.

The next window has three panels or portions (figure 3.5). The top panel shows the name of the folder and the volume it is on, along with your user name and your privileges for that folder.

The middle panel of this window shows who owns the folder (i.e., its creator). If you created the folder, then your name will appear here. Boxes with an X in them allow those privileges for the entity listed at the top. If you want to assign privileges for a different group, type the group's name in the box above. If you want to work with another folder, click on the View Another button, and you will be returned to the Folder selection dialogue.

Folders You Create When you first create a folder, AppleShare automatically makes this a private folder. As the owner, you will be able to see folders, see files, and make changes to the items in the folder. No other privileges—for a group or for everyone—are assigned automatically. This is a method of protection. AppleShare assumes that you are creating private, not public, data. To make that data available to others, you have to say so specifically.

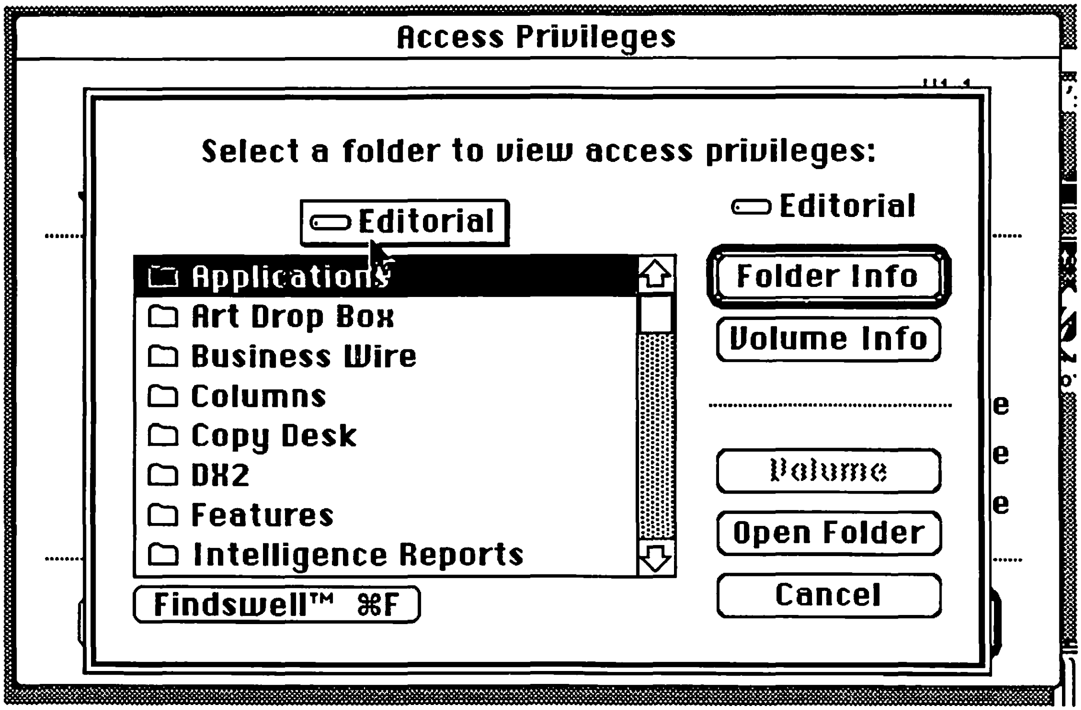


Figure 3.4 The Access Privileges desk accessory's first display.

If you are logged onto the server as a guest, however, all the folders you create will be public folders. Everyone will have all privileges for that folder, and you will not be able to change those privileges.

Folders Created by Others If you are viewing the privileges of a folder created by someone else, this display is quite different. (See figure 3.6.) The middle panel of the window now shows the name of the owner of the folder and what group he or she is a member of. Since you are not the creator of the folder, you may not change its Access Privileges, and those options are not presented.

Using PCs with AppleShare PC

AppleShare PC is a complete set of software utilities that allows PCs to connect to and use volumes on AppleShare servers. AppleShare PC

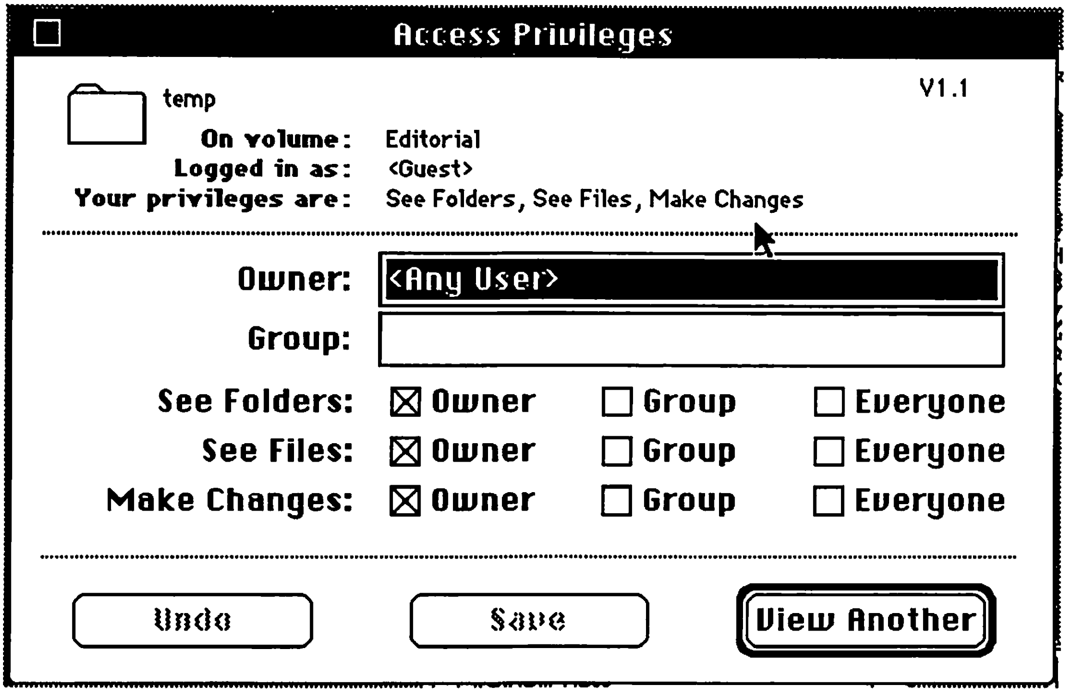


Figure 3.5 The display of privileges for a folder you have created.

software is flexible, providing both a menu-driven, memory-resident interface to AppleShare functions and the ability to execute these functions from the DOS command line or from within batch files. AppleShare PC works by assigning standard DOS drive designations to AppleShare server volumes. When connecting to an AppleShare volume, you must tell DOS how you want to refer to that volume. For example, if you have a PC with one floppy and one hard disk drive, the floppy is usually called drive A:, and the hard disk, drive C:. AppleShare PC will then allow you to call an AppleShare volume drive D: or drive E:. Once it is connected, all the standard DOS commands for working with disk drives work with AppleShare volumes.

AppleShare PC Software

AppleShare PC consists of the following software files and programs.

ATALK.EXE This program provides the software interface to allow DOS programs (including AppleShare PC programs) to use the Ap-

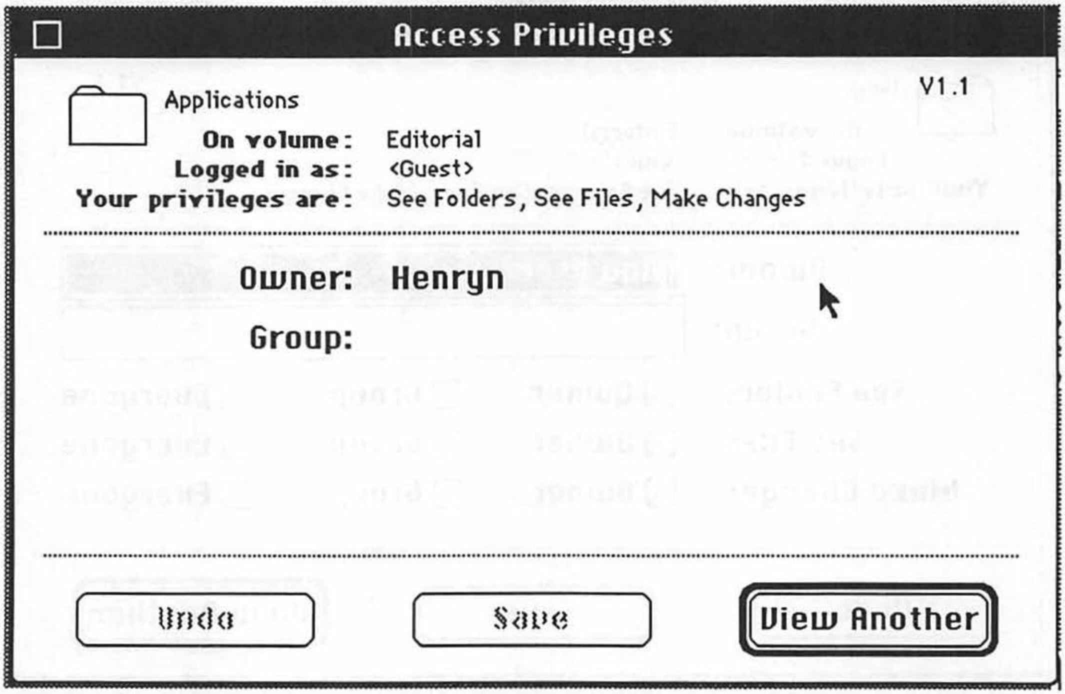


Figure 3.6 The display of privileges you have for a folder created by someone else.

pleTalk interface board. In function, it is equivalent to ATALK.SYS used by TOPS. Unlike the TOPS program, however, this is not a file that must be summoned by the CONFIG.SYS at System startup or boot time. It may be executed at any time and must be executed for the rest of AppleShare PC to run.

DA.EXE "DA" stands for desk accessory. This program, which can be executed as either a stand-alone DOS application or as a memory-resident module, provides the functions of both the Macintosh Chooser Desk Accessory and the Access Privileges DA. It also adds other functions that are appropriate to working in the DOS environment.

DA.DTA This file contains messages that are used by the AppleShare DA program, as well as user options that are "remembered" by that program for future sessions.

DA.HLP This file contains the text of various help messages that are presented by the DA.

ASHARE.COM This program allows the PC to access the AppleShare server. It translates network file system requests between MS DOS and AppleTalk Filing Protocol (AFP) formats.

MINSES.EXE This program is the session interface. A session is the logical connection between a workstation PC and a server. Separate sessions are maintained for each connection between the workstation and a server, even if they are multiple connections to the same server. This utility works in the background: There is no user interface for it, rather it is used by REDIR to communicate with ASHARE.

REDIR.EXE Another background program, this does most of the dirty work of handling redirection of output to the logical devices referred to by AppleShare. It makes the server volumes appear to DOS to be equivalent to locally connected disk drives.

ANET.EXE This program provides the command-line, DOS-like interface to AppleShare PC, allowing AppleShare PC commands to be executed from the DOS prompt, or from within batch files.

The AppleShare PC Install program—INSTALL.EXE—will automatically copy these programs to the disk you specify, either your local hard disk or a floppy disk.

Starting AppleShare PC

The AppleShare Install program will automatically modify your AUTOEXEC.BAT file to include the necessary commands to start AppleShare PC when you start the System. Here are the commands the installer will add to your autoexec file and a description of what each command does.

ATALK /M=X This line invokes the driver for the AppleTalk interface board. It is necessary for the other programs to work with the board. The "/M = X" instructs the driver to set aside this amount of memory to work with the board. During installation, the installer will ask you how much memory you want to set aside and will insert the amount of memory here.

ASHARE This line calls up the background portion of AppleShare and allows it to use resources on the server volume.

MINSES This line installs the session interface.

REDIR This line invokes the DOS redirector.

ANET AUTO "ANET" is the command-line interpreter for AppleShare PC functions. "AUTO" is a command to ANET to perform the standard automatic connections of the workstation to servers, assuming those connections have been specified. You use the AppleShare DA to specify which server volumes are to be mounted automatically, and those automatic connections are stored in a file that this command uses.

These commands will be placed in your AUTOEXEC.BAT file if you choose. However, you may not always wish to start your machine with the server connected. In that case, you can place these commands in a separate batch file, which will be executed when you choose.

> ***Note*** These commands are placed into the AUTOEXEC.BAT file by the prerelease version of the Install program. The release version may include other commands as well.

All the AppleShare PC files should, of course, be installed in the same directory. By default, the Install program will place them in the root directory or another directory that you specify. This directory should also be included in your DOS search path, so that they will be available no matter which drive or directory you are currently logged into.

> ***Note*** After you have started AppleShare, only the DA and ANET commands will need to be accessed. Strictly speaking, then, only the directory that contains these files needs to be included in the PATH statement. In some installations, it might be preferable to place ANET and the DA in locations separate from the other files. You will need to remember these locations when you update AppleShare in the future.

The AppleShare Desk Accessory

DA.EXE is the primary program you will use to handle AppleShare PC functions. This program provides all the functions provided by the two Macintosh Desk Accessories (which is why, of course, it is called "DA"), as well as some extensions to those functions that are uniquely suited to the DOS environment. DA can be executed as a standard DOS application or in its memory-resident mode, which allows it to be called up like a Macintosh Desk Accessory from within most programs.

Memory-Resident Program Memory-resident programs, also known as TSR programs (for "Terminate and Stay Resident," a special programming instruction that allows their presence), continue to remain in your computer's memory once they have been executed. Other programs that work in a similar manner include Borland's SideKick, Living VideoText's Ready, and a host of others. They are akin to Macintosh Desk Accessories in that they can be invoked without having to exit most programs.

Integrating memory-resident programs is a tricky process, and there are few standards for providing these functions in the DOS environment. If you are unable to access DA in its memory-resident mode, first examine the others that you may be using. Certain of these programs, such as SideKick, require that they be the last TSRs loaded; conflicts can arise if they are not. To help troubleshoot these problems, start with a System that has no others loaded and add programs one at a time. Pay attention to requirements of such programs as SideKick and make sure you are following all their guidelines. Since memory-resident programs use up System memory, which can be scarce in PCs with less than 640-kb (and even in those with more memory), having too many of them resident at one time can severely impact your use of some other programs. Software such as Xerox's Ventura Publisher and Paradox require a great deal of memory to run and may not run at all with several of these modules taking up memory.

To execute DA as a memory-resident program, you must include the parameter /R on the command line in this manner:

```
C:>DA /R
```

DOS is not particular about parameters sent to a program with the slash (/), so spaces are not required in this line. They do, however, help you to read the line if you have inserted it into a batch file. If you have standard DOS expanded memory, DA will use that memory for its buffers, cutting down on the demands it makes of the standard 640-kb DOS memory.

DA will execute as normal, but when you press the escape key to exit the program, it will still be available for you to call it up later. You can access the DA later by holding down the ALT key and pressing the enter or return key. (If this does not work for you, someone else may have changed the key combination that summons DA. We discuss how to change this key combination later.) Figure 3.7 shows the initial screen presented by the DA. The active element is Chooser, which allows you to connect to a server.

```
Chooser          Assignments         Special          Options

  Select a Type:                 Select a File Server:
» 1 AppleShare                   1* demo serve

  Select a Zone:                           Workstation Name:
  1 Personal Network                       Demo PC
  2 EtherTalk
                                                  AppleTalk PC
                                               Desk Accessory 1.0b2
                                           (C) 1987 Apple Computer Inc.
                                               All rights reserved.
F2:Begin Log-on  F3:Log-off  F4:Log-off all File Servers  F10:More
```

Figure 3.7 The AppleShare Chooser DA.

Stand-Alone Application To execute DA as a standard DOS application, simply type "DA" from the DOS command line. DA will act as if it were a stand-alone program (i.e., it will not remain in memory when you exit using the escape key).

Using the AppleShare PC DA

As figure 3.7 shows, the AppleShare DA works similarly to the Macintosh user interface. Menus are listed across the top of the screen, and other menu items are accessed by pulling down those menus. (It doesn't work with a mouse.) To navigate among the DA menus, use the PC arrow keys to select menus at the top of the screen. As you move horizontally to different menus, that menu appears in reverse video. Its submenus pop down from the main menu. You can use the up/down arrow keys to select the different options on the submenus. Pressing either the enter or return key when a menu is highlighted executes that item, often causing a new window to appear. To exit from any window, press the escape key. It will always take you away from an item and return you to the next higher level. Eventually, if you use it enough, you will exit the DA.

Although the DA is not shown in figure 3.7 (or any of the other figures), it also shows a status line at the bottom of the screen. This line gives you error messages. The status line can be turned off or on, depending on your preferences. When you first start using the DA, you will probably keep it on. When you become more familiar with the program, you may want to turn it off.

Figure 3.7 shows the Chooser portion of the DA. This menu performs roughly the same functions performed by the AppleShare portion of the Chooser on the Macintosh. It allows you to connect or log onto AppleShare servers and makes those available to DOS. This window contains three subwindows, called panes. The first, labeled "Select a Type," shows the types of resources that are available on the network. At present, this shows only AppleShare servers. The second pane, labeled "Select a Zone," shows the different zones (individual networks) that are tied together. (It appears only if there are zones available to you.) And the third pane, "Select a File Server," shows all the file servers that are in the selected zone. You will probably use this one most often. If you have already logged on to a server, it is indicated by an asterisk (*) next to the name, as shown in figure 3.7. If you press F3 while a connected server is selected, you will log off that server.

In figure 3.7, the "Select a Type" window has a double-line border and a » symbol next to it. (For the curious, this symbol is called the "guillemet," along with its companion «.) Pressing the tab key selects different windows, and the up/down arrow keys move the » pointer to select different items within that window.

Along the bottom of the larger double-line border of the window, additional commands, executed with the PC function keys, are listed. Many of the function keys are standard in all the windows the DA will show you. F1 will always summon help, and F10 will usually call up additional F key commands whose descriptions would not fit at the bottom of the window (even if they are not shown, you can still execute them using the F keys).

To log onto a file server, use these navigation methods to select the server. Press the F2 key to begin your connection to that server. Figure 3.8 shows the "log-on" screen. Use the arrow keys to select the manner in which you wish to log on to the server.

```
Chooser          Assignments        Special         Options
=LOGON:
 Connect to server demo server

 in zone Personal Network

»Logon as:  Guest  Registered User




 F2:Log-on
```

Figure 3.8 Beginning the process of logging onto the server.

If you are logging on as a registered user, the DA will prompt you for your user name and password, as shown in figure 3.9. When you are typing the password, it will be scrambled on screen to prevent unauthorized access by others. Pressing F2 will move you to the next step in logging on. Before you can actually use the server volume, you will need to tell the DA how to connect the server to DOS. Figure 3.10 shows this screen.

```
Chooser          Assignments        Special         Options
=LOGON:
 Connect to server demo server

 in zone Personal Network

 Logon as:  Guest  Registered User

»User Name: Steve


 F2:Log-on
```

Figure 3.9 Logging on as a registered user.

```
Chooser         Assignments        Special        Options
=╔══════════Attach Volume to DOS══════════╗══════════════════════════════════════
 ║                                        ║
 ║ Connect to Server demo server          ║
 ║                                        ║
 ║ Its DOS Name is   DEMO_SERVER          ║
 ║                                        ║
 ║ ╔═Select Volume to Attach:═══════════╗ ║
 »║1 server volume                      ║ ║
 ║ ║                                    ║ ║
 ║ ║                                    ║ ║
 ║ ╚════════════════════════════════════╝ ║
 ║                                        ║
 ║                                        ║
 ║ Connect to DOS drive: E (Free)         ║
 ║                                        ║
 ║ Connect automatically                  ║
 ║ at startup?          Yes  No           ║
 ║                                        ║
 ║                                        ║
 ╚═F2:Attach to Root  F10:More════════════╝
```

Figure 3.10 Attaching the server volume to DOS.

This screen reminds you what server and what volume you are attaching and asks what DOS-logical volume or drive letter you want to assign this server volume. The DA will present a default volume that represents your next free drive. In the example shown, drive D: has already been assigned to a local hard disk, so the DA is proposing that E: be used to refer to this server volume. To use a different drive letter, simply type one in. If you wish to include this volume with those that are connected automatically at System startup (i.e., when the ANET AUTO command is used), then select "Yes" where the DA is asking you.

The DA also permits you to log onto a volume at a directory level lower than that volume's root directory. This is a very handy function. You may not always want or need to have the entire contents of a volume available to you. Running from the PC, you do not need to access folders that contain Macintosh application programs, for example. And if the data you need is at a lower directory level, you won't need to spend as much time using DOS to navigate to that directory to get to the files.

To log on to a subdirectory instead of the root directory, press the F3 key. A window—shown in figure 3.11—will appear, allowing you to move with the arrow keys to the various directories on the server. This window displays the short DOS directory names as well as the longer Macintosh names, which we'll describe later in this chapter.

```
 Chooser         Assignments         Special         Options
=╔═\\DEMO_SERVER\!serverv.olu═══════════════════╗═══════════════════════════
 ║  john           john                          ║
 ║  JOHNS          JOHNS                         ║
 ║  Oldies         Oldies                        ║
 ║» QUINN          QUINN                         ║
 ║  SCOT'S.STF     SCOT'S.STF                    ║
 ║  SCOTT          SCOTT                         ║
 ║  !SystemF.old   System Folder                 ║
 ║  TEMP           TEMP                          ║
 ╚═F2:Attach to Directory════════════════════════╝
```

Figure 3.11 Connecting to a subdirectory on the server volume.

Using the enter key opens a subdirectory that allows you to navigate among those at a lower level. Press F2 when you've pointed to the desired directory. Once you have logged on to a server volume in this manner, the selected subdirectory will appear to you (and to DOS) as if it were the root directory on the volume.

DA also allows you to do some additional work with both the server and local volumes. The next menu to the right of the Chooser menu is the Assignments menu, and you can use the left/right arrow keys to get to it. This menu shows a list of the volumes currently mounted to your system, including both the local and server volumes. Figure 3.12 shows a sample menu that was taken from a system that had two removable hard disks (of the Bernoulli Box variety) called drives C: and D:. Normally, "Fixed Disk" would appear here to designate your hard disk(s).

```
 Chooser         Assignments         Special         Options
═══════════════╔══════════════════════════════════════╗══════════════════
              »║A  Removable                           ║
               ║C  Removable                           ║
               ║D  Removable                           ║
               ║E  AppleShare (\\DEMO_SERVER\!SERVERV  ║
               ╚═F7:Set Current Drive  ════════════════╝
```

Figure 3.12 The DA Assignments menu.

Use the arrow keys to point to specific drives in the Assignments menu. With a drive selected, the F7 key sets that drive as the current drive. This means that when you exit the DA, you will be logged on to that drive. For example, if you are logged on to drive A: before you summon the DA, pressing the F7 key with drive E: selected would cause your DOS prompt to change from "A:>" to "E:>." This provides a handy means of navigating through directories and volumes without necessarily having to master the intricacies of DOS.

The Assignments menu also allows you to work with the subdirectories on a selected volume. Point to a volume and press the enter key. The next display, shown in figure 3.13, shows the contents of the current subdirectory on that volume. In the example shown, the information includes (at the left) the DOS name of the contents of the current subdirectory, as well as the Macintosh long name. We'll talk about Macintosh and DOS directory and file names later in this chapter.

```
Chooser        Assignments      Special      Options
                 E:\
   »  .               <DIR>
      !Applica.tio    <DIR> Applications
      CLIFF           <DIR> CLIFF
      !Cliff's.fol    <DIR> Cliff's folder
      DeskTop            0 DeskTop
      EXPENSES.WK1    7168 EXPENSES.WK1
      john            <DIR> john
      JOHNS           <DIR> JOHNS

      39067K free, 41471K total

  F2:Info  F3:Delete  F4:Rename  F5:Copy  F10:More
```

Figure 3.13 List of the files and directories in the current directory.

With this window visible, press the F9 key, which will show a nearly standard DOS directory of the folder shown in figure 3.14. The additional information it shows are the attribute bits for the files. "A" means the file has not been backed up, "H" means it's hidden, "S" means it's a System file, and "R" indicates a read only file.

While viewing the directory of a volume, you can maneuver through the subdirectories on that volume. Pressing the enter key with a directory selected opens up the contents of that directory in the same way it did with a volume. If you are viewing a subdirectory, you will notice two items at the top of the list of files. Both these

```
Chooser            Assignments          Special          Options
                   E:\
                     .              <DIR>
                     !Applica.tio   <DIR>   7/14/87   4:06 PM A
                     CLIFF          <DIR>   8/27/87   3:31 PM A
                     !Cliff's.fol   <DIR>   8/24/87  10:10 AM A
                     DeskTop            0   8/21/87   8:17 AM A    H
                  »  EXPENSES.WK1    7168   8/05/87  11:32 AM A
                     john           <DIR>   8/21/87   8:54 AM A
                     JOHNS          <DIR>   8/31/87   8:26 AM A

                     39067K free, 41471K total

                   F8:Privileges  F9:Long Names   F10:More
```

Figure 3.14 A nearly standard DOS directory of the selected volume.

items are also displayed with the standard DOS dir command but are seldom used. The first consists of a single period (.). DOS uses this to refer to the current directory.

> ***Note*** Although it is rarely used, you can use this item in many standard DOS commands. For example, to delete all the files on drive A:, we usually type "DEL A:*.*". Instead, you can use "DEL A:.". Since the period refers to the entire directory, the net result is the same. This offers more help for lazy fingers.

The next item in the list consists of two periods (..) and is more commonly used in DOS commands. The two periods refer to the parent directory of the current directory—the one directly above it in the hierarchy of directories. In the Assignments window, you can move up to the parent directory by pressing the enter key with this item selected. You can use the DOS change directory command in the same way, with "CD ..".

Just as you can set a volume to be the current volume from the top level of the Assignments menu, you can set a particular directory to be the current directory in the directory level of the menu. Simply point to a directory and press F7. When you leave the DA, that will be your current directory. Some DOS programs provide a limited means for traversing a directory tree. With the DA resident, this can make these programs a little easier to operate.

Get Info The F2 key brings up a screen similar to that available to the Macintosh Finder's Get Info window. This window displays the name of the item, whether it is a file or a directory, its size (in bytes if it is a file, in number of files and subdirectories if it is a directory), the date it was created, and the date it was last backed up. It also shows the privileges you have for that item. A comment box, similar to the one on the Macintosh, allows you to post short notes about the file. Figure 3.15 shows a sample Get Info window.

```
Chooser            Assignments         Special        Options
                   E:\
                    Dir Name:    !serverv.olu
                    Long Name:   server volume
                    on Volume:   server volume
                    on Server:   demo server

                    Size:        16 file(s)

                    Created:     Mon, Jul 13, 1987  10:32 PM
                    Backed up:   Mon, Jul 13, 1987  10:32 PM

                    Privileges: See Directories, See Files
                                 Make Changes
                     Comment:
                     This box appears when you press F2
                   » from the Assignments menu.



                    F2:Save Changes  F8:Privileges
```

Figure 3.15 The Get Info display from the Assignments menu.

Note The size of a file as shown in this window can vary from the one shown in a DOS directory of the file. This has to do with Macintosh resource and data forks. In the DOS dir command, the size of the resource fork is not displayed, only that of the data fork. In the Get Info window, the sizes of both forks will be shown. We'll talk more about this later.

Privileges With either the Get Info window visible or a directory selected, the F8 key summons the Access Privileges dialogue. This provides essentially the same information as the Macintosh Access Privileges DA. It allows you to see what privileges you have for directories you did not create and to set the privileges that others may have for those you did create. This display is shown in figure 3.16.

```
Chooser          Assignments         Special         Options
                 E:\
                 Name:          Applications

                 Logged in as:  Scott
                 Privileges:    See Directories
                                See Files, Make Changes

                »Owner:         Scott
                 Group:

                                    Owner  Group  World
                  See Directories:  [X]    [ ]    [ ]
                  See Files:        [X]    [ ]    [ ]
                  Make Changes:     [X]    [ ]    [ ]

                 Change enclosed directories? Yes   No

                 F2:Toggle Privilege   F3:Save Changes
```

Figure 3.16 The Access Privileges dialogue box.

The Assignments menu can also perform a number of other useful functions. The F3 key allows you to delete a file; the F4 key renames a file or a subdirectory—a particularly handy feature, since there is no support for renaming directories provided as part of DOS. The F5 key copies files and can optionally perform file conversions as it copies them. F6 allows you to create a new subdirectory.

The Special Menu

The next menu on the DA is the Special menu, shown in figure 3.17. This menu features a potpourri of functions that increase the ease of use and functionality of AppleShare PC.

```
Chooser          Assignments         Special         Options
                                     1 Disconnect all Devices
                                     2 List all Connections
                                     3 Review Auto-Connections
                                     4 Change Extension Mapping
                                     5 About AppleTalk PC
```

Figure 3.17 The AppleShare DA Special menu.

Disconnect All Devices The first item on the menu, this allows you to log off all the AppleShare devices to which you are currently connected quickly. Before doing this, the DA will ask you for confirmation.

It is always best that you officially log off a server before resetting or shutting down your computer. While there is no specific requirement in AppleShare that you do so, it ensures that the server knows you are no longer logged into the network and makes things a little cleaner. To do so, you can use this command or the ANET command, which is described later. We'll also discuss some common types of batch files you might create for performing functions such as this.

Review Auto-Connections This item opens a window (shown in figure 3.18) that displays the volumes you currently have selected to be mounted automatically when you start the System (or when the ANET AUTO command is issued) and the drives to which these volumes are connected. The F3 key allows you to remove a selected auto-connection from the list.

```
Chooser        Assignments        Special        Options
                                  ═══Review Auto-Connections═══
                                    Connection     User       Device

                                » 1 demo server     scott      E:
                                  F3:Remove entry
```

Figure 3.18 The AppleShare volumes that are mounted automatically when you start AppleShare.

Change Extension Mapping This item allows you to tell AppleShare how to handle some of its basic file conversions and describes certain types of files to it. A more detailed discussion of this feature appears later in this chapter.

About AppleShare PC This item displays information about the program, including the version number you are running, copyright information, and how much memory is used.

The Options Menu

The Options menu (figure 3.19) allows you to change some of the parameters of how the DA works on your system. To change any of these parameters, point to the one you want to change and press the enter key. The first five items (with the brackets: [ ]) are toggles, and the enter key changes their state from on to off or the other way around. If there is an X between the brackets, that item is enabled; if not, it is not enabled.

```
Chooser          Assignments          Special          Options
                                                      » 1 [ ] Show hidden files
                                                        2 [X] Show status line
                                                        3 [X] Verify deletes
                                                        4 [X] Fast video
                                                        5 [ ] Color Monitor
                                                        6 Change Colors
                                                        7 Change Hot Key
```

Figure 3.19 The DA Options menu.

Show Hidden Files Both the Macintosh and PC operating systems use hidden files for special functions, typically System functions reserved for use by the operating system. On the Macintosh, the DeskTop file is a hidden file that is used to store information about the files on disk, including where it is in its window, its icon, and the name of its creator. On the PC, two hidden files are IBMBIO.COM and IBMDOS.COM. These are part of DOS and are required on a disk for it to be a startup disk (the format command with the /S parameter puts them there). The names of these files are the same even if the computer is not made by IBM. Enabling this option allows you to see these files and to work with them.

However, these files are usually hidden for a good reason: You are not supposed to mess around with them. Deleting the two DOS hidden files from a startup disk will cause that one to boot no longer.

> ***Note*** You should work with these files only if you know what you're doing and have good reason to be doing it. This warning is especially true if you plan to delete them.

Show Status Line As mentioned in our discussion of the DA, this line appears at the bottom of the screen. It is used for prompts and warnings and allows you to hide the status line. Most error messages and serious warnings will appear in pop-up windows.

Verify Deletes This item causes the DA to stop and ask you when you want to delete files. Most users, especially those familiar with

DOS, will keep this turned off, since it requires a few more keystrokes to delete a file. If you are new to the System or are performing some complicated deletions, keep it turned on.

> ***Note*** It is advisable to keep this feature turned on, even though it goes against the DOS grain. It is very easy to hit F3 instead of F4 (rename). Unlike files on standard DOS volumes, files on server volumes cannot be "undeleted" with utility programs once they have been deleted. Unless you really hate confirmation dialogues, save yourself the possibility of some grief.

Fast Video With this item you can update the screen with new menus and windows. On some older PCs and clones, especially those that have a monitor interface compatible with the IBM Color Graphics Adapter, this may cause "snow" to appear on the screen. If you get snow on your System and you do not like it (it's really no more than visual nuisance), keep this option off.

Change Colors For those who have color monitors, this item allows you to set the colors for most of the types of items that appear on the screen, including window borders, warning messages, and menus. This is a matter of your own choice, although good taste counts! If you have a monochrome monitor, you will be able to set attributes such as underlining, reverse video, and high intensity with this menu.

Change Hot Key The hot key is the key combination that summons the DA in its memory-resident mode. When the DA is first installed, the hot key combination is ALT-enter. If you are running the DA with other software (usually other memory-resident software) that uses this key combination for its own purposes, you may not be able to access the DA. When you select this option, the DA first verifies that this is what you want to do (see figure 3.20). If you answer "Yes" to this dialogue, the next two keystrokes you type will be the new hot key combination.

> ***Note*** Be very careful when doing this to make sure that you really are typing in the key combination you want.

> ***Note*** If you are not running the DA in memory-resident mode, you will not be able to change the hot key.

```
Chooser          Assignments         Special        Options
                                                   1 [ ] Show hidden files
                                                   2 [X] Show status line
                                                   3 [X] Verify deletes
                                                   4 [X] Fast video
                 VERIFY:                                 or Monitor
                 Once you start this process,            Colors
                 any keys you press, including <ESC>,    Hot Key
                 will become the hot key combination.

                 Do you still want to change the
                 hot key?

                 »          Yes           No
```

Figure 3.20 Verification dialogue used to change the hot key combination.

If possible, always allow the ALT key to be the first key of the hot key combination. The ALT key is most often used for memory-resident programs. Your other applications probably won't interfere with it. Many programs use the control key as the first key in a command operation, so it is likely that any control keys you use might conflict with these applications. Also, many programs, such as WordPerfect, use the ALT key in combination with one of the function keys F1–F12; so try to avoid this combination, too.

Always be sure to document the new key combination. Use a post-it note attached to the front of the computer, so you can remember the key combination and others will be able to use it, too.

The AppleShare PC DA is a flexible program. It allows you to access server volumes from within other programs and provides the full range of functions you will probably need with server volumes. AppleShare also includes a command-line interpreter that gives you flexibility and power in using the server.

Working from the Command Line

The AppleShare PC program ANET.EXE provides a command-line interface for using most of the AppleShare functions. Using this command, you can place commonly used AppleShare commands into DOS batch files, thus automating many of its functions even more. We'll discuss each of these commands in full, then provide some sample batch files for their use.

ANET.EXE can be used in one of two ways. If you type "ANET" by itself at the DOS prompt, the program will put you into an interactive mode and present its own prompt (the greater than symbol [>]). Any of the ANET commands can be typed at this prompt, with immediate response. Typing "QUIT" or "EXIT" or pressing the enter key on a blank line at this prompt halts execution of the ANET program and returns you to DOS.

You can also type "ANET" followed by a space and a command at the DOS prompt. ANET will execute, interpret the command, act on it, and return immediately to DOS. There must be a space between the word "ANET" and its commands; otherwise, DOS may return the error message "Bad command or file name." However, ANET is very flexible regarding the syntax for many of its own commands.

ANET ACCESS The access command allows you to modify Access Privileges for a directory. To use the command, first type in the first letter of the entity for which you want to assign privileges followed by the privileges you want to allow in parentheses.

The entities for which you can change privileges are the owner, group, and everyone. The privilege indicators you can use are: C (to make changes), D (to see the directory or folder), and F (to see the files). There are two ways to grant a privilege: Place a plus (+) before the privilege letter or merely include a letter. To deny the privilege, include a minus (–) before the letter or omit it.

The following example shows the privileges being assigned for a directory called "Sales" on logical drive E:, which is on an AppleShare server. This example makes John the owner of the directory and allows the marketing group to see the files and everyone to see the folder.

```
ANET ACCESS E:SALES O(CDF=(JOHN)) G(DF=(MARKETING)) W(E)
```

Since there is no C or F assigned for this directory to the world, everyone will not have the ability to see or change files. Since there is no C assigned to the group, people in the marketing group will not be able to change the files.

The order in which you assign the privileges does not matter—world can come before owner. Nor does the order in which you list the privileges for each entity. Characters may be typed in either upper- or lowercase.

