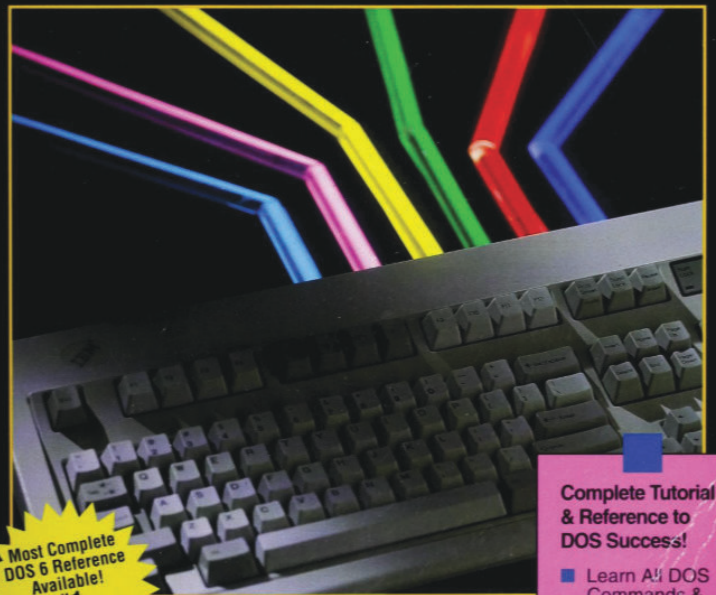


Special Edition
**USING
MS-DOS® 6**



Most Complete
DOS 6 Reference
Available!
#1
Best-Seller!

que

**Complete Tutorial
& Reference to
DOS Success!**

- Learn All DOS Commands & Utilities
- Optimize DOS Memory & Disk Space

Using MS-DOS 6, Special Edition.

Copyright © 1993 by Que® Corporation.

All rights reserved. Printed in the United States of America. No part of this book may be used or reproduced in any form or by any means, or stored in a database or retrieval system, without prior written permission of the publisher except in the case of brief quotations embodied in critical articles and reviews. Making copies of any part of this book for any purpose other than your own personal use is a violation of United States copyright laws. For information, address Que Corporation, 11711 N. College Ave., Carmel, IN 46032.

Library of Congress Catalog No.: 93-83296

ISBN: 1-56529-020-8

This book is sold as is, without warranty of any kind, either express or implied, respecting the contents of this book, including but not limited to implied warranties for the book's quality, performance, merchantability, or fitness for any particular purpose. Neither Que Corporation nor its dealers or distributors shall be liable to the purchaser or any other person or entity with respect to any liability, loss, or damage caused or alleged to be caused directly or indirectly by this book.

95 94 93 4 3 2

Interpretation of the printing code: the rightmost double-digit number is the year of the book's printing; the rightmost single-digit number, the number of the book's printing. For example, a printing code of 93-1 shows that the first printing of the book occurred in 1993.

Some of the art in this book was created by Hartman Publishing. Screens reproduced in this book were created using Collage Plus from Inner Media, Inc., Hollis, NH.

This book is based on MS-DOS Version 6.0 and earlier versions.

Publisher: Lloyd J. Short

Associate Publisher: Rick Ranucci

Operations Manager: Sheila Cunningham

Publishing Plan Manager: Thomas H. Bennett

Acquisitions Editors: Chris Katsaropoulos and Sarah Browning

CREDITS

Title Manager

Walter R. Bruce III

Product Directors

Timothy S. Stanley
Brian Underdahl

Production Editors

Tracy L. Barr
Joy M. Preacher
Kathy Simpson

Editors

Elsa M. Bell
Barb Colter
Kelly Currie
Gregory R. Robertson

Technical Editors

Matthew Harris
Dave Knispel
Eric P. Bloom
Horace E. Shelton
Jerry Ellis

Book Designer

Scott Cook

Production Team

Jeff Baker
Claudia Bell
Jodie Cantwell
Paula Carroll
Laurie Casey
Brad Chinn
Brook Farling
Carla Hall-Batton
Heather Kaufman
Bob LaRoche
Jay Lesandrini
Linda Seifert
Tina Trettin
Phil Worthington

Formatter

Jill L. Stanley

Composed in Cheltenham and MCPdigital
by Que Corporation



The text in this book is
printed on recycled paper.

Key Terms Used in This Chapter

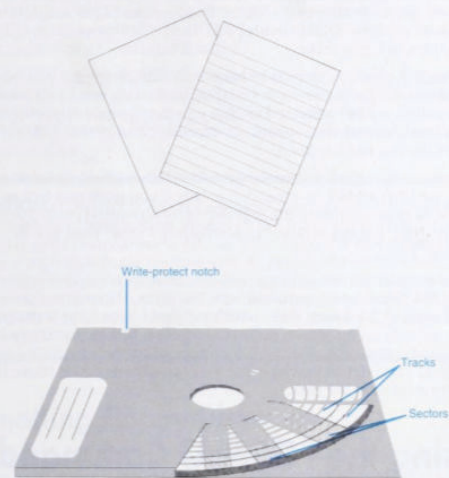
Format	Initial preparation of a disk for data storage
Volume label	A name that identifies a particular disk
Track	A circular section of a disk's surface that holds data
Sector	A section of a track that serves as the disk's smallest possible storage unit
Cluster	A unit of one, four, or eight sectors, the smallest amount of disk space that DOS allocates to a file
Disk partition	A division of a hard disk that DOS views as a separate disk
Logical drive	A partitioned section of a hard disk that DOS views as an additional hard disk
Boot sector	A special area in track 0 of each DOS disk; used by DOS to record important information about a disk's format to reference later when working with the disk
Internal command	A DOS command built into COMMAND.COM
External command	A DOS command whose instructions are stored in a file other than COMMAND.COM

Understanding Disk Preparation

Both floppy disks and audiocassette tapes use magnetic media to store information, but in some ways, disks and tapes are different. You cannot just drop a disk into a drive and use the disk; you must prepare your disks before you can store information on them. This preparation process is called *formatting*. DOS's FORMAT command performs this process for disks; you do nothing more than enter the command. FORMAT analyzes a disk for defects, generates a root directory, sets up a storage table, and makes other technical modifications.

Think of unformatted and formatted disks as being comparable to unlined and lined sheets of paper (see fig. 7.1). The lines on a sheet of paper give you an orderly way to record written information. The lines

also act as a guide for whoever reads the information. New unformatted disks are like unlined sheets of paper to DOS.

**Fig. 7.1**

Formatted disks are comparable to lined paper.

Fig. 7.2

Anatomy of a microfloppy disk.

When you format a disk, DOS creates data-storage divisions on the disk's surface. As you learned in Chapter 6, "Understanding Disks and Files," these divisions are concentric circles called *tracks* (see fig. 7.2). DOS decides what type of drive you have and then positions the tracks accordingly. DOS also writes on the disk's first track—track 0—vital information about the disk's format. Later, DOS can read this information quickly to determine how to work with the disk.

As you can see in figure 7.2, each track is divided into segments called *sectors*, which are the smallest divisions of a disk. When you issue the FORMAT command, DOS creates a special disk table called the *file allocation table* (FAT). The FAT, which is always located on track 0, logs every sector on a disk in units of one or more sectors called *clusters*. DOS stores data in the clusters and uses the track number and cluster number to retrieve the data.

A cluster is the smallest unit of storage that DOS allocates for file storage at one time. All DOS commands use tracks and clusters as road

maps that enable them to carry out their operations. A standard 5 1/4-inch double-density floppy disk (one that holds 360K of data) has 40 tracks per side. High-density 5 1/4-inch disks (1.2M) and both double-density and high-density 3 1/2-inch floppy disks (720K and 1.44M) have 80 tracks per side. DOS provides FATs and directories suited to the capacities of these disks.

Although formatting disks may become a routine job for you, always be careful during the process because formatting erases all information from a disk. If you format a disk that you used previously, everything stored on that disk disappears. Be careful not to format a disk that contains files you want to keep.

In most cases, a disk formatted with DOS 5.0 or 6.0 can be unformatted with the UNFORMAT command, but only if you unformat before you create or copy files onto the inadvertently formatted disk. (For more information, see the section "Unformatting a Disk" later in this chapter.)

One way to avoid formatting a disk by accident is to label your disks, using the paper labels supplied with the disks. Other precautions are to write-protect 5 1/4-inch disks with a tab and to open the write-protect slide window on 3 1/2-inch disks. In any case, before you try to format a previously used disk, use the DIR command to see which files are on the disk. Then determine whether you can afford to lose those files when you format the disk.

Using the FORMAT Command

The FORMAT command prepares disks for use as external data-storage media. You must format a floppy disk or hard disk before you can use it. You also can reformat disks that have already been formatted.

NOTE

When you issue certain external commands like FORMAT, DOS prompts you to insert the proper disk into the drive before DOS carries out the command. If you have a floppy-disk system, remove the disk that contains the FORMAT command, insert the disk to be formatted, and then press any key. Hard-disk systems normally are set up to search for and load an external command from a directory on the hard disk. For more information on this process, see Chapter 8, "Understanding and Managing Directories."

In DOS versions before 5.0, reformatting a disk in effect erased the data stored on the disk. Starting in DOS 5.0, however, the FORMAT

command first determines whether the disk contains data. If the disk contains data, FORMAT saves a copy of the boot sector, the FAT, and the root directory in a safe place on the disk where UNFORMAT can find the information when you later need to unformat the disk. You can unformat a disk formatted with DOS 5.0 or 6.0 only if you have not created or copied other files onto the disk.

In DOS 5.0 and 6.0, the FORMAT command clears the disk's FAT and the first character of each file name in the root directory but does not erase any data. (The program also scans the entire disk for bad sectors.) FORMAT then saves the first letter of each file name in a safe place on the disk.

CAUTION: FORMAT is a DOS command that can quickly wipe out the contents of a disk. Even though DOS 5.0 and 6.0 FORMAT command is safer than earlier versions, before you use this command, make sure that you are familiar with its syntax. FORMAT issues warning messages on-screen and prompts you for verification before formatting a hard disk. Take care when you use FORMAT, especially if you use a version of DOS earlier than DOS 5.0. Read the screen prompts, and keep backups of your data.

