FRIENDLY

# DOS

SUPER LOW PRICE

THE LEBLOND GROUP

BANTAM COMPUTER BOOKS

56214-2 ★ IN U.S. $5.99 (IN CANADA $6.99) ★ BANTAM COMPUTER BOOKS

ISBN 0-553-56214-2

50599>

# Friendly DOS

Friendly Computer Books are available for the following major software programs:

**Friendly DOS**
by The LeBlond Group
(ISBN 0-553-56214-2)

**Friendly Excel 4.0 for Windows**
by Jack Nimersheim
(ISBN 0-553-56215-0)

**Friendly Quicken for Windows**
by The LeBlond Group
(ISBN 0-553-56216-9)

**Friendly Windows 3.1**
by Kay Yarborough Nelson
(ISBN 0-553-56211-8)

**Friendly Word for Windows 2.0**
by Jack Nimersheim
(ISBN 0-553-56212-6)

**Friendly WordPerfect**
by Kay Yarborough Nelson
(ISBN 0-553-56213-4)

# Friendly DOS

The LeBlond Group

BANTAM BOOKS

NEW YORK • TORONTO • LONDON • SYDNEY • AUCKLAND

*Friendly DOS*
*A Bantam Book / February 1993*

*All rights reserved.*
*Copyright © 1993 by The LeBlond Group*
*Cover design © 1993 by Bantam Books*

*Interior design by Nancy Sugihara*
*Produced by Micro Text Productions, Inc.*
*Composed by Context Publishing Services*

*This book may not be reproduced in whole*
*or in part, by mimeograph or any other means,*
*without permission.*
*For information address: Bantam Books*

*Throughout the book, tradenames and trademarks of some companies and products have been used, and no such uses are intended to convey endorsement of or other affiliations with the book.*

ISBN 0-553-56214-2

Published simultaneously in the United States and Canada

---

Bantam Books are published by Bantam Books, a division of Bantam Doubleday Dell Publishing Group, Inc. Its trademark, consisting of the words "Bantam Books" and the portrayal of a rooster, is Registered in U.S. Patent and Trademark Office and in other countries. Marca Registrada, Bantam Books, 666 Fifth Avenue, New York, New York 10103.

---

PRINTED IN CANADA

0 9 8 7 6 5 4 3

# Contents

Preface xiii

## 1 Introducing DOS 1

Understanding DOS 1
Working with the DOS Shell 3

## 2 Working with Directories 11

The DOS Hierarchical Structure 11
Changing Drives 12
Changing Directories 13
Revealing and Hiding Subdirectories 16
Paths 18
Creating a Directory 18
Removing a Directory 19

## 3 Working with Files 21

Naming Files 21
Copying Files 22

**4 More File Operations** 29

Moving Files 29
Renaming Files 32
Renaming a File in Its Current Location 33
Deleting Files 34

**5 Running Programs** 35

Running a Program from the File List 35
Running a Program with the File Run Command 38
Running a Program Included in a Group 40

**6 Selecting Multiple Files for a Command** 45

Methods of Selection 45

**7 Different Ways of Viewing Files and Directories** 51

Changing the DOS Shell Screen Arrangement 51

**8 Controlling the File List** 59

Changing File Display Options 59
Rereading All Files and Directories 65

**9 Searching for Files** 67

Searching for a Single File 67
Searching for a Group of Files 68

**10 Building Program Groups** 71

Understanding Program Groups 71
Creating a New Program Group 74
Adding Program Items to a Group 76
Rearranging Program Groups and Items 78
Changing the Properties of a Program Item 79

**11 Customizing the DOS Shell** 81

Customizable Features 81
Changing Colors 81
Changing the Display Mode 82
Setting Confirmation Options 84
Creating File Associations 86

**12 Special File Operations** 89

Changing File Attributes 89
Viewing a File's Contents 91

**13 Using the MS-DOS Editor** 93

Editing ASCII Files 93
Creating a Memo 94
Exiting the MS-DOS Editor 98

**14 Formatting a Floppy Disk** 99

Why Format? 99
Formatting a New Disk 100
Creating a System Disk 105

**15 Copying Floppy Disks** 107
Understanding Disk Copy 107

**16 Undeleting Files** 111
Understanding Undelete 111

**17 The Undelete-Mirror Utilities Team** 117
How Undelete and Mirror Work Together 117
Setting Up a Deletion Tracking System 118
Setting Parameters for Undelete 121

**18 Backing Up Files** 123
Why Back Up? 123
Using Backup 123

**19 Restoring Files** 131
Understanding Restore 131
Restoring Files in a Specific Directory 132
Restoring the Files in a Directory and its Subdirectories 135

**20 Task Switching** 137
About the Task Swapper 137
Starting More Than One Program 138
Switching Between Programs 140
Quitting Programs 141

**21 Working at the DOS Command Line** 143

The Purpose of the DOS Shell 143
Understanding COMMAND.COM 144
Using Commands 144
Using Doskey 149
Getting Help at the Command Line 150

**22 Working with Drives and Directories** 151

Commonly Used Commands 151
Changing to a Different Drive 151
Changing to a Directory 152
Viewing the Contents of a Directory 153
Creating Directories and Subdirectories 155
Removing Directories 156

**23 Getting Information about Files** 159

Using Dir with Files 159
Getting Information about a Specific File 159
Getting Information about a Group of Files 160
Controlling the Files List 161

**24 Working with Files from the Command Line** 163

Frequently Used Commands 163
Copying Files from the Command Line 163
Copying a Single File 164

Changing the Name of a File While Copying 164
Renaming Files 166
Deleting Files 166

**25 Changing File Attributes** 169

The Attrib Command 169

**26 Running Programs from the Command Line** 173

About Executable Files 173
Passing Parameters to Programs 174
Making the Leap from the DOS Shell 175

**27 Command Line Shortcuts** 177

Cutting Down on Typing 177
Current Shortcuts 177
Previous Location Shortcuts 179

**28 Working with ASCII Files** 181

Printing ASCII Files 181
Listing ASCII Files 182

**29 Copying Entire Directories** 185

The Xcopy Command 185
Copying a Single Directory 185
Copying Subdirectories 187

**30 Creating Batch Files** 189

Understanding Batch Files 189
Creating a Simple Batch File 189
Running the Batch File 191

**31 Rebooting Your PC** 193

When Your Computer Hangs 193

**32 Modifying AUTOEXEC.BAT** 195

The PC's Boot Sequence 195
Before You Proceed 196
Loading AUTOEXEC.BAT into
MS-DOS Editor 198
AUTOEXEC.BAT Commands 199
Saving AUTOEXEC.BAT in the
MS-DOS Editor 202

**Epilogue: Where To Go from Here** 203

**Index** 205

# Preface

Most of us become computer users because we have to, because knowledge of a particular software package is needed for a job, or because computer-assisted productivity is essential to success in business. There are hundreds of reasons. Computers and software are only the means to an end. They have become a necessity of life, and this requirement shapes the way we go about learning how to use software.

Not everyone is interested in every detail of a particular program. Here is a quick, no-nonsense introduction that teaches the basic skills needed to use the software.

In approximately 200 pages, each Friendly Computer Book covers the basic features of a specific popular software in a way that will get new users up and running quickly. The result is a series of computer books that has these unifying characteristics:

- **Topic-oriented organization.** Short, self-contained lessons focus on a particular topic or area that is important in learning to use the software.

When you finish the lesson, you'll have mastered an aspect of the software.

- **Spacious layout.** Large type and a spacious layout make the books easy on the eyes and easy to use.
- **Step-by-step approach.** Numbered lists help you to concentrate on the practical steps needed to get your work done.
- **Numerous screen shots.** Each lesson contains at least two screen shots that show you exactly how your screen should look.
- **Frequent use of icons.** Many eye-catching icons—drawing attention to important aspects of the text and software—are placed throughout the book.
- **Lay-flat binding.** Friendly Computer Books stay open as you work.
- **And finally, a low, low price.**

For many users Friendly Computer Books are all they'll need. For others who want to learn more about the software, we've suggested further readings.

Enjoy the friendly approach of Friendly Computer Books!

Ron Petrusha
Series Editor

◆ *Lesson* ◆

1

# Introducing DOS

## Understanding DOS

DOS is an acronym for "disk operating system," and it is an integrated collection of programs that acts as a link between you and the PC. As soon as you turn on your PC, DOS starts running and continues to run as long as the power remains on.

This book assumes you are using DOS 5.0. If you're running an earlier version of DOS, you won't be able to complete many of the lessons presented here. (If you're not sure which version of DOS you have, use the Ver command; see the example below.)

DOS has three primary functions. First, it manages files and directories. A *file* is a collection of information stored in a PC. Files can contain programs, letters, lists, budgets, or anything else that you want to collect and store as a group. DOS has built-in tools for creating new files and adding to existing ones. These tools also

let you copy, move, erase, and rename files. Every file is part of some *directory*, which keeps files organized. If every file were located in the same place, it would take DOS a long time to search through all the files and find a particular one. Directories provide a way of addressing files so that DOS (and you) can access them quickly and easily.

Second, DOS runs programs. The "brains" of a PC consist of a microprocessor and RAM (random access memory—temporary memory that loses its contents when you turn off the PC's power). A program consists of a sequence of instructions that are placed, for the most part, one after the other in RAM. The microprocessor reads and executes each instruction in sequence. DOS controls the process that loads programs from disk into RAM and directs the microprocessor to execute them. You may have DOS run one of its many utility programs or a program supplied by a software developer, like Lotus 1-2-3.

Finally, DOS interacts with devices that are attached to your computer. DOS gives the PC access to devices such as the screen, keyboard, printers, disk drives, and CD-ROMs.

DOS isn't very flashy. When your PC first comes up, the only evidence that DOS is working is a command line that looks like this:

```
C:\>
```

Following the C:> prompt, you can enter commands that you want DOS to execute. Try this example:

1. Type **ver**.
2. Press **Enter**.

DOS displays the following message showing its version number:

```
MS-DOS Version 5.00
```

⇨ ***Note:*** If someone else set up your PC for you, you may not see the DOS prompt when you first boot (start) your PC. Instead, you may see a menu system that lets you select various programs. Typically, such a menu system will offer an option for directly accessing DOS. Select that option.

## Working with the DOS Shell

One of DOS's most useful utility programs is the DOS Shell. This program hides much of the complexity of working directly at the DOS command line and is a good place to start when you're first learning DOS.

To start the DOS Shell, type **dosshell** and press **Enter**. After a brief pause, your screen should look something like Figure 1.1.



If the DOS Shell doesn't load, try navigating to the DOS directory. Type cd \dos and press Enter. Then attempt the dosshell command again. If the DOS Shell still doesn't load, try another drive; for example, if the DOS prompt appears as D:>, type c: and press Enter.

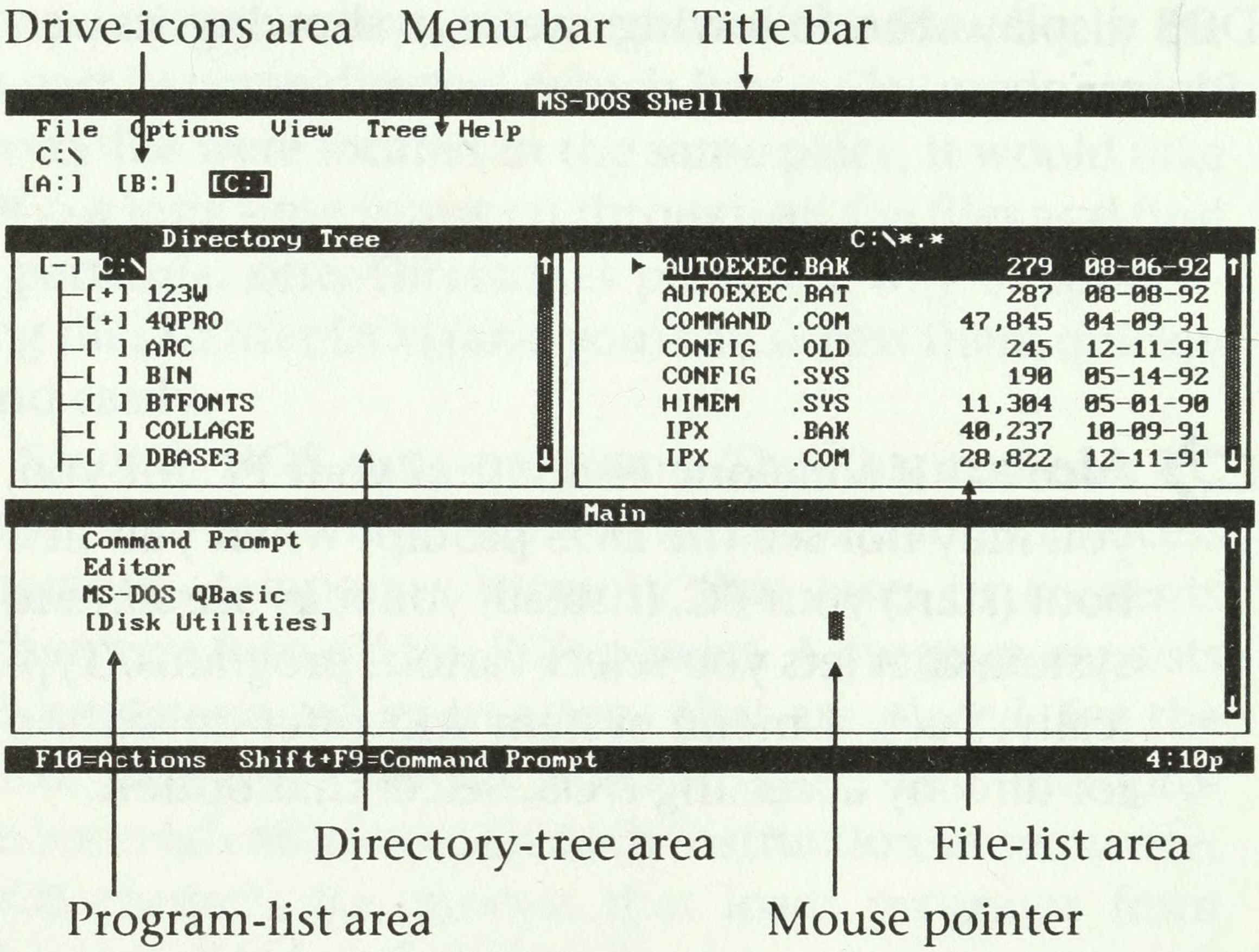



**Figure 1.1** The DOS Shell

## *Moving between Areas*

When you first start the DOS Shell, your screen is divided into four main areas:

**Drive-icons area** — This area shows in icon form the disk drives that are available on your system. For example, Figure 1.1 shows a PC with two floppy disk drives (A and B) and a hard disk drive (C). By selecting a drive icon—for example, [A:]—you can make that drive current (see Lesson 2).

**Directory-tree area** This area shows the structure of the current drive in tree form. When you choose a new drive icon, the tree that you see here changes to match the structure of the drive whose icon you've chosen.

**File-list area** This area displays the files that reside in the directory that is chosen in the directory-tree area.

**Program-list area** This area shows the programs that are accessible directly from the DOS Shell.

Initially, the drive-icons area is the active area, meaning that a drive icon (usually [C:]) is highlighted and any keys you press are directed toward that area of the screen. To move to another area using the keyboard, use Tab or Shift+Tab, as in this example:

1. To move from the drive-icons area to the directory-tree area, press **Tab**.
2. To move to the file-list area, press **Tab** again.
3. To move back to the directory-tree area, press **Shift+Tab**.

⇨ ***Note:*** If you have a mouse installed on your PC, you can use it to move around in the DOS Shell. To move to the program-list area, for example, click on that area's title bar, the one that reads "Main."

## Selecting Menu Commands

The menu bar near the top of the DOS Shell screen gives you access to many powerful commands. To activate the menu bar using the keyboard, press either **Alt** or **F10**.

Suppose you have a color screen and want to change the DOS Shell color scheme to something more lively, like Hot Pink. To do so, use the Options Colors command, as follows:

1. Press **Alt**. The File command appears highlighted in the menu bar. In addition, the first letter of each menu command is highlighted, meaning that you can select the command by typing that letter.
2. Press **o** to select the Options command, or use → to move to the Options command and press **Enter**. Figure 1.2 shows the pull-down menu that appears.
3. Select the Colors option by typing its highlighted letter, in this case, **o**; or use ↓ to move to the Colors option and press **Enter**. The DOS Shell presents a list of colors from which to choose.
4. Press ↓ repeatedly until Hot Pink is highlighted in the list. (If you go too far, press ↑ to move back up.)
5. Press **Enter** to confirm your choice. The colors of the DOS Shell title bar and highlights change to shades of pink.

To restore the DOS Shell to the default color scheme, repeat the procedure, this time choosing **Ocean** from the Colors list. To cancel a menu command at any time, press **Esc**.

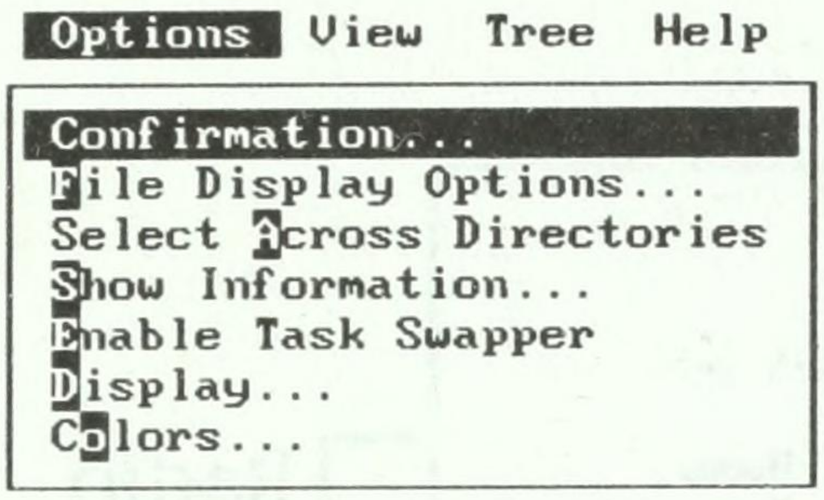



**Figure 1.2** The Options pull-down menu

***Tip***

The quickest way to select an option from the DOS Shell main menu is to press Alt+*Letter*, where *Letter* is the option's first letter—for example, press Alt+F for the File command.

***Note:*** To select a menu option using the mouse, simply click on it. To cancel a menu command, click on any area outside the menu.

As you explore the menu system, you'll discover that certain commands are grayed. This simply means that these commands are not available in the context you're currently in. For example, the Move option in the File menu appears grayed until you highlight a file in the file-list area.

## *Using Dialog Boxes*

A number of DOS Shell commands present a dialog box requesting information from you. Figure 1.3 shows an example. Typical elements contained in a dialog box are:

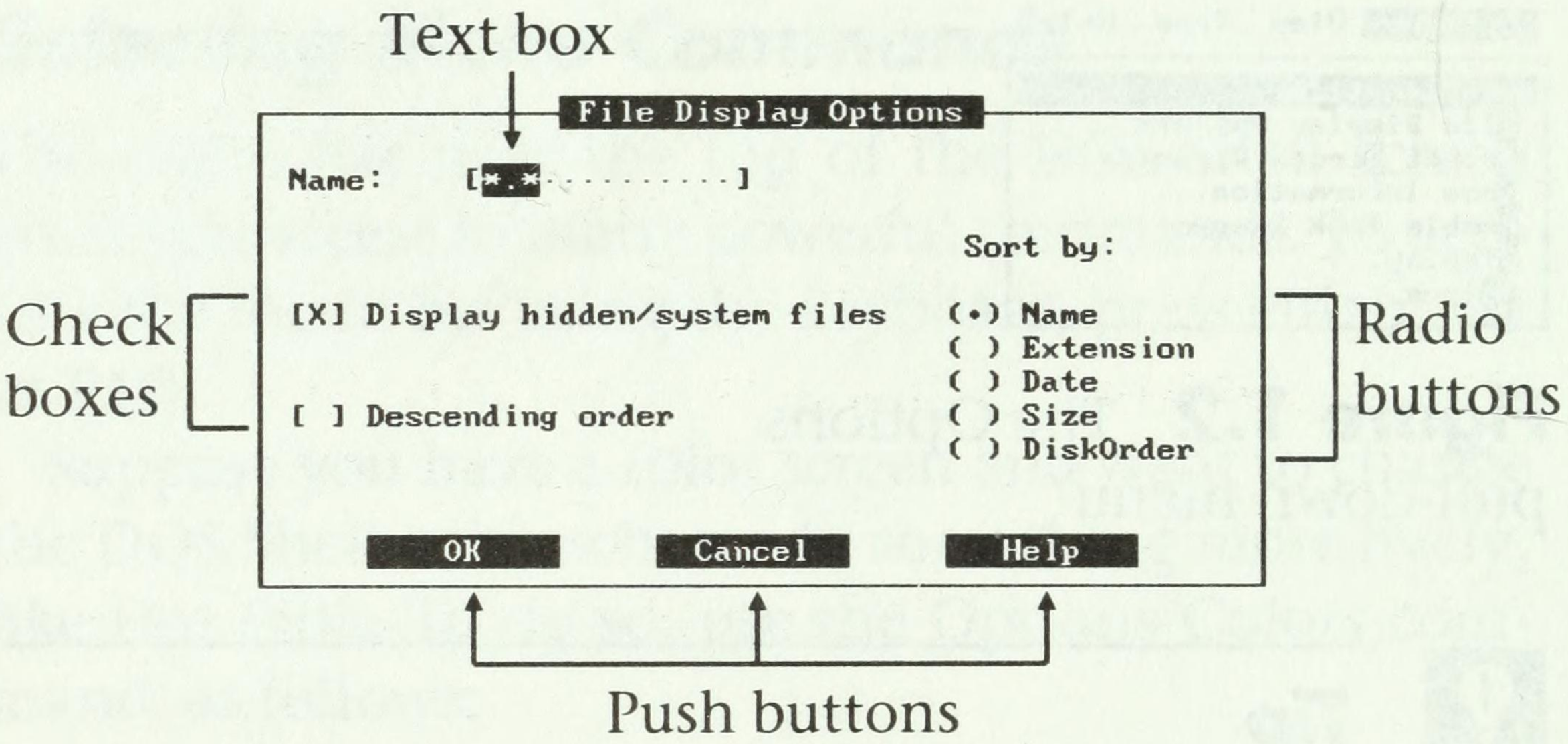



**Figure 1.3** A sample dialog box

**Text box** Accepts text input. Table 1.1 shows the keys that you can use to edit the contents of a text box. If you have a mouse, click anywhere within the text box to begin editing.

**Radio buttons** These are grouped together to show you a fixed set of choices that are mutually exclusive. To select a radio button with the keyboard, use the ↑ or ↓ keys to move to it. If you have a mouse, click on a radio button to turn it on or off.

**Check box** Acts like a switch. An X appears in the check box when it is turned on. Unlike radio buttons, check boxes are not mutually ex-

clusive. Press Spacebar to turn the check box on and off; if you have a mouse, click on it to turn it on or off.

**Push button** Initiates an action. For example, the OK push button accepts the current settings and closes the dialog box (the same as pressing Enter), the Cancel push button closes the dialog box without changing settings (the same as pressing Esc), and the Help push button displays help about the dialog box.

**Table 1.1 Keys for Editing**

| Key | Function |
|---|---|
| → | Moves right one character. |
| ← | Moves left one character. |
| Home | Moves to the start of the text entry. |
| End | Moves to the end of the text entry. |
| Del | Deletes the character at the cursor (or the highlighted characters). |
| Shift | When used with the → and ← keys, lets you highlight a group of characters. |

To move from one item to the next in a dialog box, press the **Tab** key. When items are aligned vertically—for example, the two check boxes in Figure 1.3—you can also use ↑ and ↓ to move between them. When items are aligned horizontally—for example, the OK, Cancel, and Help buttons in Figure 1.3—you can use → and ← to move among them.

## Exiting the DOS Shell

To leave the DOS Shell and return to the DOS command line, use the File Exit command, or its shortcut, Alt+F4. If you've customized the DOS Shell's configuration in any way (see Lesson 11), it saves the configuration settings to disk so that it can use them again the next time you start the program.

◆ *Lesson* ◆

2

# Working with Directories

## The DOS Hierarchical Structure

A disk drive's contents consist of a hierarchical structure of files and directories. You can think of this structure as an inverted tree. At the top is a single directory, called the *root directory* or \ *(backslash) directory*. This is analogous to the trunk of the tree. All other directories branch out from the root, just as the limbs of a tree branch out from its trunk. At the bottom of the structure are files, which are like the leaves of a tree. In between the root directory and the files are other directories. These can be thought of as the branches of a tree: They lead from the root directory to other directories and files.