Another powerful feature of ANET ACCESS allows you to specify a different set of default privileges for newly created directories, on a per session basis. You do this by specifying a session name instead of a directory name. The following ANET "response file" will log on to the server "PUBLIC," attach it to drive D:, and set the privilege default so that new directories include read access for everyone:

```
LOGIN S(PUBLIC) U(JOHN DOE) D(PUBLIC)
ACCESS \\PUBLIC O(CDF) E(DF)
ATTACH D: \\PUBLIC\!APPLESH.DRE
```

ANET response files are discussed later in this chapter.

ANET ATTACH This command connects you to an AppleShare server and assigns a DOS logical device to the specified volume and directory on that server. The syntax is:

```
ANET ATTACH DRIVE SERVERPATHNAME
```

"DRIVE" refers to the logical DOS volume to which you want to attach the server and "SERVERPATHNAME" refers to the name of the server, its volume, and the folder on that volume. To connect your drive A: to the directory "SALES" on the volume "DATA" on the server "MAIN," the command would look like this:

```
ANET ATTACH E: \\MAIN\\DATA\SALES
```

Names that begin with the double backslash (\\) are known as network paths and are intrinsic to DOS networks-naming conventions. Network paths have the format of:

```
\\SESSION_NAME\VOLUME_NAME\PATH
```

Some DOS commands, such as "TYPE" or "COPY" accept full network paths. For example, after the attach command has been executed, the following two commands are equivalent:

```
TYPE E:WIDGETS.TXT
TYPE \\MAIN\DATA\SALES\WIDGETS.TXT
```

Additionally, software that is network aware will accept network paths when opening or saving files.

ANET AUTO This command performs an automatic connection to all the devices that were specified as auto connections in the DA's Chooser window.

This command makes it easy to set up a standard startup procedure for AppleShare PC. Use the DA initially to connect servers and make them automatic connections. These auto-connections will be recorded and implemented the next time you execute this command. Putting ANET AUTO into your AUTOEXEC.BAT file (along with the other necessary commands) will make the server volumes always available to you. Naive or inexperienced users can thus be protected from some of the intricacies of using AppleShare PC.

ANET CON This command simply displays a list of the current sessions.

ANET DETACH This command disassociates a DOS logical device from a server. To use this command, type "ANET DETACH" followed by the drive letter.

```
ANET DETACH E:
```

will disconnect drive E: from the server to which it was connected.

All connections can be severed with the /ALL parameter:

```
ANET DETACH /ALL
```

ANET LOGOFF This command closes a session with a server and detaches all associated drives. It takes the parameter of the session name instead of the device name. To log off your session with the server "SALES," use the following command:

```
ANET LOGOFF SALES
```

As with the ANET DETACH command, all sessions can be logged off with the /ALL parameter.

ANET LOGON This command connects you to a server, performing essentially the same functions as the Chooser portion of the DA.

The syntax for this command is:

```
ANET LOGON [Z(zone name)] [S(server)] [U(user name)] [P(password)] [D(DOSname)]
```

All these parameters or arguments are optional. If you include only the name of the server, it will log you on to that server as a guest. If you include any of these parameters, you must include the name of the server.

As we'll see in "AppleShare PC and DOS Batch Files," you can include very easily ANET LOGON and its parameters in a batch file to perform many automatic operations. You should not include the password in the batch file. Batch files are standard ASCII text, and others cannot examine their contents very easily. Not including the password in your batch file provides an additional level of protection.

The DOS name is optional; ANET LOGON creates a session name with that name. This is handy; it allows you to better manage the sessions. Any name you assign will probably be more meaningful to you when you are referring to sessions and may be easier to remember.

One of the prime purposes for using this command is to create session names that are easier to remember. The DOS session name is limited to a maximum of fifteen characters. Longer names are truncated: If you have a server called "ADMINISTRATION ACCOUNTING SERVER," then the session name will become "ADMINISTRATION_," which may not be adequate to describe the server. Creating a session name allows you to make a name that is easier to remember. On the other hand, if your server has a shorter name, there is probably no reason to use this command. Why create another name that you have to remember?

Here's an example of a logon command:

```
LOGON S(ADMINISTRATION ACCOUNTING SERVER) U(LEOPOLD BLOOM) D(DUBLIN)
```

In this command, the user "LEOPOLD BLOOM" is logging onto the server "ADMINISTRATION ACCOUNTING SERVER" and creating a DOS session name of "DUBLIN."

ANET MAP This command tells the server how to assign a Macintosh icon to a file on the server that was created by a PC. It only affects files created after you have issued the command and does not affect any files already stored on the server. File types and creators, and how AppleShare deals with them, are discussed in more detail in "AppleShare PC and Macintosh & DOS Files."

MAP can be used with or without arguments. If you use it without arguments, it displays the current defined extensions and their file types.

ANET TYPE This command is similar to ANET MAP, except it changes the type of a file that already exists on the server.

That concludes our discussion of working with ANET from the DOS command line. In the next section, we'll talk about using ANET to automate some AppleShare PC functions, using handy DOS batch files.

AppleShare PC and DOS Batch Files

As already mentioned, many AppleShare PC commands can be placed into batch files to make working with the System easier and more consistent. If you are not familiar with DOS batch files, consult your DOS Owner's Manual, or any number of good DOS tutorial books that are available. Chapter 2C also contains some hints about using batch files.

AppleShare PC can work in batch files through its ANET.EXE program. As already discussed, to use ANET include at the command line the command "ANET" followed by any of the arguments or commands that it can understand.

ANET also supports a second way of working with batch-type files from the command line. In this manner, all the commands that you want ANET to execute can be stored in a separate text file, which is used as its input. In a batch file, including ANET on each command line slows execution: ANET must be loaded into memory each time the command is executed. To implement this method, create a separate text file that contains each of the log-on commands you want executed—one command to a line, without the word "ANET" on the line. These are called ANET response files. Then, from the command line, type

```
ANET @ FILENAME
```

"FILENAME" is the name of the file containing the ANET commands. This is a standard DOS technique called redirecting. It is similar in principle to the process of redirecting involved in using printers or sending output to a file. Instead of using the less than symbol (<), you can use the "at" symbol (@). A sample of this method of using ANET is discussed with the batch file LOGON.BAT.

An AUTOEXEC.BAT File An AUTOEXEC.BAT file is, of course, executed automatically when the computer is first started. The AppleShare PC Install program will, if you choose, automatically modify your AUTOEXEC.BAT file to include the necessary commands to start AppleShare PC on your System.

However, you might not always want AppleShare to be active. If you are running large programs or simply do not plan on using the server disk, there's no need to take up the memory that AppleShare requires. The AUTOEXEC.BAT file can be modified so that it pauses and asks the user whether or not AppleShare PC should be started.

Listing 3.1 shows a sample AUTOEXEC.BAT file that does this. This batch file relies on the program ASK.COM being available to DOS (in your root directory or in another directory that is accessed via the path command). ASK.COM is part of the excellent Norton Utilities, which no serious PC user should be without. There are several other similar programs available—many of them in public domain—and they work in the same manner.

Listing 3.1 An AUTOEXEC.BAT file that pauses and asks if AppleShare PC is to be loaded.

```
AUTOEXEC.BAT

ECHO OFF
PROMPT $P$G
PATH=C:\;C:\DOS;C:\BAT;C:\ASHARE
REM ===================
CLS
BEEP
ASK Start AppleShare?
IF ERRORLEVEL 1 GOTO END_ASHARE_LOAD
ATALK /MEM=10K
IF ERRORLEVEL 1 GOTO END_ASHARE_LOAD
\ASPC\ASHARE
IF ERRORLEVEL 1 GOTO END_ASHARE_LOAD
\ASPC\MINSES
IF ERRORLEVEL 1 GOTO END_ASHARE_LOAD
\ASPC\REDIR
IF ERRORLEVEL 1 GOTO END_ASHARE_LOAD
```

continued

Listing 3.1 (cont.)

```
REM ====================
REM find out if the user wants to automount the volumes
ASK Automatically connect?
IF ERRORLEVEL 1 GOTO NOAUTO
GOTO ASKDA
\ASPC\ANET AUTO
IF ERRORLEVEL 1 GOTO END_ASHARE_LOAD
:NOAUTO
REM ==========
:ASKDA
REM start the DA?
ASK Load the AppleShare DA?
IF ERRORLEVEL 1 GOTO NODA
\ASPC\DA /R
:NODA
ECHO The DA was not loaded.
:END_ASHARE_LOAD
CLS
ECHO AppleShare not loaded
PAUSE
REM insert other startup commands here
```

The first thing this batch file does is modify the standard DOS prompt. The line "PROMPT PG" causes the prompt to include the name of the directory you are currently logged into, in addition to the disk drive designation. This makes it easier to keep track of where you currently are on disk. There are other modifications that can be made to the prompt command; this is a minimal one.

The path command is a standard DOS command that tells DOS where on disk it should look for commands, programs, and batch files. Again, this is a minimal path that is set up to look for commands in the root directory, the DOS directory, the batch file directory, and the directory that contains the AppleShare PC commands.

The line "ASK Start AppleShare?" prompts the user as to whether to start AppleShare running. ASK.COM presents the question to the user and waits for an answer (either a "Y" or an "N"). If the user answers "N," an error is sent to the batch file. The batch file interprets this error and branches its execution accordingly. In this case, it is branched to the section of the batch file called "END

_ASHARE_LOAD," which reports to the user that AppleShare has not been loaded.

The next of our modifications (in the line "ASK Automatically connect?") asks if the user wants to start the automatic connections with the server. This is useful if the PC is to be used by different people: Each user may want to use different servers or different volumes on the same server. So the person who starts the System may not be the person who will actually log on to the server.

If the user answers "No," the line "\ASPC\ANET AUTO" is not executed, and control flows to the labeled section "ASKDA." This asks the user whether DA is to be loaded. If so, the DA is loaded in its memory-resident mode.

ASHARE.BAT Listing 3.2 shows a modified version of the AUTOEXEC.BAT file that is used to start AppleShare after the System has already been started. It has all the same features of the autoexec batch, but it does not ask whether AppleShare should be started nor does it set the prompt command or clear the screen.

Listing 3.2 The ASHARE.BAT file.

```
ASHARE.BAT

ECHO OFF
ATALK /MEM=10K
IF ERRORLEVEL 1 GOTO END_ASHARE_LOAD
\ASPC\ASHARE
IF ERRORLEVEL 1 GOTO END_ASHARE_LOAD
\ASPC\MINSES
IF ERRORLEVEL 1 GOTO END_ASHARE_LOAD
\ASPC\REDIR
IF ERRORLEVEL 1 GOTO END_ASHARE_LOAD
REM ====================
REM find out if the user wants to automount the volumes
ASK Automatically connect?
IF ERRORLEVEL 1 GOTO NOAUTO
GOTO ASKDA
\ASPC\ANET AUTO
IF ERRORLEVEL 1 GOTO END_ASHARE_LOAD
```

continued

Listing 3.2 (cont.)

```
:NOAUTO
REM ==========
:ASKDA
REM start the DA?
ASK Load the AppleShare DA?
IF ERRORLEVEL 1 GOTO NODA
\ASPC\DA /R
:NODA
ECHO The DA was not loaded.
:END_ASHARE_LOAD
CLS
ECHO AppleShare not loaded
PAUSE
REM insert other startup commands here
```

LOGON.BAT Once AppleShare has been started, a second batch file can be used to make the actual connection with the server. Listing 3.3 is the batch file LOGON.BAT. This is a batch file for the user John Doe, who is logging onto the root directory of the server "SALES." Doe has a machine with one internal hard disk (drive C:), and a floppy (drive A:). He is performing connections with the folders (subdirectories) "JANUARY" and "FEBRUARY" on the volume "MAIN."

Listing 3.3 A LOGON.BAT file that performs an automatic log-on to a server and a volume.

```
LOGON.BAT

ECHO OFF
ANET LOGON S(SALES) U(JOHN DOE)
ANET ATTACH D:\\SALES\MAIN\JANUARY
ANET ATTACH D:\\SALES\MAIN\FEBRUARY
```

The first of these lines does the log-on to the server and establishes the name of the user. The next two lines connect the server to the workstation.

However, this same operation can be done in another way, one that allows a little more flexibility. A revised LOGON.BAT is shown in listing 3.4. This takes advantage of the ability of ANET to take its input directly from a second file, instead of from the keyboard or the command line. The secondary text file is also shown in Listing 3.4.

Listing 3.4 Two files that work together to perform the log-on to an AppleShare server.

```
LOGON.BAT
ANET @ %1

DOE
LOGON S(SALES) U(JOHN DOE)
ATTACH D:\\SALES\MAIN\JANUARY
ATTACH E:\\SALES\MAIN\FEBRUARY
```

In the example, the batch file is called "LOGON.BAT," and the secondary file is called "DOE." To perform the log-on, John Doe would type "LOGON DOE" at the command line. In the batch file, he would type "ANET @ %1." As already discussed, the "at" symbol (@) tells ANET to take its input from a text file. The "%1" is a batch file parameter; it means the same as the first word typed on the command line after the name of the batch file. In effect, this substitutes "DOE" for "%1" and performs exactly as if you had typed in "ANET @ DOE" at the command line. Of course, for this to work, the file "DOE" must be in the current directory; otherwise, the command line should include the full path of the file "DOE." For example, if the file "DOE" is in a subdirectory called "LOGONS" on drive C:, the command line would read:

```
LOGON C:\LOGONS\DOE
```

In this manner, you can create separate text files for the different people who might be using a machine. Each person will be able to perform their standard log-on procedures by typing "LOGON" at the DOS prompt followed by their name.

> ***Note*** In both these batch files, the password argument is not included in the text of the log-on statement. You can include the password in the batch file, so that the user does not have to stop and type it in during the log-on process. However, batch files are not secure from the prying eyes of

others. To prevent unauthorized entry into your files, keep your password to yourself.

You could revise this file many different ways. For instance, you might change it so it can accept several names at the prompt, performing several log-ons at once. Or you might make it so that it first logs off all other users before performing the next log-on.

SHUTDOWN.BAT The Macintosh Finder shutdown command is a very handy feature. On a Macintosh II, this actually turns off the computer. On all Macintoshes, it first disconnects you from any server volumes. The PC, unfortunately, has no similar command; however, you still should disconnect from any servers before turning off the machine.

Listing 3.5 is a brief shutdown batch file that simply logs off all the servers and tells the user that it has done so. AppleShare on a Macintosh is a resilient program that is able to recover well from errors such as these, but you should still let it know that you have logged off the servers, in case there are files that are left open.

Listing 3.5 A shutdown batch file.

```
SHUTDOWN.BAT

ANET LOGOFF /ALL
ECHO Disconnected from all servers....
```

These batch files illustrate just some of the possibilities of working with AppleShare PC.

AppleShare PC and Macintosh & DOS Files

In chapter 1, we discussed some of the differences between the manner in which the Macintosh and the PC deal with files on disk. In chapter 4, we'll take a closer look at some of these differences, particularly regarding ASCII text files. In the following material, we'll examine some of the facilities that AppleShare PC provides for translating certain files between the two operating systems. Some of this material won't make complete sense without some of the material that is in later chapters. You might want to refer back and forth for clarification.

File Names

On the Macintosh, file names can be up to thirty-one characters long and can contain any character except the colon (:). On the PC, file names can be only eight character long, plus a three-character extension. Several characters are illegal—they cannot be part of DOS file names. Chapter 1 contains a complete discussion of these file-naming differences. AppleShare PC must first translate the names of the DOS program files that reside on Macintosh servers into a form that DOS can recognize.

First, if a Macintosh file name also happens to be a legal DOS file name, then no change takes place in that name. This will probably not be the case for files that were created on a Macintosh. Mac users are just too accustomed to typing in longer, more descriptive file names. It will, of course, always be the case for any files actually created by a PC.

If a file name contains any illegal characters, AppleShare PC first removes those characters. If the resulting file name is legal (if it is ten characters long or less), then that becomes the file name, as viewed by DOS, with the addition of an exclamation point (!), which becomes the first character of the file name. The next seven characters are used as the actual file name, and the final three characters become the extension.

If the file name is longer than ten characters, AppleShare PC simply truncates or shortens the file name so that it will fit into the DOS mold. Some examples are shown in table 3.1.

Table 3.1 Mac File Names Shortened for DOS.

Mac Name	DOS Name
Sales Folder	!SALESFO.LDE
January Sales	!JANUARY.SAL
homework.txt	HOMEWORK.TXT

In this table, the altered form of the name includes the exclamation point as the first character. You can use the exclamation point to help determine whether a file was created on a Macintosh. In referring to the file from within a DOS program, you must include the exclamation point as part of the file's name, or DOS won't see it.

In chapter 4, we'll discuss in detail some of the differences between Macintosh and PC files and how to transfer those files between the two systems. AppleShare PC, like TOPS, provides some service that make much of this work easier.

AppleShare PC provides special tools for dealing with three types of files: binary, Mac Text, and DOS text. A binary file will often be a program, an image, or some other kind of file that is not normally readable as an ASCII (or formatted) text file. A DOS text ends each line with a carriage return and a linefeed as do most DOS ASCII text files—see chapter 4C on word processing). A Mac Text file is used most commonly on the Macintosh: Lines end with a carriage return and not the carriage return/linefeed combination.

AppleShare PC allows you to specify that a particular file (or group of files with the same extension) is of a certain type. Two commands allow you to do this: ANET MAP and ANET TYPE. For either of these commands to work, the file(s) in question must reside on the server; they do not affect files saved to local PC disks, and their primary purpose is to better effect the transfer of files between the two systems.

When any file is placed on the server, the Macintosh must know what kind of icon it should give to that file. If an application does not include the file type and creator information as part of the resource fork for that file, then the Finder gives it a generic icon. This icon shows no information about what kind of file it is—it can't because there isn't any. These files are called binary files and the server makes no assumptions about their contents. The default type for any PC file saved on the server is the binary type. AppleShare PC provides two commands that allow the file server to assign icons to files created by the PC and saved on the server.

ANET MAP is used to work with files that have not yet been created. It allows you to specify that all future files created with a specific extension are a certain kind of file and should have the appropriate icon.

For example, suppose you want to specify that all the files you create that have the extension .TXT are DOS text files. To do this, you issue the MAP TXT DOS-TEXT command. (This command can be executed from the DOS command line by preceding it with "ANET" and a space. It can be executed as written from the ANET prompt or can be placed in an ANET response file.) After this command has been executed, all files created by the PC and saved on the server with this extension will be given the appropriate icon.

If you are working with a word processor that has a counterpart in the Macintosh world or can read PC-formatted files directly, you can then tell AppleShare PC that these are Mac Text files. For example, the default extension for files created by Microsoft Word on the PC is .DOC. Since Microsoft Word on the Macintosh can read these files directly, you can use the MAP DOC MAC-TEXT command. Microsoft Word on the Macintosh will show these files in its Open dialogue and allow you to open them.

The ANET TYPE command works for files already stored on the hard disk, allowing you to change the types of those files. This command works in the same manner as ANET MAP, only it takes as its first argument the full name of the file in question. Its syntax is:

```
ANET TYPE FILENAME FILETYPE
```

Here "FILENAME" can be any path of a file. It may include the full network pathname (\\SERVER\DIRECTORY\FILENAME), or it may include legal DOS "wildcards" (\\SERVER\DIRECTORY*.TXT, for example).

You can use this command if you have already placed a number of files on the server volume, before you started using the MAP command.

Both ANET TYPE and ANET MAP have analogs in the AppleShare PC DA. To change the type of a file from the DA, use the Assignments menu to locate and select the file you wish to change and summon the File Info window for that file. Press the enter key one or more times to move the cursor to the "Type" listing below the standard file information. If the file is of a type that the DA recognizes, a pane will pop open, allowing you to choose among the three types recognized by the DA. To change the file mapping, select "Change Extension Mapping" from the Special menu. A dialogue will appear that allows you to specify which file extension you wish to map to choose from among the three types of files recognized by AppleShare.

Neither ANET MAP nor ANET TYPE perform any file translation. DOS text files will retain their carriage return/linefeed pairings. The AppleShare PC DA, though, can convert files when it copies the file from a local disk to the server or the other way around. When you use the DA's copy function (after selecting a file or folder in the Assignments menu), the DA presents a conversions dialogue. This dialogue allows you to select a conversion, and it defaults to no conversion. If you wish to convert from Mac Text to DOS text, use the cursor to highlight the appropriate option, and press Enter. Do the same for the DOS text to Mac Text conversion.

> ***Note*** If you are creating files to be saved on the server and only used by other PCs on AppleTalk, you don't need to worry about this extension mapping. PC users will never see Macintosh icons, and the file extensions will be sufficient for most PC users and programs. These functions are primarily a convenience provided for transferring files between the two systems.

Working with the PC and the Macintosh Using AppleShare

Since AppleShare PC is still in its unreleased state as this book is being written, it is difficult to come up with many tips on how to use it. What follows are some general tips that should prove useful.

Organizing the Server

In a Macintosh-only environment, data on the server will usually be grouped by project. That is, a specific department will typically have their own folder or volume. That folder will usually contain other folders that are specific to particular jobs or projects.

For example, the marketing group will have their own folder in the root directory or top level of the server hard disk. That folder will contain other folders that share the same type of information or relate to certain specific projects. All the advertisements and brochures for a certain product will probably be grouped together to make them easily accessible and to allow easy working with the Access Privileges. This will allow those who are working on the project to share the data.

Adding a PC to the network complicates things. Many of the files created on the Macintosh that relate to a specific folder will not need to be used on a PC and vice versa. In this instance, the folder should also have another folder inside it that separates the PC files from the

Macintosh files. This will make it easier for PC users to navigate the data and get at the material they need to use. The marketing department may have their sales figures and projections stored in a 1-2-3 worksheet or their mailing list for press releases stored in a dBASE file. Those files should be kept separate from the files that make up the project itself. (Additionally, the copywriter for the brochure does not need to have access to those files, so assigning different privileges for the folder that contains sales projections is a good way to keep those confidential.)

Generally, you should organize the server disk, so that the group or department that uses common information has its own folder at the top level of the directory structure, with different folders in that folder to hold information about specific projects. Project folders, in turn, should be divided into more folders that contain specific types of information.

Segregating Data

While we already mentioned this, you should keep data divided between PC-created data and Macintosh-created data. This reduces confusion: Mac owners may not be able to do much with certain types of PC-produced files and vice versa. It also provides another level of security for data on the hard disk. If PC owners are examining a file, the chances are greater that they will recognize that the file is meaningful and not delete it.

This raises an important issue. Files on the server should not be deleted unless you are absolutely sure that the file is not needed by someone else. The corollary: Delete all files that you are absolutely sure are not needed.

Much of this advice goes against the grain of how PC hard disks are often organized. Most PCs have a file structure that puts an application in its own subdirectory, with files created by that application either in the same subdirectory or in a data subdirectory that is a child of the first. This method arose for several reasons. First, early PC programs, such as WordStar 3.3, often could not work with the DOS subdirectory structure and, thus, required that their files be kept in the same subdirectory. In the PC world, data communication between two programs is not as common as might be, so different programs did not necessarily need to have access to the same files; this is changing. Finally, since DOS provides no inherent means of determining which application created a particular file (aside from extensions, which are not always used or are inconsistently used), the subdirectory structure was one thing that assisted in identifying a file's creator.

Setting Up Drop Boxes

Each user on the network should have a drop box into which other users can place files directed to that user. Drop boxes can serve as a minimal version of electronic mail or can contain files that you want another person to review or edit. Each user must create his or her own drop box. To create your own drop box, create a folder on the server and give that folder your name. Since you will want to protect the contents of that folder from the prying eyes of others, you will not assign the See Files or See Folders privileges to that drop box. Instead, assign the privilege of Making Changes to that folder to everyone (or the world). The result is a folder that only you can see the contents of but that everyone else can make changes to. They will be able to copy files into that folder (thus making changes), but they won't be able to find out what is in it.

CHAPTER 4

Coexistence: Transferring Files Back and Forth

Part A INTRODUCTION

Once you get the network hooked up and operating, new problems arise, the more so when you are integrating different computers. You might be able to use your PC to access Macintosh files and vice versa, but you still need to deal with some various incompatibilities between the machines. Your PC might be able to "see" Macintosh files, but will it be able to do anything with them? That's what this section is all about. Before getting into the specifics of file transfer, let's talk about some general principles.

In chapter 1, we saw how Macintosh and PC files differ from the point of view of the operating system. In this section we'll talk about how file formats, as used by the different kinds of programs, can differ and how to deal with this in an environment where we need to share files between two programs.

Most programs today store their data in a format specific to the program itself. To take advantage of special features to set themselves apart from other programs, most manufacturers store their data in their own formats. This allows them to implement specific features and get at their data quickly and easily. To transfer information between programs, therefore, we need facilities that allow us to translate files from their native formats into formats that can be read by other programs.

The Pivot File

The key to file transfer is the pivot or intermediary file. Given the wide range of programs we use, we naturally store data on disk in a wide variety of ways. It is not surprising that a word processor on the Macintosh cannot directly read files from most word processors on the PC. The same holds, of course, for databases, spreadsheets, and the other kinds of programs.

Many software packages come with special conversion utilities—small programs that allow them to read and write files to and from their native formats. These utilities are indispensable when transferring data from program to program.

However, it is not likely, or even possible, that any given conversion program can handle all the conversions we would like to throw at it. For example, the WordPerfect Convert utility can handle conversions to and from WordStar but not to MacWrite. The same is true of the Convert utility for Microsoft Word 3 on the Macintosh: It can convert to PC Word format but not to WordPerfect.

Instead we use intermediary or pivot formats. In this example, Word 3 on the Macintosh includes a utility that allows us to save files to DCA (document conversion architecture) format. WordPerfect, in turn, can translate files from DCA to its native format. Using the pivot of the DCA format, we can move files between these two programs. By supporting standard pivot formats (e.g., DCA), software manufacturers are relieved of much of the burden of having to write specific conversions between a myriad of different programs.

The pivot file is key to the transfer process. We will use this file over and over again.

Unexpected Uses of Software

In the early days of computers, software usually performed one function and one function only. A text editor allowed you to edit text.

You used another program to format the text and yet another to print it. With modern software, the distinctions blur somewhat. Although the fad of a few years ago for integrated programs, which tried to be all things to all people, has passed somewhat, today's software often performs more than one function. Your spreadsheet program now no longer just manipulates numbers, it can often assist in making and printing very nice charts from those numbers. Word processors are taking on more and more of the functions of desktop publishing programs. And there is at least one desktop publishing program that sports an integrated spreadsheet.

We can use our software in surprising ways to help us transfer data between programs, too. PageMaker on the Macintosh can read certain formatted text files from the PC and save them in Microsoft Word format for the Mac. Excel can read in text files created by a PC database and help us transfer them to a Mac product; it can also go the other way.

Recipes for doing this are hard to write: The circumstances vary considerably. The key is an extended awareness of a program's capabilities and of how those capabilities fit into a system approach to networking and data transfer. Only by exercising our software and by attempting to do new things with it can we discover its hidden powers.

Standards

The Need for Standards

In moving data from one program to another we rely on standard file formats that our different programs can read and interpret. Without at least some standards, this job would be difficult, if not impossible. When programmers create a new piece of software, they give much thought to the format that the data will take in their program. This is important in that it allows them to create unique features for their software and gives them great power in accessing and manipulating the data.

The need for a standard format for document interchange is a function of simple mathematics. There are at least fifty word processors for PC machines alone and another score for the Macintosh, with more new products appearing regularly. Add to that a couple of dozen word processors for minis and mainframes, and the number of necessary product-to-product translators grows into the thousands—too many for every developer to implement for every program. The same is true with other formatted data, such as that generated by spreadsheets and databases. With a standard format, each program has to translate only its files into that format to enable it to exchange data with all other products that read that format.

What's Wrong with Standards

Standards are, in some sense, the lowest common denominator.

Any programmer or manufacturer who wants to create, say, a new word processor must make many choices and decisions that affect the design process. One of the first steps is to determine what features the program will have. Almost always, some features will be added that are not generally available on other word processors, or perhaps a certain combination of features will be included that do not exist on other products. This is a crucial marketing, as well as a programming, step. To sell a product you must differentiate it in some way from other competing products. One of the ways you do this is by adding new or different features.

Now it gets complicated. Assume that as a programmer you give your word processor a new or unusual capability: You want it to be able to include very complex mathematical equations within the text. The user would then be able to type in characters that are only a half line high or less, alongside characters that might be as tall as two or three times the height of a regular character. And you want to be able to do it without adding extremely complicated codes to the text. It is to be a what-you-see-is-what-you-get program. You'll need a character set that includes a number of different, unusual mathematical figures, as well as a fairly large selection of the Greek alphabet.

There are a fair number of programs around that do just that. On the Macintosh, certain aspects of it are very easy since many mathematical, symbol, and Greek fonts are available. However, putting in the programming instructions that give your program the desired flexibility in laying out the characters is another story. You develop a unique file format that stores the formatting instructions (e.g., character height and placement) in a compact form. Your program can read and write the files quickly and reliably, and you are sure that it presents on the screen what the user wanted. Fine! The people that purchase and use your program are happy.

But if you want to place the file into PageMaker, you will be unhappy. PageMaker can place files that were created in Microsoft Word, but it cannot place those created by your program. If other users want to transfer formatted files to other word processors on the PC, they too have problems. You need to be able to write to a standard format that these other programs can read. In other words, you need to use a standard format.

This example is probably possible to accomplish using DCA or Microsoft's RTF formats. (See part C for a discussion of using DCA and RTF.) But it is not possible using ASCII or many word-processing formats (e.g., MacWrite or WordStar). Microsoft Word might be able to do it, but if it can, why did you write your program? The point is: Adding new features to a program disrupts the data structure of a file on disk. When a new program is implementing features not previously available, it needs a format to hold them in on disk.

Today, we have a multitude of programs, all, in a sense, speaking different languages. The words of one cannot adequately describe another. And given the nature of the problem, things are not likely to change for a while. Until someone comes up with a technique wherein we can include a full description of a file's data structure along with the file, we'll be in the same situation we are in now. Years ago, the authors of word processors did not envision laser printers and page description languages, so today we are struggling with programs written for daisy wheel printers and trying to get them to do desktop publishing. We are probably not aware today of the demands that such things as CD-ROM, HyperText, and integrated video editing will make on our data structures.

About This Chapter

For many readers, much of the material in this chapter will be somewhat extraneous. Many of the file transfers they will need to make will be easy to accomplish. Microsoft Excel will read Lotus 1-2-3 files directly and without too much loss of information. Some Macintosh word processors will read text files, or even formatted files, directly from the PC without even using pivot files. In these cases, transferring the files is easy and, with the exception of making some trivial changes to the imported files, painless. For other users, common utilities such as MacLink, the TOPS Translators, or Apple's File Exchange Facility will do the trick.

However, all readers should at least skim through this chapter. You might have data in a format that is not covered by one of the translation programs. Or, you might have several choices of which format to use (if you are taking data from, say, Samna on the PC to Word 3.0 on the Macintosh, you have many options: ASCII text, WordStar file format, MultiMate file format, DCA, and more). For users who can choose how to save data, this chapter should point the way to an increased understanding of the different translation formats and allow you to better choose the right one for the right job.

Translating files between the two machines gets easier almost every day. New programs or new versions of old programs arrive that support more foreign file structures. More and more file structures are added into the translation programs. But it is still not necessarily an easy task, and a good understanding of what's going on behind the transfer process only makes it easier.

Part B *TRANSLATION PROGRAMS AND UTILITIES*

There are several programs available, including some that are public domain or shareware, that can do much to facilitate the translation of files from one format to another.

Two of the leading programs—the TOPS Translators and the Apple File Exchange Facility—are based on the technology and design of DataViz' MacLink program, which has been around for some time. We'll talk about MacLink in the most detail; both of the other translators work in about the same way.

MacLink Plus

MacLink Plus, from DataViz, was one of the earliest translation programs available. It appeared on the market before two machines could be connected on a network, so it includes some features that are handy for nonnetworked operations, but are somewhat extraneous when working over the network. They do not get in the way.

MacLink Plus provides two features: one that allows the PC and the Macintosh to be connected via a modem or a serial cable and the other that does the actual file translation. MacLink Plus comes with two disks: one for the PC and one for the Macintosh. If you are using MacLink Plus to transfer files via the serial interface or modem, you'll need both. Since only the Macintosh program does the actual translation, in network operations you'll need only that disk. We won't discuss the modem or serial transfer operations; they are clearly documented in the manual.

MacLink Plus file translation can work with any disk that is available on the desktop to the Macintosh. That disk or volume can be a locally connected drive, or it can be a drive available to the Macintosh over the network. It can perform batch operations. In other words, it can perform the same translations on a number of files, making operation much easier and requiring far less intervention on the part of the user.

The MacLink Plus file conversion process has two parts: selecting the formats you will be translating to and from and selecting the files to be translated.

Selecting the Translators

Figure 4.1 shows the Set Translators screen of MacLink Plus. The window on the left shows the Macintosh file formats that MacLink Plus supports. The right-hand window shows the "foreign" (PC) formats supported. Above these two windows is a direction arrow that you use to indicate which way you are translating the files. Click in the button above the Macintosh translators to indicate that you will be translating files from DOS formats to Macintosh formats. The way the direction arrow is pointing will tell you how it will translate. Clicking on the arrow itself reverses its direction.

The MacLink Plus translators exist in pairs. That is, for each option in one of the windows (either the source or destination), at least one matched pair appears in the other window.

For example, if you click on MacWrite format in the left-hand (Macintosh) window, only word-processing or text files—MultiMate, MS Word, WordStar, DCA, or text—will be available in the right-hand window. The same is true if you choose Jazz in the Macintosh window: Only those formats that are appropriate will be shown in the other window on the PC side.

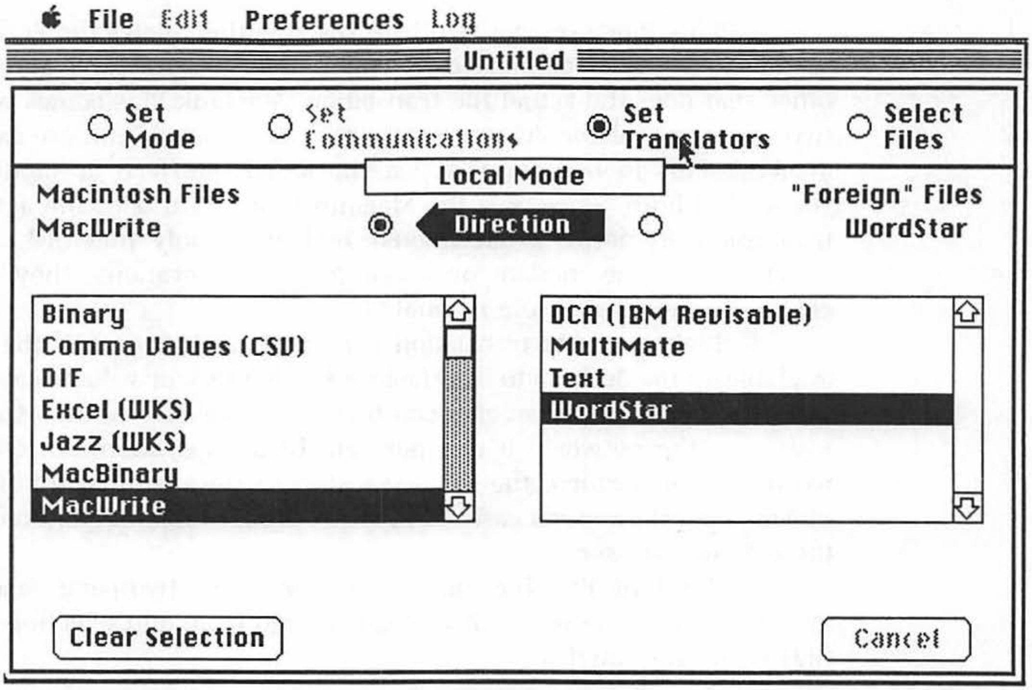


Figure 4.1 MacLink Plus' Set Translators screen.

Sometimes when using the translators, you'll encounter situations where your programs are not supported directly. For example, if you need to convert text from WordPerfect on the PC to MacWrite on the Macintosh, that format is not supported directly. Instead, you need to use WordPerfect's Convert utility to first translate the WordPerfect file into one that MacLink can work with—MultiMate, DCA, or WordStar. You can then use MacLink to translate the file from one of those formats to MacWrite. As we'll see in part C, a little something is usually lost with every translation we make. In this case, we are doubly removed from our original WordPerfect document.