Formatting Floppy Disks

Formatting floppy disks from the DOS Shell is different from formatting floppies from the command line. Both methods are described in the sections that follow. DOS also provides several formatting options through command-line switches. See the section "Using FORMAT's Switches" later in this chapter for a discussion of how to use the switches when you format disks.

Mixing formatted disks with unformatted disks is an easy mistake to make when you are working with a fresh box of disks. To avoid a mix-up, place a mark on the label of each disk that you format. The mark may be as simple as a dot, a check, or the letter *F* (for "Formatted").

TIP

Before you begin formatting your first floppy disk, mark the disk's label to indicate that the disk will be formatted. If your computer isn't running, boot from the hard disk or insert the DOS Startup disk into drive A, and then boot your computer.

Understanding Backups and the Care of Data

Desktop computers commonly contain hard disk drives that can store thousands of pages of information. If your computer contains a hard disk, it almost certainly contains millions of bytes of software—the programs that run your computer. The hard disk probably also contains thousands or even millions of bytes of data—the information you or someone else has typed into the system. The software and data represent a significant investment in money and effort that would be lost if your hard disk were damaged or erased.

This chapter explains how to protect your computer files from a variety of menaces such as static electricity, excessive heat, and erratic electrical power. Most important, in this chapter you learn how to use several DOS commands, including XCOPY, and the two hard-disk backup programs, MSBACKUP and Microsoft Backup for Windows, to back up the files on your system and then to restore these files in the event of damage to or erasure of your disk.

NOTE

If you have never learned how to back up your disk files or produce duplicates, now is time to learn. If you use the ideas covered in this chapter, you can avoid the sinking feeling that comes from losing a significant amount of hard work to a hard disk crash, accidental disk format, or accidental file erasure.

Key Terms Used in This Chapter

Surge protector	A protective device inserted between a power outlet and a computer's power plug. By acting as a circuit breaker, a surge protector helps block power surges that can damage the computer's circuits.
Static electricity	A high-voltage charge that builds on objects (including people) and that can be discharged when another object is touched. Static electricity discharges can damage electronic circuits.
Ground	An electrical path directly to earth. Grounds can dissipate static discharges safely. A PC chassis normally is grounded, but this grounding is not always adequate to stop a static discharge.
Voltage regulator	An electrical device that keeps voltage fluctuations from reaching an electrical device. Regulators usually don't stop all power surges.
Full backup	A special series of disks containing all the data stored on a hard disk, along with information about the previous location of the data files.
Intermediate backup	A series of disks containing copies of all files modified since the most recent full backup.

Avoiding Data Loss

Today's personal computers are reliable and economical data processing machines. Like all machines, however, computers are subject to failures and operator errors. Table 13.1 lists some hardware and software problems discussed in this chapter and suggests ways to prevent these problems.

Table 13.1 Hardware and Software Problems and Prevention Techniques

Problem	Prevention
Static electricity	Use antistatic liquid or floor mat; place a "touch pad" on desk.
Overheating	Clean clogged air vents; remove objects that block vents; use air-conditioned room during the summer.
Damaged disks	Don't leave disks to be warped by the sun; use protective covers; avoid spilling liquids on disks; store disks in a safe place; avoid magnetic fields from appliances (televisions, microwave ovens, and so on); do not use a ball-point pen to write on flexible disks.
Data loss	Keep current backups of your hard disk and any diskettes that contain important data; scan for viruses; use delete-tracking.
Viruses	Always scan a questionable disk for viruses before copying any files to your hard disk.

Always be cautious about your computer's environment. If your power fluctuates and lights flicker, you may need to purchase a *line voltage regulator* from your computer dealer.

Your computer may perform erratically when it is too hot. Because circuits are not reliable when they overheat, you may get jumbled data. Make sure that your computer has room to breathe by cleaning the air vents.

You generate static electricity on your body when humidity is low, when you wear synthetic fabrics, or when you walk on carpet. Just by touching your keyboard while carrying a static charge, you can send an electrical shudder through your computer, causing a data jumble or circuit failure. Fortunately, you can avoid static problems by touching

your grounded system cabinet before touching the keyboard. If static electricity is a serious problem for you, ask your dealer about antistatic products, which are inexpensive and easy to use.

Preventing Software Failures

A minority of software packages have flawed instructions called *bugs*. Software bugs are usually minor and rarely cause greater problems than keyboard lockups or jumbled displays. The potential does exist for a software bug to cause your disk drive to operate in an erratic way, however. Fortunately, most companies test and debug their software before marketing the packages. Performing a backup of your disks is your best insurance against damage caused by bugs.

Software *viruses* are purposely created software bugs with the potential to cause a great deal of damage to your files. Most viruses are additional program codes hidden in a COM file like COMMAND.COM. A virus most often does its damage by altering the disk's file allocation table (FAT) and marking clusters as bad or unavailable. A virus also has the capability of initiating a FORMAT or destroying a disk partition table.

Viruses "multiply" through the exchange of software between users. If you load an infected COM file onto your PC, the virus can quickly infect your COMMAND.COM file. Unfortunately, you can back up an infected file, and when the file is restored, the virus continues to do its dirty work. Your best defense against a virus is to install only reputable software from a source who can attest to the program's operation. If you are hit by a virus, you may have to fall back to your write-protected master DOS disks and rebuild your system from the ground up. DOS 6.0, however, includes a pair of programs—Microsoft Anti-Virus (MSAV.EXE) and Windows Anti-Virus (MWAV.EXE)—that can monitor your system constantly to detect viruses. If necessary, these programs can also remove any viruses they find. For details on these programs, see Chapter 18, "Getting the Most from Your Hard Drive."

Preventing Mistakes

As you gain skill, you may use DOS commands that can result in the accidental loss of files. When you use commands such as COPY, ERASE, and FORMAT, you may inadvertently remove important data. For this reason, study what you typed before you press Enter. It is easy to develop a typing rhythm that carries you straight through confirmation prompts into the clutches of disaster.

Although you can use Ctrl-C, Ctrl-Break and, if necessary, Ctrl-Alt-Del to abandon commands, you are stopping what is already underway, and these "panic buttons" may not contain the damage. Because you are likely to make mistakes at one time or another, always have a backup copy of all important data.

DOS versions since 5.0 are more forgiving than previous versions of DOS. These DOS versions include a safe, reversible FORMAT command, as well as an UNDELETE command that you can use to recover from an accidental DEL or ERASE command.

For Related Information

- ▶▶ "Understanding How Viruses Spread," p. 725.
- ▶▶ "Guarding against Infection," p. 732.



Combining Copying and Directory Management with XCOPY

The external command XCOPY is an enhanced version of COPY that, among other capabilities, can create directories on the destination disk. XCOPY is therefore handy for copying a portion of your directory tree to another disk. XCOPY is a sophisticated copy command with many uses.

The XCOPY command addresses the needs of three types of computer users: those who use more than one computer, those who have hard disks, and those who want more control over which files are copied than is provided by the standard COPY command. Almost all PC users fit into at least one of these categories, so XCOPY is an important command to know and use.

XCOPY syntax is similar to COPY syntax, but the switches are more complex. XCOPY syntax is as follows:

```
XCOPY ds:paths\filenames.exts dd:pthd\filenamed.extd
/V/P/W/S/E/A/M/D:date
```

In this command, *s* means source. The *source files* are the files to be copied. If you omit the source disk drive name (*ds:*), the current disk drive is used. If you omit the source path name (*paths*), the current directory on the drive is used. You can use wild cards in the source file name (*filenames.exts*). If you omit the file name, XCOPY assumes that you want to specify the wild card *.* and copies all files in the given path. You must include at least one source parameter.

With XCOPY, *d* means destination. If you do not specify a destination disk (*ddf*) or path name (*pathd*), XCOPY uses the current disk directory. If you omit the destination file name (*filenamed.extd*), files on the source disk retain their names on the destination disk.

In an XCOPY operation, DOS may not always recognize whether a particular parameter refers to a file or to a directory. When ambiguity arises, XCOPY asks whether the destination is a file name or path name.

Consider the following command, for example:

```
XCOPY C:\WORDS\*.* A:\WORDS
```

If no directory named WORDS exists on the destination disk, DOS cannot determine whether you intend to create a file or a directory named WORDS on the A disk.

XCOPY displays the following message:

```
Does WORDS specify a file name
or directory name on the target
(F = file, D = directory)?
```

Press F when the destination (target) is a file name, or D when the destination is a directory. Unlike COPY, XCOPY creates directories on the destination disk as needed.

NOTE

If you append a backslash to the name of the target directory, XCOPY automatically assumes that the target is a directory, not a file. In the previous example, if you use the command XCOPY C:\WORDS*.* A:\WORDS\, you will not have to specify that the target is a directory. XCOPY creates the \WORDS directory if it does not exist.

Understanding the Operation of XCOPY

XCOPY is best described as a hybrid between COPY and BACKUP/RESTORE. XCOPY and COPY duplicate files between directories and disks. Unlike COPY, however, XCOPY does not copy files to a nondisk device, such as the printer (PRN) or the console (CON). Like BACKUP and RESTORE, XCOPY can copy files selectively and traverse the directory tree to copy files from more than one directory. XCOPY also can make a destination directory when one does not exist. This directory capability makes XCOPY useful for duplicating a directory branch onto another disk.

Like COPY but unlike BACKUP, XCOPY copies files that are directly usable. (You cannot use files processed by BACKUP until you have processed them with RESTORE.)

When using the XCOPY command, consider the following guidelines:

- XCOPY cannot copy hidden source files.
- XCOPY does not overwrite read-only destination files.
- If a file parameter is omitted in the XCOPY syntax, XCOPY assumes the *.* full wild-card pattern as the file parameter.
- If you include the /D switch, the date parameter must be entered in the format of the system's DATE command or in the format indicated by the latest COUNTRY command.
- The /V switch performs the same read-after-write checking as the SET VERIFY ON global verify flag.
- To use XCOPY to copy empty source subdirectories, you must specify both the /S and /E switches.

Understanding XCOPY's Switches

XCOPY has eight switches, which are described in this section. The Command Reference section gives a complete table of XCOPY's switches.

/V is the familiar verify switch. XCOPY verifies that it has copied the files correctly.

/W causes XCOPY to display the message Press any key to begin copying file(s) and waits for you to insert the source disk. This switch is particularly useful in batch files.