Figure 2.1 shows the structure of a typical hard drive. The ellipses (...) under the names of directories indicate that these directories may branch out into other subdirectories, which for reasons of space are not shown. Any directory that is inside another is called a *subdirectory*. In Figure 2.1, for example, BUSINESS and PERSONAL are subdirectories of LETTERS. Any directory

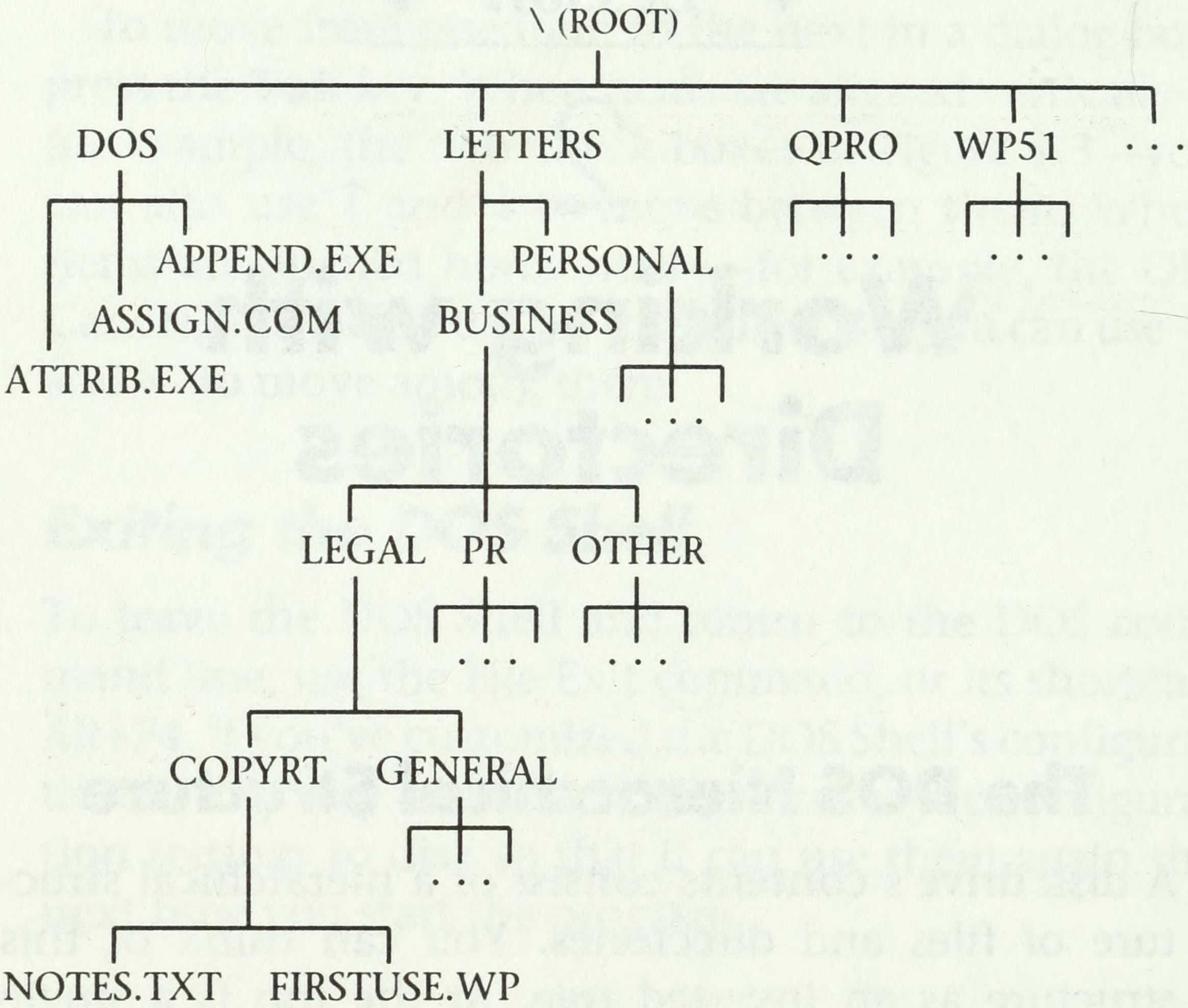



**Figure 2.1** A typical directory structure

that contains a file or subdirectory is called the *parent directory* of that file or subdirectory. In Figure 2.1, \ is the parent directory of several directories, including DOS, LETTERS, QPRO, and WP51. Notice that each directory name in Figure 2.1 is a maximum of eight characters; this is a rule in DOS.

## Changing Drives

When you first start the DOS Shell, it usually focuses its attention on drive C. You can tell that this is so because the drive C icon is highlighted in the drive-icons area and the directory-tree area shows it hierar-

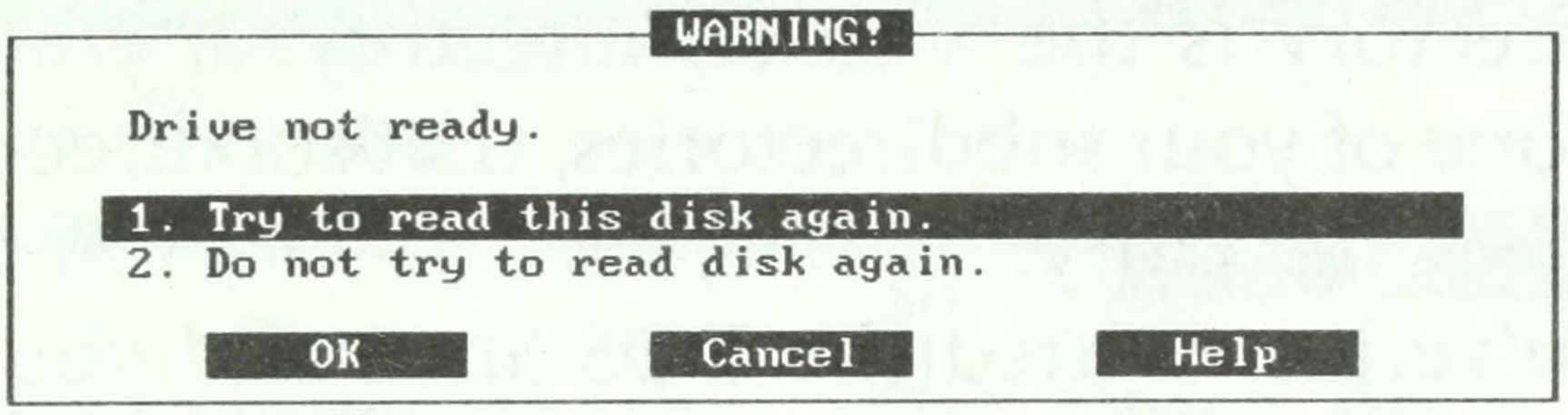



**Figure 2.2** Drive warning message

chical structure. The drive whose contents you are viewing is known as the *current drive*.

Suppose you want to make drive A the current drive. Here are the steps:

1. Insert a formatted floppy disk into drive A; any formatted disk will do. (See Lesson 14 to learn how to format a floppy disk.)
2. If necessary, press **Tab** to move to the drive-icons area.
3. Use → to highlight the **[A:]** icon and press **Enter**.

⇨ ***Note:*** If you're using the mouse, double-click on a drive's icon to make it the current drive.

If you don't insert a formatted floppy into drive A before selecting its icon, your PC will appear unresponsive for a few seconds and then show a warning message like the one in Figure 2.2, indicating that drive A is not ready. To clear the message, press **Esc** or insert a formatted disk into the drive and press **Enter**.

## Changing Directories

The *current directory* is the directory you are in at any given time. For instance, when you first boot your PC,

the current directory is the \ (root) directory. If you then move to one of your subdirectories, that subdirectory is the current directory.

Suppose you've just started the DOS Shell and you want to make the DOS directory the current directory. Perform the following steps:

1. Press **Tab** to move to the directory-tree area.
2. Press ↓ repeatedly until the DOS directory is highlighted. (Use ↑ if you've gone too far.)

Figure 2.3 shows how your screen appears after making the DOS directory the current directory.

***Tip***

Depending on the layout of your hard disk, the DOS directory may not appear at the top of the directory tree when you first start the DOS Shell. If this happens, press ↓ several times before it scrolls into view. Table 2.1 shows the keys that you can use to move within the directory tree. (These keys work similarly in the file list and other lists in the DOS Shell.)

Notice that as you move among the directories, the file-list area changes its contents to show the files in the highlighted directory. Again, the directory whose contents you are viewing at any given time is the current directory.

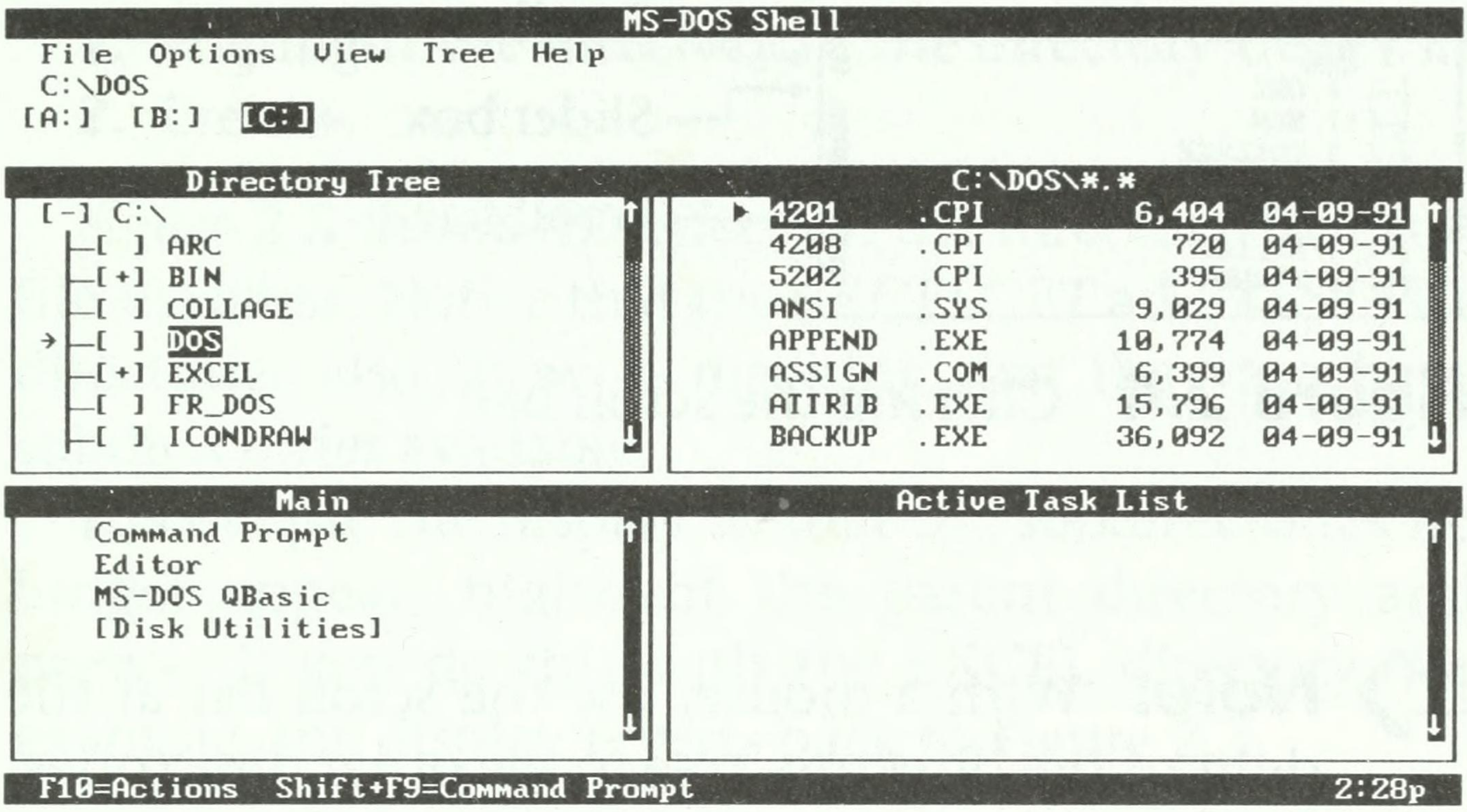



**Figure 2.3** After making the DOS directory current

## Table 2.1 Keys in the Directory Tree

| Key | Purpose |
|---|---|
| Home | Moves to the first directory in the directory tree, which is the root directory of the current drive. |
| End | Moves to the last directory in the directory tree. |
| PgDn | Moves down by approximately 8 directories. |
| PgUp | Moves up by approximately 8 directories. |

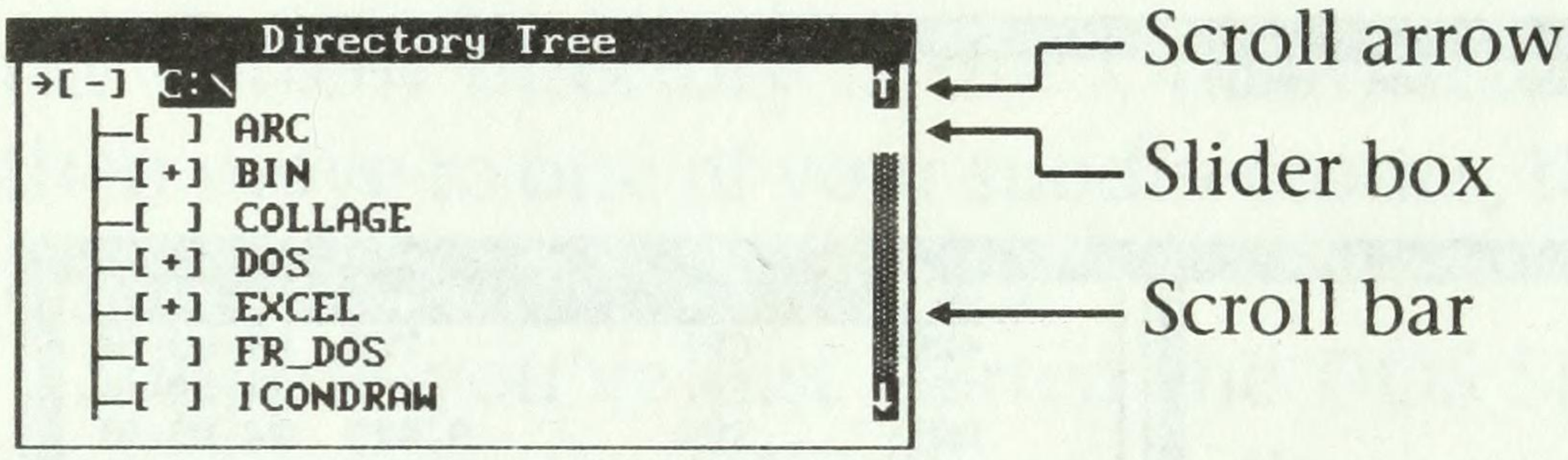



**Figure 2.4** Clicking the scroll bar

⇨ ***Note:*** With a mouse, use the scroll bar at the right of the directory tree to see other directories, as shown in Figure 2.4. (If you're new to the mouse, "dragging" is the term used to describe the procedure of holding down the left mouse button as you move the mouse, and then releasing the mouse button.) Click on an arrow to scroll one directory at a time; to scroll continuously, point to an arrow and hold down the mouse button continuously. To move in large increments to a different area, drag the slider box in the scroll bar. Click above or below the slider box to scroll approximately eight directories at a time.

## Revealing and Hiding Subdirectories

When a directory has a + preceding it in the directory tree, it means that it has subdirectories available. For example, if you want to see the subdirectories available from the EXCEL directory in Figure 2.3, use the following steps for revealing the subdirectories from the keyboard:

1. Highlight the directory in the directory tree.
2. Press +.

Figure 2.5 shows the effect on the directory-tree and file-list areas. Notice that the EXCELCBT and XLSTART directories also show +, meaning that they too have subdirectories available.

To collapse the display so that the subdirectories no longer appear, highlight the parent directory and press –. If you do this with the EXCEL directory, for example, the display reverts back to Figure 2.3.

**Tip**

The quickest way to reveal all available subdirectories on the current drive is to press Ctrl+* (asterisk) from within the directory tree.

If you're using the mouse, click on the + preceding a parent directory to show its subdirectories; to hide subdirectories, click on the – preceding the parent.

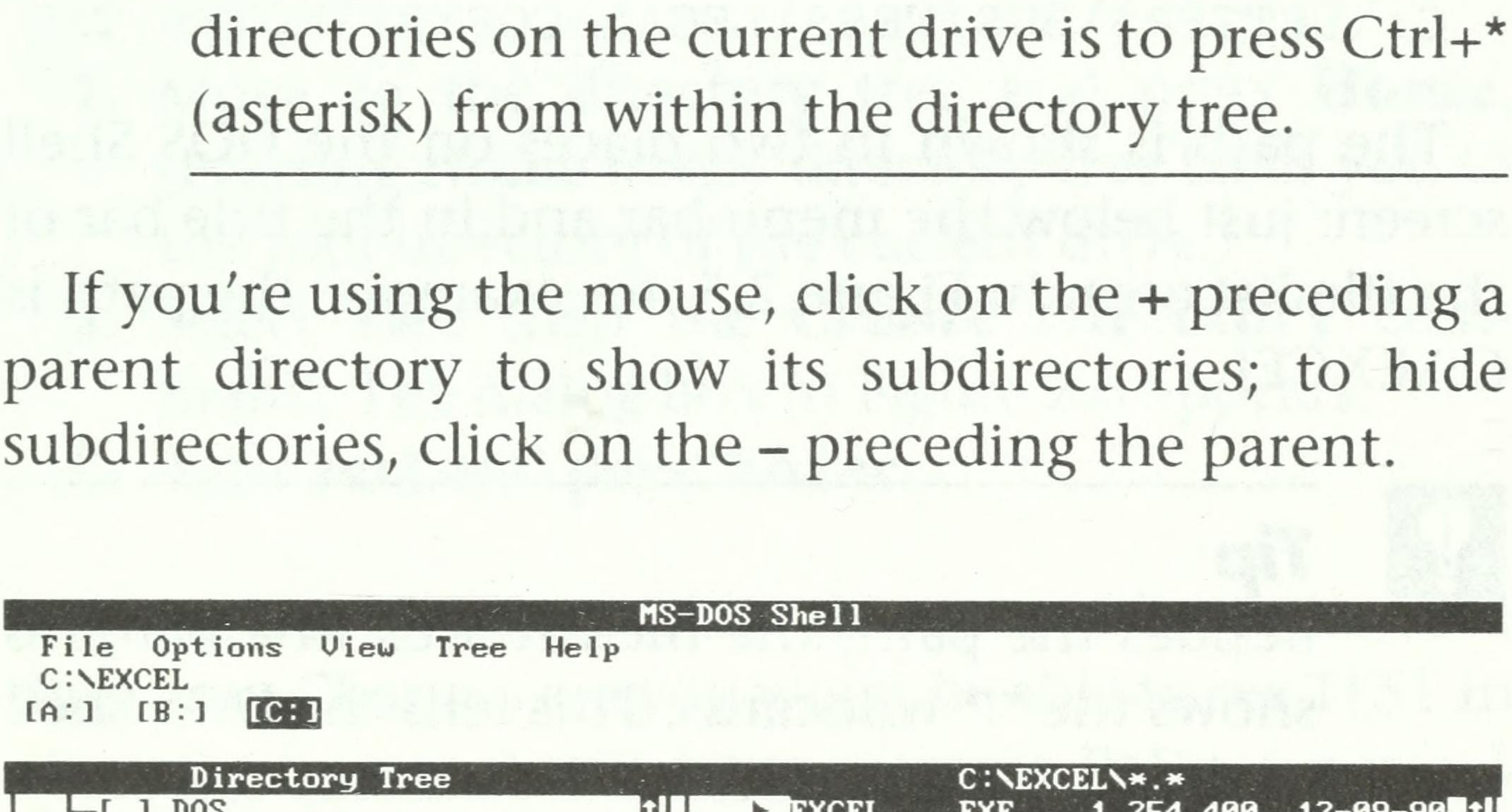



**Figure 2.5** After pressing + to view subdirectories

## Paths

The *path* of a file or directory is the list of all the directories that form the connection between the root directory and the item in question. The path begins with the drive letter followed by a colon—for example, C:. It then includes a \ (backslash), which is the name of the root directory. A backslash also separates the names of the directories in a path. For example, here is the path for the file NOTES.TXT shown in Figure 2.1.

```
C:\LETTERS\BUSINESS\LEGAL\COPYRT
```

To show the path and filename together, use

```
C:\LETTERS\BUSINESS\LEGAL\COPYRT\NOTES.TXT
```

The path is shown in two places on the DOS Shell screen: just below the menu bar and in the title bar of the file-list area. In Figure 2.5, for example, the path is C:\EXCEL.

***Tip***

Besides the path, the file-list area title bar also shows the *.* wildcards. This tells the DOS Shell to display *all* the files in the current directory. (See Lesson 8 for more on wildcards.)

## Creating a Directory

To create a new directory off the current directory, you use the File Create Directory command. Try the following example to create a directory named TEST off the root directory:

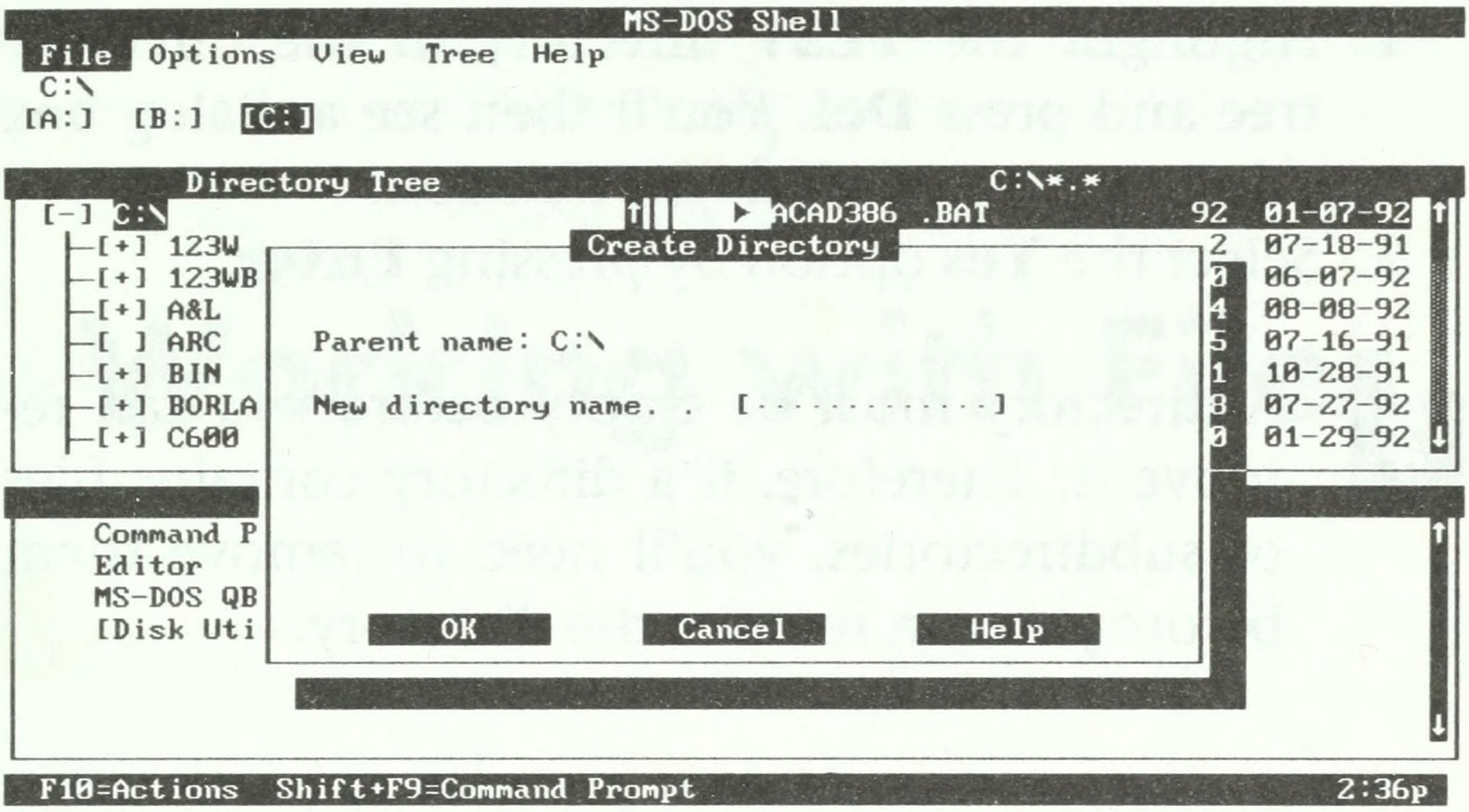



**Figure 2.6** The File Create Directory dialog box

1. Move to the directory tree and press **Home**. (Pressing Home in the directory tree takes you to the root directory of the current drive.)
2. Select **File** then the **Create Directory** command. The dialog box in Figure 2.6 appears.
3. Type **test** and press **Enter**.