File Selection

Once you select the translation you wish to take place, you need to select the files you wish to translate. To do this, click on the Select Files button at the top of the window. This brings up the screen shown in figure 4.2.

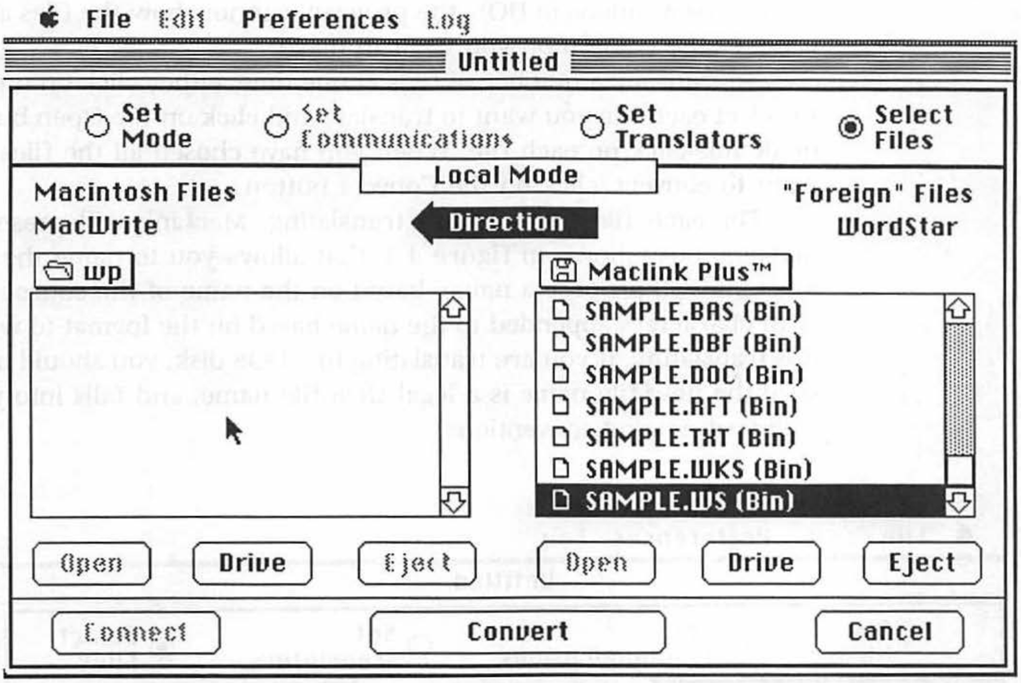


Figure 4.2 MacLink Plus' Select Files screen.

In this screen, the left-hand window shows the disk to which we want to save the new files. Initially, this will show the current default drive and folder. You can use standard Macintosh techniques for selecting the drive and folder to which you want to save the files. Double-clicking on a folder opens it, clicking on the Drive button cycles between the various mounted volumes, and clicking on the Eject button ejects a floppy disk.

The same is true for the right-hand window. This window, however, will allow you to select files by clicking on them. Again, standard Macintosh disk-handling techniques work here. If you are using TOPS, the volume shown in this window can be a mounted volume from a PC. (In fact, mounted volumes can appear in either window.) That way, you do not necessarily have to copy the files to your local disk before translating them. MacLink can copy as it translates.

This window will show all the files on the source volume—regardless of file type or creator (if it is a Mac volume) or its extension (if it is a PC volume). Since there are few, if any, standard file-naming conventions in DOS, the program cannot show the files applicable to the translation you want to make.

To translate a number of files at one time, either click on the file to select each file you want to translate and click on the Open button or double-click on each file. When you have chosen all the files you want to convert, click on the Convert button.

For each file that you are translating, MacLink will present a dialogue box, shown in figure 4.3, that allows you to name the file. MacLink will propose a name, based on the name of the source file, with characters appended to the name based on the format to which it is translating. If you are translating to a DOS disk, you should make sure the new file name is a legal DOS file name, and falls into your standard naming conventions.

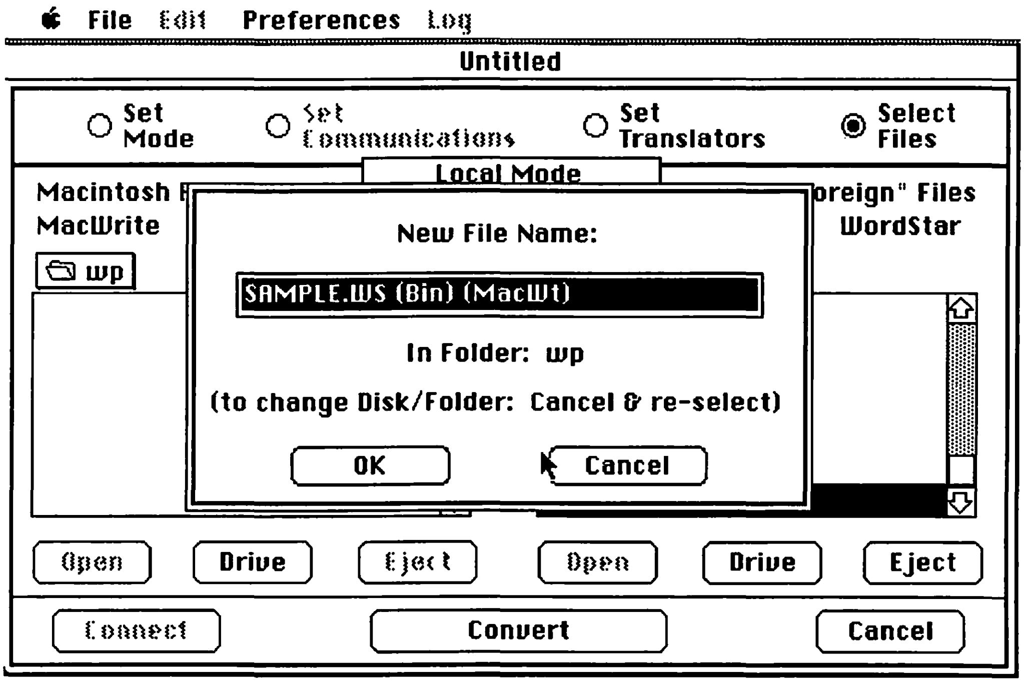


Figure 4.3 Screen before translating a file.

Figure 4.4 shows the screen that MacLink presents as it translates the file. You can choose "Cancel" to halt an operation if the program seems to hang up or if you realize you have selected an inappropriate translation or file. When the translation is complete, the Cancel button is replaced by a Done button, and the Print Log button becomes active. "Print Log" allows you to print the log of errors that occurred during the translation.

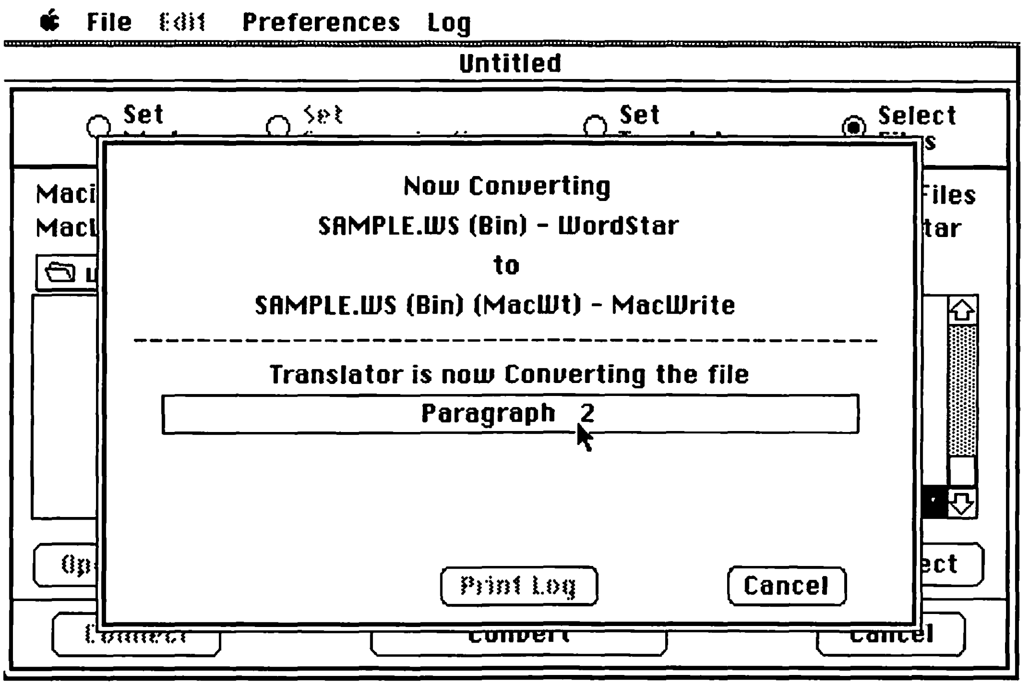


Figure 4.4 The MacLink Plus status dialogue shows the progress the program is making toward translation.

Supported Translations

Word Processing MacLink Plus supports the following word-processing formats on the PC: MultiMate, DCA (RFT), WordStar, and ASCII text. On the Macintosh, it supports ASCII text and MacWrite.

Part C discusses translating these files in more detail. MacLink supports the following features: margins, rulers, paragraph indentation, tabs, tabular tables, underline, italics, print styles, bold, superscripts, and subscripts. Depending on the translation method you choose, it may also ignore such things as hard page breaks, headers, footers, footnotes, and font changes.

Spreadsheets MacLink Plus supports the following spreadsheet formats: SYLK (Multiplan and others), WK1 (Lotus 1-2-3, version 2), WKS (1-2-3, version 1.0a and other compatible spreadsheets), WRK (Symphony), DIF, and some others.

Before using MacLink or another translator for spreadsheet files, look at the spreadsheet you are using to make sure you need to do the translation. Excel on the Macintosh can read and write WKS and SYLK files directly.

Databases MacLink Plus supports the standard pivot formats of comma- and tab-delimited files. You can even use it to translate from the DOS standard comma-delimited format to the standard Macintosh tab-delimited format. Since virtually every Macintosh database program (as well as spreadsheets and word processors) can read the tab-delimited format, this is very handy. MacLink Plus also supports translating files to and from the dBASE II or III formats to and from delimited file formats, DIF, and SYLK. Chances are your database program will support one of these features.

MacLink Plus is a flexible program. While it does not support the native format of every program you will be using, it does support pivot formats that most programs can use.

The TOPS Translators

The TOPS Translators is a modified version of MacLink Plus that works well with TOPS. Since you have the connections made to the different computers with TOPS, you do not need the communications options of MacLink Plus with TOPS, so it is not included. In all other respects, it works in the same manner as MacLink Plus.

The Apple File Exchange

At the time of this writing, this program was not yet available. With the imminent release of Apple's new System software, this program will be included with all copies of that System's software.

One unique feature that the Apple File Exchange will offer is the ability to change the translation options available in the program. This will be in the form of a set of drivers that can be installed to work with the program and will allow software manufacturers to develop specific drivers for their own programs. For example, a manufacturer of word processors will be able to include a specific driver for their program, allowing conversions to be direct from, say DCA, to their own format, thus eliminating the need for a pivot format such as MacWrite. The same will be true of spreadsheet and database operations. It is one step closer to offering a transparent and universal means of file translation. The user interface appears to be similar to the one for Apple's Font/DA Mover, so it will be familiar to most Macintosh user.

Other Translation Programs

While most Macintosh owners will have the Apple File Exchange program and all TOPS users will have the TOPS Translators, not everyone will have the translators for some of the conversions we need. And since they all work on the Macintosh, some programs for the PC can come in handy.

A couple of DOS-based programs written in a generic BASIC for the PC follow. Many thanks to Eric Alderman, who wrote these programs and graciously permitted their use here.

When working with simple ASCII text files that are to be transferred to the Macintosh, a small program that can manipulate those files is useful. While many utilities are available that perform the functions of these basic programs, they can be useful if you do not have any of the other utilities.

These programs can be entered directly into BASIC and compiled with most BASIC compilers (e.g., QuickBasic and TurboBasic) with only minor changes, if any. Lines that begin with a single quote (') are remarks and are used to explain the functions of the program. They can be entered as listed and are helpful for modifying the program later.

See the manual that came with your version of BASIC or your compiler for information about entering these programs or compiling them.

LFR. BAS

As we'll see in part C, DOS places a linefeed character along side the carriage return character at the end of every line or paragraph in an ASCII text files. The Macintosh needs to have only a carriage return at the end of a line or paragraph. The following simple PC BASIC program can be used to eliminate those linefeeds from a text file.

```
5 ' LFR.BAS - Program to add linefeeds to an ASCII (text) file.
6 ' © Copyright 1986 by Eric Alderman
7 ' Remarks © Copyright 1987 by Stephen L. Michel
9 ' clear some memory, clear the screen, turn off the function keys
10 CLEAR:CLS:KEY OFF
20 PRINT "Program to strip LFs from text files":PRINT
25 ' Get the name of the file to convert
30 PRINT "Hit RETURN to quit":PRINT
40 INPUT "Input file name";INF$
45 ' If the user hit RETURN—i.e., if the variable INF$ is empty
46 ' then quit the program
50 IF INF$ = "" THEN 200
55 ' Get the name of the file to write
60 INPUT "Output file name";OUTF$
65 ' open the file to convert as "i"—an input file
70 OPEN "i",1,INF$
75 ' open the file to write as an output file ("o")
80 OPEN "o",2,OUTF$
85 ' get the length of the file
90 LNFL = LOF(1)
100 CLS
105 ' Read a line from the file
110 LINE INPUT#1,LIN$
115 ' if its the end of file ("EOF(1)") then stop the converting
120 IF EOF(1) THEN 190
121 ' BASIC reads from a file until it:
122 ' 1) reaches a carriage return or
123 ' 2) has read in 255 characters
124 ' if it has read 255 characters, there is no carriage return, so
124 ' write the characters to disk
125 IF LEN(LIN$) = 255 THEN PRINT#2,LIN$;:GOTO 140
126 ' BASIC does not make the linefeed & carriage return
127 ' part of the text it has read in, so you don't need
128 ' to remove it. Add the carriage return to the text
129 ' when you write it to disk.
```

```
130 PRINT#2,LIN$;CHR$(13);
135 ' determine where in the file we are
140 POSFL=POSFL+LEN(LIN$)+2
145 ' calculations to determine the percent of the file completed
150 DONE=INT(POSFL/LNFL*100)
160 DONE$=RIGHT$(STR$(DONE),LEN(STR$(DONE))-1)
165 ' Tells the user how much of the file is processed
170 LOCATE 1,1:PRINT "Processing file: ";INF$;" - ";DONE$;"% complete."
180 GOTO 110
185 ' Close the files, and go back to find out if there are any
186 ' more files to be converted
190 CLOSE: GOTO 10
195 ' no more files—close the files & quit
200 CLOSE : SYSTEM
```

LFA.BAS

When Macintosh programs save text only files to disk, they put a carriage return at the end of every line or paragraph (see part C). DOS, however, needs to have a linefeed character along with the carriage return to determine the end of a line. This program adds the linefeed characters.

All lines perform approximately the same functions as in LFR.BAS. Only those lines that are different in function are here explained.

```
5 ' LFA.BAS - Program to add linefeeds to an ASCII (text) file.
6 ' © Copyright 1986 by Eric Alderman
7 ' Remarks © Copyright 1987 by Stephen L. Michel
10 CLEAR:CLS:KEY OFF
20 PRINT "Program to add LFs to text files":PRINT
30 PRINT "Hit RETURN to quit":PRINT
40 INPUT "Input file name";INF$
50 IF INF$="" THEN 200
60 INPUT "Output file name";OUTF$
70 OPEN "i",1,INF$
80 OPEN "o",2,OUTF$
90 LNFL=LOF(1)
100 CLS
110 LINE INPUT#1,LIN$
120 IF EOF(1) THEN 190
125 IF LEN(LIN$)=255 THEN PRINT#2,LIN$;:GOTO 140
126 ' Since BASIC does not make the carriage return part of the
```

```
127 ' string that was read in, it is necessary to put both it and the
128 ' line feed character at the end when reading it out.
129 ' The carriage return is ASCII 13, the linefeed ASCII 10.
130 PRINT#2,LIN$;CHR$(13);CHR$(10);
140 POSFL = POSFL + LEN(LIN$) + 2
150 DONE = INT(POSFL/LNFL*100)
160 DONE$ = RIGHT$(STR$(DONE),LEN(STR$(DONE)) - 1)
170 LOCATE 1,1:PRINT "Processing file: ";INF$;" - ";DONE$;"% complete."
180 GOTO 110
190 CLOSE:GOTO 10
200 CLOSE
```

HyperCard

HyperCard, introduced by Apple Computer in August 1987, is something of a jack-of-all-trades utility. It can act as a paint program, a database, or a programming environment. Like BASIC on DOS machines, it is included with every computer and provides an easy mechanism for users to write their own programs. It can perform some of the same functions as BASIC—it wouldn't be too hard to write the same linefeed utilities in HyperCard. However, since there are other Macintosh utilities that do the same thing, that really doesn't make much sense. Also, HyperCard is fairly slow at the kind of raw input/output with text files, so these utilities are impractical.

Instead, we'll talk a little about importing data to HyperCard. This material probably should have been included in part D when we look at databases, but HyperCard is a little uncomfortable in that realm.

HyperCard includes no built-in functions for importing data from text fields. It does, though, include a nice mechanism for importing MacPaint (or compatible) bitmaps, so we do not need to talk about that.

Listing 4.1 a sample HyperCard script that can be used to import text files that are tab-delimited (i.e., tabs separate the fields in the file and returns separate the record—see part D for a more complete discussion of this). To use this script, create a button in any HyperCard stack and type the script into it. The lines that begin with "—" are remarks (like the rem statements in BASIC). You do not need to enter these lines if you don't want, but they might help when you are debugging or modifying the script. Essentially, this script creates a moderately complete HyperCard document, which you can modify for your own HyperCard purposes.

Listing 4.1 A script for importing a tab-delimited file and making a HyperCard stack out of it.

```
on mouseUp
   global inFile
   global error
   newStack
   show message box at 33,296
   getFile
   if error is not empty then
      exit mouseUp
   end if
   goodFile
   if error is not empty then
      exit mouseUp
   end if
   createStruct
   importText
   choose browse tool
end mouseUp

on newStack
   — Creates the new stack
   doMenu New Stack . . .
end newStack

on getFile
   global inFile
   global error
   — finds the file to import from
   ask "Read from what file?"
   if it is empty then
      put "true" into error
   else
      put empty into error
      put it into inFile
   end if
end getFile
```

continued

Listing 4.1 (cont.)

```
on createStruct
  global inFile
  global error
  global countFields
  put 0 into countFields
  open file inFile
  put "Now analyzing the structure & counting fields . . . "
  put "dummy" into it
  — simply loops through the first record until it hits a return
  — and adds one to countFields every time it finds a tab
  repeat until it is empty
    read from file inFile until tab
    add 1 to countFields
    if it contains return then
      exit repeat
    end if
  end repeat
  close file inFile
  put "There are " & countFields & " fields in the file"
  domenu "background"
  open file inFile
  — now we are creating the structure, naming the fields,
  — and placing them.
  — show the data in the first record as the field name (in case
  — the field names in the first record & to help
  — in remembering what each field contains)
  repeat with x = 1 to countFields – 1
    read from the inFile until tab
    if it contains tab then
      delete last character of it
    end if
    ask "Please type a name for field" && x with it
    put it into fieldName
    newField fieldName
  end repeat
  — this does the same for the last field
  read from file inFile until return
  ask "Please type a name for field" && countFields with it
  put it into fieldName
```

continued

Listing 4.1 (cont.)

```
   newField fieldName
   — all done!
   domenu "background"
   close file inFile
end createStruct

on importText
   global inFile
   global error
   global countFields
   open file inFile
   put "dummy" into it
   put 1 into recNum
   — loops through the cards & fields until the end of file.
   — this usually creates one or two extra cards, but it
   — was easy to code.
   put the seconds into startTime
   repeat until it is empty
      put "Now on record #" & recNum
      repeat with x = 1 to countFields — 1
         read from file inFile until tab
         — this line removes the tab
         delete last character of it
         repeat while it contains quote
            delete char offset(quote,it) of it
         end repeat
         put it into field x
      end repeat
      — now do the same for the last field
      read from file inFile until return
      put it into field countFields
      domenu "new card"
      add 1 to recNum
   end repeat
   — all done!
   put the seconds into stopTime
   close file inFile
   put stopTime - startTime into timeTook
   divide timeTook by 60
```

continued

Listing 4.1 (cont.)

```
   answer "Done!" && recNum && "records written" with "Ok"
end importText

on goodFile
   — checks to make sure the file exists.
   — HyperCard automatically creates a new file when you
   — issue the open command. So we read from the file
   — to see if there is anything in it.
   — Be sure to delete any excess files created by this.
   — You can probably type in a path name to a file, if it is not
   — in the same folder as HyperCard.
   global inFile
   global error
   open file inFile
   read from file inFile until return
   if it is empty then
      put "That file does not exist"
      put "true" into error
      close file inFile
      exit goodFile
   end if
   close file inFile
end goodFile

on newField fieldName
   — this routine simply creates the field.
   — this lets us add to a background that already contains fields.
   put the number of background fields into numFields
   put numFields + 1 into thisField
   put "click where you want the field to go"
   wait until the mouseClick
   put the mouseLoc into here
   domenu "new field"
   put first item of here into right
   — fixed current length of field on screen
   add 150 to right
   put second item of here into bottom
   add 20 to bottom
   put here && ","&& right && "," && bottom into thePlace
```

continued

Listing 4.1 (cont.)

```
   — these can be changed for different field defaults
   set rect of background field thisField to thePlace
   set style of background field thisField to rectangle
   set name of background field thisField to fieldName
end newField
```

This script has been fairly well tested, but it is possible that there are situations it won't handle. In any case, it is presented as a sample of what HyperCard can do and how to do it. With some work, you could modify it to handle DIF, SYLK, or comma-delimited fields.

In addition to the software products mentioned in this part, there are a number of public domain or shareware products available that can perform a wide variety of translations. User's groups are a great source for these kinds of utilities, as are bulletin boards and information services such as GENIE or CompuServe.

Part C *WORD PROCESSING*

Virtually everyone with a computer does some word processing. And these days, almost everybody who has PCs and Macintoshes needs to move files created by their word processors from one machine to the other. Perhaps you want to incorporate a WordStar file into a ReadySetGo document on the Macintosh. Or perhaps you want to take a file that was created in MacWrite and edit it with MultiMate on the PC. This chapter will tell you how to make these transfers. Obviously, we cannot cover all contingencies. With the number of word processors around, it would take something akin to an encyclopedia to tell you how to move files from each and every program to each and every other program; the encyclopedia would be out of date as soon as it was published. Instead, we'll discuss general principles, with specific asides for the most popular and powerful programs. If you cannot find a specific answer in here, try some of the general principles.

The Problem

You may not realize it, but two kinds of information are stored with the documents produced by your word processor.

The first is the raw text that you've typed at your keyboard. Without thinking about it, you may have assumed that it is this text that is stored in your document file, and nothing more.

The second part consists of the formatting instructions that tell the program how you want the text to appear on the page. This information includes, for example, the instructions you have given that tell the program to underline a particular word or to format the document with right-justified margins. Typically, these instructions will be included in the document as embedded codes. When you bring the document onto the screen or print it out with your word processor, the program interprets these codes and follows them as it prepares the document for editing or printing.

The problem arises when you try to transfer a file from one program to another. If you have used more than one program for word processing, you have an idea of what is involved here. For example, in WordPerfect on the PC, you hold down the shift key and press the F6 key to center a line of text. In WordStar, you use a different set of commands. Just as you instruct the programs to center that line in different ways, they store that instruction in different ways. That's why, when you open a WordStar document in WordPerfect, it does not appear with formatting intact. As we shall see, a standard exists for encoding or interpreting the standard alphanumeric characters, but not for formatting the instructions.

This problem is compounded when transferring files between the Macintosh and the PC. Besides supporting the ability to show and print an a or a boldface **a**, the Macintosh can format that a in a number of other ways (e.g., shadowed, outlined, bold and shadowed, and outlined and bold). Furthermore, it can display that a in any number of different typefaces or fonts.

In this part of chapter 4, we'll discuss these problems in more detail and provide some guidelines on how to overcome these incompatibilities between programs running on the two machines. While we'll discuss some programs in particular, we'll try to do it in a generalized way that should give you enough information to attempt file transfers of almost any program.

The Levels of Compatibility

Basically, there are three levels of compatibility between programs created on the two machines. While we'll discuss them in detail, a short summary of each follows in order of increasing power.

ASCII ASCII stands for American Standard Code for Information Interchange and is pronounced "ASK-E." (Although that pronunciation seems natural to me, I have encountered people saying it in more ways than most of us can imagine.) Everything stored on a computer disk is represented, of course, in the form of a number. Virtually all programs agree on what constitutes an a and most other alphanumeric characters. ASCII consists of a set of numbers, from 0 through 128. Each letter of the alphabet, and each numeral, as well as many punctuation marks and symbols, are assigned a number in the ASCII format. For example, in ASCII, a lowercase a is represented by the number 63, a b, by 64, followed by all the lowercase numbers, the uppercase numbers, and punctuation marks and symbols.

> ***Note*** Both the PC and the Macintosh also use extended ASCII codes—numbers between 128 and 255—to show text characters that are not defined in the standard ASCII character set. This compounds the problem of converting files, especially those files that employ foreign language characters. Appendix B shows the full character sets of both computers.

Interchange or Common File Format Here one program saves data in a format that is not its native format (i.e., the standard manner in which it saves data). In other words, it allows you to save it in a different format, in the same way most word-processing programs allow you to save only the text portion of the document. The destination program can then read that format and interpret it. The leading standard file format is DCA, for document conversion architecture. This is an IBM file format, supported by many programs that run under DOS. Typically, programs from IBM (e.g., the DisplayWrite series) use this format as their native function. Another file interchange format is RTF from Microsoft. (RTF stands for rich text format.) This is a new standard that is supported right now by only Microsoft Word 3 on the Macintosh, although more programs are becoming available that work with it.

Most of the leading file translation facilities—DataViz' MacLink, the TOPS Translators, and the Apple File Exchange program—support DCA transfers to MacWrite. Once the file is in MacWrite format, you can read it into most other Macintosh word processors.

Direct Compatibility This is the easiest to work with and the ideal solution. In this case, a program can read (and usually write) the format of another word processor directly. Microsoft Word 3 on the Macintosh, for example, can read (and write) files in MacWrite, Microsoft Word 1.05, and DOS Microsoft Word files directly. WordPerfect on the Mac can directly read DOS WordPerfect files. Obviously, this is the best solution, since it is the most transparent to the user: You do not have to worry about the formatting or go through exchange procedures.

These three levels of word-processing compatibility do not address some of the things you'll come upon in transferring word-processing files, such as separate Convert programs that are available to handle many file formats. When using those programs, though, you usually work a variation on one of these themes.

General Tips

Before we discuss file formatting in detail, here are a few tips for working with documents that are to be transferred from one program (or computer) to another.

Format for Your Destination

When using a word processor, we do not always write with the idea that we are formatting for the final output. That is, our documents are sometimes written using one program and formatted and printed using another. An example is the document that is written to be layed out with a desktop publishing program. In this case, it is important that we understand which formatting is appropriate to insert with the word processor and which is appropriate to do with the desktop publishing program. Generally, anything that affects how the characters will appear on the page is best to do with a word processor (e.g., boldfaced or underlined words). Formatting that we should leave to the desktop publishing (or other final layout program) includes such things as columns on a page, margins, headers and footers, and perhaps font selection. By doing this, we can eliminate some of the confusion that might result from losing some of these instructions during the file transfer process. Also, if you are working on the Macintosh, most desktop publishing programs (and virtually all PC word-processing programs) will not accept graphics pasted into the text. Consequently, you should leave them out.

In the same manner, you should be aware of what formatting is supported by your destination program. If you are writing in WordPerfect, for example, and you know that you are going to use DCA to pass the text to Microsoft Word, be aware that DCA does not support a double-underlining format (even though Microsoft Word does), so there's no point in using it. That formatting won't transfer.

Don't Use Spaces for Alignment

Another piece of advice, which might appear to go against the grain of what was just said, is to use some of the built-in function of the word processor to do some routine formatting. This is most important when it comes to lining up columns with tabs. Generally, you should not use spaces to line up columns of text on the page. You can easily get into this habit, especially when working on the PC.

On the PC, all text is presented on the screen (and most printers) in a monospaced font, as opposed to a proportionally spaced font. That is, each character on a line is given the same width on that line. In a proportionally spaced font, each character is allotted space according to its width. A w, for example, is given more room on a line than an i. Most Macintosh fonts are proportionally spaced. It gives text on a page a more pleasing appearance and is easier to read. The problem is, spaces are handled in much the same manner. If you fill up a line with spaces (to make columns line up), the characters on the line below will probably not have the same spacing values. When the printer (or program) is then formatting proportionally, it won't know how to allocate widths to the spaces. Your columns won't be aligned on the paper or when you transfer the document to another program that does support proportional spacing.

Virtually all transfer formats support at least left-aligned tabs. Left-aligned tabs are the default in most programs and are like typewriter tabs: The left edge of tabbed words or numbers line up with the tab stop. By using tabs, the word processor knows where on the page the tab will be set and can start printing your column at that point, regardless of the width of the characters that come before or after it.

Using tabs also makes it easier to redesign your columns: instead of having to remove spaces from what might be lengthy columns of spaces, you have only to reset your tabs.

On the Macintosh, you can get around this by formatting documents that have used spaces in a monospaced font. Monospaced fonts, such as Monaco and Courier, allow each character the same width on the page. You sacrifice some of the clarity of the document for the ease of keeping the columns lined up. One suggestion is to format only the columnar material in Monaco or Courier, while leaving the rest of it in a proportionally spaced font.

Some programs, unfortunately, subvert your good intentions here. WordStar 3.3 and Q&A Write, for example, do not insert tab codes when you press the tab key. Instead, they insert the required number of spaces into the text to make it line up with your tab stops.

ASCII Text Files

As already mentioned, ASCII is the lowest common denominator in dealing with files. ASCII files are also commonly known as "text files," "non-document files" (in the case of WordStar), or "non-formatted files." Virtually every word processor I have ever seen supports some kind of saving and retrieving of standard ASCII files. If yours does not support ASCII, chances are it won't support any of the other formats. I suggest getting rid of it and finding a new word processor.

See appendix B for a chart of the ASCII characters for both the PC and the Macintosh. PC ASCII codes are fairly standard. Macintosh ASCII charts, however, can vary somewhat depending on the font used. The table in appendix B uses the default Geneva font for the Macintosh, so it's as standard as it gets.

An ASCII file consists of nothing more than the characters you typed that represented the text you wanted to print. With only a couple of exceptions, ASCII files contain no formatting codes or commands. Typically, the only nonprinting or nonvisible codes that will be inserted into an ASCII file are the carriage return/linefeed codes that end a line or a paragraph and, sometimes, tab codes (although not, of course, tab settings).

PC DOS ASCII Files

PC ASCII files can vary in several important respects. There is a standard manner and a generic word processor manner (to borrow a phrase from WordPerfect).

A standard DOS text file usually consists of a stream of characters less than eighty characters (the width of the standard screen) long, followed by a carriage return/linefeed pair. The carriage return /linefeed pair is a holdover from the days when computers addressed their screens as Teletype or typewriter-like devices. (Indeed, the first computer input/output devices were Teletypes.) A carriage return character (ASCII character 13) was needed at the end of the line to tell the Teletype that the next character should appear on the left edge of the page. A linefeed character (ASCII character 10) was needed to instruct the Teletype to push the paper through so that the text was printed on the next line.

You can see this in effect when you type a document to the screen from PC DOS. If the file is not in the standard ASCII format (i.e., if there is no linefeed at the end of the line), new lines will start at the left edge of the screen, but they will overprint preceding lines. Just as the Teletype did, DOS needs the linefeed character to move to the next line. As we'll see, the Mac does not do things this way. When we open DOS text files on the Mac, we need to strip these linefeeds.

With most DOS word-processing programs, when you save files in ASCII format, they append a carriage return/linefeed pair at the end of every line and two at the ends of paragraphs. The assumption is that you are not going to use this text file in a program that can handle its own word wrap, that it needs these characters at the end of the line. This is a nuisance when you are transferring the text into another word-processing program. Carriage return/linefeed combinations are interpreted by most programs as being codes for the end of the paragraph. So, in the destination program, you have a document composed of many paragraphs, each only one line long! Fortunately, as we'll see, it is not too hard to deal with this.

When you create a standard DOS text file it does not store the tab character (ASCII character 9). Instead, most word processors or other programs will pad the tab stops with spaces (i.e., they will insert enough spaces into the document to make the columns line up). As already mentioned, this is unfortunate and can cause extra work when you are transferring an ASCII file into a program that supports proportional spacing.

WordPerfect, MicroSoft Word, MultiMate, and some other DOS programs allow you to save files in a generic word processor format (although it may be referred to in different terms). This is an ASCII file that addresses some of the problems mentioned above and makes it easier to transfer word-processing documents.

Files saved in a generic word processor format do not include a carriage return/linefeed combination at the end of every line. This combination is only inserted at the end of paragraphs, making editing much easier. WordPerfect's generic word processor format uses only a carriage return at the end of a paragraph.

Additionally, this formatting convention inserts a real tab character—no spaces—where tabs are. Since most word-processing programs use the tab character to mean a tab, this helps when editing columns and when dealing with proportional spacing.

Macintosh Text Only Files

Macintosh text only files are almost exactly like the DOS generic word-processing files. Since the Macintosh is a graphics-based computer, it has almost completely divorced itself from the heritage of the Teletype and does not need the carriage return/linefeed combination to indicate the end of a line of type. The text-editing routines built into the Macintosh ROM Toolbox give to virtually every program wordwrap and other text selection and editing powers.

DOS Text Files on the Macintosh

Since DOS text files can vary somewhat from Macintosh text files, we must convert them before we can use them to their best advantage on the Macintosh. Basically, this gets down to stripping the linefeed character from the carriage return/linefeed combination that many DOS programs insert at the end of every line of text and removing the carriage returns from the end of every line. See figure 4.5 for an example of what a DOS text file looks like when it is opened with MacWrite. The small boxes signify linefeed characters that DOS inserts after the carriage return character to end a line.

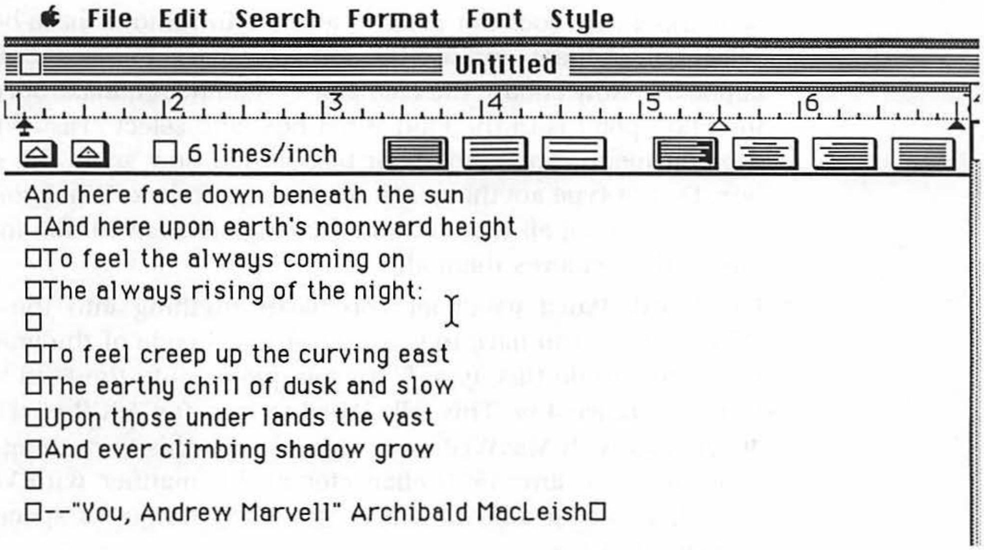


Figure 4.5 A text file created on an IBM displayed in MacWrite.

There are several ways to remove these linefeed characters.