/P is the prompt switch. XCOPY displays the name of the file to be copied and asks whether the file should be copied. Press Y to copy the file or N to skip it. When you answer Y, XCOPY immediately copies the file.

The */S* and */E* switches affect the way XCOPY handles additional subdirectories. These two switches tap the full power of XCOPY. COPY limits itself to handling the files from one directory; XCOPY starts with the named or current directory (source) and can process the files in all additional subdirectories of the source directory, all lower subdirectories of these subdirectories, and so on. XCOPY traverses the subdirectory tree and can copy complete directory branches from one disk to another.

The */S* switch copies subdirectories and the directory specified in the command. The */S* switch instructs XCOPY to copy all designated files from the current and subsequent subdirectories to a parallel set of subdirectories on the destination disk drive. Assume, for example, that you want to copy to the disk in drive A all the files in the C:\WORDS directory and all files in the two subdirectories of C:\WORDS: \WORDS\LETTERS and \WORDS\CONTRACTS. You want XCOPY to create on the destination disk any directories necessary to hold the files from the source disks. To perform this operation, you issue the following command:

```
XCOPY C:\WORDS\*.* A: /S
```

The command executes in the following sequence:

1. XCOPY copies all files in \WORDS to the current directory on drive A.
2. If the subdirectory \LETTERS doesn't already exist on drive A, XCOPY creates \LETTERS. XCOPY then copies all files from WORDS\LETTERS to the subdirectory \LETTERS on drive A.
3. Similarly, XCOPY creates \CONTRACTS, if necessary, on the destination disk and copies all files from the subdirectory WORDS\CONTRACTS to the corresponding subdirectory on drive A.

In essence, XCOPY lifts a copy of the directory tree starting at the source directory and transplants the copy to drive A.

NOTE XCOPY does not place the copied files from the subdirectories into a single directory. XCOPY places files from one subdirectory into a parallel subdirectory. If the subdirectory does not exist on the destination, XCOPY creates the subdirectory.

The */E* switch affects the operation of the */S* switch. If XCOPY encounters an empty subdirectory on the source drive, XCOPY */S* skips the empty subdirectory. If you give the */S* and */E* switches together, XCOPY also creates empty subdirectories on the destination drive. The */E* switch has no effect without the */S* switch.

The switches */M*, */A*, and */D:date* add extra filters to the *filenames.exts* parameter. The */M* and */A* switches tell XCOPY to copy only files that have been modified since the last archival procedure that turned off the archive attribute. In Chapter 8, "Understanding and Managing Directories," you learned about the archive attribute stored in the directory with each file name. When you create or change a file, this attribute is turned on. XCOPY, when used with the */A* switch, copies only files that have their archive attribute set. The command does not affect the archive bit. When you use the */M* switch, XCOPY copies only files with the archive bit set and then turns off the archive attribute for each archived file.

NOTE

The BACKUP command can select files based on the archive attribute. If XCOPY has cleared this flag, however, BACKUP does not process the file. If you use the commands XCOPY */M* and BACKUP */M* (the switches have identical meaning for the two commands), the backup you make using BACKUP may not be complete.

The */D:date* switch selects files based on their *directory date*, the date of the file's creation or modification. XCOPY copies files created or modified on or after the date specified.

Using XCOPY Effectively

Using XCOPY, you can control by date or archive attribute the files copied; you can copy complete subdirectory trees; and you can confirm which files to copy. The command has several ideal uses: copying files selectively between disks or directories, performing a quick hard disk backup (backing up only a few critical files in several subdirectories), and keeping the directories of two or more computers synchronized.

With COPY, your control is limited. COPY duplicates all files that match the given name—an all-or-nothing approach. If you use the */P* switch with XCOPY, however, DOS asks whether you want to copy each file.

XCOPY is practical to use if you want to make backup copies of less than a disk full of files from several directories. Instead of BACKUP, you may prefer to use the command XCOPY */A* to select files that have changed since the last backup.

Keep the following points in mind when you use XCOPY to back up a disk:

- If you suspect that the files cannot fit on one disk, be sure to use the /M switch. As XCOPY copies each file, the command resets the file's archive attribute bit. When the destination disk fills, XCOPY stops. Change disks and restart the XCOPY command, again using the /M switch. XCOPY copies the files whose archive bit has not yet been reset.
- XCOPY cannot break a large source file between destination disks. If you need to back up a file that doesn't fit on a single floppy disk, you must use BACKUP.

A favorite use of XCOPY is to synchronize the contents of the hard disks of two computers. Many people have one computer at work and another at home. If both computers have hard disks, keeping the copies of programs and data files current is a major task. Which files did you change today? Which machine has the more current version?

When you want to keep separate hard disks synchronized, you may find XCOPY's /A, /D:date, and /S switches especially useful. The /S switch forces XCOPY to traverse your disk's directory structure, playing a hunting game for source files. Whether you use /A or /D depends on how often you copy files between the machines. If you copy files between the machines frequently, you may prefer the /A switch. If you let many days pass between synchronizing your computers' contents, you may find that the /D switch works better. Use the /D switch if you have run BACKUP on the source machine since you last used XCOPY. BACKUP resets the archive attribute so that XCOPY's /A switch does not catch all files changed between XCOPY backups.

Duplicating a Directory Branch

For this example, assume that your hard disk has a subdirectory named \WPFILES that contains a few word processing files. \WPFILES also has two subdirectories. The first, \WPFILES\MEMOS, contains your current memos. The second, \WPFILES\DOCS, contains your document files. You want to keep a current set of the files in these three directories stored on a floppy disk. To copy all the files in this directory branch to the floppy, issue the following command:

```
XCOPY C:\WPFILES A:\WPFILES\ /S
```

XCOPY immediately begins to read the source directories. DOS displays the following messages:

```
Reading source file(s)...
C:\WPFILES\LET9_1.WP
C:\WPFILES\LET9_2.WP
C:\WPFILES\LET9_3.WP
C:\WPFILES\LET9_4.WP
C:\WPFILES\LET9_5.WP
C:\WPFILES\DOCS\SCHEDULE.DOC
C:\WPFILES\MEMOS\SALES.MEM
7 File(s) copied
```

Because you included the /S switch, XCOPY copied the files in C:\WPFILES, C:\WPFILES\MEMOS, and C:\WPFILES\DOCS. The A:\WPFILES path parameter causes XCOPY to ask whether the name specifies a directory or a file. \WPFILES conceivably could be a user file in the root directory. The full path name of each file is echoed to the screen as the file is copied to drive A. When the command finishes, the \WPFILES directory branch has been copied to drive A.

As another example, assume that your PC experiences a hardware failure and needs to go to the shop for repairs. You want to use your floppy disk that contains the \WPFILES directory branch to place the \WPFILES directory branch on a second computer's hard disk. Enter the following command:

```
XCOPY A: C: /S
```

XCOPY reverses the copy process described in the preceding paragraphs by copying all the files on drive A to corresponding subdirectory locations on drive C.

Comparing Files with FC

The external command FC (for *file comparison*) compares two files or two sets of files to find differences in the files. Any differences are reported on-screen. When a copied file is extremely important, you can use FC to compare the file with the original. If differences are found, you know that a problem may exist with the copy. (Normally, DOS detects data integrity errors while reading and writing files. But if you want to be sure that two files are the same, you can ease your mind by using FC.)

NOTE PC DOS Versions 1.0 through 5.0 and MS-DOS Versions 3.3 through 5.0 include a simpler file-comparison command, COMP. This command, however, is less versatile than FC, provides less information, has fewer options, and cannot compare files of different lengths.

The FC command has two general syntax forms. One form uses the /B switch for a forced binary comparison; the other form uses the remaining switches in an ASCII comparison. The two forms of syntax are as follows:

```
FC /B d1:path1\filename1.ext1 d2:path2\filename2.ext2
```

or

```
FC /A/C/L/LBn/N/nnnn/T/W d1:path1\filename1.ext1  
d2:path2\filename2.ext2
```

NOTE Although placing all switches before other parameters in an FC command is conventional (and required in some earlier versions of DOS), this practice is not necessary in DOS 6.0.

d1: is an optional parameter that specifies the drive containing the first file to be compared. If the first drive parameter is omitted, FC assumes the default drive.

path1 is the path of the directory containing the first file. If the first file's path is omitted, FC assumes the default directory.

filename1.ext1 is the file name of the first file. This parameter is mandatory. If you use wild-card characters in this parameter, all files matching this parameter are compared with the second file.

d2: is the drive containing the second file to be compared. If the second drive parameter is omitted, FC assumes the default drive.

path2 is the path of the directory containing the second file. If the second file path is omitted, the command assumes the default directory.

filename2.ext2 is the file name of the second file. This parameter is mandatory. If you use wild-card characters in this parameter, the first file is compared with all files matching it.

NOTE If you use wild-card characters in both the first and second file names, FC compares the files as sets. That is, only those files whose names match in other respects are compared.

If you use the command FC *.WK1 *.BAK, for example, FC compares each worksheet file having the extension WK1 with the worksheet file having the same base name and the extension BAK. FC does not compare every worksheet file with every backup file.

/A instructs FC to abbreviate its output (DOS 3.2 and later versions), displaying only the first and last lines of each set of differences separated by an ellipsis (...).

/B performs a binary (byte-by-byte) comparison, showing the hexadecimal address and value of every differing byte.

/C causes FC to ignore the case of alphabetic characters when making comparisons.

/L instructs FC to compare the files in ASCII mode, even when the files have EXE, COM, SYS, OBJ, LIB, or BIN extensions (DOS 3.2 and later versions).

/LBn sets the number of lines in FC's buffer to *n*. The default number is 100 (DOS 3.2 and later versions). If the number of consecutive non-matching lines exceeds the buffer size, FC aborts the compare operation.

/N instructs FC to include the line numbers of lines reported in the output (DOS 3.2 and later versions).

/nnnn establishes the number of lines that must match after a difference in order to resynchronize FC.

/T instructs FC to view tab characters as literal characters rather than tab-expanded spaces (DOS 3.2 and later versions).

/W instructs FC to compress *white space*—tabs, empty lines, and spaces—into a single space for purposes of file comparison.