DOS creates the new TEST directory off of the root directory. Chances are you won't be able to see TEST in directory tree and will have to press PgDn (or use ↓ repeatedly) to bring it into view.

## Removing a Directory

To remove a directory, you use the File Delete command (or press Del, its shortcut). For example, to delete the TEST directory you created in the previous example, follow these steps:

1. Highlight the **TEST** directory in the directory tree and press **Del**. You'll then see a dialog box asking you to confirm the deletion.
2. Select the **Yes** option by pressing **Enter**.

A directory must be empty before you can remove it. Therefore, if a directory contains files or subdirectories, you'll need to remove them before you can remove the directory.

◆ *Lesson* ◆

3

# Working with Files

## Naming Files

Filenames can be up to eight characters long with a three-character extension. If you attempt to use a name that is longer than eight characters, DOS disregards the extra characters. Here are some additional rules that apply to filenames:

- You can use any combination of letters, numbers, underscores (_), and hyphens (-). However, you cannot use the *, ?, |, /, <, or > characters because these have special meaning in DOS. In addition, you can use a period only once in a filename, to separate the filename from its extension.
- You cannot use spaces.
- You can use either uppercase or lowercase letters.

The following are all acceptable filenames:

| | | | |
|---|---|---|---|
| CASH-93.WQ1 | 75%_PLAN.TX | MKT&SLS.WP | _BACKUP.BAK |
| $60PRICE.BAS | 07-23-93.XLS | 4403.DRV | #D11.C |

Although file extensions are optional, most veteran DOS users take advantage of them for these reasons:

- They make it easier to distinguish files from directories. (Directories share all the same naming conventions as files, but are rarely given extensions.)
- They help identify the type of file you are dealing with—for example, .TXT usually means a text file and .BAK a backup file.
- Most application programs (other than word processors) automatically assign their own unique file extensions—for example, .XLS for Excel worksheet files and .WQ1 for Quattro Pro files.
- Using the same file extension for files that have a similar purpose lets you match them all by a single pattern and a wildcard character. For example, if you have several word processing files concerning legal action you have pending against widows and orphans, you might want to give them a .w&o extension, just to indicate to yourself what these files are about. Then, whenever you want to do some operation on these files, you could refer to them as *.w&o. (Lesson 8 will tell you all about wildcard characters.)

## Copying Files

To copy a file using the keyboard, select it in the file-list area and then use the File Copy command (or its shortcut, F8). This sequence lets you make a copy of a file with a new name or in a new directory and leave the original file as it was. It's handy for making backup copies of important files and when you want to have a file available in more than one directory.

The method for using File Copy is slightly different depending on whether you want to copy a file within a directory or copy the file to another directory.

## Copying a File Within a Directory

To copy a file within a directory, you must give it a different filename. For example, suppose you want to make a copy of the file README.TXT in your DOS directory and name it TEST.TXT in that same directory. Here are the steps:

1. Select the **DOS directory** in the directory tree. All its files will appear in the file list.
2. Press **Tab** to move to the file list and press ↓ repeatedly until README.TXT is highlighted.
3. Select **File** then the **Copy** command or press **F8**. The dialog box shown in Figure 3.1 then appears.
4. Type **test.txt** for the To: location and press **Enter**.

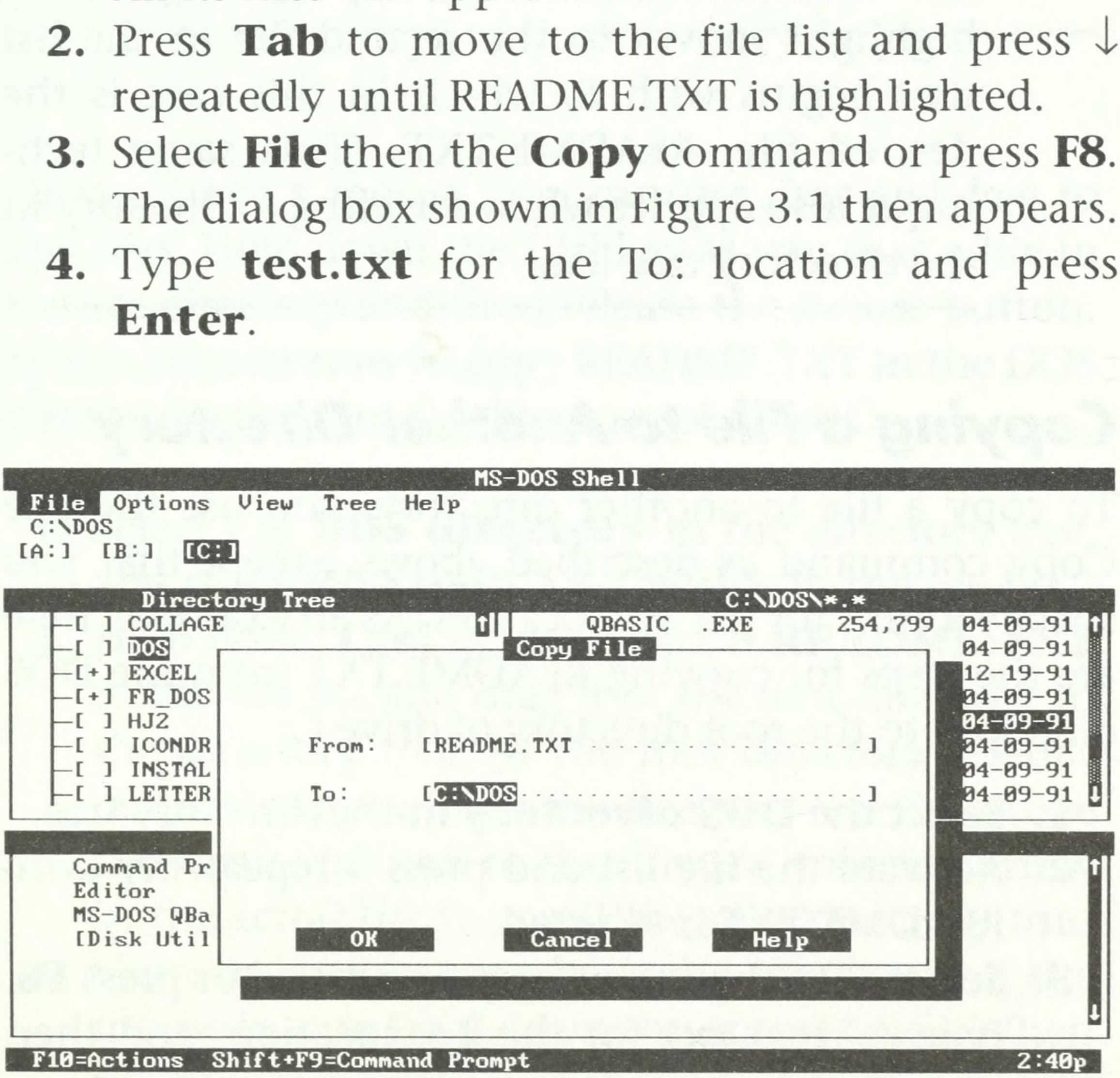



**Figure 3.1** The File Copy File dialog box

DOS makes a copy of the file and names it TEST.TXT. To see the new file in the file list, press **PgDn** until it comes into view. (In Lesson 4, you'll learn how to delete this file.)

***Tip***

A quick way to select a file in the file list is to press the first letter of its name. For example, to select README.TXT when viewing the contents of the DOS directory, simply press **r**. The highlight immediately shifts to the first file in the list that begins with that letter, which in DOS 5.0 is RAMDRIVE.SYS. If you press **r** again, the highlight moves to the second file in the list that begins with *R*, which in this case is the desired file, README.TXT. (This same technique also applies when selecting a directory in the directory tree list.)

## Copying a File to Another Directory

To copy a file to another directory, you use the File Copy command as described above, except that you specify a path for the To: location. As an example, here are the steps for copying README.TXT from the DOS directory to the root directory of drive C:

1. Select the **DOS directory** in the directory tree.
2. Activate the file list and press ↓ repeatedly until README.TXT is selected.
3. Select **File** then the **Copy** command or press **F8**, type **c:\test.txt** for the To: location, and then press **Enter**.

### Tip

When copying files to different locations on the current drive, the drive portion of the path is optional. For example, you could use \test.txt in step 3 of the previous procedure instead of c:\test.txt and still get the same results. In addition, if you don't want to rename the file in the process of copying it, you can omit the name in the To: location. For example, if you specified \ for the To: location in step 3, DOS would create README.TXT on the root directory of drive C. (See Lesson 27 for more on these and other command line shortcuts.)

If you have a mouse, you can use *drag-and-drop* to copy files. Hold down the Ctrl key as you drag a file to another directory and then release the mouse button. Try this process now to copy README.TXT in the DOS directory to the root (\) directory of drive C:

1. Select the **DOS directory** in the directory tree, and make **README.TXT** visible in the file list.
2. Hold down **Ctrl** as you click on **README.TXT** in the file list and drag it to the root directory in the directory tree. (If the root directory doesn't currently appear with the directory tree list, move the mouse pointer to the upward pointing arrow in the scroll bar to scroll the root directory into view; be sure to keep both the Ctrl key and the mouse button held down the entire time.) When the mouse pointer is at the root directory, you'll

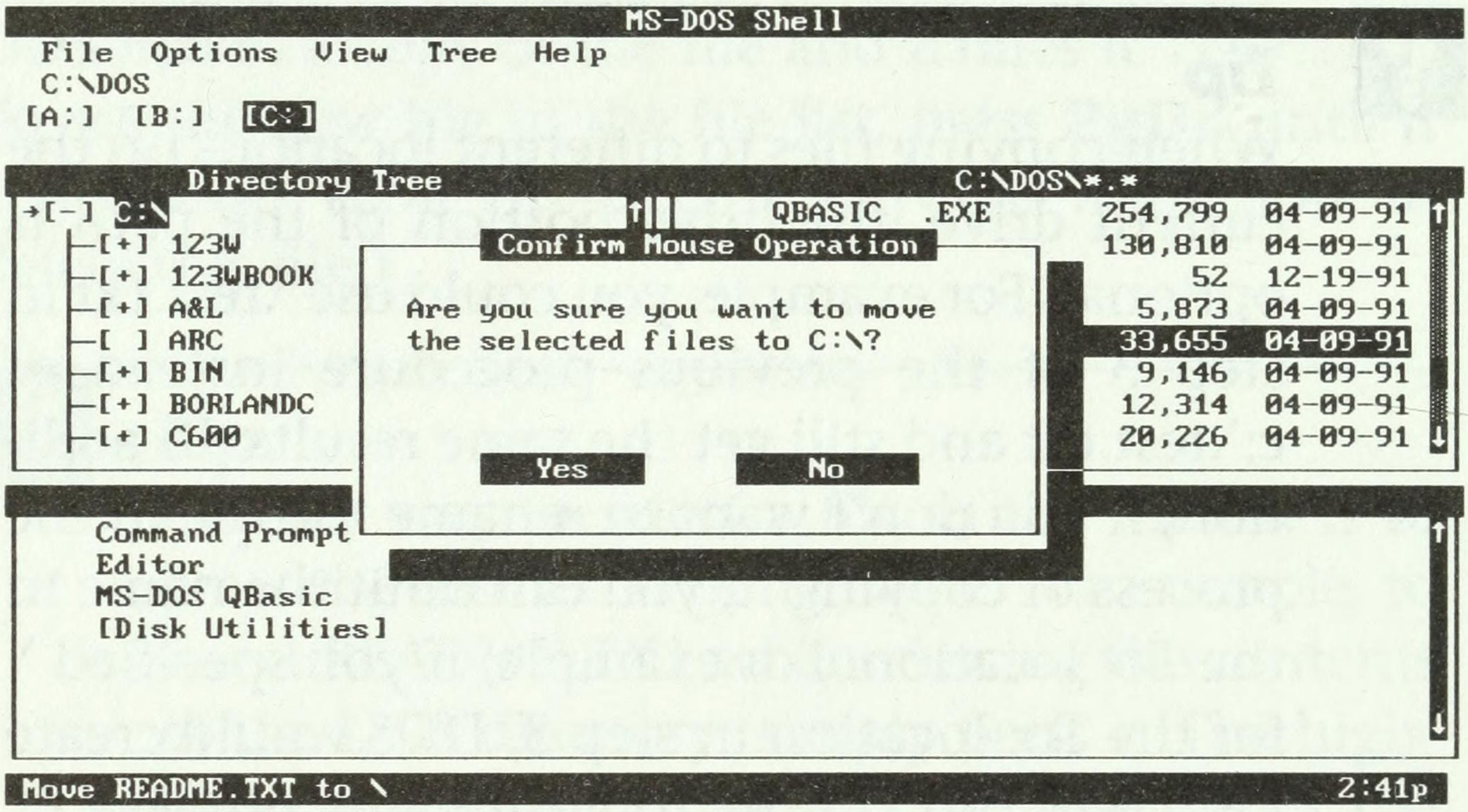



**Figure 3.2** Confirmation when moving a file

see the following message at the foot of the screen:

```
Copy README.TXT to \
```

**3.** With the mouse pointer located on the root directory, release the mouse button. Figure 3.2 shows the dialog box that appears asking you to confirm the operation.

**4.** Click on the **Yes** button.

---

***Tip***

When copying and moving files using drag-and-drop, you can save considerable time and trouble by using the Dual File Lists view, which presents two directory trees and file lists at the same time, allowing you to copy and move files between them. To access the Dual File Lists view, use the View Dual File Lists command (see Lesson 7).

---

⇨ ***Note:*** Using drag-and-drop to copy files only works when you want to copy a file to another directory (or drive) and keep the same filename.

## *Copying Files between Drives*

To copy a file from one drive to another using the keyboard, you must specify the target drive and path for the File Copy command's To: location. Here are the steps for copying README.TXT in C:\DOS to the root directory of drive A. (You'll need a formatted floppy disk—see Lesson 14—to try this example.)

1. Select the **DOS directory** in the directory tree and then select **README.TXT** in the file list.
2. Choose the **File Copy** command or press **F8**.
3. Type **a:**\ for the To: location and press **Enter**.

To see the copy of README.TXT on drive A, move to the drive-icons area and make drive A the current drive.

If you have a mouse, you can perform this same example using drag-and-drop. Here are the steps:

1. Select the **DOS directory** in the directory tree and make **README.TXT** visible in the file list.
2. Click on **README.TXT** in the file list and drag it to the **drive A** icon in the drive-icon area above. (Holding down the Ctrl key isn't necessary when copying files between drives.)
3. Release the mouse button.
4. Select **Yes** (press **Enter**) in response to the confirmation message.

◆ *Lesson* ◆

# 4

# More File Operations

## Moving Files

To move a file using the keyboard, you use the File Move command or its shortcut, F7. For example, suppose you want to move the file APPNOTES.TXT from the DOS directory to the root directory of drive C. Follow these steps:

1. Select the **DOS directory** in the directory tree and then **APPNOTES.TXT** in the file list.
2. Choose **File** then the **Move** command. Figure 4.1 shows the dialog box that appears.
3. Type **c:** for the To: location and press **Enter**.

APPNOTES.TXT is removed from C:\DOS and now appears in C:\. To see the file in its new location, select **C:** in the directory list (press **Home** after moving to the directory-list area).

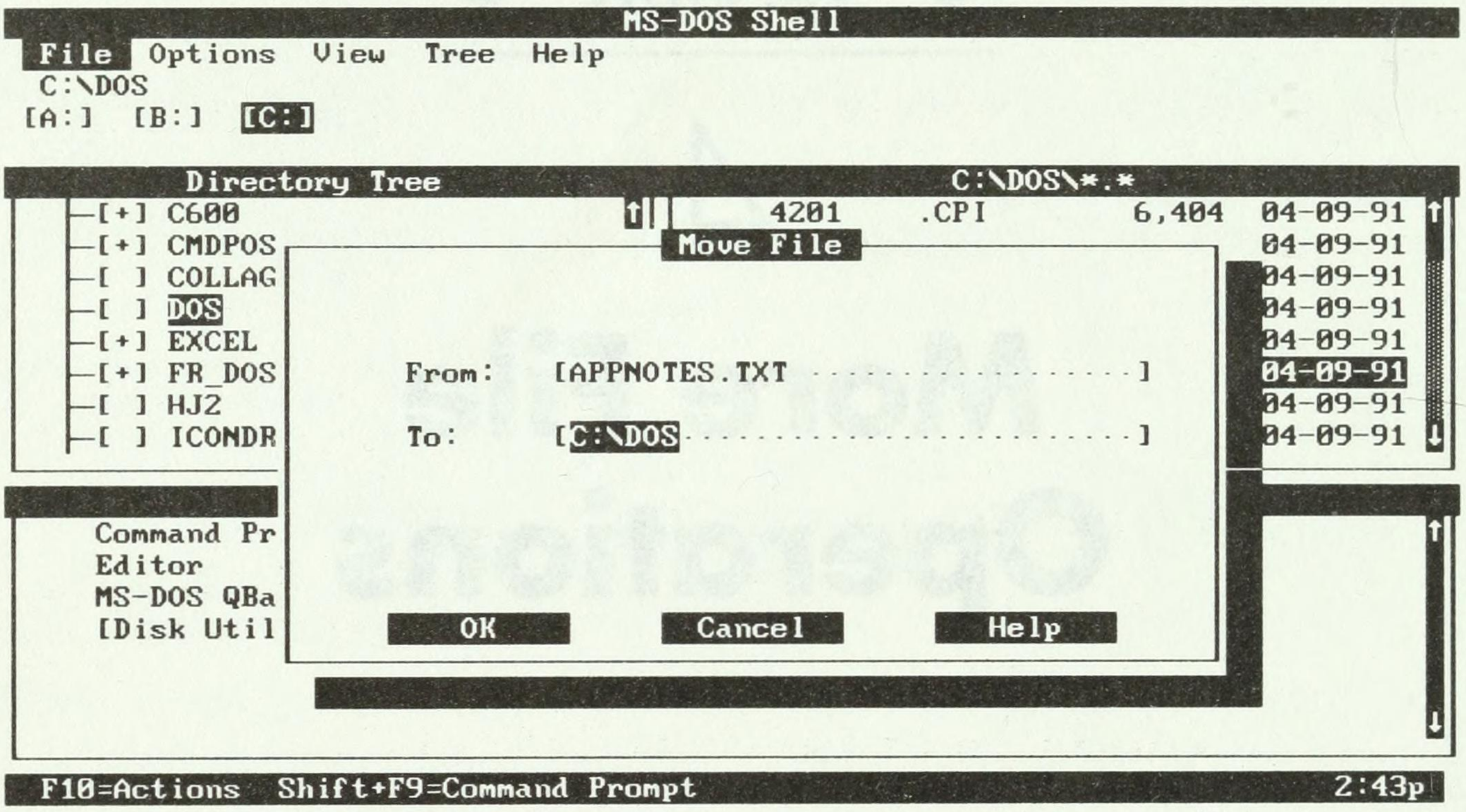



**Figure 4.1** The File Move File dialog box

To return APPNOTES.TXT to its original location in the DOS directory:

1. Select **APPNOTES.TXT** in the file list for C:\.
2. Choose **File** then the **Move** command, type **c:\dos** for the To: location, and press **Enter**.

---

***Tip***

You can also use File Move to move a file to another drive. Just enter the new drive and target path in the To: location. For example, if you used d:\backup for the To: location in the previous procedure, DOS would copy APPNOTES.TXT to the BACKUP directory on drive D.

---

To move a file using the mouse, drag it from the file list and drop it in a new directory in the directory tree. To move a file to a new drive, use the same technique

but hold down the Alt key as you drag the file and drop it on a drive icon. (If you don't use the Alt key, the DOS Shell assumes you want to copy the file to the new drive instead of moving it.)

### *Tip*

Just as when you're copying files, using the Dual File Lists view is handy when moving files because it lets you see the contents of two directories at the same time (see Lesson 7).

## *Duplicate Filenames*

Imagine that when you were moving APPNOTES.TXT to the root directory of drive C, a file with that same name already existed in that location. You'd then see a confirmation dialog box like the one in Figure 4.2 asking whether you want to overwrite the existing file

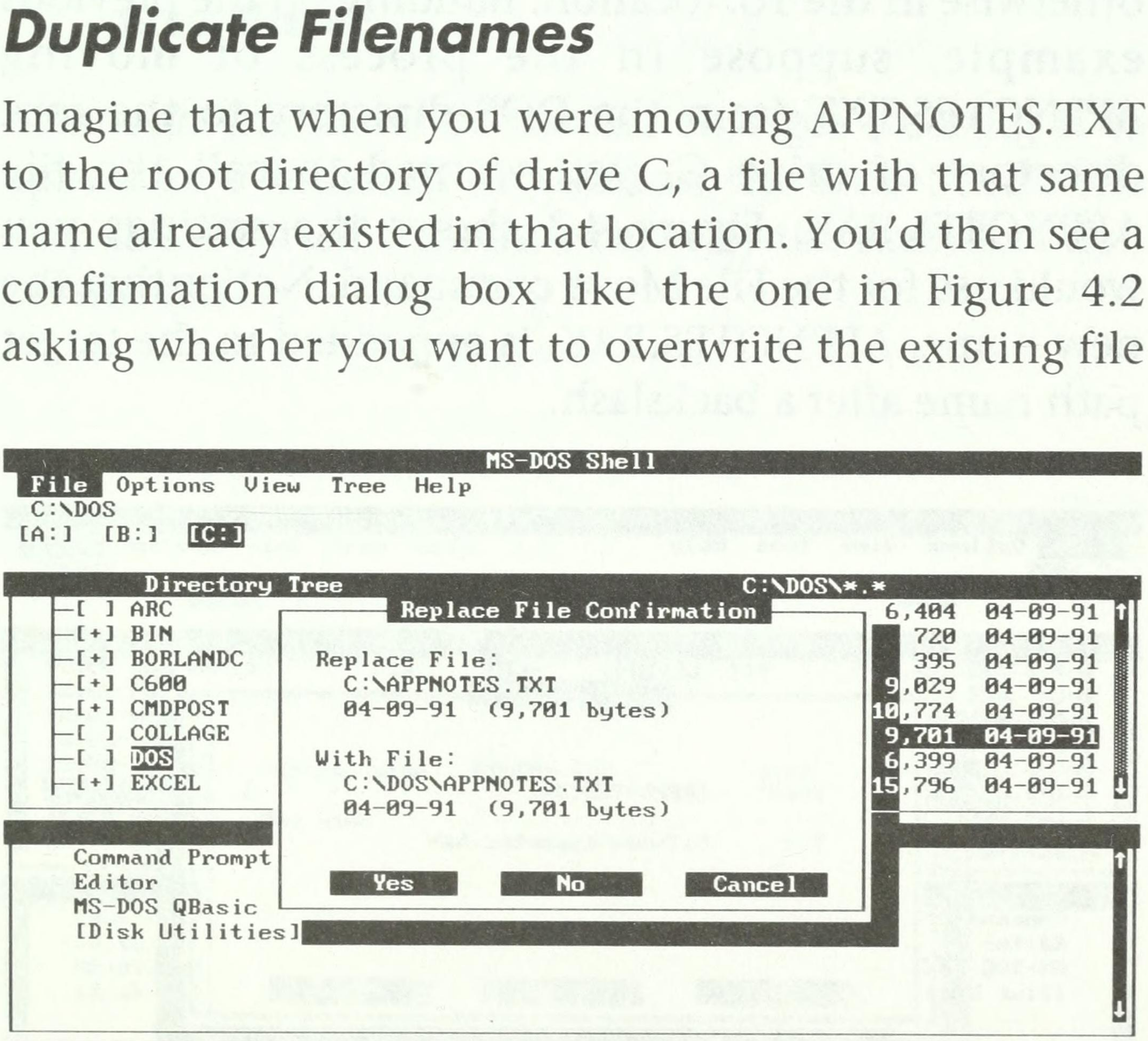



**Figure 4.2** The dialog box when overwriting an existing file while moving

with the one you're moving. At this point you must either select Yes, which replaces the file; or No or Cancel, both of which leave the original file intact and abandon the File Copy operation.

# Renaming Files

## *Renaming a File As You Move It*

When you use the File Move command to move a file into a different directory, the file you move will have the same name in the new directory, unless you specify otherwise in the To: location. Building on the previous example, suppose in the process of moving APPNOTES.TXT from the DOS directory to the root directory of drive C, you wanted to call the file APPNOTES.BAK. Figure 4.3 shows the settings you would use for the File Move command. Notice that the new name, APPNOTES.BAK, is appended to the target path name after a backslash.