1. Use the TOPS DA to remove carriage return/linefeed combinations. When using the Copy button in the TOPS Desk Accessory, pressing the option key brings up a dialogue box (see figure 2.2) that asks if you want to copy all the data that is in a file or the text only. Selecting "Copy Text Only" instructs TOPS to remove many of the control characters that might be part of the file (see chapter 2B for a discussion of this option). Only the linefeed characters are removed; the carriage returns, which will still need to be changed, remain.
2. Remove the linefeeds with AppleShare PC. As discussed in chapter 3, AppleShare PC allows you to map certain extensions on the PC so that these linefeeds are removed automatically.
3. Use TOPS Translators, Apple File Exchange, or MacLink Plus: These programs automatically remove the linefeeds from the paired sets. If you use any of these programs to translate files, then it is handled for you.
4. Use MacWrite. See figure 4.5 for an example of how a text only document in MacWrite looks with the linefeeds visible as small boxes (□). (The box character is a standard Mac character for

something that does not print in a font.) To remove these boxes, select one of them, and use the copy command to place it on the clipboard. Now choose the change . . . command, make sure the insertion point is in the Find What box, and select "Paste" from the edit menu. The Find What box should now show the small box. Do not type anything into the Change To box. What you are doing is finding all the boxes and changing them to nothing. In effect, this removes them all.

5. Use Word. Word won't let you paste anything into the Find What box, so you have to type in the ASCII code of the linefeed character. To do this, type "^10" (no quotes) into the Find What box (see figure 4.6). This tells Word to look for ASCII character 10. Just as with MacWrite, you can change this to nothing. You can search for any ASCII character in this manner with Word, as well as being able to search for such things as spaces or carriage returns.

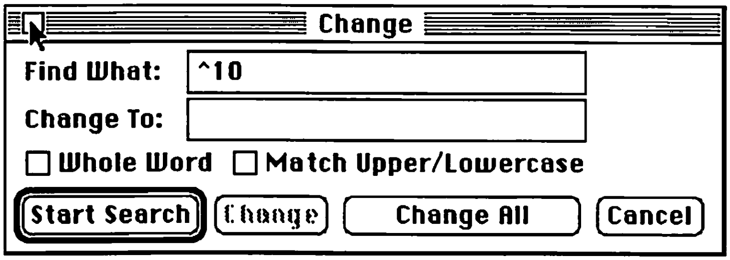


Figure 4.6 In Microsoft Word 3, you can search for a specific ASCII code by typing in the caret (^) followed by the ASCII number.

Remove the Excess Carriage Returns As mentioned above, many DOS programs will place a carriage return at the end of every line of text, which can be a real nuisance for working with the text in a word processor or desktop publishing program. Removing the excess carriage returns is a little harder. You do not want to remove all the carriage returns, since you want to retain the paragraph structure of the original document. You want to go from making every line a paragraph to retaining the original paragraphs.

Usually, when you have created a document on the PC, you have entered two returns to separate paragraphs: one to end the last line of the paragraph and one to add a blank line between the paragraphs, as on a typewriter. If this is the case, removing the excess carriage returns is easy.

The way to do this is to first search for all occurrences of two carriage returns in a row. In Word, you do this by typing "^p^p" into the Find What box of the Change . . . dialogue box. (^p is how you tell Word to find a carriage return. The "p" stands for paragraph.) Now replace all these with something that won't occur elsewhere in the text. I use "zzzz." Now search for all the carriage returns (^p alone this time) and change them to nothing, to delete them. After this has been done, your document should be pretty messy: There will be no carriage returns at all, so that it appears as one big paragraph on the screen. To fix it, search for all the occurrences of "zzzz" (or whatever you changed the two carriage returns to) and replace them with a single carriage return. In Microsoft Word, this can take quite a while to do, but when it's finished, you should have an editable document.

Since there are several word processors available on the Macintosh, I cannot show you exactly how to do this for all of them. This technique should work for most of them, as long as they allow you to either paste what you are searching for into the Find What box or, as with Word, specify the ASCII character you want to find.

Macintosh Text Files on the PC

Using Macintosh text only or ASCII files on the PC is essentially nothing more than reversing the two steps already mentioned. Since the Macintosh does not pair linefeeds and carriage returns, they must be added to the file, preferably before it is transferred to the PC. Typically this shows up on the PC when your document being converted to one long paragraph, which is definitely an undesirable effect!

Unfortunately, you can easily do this in many ways. Most DOS programs simply ignore an unpaired carriage return and strip it from a file as they read it in. WordPerfect and Microsoft Word work this way. Here are a couple of suggestions on how to do this.

1. Using your word processor, modify the file on the Macintosh before sending it to the PC. This technique is similar to that used

to remove excess carriage returns. Simply search for all the returns in your document and replace them with a unique string of characters that won't be found as part of the normal text. When you load the file into the PC, you then reverse the process, replacing the characters with a return.

This does not work well with very long documents. Working the process on both ends can take a long time. If you're not careful in choosing your dummy characters to represent returns, your text will become a real mess.

2. Use a public domain program. Many public domain programs and utilities for both the PC and the Macintosh can perform this operation.
3. Use the WordPerfect Clipboard. When opening a text file, WordPerfect strips out any returns that are not paired with linefeeds, thus losing all the paragraph markers in your document. If you use WordPerfect Library, however, you can use the clipboard function.

   For some reason, the WordPerfect clipboard changes unpaired returns to paired linefeed/carriage returns when you open a text file from the clipboard option of the library main menu. After putting the text file on the Clipboard, enter WordPerfect and retrieve it from the Clipboard. Again, this works best with short documents, since there are limits to what the WordPerfect Clipboard will hold.
4. Use the MultiMate conversion utility, which is one of the best available. It allows you to specify that the source file (or destination file for transfers the other way) use only a carriage return and does not pair it with a linefeed. If you are using MultiMate, this is obviously the way to go.

Interchange or Common File Formats

Beyond ASCII, which has little or no provisions for storing any kind of formatting information, programs can use common file formats to transfer files to one another. Typically, a program will not use one of these formats (e.g., DCA) in which to store its own files. Instead, it will have a Convert program or a conversion option that can be used to modify the file to put it into one of these formats.

Basically, the industry—software manufacturers, users, and programmers—is crying out for a good, flexible standard for storing formatted text. As of this writing, there is one clear standard method, a couple of new standards that manufacturers are trying to support, and several de facto standards. Unfortunately, the prospects for universal adoption of a file format are not likely: Manufacturers hesitate to adopt standards proposed by their competitors. Any standards are likely to be at least somewhat restrictive, not supporting all the features that can be part of a word-processing program.

These are pivot formats. That is, only a few of them are used in the native mode by word processors on either the PC or the Mac. Rather, a file is converted by the source program (or accompanying utility) from the native format into the pivot format. The destination program then can translate it into its own format. As we'll see, something is lost with each of these translations.

Using pivotal exchange formats takes the burden for writing translation utilities off the shoulders of software developers. Instead of the Macintosh software authors having to write a conversion from every leading PC program, for example, they need only write programs that convert from the pivotal architecture. This reduces the authors' work, improves standard file transfers, and in general makes life easy for everybody.

Document Conversion Architecture

IBM's Document Conversion Architecture (DCA) is the leading formatted text architecture of today. DCA was introduced in 1980 as an internal IBM standard allowing interchange of text among the many different IBM word-processing systems, from their DisplayWriter word processor to their mainframes. DCA is not the native format of any of IBM's word processors. Rather it is a pivotal or intermediary file structure.

There are two types of DCA available: DCA Revisable Form Text (DCA RFT), which is designed for the exchange of documents that will be altered after they are received, and DCA Final Form Text, which, as you might guess, is for those that won't be edited after translation. Most microcomputer systems use DCA RFT almost exclusively, so that's what we'll talk about.

The good news is that almost every leading MS DOS and several leading Macintosh word processors feature conversion utilities that can read and translate DCA files. Additionally, many widely available translation facilities (e.g., those from Apple, TOPS, and DataViz) can work with DCA, giving support for almost every word processor available.

The bad news is that DCA is an aging standard, devised before the wide use and acceptance of such things as laser printers, mixed text and graphics, and desktop publishing. Many modern word-processing programs feature a wide range of options that DCA does not support. When creating a document that is to be used on a system different than the one it is created on, consider which native features of a program might be supported or not supported in a destination program.

Shortcomings

Some of the features that DCA either does not support at all or supports in a limited or nonstandard manner, follow.

Double Underlining Many word processors (e.g., WordPerfect, Microsoft Word, and MutliMate on the PC and Word 3.0, PageMaker, and FullWrite on the Mac) support double underlining. DCA RFT has no provision for double underlining, so most conversion programs will convert this to single underlining.

Bold Printing This works fine with most DCA conversions. However, IBM supports bold printing in a different way in DisplayWrite 3 and DisplayWrite 4 than in earlier versions of that PC program. Typically, this does not affect your work, as most DCA conversion programs support the newer method.

Headers and Footers DCA RFT can support up to four pairs of headers and footers: two even headers and footers and two odd headers and footers (i.e., two each for even and odd pages). Most Macintosh programs support but one header and footer per page, although it can have multiple lines.

Shadow Print Some PC word processors (e.g., MultiMate) support shadow printing. The Macintosh, too, has an option for shadow printing, although it is handled in quite a different manner than that of

MultiMate. But DCA RFT has no shadow options, so this is usually converted to bold text when translated.

Overstrike Many professional word processors feature automatic overstrike of characters to print, say a slash (/) on top of regular text to indicate it has been struck out. This is handy to indicate sections of text that are to be deleted, when many readers must approve a document. DCA does not support overstrike directly. Some programs, when converting into DCA will print the original character, then a backspace character followed by the slash. The Macintosh has no backspace character, so these are printed as boxes in most fonts.

Font Change DCA does not support Macintosh font names.

Pitch or Font Size DCA allows only one font size per line. Each line is formatted according to the pitch of the first character on the line.

These are some of the major shortcomings of the DCA RFT formatting standard. As always, standards will leave something to be desired and will promise something more than they can deliver. Various manufacturers will implement them in different ways. Still, DCA represents a major improvement over ASCII as a medium for transferring documents between applications.

Most Macintosh programs will neither read nor write DCA files directly. Instead, you can use a conversion utility to convert DCA files to MacWrite files. Since most Macintosh word processors feature a way to read MacWrite files, you can then use them in your word processor. Additionally, Microsoft provides a program called "DCA Convert" with Word. This converts DCA files into Microsoft's RTF, which Word can then work with.

Rich Text Format

With the release of Microsoft Word 3.0 in early 1987, Microsoft also announced a new document transfer specification. Called Rich Text Format or RTF (despite the potential for confusion between Rich Text Format and DCA RFT, we will hereafter refer to this specification as RTF), the MicroSoft specification is much more flexible than DCA. Only time will tell whether or not RTF will come to replace DCA as the dominant conversion architecture. At this writing, we cannot know how prevalent this standard will become.

Not surprisingly, RTF coding is closely aligned with the kind of formatting that Microsoft employs with Word. That is, RTF coding is based on the separate elements of Word documents: sections, paragraphs, and characters. Given this, it might be difficult for manufacturers of other word-processing programs that format in a different manner to implement RTF for their products.

RTF lives up to its name by being a rich standard for coding text. It can include information about font name, size, and style, as well as bitmapped pictures, and it can employ style sheets to assist in flexible and consistent formatting.

De Facto Standard Document Conversion Formats

Aside from the ASCII, DCA, and other pivot formats, there are two de facto standard formats for the PC and the Macintosh. On the PC side, there is WordStar; on the Macintosh side, there is MacWrite.

WordStar

WordStar, from MicroPro International, was one of the first successful word-processing programs available for any machine. Its original implementation was for the 8-bit CP/M operating system. Shortly after the introduction of the IBM PC, WordStar became available for DOS as well. Since the PC implementation featured the same commands and user interface as the very successful CP/M implementation, success for the product was assured. Many people automatically went to WordStar as they upgraded their systems and the wide variety of peripheral products (i.e., instructional books) applied to WordStar on the PC as well as CP/M.

Given WordStar's wide acceptance, it was destined to become a standard for file interchange formats as well. Virtually every word-processing program for the PC that features a convert utility can read and convert WordStar files. And virtually every utility for the Macintosh that converts files to DOS formats will transfer files to WordStar. Like DCA, it can act as a pivot in the file transfer process: Convert your Mac-formatted document to WordStar format, then use your DOS program's convert facility to convert from WordStar format.

As with most of the transfer formats we've discussed, WordStar does not support the full range of features used by modern word-processing programs. Here are some of the problems encountered when using WordStar to convert documents. Unless otherwise noted, this discussion refers to WordStar version 3.3, not to the recently introduced version 4.0. Most file transfers will use the version 3.3 formatting codes.

Tabs One of the big problems with WordStar is that it does not use tabs. Even when typing text into WordStar, when you strike the tab key, it does not insert a tab character. Instead, it inserts the number of spaces necessary to line the text up with the tab setting. This gets problematic when you are using the WordStar format to transfer files from the PC to the Macintosh. As discussed earlier, lining up columns with spaces does not contribute to good looking text when different font sizes and proportional spacing are used. It also makes it harder to reformat text for changing margins or tab settings. In the destination program, those spaces must be converted to tabs. Another way around this is to format the document in a Macintosh monospaced font. When most Macintosh translator programs (e.g., MacLink or the TOPS Translator) process a WordStar file, they automatically put the document into the Monaco or Courier fonts, neither of which is proportionally spaced.

Fonts and Other Graphics-Related Features WordStar does not support the naming of different fonts nor will it include graphic items generated on the Macintosh when transferred to the PC.

Dot Commands Much of WordStar's formatting is accomplished with special dot commands that are placed at the beginning of a line of text. For example, a .HE command tells WordStar to insert a header. These dot commands are acted upon by the program as it is printing. There is little consistency in how conversion utilities will handle WordStar dot commands. Some conversion programs, such as the one that comes with the PC version of Microsoft Word, will ignore most of the dot commands when formatting the text. This Convert program will place three asterisks (***) in front of the dot command, allowing you to search for them. Other programs will try to interpret these commands and translate their formats into the destination program.

One way to search for dot commands—when they are not modified as with PC Word—is to search for a carriage return, followed by a period. Dot commands always appear at the left-hand margin on their own line of text, so a return will always precede them. At least this lets you move quickly through your document, searching for them and perhaps interpreting them yourself.

MacWrite

For the first two years of the Macintosh, Apple provided MacWrite free of charge with every machine. This allowed it to become the de facto text standard to the Macintosh, and virtually every Mac word-processing program will read and write MacWrite files. If you see one that does not, do not buy it. Many software manufacturers put some of their documentation or notes on disk in MacWrite format, and most transfer utilities support MacWrite. It is very handy as the Macintosh pivot program in transferring files.

While MacWrite is a very capable program, it too is showing its age. Many of the newer Macintosh word processors support a wide range of features that are not implemented in MacWrite (e.g., mixing graphics and text side-by-side on the same line, style sheets, and multiple columns).

However, in transferring PC documents to the Mac or vice versa, MacWrite is very capable. It supports many of the more common formatting conventions of both PC- and Macintosh-based word processors. Only when transferring very complex documents from one program to another do MacWrite's shortcomings become noticeable. Since it provides no provisions for such things as multiple columns, numbered lines, and footnotes, most of these formats will not transfer or will transfer incorrectly.

Direct Compatibility

The final level of compatibility is direct compatibility between Macintosh- and PC-based programs. As already mentioned, this is the most useful type of compatibility to have in the networked environment.

Microsoft Word

At this writing, Microsoft Word 3 is the only program that can directly read and write DOS-formatted text files. Since Microsoft makes versions of its word processor for both machines, this is no surprise.

However, Microsoft's file compatibility has some limitations. Microsoft Word 3 on the Macintosh can read files created by Word 3 on the PC (with much of the formatting used in Word 4), and it can write files that can be read directly by Word 4 on the PC. Word 4 on the PC cannot read or write files that have been saved in the "native format" of files created on the Mac; the Macintosh version must save them in a format specific to PC-based Word.

The two versions of Word also use style sheets in different manners. On the PC, Word saves a style sheet as a different document, with the extension .STY. On the Macintosh, the style sheet is saved either as part of the document or as a standard set of styles used by all documents created by the program. When reading a file created on the PC, Word 3 on the Macintosh asks if a style sheet has been used for the document, then it allows you to open that style sheet. The PC style sheet becomes part of the Macintosh document. When saving files from the Mac to the PC version, though, Word does not create a separate style sheet. Rather, it formats the document using the style sheet that is in effect. To incorporate or create a DOS style sheet, you must use PC Word's style sheet commands to insert the correct style information.

WordPerfect

Another program that supports close to the same formats on the different machines is WordPerfect. WordPerfect is unavailable as this is written. Preliminary copies of the program, as well as the announced intentions of WordPerfect Corporation, show that the Macintosh program will directly read files created on the PC. Since the Macintosh version will be able to include features not available on the PC (e.g., different fonts and sizes and mixed graphics), WordPerfect 4.2 on the PC has been written to simply ignore commands that are not appropriate to the machine. However, just as with Microsoft Word, WordPerfect for the Macintosh will need to save its files in a special format so that WordPerfect 4.2 on the PC can work with them. Many of the functions of Macintosh products (e.g., graphics and fonts) are simply too difficult to make a PC understand.

Conversion Programs

Most of the conversion programs available—including MacLink, the TOPS Translator, and the Apple File Transfer Utility—work in ways we've already discussed. Using them, typically a pivot format (e.g., DCA, MacWrite, and/or WordStar) will be used to translate files back and forth. Apple's File Transfer Utility, since it is modular, will be one of the most powerful. With this utility, software manufacturers will be able to write translation modules that the utility can use to write files directly to its native format. These programs all feature capable manuals that explain how to use them, so we won't go into the nuts and bolts here. Most support one or more of the transfer formats discussed in this chapter.

Beyond the conversion programs, you can convert documents in ways perhaps surprising to the manufacturers of those programs. One good example is in Aldus' PageMaker. We'll discuss some of the details of using PageMaker in part 4E, but its ability to save text-formatted text files bears some discussion here.

PageMaker 2.0 on the Macintosh can directly read files written by PageMaker on the PC. PC PageMaker version 1.0a (and higher) can directly read files created by PageMaker 2.0 on the Macintosh. Beyond merely allowing you to exchange PageMaker files on the PC, this allows the transfer of text (and graphics) between the two machines and some translation from one format to another.

To start with, PageMaker 2.0 on the Macintosh is adept at reading formatted files created by several word-processing programs on the PC. It can read files created by XyWrite, WordPerfect, and WordStar directly, over a network or after the file has been transferred to a local disk. PageMaker's rules for this—the only way it can know what program created the file—require that certain specific DOS extensions be used as part of the file name. The extensions that PageMaker requires for these files follow.

.WS for WordStar

.XYW for XyWrite

.WP for WordPerfect

When PageMaker attempts to place these files into a document, it uses the extension to discover what program created the file. It then knows how to read the formatting instructions from the program and applies correct formatting to the text it places. This works very nicely if you are using PageMaker on the Mac to lay out pages with files created on the PC. Most PC conversion programs feature some way of getting formatted text at least into the WordStar format.

However, we can make PageMaker do some of our file conversion bidding when using PageMaker 2.0's Export . . . command.

> ***Note*** The Export . . . command is not available in version 1.0a of PC PageMaker, although it might be added to future versions. This is a one-way transfer only.

The Export command allows PageMaker to save files out to Word 3.0 or Write Now (version 2.0a) format. Using this command, it is possible to place into PageMaker a WordStar, WordPerfect, or XyWrite file and save it to disk formatted for Macintosh Word 3 or Write Now.

Using PageMaker to transfer files is an outgrowth of its native ability to deal with files created on both machines. This example has been used not only to show how to use PageMaker itself to assist in formatted file transfer but also to illustrate how the uses of a program might go beyond what its creators had planned. As we use software, with an open mind and a sense of exploration, uses such as this one become more commonplace, and we find ways of doing things we previously thought to be difficult or impossible.

Mutiuser Word Processors

It is not uncommon for more than one person to want to work on the same file at the same time. That's one of the reasons we install networks. However, there are problems with this. For example, suppose that user A opens a particular file and is making changes to it. After he opens it, so does someone else. That person, too, makes changes to it and saves it to disk before user A does. When user A then saves the file, the changes made by user B will be overwritten. In an ideal software world, there would be some way that the software could be intelligent about this—perhaps by informing user A that user B had made changes and offering some way to allow user A to reject or accept those changes as part of the edited document. That would be ideal, but the software technology to support that is not available yet.

Instead, word-processing software that is network aware usually will lock out second users of a document. Microsoft Word 3 on the Macintosh is an example. When user A opens the file in Word, that file is then marked as a locked file for subsequent users. When they open the file, they are warned that they won't be able to save the changes to the same file. They will be able to save it, however, to a new file name. Later, the two sets of changes can be merged. WordPerfect on the PC works the same way.

Unfortunately, as yet no products work in a multiuser mode for sharing files between the Macintosh and the PC. Both Word and WordPerfect, the only two programs available today for both machines, use different formats for their PC and Mac versions and require, on the Macintosh, that you save the files in a PC-compatible format before the PC version can open them. At least that helps clear up some of the multiuser problems between the two machines.

Another solution, with TOPS, is to publish the folder containing the files in question as "one writer only." With such a volume, only the person who first opens that volume can make changes to the volume. To other users, it will appear as a locked volume; they will be able to look at the files, but they will not be able to save changes back to the same volume.

Part D DATABASES AND SPREADSHEETS

Database and spreadsheet applications are among the most common in the microcomputer world. Indeed, the first spreadsheet program—VisiCalc for the Apple II—was instrumental in bringing about the acceptance of the personal computer in business. Many of us use spreadsheets in everyday life, for everything from cost analysis to record keeping. We use databases, too, to assist us in everything from mailing list management and sales contact follow-up to personal reminder lists to elaborate order entry and inventory applications. Much of the information that is shuttled around our computers is of one of these two types. Many modern database programs have spreadsheet-like functions that make the insertion of calculations and sensitivity analysis possible. Most modern spreadsheets feature some database functions such as sorting ranges of cells and selecting particular records based on the values of specific cells. A few spreadsheet products, such as Lotus Symphony and Paperback Software's VP Planner, feature fairly complete database functions as part and parcel of their spreadsheet functions. As our computers and programs become more powerful, the distinctions between the two types of software are disappearing.

Database and spreadsheet files are similar. A database file consists of a set of records, which in turn are broken down into fields. A spreadsheet consists of cells in an array of horizontal elements (the rows) and vertical elements (the columns).

In a spreadsheet that is rectangular, each row is the same as a database record, and each cell in that row is about the same as a database field. Most database files can be loaded into spreadsheets—usually through the DIF or WKS pivot file format—as long as the file is not too large for the free memory available to the spreadsheet program. However, spreadsheets do not always have rectangular structures, so the reverse may not always be true.

Given these similarities in structure, the techniques for transferring database and spreadsheet information are much the same. Many of the pivot formats we discuss in this chapter are common to database and spreadsheet files.

Even though database and spreadsheet files are similar, they are also different. Database applications may not contain calculations that refer to other records in the database; spreadsheets almost always do. We'll use different techniques to transfer the different types of information. The following discussion starts with an examination of database information, with tips on how to transfer information between databases. It then leads into spreadsheets, with some tips on how to transfer data between spreadsheets and databases.

Databases

Transferring files between databases and between database programs and other programs that can manipulate, display, or analyze database information is a complicated topic. At times you might need to recreate in one database program all the functions that are present in another. This can involve much work, usually including some database programming. At other times, you may need only to transfer a database report into a program that can format and print it nicely. That is much easier.

As with some of the other topics discussed in this section, it would take a fairly large book to list all the possible formats that database programs use and all the possible ways to transfer their information. And, of course, it would be hopelessly out of date as soon as you bought it.

Instead, we'll examine some general hints that should work for many situations, with the major database products for both the PC and the Macintosh. Mostly, we'll discuss the various common file or pivot formats that can be used to transfer information between two database programs and between such things as databases and spreadsheets. As always, good familiarity with the programs you are dealing with, along with a general understanding of the structure of the data you want to transfer and your goals in doing the transfer, is the key to successful work.

Database Defined

A database is an ordered set of information. Typically, a database is a filing program that allows you to enter information in an organized manner and perform operations on that data, such as sorting, reporting, and making calculations. A database or filing program will usually allow you to design a form on the screen and enter your information into that form.

A database file is composed of fields and records. An understanding of these two terms is crucial to being able to manipulate database information, so some discussion is in order.

By way of example, let's discuss a name and address database, one of the simplest and most common applications. It might have a screen that looks like the one shown in figure 4.7. (Except for the graphic elements and fonts, a screen form of a PC database program is essentially the same.) This screen represents the database record. A record is all the information that is related to one form on the screen. Within the record, there are fields. The characters you type into "First Name," represent the first name field, the characters in the address box relate to the address field, and so on. This is a simplified view of a database. More complex structures are available, but this basic structure is common to all databases. They are all composed of fields and records. The trick in transferring information between database programs is to make sure that the structures of the two databases—the one that created the data and the one that is to receive it—are the same.

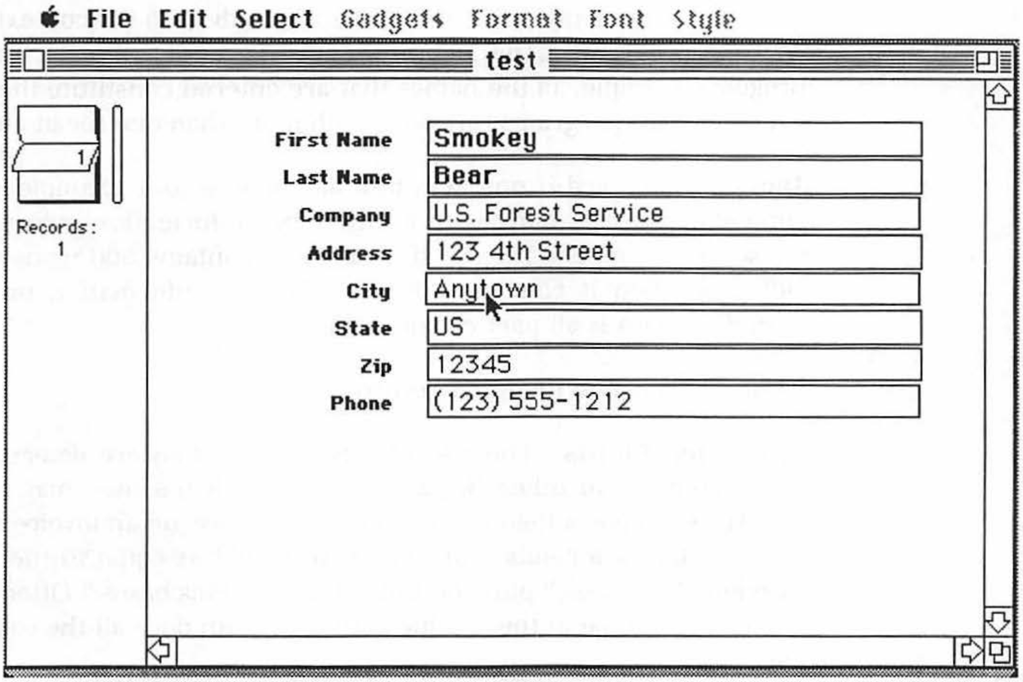


Figure 4.7 A screen form of a typical Macintosh database application.

Some Other Terms Defined

Many of these terms are discussed in appendix A, but here we'll discuss them in more detail.

Database This word can refer to both the filing program and the set of data created by it. The database or filing program is the one with which you design screen forms, enter the information, manipulate it, sort it, and report on it. The manipulation can include finding certain instances of the data (finding, say, all the "Bears" in our name and address example), doing calculations (determining how much Smokey Bear owes on all his past invoices), and more.

The second definition describes the information that is actually contained in the file created by the database. This includes both that information you type and that calculated by the program.

The context should make these two meanings clear.

File When working with databases, "file" has a distinct meaning. Generally, a file is any unified set of data on a disk. In the context of a database program, a file is a unified, ordered set of data. In the foregoing example, all the names that are entered constitute the file. Some database programs can work with more than one file at a time.

Record A record is one item in a database. In our example, each name and address, along with all the other information entered on one screen form, is a record. If a database contains 500 names and addresses, then it contains 500 records. The information on our example screen is all part of one record.

Field This is one item in a record.

Calculated Fields These are fields whose values are determined by the contents of other fields and into which the user may enter data. For example, a field called "Total Purchases" on an invoice form might contain a formula that defines this field as equal to the field "Previous Purchases" plus the field "Current Purchases." Often, the user does not type in these values—the program does all the calculations.

Relational This is a fairly loaded word in the database game. Generally, "relational" has come to mean the capability of a program to unite and cross-reference information contained in separate database files. (To the knowledgeable, this definition isn't adequate. This term has complex meanings that have changed over the years. I use it here in the sense in which it is generally used in the microcomputer industry and hope I'm not offending anyone.) Since a relational database contains separate files, for the purpose of transferring data, each of those files must be handled separately.

The Problem

As is the case with word processors, spreadsheets, and most other kinds of software, each programmer and manufacturer structures a program's data in a unique way. This allows that programmer to implement unique features in a program and to access the data quickly and in unique ways. The internal data structure that a program uses is usually one of the first things a programmer creates when planning software.

In the database realm, structuring data is further complicated by the fact that you have the power to set up your own data structures. Given a common problem (e.g., tracking inventory), no two database users are likely to use exactly the same methods to organize information. In a typical example of a database that is used to track potential clients, one user might set it up so that the client's name is one field and is entered "Smokey Bear." Another might separate the first name from the last to make sorting on the last name easier. And yet a third might still keep the two names in one field, only specify that the name is entered as "Bear, Smokey." This is what gives database programs their power: We are able to tailor the information and its structure to suit our purposes.

However, this flexibility requires that we do some more work when we transport information from one database program to another. Unlike a word processor, where a program can generally receive an ASCII text file from another without having to be aware of the structure imposed on it by the creator of the text, a database file consists of information that is highly ordered. We must, therefore, do some preliminary work in each program—the source program as well as the destination—before we can actually transfer the data.

Document the Source Database The first step in file transfer is to document the structure of the file in the source database. Ideally, this documentation begins the moment we create any new database. The documentation should include a listing of each field in the database, along with all the information necessary to duplicate that field definition in another database. Figure 4.8 contains an example of a form that might be used to document databases, showing the minimal information that should be kept for each file. Such documentation is important to the individual user or one in a small organization. It is mandatory to those in larger organizations, where many people might be using a database or be called upon either to modify it at a later date or to transfer its data to another program. While the form discussed here is designed particularly to be used in cases where data needs to be transferred, it is also very useful in other situations.

File Name:
Date Created Creator:
Date Modified: Modifier:

Program used:
Does this file cross reference any other separate files?
Which ones?
Describe the links:

Field Name	Field Type(1)	Field Format(2)	Length	Calculations
________	________	________	______	________
________	________	________	______	________
________	________	________	______	________

(1) Field types include: Text, Number, Date, Time, Graphic (generally for Macintosh programs only).
(2) The field format includes:
Text fields: alignment of the text within the field (left, centered, right)
Number fields: formatting for dollars (usually preceded by the $), percents, and the decimal accuracy of the number (i.e., if it is an integer or if not how many decimal places are figured).

Figure 4.8 A form for documenting the record structure of a database.

Each field in the file should be listed and fully described on the form. A rundown of information that should be kept for each follows.

Field type This includes the most basic information. It should tell whether the data contained in that field is text, a number, a date or time, or a graphic (for Macintosh programs only). If the field type is different from one of these general types, that should be noted and detailed. Some programs have data types that are not found in others.

You must note that text fields can, and often do, contain numbers. A phone number is an important example, as is a social security number. We enter these digits into text fields because we do not want the program to evaluate them as numbers. If we entered a phone number into a numeric field, for example, it might perform a calculation on it: 555-1212 would become −657 as the program subtracted 1212 from 555. A leading 0 in a zip could would normally be dropped, too.

Note Some programs, such as HyperCard, do not require that you assign any particular type to a field name.

Field format In the case of text fields, most programs use the format of the field to indicate whether that data should be left- or right-aligned on the screen form. For date and time fields, the format will usually indicate the manner in which the date and time is displayed (i.e., 2/15/88 or 2-15-88 or February 15, 1988). Many programs can work with all these formats and more. Some also store the date internally as a number and figure the date based on that number. Since there are many ways of numbering days (depending on what you call the "first day"— the Macintosh uses January 4, 1904), you should make sure the formatted date is what is transferred, if possible, and not the number. Some programs, such as dBASE III on the PC, cannot import a date from a text file. You must bring the date in as text and write a dBASE procedure that translates the text date into something dBASE can understand.

Field length Some programs require that you specify the maximum number of characters a field can contain. This applies specifically to text fields. Some programs (e.g., Double Helix on the Macintosh) allow you to have a virtually unlimited field length, and some put severe restrictions on the length of a text field. In either case, this should be noted in your documentation. If you have a field that is designed to hold large amounts of text, that should be specified. It may not be transferable to another program. If the source program does not require you to state the field length, you should, before transferring the data, attempt to find the length of the longest item that is actually contained in that field, to make sure the destination program can hold it. Most programs, when reading from text files, will simply truncate the text to fit within their specifications. You might lose some information.

Calculations The value of a calculated field is based on the value or values entered into other fields in the record (or it can be on values entered into other files, in the case of relational programs). Usually you do not enter data manually into a calculated field. If a field is calculated, the formula used to derive its value should be shown on the form. Calculated fields will not always be exported. If you are taking the information into another database program, you will probably want the destination database to have the capability to perform the same calculations itself. In that case, being clear about the formulas in the database documentation is important. Calculations must be documented in a clear manner that is independent of the syntax of the database program. In that way, the formula can be understood by others, who may not be familiar with the syntax of the source database.

If you are transferring data into another program for purposes other than data entry or do not wish to duplicate the functions of the original database, you must consider other things. If the data is going to a program, such as a word processor or desktop publishing program, you will want to export the calculated fields that have a bearing on the information you are trying to illustrate. If you are transferring the data to a spreadsheet or graphics program, you might want the spreadsheet to perform its own calculations, or you might just want it to contain those numbers to generate a chart.

Spreadsheets

Figure 4.9 shows a sample screen from an Excel spreadsheet on the Macintosh, and figure 4.10 shows the same screen as it appears in 1-2-3 on the PC. As you can see, a spreadsheet consists of an ordered group of data—usually numbers. This data is organized into columns, which are lettered, and rows, which are numbered. Each combination of a row and a column is called a "cell," and each cell contains one item of information—that information might be a label (text, usually describing other entries), a number, or a formula.

A formula instructs the spreadsheet program to make calculations on the values that are included elsewhere in the spreadsheet and to put the resulting value in the cell that contains the formula. If you change one of the numbers that is referred to by a formula, then the value calculated by that formula changes immediately. That's simple enough, and this simplicity of premise is one of the things that has made spreadsheets one of the most powerful and popular applications programs for personal computers.

In general, transferring spreadsheet data from one spreadsheet program on the Mac to another on the PC is very easy. This is due to the market dominance of 1-2-3 on the PC and Excel on the Mac.

On the PC, Lotus 1-2-3 is the dominant spreadsheet program and has been for several years. Recent studies show that it has a market share in excess of 60%. For this reason, manufacturers of other

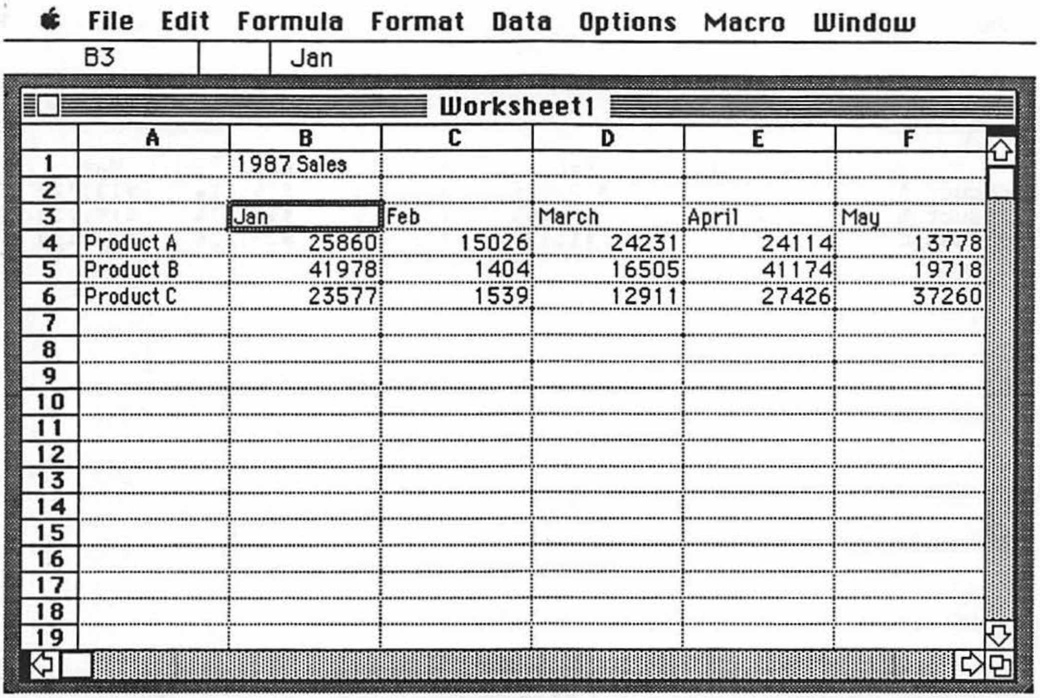


Figure 4.9 A representative screen from Excel on the Macintosh.

spreadsheet products for that machine (and for the Macintosh as well) have made their programs able to import data directly from the 1-2-3 format, known as the WKS format for the standard file extension appended to 1-2-3 data files.