Understanding the Operation of FC

FC works in two modes: ASCII and binary. FC defaults to ASCII mode comparison when the files to be compared do not have EXE, COM, SYS, OBJ, LIB, or BIN extensions.

In ASCII mode, FC compares two files on a line-by-line basis. Lines from both files are held in a line buffer. FC uses the lines in the buffer to compare the first file to the second.

If FC detects a difference, FC displays the first file name followed by the last matching line and the mismatching line(s) from the first file. FC then displays the next line to match in both files.

After displaying mismatch information about file 1, FC repeats the same sequence for file 2. The file 2 name is displayed first, followed by the last matching line and the mismatching lines from file 2, ending on the next line that matches in both files, thus synchronizing the two files.

FC can help you determine whether the contents of two files are different by showing you the extent and location of any mismatch FC finds. You can use this output as an alternative to a side-by-side comparison of the file contents.

In binary mode, FC compares two files byte for byte. At the first difference, the byte offset position in the first file is reported along with the value of the two files' bytes at the position. The offset and byte values are reported in hexadecimal (base 16) form. This form of FC is essentially equivalent to the older COMP command.

TIP

If you are comparing two files that are not the same, you can quickly stop the reporting of differences by pressing Ctrl-C or Ctrl-Break.

In binary mode, FC does not attempt to resynchronize the two files by finding an adjusted point of byte agreement. If one file has an additional byte at one place in the file, FC reports the additional byte and all subsequent bytes of the file as mismatches.

If one file is longer than its comparison file, the binary mode compares as many bytes as are present and then reports that one file is longer. When a binary file comparison results in a long listing of differences, you may want to stop the FC operation by pressing Ctrl-C.

Only one switch is available in the binary mode. The /B switch causes the comparison to be binary even if file extensions indicate that the files are not binary. You use the /B switch to compare two text files in binary mode. You may find situations in which you would rather have the binary-mode output format of FC than the ASCII mode format. Binary mode format reports differences as pairs of hexadecimal values. You then can see the values of characters, such as Ctrl-G (bell), that do not produce printed output.

When you use FC, keep in mind that the default number of lines that must match in an ASCII comparison after a difference has ended is two. The files then are considered resynchronized. The number of "must match" lines can be changed using the /nnnn switch by setting *nnnn* to the desired value.

Using FC To Compare a Copied File to Its Original

Suppose that you are copying the ANSLSYS file from your hard disk to a floppy disk to use for an important demonstration on another PC. When the copy completes, you set the disk on the edge of your desk and go to the break room to get coffee. When you return, you notice that the disk has fallen off your desk and landed against the small transformer that runs your cassette recorder. You are worried that the magnetic field from the transformer has damaged ANSLSYS. To verify that the copied ANSLSYS is good, you compare it to the original by using the following command:

```
FC A:ANSLSYS C:\DOS\DRIVERS\ANSLSYS
```

After a few seconds, FC reports FC: No differences encountered. The copy of ANSLSYS seems to be good.

Comparing Two Text Files

The following example demonstrates the operation of FC. Suppose that two similar text files, ORIGINAL.TXT and ANOTHER.TXT, are located in the default directory of the current drive. ORIGINAL.TXT contains the following:

```
This is the first line.
This is the second line.
1
2
3
4
5
This is the last line.
```

ANOTHER.TXT contains the following:

```
This is the first line.
This is not the third line.
1
2
3
4
5
6
7
8
9
This is the last line.
```

Note that ANOTHER.TXT has four more lines than ORIGINAL.TXT, and that the second lines of the files contain differing text. These simple files illustrate how FC reports differences. You can use the principles illustrated here to understand the result of comparisons of more complex files.

If, for example, you want to compare the two files that are in the same directory, you issue the following command:

```
FC ORIGINAL.TXT ANOTHER.TXT
```

FC reports the following:

```
Comparing files ORIGINAL.TXT and ANOTHER.TXT
***** ORIGINAL.TXT
This is the first line.
This is the second line.
1
***** ANOTHER.TXT
This is the first line.
This is not the third line.
1
*****

***** ORIGINAL.TXT
5
This is the last line.
***** ANOTHER.TXT
5
```

```
6
7
8
9
This is the last line.
*****
```

FC displays the lines before and after the mismatched line, if any exist. FC also finds mismatches that are not on equivalent lines so that it can match text even when one file contains material not found in the other. The second report concerning ANOTHER.TXT shows all the lines between the one containing the numeral 5 and the one containing the text This is the last line. Only these two matching lines appear in the report concerning ORIGINAL.TXT.

Understanding Microsoft Backup

Compared to floppy disks, hard disks have many advantages. Hard disks are faster, have larger storage capacities and root directories, support multiple partitions, and never require a disk swap. A file on a hard disk can be many times the size of a file on a floppy disk. Commands like COPY and XCOPY enable you to keep a few duplicate files on a floppy for backup purposes, but DOS provides a pair of special programs specifically designed for the big jobs: Microsoft Backup (MSBACKUP.EXE) and Microsoft Backup for Windows (MWBKUP.EXE).

By default, the Setup program installed one of these programs when you installed DOS 6.0 on your computer. If your system has Microsoft Windows, however, you have the option of installing both programs. As you might expect, Microsoft Backup for Windows can be used only within Microsoft Windows. The DOS version, however, can be used either at a command prompt or in a DOS window within Microsoft Windows.

Both backup programs are full-featured programs that enable you to perform the following tasks:

- Copy files from a hard disk to another disk, usually a floppy disk.
- Restore the copied files to their original location or to another disk of your choice.

- Compare files on your backup copies with the originals to ensure their validity.

Both programs have several types of menus and many options to enable you to copy only those files you want to copy and to ease the process of making backup copies on a regular basis. Because the programs are rather similar, this chapter focuses on MSBACKUP. Relevant differences in the Windows version are noted.

NOTE

If you do not now use Microsoft Windows but plan to add that operating environment to your system at a later time, you must reinstall DOS 6.0 to install the Windows version of the backup program.

NOTE

The files copied to the backup disk do not have the same format as files on your hard disk or files copied with the COPY or XCOPY commands. The directories of disks written by the backup programs show only one file per disk, taking all the available space, as in the following example:

```
Volume in drive A is DEFAULT FUL
Directory of A:\
EE21123A 001 1457664 11-23-92 4:11p
1 file(s) 1457664 bytes
0 bytes free
```

These files may actually contain the data that originally appeared in many files on your hard disk, along with information about their original location. Before you can use these files, they must be restored, using the Restore portion of one of the Backup programs.

Both versions of the Backup program can read backup disks created by the other version. You need not worry, therefore, about losing access to backup data if you switch from one version of the program to the other.

Backup disks contain not only special copies of files but also directory information about each file. This information enables the programs to copy the backup files to their original locations in the directory tree.

The Backup programs can spread a single file across more than one floppy disk. This capability enables you to copy to multiple disks files that are too big to fit on a single disk. If a backup operation uses more than one disk, each disk is linked internally to the next disk to form a backup set.

During the backup operation, each disk in a set is filled to capacity before the next disk is requested. If a file is only partially written to a backup disk when the disk reaches full capacity, the remainder of the file is written to the next disk in the set.

Like XCOPY, the Backup programs can copy files selectively. You can specify a directory or branch of directories, a file name, a file name patterned after a wild card, and additional selection switches. By taking advantage of the programs' selectivity, you can maintain more than one set of backup disks, each set having its own logical purpose.

You can, for example, keep one backup set that contains every file on your hard disk. This set is insurance against data loss resulting from a hard disk failure or crash. You may have another backup set that contains only the files that have their archive attributes turned on. With an archive set for each day of the week, you can recover a week's worth of data between complete backups. When your hard disk fails, you replace it, restore using the full set, and then update with the daily sets.

You can lose the data on your hard disk in many ways. If you have not replicated the data, you may be forced to re-create the data and suffer the consequences of permanent data loss if you cannot re-create data. If you have never experienced a disk-related failure, don't be overconfident. With 10 million hard disk users working on their computers for an average of 8 hours per day, over 250,000 people will experience a hard-disk-related failure this year. You could be next.

Configuring the Backup Programs

Before you can create backups of your data with either backup program, you must first test the program for compatibility. If you have not previously run the program, the test is performed automatically the first time you do so. A special program runs through some of the backup program's menus, making selections as though you were running the program yourself. This program tests your floppy disk drives and the speed of your processor chip and hard disk. Periodically, dialog boxes appear and ask you for permission to continue. You respond by selecting "buttons" similar to those you have seen in dialog boxes in the DOS Shell and the DOS Editor. You select options in the same way: by pressing the highlighted letter, by moving the highlight to the desired button with the Tab or Shift-Tab key, or by clicking the desired button with the mouse. In the DOS version, a highlighted button has pointers at its left and right ends and is a different color from the other buttons. In the Windows version, the highlighted button is simply a different color.

The program then displays a dialog box that gives you the option of choosing the drive and medium you want to use for the compatibility test (see fig. 13.1). To select a different capacity disk, move the highlight to the appropriate entry with the cursor keys; then press the space bar. Choose OK when the setting conforms to your choice. (If you have only one low-density drive, this option is not available.) If you have two drives of the same type, the program uses both, alternating backup disks between the drives.

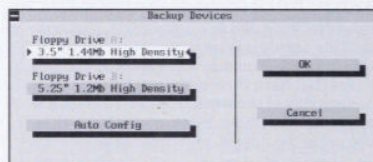


Fig. 13.1

Choosing a disk drive and capacity for backups.

The program then selects a group of files to back up and completes a backup. This procedure requires two floppy disks of the type you selected. Make sure these do not contain any data you want to save because the disks will be erased completely. The program instructs you when to insert the disks. When the backup is complete, the program compares the data on the backup disks with the source files to verify that the backup is accurate.

NOTE

Even after you configure one of the backup programs, you still have to configure the second program. However, the backup files that either program creates can be read by the other program.

Your computer will probably pass the compatibility test. If it does, you can proceed to make backups with the program. However, the program may display the following warning message:

DMA Buffer size too small. You will not be able to perform a backup, compare, or restore until the DMA buffer size is increased. See 'Troubleshooting' in your MS-DOS manual.