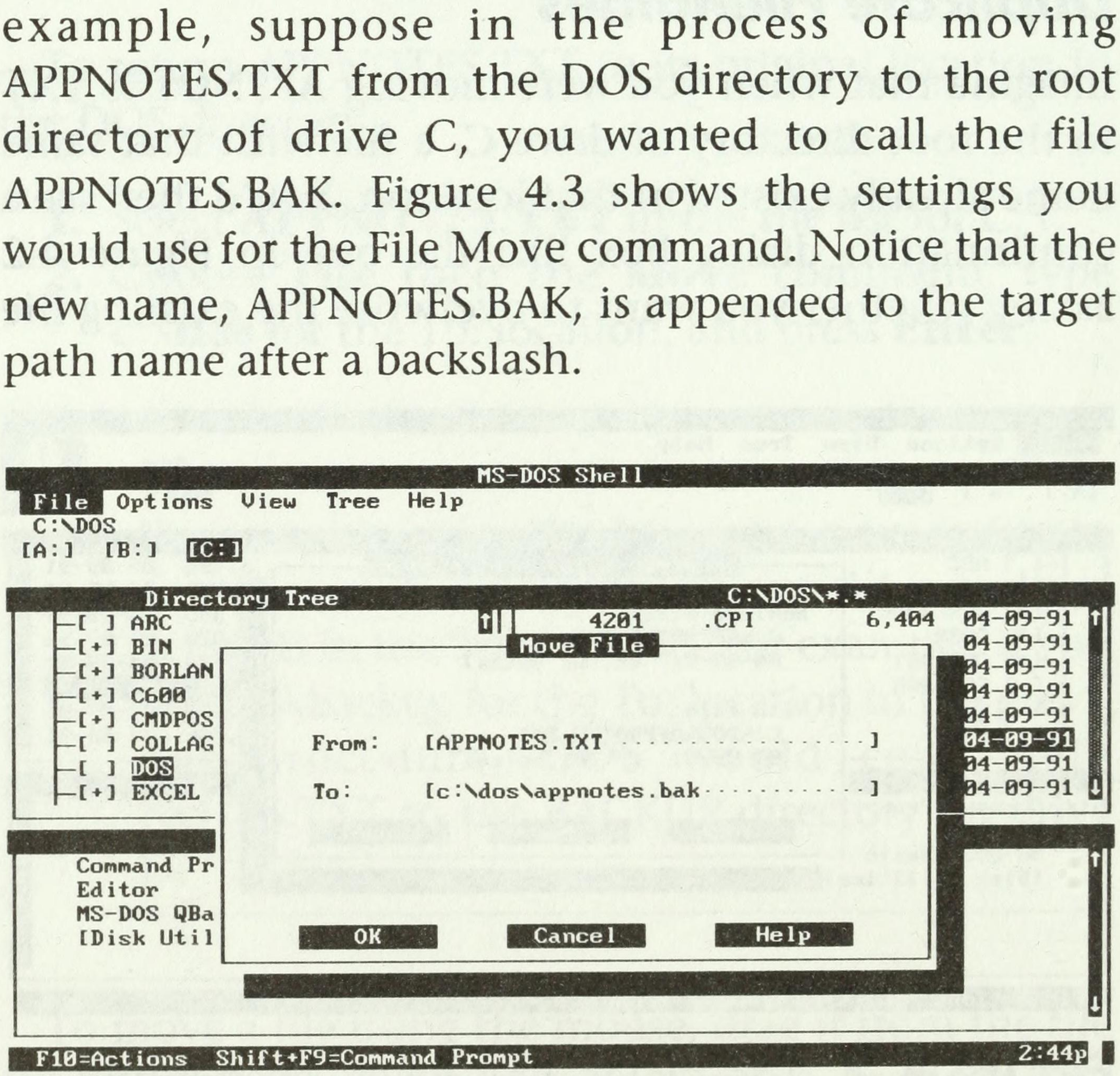



**Figure 4.3** Renaming APPNOTES.TXT

## Renaming a File in Its Current Location

To rename a file in its current location, you use the File Rename command. For example, to rename the file README.TXT in the DOS directory to DOS5READ.ME use these steps:

1. Select the **DOS directory** in the directory tree and then select **README.TXT** in the file list.
2. Choose **File** then the **Rename** command. Figure 4.4 shows the dialog box that appears.
3. Type **DOS5READ.ME** for the new name and press **Enter**.

Although the name of the file changes (causing it to appear in a different location in the DOS Shell's file list), the file itself remains in exactly the same location on the disk.

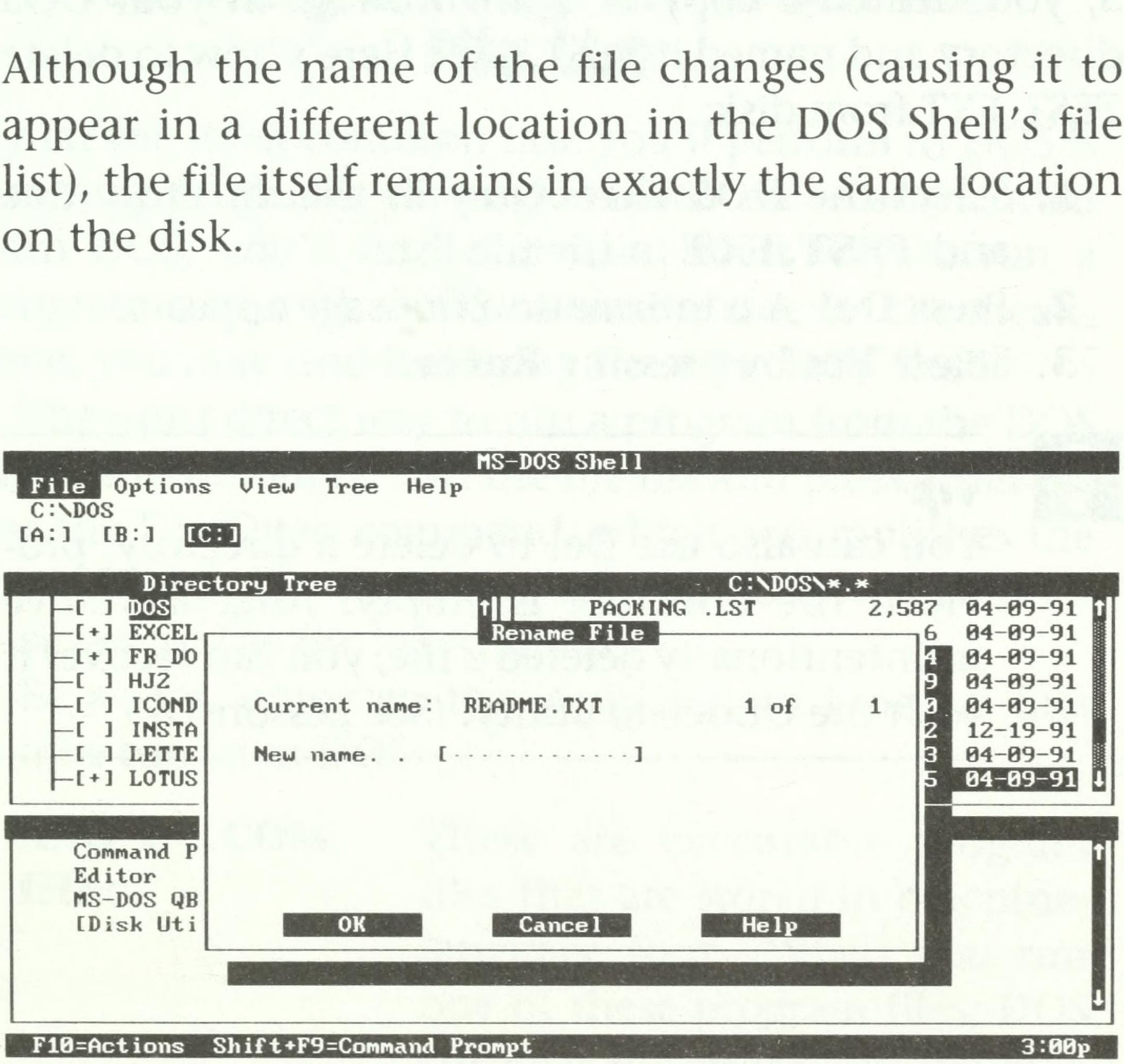



**Figure 4.4** The File Rename File dialog box

The File Rename command won't let you overwrite an existing file. Therefore, if a file already exists with the new name you've chosen, you'll see the error message "Access denied." If you find this a nuisance and want to overwrite the file anyway, use the File Move command to rename the file instead.

## Deleting Files

When you no longer need a file, you can remove it from disk using the File Delete command or its shortcut, Del. For example, if you followed along in Lesson 3, you created a copy of README.TXT in your DOS directory and named it TEST.TXT. Here's how to delete TEST.TXT from disk:

1. Select the **DOS directory** in the directory tree and **TEST.TXT** in the file list.
2. Press **Del**. A confirmation message appears.
3. Select **Yes** by pressing **Enter**.

***Tip***

You can also use Del to delete a directory, provided the directory is empty. And, if you've unintentionally deleted a file, you can recover it with the Undelete utility. (See Lesson 16.)

◆ *Lesson* ◆

# 5

# Running Programs

## Running a Program from the File List

By far the most common task you'll perform in DOS is running programs. As you become more acquainted with DOS, you'll find that the easiest way to run a program is from the DOS command line; in the meantime, you may find it simpler from the DOS Shell.

The most direct way to run a program from the DOS Shell is to highlight it in the file list and press Enter (or use the File Open command, which accomplishes the same thing). If you have a mouse, double-click on a program in the file list to run it.

Be aware, however, that only certain types of files can be run from DOS:

**.EXE or .COM files** These are executable program files that are stored in machine-language form. When you run one of these program files, DOS copies it into RAM and the microprocessor executes it.

**.BAT files** These are batch files, a special form of text file that DOS reads and interprets. Batch files are used primarily to run executable program files in succession—for example, a screen capture program followed by a spreadsheet program. But as you'll learn in Lesson 30, you can use them for much more than that.

The DOS Shell also has the unique ability to run two other types of files, which can't be run from the DOS command line.

**.TXT files** These are ASCII (American Standard Code for Information Interchange) text files that are automatically loaded into the MS-DOS Editor for viewing and changing (see Lesson 13).

**.BAS files** These are QBasic program files. DOS 5.0 comes with four programs written in QBasic (GORILLA.BAS, MONEY.BAS, NIBBLES.BAS, and REMLINE.BAS) that you can execute directly from the DOS Shell.

***Tip***

The only reason that you can "run" .TXT and .BAS files from the DOS Shell is that file associations have been created for them. You can create your own file associations using the File Associate command (see Lesson 11).

To get a feel for how to run a program from the DOS Shell, try this example:

1. Select the **DOS directory** in the directory tree.
2. Choose **HELP.EXE** in the file list and press **Enter**. After a slight pause, you'll see the screen in Figure 5.1, which is the first of four screens giving brief descriptions of all the available DOS commands.
3. Press **Enter** (or any other key) four times to view all the available screens. When HELP.EXE has run its course, you'll see the following message:

```
Press any key to return to MS-DOS Shell....
```

4. Press **Enter** (or any other key) again to return to the DOS Shell.

```
For more information on a specific command, type HELP command-name.
APPEND   Allows programs to open data files in specified directories as if
         they were in the current directory.
ASSIGN   Redirects requests for disk operations on one drive to a different
         drive.
ATTRIB   Displays or changes file attributes.
BACKUP   Backs up one or more files from one disk to another.
BREAK    Sets or clears extended CTRL+C checking.
CALL     Calls one batch program from another.
CD       Displays the name of or changes the current directory.
CHCP     Displays or sets the active code page number.
CHDIR    Displays the name of or changes the current directory.
CHKDSK   Checks a disk and displays a status report.
CLS      Clears the screen.
COMMAND  Starts a new instance of the MS-DOS command interpreter.
COMP     Compares the contents of two files or sets of files.
COPY     Copies one or more files to another location.
CTTY     Changes the terminal device used to control your system.
DATE     Displays or sets the date.
DEBUG    Runs Debug, a program testing and editing tool.
DEL      Deletes one or more files.
DIR      Displays a list of files and subdirectories in a directory.
---More---
```

**Figure 5.1** The initial HELP.EXE screen

# Running a Program with the File Run Command

Another way to run a program from the DOS Shell is to use the File Run command. This command is useful for two reasons: First, you don't have to highlight a program in the file list in order to run it, and you can include command-line parameters after the program's filename.

For example, suppose you want to run HELP.EXE without going to the trouble of highlighting it in the DOS directory. Here are the steps:

1. Select **File** then the **Run** command. You'll then see the dialog box in Figure 5.2.
2. Type the name of the program preceded by its path and a \—in this case, **c:\dos\help**—and press **Enter**. You'll then see Help's initial screen, as shown in Figure 5.1.

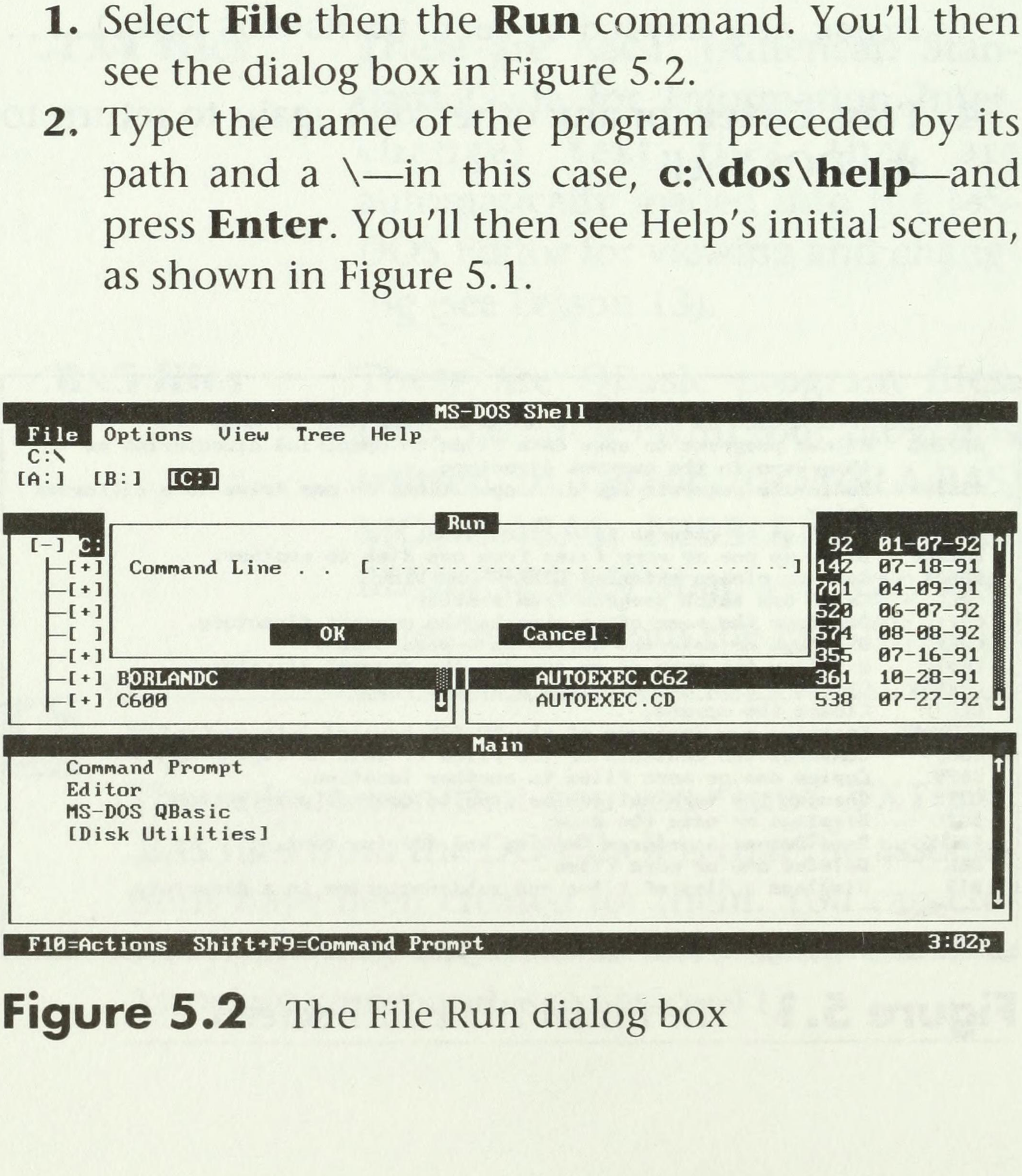



**Figure 5.2** The File Run dialog box

3. Press **Enter** (or any other key) repeatedly to see all the available screens, and finally, return to the DOS Shell.



***Tip***

When running an .EXE, .COM, or .BAT file in DOS, you don't need to include its file extension. That's why you were able to type c:\dos\help instead of c:\dos\help.exe in the previous example.

The Help program is also capable of giving you more detailed information about individual DOS commands. To see this information for the Undelete command (UNDELETE.EXE in the DOS directory), you must include its name as a parameter following Help's path and program name. Here are the steps:

1. Select **File** then the **Run** command.
2. Type **c:\dos\help undelete** and press **Enter**. Figure 5.3 shows the screen that appears.
3. Press **Enter** (or any other key) to return to the DOS Shell.

```
Restores files which have been deleted.

UNDELETE [[drive:][path]][filename] [/LIST ¦ /ALL] [/DT ¦ /DOS]

  /LIST  Lists the deleted files available to be recovered.
  /ALL   Undeletes all specified files without prompting.
  /DT    Uses only the deletion-tracking file.
  /DOS   Uses only the MS-DOS directory.

MIRROR, UNDELETE, and UNFORMAT Copyright (C) 1987-1991 Central Point Software,
Inc.


Press any key to return to MS-DOS Shell....
```

**Figure 5.3** The Undelete command help screen

## When File Run Doesn't Require a Path

When starting a program with the File Run command, there are two cases when you don't need to precede the program's filename with its path:

- When the program is in the current directory,
- When the program's directory is included in your search path. The *search path* is a list of directories that DOS searches when it can't find the program you're trying to run in the current directory. And because the DOS directory is almost always included in your search path, you don't usually need to provide any path information when running DOS commands. For example, you could type **help undelete** in Step 2 of the previous example, and DOS would have no trouble finding HELP.EXE in the DOS directory and executing it. (See Lesson 32 for more information on the search path and how to modify it.)

## Running a Program Included in a Group

Another way to run a program from the DOS Shell is to select its name in the program-list area. To run the MS-DOS Editor using the keyboard, for example:

1. Press **Tab** to move to the program-list area.
2. Highlight **Editor** in the list and press **Enter**. Figure 5.4 shows the dialog box that appears requesting the name of a text file to edit.
3. Type **c:\dos\readme.txt** and press **Enter**. You'll then see the MS-DOS Editor screen with

the README.TXT file from the DOS directory displayed and ready for viewing or editing.

4. Use **PgDn** and **PgUp** to view the file.
5. To leave the MS-DOS Editor and return to the DOS Shell, select **File** then the **Exit** command.

To run a program using the mouse, double-click on its name in the program-list area.

Table 5.1 shows the names of the programs that are available in the program-list area and their corresponding program files. (All the program files are located in the DOS directory.) The program names that are listed in the table are actually part of what is known as the Main program group.

***Tip***

If you select "Command Prompt" in the program-list area, you'll temporarily leave the DOS Shell. To return to the DOS Shell, type exit and press Enter.

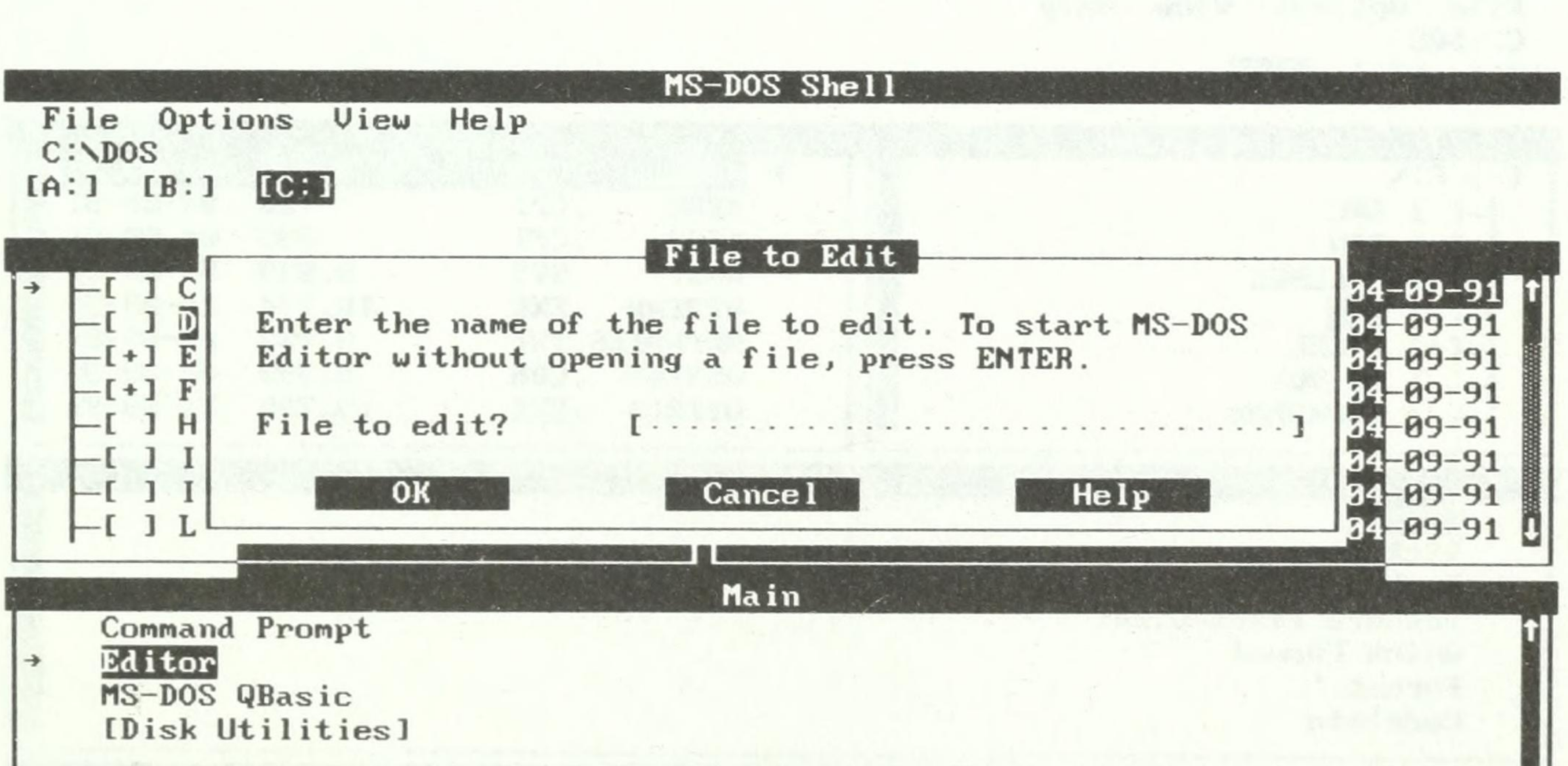



**Figure 5.4** Running the MS-DOS Editor

**Table 5.1 Main Program Group Programs**

| Name | Program File |
|---|---|
| Command Prompt | COMMAND.COM |
| Edit | EDIT.COM |
| MS-DOS QBasic | QBASIC.EXE |

You can access a second program group—the Disk Utilities group—by highlighting **[Disk Utilities]** in the program list and pressing **Enter**. Figure 5.5 shows the new program names that appear in the program list. Each of these names also corresponds to a DOS utility, as shown in Table 5.2. To return to the Main program group, highlight **[Main]** at the top of the program list and press **Enter**.