On the Macintosh, the dominant spreadsheet program is Microsoft's Excel. Like most modern spreadsheet programs, Excel is able to read 1-2-3 files directly. So, transferring a 1-2-3 file to the Macintosh is as simple as getting the file to a disk the Mac can read (either a local disk or one available to the machine through a network) and then loading the file into Excel. To transfer a file the other way, you must tell Excel to save the file in the 1-2-3 WKS format; Excel will perform the necessary translation as it saves the file. To load it into 1-2-3 then is easy. The file must have a name that includes the .WKS extension and must be on a disk (either local or networked) that the computer running 1-2-3 can read.

Recently, new spreadsheet programs for both computers have been announced, including versions of Excel for the PC and 1-2-3 for the Macintosh. Whether any of these programs will actually chal-

```
B3: ^Jan                                                                READY

          A             B          C          D          E          F
1                   1987 Sales
2
3                      Jan        Feb        March      April       May
4   Product A        $25,860    $15,026    $24,231    $24,114    $13,778
5   Product B        $41,978     $1,404    $16,505    $41,174    $19,718
6   Product C        $23,577     $1,539    $12,911    $27,426    $37,260
7
8   Totals
9
10
11
12
13
14
15
16
17
18
19
20

09-Oct-87   10:37 AM
```

Figure 4.10 The same spreadsheet shown in figure 4.9, as it appears in 1-2-3 on the PC.

lenge the standards remains to be seen. What is clear is that file compatibility will be retained: All the new products I have seen support the Lotus WKS standard format. Many times, standards work for us.

Some General Pivot Formats

As in transferring most data between programs on the different machines, the pivot or intermediate format is the key. Very few database programs can read directly the formats of other database programs. A pivot file, of course, is one that is not the native format of either the source or destination program but is one that both programs can read and write. All the pivot formats discussed next are ASCII text files, containing few control characters to format the data. They are usually very structured files, to delimit the fields and records. And, being text files, they do not contain any graphics information.

One nice side effect of these files being ASCII text files is that you can view and edit them using most standard editors or word processors. When something mysterious is happening so that the information is not being transferred correctly, you can easily load the file into an editor and explore the structure of the file. The major caution is to make sure that, when you are finished editing the file with the word processor, you save it back to disk as an ASCII text file. Formatted information (beyond the field and record delimiters) has no place in a database file and will usually make it impossible for the destination program to read it.

Spreadsheets are also handy for examining the contents of a pivot file. Most spreadsheets, such as Excel on the Macintosh, can read delimited files (as well as SYLK and DIF). In a spreadsheet view of a database, each record is one line on the worksheet, with each field representing one cell in the line. Spreadsheets are good places to make global changes in the data (e.g., changing the format of date fields or other numeric fields) to make it easier for the receiving program. Remember, though, that spreadsheets can only work on files small enough to be held entirely in RAM, so it is possible that your file will be too large for the spreadsheet. Also, if you have large text fields, the spreadsheet may truncate them to fit into cells. Most spreadsheets can hold only up to 256 characters per cell.

You do not really need to understand the details of these file formats to be able to use them. Most database products can deal with at least one of these formats, and this information is presented in part to help you decide which format to use when you have a choice.

Comma-Delimited Format

In a sense, all the pivot formats discussed herein are delimited formats, since "delimited" means to establish the limit or boundaries. It simply means that a special character, or characters, is used to separate the contents of various fields from one another, and another is used to indicate the end of a record. The difference is that some of the more specific formats can sometimes include information about calculations, field names, and more.

Our example filing application in figure 4.7 is a typical screen of a database program for the Macintosh. The principles are the same for PC products, only the screen appearance is different. This entry screen is designed for a name and address database. As you can see, there are eight fields in this record: First Name, Last Name, Company, Address, City, State, Zip, and Phone. When scrolling through the records in the file or when entering new records, this is the screen you see. The database program probably uses a specific native format for holding the data, one that probably cannot be read by other programs.

> ***Note*** This screen has specific character formatting: The data you type in is shown in boldface. Even if another program can support the various type styles and formatting options, we still do not have a way to transfer that information.

When the information is written to a file on disk, you need a way to separate the contents of the different fields and to separate the records from one another. Both the source program and the destination program must agree on what is used to separate the fields, so the data does not get scrambled in the process. The characters you use to separate the fields and records are called the delimiters.

The most common delimiter used to separate fields on the PC is the comma (,—ASCII character 44). The most common delimiter of records is the carriage return. Using the comma-delimited format, the record on our sample screen is saved on disk; it will look like this:

```
Smokey,Bear,U.S. Forest Service,123 4th Street,Anytown,US,12345,(123)555-1212↵
```

When the source program, writes the file to disk, it first writes the contents of the first name field, then a comma, then the last name field, and so on. A carriage return (↵—ASCII character 13) marks the end of the record.

> ***Note*** Unlike word-processing files created on DOS machines, the linefeed character (ASCII 10) may not be included in the file. Most programs will strip out the linefeed when reading the file in, anyway.

The destination program reads from the file until it encounters a comma, puts that data into the first field (which need not be called "First Name"), reads until it encounters the next comma, and so forth until it finds the return, at which time it moves to the next record and does the same until it reaches the end of the file.

Using comma-delimited files to transfer data is fairly easy. Most database programs support this format. The problem with comma-delimited fields is that it is possible, even likely, that some of the fields may contain commas as part of the data.

For example, in our name and address file, we might have set it up so that First Name and Last Name are combined into one field, and it is specified that the last name be entered first, followed by a comma, and then the first name. The file as written in comma-delimited format, would look like this:

```
Bear,Smokey,U.S. Forest Service,123 4th Street,Anytown,US,12345,(123)555-1212↵
```

It's almost exactly the same, except that "Smokey" and "Bear" are in different positions. However, if the destination file was created to accept the first and last name as part of the same field, the data would get scrambled. The destination program would read "Bear," encounter the comma, and insert that into the first field (which might be called "Name"). Next it would read "Smokey," encounter the comma, and put it into the next field, "Company." Clearly, the file would be scrambled beyond use.

Programs that work with comma-delimited fields usually solve this problem by enclosing the fields that contain commas as part of their data in double quotes ("—ASCII character 34). The source program writes the file to disk so that it appears like this:

```
"Bear,Smokey",U.S. Forest Service,123 4th Street,Anytown,US,12345,(123)555-1212↵
```

Upon encountering the first double quote, the destination program realizes that the next comma is part of the data, not a delimiter. When it reaches the second double quote, followed by a comma, it knows that it has reached the end of the field. The destination program should remove the double quotes from the file as it is putting the data into the fields (some programs, though, such as Double Helix, do not do this—the double quotes appear as part of the field data). This format is often called a merge format, since it is used by Microsoft Word and many others for their mail merge format.

When creating a file of this type, some programs enclose all fields in quotes, whether or not they contain commas. FileMaker Plus on the Macintosh, for example, does this.

Merge Files As already mentioned, the merge format is a variation of the comma-delimited file. This format is generally used to merge records from a database—most often names and addresses—into a letter created on a word processor. In this manner, of course, the same letter can be sent to a number of people, and each letter is personalized so that it has the name of the specific individual somewhere in the letter.

A file structured in the style of the merge format usually differs from the comma-delimited format in only one respect. The first record in the merge file contains the names of each of the fields that appears in the following records.

The first record of our name and address sample, along with the field names, looks like this on disk:

```
First Name,Last Name,Company,Address,City,State,Zip,Phone Number↵
Smokey,Bear,U.S. Forest Service,123 4th Street,Anytown,US,12345,(123)555-1212↵
```

This makes accessing the file and referring to it from within your word processor easier. You can let the word processor worry about the names of the fields. It looks at the first record in the file to get the filed names, then replaces those field names in the letter with the information represented by that name in the list file.

For example, the opening of a letter prepared in WordStar and used with MailMerge to merge the letter with your name and address list would look like this:

```
&First Name& &Last Name&
&Company&
&Address&
&City& &State& &Zip&
Dear Mr. &Last Name&:
Remember, &First Name&, only you can prevent forest fires . . .
```

When merging with WordStar, the ampersand (&—ASCII character 38) encloses the variable name. When WordStar's MailMerge program processes this file, it encounters those variable names and knows to substitute that name with a specific field from the merge file. By having the names of the fields at the top of the merge file, it knows what each file is called.

When printed, this letter would read this way:

```
Smokey Bear
U.S. Forest Service
123 4th Street
```

Anytown, US 12345

Dear Mr. Bear:

Remember, Smokey, only you can prevent forest fires . . .

There are a couple of things to note about this example. First, the comma placed between the city and state names in the letter was outside of the & field names. That prevents MailMerge from reading the comma as part of the field name. All spacing between the merge fields is handled from within WordStar. When merging, most programs strip or remove all spaces from the beginning and ends of fields (most database programs do this, too). Second, the assumption in this example that everyone in the list was male (the line that reads "Dear Mr. &Last Name&:"). In real life, this will probably not be the case, and you should probably include in name-and-address application another field that would include the honorific you use to refer to the person (e.g., "Mr." or "Ms.").

Although this example has been keyed to WordStar's method of merging files, the basic techniques are similar for other word-processing systems. Microsoft Word, for example, uses a similar mechanism on both the PC and the Macintosh. MultiMate will read the data from a file formatted in this manner, but you will need to define the names of the fields in your MultiMate document and delete the first record of the merge file. Otherwise, MultiMate will produce one letter that merges the field names into the letter.

If your database program does not support the merge format directly (i.e., if it does not place the field names at the top of the file), you can use your word processor, or even a spreadsheet, to do this. Load the file containing the names, add one record at the top that contains the field names, and save it as an ASCII document on disk.

WordPerfect on both the PC and the Macintosh uses a very different means of merging files, with a different merge format. The files WordPerfect uses to contain the names are called "Secondary Merge Files," and they use specific control characters to delimit the fields. Most database programs do not directly support the WordPerfect Secondary Merge File format. However, the wpconvert utility that is included with WordPerfect can translate files from DIF or delimited formats to the Secondary Merge File format. In this case, be sure to remove the first record containing the field names.

If your database program does support the merge format, you can use this to reconstruct the field names in the destination from what they are in the source. If you are unsure of the field names, you can write the file to disk in the merge format, open that file with your word processor, and get the names of the fields from the first line of the file. Since the source program is writing the fields to the file in a different order than they appeared in the screen form, you can also ascertain in which order they will be read in the destination program. This helps assure the successful transfer of your data.

A few programs, such as dBASE Mac, can construct the structure of a file based on the information that is held in the first record. This is very handy, since it means you do not have to worry so much about the order that fields appear in the source file, nor do you need to worry about the names of the fields. dBASE Mac will create the file for you and assign names to those fields.

Tab-Delimited Format

The comma-delimited format, or a variation of it, is more common on the PC than on the Macintosh, although most Mac programs support it. Most Macintosh programs delimit their fields using the tab character (↳—ASCII character 9). Some PC programs can use the tab as a delimiter as well. Since virtually every database program prohibits you from entering a tab character into a field as data (usually, the program to move to a different field when you hit the tab key), the conflict of having a delimiter character also appear as part of the data does not arise.

Our file delimited by tabs will look like this on disk:

```
Smokey↳Bear↳U.S. Forest Service↳123 4th Street↳Anytown↳US↳12345
    ↳(123)555-1212↵
```

The tab-delimited format is easier to work with then the comma-delimited format. It can be used to transfer data between most Macintosh programs and between many PC programs and Mac programs. Other programs, such as Excel on the Macintosh, can also read tab-delimited files.

Delimited Formatted Files and Spreadsheets

When a spreadsheet program reads a delimited file, it works about the same way as a database program. It reads until it finds the delimiter, puts the value into a specific cell (instead of a field), reads until it finds the delimiter, and so on. When it encounters a return, it creates a new row on the spreadsheet instead of creating a new record.

When a spreadsheet writes to a delimited file, the process is reversed. Starting at the top right-hand corner of the worksheet, it firsts writes the contents of that cell to disk, followed by the delimiter. In this manner, it writes all the fields in the first row, then ends that row with a return. It then goes to the second row and does the same, and so on throughout the spreadsheet.

Transferring data from a database to a spreadsheet is much easier (and is done more often) than going the other way around. There is not much reason to write the entire contents of most spreadsheets to databases, since the information won't always apply to fields and records. Just because a particular piece of information is on the same line as another does not mean it is related in the sense that one field in a record is related to another field in the same record in a database. So it won't always be necessary to transfer the entire contents of the spreadsheet. However, as we've said, many spreadsheets contain database functions as part of them. This database information will often need to be transferred. Some of the conversion programs, such as Lotus' TRANS.EXE program, allow you to select a particular range in a spreadsheet—usually containing the database information—and to transfer just that information.

Some spreadsheets, such as 1-2-3, cannot read or write delimited files directly. Instead, they use conversion programs, which come as part of the spreadsheet program, to translate from or to their native formats from or to the delimited formats. It is possible, with 1-2-3, to transfer information to a text only (no formulas) file from within 1-2-3, with its Print to disk function. When you use 1-2-3's /p command to print, the next option is "File." When you select this option, 1-2-3 prints the data to a text file on disk, not to the printer. Unfortunately, 1-2-3 does not delimit the cells when it performs this function. Instead of inserting tabs between the columns, it inserts enough necessary spaces to make the columns line up. This means that most other spreadsheets and/or databases will not load that information into separate cells when they read the file. Instead, all the information on one line will appear to be one record, or one entry in a cell.

This is unfortunate; 1-2-3 does not have a built-in way to create a delimited file. The 1-2-3 Convert program (TRANS.EXE) can handle dBASE II, dBASE III, and DIF formats, but it cannot handle delimited formats. The 1-2-3 macro command language, however, allows you to open a file and write to it directly. In this manner, you can create a macro that opens the file, writes the contents of a cell to the file, writes a tab to the file, and goes onto the next cell. Other solutions could include writing a small basic line to remove the spaces and replace them with tabs, or to do it from a word processor. There are other convert programs available, including the TOPS translators and MacLink, that can translate from 1-2-3 or dBASE formats into delimited files.

1-2-3 can read delimited files with its file import command. When you type "/fi" to call up this command, 1-2-3 asks if you want to import the file as text or as numbers. Importing the file as text causes all lines to be read as labels; they are placed into one cell, regardless of delimiters. Reading the file as numbers causes 1-2-3 to parse the file and separate the delimited fields into different cells.

On the Macintosh (and, presumably, with the PC version), Excel can directly read and write delimited files. When reading a delimited file, it automatically parses the file so that different fields, separated by tabs, are placed into different cells. When you save an Excel file as text only, the program automatically places tabs between cells and returns at the end of the row, when it writes the file to disk.

SYLK Format

SYLK is a format originally developed by Microsoft to facilitate data transfer between their MultiPlan spreadsheet on the Macintosh and other computers. SYLK, which stands for Symbolic Link, is an ASCII representation of the data in a spreadsheet or database, structured in such a way that a program that reads this format can reconstruct the spreadsheet entirely, including the formulas that make up a calculation. However, most databases only use SYLK to transfer data; they do not use it to transfer formulas.

A complete discussion of the SYLK format is beyond the scope of this book and is not necessary for using the format. Unlike the comma- and tab-delimited and merge formats, you will probably never need to modify a SYLK file directly from within a word processor or other text editor.

What is important about SYLK is that it is a format that can facilitate transfer of specific data between products that include some awareness of what the data contains. While all the data in a SYLK file is represented as text, of course, the SYLK format does support such things as telling the destination program that a particular field is a date or a number, for example. And, when transferring data between spreadsheets, it does include information about the calculations that were used to create the data.

SYLK's power derives from its ability to create a header for the file. The header includes data that describes the file, including field names, and tells something of the structure of the file. This structure information includes how many fields and records there are in the file.

Since it is a Microsoft product, Excel can read and manipulate SYLK files very well. 1-2-3, on the other hand, cannot deal with them at all. You will need to use an intermediate conversion program to translate files from SYLK into another format—either delimited, dBASE format, or DIF to allow 1-2-3 to read those files.

Data Interchange Format

Data Interchange Format (DIF) was developed by Software Arts, the manufacturers of VisiCalc. Like SYLK, DIF was developed by a specific manufacturer to facilitate transfer of data between their own products, but it has come to be widely used. Since VisiCalc was one of the first truly successful microcomputer applications, this format is supported by virtually every database and spreadsheet program, as well as all translation utilities.

Like SYLK, DIF includes header information that programs use to indicate how many fields and records are in the file. In DIF terminology, these are called "Vectors" and "Tuples." Vectors are fields, and Tuples are records.

Ideally, with DIF or SYLK, programs should be able to develop the structure of the database based on the information that is contained in the file. In practice, most software cannot do this.

A De Facto Pivot Format: dBASE III Plus

dBASE is such a common program, and such an important one, that an entire chapter, if not an entire book, could be written about file transfer to and from this program. dBASE was one of the first truly powerful database products available for microcomputers. It dates back to before the era of the IBM PC, to the time of the popularity of the 8-bit CP/M operating system (otherwise known as the dark ages). Because there are so many copies of dBASE in existence, and, consequently, there is so much data in that format, many programs can read and write those files directly. Virtually all the standard translation programs support the dBASE file format, so it is a very useful pivot format.

> ***Note*** There are actually two versions of the dBASE format. dBASE II, which was a product for the CP/M operating system and an early product for the PC line, had one format. When Ashton-Tate made the transition to dBASE III, a more powerful version of the program designed for the 16-bit PC environment, the older file structure would not support the new features they deemed important. If you have older files, in the dBASE II format, they should be translated to the newer format first, although some conversion programs (e.g., Lotus' TRANS.EXE utility) will support both formats.

Exporting from dBASE dBASE uses two commands to export data, the copy and export commands. Copy merely duplicates all or part of an active database file to a new file. Use this if you want to transfer only a portion of a dBASE file to another product. First, select the records you want to copy, then use this command to write them to a new file. You can then use either the export command to process that new file or work with that file with another program.

The export command allows you to save dBASE files to the file format of Software Publishing Corp's PFS format. If you have a screen form associated with the current file, then that format will be used to create the new file. If your destination database does not support the dBASE format directly, you can use the export command to save the data to a PFS formatted file. Then use either one of the PFS programs to export the file to a delimited format or use a translation program that supports this format.

Most data transfers from dBASE formats will work best if you use one of the file translation utilities. These utilities can read dBASE files and convert them to delimited files, which your destination program can probably read.

In the spreadsheet world, 1-2-3 can read and write dBASE II or III files with its TRANS.EXE conversion program. Writing a dBASE file with this program requires that the portion of the spreadsheet you transfer be a defined 1-2-3 database, including headers at the top of the column to define field names. In this manner, it can totally create the database structure in dBASE format. Some other programs, such as VP Planner from Paperback Software, can deal with dBASE III files directly, from within the spreadsheet program.

Another Common Pivot: Lotus 1-2-3 Format

Just as dBASE is the dominant and most influential database program on personal computers, so 1-2-3 is the dominant spreadsheet. Virtually all spreadsheet programs can read and write files in this format. Most can do so directly; some must use separate conversion utilities.

Transferring Spreadsheet Files Between 1-2-3 and Excel

Excel on the Macintosh is able to read directly files created by 1-2-3 on the PC. These files must be of the type WKS used by 1-2-3 release 2.

Even though Excel can directly read 1-2-3 files, not all the original spreadsheet can be converted into Excel format. Certain things, such as 1-2-3 macros, cannot be converted at all: Excel's command structure is too different from that of 1-2-3 to allow for automatic translation of the macros. Some formulas, range names, and other items may not be translated correctly. We'll talk about some of those items now.

To convert a 1-2-3 spreadsheet to Excel format is simple. Either copy the 1-2-3 file to a Macintosh disk or have it on a volume that is available to the Macintosh. When you open the file with Excel, that program will automatically recognize that it is a 1-2-3 file and translate the formulas as it opens the file.

To convert an Excel file to 1-2-3, choose "Save As . . . " from Excel's File menu, click on the button labeled "WKS," type in a file name, and save the file. You may save the file directly to a volume that is available to a PC (either on the AppleShare server, if you have privileges for that volume, or on a PC disk if you are using TOPS), or to your own disk for copying later. Excel will translate the file into 1-2-3 format as it saves the file.

Note When naming the worksheet, you must give it a legal DOS name. For 1-2-3 to recognize that it is a worksheet, it must have the three character extension .WKS.

As it is opening or saving a file in WKS format, Excel may encounter something that it cannot convert. If it does so, it will display a dialogue box that asks if you want to see all the errors it encounters as it encounters them ("Show All") or whether you just want to see the total number of errors at the end of the translation process ("Show Total"). Generally, you should choose "Show Total," as that will speed the conversion process so that you won't be presented with a number of dialogues during the process. On the other hand, the Show Total option does not give you much information about the nature of the errors that occurred. If you want to make notes about those errors, choose "Show All" and write down the errors as they occur.

If Excel cannot convert a formula during either of the conversions, the text of the formula will be maintained. That text will be entered into the specific cell as a label or text entry, and no calculation will take place. This may cause errors elsewhere on the worksheet. It does, though, allow you to analyze the formula in question and determine what steps you may take to translate it to the foreign syntax.

Carefully planning spreadsheet style and structure can help you minimize the errors that occur. If you create worksheets with the thought in mind that they are to be used on alien software products, you can probably eliminate the errors all together. The best way to do this is to use a subset of the formulas available in either product: only those formulas that work on both machines. Doing this may make programming in one or the other of the spreadsheets more difficult, but it can save work and translation confusion if you are making worksheets that are to be used on both machines. Alternatively, you can segment a worksheet so that there is some duplication in it. One portion of the worksheet may have formulas that are specific to Excel, and another may have formulas that are specific to 1-2-3.

In general, Excel has more features and more formulas than 1-2-3. Excel can read more 1-2-3 formulas than 1-2-3 can read Excel formulas. If a worksheet that is to be used on both machines is constructed first in 1-2-3 and then translated to Excel, you will need to translate less.

As already mentioned, at the time of this writing, Microsoft was in the process of preparing a new version of Excel for the PC. According to reports in the press, this new version features enhanced compatibility with 1-2-3, including the ability to read not just 1-2-3 formulas but the macros as well. It is difficult to tell what impact this will have on the Macintosh version of Excel. For example, if the new PC version can read 1-2-3 macros, will it then be able to also translate those into its own macro syntax? If so, will it then be able to save those macros to the Macintosh Excel file format? If so, then we've come into a new era of file compatibility, and we should all upgrade as soon as possible to the new versions.

Multiuser Software

One of the goals we often have when setting up a network is to allow several individuals to work on the same file at the same time. Products that support this are called "multiuser." For example, in a database application, we might want one individual at the order desk, entering orders into our accounting system. Another person might be inputting new goods received. Both these operations should automatically adjust the amount of inventory on hand, and they should do it virtually instantaneously.

A multiuser database must support record locking. In a multiuser situation, two users could be working on the same record simultaneously. If both make changes to that record, the software must take both those changes into account. Generally, multiuser databases lock out multiple users of the same record. In that manner, only one person may change a record at a time.

At the time of this writing, several products are available for the Macintosh and the PC that support this multiuser capability on a network. On the Macintosh side, database programs such as Odesta's Helix, Blythe's Omnis III Plus, and Acius' 4th Dimension offer this capability. On the PC side, there are many programs, such as Ansa's Paradox, dBASE III Plus, and rBASE that can allow users to share files on a network. Unfortunately, at the time of this writing, no major products are available that support the simultaneous use files by users on both machines. In Fall 1987, Blythe announced Quartz, a product for the PC that runs under Windows. Similar, but more powerful than their Macintosh product, Omnis III Plus, it does not use the same file format as the Macintosh product. An upgrade for the Macintosh product due in 1988 will allow it to use the same file format, but it is unclear when multiuser capabilities between the two systems will be supported. The same is true of Ashton-Tate's dBASE

III Plus and dBASE Mac. dBASE III Plus supports multiple users on a network, but dBASE Mac does not. dBASE Mac can read dBASE III files but cannot really work with them without some translation. Eventually, it is to be hoped, there will be more convergence between the database products on both machines.

Multiuser spreadsheets are a different manner. When a spreadsheet is called up, typically the entire file is loaded into memory, and multiple individuals can then work on the spreadsheet at the same time. The problem arises when both users save their spreadsheets. The last one saved is the one that appears on disk, and when user B saves a spreadsheet, he might destroy the changes made to it by user A who just saved it.

Unfortunately, few spreadsheets are available today that support true multiuser work in spreadsheets. Instead, that will require that those who are using spreadsheets on a network decide on such things as file use and naming conventions (append a number to the file name to make the version number clear). In TOPS, directories that contain spreadsheets should be published as "One writer" from the Macintosh, so that only one person will be able to change those files at a time. Others will be able to manipulate the spreadsheet but won't be able to save their changed versions of the file.

Part E *GRAPHICS AND DESKTOP PUBLISHING*

Besides the standard functions of word processing, filing (databases), and number crunching (spreadsheets), people do many other things with PCs and Macintoshes. Here we'll talk about some of the other kinds of data that you can generate on your computers and how to transfer it.

Graphics

The Problem

As is the case with text files, there is no standard way for programs to store graphics information. The situation is somewhat better on the Macintosh than on the PC, but you still must consider much in dealing with graphics applications and their data.

First, it is important to understand that there are different types of graphics data. They usually correspond to the type of program that created them and to what use is made of the data. Roughly speaking, there are four types of graphics information: bitmaps, object-oriented or vector graphics, scanned images, and Encapsulated PostScript.

Bitmap Graphics

A bitmap graphic file is typically created by a paint-type program. Bitmapped programs treat the entire screen or drawing area as a collection of dots and store the file as an array of 0s and 1s—0s for the white space, and 1s for the black marks. Typically, all the dots are turned off (i.e., they are white or have the value of 0) when you start drawing. When you use a drawing tool on the screen, you turn on the dots or make them black. Usually, programs that create bitmaps offer a range of artistic tools, such as a paintbrush, that can assume different shapes and tools for drawing lines, circles, and boxes. Often you can type text into these programs. Since the program stores the file as a bitmap, you usually cannot go back and edit the picture without doing a lot of erasing. For example, if you wish to make a circle smaller in a paint-type program, you must first erase the original circle (being careful to not erase other parts of the picture) then draw the circle again.

On the Macintosh, programs that create bitmaps include MacPaint, FullPaint, and one of the layers of SuperPaint. Programs that access bitmaps include the PictureBase, Art Grabber, and Artisto desk accessories, and the desktop publishing programs. Furthermore, the Macintosh clipboard can access bitmaps, so that these kinds of images can be placed into many other types of files, such as word processors. Since one of the first two programs available for the Macintosh was MacPaint, the bitmap format that MacPaint uses has become a standard file format. Most programs that create bitmaps can read and/or save files to this format. This simplifies the tasks of editing, using, and transferring the files.

Typically, bitmap files on the Macintosh will have the file type PNTG.

In the PC world, there is more chaos regarding file formats.

Since the PC has always been more of a text-oriented computer and since no particular graphics program has really reached prominence as a standard, there is less agreement as to how to store a bitmapped file. And, of course, few programs on the PC other than specific graphics applications or desktop publishing programs can include graphics information. DOS-based programs that can create bitmaps include almost all programs with the word "paint" in their name: PC Paintbrush, Publisher's Paintbrush, Windows Paint, Gem Paint, and more. Fortunately, many of the paint programs for the PC are derived from ZSoft's PC Paintbrush and, therefore, can deal with many of the formats.

Color images, more common on the PC than on the Macintosh at the time of writing, add new levels of complexity to the transfer problem. Instead of storing just one bit (i.e., 0 or 1) for each dot on the screen, color images must store more bits. You can describe a black and white drawing with just one bit per screen dot, but color, obviously, requires more bits. Since color is recent to the Macintosh, there are still no standards for its representation in a file. For images that contain different colors, or different shades of grey, the TIFF format is used.

Transferring Bitmaps from the Macintosh to the PC

As with the other types of files discussed in this book, you can transfer bitmaps between various programs on the Mac and the PC several ways. Several programs—notably the desktop publishing programs PageMaker and Ventura Publisher—can directly open MacPaint files. With these programs, using a MacPaint file involves little more than making the file accessible to the program (whether accessing it directly from a DOS disk or over a mounted network volume) and making sure that the name of the file is consistent with what the PC program expects. As with other types of files, too, once you have transferred the file into one program, it is often possible to use utilities that are part of that program to convert the file to other formats.

PageMaker on the PC PageMaker expects MacPaint files to have a legal DOS file name that includes the extension .PNT. PageMaker allows you to place MacPaint files directly into the document and use all the standard PageMaker tools (cropping and resizing) to manipulate those files.

Once you have placed a graphic image into PageMaker, you can put that image onto the Windows Clipboard and so paste it into any other Windows program that accepts bitmaps from the Clipboard (e.g., Windows Paint).

Ventura Publisher Like PageMaker, Ventura Publisher version 1.1 (VP) can read MacPaint type files directly. With Ventura, you use the Load Text/Image command from the file menu. Click on "Image" then on "MacPaint," to let Ventura Publisher know what type of file you are loading. When placing documents into Ventura Publisher, you should not do so over a networked disk. Since VP does not actually make a copy of the image and include it as part of the chapter but rather remembers where it got the file, it will want to reload that image from the same location when you next open the chapter. Since it is not likely that you will always have the same set of volumes mounted in the same manner every time you open Ventura, the program could become confused and not be able to access the file. You should first copy the file from a Macintosh disk to a subdirectory on your PC disk and place it on a local disk before incorporating it into Ventura Publisher.

One of the problems with using Ventura Publisher to import MacPaint files is that it always loads the entire page. A MacPaint file always has a maximum size of an 8-1/2 × 11 in. sheet of paper. Ventura Publisher will always assume your MacPaint file fills this entire sheet and will so scale the entire page to fit it into the frame you have created in the VP chapter. This makes it hard to work with these images. You need to tell Ventura not to scale the image (i.e., to keep it the same size as the original) and then use Ventura Publisher's cropping ability to resize the frame to hold the part of the graphic you want to display.

The Graphics Link This is a program from PC Quik-Art that is designed to convert bitmap files from one format to another. Since one of the formats it supports is MacPaint, it is fairly easy to use this program to move bitmaps from one machine to the other. If you are converting a file from the Macintosh to a PC format, the operation is straightforward. The only requirement that The Graphics Link makes is that you provide an extension for your MacPaint files. I recommend that you use a common extension (not a bad idea, anyway, in the Mac world, even if the operating system does not use them). You might use .PNT for MacPaint files, since that is also what PageMaker expects.

Figure 4.11 shows the screen you use to tell The Graphics Link what types of files you want to convert. Converting from a MacPaint to, say, a PC Paintbrush format is an easy process.

```
(c) 1987 TerraVision Inc.      Version 1.50      Licensed to PC Quik Art Inc.
╔══════════ Source File Type ══════════╦══════════ Target File Type ══════════╗
║                                      ║                                      ║
║   PC Paintbrush+ .PCC or .PCX        ║                                      ║
║                                      ║                                      ║
║   Microsoft Windows Paint .MSP       ║                                      ║
║                                      ║                                      ║
║   GEM/Ventura Publisher .IMG         ║                                      ║
║                                      ║                                      ║
║   PageMaker TIFF .TIF                ║                                      ║
║                                      ║                                      ║
║   RIX EGA Paint 2005 .PCQ            ║                                      ║
║                                      ║                                      ║
║   Macintosh MacPaint                 ║                                      ║
║                                      ║                                      ║
║   Dr. Halo .CUT                      ║                                      ║
║                                      ║                                      ║
║   Standard BLOAD <.BLD only>         ║                                      ║
║                                      ║                                      ║
║   Mouse Systems PC Paint+ .PIC       ║                                      ║
║                                      ║                                      ║
╚══════════════════════════════════════╩══════════════════════════════════════╝
```

Figure 4.11 The main screen from The Graphics Link.

1. Either mount a Mac volume containing graphics on the PC or copy your MacPaint file to a PC disk. If necessary, rename the file so that is has a .PNT or other extension.
2. Start The Graphics Link. If you have copied the file you want converted into the same PC subdirectory that contains The Graphics Link and if you want to save your file to that same directory, start the program by simply logging into its directory and typing "TGL" at the command line.

   For example, to convert a file from a mounted Mac Volume (drive E:) to a subdirectory on a PC disk that contains picture files (C:\PICTURES), type the following at the DOS command line:

```
C:\>TGL E: C:\PICTURES
```

This only works if you are currently logged into the directory that contains The Graphics Link or if that directory is in your search path. The first parameter that "TGL" takes at the command line indicates the source for images you want to convert. The second shows where it should place them. Alternatively, you can change the subdirectories for the source and destination files from within The Graphics Link.

3 Select "Convert Files" from the first screen that appears, by using the arrow keys to point to the selection and pressing the enter key. This will take you to the conversion screen (shown in figure 4.11). The screen is divided into two halves, one showing the source files and one showing the target files. At the bottom of each window, TGL shows the volumes and/or directories TGL will use for files.

4. Using the arrow keys, move the highlight to "Macintosh MacPaint" in the left-hand side of the screen, under the heading "Source file." Press the enter key. Next move the pointer to the type of file you want to convert to in the target window. (The target window, as the source window, shows the normal extensions for each file type in the DOS world.) The Graphics Link automatically appends these extensions to converted files. Press the enter key when you have pointed to the file type you want.

5. The Graphics Link will now display a small dialogue box, asking for the extension for the MacPaint files. Type in "PNT" or whatever you use and press return.

6. The Graphics Link will now ask you to point to the files you want to convert, showing only the files in the source volume/directory that match the extension you've typed for a MacPaint file. One of the nice features about The Graphics Link is that it allows you to select a number of files and convert them in one operation. This means you can set it up, and it will operate unattended. This feature is great if you have many bitmaps to convert.

   To select the files you want to convert, point to them using the cursor keys and press T for tag at each file you want converted.

   One of the idiosyncrasies of The Graphics Link is that it inverts files when it converts them. That is, what is black on the Mac screen will appear as white when you show it on the PC screen, and vice versa. This is an outgrowth of the nature of the machines: The PC typically shows lighted characters against a black background, while the Mac is just the opposite. This is fine if you are converting files to use in another Paint program; they usually have invert tools of their own. However, if you are going to be using the picture in a different manner, you can tell The Graphics Link not to reverse them by pressing R at each file.

When you've tagged all the files you want converted, press the enter key.

7. The Graphics Link will now convert the files, one by one. For each file, it must first read the file into a buffer, analyze it, convert it, then write it to disk. A status box shows the file being processed, how many more files it has to process, what it is doing at the moment, and the percentage it has completed.

The process for converting files from the Mac to the PC is complete. The Graphics Link has added the correct extensions onto the files. If you have specified the correct directories at the command line, you need do no more than use the program of your choice to edit or use the bitmap.

If you are converting the file from the PC to the Mac, however, there is still one more step. If you have used The Graphics Link to write the file to a PC disk, you must first copy it over to a Macintosh disk. However, since The Graphics Link is a DOS-based program, it does not include the type and creator of the file as part of the file. To use the bitmap with any Mac program, you must provide this information. Briefly, you must tell the Mac that the file is a PNTG type, if you want to either open it with MacPaint or FullPaint or place it with PageMaker. You must also provide it with a creator, so it can correctly display an icon or be opened from the desktop.

The Graphics Link is a handy program. If you want to convert bitmaps to use with other paint programs, it is one of the few utilities available. Of course, if you use PageMaker PC or Ventura Publisher, it is not really necessary. These two programs both read MacPaint files and can be made to do some conversions themselves. However, if you have many files to convert, The Graphics Link works well in a batch mode.