If this message appears, and you have an 80386 or 80486 microprocessor, use the `DEVICE` command to load the memory manager `EMM386.EXE` in your `CONFIG.SYS` file. Add the parameter `D=64` to the command, as shown in the following:

```
DEVICE=C:\DOS\EMM386.EXE D=64
```

If you have an 8088 or 80286 microprocessor and do not use an expanded memory manager, you will not encounter this problem, but the program may still fail the compatibility test. (See Chapter 17, "Configuring Your Computer," for details on `EMM386.EXE` and the `CONFIG.SYS` file.)

Most commercial expanded memory managers include a parameter that you can use to enlarge the DMA buffer. If you use an expanded memory manager other than `EMM386.EXE`, consult the documentation for details.

T I P

When you finish running the tests—and assuming that you selected the default buttons—the programs save all the settings on every menu to a file called `DEFAULT.SET`. Continue to select the default buttons until you return to a DOS prompt or the Windows Program Manager.

Understanding Microsoft Backup Functions

Both the DOS and Windows backup programs have five basic functions: Backup, Compare, Restore, Configure, and Quit. In the DOS version, the dialog box shown in figure 13.2 contains buttons representing each function. In the Windows version, the functions are represented by a series of icons near the top of the screen (see fig. 13.3).

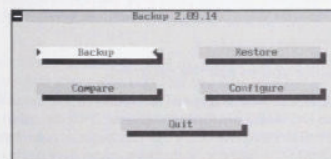


Fig. 13.2

The opening DOS Backup dialog box.

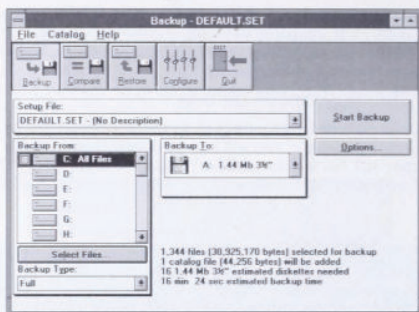


Fig. 13.3

The opening Backup for Windows screen.

The functions are listed in the following:

- **Backup** duplicates all or selected files from your hard disk on floppy disks.
- **Compare** compares the files in a set of backup floppy disks to their source on the hard disk to verify that the copies are accurate.
- **Restore** copies the files on your backup floppy disks to the hard disk, in either their original location or a new location, or to another floppy disk in usable form.
- **Configure** enables you to select default settings for the program. In the DOS version, the following choices are available:
 - Change the number of rows of text shown on the screen.
 - Switch to a display using normal text characters instead of the graphics characters shown in the illustrations here.
 - Adjust the way your mouse behaves.
 - Change the screen colors.
- **Quit** enables you to exit the program.

The Configure options, which appear in figure 13.4, are similar to the options found on DOSHELL's Options menu. You can also select a different drive or disk capacity for your backups or send your backups to a *DOS path*—that is, a directory on your hard disk.

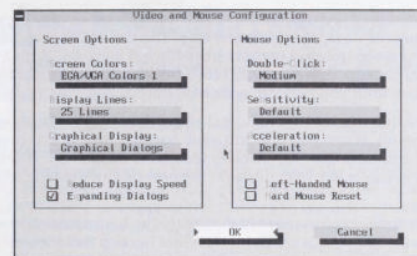


Fig. 13.4

Changing DOS Backup's display options.

Both programs have pull-down menus at the top of the screen. The File menu enables you to load and save *setup files*—files that contain all the settings you will use for a particular type of backup or restore. The File menu also enables you to print the contents of *catalog files* containing a list of the files that have been backed up. Both backup programs have a Help menu as well, which explains techniques and procedures. The Windows Backup program has a third menu, a Catalog menu, which enables you to select catalog files for restore or comparison. This menu does not appear in the DOS version; however, you can select catalog files after you have chosen to perform a restore or compare.

Setting Up a Backup Policy

Every PC user should develop a *backup policy*, a defined method of backing up data on a regular basis. Your policy may include making and keeping more than one backup set. The backup programs allow for three basic types of backups:

- A **full** backup copies every file on your hard disk. Keep at least two full backup sets so that you can re-create your system if your hard disk is destroyed; you need not make both copies at the same time.
- An **incremental** backup copies only new files or files that have been changed since the last full backup was performed. You use an additional set of disks for the incremental backup set. You may complete several backups between full backups, depending on how many files you work with and how large the files are. Each time you perform an incremental backup, you add any files that are new or modified since the previous new or incremental backup.

- A *differential* backup, like a partial backup, copies only files that are new or have been changed since the last full backup and requires a separate set of disks from the full backups. With a differential backup, however, you reuse the same floppy disks for each backup until you perform your next full backup.

The distinction between incremental and differential backups lies with the archive bit. When you perform a full backup, the backup programs turn off the archive attribute for any file that is backed up. When you modify or create a file, DOS turns on the new or modified file's archive attribute.

When you perform an incremental backup, the backup programs turn off the archive bit so that each incremental backup includes only files created or modified since the previous incremental backup. When you perform a differential backup, however, the backup programs *do not* turn off the archive bit. Therefore, every file modified or created since the previous full backup is copied. The differential backup is appropriate if you work with the same few files daily and do not need several generations of each file.

TIP

You can use the ATTRIB command to determine the current archive attribute setting for each file on your system. If you want to make sure that a particular file is backed up by an incremental backup procedure, use ATTRIB to turn on the archive attribute.

You can perform a full backup weekly, monthly, or at some other interval. The intervening (incremental or differential) backup intervals reduce your risk of data loss between full backups. The time between performing intermediate backups depends on the amount of risk to your data you are willing to take. In a business setting, you may reduce your risk to an acceptable level by performing an intermediate backup every other day. If your PC activity level is high, you may need to perform an incremental backup daily. If your PC activity is minimal, a differential backup once every two weeks may be frequent enough to reduce your risk to a manageable level.

In addition to these three types of backups, you can create backup sets for special purposes. You can select files to include in, or exclude from, the backup. You can select these files individually, by directory, according to a wild-card pattern, by date, or by attribute, and save the selections in a setup file. You learn how to select files for backups later in this chapter.

Issuing the MSBACKUP Command

The syntax for the MSBACKUP command is shown in the following:

MSBACKUP *filename* */Type* */video*

All of the switches are optional.

filename is the name of a setup file. If you created setup files for different types of backups, you can load one automatically by specifying the file on the command line. If you do not specify a setup file, MSBACKUP loads DEFAULT.SET and applies the settings to the drive specified in that file. If no drive is specified, MSBACKUP applies the settings to the default drive.

/Type specifies the type of backup. Specify the type with a single letter: */TF* for full, */TD* for differential, */TI* for incremental. The */Type* switch overrides the backup type specified in the setup file.

/video specifies the type of video display to be used. Use */LCD* for laptops or other LCD screens, */BW* for black and white, or */MDA* for monochrome displays, including those attached to Hercules-type adapters. Do not specify a video type for a color display.

For Related Information

- "Understanding CONFIG.SYS," p. 630.



Using Microsoft Backup

DOS provides a pair of special programs specifically designed for the big jobs: Microsoft Backup (MSBACKUP.EXE) and Microsoft Backup for Windows (MWBKUP.EXE). Using Microsoft Backup, you can perform a full, incremental, or differential backup; compare backups files to the original files; select which files to backup; and restore backup files. The following sections explain how to perform these tasks.

Performing a Full Backup

In this section, you learn how to complete a full backup. First you learn how to use MSBACKUP and Microsoft Backup for Windows.

Performing a Full Backup with MSBACKUP

To perform a full backup in DOS, follow these steps:

1. Enter the following command:

```
MSBACKUP
```

The program scans drive C (unless you specified a different drive and saved the information in the default setup file, DEFAULT.SET), reads its directories, and then presents its opening screen.

2. Choose the Backup button. As shown in figure 13.5, the default setup file is loaded, and the selected backup type is Full. To perform a full backup, select the drive to back up from the Backup From list by double-clicking the appropriate drive, or by using the cursor-movement keys to move the highlight to the drive and pressing the space bar. You may choose more than one drive. Each time you choose a drive, Backup scans the drive and adds the selected files to the totals.

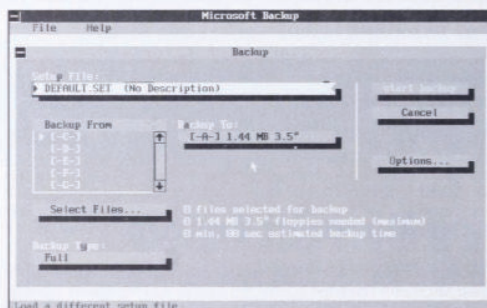


Fig. 13.5

The Backup screen.

As you select files to back up, the Backup screen shows you exactly what is required. A message similar to the following appears:

```
1,316 files (with catalog) selected for backup
24 1.44 MB 3.5" floppies needed (maximum)
15 min, 50 sec estimated backup time
```

Backup formats any target disks that are not formatted and compresses the data on the backup disks so that the capacity of the disks required may be less than the amount of data you need to back up.

NOTE

The Windows Backup program indicates how many bytes of data are selected for backup and compresses the files so that fewer backup disks and less time are required to complete a backup.

3. Choose Start Backup. The program begins by creating a backup catalog, listing the files to be backed up and the options chosen. The following message appears:

```
Insert diskette #1 in drive A:
```

As the backup progresses, a display similar to figure 13.6 appears.

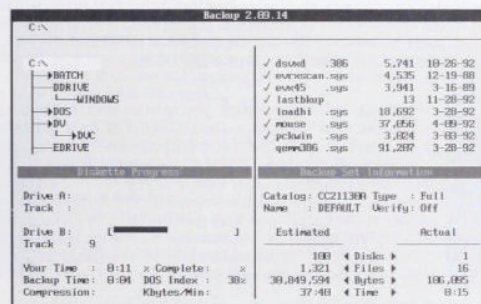


Fig. 13.6

Viewing the progress of a backup.

4. When the disk is full, the program prompts you to insert the next disk, and so on until the backup is complete.

As you remove each disk, label the disk "Full Backup #nn," where nn is the number of the disk in the series. Date the label also. When the backup is complete, the opening screen reappears.

Comparing the Backup to the Original Files

The first time you use a series of disks for a backup, compare the backups to the originals to verify that the disks are readable and accurate. You don't want to discover that your backup is unusable when you want to use it to recover a hard disk!