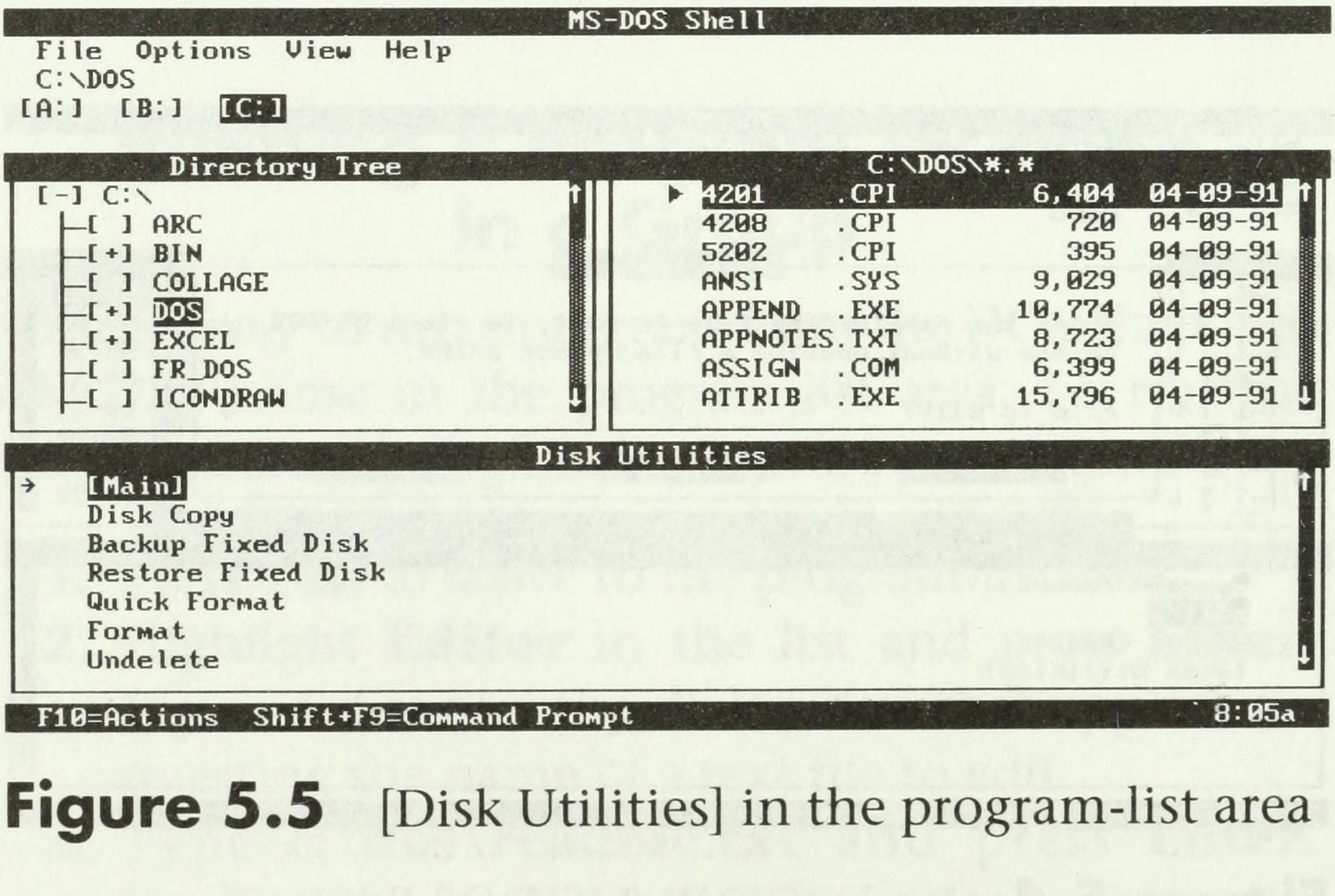



**Figure 5.5** [Disk Utilities] in the program-list area

## Table 5.2 Disk Utilities Group Programs

| Name | Program File |
|---|---|
| Disk Copy | DISKCOPY.COM |
| Backup Fixed Disk | BACKUP.EXE |
| Restore Fixed Disk | RESTORE.EXE |
| Quick Format | FORMAT.COM (with the /q switch) |
| Format | FORMAT.COM |
| Undelete | UNDELETE.EXE |

Lesson 10 will explain how to add your own programs to the Main and Disk Utilities program groups. This lesson will also teach you how to create your own program groups.

◆ *Lesson* ◆

6

# Selecting Multiple Files for a Command

## Methods of Selection

Occasionally, you'll want to select a group of files from the file list before applying a command to them. You might do this, for instance, before copying a group of files from one directory to another.

The DOS Shell provides three main ways to select files using the keyboard or the mouse. You can:

- select consecutive files,
- select nonconsecutive files,
- select all the files in the current directory.

## *Selecting Consecutive Files*

To select a group of consecutive files—that is, files that are listed next to one another in the file list—hold down the Shift key as you press ↓ or ↑ to extend the selection.

For example, suppose you want to select all the files in the DOS directory that begin with the letter S. Perform the following:

1. Select the **DOS directory** in the directory tree.
2. Move to the file list and press ↓ repeatedly until the first file that begins with the letter S—SETVER.EXE—comes into view. (A quick alternative is to press **s** to move directly to the file.)
3. Press **Shift** and hold it down as you press ↓ five times to highlight all the files that begin with S. (If you go too far, just press ↑ while you continue to hold **Shift** down.) Figure 6.1 shows how your screen should appear.

If you were to select the File Copy command now, you'd see that all the files were listed one after the other in the From: location, each separated by a space. All you would need to do to copy the files to another directory would be to provide the name of the directory for the To: location; for example, **c:**.

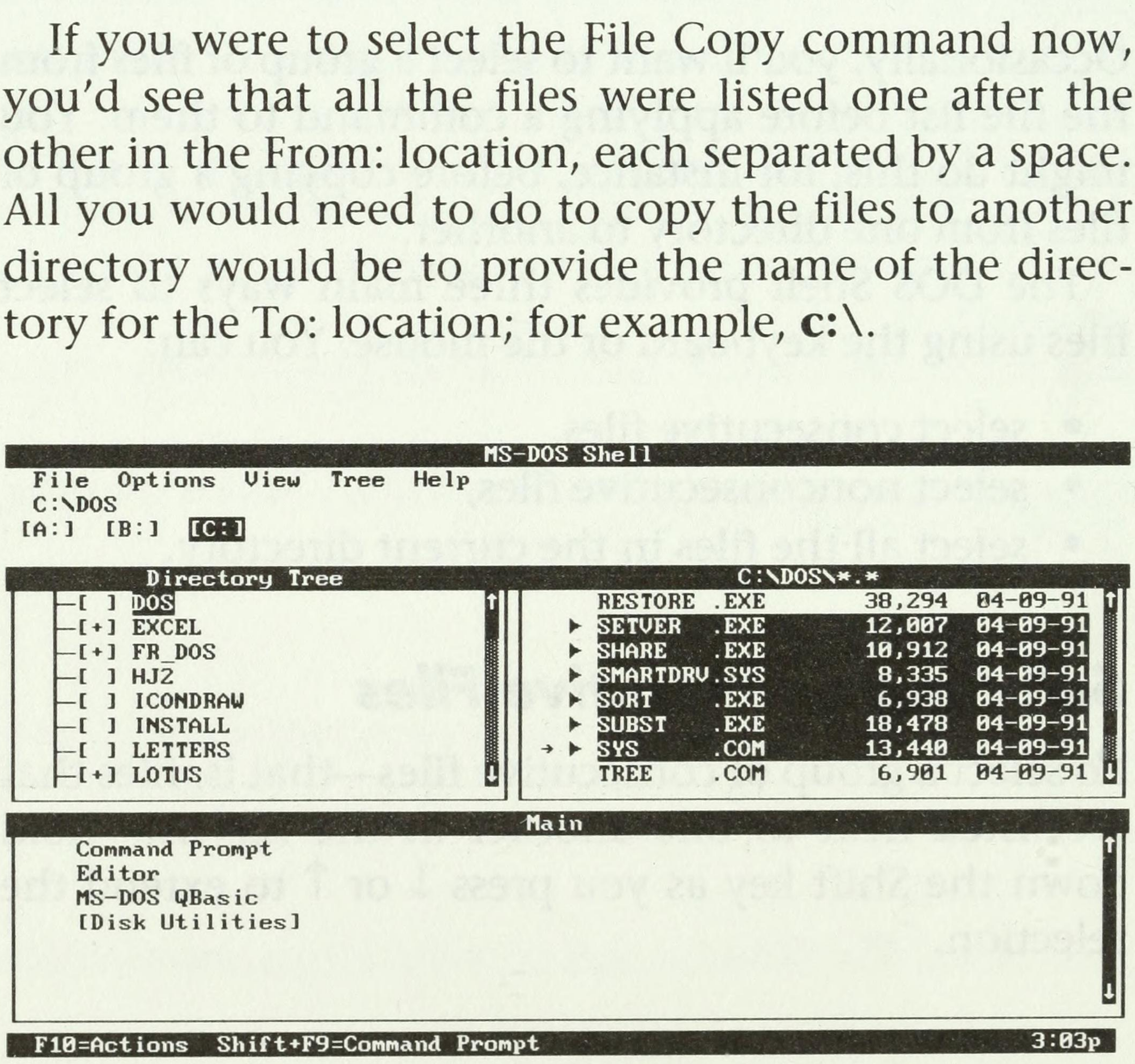



**Figure 6.1** Selecting consecutive files using Shift

**Tip**

After highlighting a group of consecutive files with Shift, you can eliminate the selection by releasing the Shift key and using ↑ or ↓ to move the highlight to another file. (You can also use Ctrl+\ to eliminate the current selection.)

To select a group of consecutive files using the mouse, start by clicking on the first file you want included in the group—for example, SETVER.EXE in Figure 6.1. Then hold down the Shift key as you click on the last file you want in the group—for example, SYS.COM. All the files between (and including) the first and last files you clicked on will immediately appear highlighted.

## Selecting Nonconsecutive Files

Besides selecting files that are listed consecutively in the file list, you can also select nonconsecutive files. To do so using the keyboard, you must use a combination of Shift+F8 and the Spacebar. Let's say you want to select all the same files as in Figure 6.1, except SMARTDRV.SYS. Here are the steps:

1. With the **DOS directory** selected in the directory tree, move to the file list and press **s** to select the first file that begins with the letter S, SETVER.EXE.
2. Press **Shift** and hold it down as you press ↓ once to extend the selection to SHARE.EXE.
3. Press **Shift+F8** to turn **Add** mode on. When Add mode is on, the word "ADD" is displayed in the

lower right-hand corner of the MS-DOS Shell window.

4. Press ↓ twice to move down to SORT.EXE.
5. Press **Spacebar** to include SORT.EXE.
6. Press **Shift** and hold it down as you press ↓ twice to extend the selection to SYS.COM. Figure 6.2 shows the results.
7. Press **Shift+F8** to turn **Add** mode off. The ADD indicator disappears from the lower right-hand corner of the screen.

***Tip***

When Add mode is on, the Spacebar acts as a toggle key. This means you can use the Spacebar to add the highlighted file to or remove it from the current selection. An arrowhead appears to the left of the filename when it is part of the current selection.

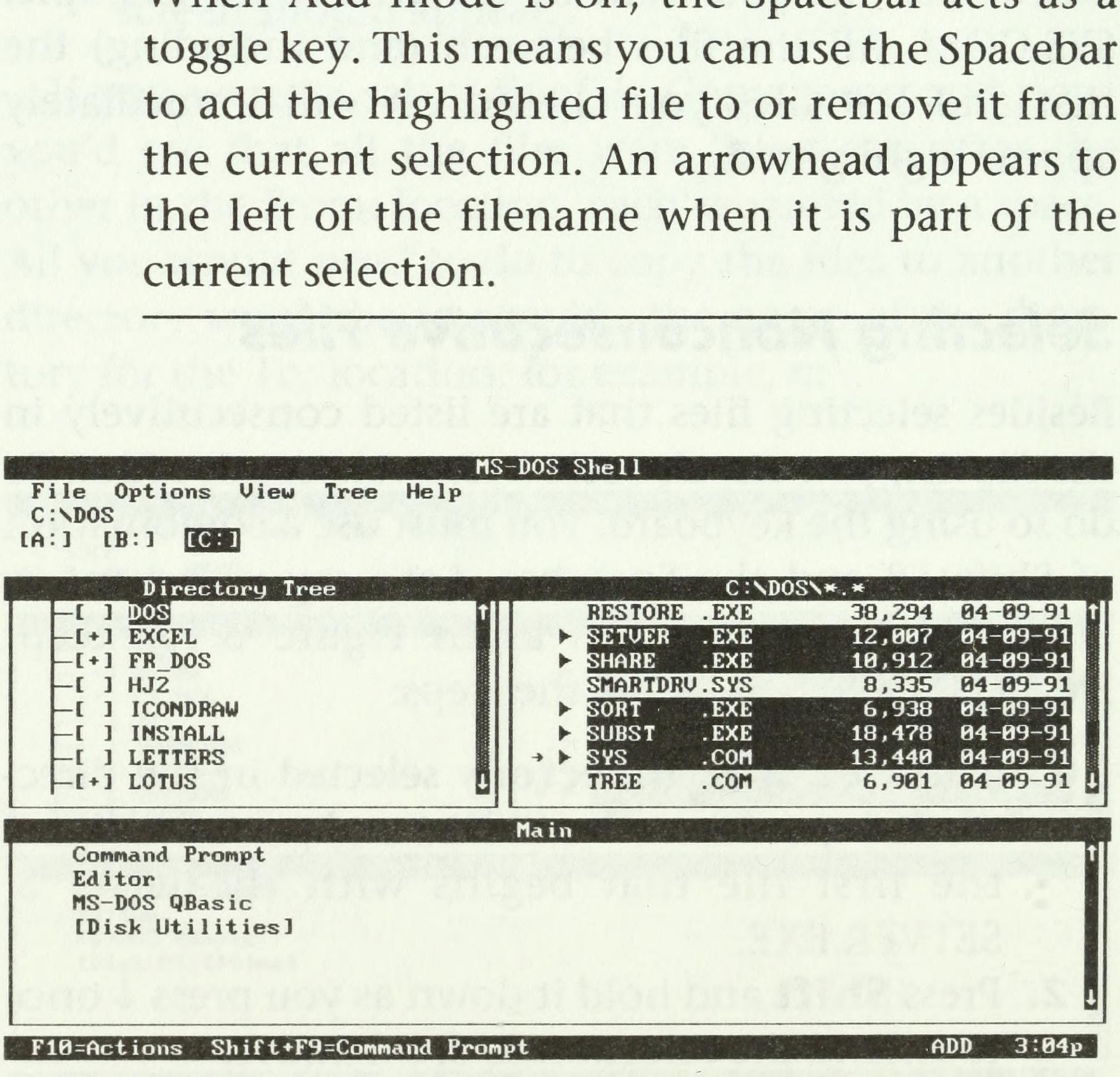



**Figure 6.2** Selecting nonconsecutive files

To select nonconsecutive files using the mouse, start by clicking on the first filename you want selected. Then press **Ctrl** as you click on each additional file you want included in the selection. To remove a file from the current selection, click on it again.

## Selecting All the Files in a Directory

The quickest way to select all the files in the file list is to press **Ctrl+/**, or use **File** then the **Select All** command. If you want to cancel the current file selection, press **Ctrl+**, or choose **File** then the **Deselect All** command.

◆ *Lesson* ◆

# 7

# Different Ways of Viewing Files and Directories

## Changing the DOS Shell Screen Arrangement

When you first load the DOS Shell, its directory tree, file list, and program list arrangement seems cast in concrete. In reality, though, you can easily change the arrangement of these areas using the View menu. Commands in this menu let you display a single file list, two file lists, or one all-encompassing alphabetical file list that ignores directories. In addition, you can also display a list of programs and directories or a list of programs only.

### *Single File List View*

With the Single File List view, the screen is divided in two with the directory tree on the left and the file list on the right. Figure 7.1 shows an example of this view. In order to access it, you use the **View Single File List** command.

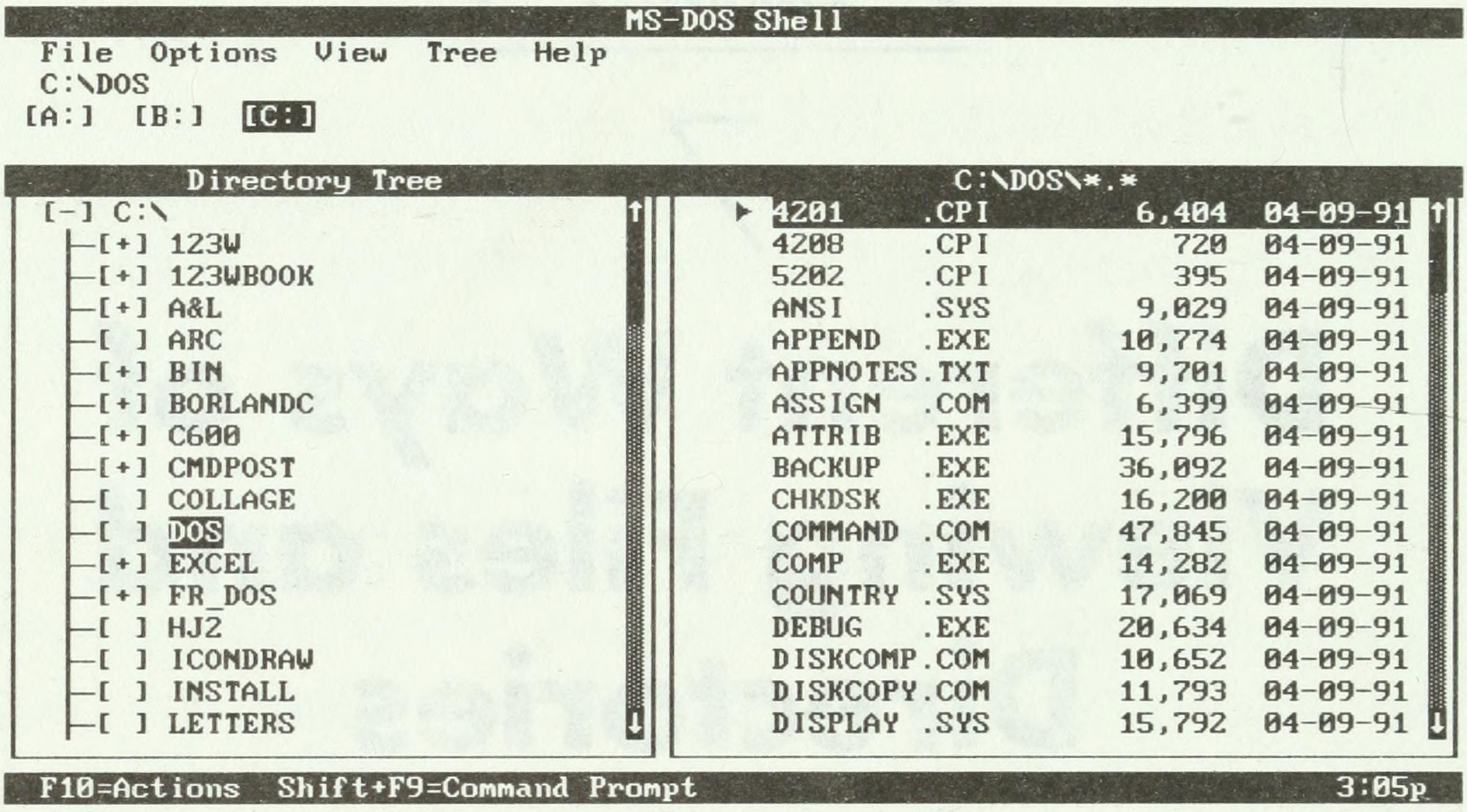
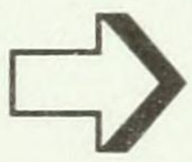



**Figure 7.1** The Single File List view



***Note:*** When you change the DOS Shell view setting, it saves the new setting to disk so that it can configure itself the same way the next time you start the program.

## *Dual File Lists View*

When copying and moving files with the mouse, it's often handy to see two directory trees and file lists at the same time. That way, you can more easily drag-and-drop files between them. To see this type of view, you use the **View Dual File Lists** command. Figure 7.2 shows an example.

***Tip***

After using the View Dual File Lists command for the first time during a session, you may need to activate the second (lower) directory tree or file list to have them show the proper contents.

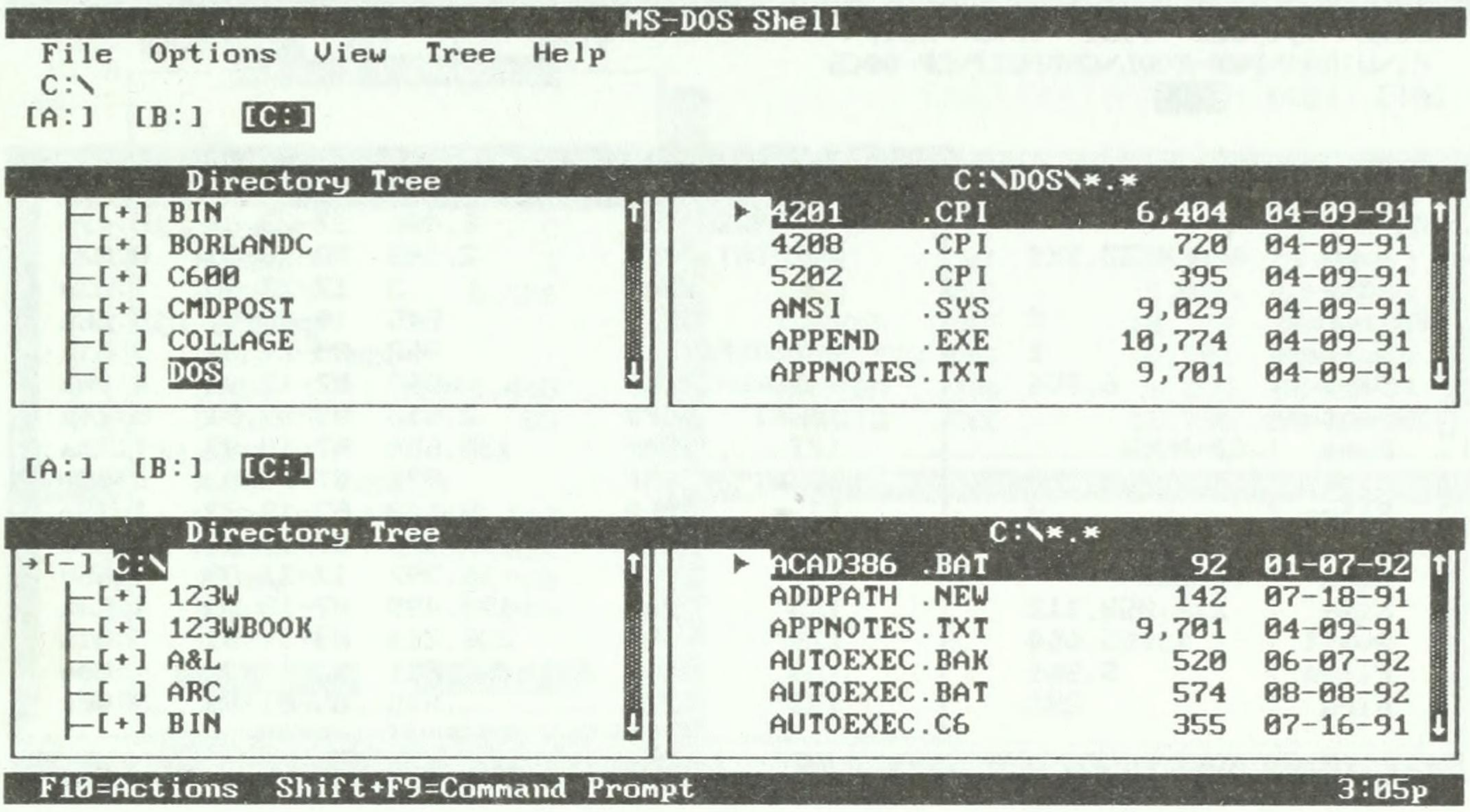
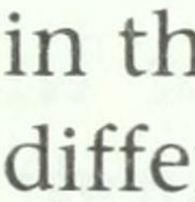



**Figure 7.2** The Dual File Lists view

## All Files View

By using the **View All Files** command, you can have the DOS Shell show one all-encompassing list of files from the current drive. The files are sorted alphabetically, regardless of their directories. Figure 7.3 shows an example of this view. Notice that the first two files in the list have the same filename. They are located in different directories, however. (The directory for the currently selected file, C:\WIN31\PWR-TOOL\CMDPOST\CP-DOCS, is shown below the menu bar at the top of the screen. This line is updated with the appropriate directory information whenever you highlight a new file.)



### Tip

The All Files view is the only way to see the time that a file was created from within the DOS Shell. The time appears in the farthest right column of the screen.