Besides being able to convert files, The Graphics Link also allows you to scale files—make them larger or smaller. This is especially handy if you need to use a MacPaint file in Ventura Publisher. In fact, given the manner in which Ventura Publisher reads MacPaint files, this is probably the preferred method to use. First, use The Graphics Link to convert the file to the format of a DOS painting program, then load it into Ventura Publisher.

Object-Oriented Drawings

The second type of graphics file is the object-oriented drawing. Briefly, object-oriented programs treat what you draw as discrete objects. Recall that if you are using a bitmap, paint-type program and you want to change the size of a circle, you must first erase the circle, then draw a new one. However, a draw-type program remembers that you used a specific drawing tool (e.g., an oval) to create a particular object on the screen. To change the size of the circle, select the object, then use whatever tools the program gives you to resize it. You do not have to start over. Whereas bitmap programs store a file as an array of 0s and 1s, draw programs store information that tell them that you have placed, say, a circle of a certain size at a certain location on the page.

Macintosh programs that work in an object-oriented manner include MacDraw, MacDraft, Cricket Draw, and the draw layer of SuperPaint. Additionally, the Macintosh Clipboard (and Scrapbook) allows you to cut and paste objects. Files that contain objects are usually called "PICT" files. A MacDraw or Cricket Draw file can be saved as a PICT file by clicking in the appropriate box in the Save as . . . dialogue box. Both programs can open PICT files; however, such files lose a little of the program's normal internal formatting.

There are advantages to creating files with a PICT format. Object-oriented files maintain within themselves information about the objects that assist desktop publishing programs, word processors, and the like in importing the images. These images can be resized within other applications and maintain more information about what they should look like. Often, when using images within a desktop publishing program, you must resize the images to make them better fit within a page layout. With bitmap images, this can lead to some distortion, especially with tightly packed grey patterns. This is not true with object-oriented images. A circle, arc, box, line, or text retains the information about such things as fill patterns and line width that you put into it with the original program. So changing the size of the image usually does not distort it. Printing these files to a PostScript printer is also usually faster and preserves more of the appearance of the original image you created on the screen. Most programs that create business graphics (i.e., Lotus 1-2-3 in its PIC format and Excel Charts) are saved in an object-oriented manner. When resized, the patterns and labels that are part of these files maintain their integrity.

In the PC world, there are many programs that work in an object-oriented mode. As in the Mac world, most of these will have the word "draw" (as opposed to "paint") in their name (e.g., Windows Draw and Gem Draw Plus). Additionally, Computer Aided Design (CAD) programs are object- or vector-oriented, and the leading desktop publishing programs will be able to incorporate their files (though sometimes you must first save them to a special format).

Unfortunately, there are fewer standards for storing object information than there are for bitmaps; therefore, transferring these types of files between the two systems is very difficult. As of this writing (November 1987), there is no object version of a program like The Graphics Link for these types of files.

At this point, it is best to first convert object-oriented drawings to another format (e.g., a bitmap or an Encapsulated PostScript file) and then convert it to the other computer. This is not the best of solutions: Bitmap illustrations do not resize or print as well as object-oriented drawings, and the support in the PC world for Encapsulated PostScript is not as good as it should be, yet.

Encapsulated PostScript

Encapsulated PostScript (EPS) is the third standard for transferring and printing graphic images. It works very well for such programs as desktop publishing programs that will use laser printers featuring the PostScript page description language to print the documents.

PostScript is the page description language created by Adobe Systems for printers. It is a true programming language that is specifically designed to tell a printer how and where to place items on a page. It is used in the Apple LaserWriter printers and in printers from a wide variety of other manufacturers. In desktop publishing, it is the dominant means of communicating a page design from desktop publishing software to a printer.

Essentially, an EPS file includes the information about the drawing in the form of a series of programming instructions written in the PostScript programming language. Programs such as Adobe's

Illustrator and Cricket Draw on the Macintosh can create files in this format, and more new ones are coming. Most desktop publishing programs—including the two leading PC programs and all the Macintosh programs—support this standard.

There are two forms to the EPS format: one for the Macintosh and one for the PC. The Macintosh EPS format includes an object-oriented representation of the drawing as a header to the file. This allows a program such as PageMaker on the Macintosh to display the EPS data as it will appear on the page. When you resize the image, using PageMaker's cropping or sizing tools, the PostScript portion of the file is also changed to print the file on the page accurately. This makes the design of a desktop published document easier: You can see directly what is going to be printed.

Unfortunately, since there is no standard mechanism for representing object-oriented graphics on the PC, the Encapsulated PostScript format is not able to include that information in files created for the PC. Instead, when you place an EPS file into either Ventura Publisher or PageMaker, these programs must place a marker that shows you how large the image is on the page. This marker takes the form of a rectangle with a large X in it. You can resize this marker, but you still cannot actually see the PostScript material until you print it out.

For PageMaker PC to recognize that a file is of this type, it must have the extension .EPS. Ventura Publisher has an option in its Load text/file dialogue box that allows you to specify that the file you are loading is of this type. Besides the desktop publishing programs, other programs (e.g., Microsoft Word 3 on the Macintosh) now allow you to place EPS files.

Encapsulated PostScript is a very handy means for describing images on a page: It allows the printer to work at its greatest resolution, no matter which of the many PostScript printers you are using. The use and capabilities of this format will probably grow over the coming months as PostScript solidifies its hold on desktop publishing.

Tag Image File Format

Tag Image File Format (TIFF) is a special format developed by Adobe Systems, Aldus Corporation, and Microsoft Corporation for the representation of images that contain grey scales, usually images produced by a scanner, although newer graphics programs for both the PC and the Macintosh allow you to create these files directly.

The reason for the existence of TIFF is simple. Bitmaps are good for representing images with only one level of black. A dot on the screen or paper is either black or white. That requires only one bit for each dot. If you need to display more than one shade of black or more than one color, you need to store more bits for each dot on the screen. Unless there is some way to compress these bits, the size of a file increases dramatically with the number of shades or colors you want to include in the image.

TIFF, then, provides a common method for describing these complex images. Using TIFF, programs also have a means of showing the image on the screen. Most monitors still use only one bit per dot on their screens, although that is changing fast.

PageMaker PC recognizes a file as a TIFF file when it has the extension .TIF. Ventura Publisher has an option in its Load text/file dialogue for this type of file.

Desktop Publishing Programs

While we've been talking about desktop publishing programs and graphics throughout this chapter, here we'll turn to desktop publishing programs and word-processing files.

PageMaker

PageMaker is the desktop publishing program that virtually created that software genre on the Macintosh. Since its release in early 1985, which coincided with Apple's release of the LaserWriter Plus, it has been the dominant desktop publishing program, even though there are now others on the market that surpass it in capabilities—especially for dealing with longer documents. In early 1987, Aldus Corporation released PageMaker for the PC, adding to its stature in the industry.

The two versions of PageMaker can share files. A file created in PageMaker on the PC can be read directly into the Macintosh version. And a Macintosh PageMaker document can be read directly into the PC product, as long as it has the extension .PUB as part of its file name.

PageMaker on the Macintosh This program is able to place text files created by MacWrite, Microsoft Word, and Write Now, as well as ASCII text files created by other word processors. Furthermore, PageMaker Macintosh has an extensible facility for file importing that should allow developers of new word-processing programs to create special drivers that allow their files to be placed into PageMaker as well. As we saw in part 4C, PageMaker can also save text files to MacWrite, Write Now, or Microsoft Word formats, allowing you to use this as a tool to translate between programs.

For graphics programs, PageMaker can place standard bitmaps (those that have the PNTG file type) of the type created by MacPaint or FullPaint. It can place object-oriented files that have the type PICT and Encapsulated PostScript files. PageMaker can also deal with any data that is on the Macintosh Clipboard, allowing you to pass to it data of many different types.

PageMaker on the PC This program can place files created by different word processors. Table 4.1 is a list of all the files that PageMaker PC can recognize, along with the extensions these files must have for PageMaker to recognize them. These word processors include Microsoft Word, WordStar, XyWrite III, WordPerfect, Microsoft Windows Write, and MultiMate. Additionally, it can read and understand files that are formatted in the DCA format. If your word processor does not support one of these files directly, there is probably a means of translating the file to one of these formats.

Table 4.1 lists the file types that can be read into PageMaker for the PC, along with the extensions required by PageMaker to allow it to recognize those files.

If you have both the PageMaker products, you have a good way to translate files from PC word processors to the Macintosh. Simply create a PageMaker document on the PC, place into it the document you want to convert, save the file, and transfer it to the Macintosh. With the file open in the Mac version of PageMaker, select the block of text that contains the file, and choose the Save Text option from the file menu. This will save the file to the format you select in the dialogue box that appears. If you have a PC file in XyWrite format and want to use it in MacWrite, this is one of the best ways to assure that formatting is kept intact.

Table 4.1 PageMaker File Compatibility.

Application	PageMaker File Type
AutoCad	.PLT
DCA	.DCA or .RFT
Encapsulated PostScript	.EPS
In-a-Vision	.PIC
Lotus 1-2-3	.PIC
Microsoft Word (PC)	.DOC
MultiMate	.DOC
PC Paint	.PIC
PC Paintbrush	.PCX
Publisher's Paintbrush	.PCX
Lotus Symphony	.PIC
ASCII text	.TXT
TIFF files	.TIF
Windows Draw	.PIC
Windows GDI Metafiles	.WMF
Windows Paint	.MSP
Windows Write	.WRI
WordPerfect	.WP
WordStar	.WS
XyWrite	.XYW
MacPaint	.PNT

PC PageMaker can also read graphics files created by a wide variety of programs, also shown in Table 4.1. If you have a graphics program that is not supported directly, there might be other ways. For example, Microsoft Windows on the PC supports a clipboard function much like that of the Macintosh, so any information that you can place on the Windows Clipboard can be pasted into PageMaker PC.

Graphics images can be transferred to the Macintosh from the PC in the same manner. An image placed into a PageMaker PC document can be placed on the Macintosh Clipboard after the PageMaker document has been opened on the Macintosh and then can be treated just as any other Macintosh data.

The two PageMaker products provide a great interface between the Macintosh and the PC and are a model for other companies developing software for both machines. In a network situation, these two products can be set up in such a way that their operation across the network is virtually transparent to the user.

Ventura Publisher

In the PC world, another popular desktop publishing program is Ventura Publisher. Ventura Publisher is a very powerful program featuring style sheets and other features that make it ideal for working with longer documents.

Ventura Publisher can accept text from a wide variety of PC word-processing programs, including Microsoft Word, WordPerfect, WordStar, and MultiMate. And, like PageMaker for the Macintosh, it can work as a translation program, through its file/type/rename function. This function, which is found on the edit menu, allows you to save a word-processing file from one format to another. For example, a WordStar file can be translated into Microsoft Word in this manner and provide a better translation than does Microsoft's Convert utility.

There are a couple of caveats that are necessary to state about working with Ventura Publisher in a network environment. First, VP is a memory-hungry program; it likes to have virtually all the 640-kb DOS memory available to it while it is operating. Both TOPS and PC AppleShare use a fairly large amount of memory to operate (as do other network programs). Either Ventura Publisher will not work at all while memory-resident software is installed, or the size of the documents you can work with will be severely limited. It's best to shut down your system from the network and remove the network software from memory before starting Ventura Publisher. If you need to work with files that are on a network server, copy them to your local disk first.

Second, Ventura Publisher has a unique way of dealing with the files that you place into its chapters. Most desktop publishing programs, such as PageMaker, make a copy of the document that you are inserting into the desktop publishing program when you place the file. Ventura Publisher operates differently. Instead of making a copy of the file, it "remembers" from where on the system (i.e., the disk and directory) that file was loaded. The next time you open the chapter in question, Ventura Publisher looks into the same location on disk to find the file. There are a couple of consequences to this. First, if the file is loaded from a network server, Ventura will attempt to load the file from the server when next you open the chapter. If the server was previously mounted as drive D:, that's where it will look the next time. If you do not have any servers connected the next time you open Ventura or if they are mounted in a different manner (say, to drive E: instead), the search will fail. You will need to do some work to reconstruct your chapter. For this reason, it's best if, before placing the file into Ventura, you copy it to a special location on your local disk.

Other Desktop Publishing Programs

Aside from PageMaker, there are now a score of desktop publishing programs available for the Macintosh and the PC. Most do not have the capability of the two leaders to work with files created on other machines, although there are many otherwise very capable programs available (e.g., XPress from Quark and ReadySetGo from LetraSet). Undoubtedly, as these programs mature, the capabilities of them to receive data from more and more programs will increase.

CHAPTER 5

Managing the Network

Managing a computer operation, especially one that involves a network of mixed machines, is a topic large enough for a book in its own right. The duties of a network administrator can be very defined, as they are in AppleShare, or there can be no network administration required at all, as is the case with TOPS.

However, in any organization, some individual (or department, if the organization is large enough) eventually ends up taking a major responsibility for computer education and support. In a large organization, that will probably be some branch of the data-processing department. In a smaller one, someone may perform the function part-time, and it might not even be an official part of the job. In any case, people like to have to go to one person to ask questions, get support, find out about the latest developments, and much more. Vendors, too, whether in a large software company or the local computer store, appreciate having calls for support come from at a least a limited number of individuals in an organization. In a networked office, that person will, by default, become at least the de facto network administrator.

Whether or not there is an official network administrator, most operations work best if some responsible party is available to handle many of the responsibilities of network administration. These will range from the specific rights and responsibilities assigned by AppleShare to general, often undefined, responsibilities for establishing procedures and dealing with vendors. In the latter case, these responsibilities will vary considerably from organization to organization. Much of the information in this book is addressed to the administrator of the network, but in the following pages, we'll discuss some of these responsibilities and talk about how to meet them.

Specific Responsibilities

As discussed in chapter 2C, the network administrator has specific, defined powers and responsibilities. Only the network administrator may assign registered user names, and only the network administrator may put users into groups. The AppleShare Admin program is used to perform both these functions.

When registering users, you need to remember a couple of things. First, of course, you should make sure that each user name is unique. This is not very hard, but you should consider some things. It is not necessary, for example, that the user name on AppleShare be the same as the actual user name. It can, for example, only consist of the individuals first name and last initial, so that John Doe becomes JohnD. If you are using a scheme such as this, you should decide on some mechanism to avoid confusion between John Doe and John Deere. Both the John's won't have to worry about their own names—it's easy enough to remember your own registered user name—however, the scheme you have used to assign these abbreviated names should be consistent enough so that other users can tell the two Johns apart.

Passwords should function in much the same way. Since AppleShare's Access Privileges are one of its most important and powerful features, the passwords you assign must be meaningful to the individual using the password but not so meaningful that others can easily figure out the password. Using the individual's middle name, for example, might seem like a good scheme, but if the personnel records of employees were to become available, it would be easy for others to find out passwords. Some sort of random string of characters is best to use, as long as it is not easily decipherable by others.

Grouping users in AppleShare is very important as well. In most cases, groups will be assigned according to the department to which the various individuals belong. However, keep in mind that additional groups can be set up. Such groups might include all the managers at a particular level or all the corporate officers.

Data Backup

The most important thing that can be said about data in general is that you should back it up, and back it up again. This is the most important piece of advice one experienced user can give to another, which is why it is a separate section. In many ways, considering the time and money that could be involved, this is the most important advice in this book. Ignore it at your own peril. Those who do not will eventually learn their lesson when they have to reconstruct lost data. But that can cost both the individual and the organization lost time and money. At least three levels of backup should be kept at all times.

Individual Files

All users should be responsible for keeping backups of their own files. Whether or not these files are actually located on a file server, all users should make backups of their important files at least once a day (or whenever the existing backups are obsolete). Usually these file backups can be accomplished by the utilities provided in the operating system of the computer (using the Finder's copy function or the DOS copy command), without having to resort to dedicated backup programs.

Hard Disks

All computers that have hard disks on them, whether or not they are servers, should have those hard disks backed up daily. Usually, commercial backup programs can do incremental backups that only back up to floppy or tape the files that have been changed since the last backup. This saves some time in the daily backup process.

The Entire Company Data Files

This procedure can be very complicated, but regular backups of all the data on the network should be made and stored offsite. It is not a bad idea to do this weekly. Ideally, this will include all data on hard disks that are servers on the network, as well as data that is kept on floppy disk. In case of robbery, insurance covers the cost of stolen computers, but it cannot cover the cost of lost data.

The network administrator should establish similar guidelines and make sure that all users follow them. Remember, when an individual in an organization loses data, the entire organization loses the data, too. It might take a few minutes and cost a few dollars to back up data each day, but it's well worth the price.

General Responsibilities

Beyond the specific responsibilities already mentioned, the administrator or administrators should handle a wide range of general responsibilities. These responsibilities are not presented in any hierarchy: The needs of various organizations will cause different priorities to be placed on the different items.

Solving Problems

Sometimes it seems as if nothing will work right with a computer. This is compounded when a network is involved. The network administrator or manager is responsible for getting to the bottom of all problems associated with computer use in an organization, whether or not the problem seems to be related to the network.

In the case of the network, this should include setting up the actual, physical network. If someone else (e.g., an outside consultant) is brought in to do the setup and initial training, the administrator should be familiar enough with the installation to do much of the later troubleshooting. Many of the problems that are encountered in day-to-day computer use are easy to figure out, and working with these problems can save the organization money that it might otherwise spend on outside consultants. At the very least, the administrator should have a basic enough knowledge of the network and how it functions to perform some of this common troubleshooting.

Problem solving has a wide set of ramifications. It can range from making sure that the network is connected correctly to helping users decide on the best means of solving a particular problem. In all cases, if requests for help are channeled through one network administrator, it at least eventually makes problems easier to solve.

Establishing Standards and Practices

When groups of people work together, they must agree on certain standards. If you are working without computers, everyone must agree on a certain physical filing method. Everyone should be able to find the materials they need when they need them. This is equally important in a computer system; people still need to find things, to be sure of where things are, if they are on a hard disk, just as if they are in a filing cabinet. The network administrator must establish these standards and make sure that all others adhere to them.

What is important is not so much the particular scheme of organization of materials on a hard disk but the consistency of that scheme. Certain types of files should be on certain server volumes or in certain folders.

Beyond establishing methods and locations for storing files, the network administrator should establish a number of other standards and practices. In a network with only one or a few laser printers, for example, conflicts between various users who need to use the printers at the same time are bound to occur. Establishing standard practices for scheduling printing can eliminate some of these conflicts and put in place a standard method of adjudicating between the conflicts when they do arise.

These are just two of the types of standards that should be established. Others will arise as the network grows and matures and as the users (and administrators) become more familiar with the network functions. Again, what is important is that standards are established (although that becomes important) but that all users are aware of the standards.

Evaluating Software and Establishing Specifications

New software appears almost daily. Microcomputer users are constantly hit with a barrage of new product literature and advertisements especially if they are magazine readers. In a situation with many users on the network, many individuals will have their own preferences as to which programs to use and will want to always be using the latest and greatest program. In general, that is to the good: New software brings new capabilities, making work easier or allowing people to do things they could not do before.

However, there is a down side to the flood of new software. New programs, and even new updates to old programs, can be expensive, especially if several or many copies must be published for individual network users. Additionally, new programs may have bugs in them that can cause havoc on a network. An unstable program run on a TOPS server, for example, can crash, perhaps causing data loss to those who are using that server disk.

The network administrator must also establish guidelines as to which software is used on the network and evaluate which new programs should be purchased. Such software evaluation probably should not be left to one individual, who might not always be conversant with the specific needs of the individual user. However, the network administrator, before permitting software to be used on a network, should at least ascertain—perhaps through experimentation—that the software is not hostile to the network.

Updating old programs is a special case of adopting new ones. Sometimes, manufacturers will charge for updating programs, which can be quite expensive if there are many users on the network accessing that software and thus requiring updates of many copies of the same program. Before committing to an update of a program, make sure that the update includes features that are needed or are very desirable. If the update is just to fix some problems with the existing version, then that update should either not cost very much or be free. If it offers significant enhancements, then it might be worth the investment. Many users on the network, though, will want the latest and greatest version just for the sake of having it. Before committing your organization to purchasing the new version, make sure the enhancements are significant for your operation.

In general, the network administrator should be somewhat conservative regarding new software and updates. This is especially true regarding system software—the software that affects all other software and users on the network. On the Macintosh, as network administrator you should be conservative about acquiring such things as desk accessories, printer spoolers, and similar things that can affect the performance of other software (especially on systems that are TOPS servers). The same is true with the PC: Carefully test all memory-resident programs before putting them into wide use on the network. The ways that programs interact with one another is complicated. It is better that you wait a few days or weeks to install that latest and greatest utility than to jump right in and be sorry about it later.

Software Licensing

As discussed in the preface, all organizations must adhere to the license agreements implicit in the purchase of software. If you have many users of one program, you should have a legitimate copy of the software (unless the software was purchased under a specific site license) for each of those users. The network administrator is responsible for making sure that all software is being used legally, (i.e., within the boundaries of the agreement).

This might mean that for each user of a specific program there is a legitimate copy of that program, purchased for that user. It will also mean that software is not copied freely for the home use of employees or that copies are not sent to other sites. The network administrator should take responsibility for educating users as to these responsibilities.

Dealing with Vendors

Dealing with vendors includes purchasing new hardware or software, purchasing training, overseeing installation, and handling technical support after the purchase.

Whether it's your local computer store or a large software manufacturer, vendors like to deal with just one individual from any organization. This is especially true of support problems. If just one individual (or a small group of individuals) is in charge of getting support from vendors for computer problems, the number of problems for which an organization has to get support will eventually decrease. Ideally, the individual users will turn to the network administrator for support. Eventually, the administrator will become familiar enough with the problems that many of them can be solved without having to call the vendor or software manufacturer.

The same is true for purchasing hardware or software. Only the administrator, or other party who is actually responsible for purchasing goods, should deal with vendors regarding possible purchases. If an individual user needs a particular piece of hardware or software, that request should be put through a channel that evaluates both whether the purchase is actually necessary and whether the particular item is the one needed (as opposed to a competitor). In this way, the resources of the institution are better directed, and different employees, who might be seeking the same solutions, are not working at cross-purposes to one another.

Not least among the duties of dealing with vendors is reporting back to them their own performance. Let them know when they or their software performs well, and also let them know what new features you would like or what things do not work correctly.

Monitoring System Performance

Few networks, hard disks, or file libraries will be set up from the start to perform their best. They must be tweaked or tuned from time to time, like a car, to achieve their best performance. This might include changing the network configuration to put resources near those who use them frequently and making sure that oft-used files are available handily to those who need them. This should include the human element in the network, which is often the most difficult to evaluate.

On a network, performance is a complicated result of many things. There are some programs, such as Traffic Watch from Farralon Computing, that can help you determine where the bottlenecks are. Moving server hard disks around on the network or changing the physical configuration of the network can improve performance dramatically.

You can also monitor the performance of the network by monitoring the complaints or praises it gets from users. Create some means on the network (e.g., a drop box) that individuals can use to let you know how the system is performing (for good or ill).

Maintaining a System Log

A system log is essentially your own internal documentation that describes your installation. It should include information about what computers are in use and where. It should also include information about the various hard disks, printers, modems, and other peripherals on the network. Software, too, should be included in the system log.

The reason for keeping a system log is straightforward. It is beyond most of our capacities to remember which items are being used where or how they are set up. A log provides a place to record this information so that the administrator (or others) can refer to it later.

For Macintoshes, the system log should include what applications programs are stored on which hard disks, the fonts and desk accessories that are in use, which hard disks are connected to which machines, and more. In a TOPS network, it should probably also include information about which users are using which server disks and which disks are always published. Insofar as it is possible, include a map showing which files are in what folders or subdirectories on the server disks.

For a PC, the system log should include all this information plus some. Interface boards on the PC can have different settings in order to work correctly: which DMA channels or interrupts they are using and the like. A system log that includes these items will be of immeasurable help when you need to change the configuration of any system or of the network.

User's experiences, tips, horror stories, bug reports, and more should all be entered in the system log in some form. This can help you improve the performance of the network and make troubleshooting much easier.

Establishing a System Library

A system library includes such things as software documentation, a copy of the system log, books, and reprints of articles. By keeping a system library and establishing procedures for borrowing material from that library, you can ensure that all users on the network have information available to them when they need it.

Educating Users

The network administrator is also responsible for user education. At its most basic level, this includes making sure all those on the network know how to the use the network to its best advantage and educating them as to the standards and practices established for the network. At a more complicated level, this will include training for all new applications that are installed on the network. The system library and system log are excellent tools for user education.

Whether or not you are assigned specific responsibilities as a network administrator or if the job just falls on you because you are the most experienced user in the organization, these tips can help you and your co-workers get more out of the network.

APPENDIX

A

Glossary

There are five types of terms defined here:

1. TOPS specific terms are noted with the abbreviation TOPS before the definition.
2. AppleShare terms are noted with AS.
3. Macintosh terms are noted with MAC.
4. PC terms are noted with PC.
5. Standard personal computer or networking terms, whose meanings are the same across the various computers and networking systems discussed here, are not set off.

If a definition has a general meaning, as well as a specific meaning in one of these areas, the general meaning is listed first.

access—To use a network resource.

Access Privileges—(AS) Those privileges that are given to, or withheld from, users to open and alter a folder (or directory) and its contents. The Desk Accessory that allows users to review and set the privileges of a folder.

Admin key—(AS) The password that is entered by the AppleShare administrator when accessing specific functions of the software.

administrator—

1. Sometimes called the "network administrator," this is the person responsible for setting up and maintaining the network.
2. (AS) The person who sets up the server. This person registers users and their passwords, creates groups, and maintains the server.

alphanumeric—A character that may be a letter, a number, or a special symbol. The typewriter-like keys on a computer keyboard are also called the "alphanumeric" keys, as opposed to the arrow or cursor keys, which do not generate a character when pressed, or the F keys, which generate special codes that programs can interpret in their own ways.

ALT key—(PC) Special key on the PC keyboard. When this key is held down and another key is then pressed, the code the second key generates is modified. It is typically used to send a command to the software, as opposed to generating a special symbol, although this is defined by the program running.

ANSI—American National Standards Institute. An official body that publishes standards in computing and related fields, such as computer languages.

AppleTalk connector—A piece of equipment consisting of a connection box, a short cable, and a plug that allows a device to be part of an AppleTalk network. AppleTalk connectors for the 512-kb Macintosh, the LaserWriter and LaserWriter Plus, and the Apple and TOPS interface cards for the IBM PC have twenty-five pin plugs. Connectors for the Macintosh Plus, SE, and II as well as for the ImageWriter have nine pin connectors.

AppleTalk network—The cables, connectors, and software that link computers and peripheral devices (e.g., printers and modems) in a local area network. (See *Local Area Network.*)

application—A program that allows you to create, alter, print, or view documents.

ASCII—American Standard Code for Information Interchange. A system for representing numbers and letters by a pattern of seven bits. Most microcomputers use ASCII to represent the numbers and letters they can display. ASCII defines only 128 characters, of which the first thirty-two are for control characters; the rest are for letters and characters.

Both the PC and Macintosh have extended the ASCII codes to contain 256 characters; the machines agree on only the first 128 characters, and then only the printable characters from 32 to 127 are exactly the same for each machine. Certain standard control characters, such as the tab, the return, and a couple of others, mean basically the same on both machines.

Since ASCII codes only number from 0 to 127, this leaves out a great deal of the kind of information that can be generated from within a program. There is no way, within ASCII, to represent an underlined or bolded character, for example.

See appendix B for a table of ASCII characters on both the PC and the Macintosh.

ASCII file—A file that contains only the alphanumeric and control characters described in the ASCII format. An ASCII file will typically contain very little of the formatting that is possible from within a program.

Creating an ASCII file is usually accomplished by saving a file in a "nondocument" or "text only" format. At its lowest level, this is probably the most reliable way to transfer files from the PC to the Macintosh.

While an ASCII file cannot typically contain any formatting information, it can be structured. (See *DIF, SYLK,* and *delimited file.*)

attribute—

1. A characteristic of a file or a program that is set or controlled by the operating system.
2. (PC) Files on DOS disks can have four attributes: system, read only, archived, or hidden.

AUTOEXEC.BAT—A special batch file that is executed automatically whenever DOS is started. This file is used to include special instructions that the user wants to have *ex*ecuted *auto*matically on system startup.

background—A process that happens independently of things the user is doing. Print spoolers work in the background (i.e., they print while the user is able to do other things in the foreground). Typically, background applications use fewer of the computer's CPU resources. (See *foreground.*)

backup—

1. The practice of keeping a separate copy of important files and documents.
2. The act of saving all data on a hard disk onto tape or floppy disk.
3. The separate copy so made.

batch file—(PC) A text file containing a series of DOS commands. Instead of typing each of the commands separately at the keyboard, they can all be executed in sequence simply by typing the name of the batch file at the DOS prompt.

binary file—A file that is composed of data that is not readable as ASCII and that stores programs, pictures, and some other information.

BIOS—(PC) Basic Input Output Services built into the ROM of a PC. A collection of functions that handles the low-level operation of the computer. It is the BIOS that makes the PC a PC, and it is the BIOS that is copyrighted by IBM. Numerous companies have cloned or made compatible BIOS chips.

bit—The basic unit of computer storage. A bit can have a value of either 1 or 0. Eight bits make up one byte.

bitmap—A file—usually of a graphic image—that consists of any array of dots (bits) that can either be black or white.

board—A printed circuit card that fits inside a computer, giving it capabilities it probably did not have without the board, such as allowing a PC to connect to the AppleTalk network.

boot—To start a computer.

bridge—A device that is used to connect separate networks.

buffer—Any portion of RAM that is set aside as a temporary holding tank for data going somewhere else. This RAM can be part of the standard RAM of your computer, or it can be located externally to it. There are many print buffers that consist of boxes that sit between your computer and printer. Such buffers, while speeding the time control of your computer returns to you, do not speed the printing process itself.

bug—A programming error that causes the program to act in ways contrary to the way its designers intended. At best, bugs are irritations that get in the way of your taking advantage of all features of a program. At worse, they can cause significant loss of work or destruction of data.

All software has bugs; the best programs simply have fewer of them.

byte—Eight bits. A byte typically represents one character. It is the common unit of measure for computer memory and storage.

cable—A bundle of wires with connectors on the ends. Used to connect various items to one another.

cache—A special instance of a buffer; a memory buffer between the CPU and disk storage. Data that is requested frequently from the disk is held in the cache so that successive requests for it come from RAM instead of the disk, thus increasing performance. On the Macintosh, the amount of RAM cache is set from the Control Panel Desk Accessory.

card—See *board.*

carriage return (CR)—

1. In the ASCII system, the carriage return is represented by code 13 and is produced by the return or enter key. In a file, the carriage return code usually indicates the end of a line or of a unified group of data.
2. (PC) A carriage return is generated when the enter key is pressed on the keyboard. Many programs use this key to discern when an entry is finished.
3. (MAC) Also generated by the return key. The separate Mac enter key may or may not—depending on the application—also generate a code 13.

cell—A specific location in a spreadsheet, usually referred to as "Row 1, Column 1" or "Row A, Column 1."

character—Any printable or nonprintable item that can be generated by pressing a key on a keyboard or can otherwise be displayed on the screen. A character is represented on disk and in memory by 1 byte.

Chooser—

1. (MAC) A Desk Accessory that allows the user to give the Macintosh several types of information, including, among others, what printer to use, what AppleShare volumes to log onto, and what network name to use.
2. (AS) In the PC version of AppleShare, the Chooser is one option that is accessible through the DA utility. It allows one to select and use network resources.

click—Using a mouse to point at some place on the screen, then press and release the button quickly.

client—(TOPS) A station that is using resources (disk or printer) that have been published by a server is that server's client.

clipboard—

1. (MAC) The holding place for items that were last cut or copied. The clipboard is never accessed explicitly: When you cut or copy, items are placed on the clipboard implicitly. When you paste, information is copied from the clipboard to your document. The clipboard typically only contains one object—whatever was last cut or copied. Pasting does not affect the clipboard; you can paste something as many times as you want.
2. (PC) Microsoft Windows implements the clipboard in much the same manner as the Macintosh.
3. Other programs (e.g., WordPerfect) and the WordPerfect Library implement a clipboard as a text file on disk.

close—To remove a file from the active part of the program you are using, as when you use the Close menu item in most Macintosh programs.

comma—ASCII code 44, entered by pressing the comma (,) key on a keyboard. Many databases use the comma to separate fields within a record, when they are creating ASCII text representations of their data.

command—An instruction given by the user to have a computer perform a specific action.

command key—(MAC) A special key on the Macintosh keyboard. When pressed along with another key, this key is used to give commands to the current program.

CON:—(PC) The DOS logical device that represents the console.

concurrent application—A program that runs simultaneously with other programs.

CONFIG.SYS—(PC) A special file on a DOS disk that is executed when the system is started and before DOS is loaded into memory. This file usually contains references to device drivers and instructs the system to use them. It will often contain other information about such things as the amount of memory to set aside for disk buffers and instruct DOS as to how many files it should allow programs to open at one time. At all times, the active CONFIG.SYS file should have at least these two lines:

```
files = 20
buffers = 20
```

Since the CONFIG.SYS file is activated before DOS loads into computer memory, the system must be restarted or rebooted for changes to CONFIG.SYS to take effect in the machine.

control character—

1. ASCII codes 1–32, which have no printable characters associated with them. Usually used to give instructions to an output device, such as a screen or printer. Older, character-oriented terminals used control codes to manipulate the cursor or text on the screen. Many dot matrix and daisy wheel printers perform certain actions when they receive a control character. Some common control characters discussed in this book are the carriage return (ASCII 13), linefeed (10), and tab (9).
2. (PC) Any key pressed by holding down the control key.

control panel—(MAC) A Desk Accessory that allows users to set certain system defaults for such things as the key repeat rate of the keyboard, the sound of the speaker, and the system date and time, among others.

copy protection—The nefarious practice followed by some software publishers that prevents their software from being copied.

crash—For a computer or peripheral (e.g., a printer or modem, though only rarely) to suffer such a disastrous chain of events that the software operating it no longer suffers. Often it will need to be reset or even turned off and turned on again to be used.

create—To cause a new file to be placed on to a disk.

creator—

1. The program that writes the file to disk.
2. (MAC) The Finder information that contains a four-letter code that the Finder uses to open a file and to present its correct icon.

cursor—

1. A flashing or stationary marker that indicates where data is expected to be entered on a display.
2. (MAC) The cursor also can refer to the mouse pointer that moves around the screen as the mouse is moved on the desk. The insertion point usually is used to refer to the standard cursor.
3. (PC) The cursor may be a small box or underline.

custodian—(AS) A special user of an AppleShare network. This user—who is not really any specific individual—becomes the owner of all server volumes and folders when the volume is first created under AppleShare. Ownership of all otherwise unassigned folders is given to the custodian. The custodian is typically the administrator and has a separate password. The administrator can log on to the server from a workstation just as a registered user does and can perform certain functions once logged on.

database—

1. An application program that allows you to create ordered information that can be sorted, printed, and reported upon.
2. The file that is created by a database program.

dedicated server—A file server that is not, or cannot perform any function simultaneously with its file-serving function. AppleShare servers are dedicated to the serving function.

default—A value that is presented to the user by a program as a standard choice.

delete—(PC) The DOS command that allows erasing of files from a disk, often written as "del."

delimited file—A text file in which fields and records are set apart (delimited) by certain characters. Virtually any character can be used to delimit files, as long as both the program that is writing the file to disk and the one that is reading it agree on the delimiter character. The most common characters used as delimiters are the comma (ASCII code 44) and the tab (ASCII code 9). The tab character is mostly used by Macintosh programs; the comma, by PC programs. When a comma is used as the delimiter, the program must put fields that naturally contain commas (e.g., the last name, first name combination as in "Bear, Smokey") into quotes to keep the destination program from reading them as two separate fields.

deselect—(MAC) To cause the currently selected item (e.g., icon, text, and volume) to no longer be selected. Usually this is done simply by clicking somewhere other than the current selection.

Desk Accessory—(MAC) Any of the many programs that are available under the Apple menu. These are small programs that are, usually, available no matter what other program may be running.

desktop—

1. (MAC) The grey area that appears behind any window that is visible on the Mac screen.
2. The display of windows that is visible from the Macintosh Finder.
3. The invisible file on each disk that stores icon and other information about each file.

desktop publishing—The software technology that allows graphics, fonts, and text to be mixed and presented in an attractive manner on a printed page, using a high-quality printer such as a Laser Printer or typesetter.

device—Any piece of equipment that can be attached to a network: a computer, a printer, a file server, or other peripheral, including bridges.

device driver—

1. Software that allows a computer to work with hardware that is not part of the standard hardware that is used with a computer. CD-ROM disks, mice on the PC, and AppleTalk interface cards are examples of hardware devices that require the use of drivers.
2. (PC) The software that works as a device driver typically has an extension of .SYS (for System). Most often, a reference to this software is placed in a CONFIG.SYS file that is read and executed by the computer before DOS is loaded.

dialogue box—

1. (MAC) A box appearing on the screen that requests information from the user. Sometimes they consist of nothing but warnings; other times they might request significant information.
2. (PC) While a standard interface for the PC has yet to emerge, many programs implement dialogue boxes similar to the Macintosh.