To perform a compare, follow these steps:

1. Choose **C**ompare. A screen similar to the main Backup screen appears.
2. Choose Backup Set Catalog, and select the setup file you used for the backup. (If you haven't saved any setup files, this will be DEFAULT.SET.)
3. Select the drive or drives from the Backup From list; the number of files to compare changes from 0 to the number of files you backed up.
4. The program prompts you to insert each disk in turn, and a progress screen keeps you apprised of events.

At the end of the process, the dialog box indicating that no errors were found or the errors were corrected will probably appear. If this is not the case, you may want to discard the disks that contain the errors, replace them with new ones, and repeat the backup.

NOTE

When you quit the program, you may be told that you have not saved your changes in the DEFAULT.SET setup file. In this case, you are asked if you want to save your changes or discard them. You may find it more helpful to save your settings explicitly to other files, which you can load either from within the program or from the command line. To save a backup setup, choose **S**ave **A**s from the File menu, and specify a file name and, optionally, a drive and directory where the setup file should be stored. The program automatically supplies the extension SET.

Performing a Full Backup in Windows

The procedure for performing a full backup in Windows is essentially the same as for DOS. The primary difference is the appearance of the screen. When the program loads, you see a message warning you not to use your disk drives while a backup, restore, or compare operation is in progress. You can prevent this message from reappearing each time you load the program by checking the Disable This Message box.

Otherwise, you proceed the same way you would in DOS. You can work in other programs while the backup is proceeding. A beep informs you when you must change disks.

Performing Intermediate Backups

Two types of intermediate backups are possible: incremental and differential. The difference depends on the status of the archive bit for selecting files. The incremental backup resets the archive bit after completion, and the differential backup does not.

Performing an Incremental Backup

To perform an incremental backup, choose Backup Type, and then choose Incremental from the dialog box. The backup proceeds as the full backup proceeds. Mark each disk in the set with the number and the date belonging to the increment for this backup. Store the completed set in a safe place, and avoid mixing the disks with disks from another backup set.

For example, you may perform an incremental backup every Tuesday and Friday and run a full backup every two weeks. Under this plan, you accumulate four incremental backup sets before you need to repeat the backup cycle and reuse the disks.

NOTE

No matter what type of backup you perform, the backup programs erase the target disks; therefore, each incremental backup must start on a new target disk. You cannot add backup files to disks that have been used as part of a set. By default, the backup programs warn you when the disk you are using is part of an existing backup set and give you the options of overwriting the disk, placing a different disk in the drive, or canceling the backup.

Performing a Differential Backup

If you use the differential backup method, you have just one intermediate backup set. The first time you do an intermediate backup in a backup cycle after a full backup, start with disk number 01 in your intermediate backup set. You may need more than one disk to complete the backup. Because the differential method does not reset the archive bit, the next intermediate backup set includes all the files included in

the first intermediate backup set. Consequently, you can reuse the same backup disks for each intermediate backup set. To perform a differential backup, select **Differential** from the **Backup Type** dialog box, and proceed as you would with a full backup.

The differential method has the advantage of using fewer backup disks than the incremental method. On the other hand, you retain only one intermediate copy of your files that have been modified or created since the previous full backup. If you think you may need to examine successive iterations of a file, use the incremental method.

Special-Purpose Backups

You are not limited to full and intermediate backups of entire disks. You can specify any file or group of files to be included in a backup. When you complete a project, for example, you may want to back up all the files associated with that project in a special series for archival purposes. Or you may want to back up such files daily in a series of incremental or differential backups. Both backup programs provide the means to make any number of special-purpose backups.

To select a group of files for a special backup, first make sure that your backup type is **Full**. Otherwise, you cannot include files that do not have their archive bit set. Next, choose **Select Files**. The backup program scans the default (or selected) drive to see how many files and directories are on it and displays a screen similar to the DOS Shell, with drives at the top, directories at the left, and files at the right (see fig. 13.7). The status line displays the following message:

Select entire directories with right mouse button or Spacebar

You also see a series of buttons at the bottom of the screen. The Shell-like window and the buttons provide two different (although complementary) ways to select files for backing up. These procedures differ somewhat in the two backup programs.

Selecting Files Manually

When you choose **Select Files** and go to the selection screen, you see that the root directory of the current drive is highlighted. To select an entire directory, move the highlight to it; then double-click, press the space bar, or click once with the right mouse button. A pointer appears next to the root directory, and check marks appear next to the file

names. To select only some of the files in a directory, select the files individually in the file window, or select the directory containing them and then deselect the files you want to exclude. To select or deselect a file, follow the same procedure you use to select a directory.

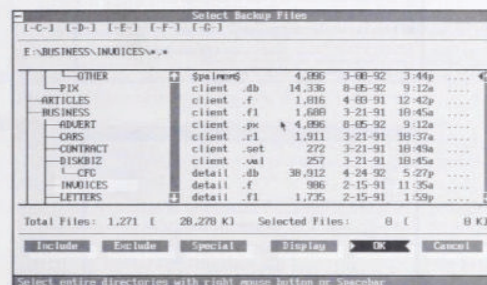


Fig. 13.7

Selecting files to back up.

When only some of the files in a directory are selected, the pointer next to the directory name changes to this symbol: >.

Press **Alt-N** to select the **Include** button, and then choose **OK**. This action selects all the files in all the subdirectories of the root directory—that is, the entire drive. Choose **OK**; the Backup screen reappears.

You select files and directories for backup in Microsoft Backup for Windows the same way as in MSBACKUP. The main difference is the manner in which selection information is displayed. A series of symbols lets you know in detail which directories have some or all files selected and whether some or all the selected files will be backed up. (Files may be selected and not backed up if you have chosen an intermediate type of backup, or if you have given the program explicit instructions not to back up some of the selected files, as described in the next section.) Choosing the **Legend** button displays the description of the symbols shown in figure 13.8.

If you want to reuse the selections you have made, be sure to save the selections in a setup file; if you don't do this, your selections are saved in the **DEFAULT.SET** file. To save your selections in a setup file, choose **Save Setup As** from the **File** menu. The **Save Setup File** dialog box appears, enabling you to give the backup set a name and a description (see fig. 13.9). You can optionally choose a different drive and directory for the setup file. (The default is **C:\DOS**.) Figure 13.9 shows the Windows version of the dialog box, but the DOS version is very similar.

Fig. 13.8

File selection legend in Microsoft Backup for Windows.

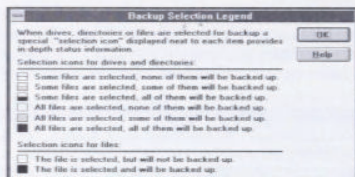
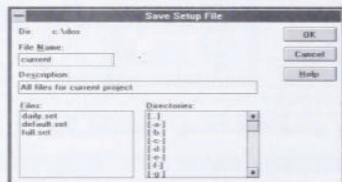


Fig. 13.9

The Save Setup File dialog box.



The next time you want to perform a backup of this type, load the setup file. In the DOS version, you can load the setup file from the command line or by choosing **Open Setup** from the **File** menu. In the Windows version, you can load the setup file from within the program using the same command. (The **Open Setup File** dialog box is similar to the **Save Setup File** dialog box.) Under some mysterious circumstances, you sometimes have to choose the appropriate drive from the **Backup From** list to activate your selections; usually, however, you do not.

Choosing Files Using Selection Criteria

The buttons at the bottom of the file selection screen give you many ways to select files to include in or exclude from a backup. These buttons also control other aspects of the program:

- **Include** (Include in Microsoft Backup for Windows) and **Exclude** enable you specify a path to include or exclude, and a file name. The file name may include a wild-card pattern. You can optionally check a box to include all subdirectories of the specified path.

In addition to selecting files to include, you may want to *exclude* the following types of files, even from a full backup:

Configuration files that are regenerated and updated every time you use an applications program

Backup files created by applications programs (usually having the extension BAK, or some other extension including the characters B and K)

Temporary files (usually having the extension TMP or an extension including the \$ sign)

- **Special** enables you to exclude read-only, hidden, system, or copy-protected files, or select files by date.
- **Display** enables you to determine whether the files appear in the file window sorted by name, extension, size, date, or attribute, and whether selected files appear before unselected files when the screen is refreshed.

The **Display** button in Microsoft Backup for Windows, in addition to the functions available in MSBACKUP, enables you determine which of the directory data appears in the file window. You can selectively exclude the file date, file time, file size, and file attributes. You can also rearrange the display so that the directory window appears above the file window instead of to its left.

Microsoft Backup for Windows includes two other buttons:

- **Legend** displays the Backup Selection Legend window (refer to fig. 13.9).
- **Print** enables you to print a list of the files selected for backup, either to the printer or to a file. (You can print the contents of the setup file from either program by choosing **Print** from the **File** menu.)

Editing the Include/Exclude List in MSBACKUP

When you choose **Include** or **Exclude**, the dialog box contains a button labeled **Edit Include/Exclude List**. Choosing this button produces the dialog box shown in figure 13.10. You can select an entry to copy, delete, or edit.

To edit a selection, choose **Edit**. This command displays the **Edit** dialog box shown in figure 13.11. In this dialog box you can specify a drive, path, and file name, including wild-card patterns. You can choose whether to include or exclude the specified files and whether to apply the selection to subdirectories of the current directory.

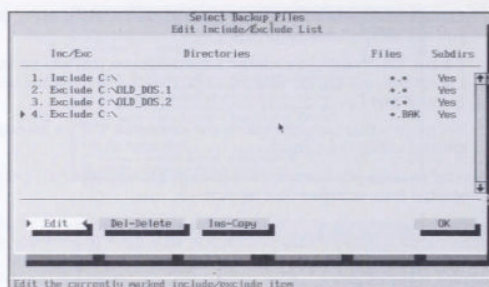


Fig. 13.10

The MSBACKUP Edit Include/Exclude List.

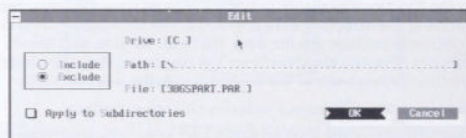


Fig. 13.11

Editing file specifications.