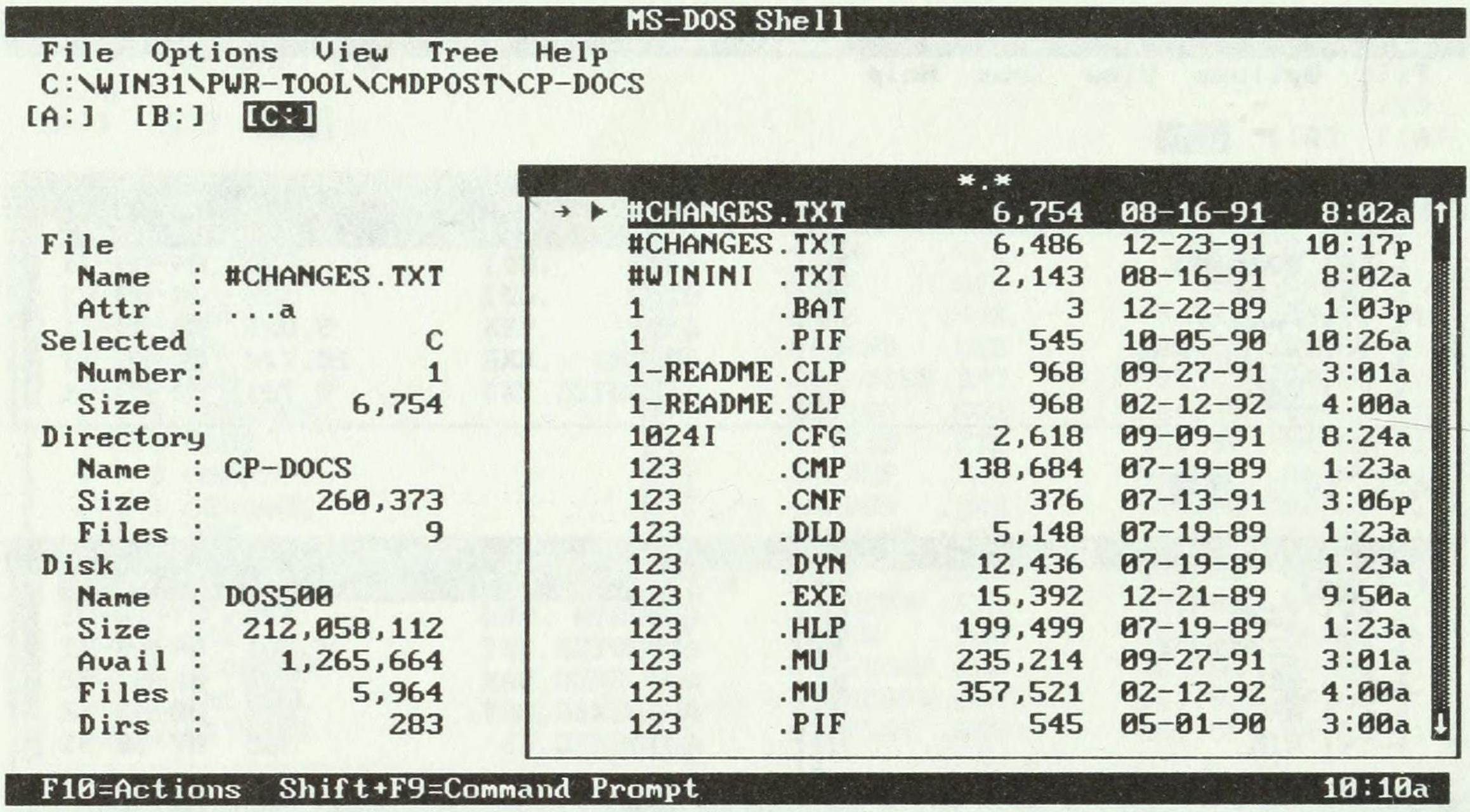
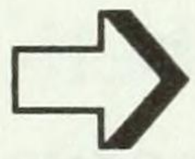



**Figure 7.3** The All Files view

The left side of the screen also shows an assortment of file, directory, and disk drive information. Table 7.1 gives an explanation of the settings.



***Note:*** The file and disk size information in the All Files view is reported in bytes, where a byte is approximately one character. You're bound to see files and disk sizes reported elsewhere in kilobytes or megabytes, so here's the conversion:

1 kilobyte (K) = 1,024 bytes
1 megabyte (MB) = 1,024 kilobytes or
1,048,576 bytes

## VIEWING FILE AND DISK INFORMATION

Another way to see file and disk information is to select **Options** then the **Show Information** command. (This command is not available in the All Files view.)

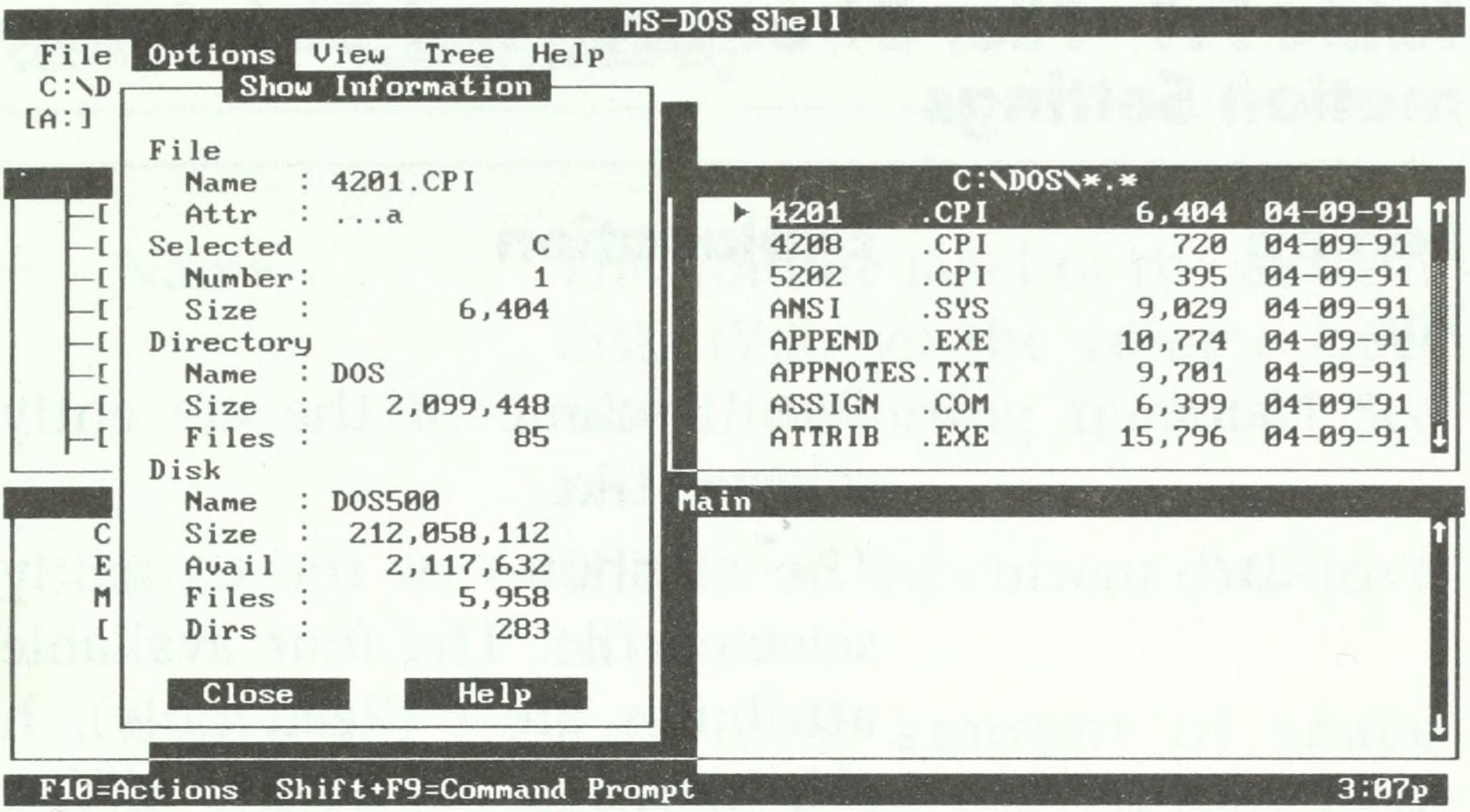



**Figure 7.4** Options Show Information dialog box

Figure 7.4 shows the dialog box that appears. Notice that it includes all the same settings as the left side of the All Files view shown in Figure 7.3.

## *Program/File Lists View*

The Program/File Lists View is the default arrangement of the directory-tree, file-list, and program-list areas that you've seen throughout the first part of this book. If you've changed to an alternate view—the Dual File Lists view, for example—you can change back to the default arrangement by selecting the **View Program/File Lists** command.

## *Program List View*

The Program List view displays a single program list, as shown in Figure 7.5. To see this view, choose the **View Program List** command.

## Table 7.1 File, Directory, and Disk Information Settings

| Setting | | Explanation |
|---|---|---|
| File | | |
| | Name | The filename of the currently selected file. |
| | Attr | The attributes of the currently selected file. The four available attributes are r (Read only), h (Hidden), s (System), and a (Archive). If an attribute is not set, it appears as a . (period). (See Lesson 12 for more information in file attributes.) |
| Selected | | |
| | Number | The number of files that are currently selected in the file list. |
| | Size | The cumulative size (in bytes) of the currently selected files. You might check this setting, for example, to determine whether the selected files will all fit on a floppy disk at once. |
| Directory | | |
| | Name | The subdirectory in which the currently selected file resides. |
| | Size | The size (in bytes) of all the files in the named subdirectory. |
| | Files | The number of files in the named subdirectory. |

***(continued)***

**Table 7.1 *(continued)***

| | | |
|---|---|---|
| Disk | | |
| | Name | The volume label of the current disk. (You set the volume label when formatting the disk. See Lesson 14.) |
| | Size | The size of the current disk drive in bytes. |
| | Avail | The total amount of space available on the current disk drive in bytes. |
| | Files | The total number of files on the current disk drive. |
| | Dirs | The total number of directories on the current disk drive. |

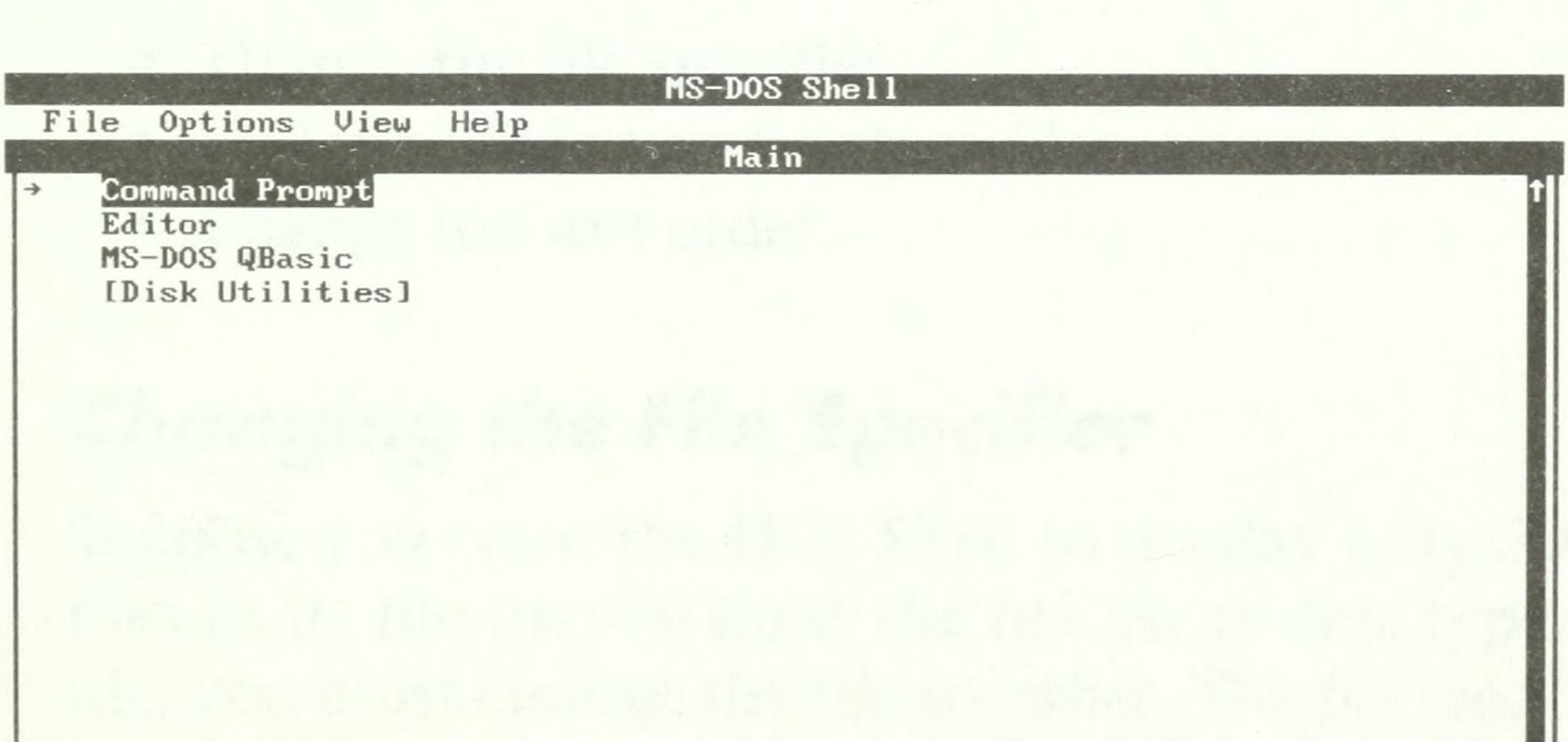



**Figure 7.5** The Program List View

◆ *Lesson* ◆

8

# Controlling the File List

## Changing File Display Options

Normally, the DOS Shell file list shows all the files in the current directory sorted by name in ascending order. By changing the file display options, though, you can limit the types of files that appear in the file list and you can alter their order. To control the file list display, you can:

- change the file specifier,
- include hidden and system files,
- change the sort order.

### *Changing the File Specifier*

Suppose you want the DOS Shell to display only .EXE files in its file list. To limit the file list to this type of file, you must change the file specifier. The *file specifier* is a pattern of characters that the DOS Shell uses to

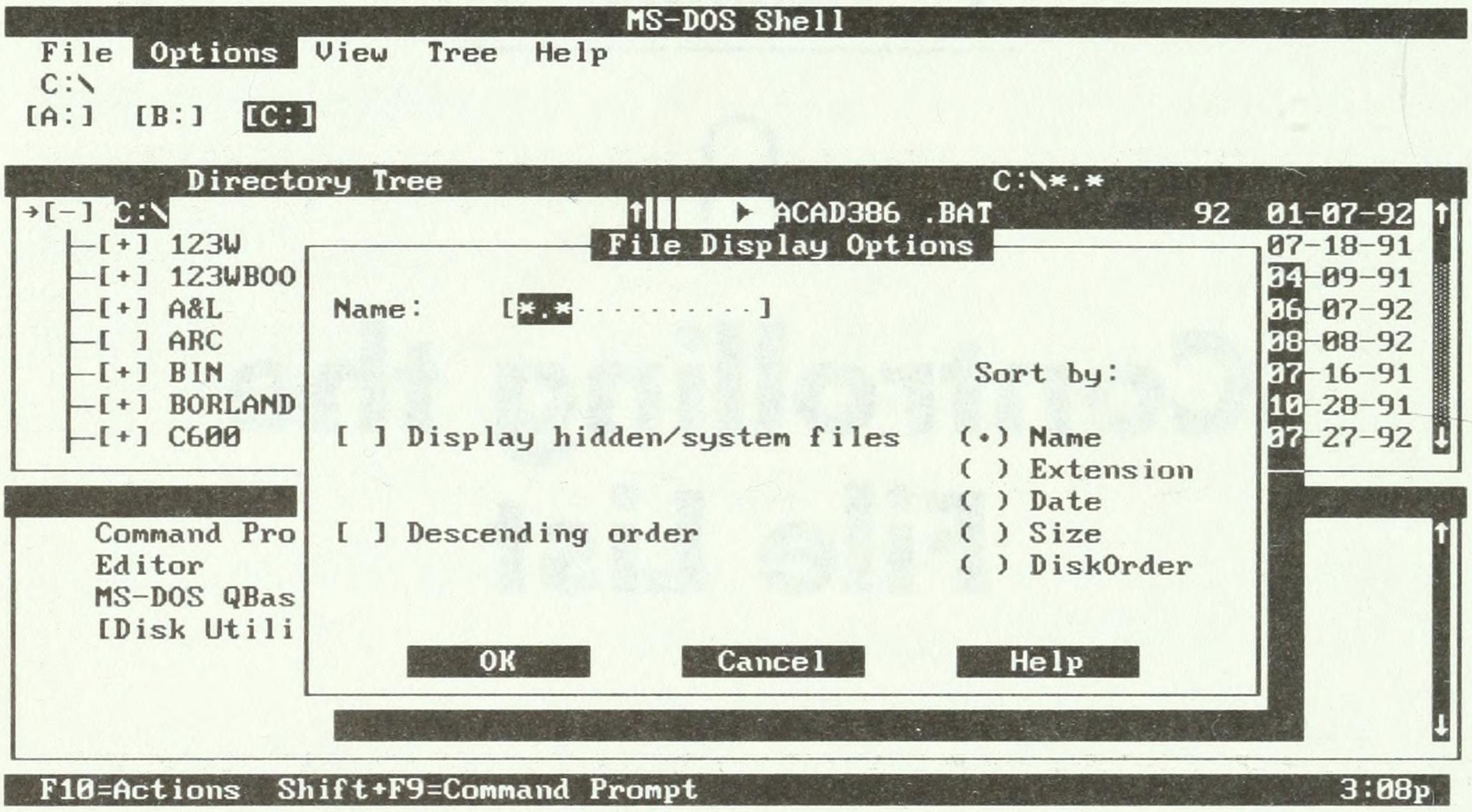



**Figure 8.1** Options File Display Options box

filter filenames. Any filenames that do not match the file specifier are omitted from the file list.

To display only .EXE files, you must establish a file specifier of *.exe by performing the following steps:

1. Select the **Options File Display Options** command. Figure 8.1 shows the dialog box.
2. Type ***.exe** in the Name: field and press **Enter**.

The DOS Shell immediately updates the file list to include only .EXE files. If there are no .EXE files in the current directory, you'll see the message "No files match specifier" in the file list. Because the root directory rarely contains .EXE files, this is the message that you're likely to see when it is the current directory. Switch to the DOS directory, though, and your screen will look like Figure 8.2.

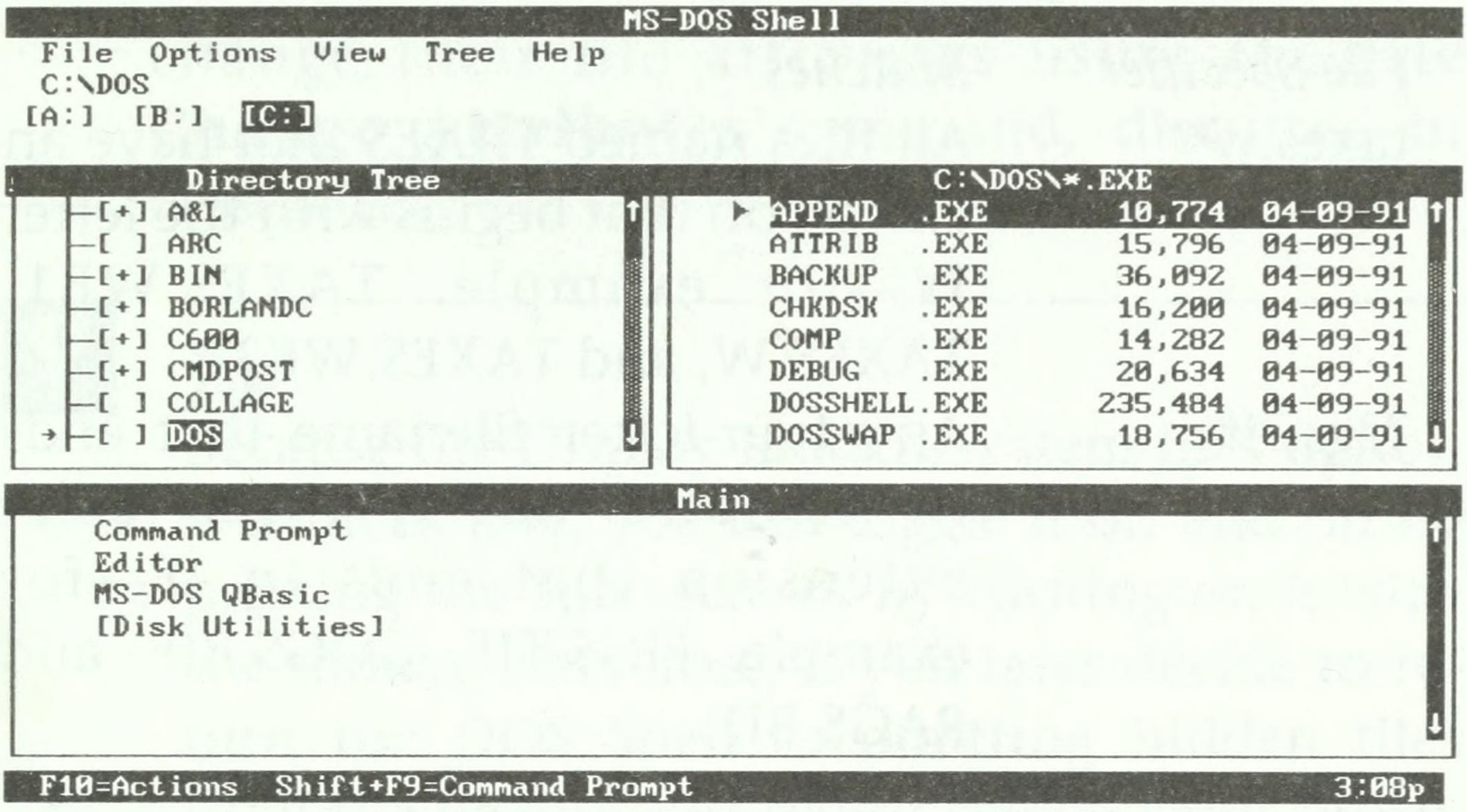



**Figure 8.2** Displaying only .EXE files

To return to displaying all the files in the current directory, repeat the previous procedure and use ***.*** for the file specifier.

## THE ? AND * WILDCARD SYMBOLS

When setting the file specifier, you can use the ? and * wildcard symbols (*wildcards* for short). The ? wildcard matches any single character, and, as the previous example showed, the * wildcard matches any group of characters. Here are some more examples:

| *File Specifier* | *Matches* |
|---|---|
| b*.* | All files that begin with the letter B—for example, B1.EXE, BLUESKY.WK1, and B_LEVEL3.BAK. |

| *File Specifier* | *Matches* |
|---|---|
| taxes.w* | All files named TAXES that have an extension that begins with the letter W—for example. TAXES.WB1, TAXES.W, and TAXES.WK3. |
| ??gs.?if | Any four-letter filename that ends in GS and has a three-letter extension that ends in IF—for example, FIGS.TIF, COGS.RIF, and RAGS.BIF. |

## Including Hidden and System Files

When you install DOS 5.0 on your system, it automatically places two hidden files in the root directory of your hard disk: IO.SYS and MSDOS.SYS. Normally, the DOS Shell doesn't show these files in the file list. To see these and other hidden files, follow these steps:

1. Select **Options** then the **File Display Options** command.
2. Press **Tab** to move to the Display hidden/system files check box.
3. Press **Spacebar** (or click on the option with the mouse) to turn the check box on. An X appears in the [ ] portion of the option.
4. Press **Enter**.

If you select the root directory in the directory tree, IO.SYS and MSDOS.SYS will now appear in the file list.

⇨ ***Note:*** Viewing hidden files doesn't change their hidden status. For that, you'll need to

change their file attributes using the **File Change Attributes** command, discussed in Lesson 12.

### *Tip*

Because the Display hidden/system files option is a check box, you can toggle it on and off by pressing the Spacebar, or by clicking on it with the mouse. Therefore, if you later decide to return the DOS Shell to omitting hidden files from the file list, just repeat the previous procedure. This time, though, pressing the Spacebar (or clicking the mouse) in Step 3 will toggle the option off (the X will disappear).

## *Changing the Sort Order*

As mentioned, the DOS Shell file list normally shows your files sorted by name in ascending order. Although this method of display is fine in most cases, occasionally you'll find it useful to change the sort order. For example, to see which files have been added to disk most recently, you could display your files sorted by date, with the most recent files listed first.

To change the file list's sort order, you use the Options File Display Options command and choose one of the following Sort by: radio buttons shown in Figure 8.1:

| | |
|---|---|
| **Name** | Sorts by filename (the default). |
| **Extension** | Sorts by file extension. |
| **Date** | Sorts by date of creation. |

**Size** Sorts by file size.

**DiskOrder** This is the actual order in which files are placed on disk; in other words, selecting this option means that no sorting takes place at all.

In addition, you can specify whether the file list uses a descending sort order by turning on the Descending Order check box.