DIF—Data Interchange Format. A file format created by Software Arts to enable information exchange between its VisiCalc program and other software packages. Although VisiCalc is gone, the interchange format remains and is used by many other programs to facilitate data transfer.

directory—

1. (PC) The listing of files in a subdirectory, disk, or volume.
2. (MAC) As with the PC, only the listing may be pictorial, consisting of icons.

disk—A device onto which a computer can write information so that it can be read later.

disk server—A network system that allows users to access disks on remote drives. A disk server usually allows only one user to access a volume at a time, so it must be divided into volumes before it can be used. (See *file server.*)

distributed—Processes that take place at more than one location are distributed processes. For example, TOPS permits many computers to act as file servers; therefore, it is a distributed file server network.

DMA—(PC) Direct Memory Access. A method by which interface boards communicate their information directly into the memory used by the CPU.

document—Anything created by an application program; information that can be entered, modified, viewed, or saved. (See *application file.*)

DOS—

1. Disk Operating System. The control software that oversees all the functions of a computer: running programs, dealing with the user, printing, and more.
2. (PC) MS (for MicroSoft) or PC (for Personal Computer) DOS refers to any of the operating systems written by MicroSoft for the Intel series of microprocessors. DOS does not refer to the OS/2 operating system announced in early 1987.

double-click—To click the mouse button twice in quick succession. (See *click.*)

down—Not working or turned off.

download—To copy information from a distant computer to a local one.

drive—When referring to a floppy disk drive, the hardware device into which the disk is inserted and which reads information from the disk into the computer or writes information from the computer onto the disk. When referring to a hard disk, the sealed disk as well as to the various mechanisms that allow it to operate.

driver—See *device driver.*

editor—An application that allows you to type, edit, and print text. Different from a word processor in that it typically inserts few formatting codes into text and, thus, might have fewer formatting and printing options than most word processors.

electronic mail/e-mail—A system for storing and distributing messages that are sent among users over the network. Refers to both the system, as well as the content of the mail (i.e., "Use e-mail to let me know when the system will be down" and "I read your e-mail on that topic").

encryption—The process of scrambling the information in a file so that it is not visible to those without the correct password.

error—A mistake. Something that goes wrong.

escape key—(PC) The key marked ESC generates ASCII code 27. Most programs use this key to allow you to instruct the software to stop doing something (i.e., escape from the current menu or program).

Escape codes are also used to instruct many dot matrix printers to change the manner in which they are printing. Many PC programs insert these codes, which typically consist of the escape character followed by another, into the text as they send it to the printer.

everyone—(AS) The user category to which you can assign privileges for any user who has access to the server.

exporting—The process of writing information from one program (the source) into a format that can be read by another program (the destination).

extension—(PC) The three-character ending of a file name. It is always preceded by a period to separate it from the file name proper. Usually the extension is used to tell what type of file a DOS file is or what program created it or both.

field—One of the items that makes up a record in a database. In a name and address file, "zip code" might be one field, "phone number" another.

field locking—The ability of some multiuser software to prevent two or more people from changing the information in one field at a time.

file—Any named, ordered collection of information stored on a disk. Documents, applications, programs, and system files on disks are different kinds of files. In a directory listing, every item in the listing is a separate file.

file server—

1. A file server makes files available to others on the network, perhaps allowing several users to access the same file at once. (See *disk server*.)
2. (TOPS) Any station on the network that has made its local resources available to others on the network.
3. (AS) A combination of AppleShare software, one or more hard disks, and the Macintosh to which the hard disks are connected and on which AppleShare is running.

file sharing—The ability of some multiuser programs to allow more than one user to access and perhaps alter a file at a time.

Finder—(MAC) The program that presents the DeskTop to the user. Sometimes mistakenly referred to as the DOS of the Macintosh.

F key—

1. (PC) Any of the ten keys that are arranged to the left (or along the top) of the PC keyboard. These keys are numbered from 1 to 10 (newer PC keyboards have twelve). There is no fixed meaning for any of these keys

(although some standards are emerging—such as using the F1 key to access help); they are assigned a meaning by the program that is currently running.

2. (MAC) Any of the small programs that can be made part of the Macintosh System File, allowing common (or uncommon) functions to be accessed by holding down the command and option keys and then pressing a number key on the top row of the keyboard. Several function keys are built into every Macintosh: command-option-3, which saves the screen as a MacPaint file, and command-option-4, which prints the current screen.

floppy disk—A magnetic disk, either 5-1/4-in. or 3-1/2-in., that contains information that can be read and altered by a computer.

folder—(MAC) A holder of documents and applications to present a more organized way of dealing with large numbers of files. Analogous to directories or subdirectories on the PC. Visible on the DeskTop with an icon that resembles a paper file folder.

foreground—Any process that is executing on a computer and using most of the computer's resources is a foreground application. (See *background.*)

format—

1. In printing, the manner in which information is presented on a page.
2. In files, the file format is the manner in which software structures information in a file.

format command—(PC) The DOS command that allows you to prepare a disk for use.

freeware—Term coined by Andrew Flugelman to define the distribution method for his program PC-Talk. Freeware software is typically not available though the usual software channels (e.g., stores and mail order). Users can receive copies though user groups, over electronic bulletin boards, or from other users at no charge. If users find the software to be useful, then they are asked to send a licensing fee directly to the program's author. (See *public domain* and *shareware.*)

glossary—A series of characters that can be entered or invoked by some other menu or keystroke selection. Unlike a macro, a glossary usually only contains text that replaces selected or typed text. (See *macro.*)

group—(AS) The user category to which you can assign Access Privileges for members of groups created by the administrator.

guest—(AS) A user who has logged onto an AppleShare server without a registered user name or a password. The guest can create folders and files on the server but cannot assign privileges to those files and folders without first reregistering with a user name and password.

hard disk—A disk that is permanently sealed into a drive. The disk drive may be an integral part of the computer system (i.e., physically built into it) or be a separate piece of hardware connected directly to the computer.

hardware—All the physical components of the computer system.

HFS—(MAC) Hierarchical Filing System. A hierarchical structure of folders and subfolders. HFS was introduced in late 1985 with the first Apple hard disk for the Macintosh and was built into the 128-kb Macintosh Plus ROMs. (See *MFS.*)

icon—A graphic representation of a file, device, or instruction.

importing—The process of reading information or data that was created by one program (the source) into the format of another program (the destination).

insertion point—(MAC) The vertical bar (which may or may not be blinking) that shows where the next typed characters will appear. It is similar to, but different than, the mouse cursor or pointer. Typically, the pointer will be clicked at a particular point for the insertion point to appear at that point.

Some DOS programs—particularly those that use a mouse—will work in a similar manner.

See *cursor.*

installation—

1. (PC) The process of making software ready to be used on a specific computer configuration (i.e., copying it to a hard disk and informing it of what kind of printer or monitor is being used).
2. (MAC) The process of adding or changing information in the System file or System folder of a disk.

interface—

1. With software, the user interface is the manner in which a program or computer presents its capabilities to the user and includes the appearance of the software and the structure of commands it features.
2. With hardware, a connection between two devices or a way to connect two devices.

interrupt—

1. To suspend or stop an operation.
2. (PC) The mechanism by which interface boards in a PC's slots inform the computer that they have data.

kilobyte—1024 bytes, abbreviated kb.

LAN—See *Local Area Network.*

launch—To start an application.

license—An explicit or implicit agreement between a software publisher and purchaser that defines the terms under which the purchaser may use the software.

licensing fee—The cost of obtaining software; the price paid for it.

linefeed—ASCII character 10. The linefeed character is part of a carriage return/linefeed combination that DOS uses to indicate the end of a line.

local—Any resource (e.g., disk, printer, or modem) that is attached directly to the computer is a resource local to that computer, as opposed to devices that are available to a computer through network services.

local area network—A group of devices linked together that communicate using the appropriate software.

log off—

1. To disconnect from a server.
2. (TOPS) Also called "unmounting."

log on—

1. To connect to a server.
2. (TOPS) Also called "mounting."

look and feel—The characteristic appearance of a piece of software or computer. A program's look and feel may include its appearance on screen, the commands it can execute, the structure of its menus, and more.

LPT1—(PC) The DOS logical device that refers to the primary printer used by the System. Usually LPT1: refers to the device that is connected to the first parallel port on the machine, but it can be redirected to other logical devices.

MacBinary—(MAC) A standard format used by Macintosh telecommunications programs. This format allows communications programs to send, along with a file, information about the file (e.g., its file type and creator, icon, and creation dates).

macro—A sequence of commands or characters that can be entered or invoked by some other keystroke or menu selection. Macros can contain simple text or quite complicated instructions that can resemble a programming language. (See *glossary.*)

mail—See *electronic mail.*

Make Changes—(AS) The Access Privilege that allows one the right to alter or change a folder's contents.

megabyte—1024 × 1024 bytes, abbreviated Mb.

memory—The part of the computer that holds the program and data that is currently in use. (See *RAM.*)

menu—A means of instructing a computer or program which operations the user wants to perform.

MFS—(MAC) Macintosh Filing System. The Macintosh operating system (also known as the flat filing system) that was used by Apple prior to the introduction of the Mac Plus. Through the Finder, the Macintosh provided a system of folders for organizing files on a hard disk. However, the structure of the disks did not provide any way for programs other than the Finder to use these disks. (See *HFS.*)

DOS subdirectories available over TOPS are MFS disks.

modem—A device that allows computers to communicate with other, distant computers over telephone lines.

mouse—A hand-held device that moves over a desktop. A rolling ball under the mouse monitors movements, which then moves the cursor on the screen. The mouse is used to draw, select text or other items, and move the text insertion point. The mouse has one button, which can be clicked, held down, or double-clicked. Additionally, certain keys (e.g., the shift or command keys) can work in consort with the mouse, changing the meanings of its clicks.

On the PC, the function of the mouse—if any—is determined by the application program. Many programs cannot take advantage of one. Additionally, while the Macintosh mouse has one button, PC mice may have two or three, and the function of additional buttons can vary considerably.

MS DOS—When DOS is sold and/or used on a non-IBM PC compatible, it is usually called "MS DOS" for Microsoft DOS. (See *DOS.*)

multitasking—The ability of a computer or program to handle more than one operation at the same time.

multiuser—

1. A computer that is designed to be used by more than one person at a time, usually through separate terminals. Mainframes and minicomputers are usually multiuser.
2. A program that is designed to allow itself, or its files, to be accessed by more than one person at a time.

name—See *network name.*

network administrator—The individual charged with maintaining the network.

network name—The name by which a computer, printer, or volume is known to others on the network.

node—A location on a network.

nondedicated server—A file server that is also capable of performing other tasks at the same time that it is operating as a file server.

open—To cause a file or program to be executed.

operating system—

1. The specific set of instructions that give a computer the ability to run programs, interact with the user, read and write to disks, and work with printers and other devices.
2. (PC) PC DOS or MS DOS, including the full range of programs and utilities that are needed to perform most elementary tasks on a PC.

option key—(MAC) A special key on the Macintosh keyboard. Holding down this key and then pressing another alphanumeric key changes the codes generated by that key. It is most often used to generate special symbols that are not available from the Macintosh keyboard.

OS/2—(PC) The name for the advanced operating system for PCs featuring the 80286 and 80386 processors. This operating system was introduced in April of 1987 by Microsoft, and IBM and is scheduled for release in early 1988.

owner—(AS) The user category that the owners of folders or volumes use to assign Access Privileges to themselves. An owner has the exclusive right to assign Access Privileges to a folder or volume.

parallel or parallel interface—(PC) A means of connecting a peripheral device to a computer; the means by which some devices communicate with the computer. A parallel connection usually consists of different wires running along side one another from the computer to the printer. Computers usually communicate with peripherals in bytes. In the parallel connection, an entire byte is sent at one time, with the different bits running next to each other (or parallel to one another) along the cable. (See *serial.*)

parameter—(PC) An optional instruction added on to a command typed at the DOS prompt.

password—A unique word or set of characters that must be entered to access certain volumes, folders, or files.

paste—(MAC) To place the contents of the Clipboard (whatever was last cut or copied) at the insertion point. (See *Clipboard.*)

path—(PC) The DOS command that tells the operating system where batch files and programs are to be found.

PC—

1. IBM's name for their line of personal computers.
2. Any IBM-compatible personal computer, running DOS.

PC DOS—When DOS is used on an IBM PC or purchased from IBM. (See *DOS.*)

peripheral—Any device (e.g., a monitor, modem, or printer) that is attached to a personal computer.

port—A hardware connection on a computer that allows peripherals to be attached directly to that computer.

PostScript—The page description language built into every Apple LaserWriter and many other compatible printers. PostScript is the language that the Macintosh (or compatible programs on the PC) uses to tell the printer what to put where on a page.

primary group—(AS) The AppleShare group with whom you are most often sharing the documents stored on the server. Primary groups, as are all groups, are assigned by the administrator.

printer—A device for putting text and/or graphics onto paper.

printer driver—

1. (MAC) A System file that allows you to print on a corresponding device attached to the network or directly to the Macintosh.
2. (PC) A series of instructions that various programs use to communicate with printers. Usually, each DOS program will contain its own set of printer drivers, and part of the installation process is to inform the software what printer driver to use.

printer resource—(MAC) See *printer driver*.

printing—That act of putting text and/or graphics onto paper. The act of communicating with a device that does that.

privileges—(AS) The rights that a user has regarding the ability to see or change files or folders within another folder. (See *Access Privileges*.)

PRN—(PC) The logical name for the connection to which DOS sends output for a printer.

program—A file that contains information instructing the computer to perform specific instructions.

prompt—

1. (PC) The A:> or C:> that DOS presents to the user when awaiting a command.
2. The DOS command that allows the user to change the information DOS displays when awaiting a command from the user.

public domain—Software that is available for anyone to use without cost. This software may be given away. (See *freeware* and *shareware*.)

publish—(TOPS) To make a volume or printer available to others on the network.

queue—The list of files waiting to be printed by a print spooler.

RAM—Random Access Memory. The main working area of any computer, where all programs operate.

RAM disk—A certain portion of the computer's random access memory that has been set aside to act as a disk drive. To the user, as well as to the computer, a RAM disk behaves just as a floppy or hard disk does, only much faster.

read—To bring the contents of a file into memory for examination or alteration; like open.

read only—

1. A file that can only be read (looked at); either the program or the operating system has marked the file "read only" so that it cannot be modified.
2. (TOPS) A volume that has been published as read only thus contains files that may not be altered by the client.

read/write—

1. A file that can be altered, as well as viewed by a user.
2. (TOPS) A volume that has been published as read/write, thus, contains files that may be altered as well as viewed by the client.

reboot—The process of restarting a computer.

record—One item in a database. For instance, in a name and address database, all the information that is associated with a specific name or entry (e.g., name, address, zip, and phone number) is part of that record.

record locking—The ability of some multiuser database programs to prevent more than one user from altering the contents of a specific record at a time.

redirection—(PC) The process of informing DOS that output, which is normally sent to one location (e.g., the screen or parallel printer), is to be sent instead to another location.

registered user—(AS) A user who has been given a name and password by the AppleShare administrator.

remote—(TOPS) Any resource that is available through the network is a remote resource.

reset—See *reboot*.

resident—(PC) See *TSR*.

resource—Any item that is shared over a network. A shared hard disk or a shared printer is a "network resource."

resource fork—(MAC) A part of a Macintosh file. In Macintosh program files, the resource fork usually contains information about menus and text presented in dialogue we boxes, icons, pictures, and the like. In a Macintosh data file, the resource fork may contain information regarding the formatting of the actual data contained in a file.

restore—

1. To take files from a backup disk or tape and return them to the disk (usually a hard disk) from which they were backed up.
2. (PC) The command that transfers files from disks created with the backup command and returns them to the hard disk.

ROM—Read Only Memory. Programming that is built into the computer. The Macintosh contains an extensive set of ROMs that contain items programmers may take advantage of to give their programs the characteristic look and feel of Macintosh software. The PC ROMs contain program code that allow the computer to start up from disk and also include a subset of the BASIC programming language.

root directory—(PC) The top-level directory of a disk or volume. All other directories—called subdirectories—are contained in the root directory.

Scrapbook—(MAC) The Desk Accessory that provides a storage place for frequently used pictures and text.

security—The concept of keeping important or sensitive files from being seen by unauthorized persons.

See Files—(AS) The Access Privilege that allows the right to open and copy documents and applications in a folder.

See Folders—(AS) The Access Privilege that allows the right to see folders within a folder.

serial—A means of communication between a computer and a peripheral device. In serial communication, data is sent to the peripheral one bit at a time. Both the interface ports (the telephone port and the printer port) on the Macintosh are serial ports. (See *parallel*.)

server—(TOPS) Any station that has published a volume or printer.

server folder—(AS) A special folder on each AppleShare volume that contains information about the server, including the information about folder privileges and registered users that the server needs to operate.

server report—(AS) A report that can be created with the AppleShare Admin program. This report includes the list of registered users, their groups, and information about how many folders and files a user has and the space they occupy.

session—(AS) A session is the logical pairing between a user and a device. The name of the first session you create is the same as the workstation to which you are attaching. Subsequent sessions with the same server have the same session name, except the last character of the session name is replaced by a number, beginning with 1.

shared—Any network devices that can be used by more than one user.

shareware—Software that can be acquired any number of ways: from other users, over electronic bulletin boards, and the like. The cost of acquiring the software is usually minimal; the creators of the program request that a sum be sent to them if the software is used.

shutdown—

1. (MAC) The Finder command that is used when the computer is to be turned off.
2. (PC) The TOPS command that halts TOPS on that machine.

single-user software—Software designed to be used and to create files that are used by one user at a time. (See *multiuser software.*)

site license—A special software license that gives the purchaser specific rights as to how software may be distributed, used, or copied within a specific location or organization.

software—A program, or set of programs, that instruct a computer to perform a series of operations.

spooler—A utility program that intercepts characters being sent to a printer (or other device), saves them on a disk, then sends them to the printer in the background.

startup volume—(MAC) The volume containing the current System folder.

station—Every computer on the network is a station. (See *workstation.*)

style—(MAC) Text formatting that governs the appearance of text. Style includes, but is not limited to, such things as boldface, italics, shadowed, and outlined.

subdirectory—(PC) Any directory that is not the root directory. Analogous to Macintosh folders.

SYLK—A file format developed by Microsoft for exchange of information between MultiPlan and other programs. SYLK files are special instances of ASCII or text files in which the text is highly structured to allow other programs to interpret that information.

system—

1. A computer.
2. The entirety of a computer installation, including the peripherals attached directly to it and perhaps all the elements of a network.

System file—(MAC) A special file containing the instructions that allow it to start up from a disk drive. The System file also contains such items as desk accessories, fonts, and other information used by programs and the computer while it is running.

System folder—(MAC) The specific folder that contains the active System file. The Finder will usually also be located in this folder, as will many other files the computer or network needs to run properly.

tab character—ASCII character code 9. It tells most word processors that the next character should be displayed at a predetermined horizontal position on the page or screen.

terminal emulation—The ability of a computer to act as if it is a terminal connected to a larger computer and to obey control signals sent to it by that larger computer.

text file—A file on disk that contains readable text. (See *ASCII file.*)

Toolbox—(MAC) The set of programming functions that are built into the ROM. These functions, available to all programs, are what give the Macintosh its appearance and character.

topology—The pattern of connections in a network.

TSR—(PC) Terminate and Stay Resident. A special type of DOS program that remains in memory so that it can be executed even while other programs are loaded.

type—

1. (PC) A DOS command that allows you to display the contents of a file on the monitor or a printer.
2. (MAC) A four-letter, all capitals part of the identifiers that the Macintosh associates with each file, showing what kind of file it is. ASCII files, for example, have a "type" on the Macintosh of TEXT.

unmount—(TOPS) The process of removing a volume or device from the list of volumes or devices a computer is now using is unmounting. This can be done through the use of the TOPS UNMOUNT command on the PC. On the Macintosh, it is done using the Unmount button in the TOPS Desk Accessory or by dragging the volume into the trash.

unpublish—(TOPS) See *publish*.

upload—To copy information from a local computer to a distant one.

user—One who is using a computer—a human being.

utility—A program that performs functions that might be different from creating, viewing, or printing documents. It is usually a small program that performs one or several specific purposes.

volume—(AS) A hard disk attached to a network.

word processor—An application that allows you to type, edit, and print text.

work group—A group of individuals who routinely work in consort with one another. A work group may consist of the individuals that make up one department (e.g., marketing, sales, or accounting). It may also consist of a subset of that department or a group within a department.

workstation—A computer that is attached to the network.

world—(AS) All the users on the network. (See *everyone*.)

write—To put information onto a disk.

zone—A group of two or more AppleTalk networks in a single large network of many interconnected AppleTalk networks. Every network is a zone. Zones do not become visible as zones until separate networks are joined with a bridge.

APPENDIX B

ASCII Character Sets

Table B.1 shows the ASCII character sets for two Macintosh fonts and for the standard IBM system.

Table B.1 ASCII Character Sets.

ASCII	Geneva	Helvetica	IBM Font	
0				
1		▯	☻	
2		▯	☺	
3		▯	♥	
4		▯	♦	
5		▯	♣	
6	\	\	\	
7	▢	▯	•	
8	▢	▯	▮	
9				tab
10			|	line feed
11				
12				
13				return
14	▢	▯	|	

Table B.1 (cont.)

ASCII	Geneva	Helvetica	IBM Font
15	□	▯	|
16	□	▯	▸
17	□	▯	◂
18	□	▯	|
19	□	▯	‼
20	□	▯	¶
21	□	▯	§
22	□	▯	▪
23	□	▯	⊥
24	□	▯	|
25	□	▯	|
26	□	▯	–
27	□	▯	–
28	□	▯	∟
29	□	▯	–
30	-	·	–
31			
32			space
33	!	!	!
34	"	"	"
35	#	#	#
36	$	$	$
37	%	%	%
38	&	&	&
39	'	'	'
40	(	(	(
41	)	)	)
42	*	*	*
43	+	+	+
44	,	,	,
45	-	-	-
46	.	.	.
47	/	/	/
48	0	0	0
49	1	1	1
50	2	2	2
51	3	3	3
52	4	4	4
53	5	5	5
54	6	6	6
55	7	7	7

Table B.1 (cont.)

ASCII	Geneva	Helvetica	IBM Font
56	8	8	8
57	9	9	9
58	:	:	:
59	;	;	;
60	<	<	<
61	=	=	=
62	>	>	>
63	?	?	?
64	@	@	@
65	A	A	A
66	B	B	B
67	C	C	C
68	D	D	D
69	E	E	E
70	F	F	F
71	G	G	G
72	H	H	H
73	I	I	I
74	J	J	J
75	K	K	K
76	L	L	L
77	M	M	M
78	N	N	N
79	O	O	O
80	P	P	P
81	Q	Q	Q
82	R	R	R
83	S	S	S
84	T	T	T
85	U	U	U
86	V	V	V
87	W	W	W
88	X	X	X
89	Y	Y	Y
90	Z	Z	Z
91	[	[	[
92	\	\	\
93	]	]	]
94	^	^	^
95	_	_	_
96	`	`	`

Table B.1 (cont.)

ASCII	Geneva	Helvetica	IBM Font
97	a	a	a
98	b	b	b
99	c	c	c
100	d	d	d
101	e	e	e
102	f	f	f
103	g	g	g
104	h	h	h
105	i	i	i
106	j	j	j
107	k	k	k
108	l	l	l
109	m	m	m
110	n	n	n
111	o	o	o
112	p	p	p
113	q	q	q
114	r	r	r
115	s	s	s
116	t	t	t
117	u	u	u
118	v	v	v
119	w	w	w
120	x	x	x
121	y	y	y
122	z	z	z
123	{	{	{
124	|	|	¦
125	}	}	}
126	~	~	~
127			
128	Ä	Ä	Ç
129	Å	Å	ü
130	Ç	Ç	é
131	É	É	â
132	Ñ	Ñ	ä
133	Ö	Ö	à
134	Ü	Ü	å
135	á	á	ç
136	à	à	ê
137	â	â	ë

Table B.1 (cont.)

ASCII	Geneva	Helvetica	IBM Font
138	ä	ä	è
139	ã	ã	ï
140	å	å	î
141	ç	ç	ì
142	é	é	Ä
143	è	è	Å
144	ê	ê	É
145	ë	ë	┃
146	í	í	┃
147	ì	ì	ô
148	î	î	ö
149	ï	ï	ò
150	ñ	ñ	û
151	ó	ó	ù
152	ò	ò	ÿ
153	ô	ô	Ö
154	ö	ö	Ü
155	õ	õ	¢
156	ú	ú	£
157	ù	ù	¥
158	û	û	┃
159	ü	ü	ƒ
160	†	†	á
161	°	°	í
162	¢	¢	ó
163	£	£	ú
164	§	§	ñ
165	•	•	Ñ
166	¶	¶	ª
167	ß	ß	º
168	®	®	¿
169	©	©	⌐
170	™	™	¬
171	´	´	½
172	¨	¨	¼
173	≠	≠	¡
174	Æ	Æ	«
175	Ø	Ø	»
176	∞	∞	░
177	±	±	▒
178	≤	≤	▓

Table B.1 (cont.)

ASCII	Geneva	Helvetica	IBM Font
179	≥	≥	│
180	¥	¥	┤
181	µ	µ	╡
182	∂	∂	╢
183	∑	∑	╖
184	∏	∏	╕
185	π	π	╣
186	∫	∫	║
187	ª	ª	╗
188	º	º	╝
189	Ω	Ω	╜
190	æ	æ	╛
191	ø	ø	┐
192	¿	¿	└
193	¡	¡	┴
194	¬	¬	┬
195	√	√	├
196	ƒ	ƒ	─
197	≈	≈	┼
198	∆	∆	╞
199	«	«	╟
200	»	»	╚
201	…	…	╔
202			╩
203	À	À	╦
204	Ã	Ã	╠
205	Õ	Õ	═
206	Œ	Œ	╬
207	œ	œ	╧
208	–	–	╨
209	—	—	╤
210	“	“	╥
211	”	”	╙
212	‘	‘	╘
213	’	’	╒
214	÷	÷	╓
215	◊	◊	╫
216	ÿ	ÿ	╪
217	[1]	Ÿ	┘
218	□	⁄	┌
219	□	¤	█

Table B.1 (cont.)

ASCII	Geneva	Helvetica	IBM Font
220	□	‹	▄
221	□	›	▌
222	□	ﬁ	▐
223	□	ﬂ	▀
224	□	‡	α
225	□	·	ß
226	□	‚	Γ
227	□	„	π
228	□	‰	Σ
229	□	Â	σ
230	□	Ê	µ
231	□	Á	τ
232	□	Ë	|
233	□	È	Θ
234	□	Í	Ω
235	□	Î	|
236	□	Ï	•
237	□	Ì	♦
238	□	Ó	•
239	□	Ô	∩
240	□		▪
241	□	Ò	±
242	□	Ú	≥
243	□	Û	ʼ
244	□	Ù	⌠
245	□	ı	⌡
246	□	ˆ	·
247	□	˜	≈
248	□	¯	°
249	□	˘	|
250	□	˙	•
251	□	˚	√
252	□	¸	ʼ
253	□	˝	ʼ
254	□		ı
255	□	ˇ	|

Note. Character 217 in Macintosh fonts is a special graphics character that is unique not just for each font, but generally for each size of a font. (This is generally true for Macintosh bitmap fonts but not for LaserWriter fonts.) You can produce these characters by holding down the option, command, and shift keys and pressing the tilde (~) key.

The Macintosh Fonts

The two Macintosh fonts shown in Table B.1 illustrate the variation that can exist between fonts on the Mac. Macintosh fonts all usually share the same character for ASCII codes through 216. Geneva is a standard Macintosh bitmap font. That is, it is designed for use on the Macintosh screen and for printing on the ImageWriter. It is a sanserif font, roughly corresponding to Helvetica on a LaserWriter. Helvetica is a font created by Adobe Systems and is built into every PostScript-compatible printer, including the LaserWriter and the LaserWriter Plus. In many fonts, characters that do not have a printing equivalent display a small box on the screen and the printer: Those characters are shown with this box where it appears on the Mac screen.

The Characters

On the Macintosh, you can use the key caps Desk Accessory to find which key combinations produce any of the symbols shown in this chart. On the PC, any of these symbols can be printed by holding down the ALT key, typing the number of that character on the numeric keypad, and then releasing the ALT key.

The characters for all the character sets are the same for characters from 30 through 127. These are the characters defined by the ASCII standard, and virtually all microcomputers have the same characters for these ASCII codes. The two machines diverge after this point. Although they do have some characters in common, they rarely have the same code. The standard translation programs will, in the case of MacLink Plus, handle this translation. The two Macintosh fonts diverge after ASCII character 216. Fonts created by Adobe Systems for PostScript will, of course, have the same characters for all the ASCII codes. However, when changing fonts from a PostScript font to one created by someone else, this can have unpredictable results.

APPENDIX

C

Working with PostScript Printers

One of the primary motivations for integrating a PC into an AppleTalk network is to make use of an Apple LaserWriter or compatible PostScript printer. This appendix examines several of the options available for making that connection. But first, a little background.

What Is PostScript?

PostScript is a page description language developed by Adobe Systems for use with laser printers or other high-resolution devices. A page description language is a means that computers (or users) can use to tell the printer what marks (letters, graphics, symbols, etc.) it should put in what locations on the page. PostScript was introduced with the Apple LaserWriter and AppleTalk in early 1985. Since that introduction, many other manufacturers have used it in their laser printers, and programs on the IBM PC have been modified to take advantage of it.

In the discussion that follows, I refer to the Apple Laser Writer frequently. Note, however, that there are a number of PostScript compatible printers on the market, from such companies as QMS, Texas Instruments, and AST. In virtually every case, my references to the Apple LaserWriter (or LaserWriter Plus) refer to these printers as well.

Essentially, PostScript is a computer programming language, expressly designed to define the appearance of pages. PostScript printers are more than just printers—they include their own computers that interpret the PostScript instructions. Often, these printer computers are more powerful than the computers that are sending along the instructions. They include a Motorola 68000 or 68020 microprocessor and 1–4 mb of RAM, and rumored models have as much as 16 mb of RAM!

Using a Macintosh program to print to a PostScript printer is no problem. Most PostScript printers have an AppleTalk connector built into them and can be plugged right into the network. On AppleTalk, they become a node on the network just as a computer does. From a Macintosh, printing to one of these printers is usually no more difficult than selecting the name of the printer to which you wish to print with the Chooser Desk Accessory and issuing the print command from the file menu. If the Laser Prep and LaserWriter printer drivers are available in your system folder, they do the rest of the work.

Two features of the Macintosh make this easy. First, the Macintosh is a graphics computer, with support for such things as proportional spacing and different font sizes built into it. Installing different fonts into the system so that they can be displayed on the screen is a relatively simple process.

Second, the Macintosh operating system offers a built-in printer driver that handles the communications between the application program and the printer. This means that not every software developer has to write unique printer drivers to communicate with all the different kinds of printers. (One negative consequence of this is that you can only use printers on the Macintosh for which there are good, standard, reliable printer drivers available.)

The PC world is very different from the Macintosh world. The PC operating system (at least before the release and general acceptance of the OS/2 operating system and its presentation manager) does not offer programs any standard way of dealing with printers. Each software manufacturer must include with their programs special printer drivers for communicating with the myriad of different printers available. WordPerfect, for example, includes drivers for well over 100 printers with their software. Second, as we discussed in chapter 1, the IBM PC and its compatibles is a character-based computer.

What this means is that the appearance of characters on the screen is controlled by the image of those characters that is built in to the computer's ROM. Only one size of each character is in ROM, and when working with most programs, only those characters can be displayed on the screen. That makes the what-you-see-is-what-you-get (or WYSWIG, pronounced "wizzy-wig") screen display something of a rarity in the PC world. (Again, as the OS/2 operating system and the presentation manager become more accepted and used in the PC world this will change.)

What this means is that it has been difficult for most manufacturers of PC software to integrate complete PostScript functionality into their programs, at least as far as the WYSWIG concept is concerned. Slowly, PC software manufacturers are integrating PostScript support into their programs, with varying degrees of success.

In the meantime, users of PCs on AppleTalk networks still want to take advantage of the capabilities of PostScript laser printers. Both TOPS and Apple have created programs that allow users to access laser printers of the AppleTalk network and to communicate with these printers in PostScript.

TOPS NetPrint

TOPS NetPrint is a program that performs several useful functions:

1. It handles "redirection" so that programs may print using an AppleTalk interface board. We discussed redirection in chapter 2. Essentially, it means that the AppleTalk interface board is made to appear to be a standard printer interface board—either serial or parallel. This is the mode in which NetPrint will most often be used, as long as the application program that is printing can produce PostScript output.
2. NetPrint can translate codes intended for the two leading printer standards to PostScript. The two leading printer standards are the Epson FX-80 (and other Epson printers as well as many others) or the IBM Proprinter. If your application cannot generate PostScript compatible output, it can almost certainly generate output compatible with one of these printers. NetPrint will interpret text and formatting codes that the application thinks it is sending to one of these printers, convert it into PostScript, and

send it along to the LaserWriter. Additionally, NetPrint can recognize special codes you insert into the document and allow you to do some things that your application program might not otherwise permit(e.g., change the font and its size).

3. NetPrint includes a print spooler. As discussed in Chapter 2, a print spooler is software that allows the computer to run applications at the same time it is sending output to the printer.

Now let's discuss implementing TOPS NetPrint and making it work with your applications.

Installing and Configuring NetPrint

Installing NetPrint is straightforward and well documented in the user manual. If you are installing on a hard disk, make sure to tell the installation program to put NetPrint into the same directory that TOPS itself is in. This makes managing the software much easier. As with TOPS itself, make sure that this directory is listed in the path statement in your AUTOEXEC.BAT file on the hard disk. This means that you do not have to be logged into that directory for NetPrint commands to work.

You can install NetPrint onto a floppy disk. Be aware, however, that, for the spooler to work, it must store output destined for the printer on a disk while it is printing it. Frequently, this can require a large amount of disk space— especially if you are printing graphics to a PostScript printer. You can tell NetPrint where to store these files (we'll discuss this shortly) and you may want to direct NetPrint to store these temporary files on a networked volume that you have mounted.

NetPrint can work in a number of different ways—either in emulation mode or not, or in a spooled mode or not. To tell NetPrint how it should work, use the CONFIGURE program that is included on the disk.

CONFIGURE can be used in two modes. The default mode, the one that is invoked if you type "CONFIGURE" by itself at the DOS prompt, is a menu-driven interface that allows you to set each of the options one at a time. This mode is well documented in the NetPrint manual, but there are a couple of things that should be discussed.

NetPrint Port The opening screen of the CONFIGURE program, asks you to tell it the name of the DOS device that NetPrint "intercepts." This is the redirection part of NetPrint. As discussed in Chapter 2, the AppleTalk interface board does not fall into any one of the standard printer interfaces recognized by DOS (i.e., one of the LPT, or parallel ports, or one of the COM, or serial ports). NetPrint, then, allows you to tell your application program to send output to one of these ports—most commonly LPT. NetPrint then resides in memory, watching for output to be directed to one of these ports. When it detects this output, it catches it and sends it out to the printer via the AppleTalk board installed in the system.

If you have a local printer attached to your system, it will most often be attached to the first parallel port—LPT1. In this case, you will want to tell NetPrint to intercept output sent to LPT2, which will still allow you to use your local printer. In this case, all you need to do to print to a different printer is instruct your program to which port it should send its output, and NetPrint will take care of the rest.

Remember, however, that your program should be using the correct "printer driver" for the printer on which you are printing. If you have an Epson printer attached to LPT1, it's not enough to tell your software to send output to that device. You must also make sure it sends output formatted for the Epson printer. If your software supports PostScript, it's not enough to tell it to send the output to LPT2. You must also make sure it is correctly sending PostScript code to that printer, by specifying the correct printer driver. Methods of giving the correct instructions to your software depend on your software and can vary greatly from program to program.