To use the Edit dialog box effectively, you first must copy an existing specification in the Edit Include/Exclude List dialog box (refer to fig. 13.10). Then select one of the copies, and choose **E**dit. After you enter the Edit dialog box, you can enter any specifications you want.

NOTE

In MSBACKUP, you need not include a drive name in a specification for inclusion or exclusion. By not specifying a drive name, you can apply the same specifications to any or all drives and select the drives from the Backup From list. In Microsoft Backup for Windows, you must include a drive name. However, you can load a setup file created by MSBACKUP in Microsoft Backup for Windows.

Editing the Include/Exclude List in Microsoft Backup for Windows

In Microsoft Backup for Windows, a single dialog box contains the list of included and excluded file specifications and the fields for editing specifications (see fig. 13.12). You do not need to edit existing specifications. You can just enter your criteria (which must include a drive name), and choose **A**dd.

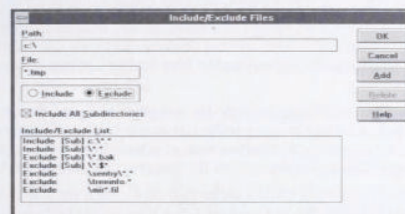


Fig. 13.12

Including and excluding files in Microsoft Backup for Windows.

Using Other Backup Options

Both backup programs give you additional options concerning their behavior. To view or change these options, use the **O**ptions button on the Backup screen. Figure 13.13 shows the resulting dialog box in Microsoft Backup for Windows, with the default options selected. The options—and the defaults—are the same in MSBACKUP.



Fig. 13.13

The Backup Options dialog box.

Understanding Batch Files, DOSKey, and Macros

When you discover that you're performing a certain task on your computer repeatedly, you should look for a way to get the computer to do more of the work. Your computer can perform most tasks faster than you can, and it doesn't make mistakes.

DOS has always enabled you to automate your use of DOS commands by creating batch files. A *batch file* is a text (ASCII) file containing a series of commands that you want DOS to execute. When you type the batch file name, DOS executes these commands in the order you specified when you created the batch file. This chapter explains how to create and use DOS batch files.

In Chapter 14, "Using the DOS Editor," you learned how to create text files with the Editor. In this chapter, you use the Editor to create batch files. You learn about batch file commands and the AUTOEXEC.BAT file, and you see some examples of batch files you can modify to suit your needs.

In addition to batch files, you can use the DOSKey program to make the command line more efficient. DOSKey stores in the computer's memory the commands you type at the DOS command line, enabling you to use these commands again without retyping them. DOSKey also enables you to store a series of commands in memory and assign them to a new command name called a *macro*. You can execute the stored commands simply by typing the name of the macro. This chapter teaches you how to use these DOSKey features.

Key Terms Used in This Chapter

AUTOEXEC.BAT	A special batch file that DOS automatically executes during the booting process. This file, usually placed in the root directory, is an ideal place to include commands that initialize the operation of a PC.
Batch file	A text file containing commands that DOS executes as though the commands were entered at the DOS prompt. Batch files always have the BAT extension.
Flow control	The capability to control the order in which DOS processes lines of a batch file.
Macro	A series of DOS commands stored in memory under a single name. You execute a macro by typing its name.
Meta-string	A series of characters that in DOS takes on a meaning different from the string's literal meaning. DOS displays substitute text when the program finds meta-strings in the PROMPT command.

Introducing Batch Files

The idea of getting a computer to do work in convenient, manageable batches predates the personal computer by several years. The earliest computers were large and expensive, and they could do only one job at a time. But even these early machines were fast. Making them wait for keyboard input between jobs was inefficient.

Batch processing was developed to make computers more productive. Collections of tasks to be carried out consecutively were put together *off-line*—that is, without using the computer's resources. The chunks,

or batches, of tasks then were fed to the computer at a rate that kept the computer busy and productive.

Today, computers are less expensive than human resources are. Batch processing enables computers to carry out a series of tasks automatically so that people don't have to waste time typing frequently used or complex commands.

A batch file is a text file that contains a series of DOS commands. Most commands used in batch files are familiar to you, having been explained in previous chapters of this book. Other commands, which control the flow of action in batch files, are available only in batch files. DOS executes the commands in a batch file one line at a time, treating each command as though you issued it individually.

You can use batch files to automate a DOS process. Using a set of commands in a batch file, you actually create a new, more powerful command. After a few experiments, you will find this DOS feature quite handy.

Recognizing a batch file in a directory listing is easy; batch files always have the file name extension BAT. You can execute a batch file by typing the file name at the DOS prompt and then pressing Enter. To execute the batch file SAFE.BAT, for example, you would type **SAFE** at the command line and press Enter. COMMAND.COM looks in the current directory for a file named SAFE that has the BAT extension; if it finds such a file, it reads the file and executes the DOS commands that the file contains.

You can use the PATH command to give DOS alternative directories to search for the batch file you enter. PATH tells COMMAND.COM where to find BAT files as well as where to find EXE and COM files.

T I P

Batch files are useful for issuing frequently used commands whose parameters you have trouble remembering. For example, you can run the Backup program by typing MSBACKUP and then selecting options from the program. But what if you have saved several sets of options in setup files for different circumstances? You could create batch files to run MSBACKUP with the proper setup file when you issue a simple command such as DAILYBAK.

Batch files can *echo* (display) messages that you designate. This text-display capability is useful for presenting instructions or reminders on-screen. You can compose messages that describe how to execute a command or that contain syntax examples and reminders. Batch files also can use the TYPE command to display text from another file.

In some ways, batch files resemble computer programs. DOS has a host of special commands that are used primarily in batch files. Although you can type some of these commands at the DOS prompt, their main use is in batch files. These commands introduce flow-control and decision-making capabilities into a batch file.

Understanding the Contents of Batch Files

Batch files consist entirely of ASCII text characters. You can create batch files in the DOS Editor, in Edlin, and in nearly any other text-editing or word processing program. (If you use a word processing program, you must use a setting that omits the special formatting and control characters that many such programs use for internal purposes.)

The easiest way to create a short batch file, however, is to use the COPY command to redirect input from the keyboard (the CON device) to a file (see Chapter 9, "Managing Your Files," for more information).

When you create batch files, observe the following rules:

- Batch files must be ASCII files. If you use a word processing program, be sure that the program saves the file without formatting characters.
- The name of the batch file, which can be one to eight characters long, must conform to DOS's file-name rules. Make batch-file names meaningful so that they are easier to remember.
- The file name must end with the BAT extension.
- The file name cannot be the same as a program file name (a file with an EXE or COM extension).
- The file name cannot be the same as an internal DOS command (such as COPY or DATE).
- The batch file can contain any valid DOS commands that you can enter at the DOS command line. (Typos cause errors.)
- You can include in the batch file any program name that you usually type at the DOS command line. DOS executes the program as though you had entered its name at the command line.
- Use only one command or program name per line in the batch file. DOS executes batch files one line at a time.

You start a batch file by typing its file name (excluding the extension) at the DOS prompt and then pressing Enter. The following list summarizes the rules DOS follows when it loads and executes a batch file:

- DOS looks first to the drive and directory specified in the command.
- Unless you specify the drive name before the batch-file name, DOS uses the current drive.
- Unless you specify a path, DOS searches the current directory for the batch file.
- If the batch file is not in the current directory or in the path you specified in the command, DOS searches the directories specified in the last PATH command you issued.
- When DOS encounters a syntax error in a batch file's command line, DOS displays an error message, skips the incorrect command, and executes the remaining commands in the batch file.
- You can stop a batch file by pressing Ctrl-C or Ctrl-Break. DOS displays the following prompt:

Terminate batch job (Y/N)?

Type **N** to skip the current command (the one being carried out) and proceed to the next command in the batch file. Type **Y** to abort execution of the batch file and return to the DOS prompt.

For Related Information

◀ "Copying from a Device to a File," p. 321.



Creating a Simple Batch File

One task that you may perform repetitively is copying and comparing files. Tasks consisting of the same series of commands are ideal candidates for batch files.

In this section, you create a simple batch file, using the COPY command to redirect input from the keyboard (the CON device) in order to create an ASCII text file. (Remember that you also can use the DOS Editor,

NOTE When you use a RAM disk and then run the DOS Shell in graphics display mode, a special RAM icon identifies the virtual disk.

Defragmenting Your Disk

If all you ever did with your hard disk was add files, the space available on your disk would be used very efficiently. DOS would add one file and then another to your disk, with each file being stored entirely within a contiguous area, followed by another contiguous file, an arrangement that enables DOS to access your files very efficiently.

Nobody, however, only adds new files to a disk. As you use your disk, you add new files, delete existing files, and add new data to existing files. Over time, data becomes scattered over the disk, and even the data within a single file may reside in chunks throughout the disk. This scattering of data is called *fragmentation*. The following sections discuss fragmentation in detail.

Understanding the Effects of Fragmentation

The more fragmented your disk, the slower your computer runs. Suppose that a third of your hard disk currently is full and that all the used space is at the front of the disk, so that you have no fragmentation. Whenever you request data from the disk, the disk heads do not have to move very far and can access your data quickly.

If the same data is spread across the disk, however, the disk head may have to move across the entire disk to access data. The longer movement requires more time, and you may notice the difference. Moreover, the disk may require more time to access each individual file. When you run a program (at least, a program that doesn't use overlays), DOS must read the program from disk into memory. That operation is much faster if the entire program is in successive locations on the disk and much slower if the disk heads must move to widely separated locations on the disk.

Fragmentation, which is almost inevitable, tends to increase the longer you use your disk. Programs that *defragment* your disk—that is, move all the data to the beginning of the disk and store each file's data in the same place—have long been available from third-party software vendors. The following section discusses DOS 6.0's built-in defragmenting program.

Understanding the Basic Operation of the DEFRAG Program

DOS 6.0 is the first version of DOS to come with a defragmenting program, called DEFRAG. You execute DEFRAG by typing **DEFRAG** at the DOS prompt. A dialog box appears, asking which drive you want to defragment (see fig. 18.1).

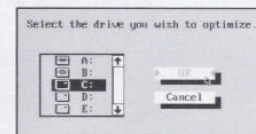


Fig. 18.1

The DEFRAG disk selection dialog box.