Suppose you want the DOS Shell to refrain from sorting your files so that you can see where on the disk they actually reside. To do so, turn on the DiskOrder radio button, as follows:

1. Select **Options** then the **File Display Options** command.
2. If you carried out the previous procedures in this lesson, you may need to return the file specifier to *.* so that the file list will show all the files in the current directory. Type ***.*** in the Name: field if necessary.
3. Press **Tab** three times to move to the Sort by: radio buttons.
4. Press ↓ four times to turn on the **DiskOrder** radio button. (If you have a mouse, click on the DiskOrder radio button to turn it on.)
5. Press **Enter**.

Figure 8.3 shows the effect of this new setting when you select the DOS directory. Notice that the files no longer appear in alphabetical order but in the order they were actually placed on the disk.

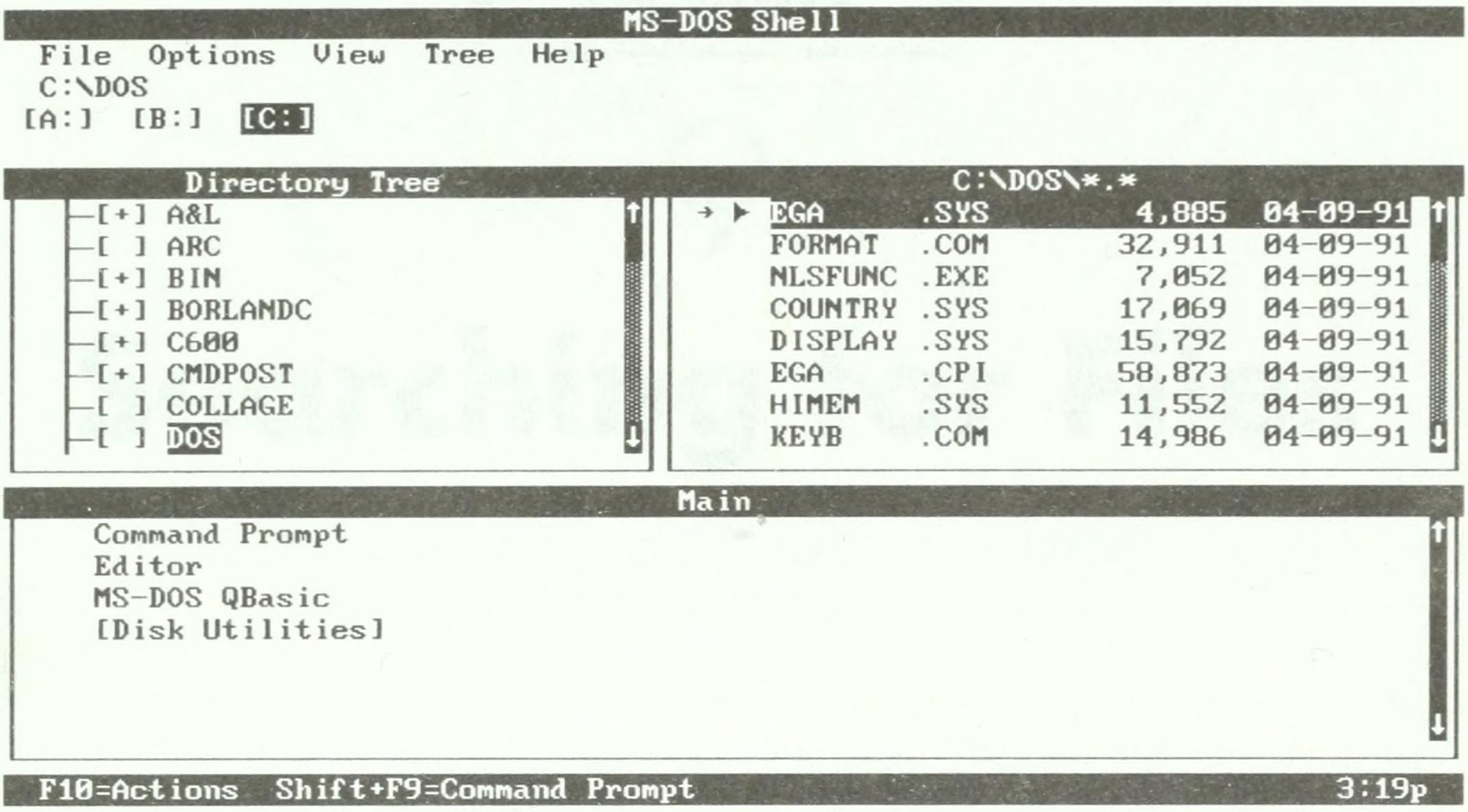



**Figure 8.3** The DiskOrder radio button turned on

## Rereading All Files and Directories

Occasionally, the files and directories listed in the DOS Shell will not reflect the true state of the disk. For example, suppose you temporarily leave the DOS Shell and execute a command at the DOS command line to create a new directory (explained in Lesson 21). When you return to the DOS Shell, the directory tree will not show the new directory you've created. To update the DOS Shell so that it shows the new directory, select the **View Refresh** command or its shortcut, **F5**. The DOS Shell will pause for a moment as it rereads all the files and directories on the current disk and then show the new directory in its updated directory tree and file list.

◆ *Lesson* ◆

9

# Searching for Files

## Searching for a Single File

As the number of files on your hard disk increases, you may forget where a file is stored. Rather than spending a lot of time poking around, consider using the DOS Shell File Search command instead.

Recall that in Lesson 4, you were asked to move the APPNOTES.TXT file from the your C:\DOS directory to your root directory (C:\). You were then asked to move it back. Let's suppose that something went wrong during that lesson and now you can't find the APPNOTES.TXT file anywhere. To locate this file, perform the following steps:

1. Select **File** then the **Search** command. DOS Shell displays the dialog box in Figure 9.1.
2. Type **appnotes.txt** in the *Search for* text box.
3. Make sure the *Search entire disk* check box is turned on. That way the DOS Shell searches your entire drive C. If you turn off this check box, only current the current directory will be searched.

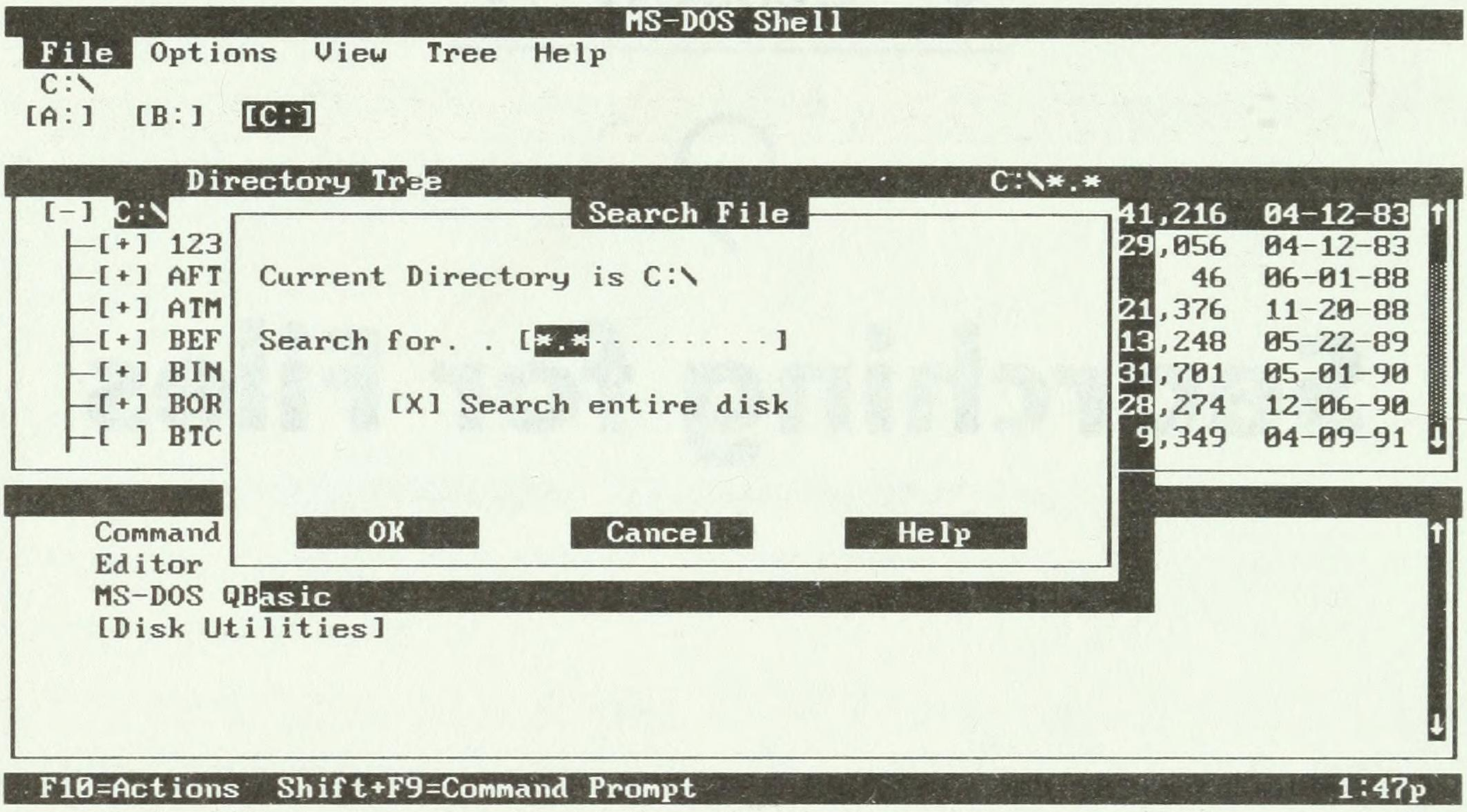



**Figure 9.1** Searching for files

(The current directory is indicated at the top of the Search File dialog box.)

4. Press **Enter** to begin the search. DOS Shell hesitates for a moment and then displays the screen in Figure 9.2 showing the location of the APPNOTES.TXT file.

When the DOS Shell locates the file you're looking for, it displays its path and name in a files list. At this point, you can note the location of the file and press Esc to return to the DOS Shell. Alternatively, you can perform an operation on the file. For example, you can copy, move, rename, or delete. If it's a program file, you can press Enter to run the program.

## Searching for a Group of Files

Although the previous example showed you how to search for a single file, you can also search for groups of

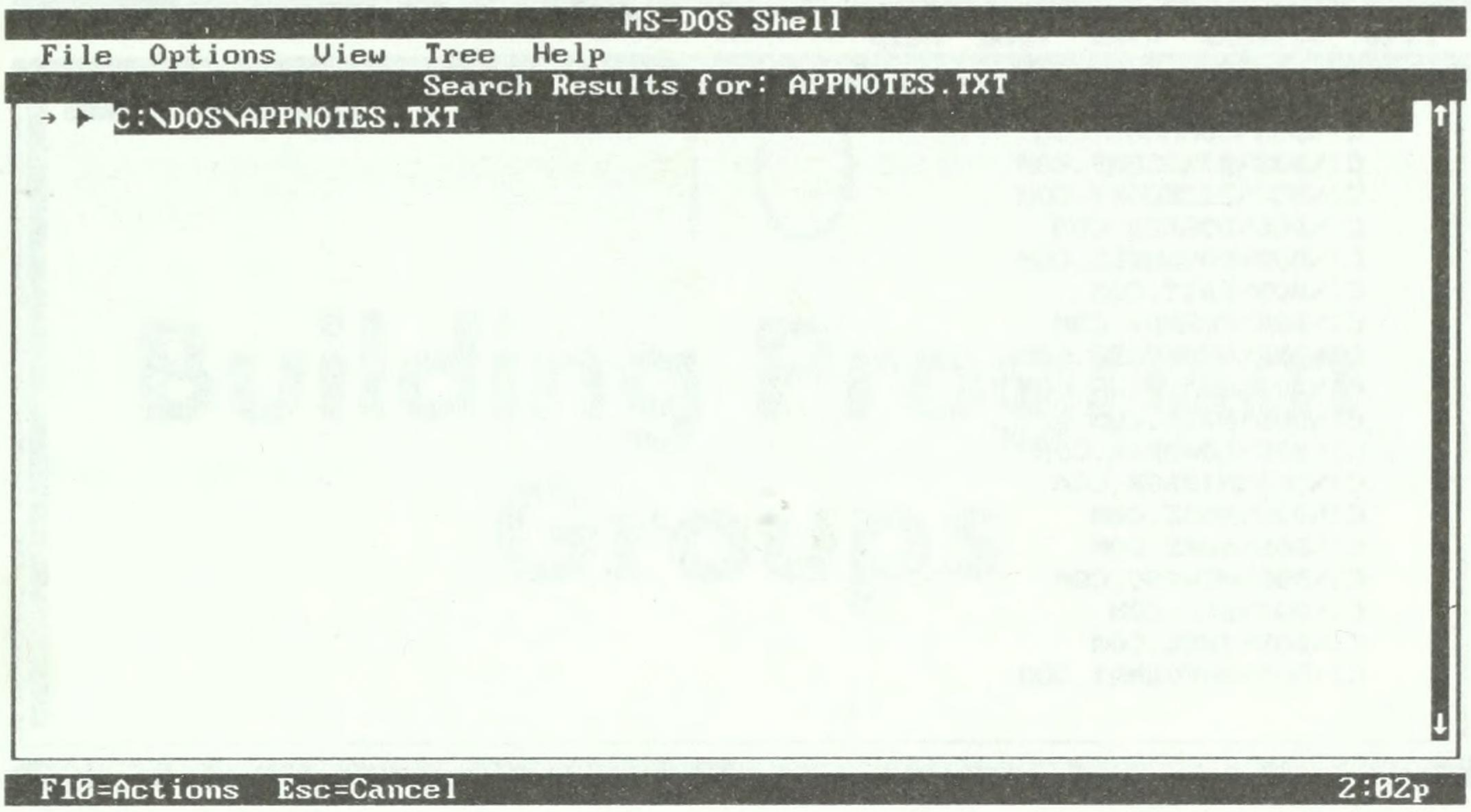



**Figure 9.2** APPNOTES.TXT in the C:\DOS directory

files with related filenames or extensions. To do so, you use wildcards. For example, suppose you want to locate all the files in your hard disk that have a .COM extension. To do this, perform the following steps:

1. Select **File** then the **Search** command. DOS Shell displays the Search File dialog box shown in Figure 9.1.
2. Type ***.com** in the Search for text box. This wild card pattern will select all files with a .COM extension.
3. Make sure the *Search entire disk* check box is turned on to have DOS Shell search your entire drive C.
4. Press **Enter** to begin the search. The DOS Shell hesitates for a moment and then displays a screen like the one in Figure 9.3 showing the location of all the .COM files on your hard disk.

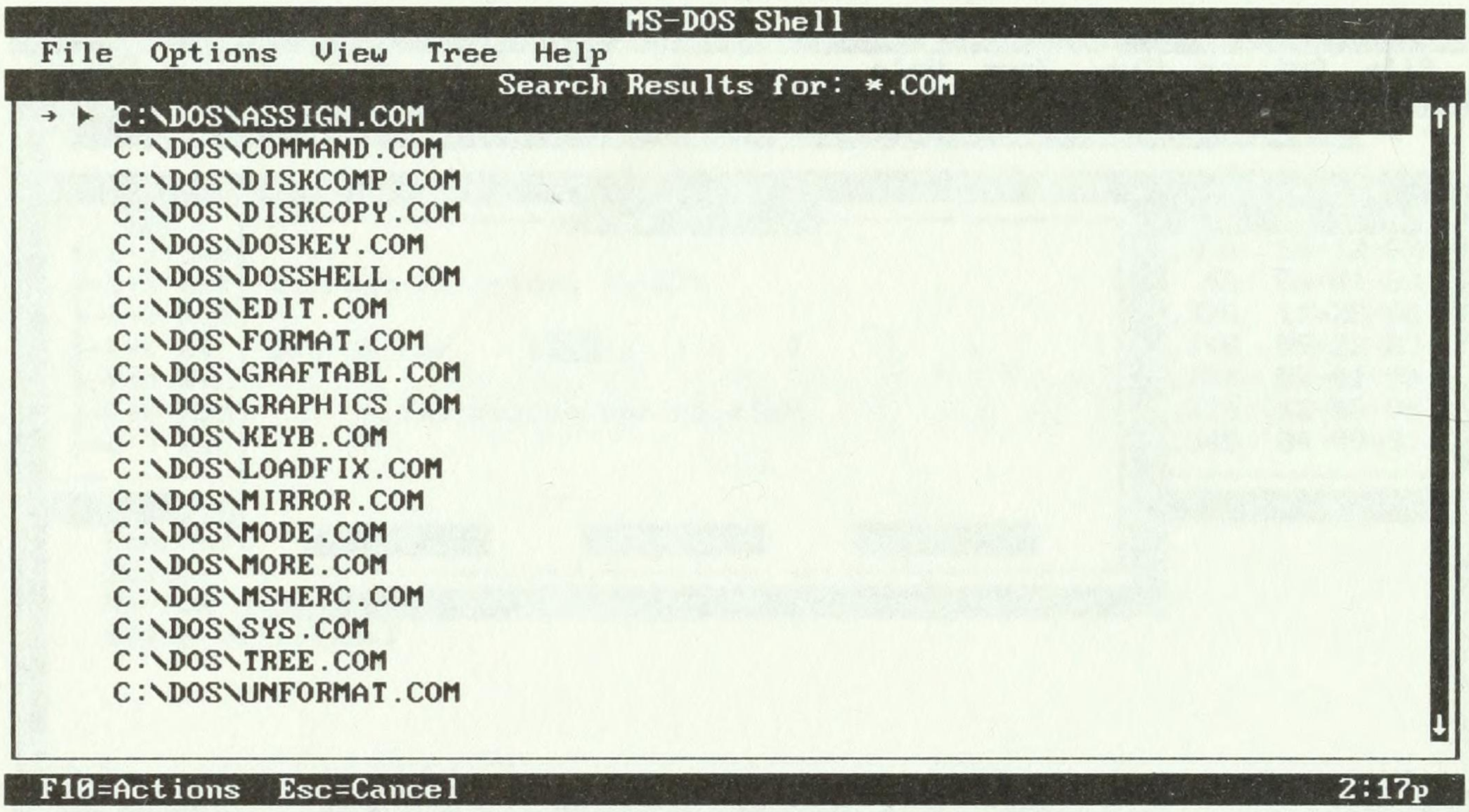



**Figure 9.3** Searching for a group of files

◆ *Lesson* ◆

10

# Building Program Groups

## Understanding Program Groups

As you learned in Lesson 5, a program group is a collection of program names that appear in the program-list area. The DOS Shell has two program groups: the Main group, which is displayed automatically when you start up the DOS Shell, and the Disk Utilities group, which appears when you select [Disk Utilities] in the program list.

Although these two program groups appear to be permanent fixtures in the DOS Shell, actually you can manipulate them to your own liking. In fact, you can perform any of the following operations related to program groups:

- Create a new program group.
- Add program items to a group.
- Rearrange program groups.
- Change the properties of a program item.

In order to work with program groups and program items, you must activate the program-list area. (You

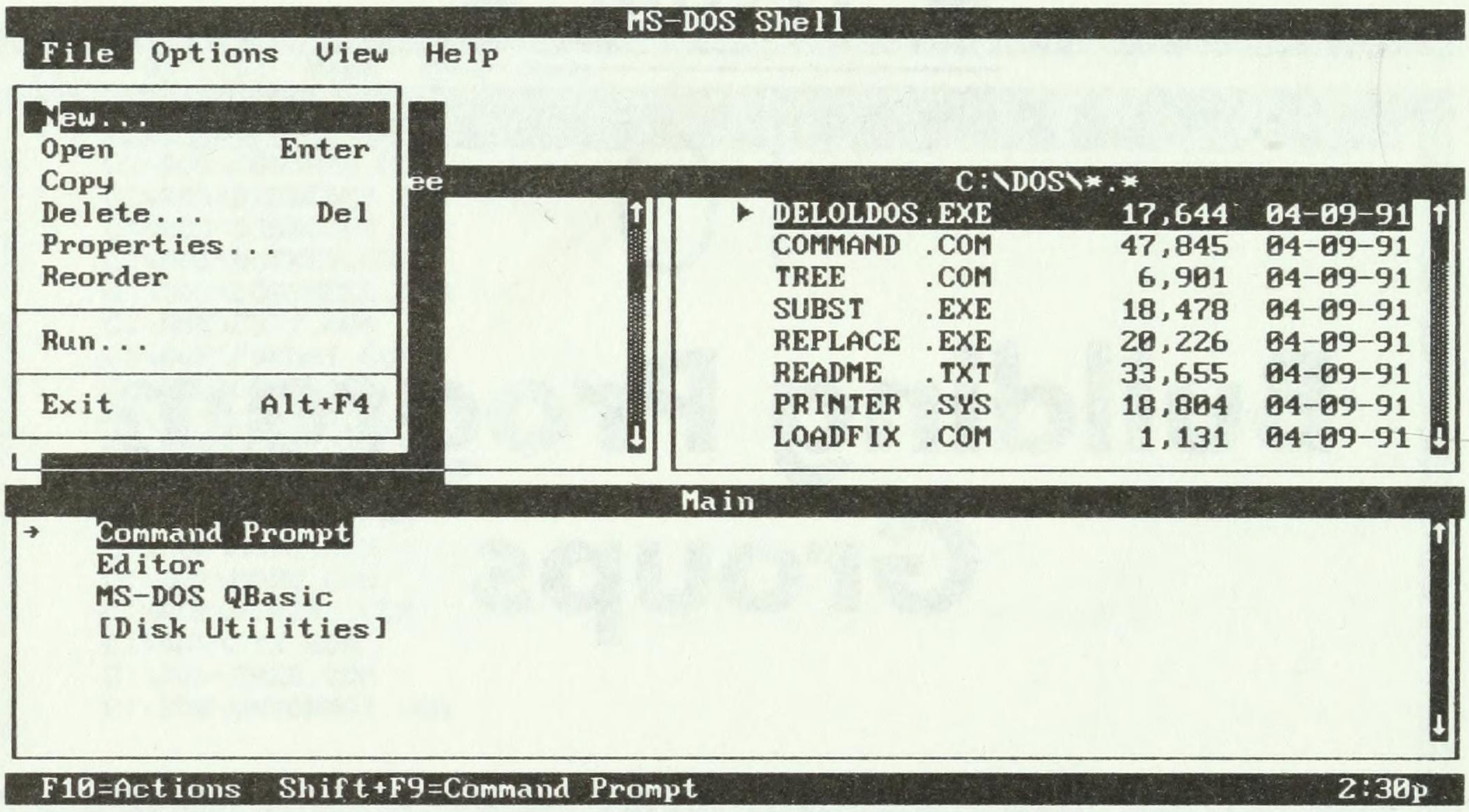



**Figure 10.1** The program-list area in the File menu

can do this by pressing **Tab** or by clicking on the area with your mouse.) If you then select the File command, the DOS Shell presents the options that appear in Figure 10.1. These options allow you to work with program groups and will be discussed in the sections that follow.

## *Contents of Existing Program Groups*

Before creating new program groups and adding program items to them, let's review the contents of the existing program groups and how to navigate among them. The items in the [Main] program group are:

**Command Prompt** Runs the MS-DOS command interpreter (COMMAND.COM), taking you to the DOS command line. (See Lesson 21.)

**Editor** Starts the MS-DOS Editor (EDIT.COM), which lets you view and edit ASCII files. (See Lessons 13 and 28.)

**MS-DOS QBasic** Starts the MS-DOS QBasic utility (QBASIC.EXE).

You can run a program item by using the arrow keys to move the highlight to the item's name and then pressing Enter. Alternatively, you can double-click on the item's name with the mouse.

The [Disk Utilities] group also contains multiple program items. To access these items, though, you must first open the group. To do this, highlight [Disk Utilities] and press **Enter** or double-click on it with the mouse. The following program items are then displayed:

**Disk Copy** Runs the Diskcopy utility (DISKCOPY.COM), which copies floppy disks. (See Lesson 15.)

**Backup Fixed Disk** Runs the Backup utility (BACKUP.EXE), which backs up files. (See Lesson 18.)

**Restore Fixed Disk** Runs the Restore utility (RESTORE.EXE), which restores backed up files. (See Lesson 19.)

**Quick Format** Runs the Format utility (FORMAT.COM) with the /q switch, which erases the contents of a disk. (See Lesson 14.)

| | |
|---|---|
| **Format** | Runs the Format utility (FORMAT.COM), which formats a disk. (See Lesson 14.) |
| **Undelete** | Runs the Undelete utility (UNDELETE.EXE), which recovers deleted files. (See Lesson 16.) |

To close the [Disk Utilities] group and return to [Main], highlight **[Main]** at the top of the list and press **Enter** or double-click on the name with the mouse.