Device The CONFIGURE program has a separate screen that lists the devices to which you want to print. This lets NetPrint know which, if any, translations it should perform. The default device, and the one most commonly used, is the LaserWriter (remember that this includes most other PostScript printers as well). However, Apple's ImageWriter printer is also capable of being installed as a separate node on the network, and NetPrint can redirect output to a networked ImageWriter. (Your software must support the ImageWriter directly; it cannot translate Epson or Proprinter output to an ImageWriter-compatible form.)

NetPrint can also be used to "spool" output to a device that is connected locally. Use the local printer option on the device menu if that's what you want to do.

Note When you choose to print to a LaserWriter or networked ImageWriter, CONFIGURE asks for the name of the printer. Since these printers are their own nodes on the network (as opposed to printers published by other TOPS stations), this works in a very different manner from the mount printer mechanism discussed in Chapter 2. The default option presented here is the " = " option. If you leave this option, NetPrint will send its output to the first available printer it finds in your zone. If there's only one printer in your zone (or on the network, if it doesn't have zones), then you can leave this setting as it is.

If you have more than one printer on the network, though, you might want to specify the name of the printer you will be using. To do this, hit the right arrow key, and CONFIGURE will show you the list of available printers. Use the space bar to move the highlight cursor to the printer you wish to use, and strike the enter key to select that printer. CONFIGURE will remember this and use that printer as the default. (Don't worry if you want to print to a different printer later—you can change the name easily.)

Translation As already discussed, NetPrint can translate printer codes and instructions from an Epson or IBM printer to PostScript. Use CONFIGURE's translation menu to set this up.

This menu first asks is if you are using the IBM extended font. This font takes advantage of one of the nicest features of PostScript—its ability to have new font definitions downloaded from the computer into the printer's memory. As discussed earlier in the book and as is illustrated in appendix B, the IBM screen font is very different from Macintosh or standard PostScript fonts. Standard Mac or PostScript fonts do not include, for example, the line drawing characters that are used on the PC. This font is very useful for PC screen dumps that you want to look exactly like the PC screen. If you enable this option and download the font, NetPrint will automatically use this font for any screen dumps you generate with the PC's shift-PrtSc key combination. For this to work, NetPrint must be set up to intercept print commands sent to LPT1, since that is where the PC printscreen command sends the screen dump.

Note If you do not have PostScript translation enabled or have not downloaded the PCScreen font, the printscreen function will not work.

To enable PostScript translation, simply select this option. The program will show you another translation screen that allows you to specify such things as the default font, font size, size of the paper tray, and number of copies to be printed.

Remember that CONFIGURE allows you to set the defaults you will be using to print. If you are working with only a couple of programs and will be using the same NetPrint mode for all the programs, you won't have to change these options later. Otherwise, you might find that you need to change them later. In this case, establish the defaults that you will need to use most often.

Changing NetPrint Options with Batch Files

In Chapter 2, we showed how batch files can be used to automate many standard TOPS functions. The same is true with NetPrint. The batch file techniques that follow can save you a lot of time when you are printing. For a discussion of creating batch files and for some specific techniques, see Chapter 2 as well as a good DOS Owner's Manual. The techniques presented below are general and do not necessarily discuss all the options available. Essentially, these options work by including specific parameters on the DOS command line with the CONFIGURE statement.

Changing PostScript Emulation

To turn PostScript emulation on from the command line or from within a batch file, use the /W parameter, followed by a (+):

```
CONFIGURE /W+
```

Until CONFIGURE is told otherwise, emulation will be turned on. This option will only work if the emulation software has been loaded into memory first. To do this, you will need to execute one of the two emulation programs (PROPS.EXE for ProPrinter emulation, FXPS.EXE for Epson FX-80 emulation).

To turn off PostScript emulation, use this command:

```
CONFIGURE /W-
```

where the minus (–) means turn it off. Remember that the emulation software will still be resident in RAM and take up about 32 kb. Unfortunately, this software cannot be removed from RAM.

Changing the Name of the Printer You Are Using

In the main CONFIGURE program, you set the default printer you will be using. However, that printer might not always be available. To set the name of a different LaserWriter to use, use this command:

```
CONFIGURE /L "L=printername"
```

where you insert the name of the printer you are using instead of "printername." If you don't know the name of the printer, you will need to use CONFIGURE in its menu-driven mode to find the name of the printer. If you wish to print to any local printer, just use "L = ".

Changing the Intercepted Printer Port

You might sometimes need to change the identity of the logical printer device that NetPrint is intercepting. For example, you may have a default of LPT2 but have a program that can only print to LPT1. To make this change, use this command:

```
CONFIGURE /L LPT1
```

Doing It All at Once

Remember that different parameters can be grouped together on the same command line, to make execution faster. To put all the commands already discussed into effect at once, the line would read:

```
CONFIGURE /L LPT1 "L=ACCOUNTING PRINTER" /W-
```

This command tells NetPrint to intercept all output being directed to LPT1 and send it to the LaserWriter named "ACCOUNTING PRINTER." It further turns PostScript emulation off.

Working with Programs That Do Not Support PostScript

As we've discussed, NetPrint offers emulation for programs that don't directly support PostScript. It can intercept output for an Epson or

IBM printer, translate it into PostScript, and send it to a LaserWriter. It further allows you to embed codes into your file, allowing you to select certain PostScript functions.

Loading Emulation Before emulation can work, you must install one of the interpreters. Decide which you will be using—FXPS.EXE for Epson emulation or PROPS.EXE for ProPrinter emulation. Install these commands into the batch file you use to start NetPrint.

Next you will need to make sure that NetPrint is configured to work in emulation mode. Run CONFIGURE in either its menu-driven mode, or turn on emulation with the /W + " parameter on the DOS command line.

You may want to set this up so that it is invoked automatically whenever you run a certain program that does not support PostScript. For example, WordStar 3.3 does not support PostScript, so you might include in your WordStar batch file the line "CONFIGURE /W + " before WordStar itself is invoked. You will also need to make sure that WordStar is installed to print to either the Epson or IBM printer.

Once you are working with NetPrint in its emulation mode, you can do a couple of things. In its default state, NetPrint will use the IBM screen font that is included with NetPrint when printing in emulation mode. If you—or others on the network— have downloaded that font to the printer, it will be used automatically. Otherwise, the Courier font will be used.

Using the IBM screen font has a couple of advantages. First, it includes all the PC graphics characters and other symbols that are not otherwise available in PostScript fonts. Appendix B illustrates the differences between the PC font and PostScript and Macintosh fonts.

Second, the IBM screen font, like Courier, is a monospaced font. This means that all characters on a line are given the same amount of space on the line. Since most programs that don't support PostScript also do not support proportional spacing very well, this assures that your printed output will look better.

If you do wish to use other fonts, NetPrint allows you to insert codes to change the formatting of the document. Suppose you want to use the Times font, in 12 points. To do this, you would type the following into your text:

```
\\\T 12 \\\
```

The first "\\\" characters tell NetPrint that the following characters are text it should interpret instead of print. The "t" tells it to use the Times font, and the "12" specifies 12 point type. The final "\\\" tells NetPrint that it should, again, print characters. If you wished to use bold or italic type, you could insert a "b" or "i" before the final "\\\".

You can insert several of these NetPrint codes into a document, but there are a few things you should be careful of. Chief among these is the fact that when you are doing this, you are working behind the back of your program. Suppose you are working in WordStar and change the typeface and the size. Since you are not using standard WordStar commands to do this, WordStar is not aware of any of these changes and will attempt to use its standard methods of placing the characters on the line. If you change to a larger or smaller typeface, it will try to put the number of characters on a line that are dictated to it by its margins. So if you are making text larger or smaller, you will need to adjust WordStar's margins (with the ^OL command for the left margin and the ^OR command for the right margin) to instruct it to put more or fewer characters on the line. The same holds true for the height of the line—you'll need to change the line height with WordStar's .LH command.

Even with the margins adjusted correctly, WordStar will not space the characters correctly on the page. Since it cannot be aware that Times is a proportional spaced font, it will place the characters in a mono-spaced mode, causing each character to be given the same width on the page. At best, character spacing will appear to be a little strange.

In summary, while NetPrint's ability to emulate a dot matrix printer is useful, it is no replacement for software that supports PostScript directly. You should either upgrade to a later version of the program that supports PostScript or use a different program. Desktop publishing programs, for example, can accept files created by a wide variety of other programs and support PostScript well.

Working with Programs That Support PostScript

It's a lot easier, of course, to work with programs that do support PostScript. These programs will generally include their own mechanisms for specifying such things as the font, size, and style of text. Desktop publishing programs also include means for mixing images in with the text.

You need to make sure of only two things when working with PostScript-compatible programs. First, any NetPrint emulation must be turned off. NetPrint must pass the PostScript code along to the LaserWriter just as it receives it. Otherwise, instead of getting a printout of your document, you will probably get a printout of the PostScript instructions that define the page. If you are working only with PostScript-compatible programs, use the CONFIGURE program to set the default to no emulation. Otherwise, use the CONFIGURE /W – command in any batch file that loads a PostScript-compatible application.

Second, you must make sure that your software and NetPrint agree as to which logical DOS printer address NetPrint is intercepting to send code to the LaserWriter. If the default set with the NetPrint CONFIGURE program is LPT1, then make sure your program is sending output to that printer port. You must, of course, also make sure that your application is using a PostScript-compatible printer driver. NetPrint takes care of the rest.

Using the AppleShare PC Printer Driver

Along with AppleShare PC, Apple includes a printer driver for use from PCs connected to the network. This printer driver offers several of the same functions of NetPrint, including printer redirection and emulation. It is, however, somewhat better integrated into the AppleShare PC environment than is TOPS NetPrint.

Choosing a printer, like choosing a server, is done from within the AppleShare PC DA program. LaserWriters and ImageWriters appear in the "Select a Type" pane on the Chooser window. Select the LaserWriter printer type, and the other panes—"Select a LaserWriter" and "Select a Zone"— will show the available options. Point to the printer to which you wish to connect, using the arrow keys. After selecting a printer and striking Enter, you will be shown a "Connect a LaserWriter" window that will allow you to specify certain options.

As with TOPS NetPrint, these options include which logical printer device (i.e. LPT1, LPT2, or LPT3) will be intercepted and sent to the LaserWriter. You will need to make sure that your program is directing its output to that same logical device.

If you always want to be using the same LaserWriter, you can tell the DA to always connect to that printer at startup. It will include the necessary commands in the ANET script that connects you with AppleShare.

Working with Programs That Do Not Support PostScript

As already mentioned, the AppleShare printer driver can translate instructions designed for the Epson series of printers to PostScript. Since the PC screen supports some graphics characters that are not available in the standard PostScript printer font (Courier) that is used for this emulation mode, you need to watch out for a few things.

Graphics Characters The PC screen font includes several graphics characters that are not available in PostScript (see appendix B). If you have any of these graphics characters in your text, they will not be printed. The printer driver will attempt to create them—usually by combining two of the standard Courier characters. If it cannot do this, it will either substitute a different character or print a blank space. Consult the manual for the printer driver for a table of these symbols.

Downloadable Fonts Many Epson printers (such as the LQ2500, which is supported in the emulation mode) support downloadable fonts. These are font definitions that can be placed into the memory of the Epson printers. They are not compatible with, nor are they to be confused with, PostScript downloadable fonts. Make sure that your program is not attempting to do this. You will, at best, get unpredictable results.

Working with Applications that Support PostScript

In this case, all you need to do is make sure that your application program is using the correct PostScript printer driver and is sending its output to the correct logical printer port that is being intercepted by the printer driver. Nothing else really needs to be done.

Using Batch Files to Connect to the Printer

In chapter 3, we discussed batch file techniques for working with AppleShare PC to connect to a server. Connecting to a printer works much the same way, and is an extension to those techniques. Use the

ANET CONNECT command to connect with a printer. This command is adequately documented in the Apple documentation. You can only change the emulation mode when you are connecting to the printer. That is, once you've connected in emulation mode, you cannot change the mode of connection to nonemulation mode without first disconnecting from the printer.

If you have more than one printer available, you can connect to one in emulation mode and to another in nonemulation mode. Then you can set yourself up so that printing to LPT1 prints in PostScript, while printing to LPT2 translates output from Epson codes into PostScript.

Summary of PostScript Printing

Both NetPrint and the AppleShare PC printer driver perform the bare minimum for printing to a networked LaserWriter: They can make the printer appear to be connected locally to the PC (printer redirection), and they can translate some output designed for a standard DOS printer into PostScript. The latter feature is really only moderately useful, given the inherent problems involved. What neither of these programs can do, what is really impossible for them to do in the current DOS world, is provide a universal driver for PostScript. Only with the possible acceptance of OS/2 will that situation change, as standard drivers for printers and other devices become more commonplace.

Index

E

F

N

O

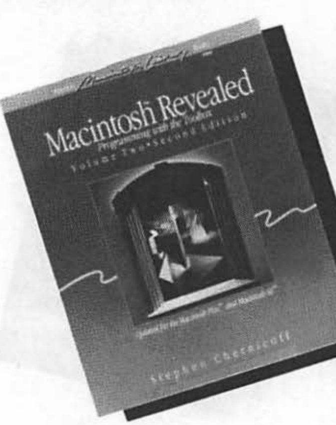
U

V

W

X

Z



Macintosh™ Revealed, Volume One: Unlocking the Toolbox, Second Edition

Stephen Chernicoff

Macintosh Revealed, Volume One is the most successful Macintosh technical programming book commercially published. Now, in addition to covering the 128K and 512K Macintosh, the second edition concentrates on the Macintosh Plus and its expanded memory management, I/O routines, graphics and text handling, and serial communications.

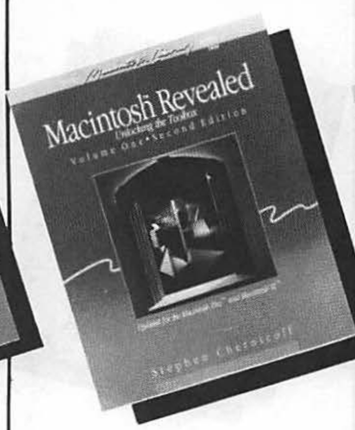
This edition has been completely updated with all the latest Mac features including the new keyboard layouts, character codes for expanded LaserWriter,™ font families and access, ROM resources, and graphic resources. Together with Volume Two, the nearly 500 ROM routines that make up the Macintosh Toolbox are described.

For serious programmers and software developers, this easily referenced book gives and in-depth look at the Macintosh's graphical interface.

Topics covered include:

- All the Tools You Need
- Putting the Tools to Work
- Thanks for the Memory
- Any Port in a Storm
- Quick on the Draw
- Summoning Your Resources
- Getting Loaded
- Upstanding Characters
- Appendices Include: Toolbox Summary, Resource Formats, Macintosh Memory Layouts, Key Codes and Character Codes, Error Codes, Summaries of Assembly Language Variables, and Trap Macros and Trap Words.

590 Pages, 7¾ x 9¼, Softbound
ISBN: 0-672-48400-5
No. 48400, $24.95



Macintosh™ Revealed, Volume Two: Programming with the Toolbox, Second Edition

Stephen Chernicoff

Providing an advanced look at the higher-level parts of the Toolbox that implemented the revolutionary Macintosh user interface, this second edition focuses on the Macintosh Plus as well as the 128K and 512K Macintosh.

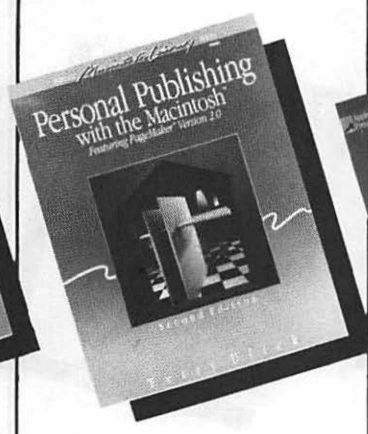
This new edition of Volume Two includes HFS, 800K double-sided disks, the new Standard File and Disk Initialization Package, window zooming, zoom boxes, scrollable menus, function key routines, and automatic text scrolling.

Software featured in the book is available on disk from the author.

Topics covered include:

- All the Tools You Need
- Keeping Up with Events
- Windows on the World
- What's on the Menu?
- Scissors and Paste
- At the Controls
- Meaningful Dialogs
- Files at Your Fingertips
- Appendices: Toolbox Summary, Resource Formats, Memory Layouts, Key Codes and Character Codes, Error Codes, Assembly-Language Variables, Trap Macros and Trap Words, and MiniEdit Source Listing

696 Pages, 7¾ x 9¼, Softbound
ISBN: 0-672-48401-3
No. 48401, $24.95



Personal Publishing with the Macintosh,™ Second Edition Featuring PageMaker® 2.0

Terry M. Ulick

Owners of PageMaker desktop publishing software or those interested in learning about desktop publishing will welcome this major revision of the dominant book on PageMaker. It now details how to take advantage of the 2.0 version.

Topics covered include:

- Assembling a Personal Publishing System
- Selecting the Right Hardware and Software
- The Printed Page
- More Than One Page
- Pages on the Macintosh
- Electronic Page Assembly
- The Grid
- Working with Type
- LaserWriter Type Styles
- Formatting Type
- Working with PageMaker
- Building Master Pages
- Placing Elements on a Page
- Adding Graphic Elements
- Linking PageMaker Files
- PageMaker Unleashed
- Printing Page Files
- High-Volume Printing
- Multicolored Pages
- Grids and Page Samples

352 Pages, 7¾ x 9¼, Softbound
ISBN: 0-672-48406-4
No. 48406, $19.95

How to Write Macintosh™ Software

Scott Knaster

Here are the insights and tools that enable you to avoid common pitfalls that have hampered Macintosh programmers.

This book thoroughly discusses memory allocation, reallocation, purging, and implicit and explicit dereferencing, as well as how to reduce memory fragmentation and segment an application.

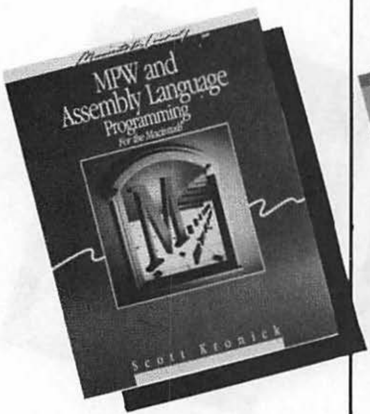
It tells how to use two powerful object-code debuggers, and how to make your software more effective. Included are valuable appendices: "Assembly Language Overview," and a list of common programming problems.

Topics covered include:

- Getting Started
- Adding Features
- Writing a Program
- Loops
- Packages in BASIC: Functions, Subroutines, and Subprograms
- Picture Windows: Using Macintosh Features

528 Pages, 7¾ x 9¼, Softbound
ISBN: 0-8104-6564-7
No. 046564, $27.95

Visit your local book retailer, use the order form provided, or call 800-428-SAMS.



MPW and Assembly Language Programming for the Macintosh™

Scott Kronick

This introduction to MPW for programmers is the first to teach Macintosh assembly language! Macintosh Programmer's Workshop (MPW) is the new programming development system for the Macintosh and one of the most sophisticated microcomputer programming development systems in existence.

This book is a clear and carefully written introduction to MPW, assembly language, Pascal, and C. It will help beginning and intermediate programmers write assembly language, Pascal, and C programs using the Macintosh Toolbox, including mouse events, windows, QuickDraw, and menus.

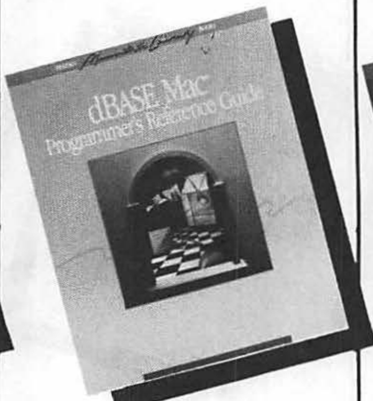
Topics covered include:

The Macintosh Programmer's Workshop
- A Sample Program in Assembly Language
- Fundamental File Commands
- StartUp and Files
- Command Language
- Make and Structured Commands

The Assembly Tutorial
- Slots
- First Lines of Assembly Code
- The ABC's of Blocks of Code
- QuickDraw Inside the Window
- Structured Programming with Blocks
- The Keyboard
- Menus

The MPW and Assembly Dictionaries
- The MPW Shell Command Language
- The 68000 Instruction Set with Directives and Toolbox Traps

352 Pages, 7¾ x 9¼, Softbound
ISBN: 0-672-48409-9
No. 48409, $24.95



dBASE Mac® Programmer's Reference Guide

Edward C. Jones

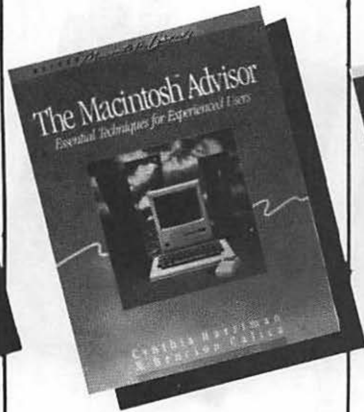
Here is a detailed guide for using and programming in dBASE for the Macintosh, written especially for those who need to take full advantage of the program's available power.

This book highlights programming application using modular code to develop complete applications for any purpose, and provides a detailed explanation of how to use the relational powers of the software.

Current DOS dBASE users will learn how to take advantage of the unique Macintosh graphical interface when they program or use dBASE Mac.

Topics covered include:
- Introduction
- Creating and Modifying Files
- Working with Display Views
- Managing Data
- Relating Files
- Creating Procedures
- System Design
- Program Flow in a dBASE Mac Procedure
- Adding and Modifying Records Under Program Control
- User Output
- Debugging Techniques
- Sharing Files with dBASE III and dBASE III Plus
- Programming Tips for dBASE III Programmers
- Complete Applications
- Appendix: dBASE Mac Commands

300 Pages, 7¾ x 9¾, Softbound
ISBN: 0-672-48416-1
No. 48416, $19.95



The Macintosh™ Advisor

Harriman and Calica

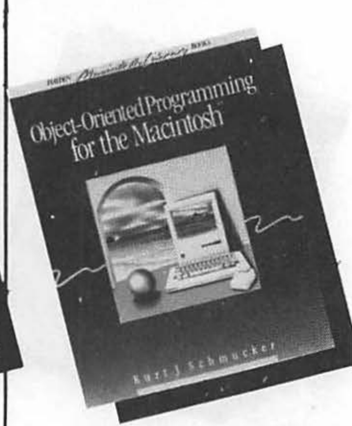
This book provides advice on shortcutting some of the Macintosh's elementary procedures to use it more productively.

Increase performance by gaining a better understanding of the Finder, RAM disks, memory management, and other features. Examine the powerful hardware options such as hard disks, RAM upgrades, and the LaserWriter.

Includes troubleshooting procedures for personal computing problems.

Topics covered include:
- The Finder: Macintosh's Operating System
- Speeding Applications
- Desk Accessories and FKeys
- Customizing Applications
- Disk Management Utilities
- RAM Upgrades and Hard Disk Drives
- Options for Better Input and Output
- IBM to Mac: Transferring Data
- Troubleshooting
- Appendices: Shopping Lists, Sources, Technical References

320 Pages, 7¾ x 9¼, Softbound
ISBN: 0-8104-6569-8
No. 046569, $18.95



Object-Oriented Programming for the Macintosh™

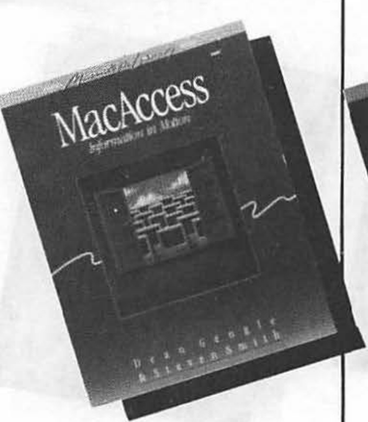
Kurt J. Schmucker

With this book, gain insight into the fundamental object-oriented concepts of objects, classes, instances, message passing and method calls, and into advanced topics like meta-classes and multiple inheritance.

Topics covered include:
- Why Object-Oriented Programming?
- The Basics of Object-Oriented Programming
- Object Pascal
- Introduction to MacApp
- Mini-QuadWorld—A Small MacApp Application
- The Most Frequently Asked Questions about MacApp
- The Flow of Events in MacApp
- QuadWorld—A Full MacApp Application
- Advanced MacApp Features
- Advanced Concepts in Object-Oriented Programming
- Smalltalk
- Lisa Clascal and the Lisa Toolbox
- An Overview of Other Object-Oriented Languages on Macintosh
- Appendices Include: The Macintosh User Interface Standard, and Object-Oriented Programming Languages, Suppliers and User Groups

624 Pages, 7¾ x 9¼, Softbound
ISBN: 0-8104-6565-5
No. 46565, $34.95

Visit your local book retailer, use the order form provided, or call 800-428-SAMS.



MacAccess: Information in Motion

Dean Gengle and Steven Smith

This book examines the software and hardware required for successful data transfer and offers a step-by-step discussion of a sample telecommunications session, clearly explaining how to send and receive text files, spreadsheet files, and other forms of data files. During detailed presentation of the telecommunications session, the focus is set on solutions to common communications problems.

It includes in-depth discussions on connecting the Macintosh™ to other computers, sharing and transferring data between machines, protocols, cabling, and conversion procedures.

With this book, you can easily and efficiently share data between your Macintosh and other computers and keep your vital information flowing to where it will do you the most good.

Topics covered include:

- Information in Motion
- Executives Backgrounder
- Telecommunications
- A SofTour of MicroPhone
- Communication Command Languages
- Links and Hints
- Advanced Topics by Section
- Appendices Include: Mac ASCII Chart, File to File Import/Export Charts, Sources Directory, Families, Bibliography for Further Reference, and Feedback and CTG Newsletter Sheet

00 pages, 7¾ x 9¼, Softbound
ISBN: 0-672-46567-1
No. 046567, $21.95

Macintosh™ Multiplan®

Lasselle and Ramsay

This invaluable resource book introduces the basics of Multiplan and then builds up to more advanced topics, quickly, easily, and logically.

Here is all the information needed to create, use, and revise Multiplan to your specifications. The clear step-by-step instructions simplify critical functions.

Topics covered include:

- Using Your Macintosh
- Entering Information in a Worksheet
- Revising a Worksheet
- Absolute and Relative Cell References
- Inserting Rows and Columns
- Changing Columns
- Templates

224 Pages, 7¾ x 9¼, Softbound
ISBN: 0-8104-6555-8
No. 46555, $16.95

Advanced Macintosh™ Pascal

Paul Goodman

Beginning where most Pascal books leave off, this book introduces programmers to the advanced skills needed to produce "real" applications.

Not only does this book discuss Pascal's "power" structures, thoroughly examining sets, pointers, and records, it also covers basic and advanced QuickDraw routines, data types, and programming techniques. In addition, it shows programmers how to access and use the Macintosh Toolbox.

With this book, you can create graphics, program dialog boxes, pull down menus, and radio buttons. The Standard Apple Numeric Environment is also presented to cover the extended data types needed for scientific and business programs.

Topics covered include:

- Macintosh System and Memory Overview
- Advanced Pascal Structures
- Files and File Programming
- ISAM
- Event Programming
- QuickDraw Techniques
- The Inline Routines—Accessing the Toolbox
- Advanced QuickDraw
- A Complete Application—The Logger
- The Standard Apple Numerical Environment
- Appendices Include: Macintosh Pascal Version 2.0, Inline Routine Trap Addresses, and Decimal to Hexadecimal Table

320 Pages, 7¾ x 9¼ , Softbound
ISBN: 0-672-46570-1
No. 046570, $19.95

Basic Microsoft® BASIC for the Macintosh™

Coan and Coan

Develop a solid foundation in programming of Microsoft BASIC with step-by-step instructions, sample programs, and screen illustrations that guide you in writing control routines, subroutines, and complete programs quickly and easily.

Learn how to access Macintosh QuickDraw ROM, highlight short and simple programs with additional capabilities, save memory with options base, initialize options, and much more.

Topics covered include:

- Getting Started
- Adding Features
- Writing Programs
- Loops
- Packages in BASIC: Functions, Subroutines, and Subprograms
- Picture Windows: Using Macintosh Features
- Pigeonholes Galore (Arrays)
- Miscellaneous Applications
- Graphics
- Using Macintosh QuickDraw Graphics Routines
- Files
- Random-Access Files
- Random-Access Address List
- Appendices: The Microsoft BASIC Menu Bar, Using the Disk, ASCII and Special Character Chart, Microsoft BASIC Functions, Listing of Programs, Solution Programs for Even-Numbered Problems

448 Pages, 7¾ x 9¼, Softbound
ISBN: 0-8104-6558-2
No. 46558, $19.95

Visit your local book retailer, use the order form provided, or call 800-428-SAMS.

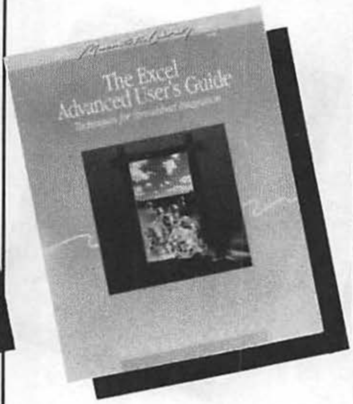
Programming the 68000
Rosenzweig and Harrison

This prime resource for programmers fully examines the power of the Motorola 68000 microprocessor. It details the assembly language processes of coding, editing, compiling, linking, and resource compiling, with thorough explanations of the 68000 instruction set and addressing modes.

Topics covered include:

- Introduction to Assembly Language
- The Addressing Modes of the 68000
- The 68000 Instruction Set
- Sample Programs
- A Programmer's Overview of the 68000 Hardware
- Macintosh Tools
- The Macintosh ROM Calls
- SimpleCalc—A Sample Application
- Some Advanced Subroutines Not in SimpleCalc
- Appendices Include: The Binary and Hexadecimal Numbering Systems, Instruction Format & Cycle Timing, Condition Codes, Error Messages, Using the Lisa Workshop, Samples of Trap Calls into the ROM, SimpleCalc Program Code

416 Pages, 7¾ x 9¼, Softbound
ISBN: 0-8104-6310-5
No. 46310, $24.95



Excel™ Advanced User's Guide
Richard Loggins

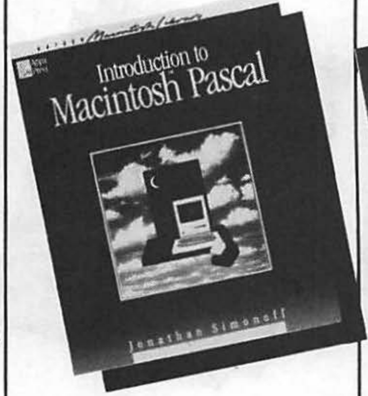
This guide provides the means to produce integrated systems of spreadsheet and supporting data that greatly increase productivity and efficiency.

With this book you can master the advanced features of Excel for designing, creating, and implementing integrated business applications. Specifically, you can design fully integrated applications; implement more than one database area on a single worksheet; establish multiple active criteria area for each database; use command macros to construct worksheets; use macros for data entry and verification; transform simple worksheets into functional menus; and replace repetitive procedures with a single subroutine.

Topics covered include:

- Advanced Basics
- Fixed Tables
- Database and Criteria
- Command Macros
- CR Macro
- Summary Report
- Forms Design

256 Pages, 7¾ x 9¼, Softbound
ISBN: 0-672-46626-0
No. 046626, $19.95



Introduction to Macintosh™ Pascal
Jonathan Simonoff

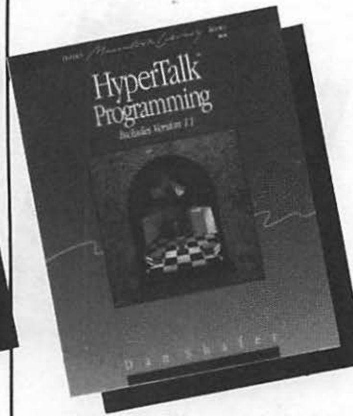
Learn to create Macintosh Pascal programs easily and quickly.

Written by an Apple documentation writer and Pascal expert, this book emphasizes the system's capabilities beginning with program design concepts and proceeding through graphics, text editing and files, handling sound, and using math operators and functions.

Topics covered include:

- Can I Be an Artist if I Can't Draw a Line?
- Lining Up
- Rectangles and Ovals
- What Type Through Yonder Window Breaks
- Fielding
- Files: A Piece of Cake
- Sound Synthesis
- Printing
- Advanced QuickDraw
- Macintosh Math
- Debugging
- Appendices: Everything about Semicolons; Help, Please, My Mac is Burning

368 Pages, 7¾ x 9¼, Softbound
ISBN: 0-8104-6562-0
No. 46562, $19.95



HyperTalk™ Programming
Dan Shafer

This exciting new book is a comprehensive tutorial covering every feature of HyperTalk including the concepts behind the language, object-oriented programming, and sample scripts for ready-made applications. It is an essential guide for all HyperCard users and programmers.

The book thoroughly explains the powerful programming aids used for the design and development of HyperCard applications, including the use of sound, graphics, and communications. It also presents language extensions and how to use them to extend HyperTalk with C or Pascal programming.

Topics covered include:

- How HyperTalk Came to Be
- Object-Oriented Programming Concepts
- HyperCard Refresher
- HyperTalk Basics and System Messages
- Control Structures
- Text & Data Management
- Sound and Music Basics
- Math Operators and Functions
- Commands
- Tips, Traps, and Techniques
- Tools and Programming Aids

450 Pages, 7¾ x 9¼, Softbound
ISBN: 0-672-48426-9
No. 48426, $24.95

Visit your local book retailer, use the order form provided, or call 800-428-SAMS.

IBM® PC and Macintosh® Networking

Featuring TOPS™ and AppleShare™

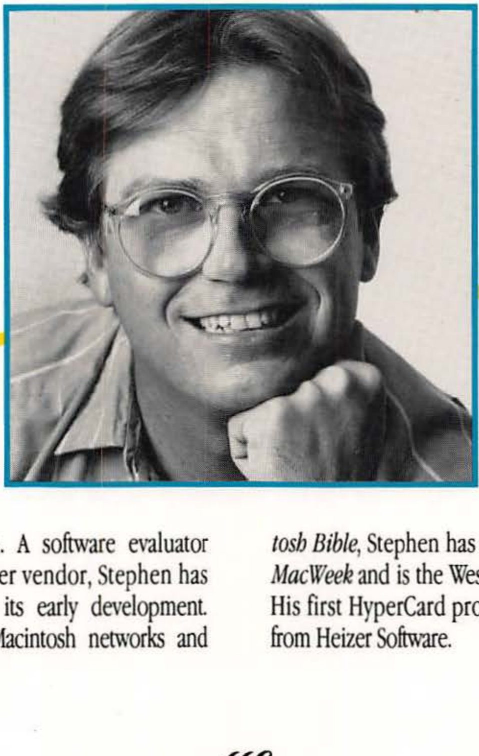
The IBM PC and the Apple® Macintosh are the world's most popular personal computers, each offering unique workstyles and advantages. It was inevitable that businesses would want to connect the two in a network to share files, storage, and printers.

IBM PC and Macintosh Networking tells you how to do just that — and do it painlessly. The author discusses two software/hardware product combinations that allow the PC and the Macintosh to be networked. These two products — TOPS and AppleShare with AppleShare PC — provide two different approaches to the problem.

This book helps you select the PC/Macintosh network that's right for you and lets you implement that network to its greatest advantage.

IBM PC and Macintosh Networking not only tells *how* the IBM and the Macintosh are different — it also tells *why* they are different. In these pages, you'll learn the capabilities of both TOPS and AppleShare software; details of the transfer of spreadsheet, word processing, and database files from one format to another; how to create the most efficient files and share printers and external disk drives. The author even explains the use of TOPS or AppleShare on a Macintosh-only LAN or PC-only LAN.

Make the most of your PC and Macintosh power. Read this book *before* you invest in a networking system.



About the Author

Stephen L. Michel has worked in the microcomputer industry since 1983 and has owned a Macintosh since its introduction. A software evaluator and manager for a major computer vendor, Stephen has been working with TOPS since its early development. He has installed and managed Macintosh networks and IBM PC networks.

A contributor to the *Macintosh Bible*, Stephen has also written magazine articles for *MacWeek* and is the West Coast editor for *CD ROM Review*. His first HyperCard program, Port Authority, is available from Heizer Software.

HAYDEN BOOKS

A Division of Howard W. Sams & Company

4300 West 62nd Street

Indianapolis, Indiana 46268 USA

$21.95/48405

ISBN: 0-672-48405-6

0 81262 48405 0