Select a drive by clicking a drive letter. Alternatively, use the arrow keys to move to the desired drive, and then press Enter. DEFRAG scans the selected drive and displays a map of the used and unused portions. You need not understand the map to use DEFRAG effectively, but if you're interested in understanding DEFRAG's analysis of your disk, study the legend at the bottom of the screen, which explains the symbols in the disk map.

After displaying the map, DEFRAG suggests one of the following courses of action:

- **Do nothing.** If the disk is not fragmented, DEFRAG tells you that you don't need to perform any operation.
- **Defragment files.** If most of the data in the used area of the disk is stored together but the individual files are scattered throughout that area, DEFRAG recommends that you defragment the files. After DEFRAG completes this operation, the same area of the disk will be in use, but the data in that area is rearranged so that the data for each file is stored together. DEFRAG usually recommends this operation for disks that are mostly full.
- **Defragment the disk.** If the used area of the disk is scattered across the disk, DEFRAG recommends that you defragment the disk. DEFRAG rearranges the used areas on the disk so that all the used portion of the disk is at the beginning of the disk and the data for each file is stored contiguously.

Figure 18.2 shows a typical recommendation from DEFRAG.

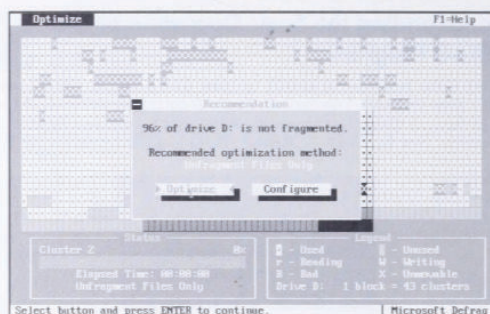


Fig. 18.2

DEFRAG
analysis and
recommendation.

If you click Optimize, DEFRAG reorganizes your disk, using the suggested method. Be prepared to wait. Although DEFRAG does its work quickly, it still may take a long time on a large, heavily fragmented disk.

When DEFRAG suggests a particular type of defragmentation, you do not have to accept its recommendation. When DEFRAG recommends that only the files be defragmented, for example, it is telling you that the extra efficiency you would gain by performing a disk defragmentation is not worth the time required by DEFRAG. However, you may be willing to let DEFRAG take the extra time to do the more complete disk defragmentation.

As figure 18.2 shows, when DEFRAG makes a recommendation, you can click Configure to select various options that control how DEFRAG works. When you click Configure, DEFRAG presents the Optimize menu, which includes more operations than just configuration—the first item tells DEFRAG to begin the defragmentation procedure. Table 18.2 lists the functions available in the Optimize menu.

If you want to change the type of defragmentation that DEFRAG performs, select Optimization Method. This selection presents a dialog box which enables you to specify whether DEFRAG defragments only files or completely defragments the disk. After you make a selection, you can select Begin Optimization from the Optimize menu; DEFRAG performs the optimization you selected.

Table 18.2 The DEFRAG Optimize (Configuration) Menu

Selection	Meaning
Begin Optimization	Begins optimization of your hard disk, using any configuration options you have selected.
Drive	Enables you to select the drive you want to optimize.
Optimization Method	Enables you to select disk optimization (full optimization) or file only (may leave empty space between files).
File Sort	Enables you to select how files will be sorted within directories, if at all.
Map Legend	Displays a legend of the symbols DEFRAG uses to show disk usage.
About Defrag	Displays information about the DEFRAG program.
Exit	Exits DEFRAG.

Sorting Directories

Normally DOS doesn't store file names in a directory in any particular order. As you create and delete files, DOS removes old names and inserts new names wherever empty space occurs. The result is that when you use DIR to list the contents of a directory, often the files appear to be listed in random order. (Some application programs, such as WordPerfect, sort this list by name each time they display the contents of a directory.)

While DEFRAG reorganizes your disk, you can tell it to organize the names of the files within the directories in one of five ways:

- **Unsorted.** This selection, which is the default, tells DEFRAG to leave the names in their current order.
- **Name.** DEFRAG organizes the directory alphabetically by file name. This option is probably the one you find most useful.
- **Extension.** DEFRAG organizes the directory by extension name; for example, all the COM files appear together, all the EXE files together, and so on.

- **Date & Time.** DEFRAG sorts the directory by the time and date the files were last modified. This option enables you to tell at a glance which files have and have not been modified recently.
- **Size.** DEFRAG organizes files by the amount of disk space they consume.

You can also specify that the sort will be ascending or descending—that is, that values get larger or smaller as the directory listing proceeds. Usually you want an ascending sort for names so that the list starts with the beginning of the alphabet and proceeds through the end of the alphabet. If you're sorting by size, however, you may want a descending sort so that the biggest files are listed first and the smallest files last.

You can specify how DEFRAG sorts directories from the DEFRAG Optimize menu by selecting File Sort. Figure 18.3 shows the File Sort menu. Alternatively, you can provide command line arguments, discussed below, to specify file sort type.

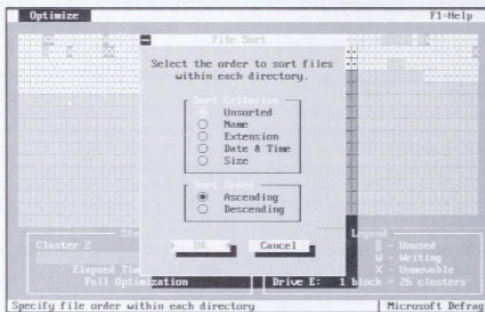


Fig. 18.3
The DEFRAG File Sort menu.

DEFRAG Startup Options

Most of the time you use DEFRAG, you will probably start it simply by typing DEFRAG. Like most DOS commands, however, DEFRAG has a number of startup options available. The complete syntax for DEFRAG is shown in the following:

DEFRAG *d* /F /U /S:order /B /SKIPHIGH /LCD /BW /G0/H

Normally DEFRAG asks you which drive you want to defragment. If you specify the *d*: parameter, DEFRAG uses that as the drive you want to defragment.

Normally DEFRAG recommends the type of defragmentation you should perform. The /F parameter tells DEFRAG to defragment the disk (leaving no empty spaces between files); the /U parameter tells DEFRAG to defragment files (possibly leaving empty space between files). If you specify one of these parameters, DEFRAG does not make a recommendation, but will immediately carry out the type of defragmentation you specify.

You can control whether or not DEFRAG sorts directory entries with the /S switch. /S is optionally followed by a colon and one or more characters that specify how you want the entries sorted within each directory. Valid letters are listed in the following:

- | | |
|---|--|
| N | Alphabetical order by name |
| E | Alphabetical order by extension |
| D | By time and date, with the oldest dates (that is, files that were last modified farthest in the past) listed first |
| S | By size, with the smallest files listed first |

For example, the following command defragments drive C: and sorts files in alphabetical order by name:

DEFRAG C: /SN

You can also place a minus sign (–) after a sort letter to specify descending sort instead of ascending. Consider these two commands:

DEFRAG C: /SS

DEFRAG C: /SS–

The first (without the minus) lists the smallest files first; the second (with the minus) lists the largest files first.

The /B switch tells DEFRAG to reboot your computer after the defragmentation process is complete.

The /SKIPHIGH switch tells DEFRAG to load itself into conventional memory. Otherwise DEFRAG uses upper memory, if available.

The /LCD switch uses a color scheme that is likely to be more readable if you are using a laptop or notebook computer with an LCD screen.

The /BW switch tells DEFRAG to use a black-and-white color scheme, which is likely to be more pleasing if you have a monochrome monitor.

The `/G0` switch disables the mouse and graphic character set. Use this switch if DEFRAG displays strange characters on your monitor.

Normally DEFRAG does not reorganize hidden files. Use the `/H` switch to tell DEFRAG to move hidden files.

NOTE

When you use DEFRAG, keep in mind the following restrictions:

- You cannot use DEFRAG to defragment drives over a network.
- You cannot run DEFRAG from Windows.
- Disk statistics reported by DEFRAG and CHKDSK differ slightly. When listing the number of directories, for example, DEFRAG counts the root directory but CHKDSK does not.

**For Related Information**

◀◀ "Using the INSTALL Command," p. 672.

Getting the Most Space from Your Hard Disk

Since the introduction of PCs in 1981, the amount of disk storage space available to the average user has increased steadily. The original PCs used floppy diskettes with less than 1M of storage space. Today, many PCs are sold with hard disks that have several hundred times that amount of space; 100M and 200M hard disks are common.

You might think that if people managed to get by with less than 1M, they should be perfectly content with 200M or more. Of course, you would be wrong. If you buy a bigger briefcase, you simply find more things to put in it. The same is true of hard disks. As disk capacity has increased, so has the need for space.

A decade ago, few software vendors wrote software that required several megabytes of disk storage because they knew that few of their potential customers' systems had that kind of storage capability. Now few vendors think twice about releasing software that requires 5M or 10M of disk storage; they expect that most people will have that much space to spare.

Although you cannot increase the physical capacity of your disk without buying a larger disk, you may be able to increase the amount of data you can store on your disk by using the techniques discussed in this section.

Increasing Space by Deleting Unnecessary Files

Much of the software available today is very complex, offering many features that you probably will never use. As a result, many of the files that are copied to your hard disk when you install a software program are files that you never open. You may be able to find and delete some of these files, thereby freeing some of your hard-disk space.

Some programs, such as Microsoft Word for Windows, ask which features you expect to use and install only the files that are appropriate for you. If you later change your mind, you can install the missing pieces later.

When you install DOS, dozens of programs are installed on your hard disk. If you are short on disk space, you may want to browse through these programs and delete those that are not important to you. Consider the following examples:

- DOS 6.0 includes many files that enable DOS to work with foreign-language character sets, including German, Swedish, and French. If you expect to use your computer only in the United States, deleting these files can free disk space.
- Some DOS 6.0 utilities may duplicate functions that are available in other programs installed in your system. If you use a backup program (such as Norton Backup or FastBack), for example, you can delete the DOS backup program, MSBACKUP.

The manual that comes with DOS 6.0 lists some of the programs and files that you may not need and can delete to make more space available.

Using Disk Compression

If you were to examine the file that contains the text for this chapter, you would find that many words—such as *DOS* and *the*—are repeated frequently. Most data includes such repetition—patterns that occur over and over within a file, whether the file contains English text, customer data, or machine instructions.