## Creating a New Program Group

In this section, you'll learn how to create a new program group named "Other Utils." (The next section will show you how to add program items to this group.) To create this new group, perform the following steps:

1. Activate the **program-list area**, if you haven't already.
2. Select **File** then **New**. The DOS Shell displays the New Program Object dialog box in Figure 10.2.
3. Turn on the **Group** check box by pressing the **Spacebar**.
4. Press **Enter**. The DOS Shell displays the *Add Group* dialog box shown in Figure 10.3 prompting you for a name for the new group.

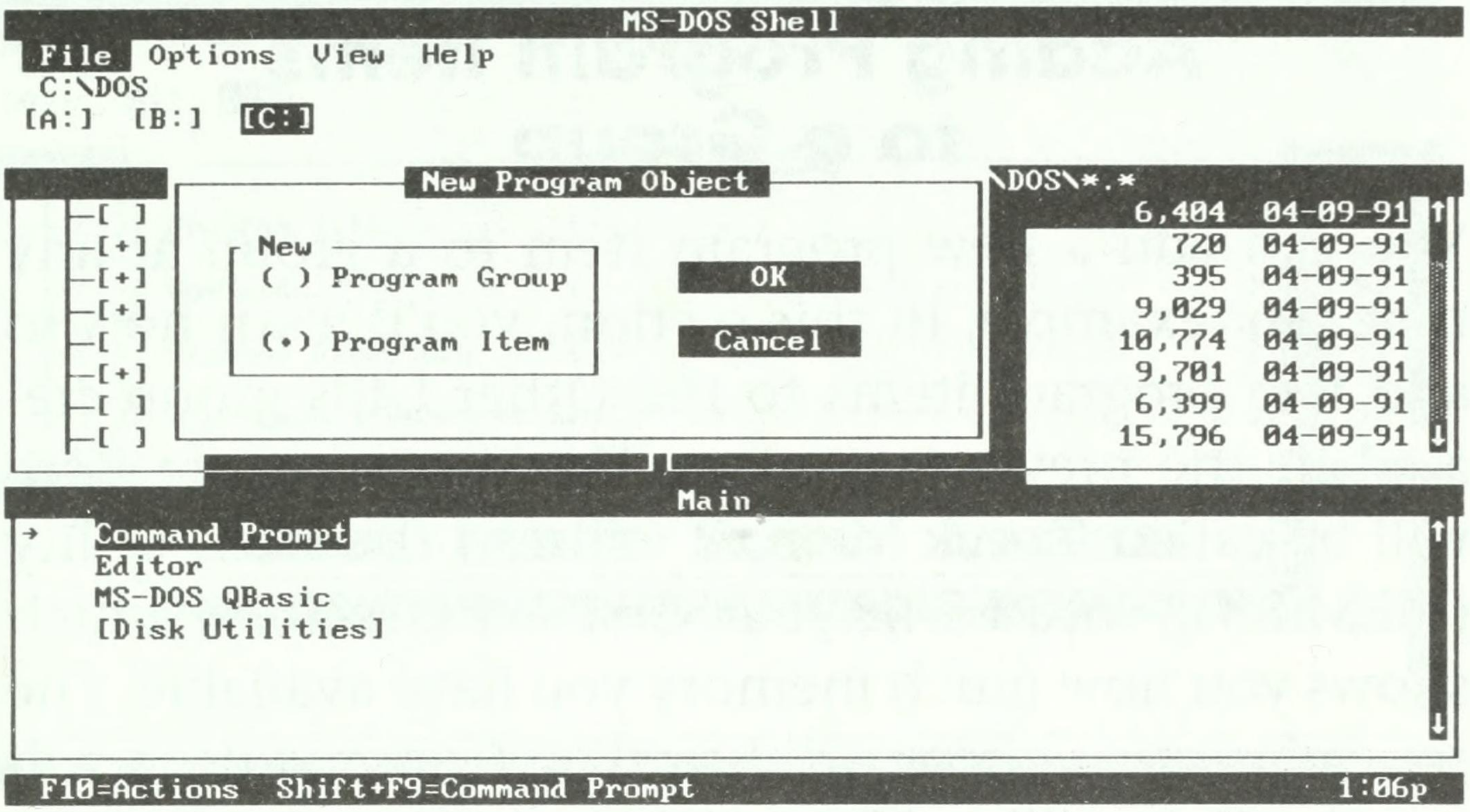



**Figure 10.2** The New Program Object dialog box

**5.** Type **Other Utils** in the Title text box and then press **Enter**. The DOS Shell adds your new group to the program-list area.

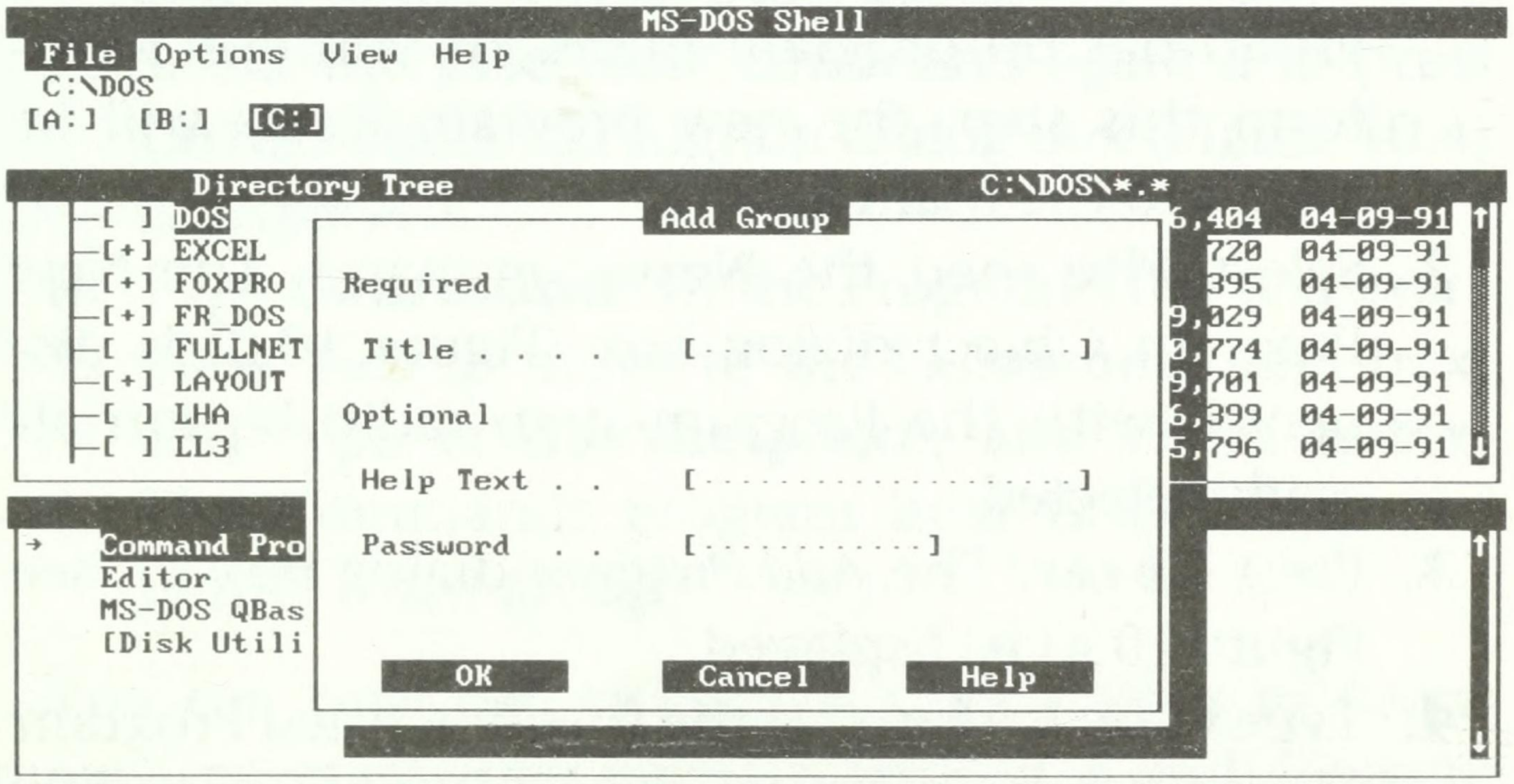



**Figure 10.3** The Add Group dialog box

# Adding Program Items to a Group

You can add a new program item to a group at any time. For example, in this section, you'll learn how to add two program items to the Other Utils group created in the previous section. The first program item will be called Check Mem; it will run the Mem utility (MEM.EXE) located in your C:\DOS directory, which shows you how much memory you have available. The second program item will be called Commands; it will run the Help utility (HELP.EXE) located in the C:\DOS directory, which shows a list of DOS commands with a brief description of what they do. To create these program items, perform the following steps:

1. Highlight the **[Other Utils]** group and press **Enter** or double-click on the group. This opens the group and displays its contents. Presently the group has no program items. (If you don't perform this step, the new program items will be added to the Main group.)
2. Select **File** then the **New** command. The New Program Object dialog box (Figure 10.2) is displayed with the Program Item radio button already selected.
3. Press **Enter**. The *Add Program* dialog box in (see Figure 10.4) is displayed.
4. Type **Check Mem** on the first line titled Program Title. This is the name that will appear in the program-list area.

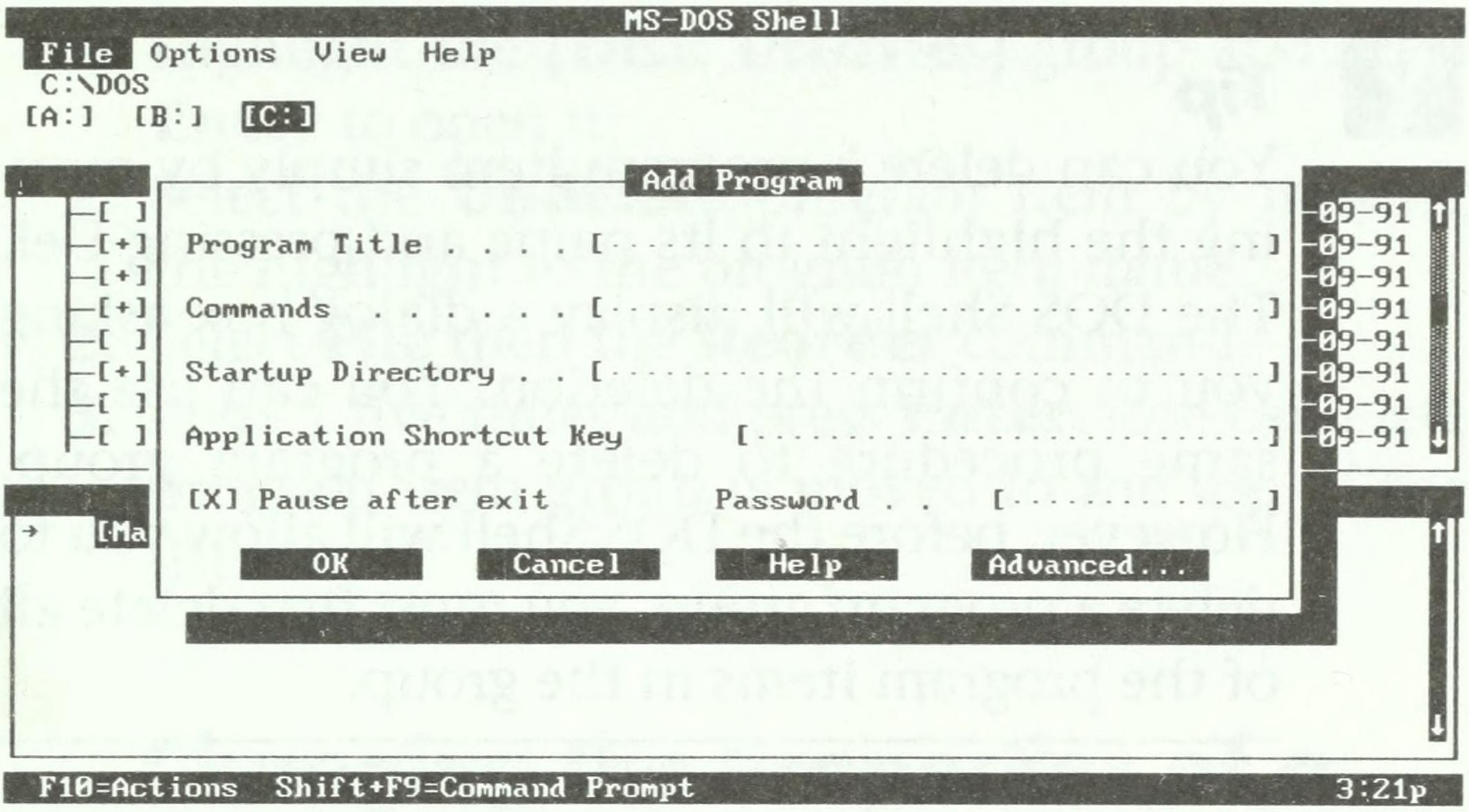



**Figure 10.4** The Add Program dialog box

5. Press **Tab** to move to the Commands text box, and type **c:\dos\mem.exe**. This is the program that the new item will run.
6. Press **Enter**. The Check Mem program item is added to the [Other Utils] group.
7. Select the **File New** command again and press **Enter**. The Add Program dialog box (Figure 10.4) is displayed.
8. Type **Commands** in the Program Title text box.
9. Press **Tab** to move to the Commands text box, and type **c:\dos\help.exe**, and press **Enter**. The Commands program item is added to the [Other Utils] group.

You can now run either the Check Mem or Commands program item by highlighting it and pressing Enter or by double-clicking with the mouse.

**Tip**

You can delete a program item simply by moving the highlight to its name and pressing Del. The DOS Shell will display a dialog box asking you to confirm the deletion. You can use the same procedure to delete a program group. However, before the DOS Shell will allow you to delete a program group, you must first delete all of the program items in the group.

## Rearranging Program Groups and Items

You can change the arrangement of program groups and program items in the program-list area by using the File Reorder command. For example, suppose you use the [Disk Utilities] program group more than any other and would prefer that it appear at the top of the list in the [Main] group. To make this change, perform the following steps:

1. Select the **[Disk Utilities]** group by highlighting its name.
2. Select **File** then the **Reorder** command.
3. Press ↑ three times and then press **Enter**. The [Disk Utilities] group is moved to the top of the list.

Now let's suppose you frequently use the Undelete program item in the [Disk Utilities] group and would like to see it appear at the top of the list within the group. To make this change, follow these steps:

1. Highlight the **[Disk Utilities]** group and press **Enter** to open it.
2. Select the **Undelete** program item by moving the highlight to the program item name.
2. Select **File** then the **Reorder** command.
3. Press ↑ five times and press **Enter**. The Undelete program item group is moved to the top of the list.

## Changing the Properties of a Program Item

The properties of a program item also can be changed—for example, its name, the program it runs, and so on—by using the File Properties command. In the previous section, you created a new program group called [Other Utils]. Within that group, you created a program item named Check Mem that runs the Mem utility (MEM.EXE), which shows how much memory you have available. In this section we'll show you how to modify that program item to run an entirely different utility, Chkdsk (CHKDSK.EXE), which displays information about your hard and floppy disks. To modify this program item, perform the following steps:

1. If necessary, open the **[Other Utils]** group by moving the highlight to its name and pressing **Enter** or double-clicking on it with the mouse.
2. Select the **Check Mem** program item by moving the highlight to the item or clicking on it.
3. Select **File** then the **Properties** command. The DOS Shell displays the dialog box shown in Figure 10.5, listing the current properties for the item.

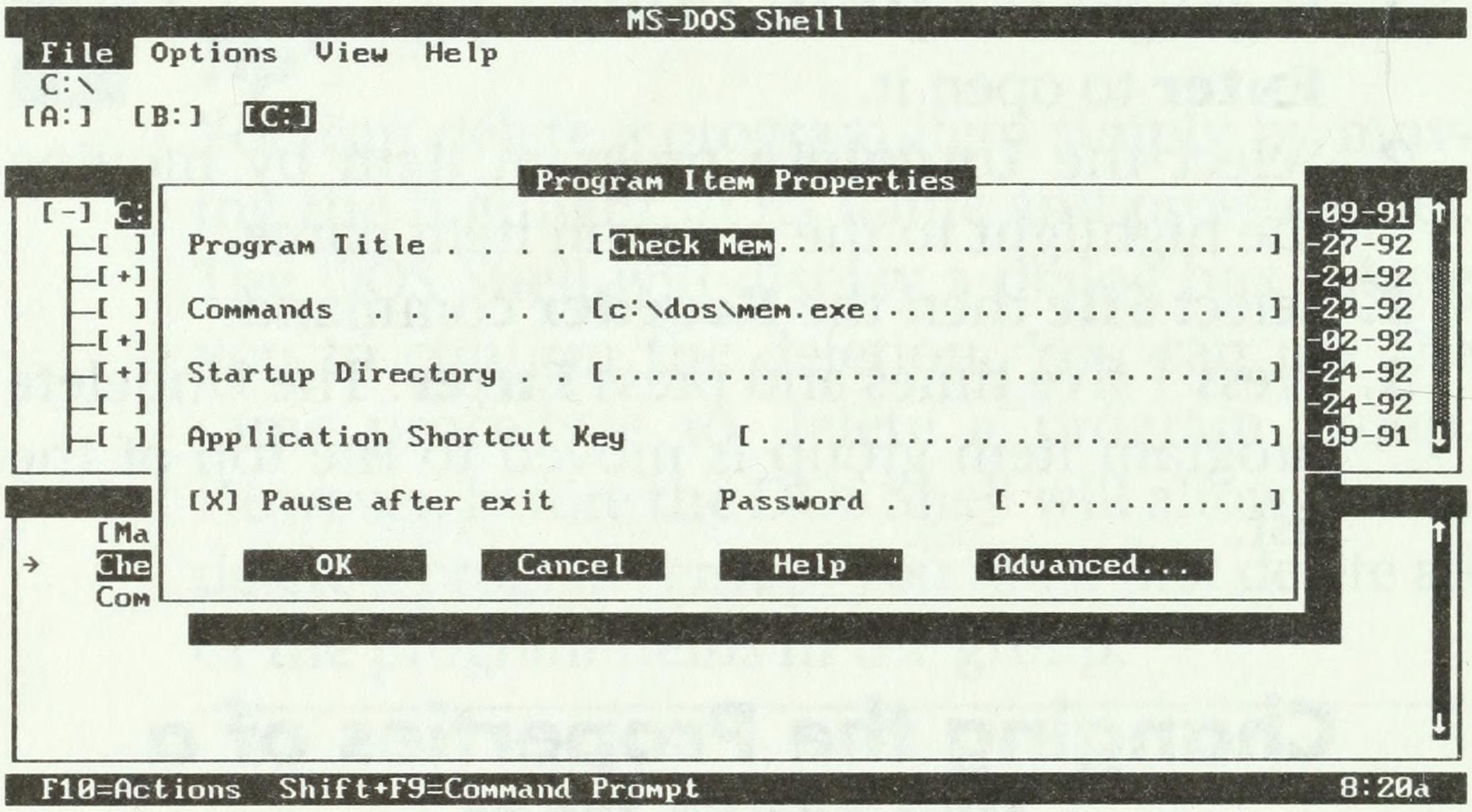



**Figure 10.5** Program Item Properties dialog box

4. Type **Check Disk** in the Program Title box.
5. Press **Tab** to move to the Commands box and type **c:\dos\chkdsk.exe**.
6. Press **Enter**. The modified program item now appears in the [Other Utils] group.

When you highlight the program item and press Enter or double-click on it, the DOS Shell runs the Chkdsk utility instead of the Mem utility.

◆ *Lesson* ◆

11

# Customizing the DOS Shell

## Customizable Features

The DOS Shell includes several commands that let you customize both its appearance and operation. In this lesson, you'll learn how to use these commands to do the following:

- Change the DOS Shell display colors.
- Change the display mode from the default text mode to the graphics mode.
- Turn off confirmation dialog boxes that are displayed when you delete or replace a file.
- Create file associations, which establish links between data files and their associated programs.

## Changing Colors

To change the DOS Shell display colors, select **Options** then the **Colors** command. (You may recall that this command was discussed briefly in Lesson 1 under "Selecting Menu Commands.") The DOS Shell displays

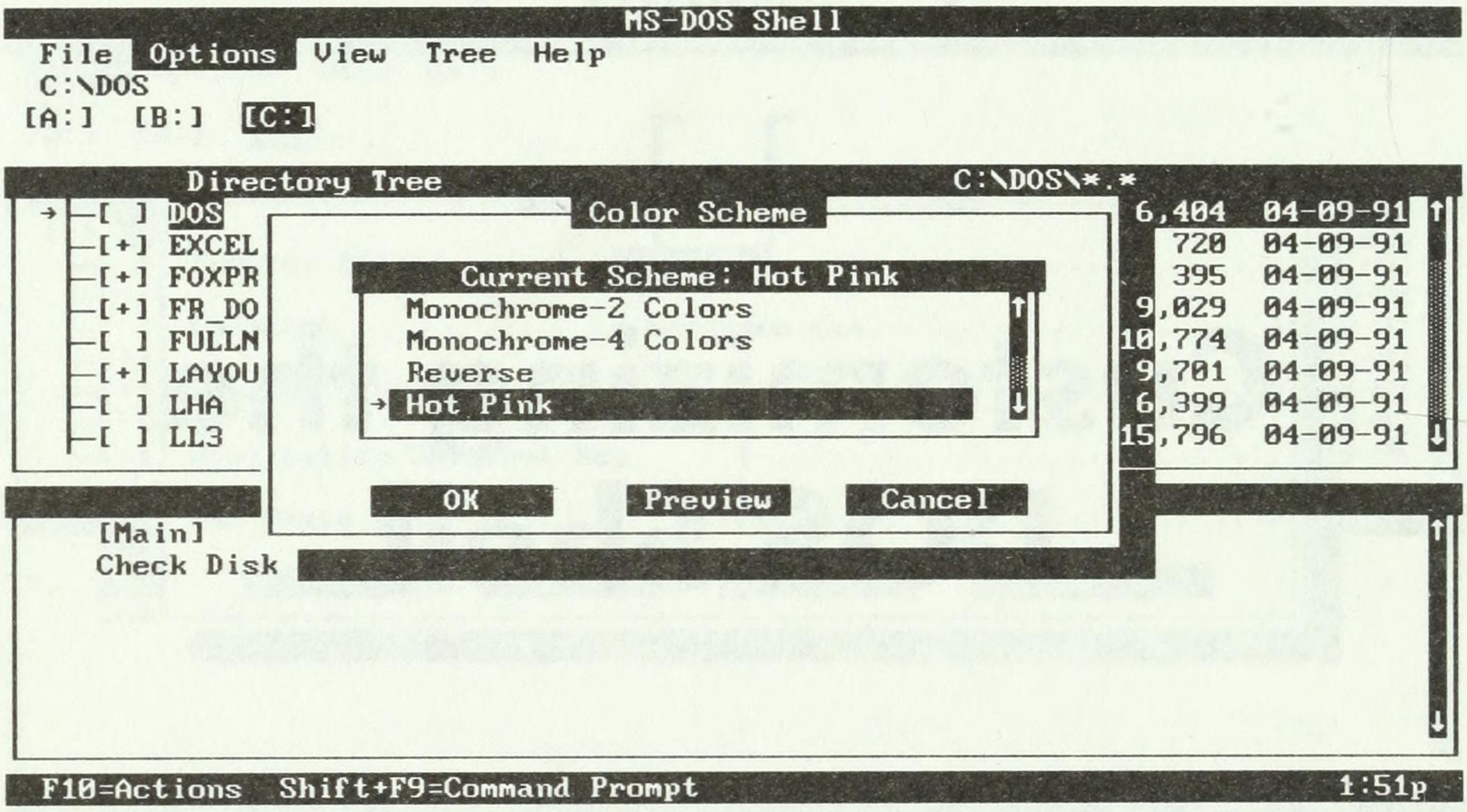



**Figure 11.1** The Color Scheme dialog box

the Color Scheme dialog box seen in Figure 11.1, which contains a list of names that correspond to different predefined color schemes. (The default color scheme is Ocean.) To try a different color scheme, use the ↑ and ↓ keys (or the scroll bar, if you are using a mouse) to scroll through the list of names until you find one that looks interesting. For example, Hot Pink is selected in Figure 11.1. To confirm your selection, press **Enter**. The DOS Shell adjusts its display colors to the color scheme you've selected.

## Changing the Display Mode

When you run the DOS Shell for the first time, its screen is displayed in *text mode*. With this setting, the DOS Shell screen can be displayed on older display adapters and monitors. A display adapter is an electronic card that plugs into a slot in the back of your

computer. This card, in turn, is connected to your monitor by means of a cable. Generally, the display adapter and monitor are purchased together and are of the same type—for example, Hercules, EGA, VGA, or Super VGA. Most modern display adapters and monitors, however, are also capable of displaying images in second mode known as *graphics mode*. In graphics mode, the shape and size of text and images on your screen appear more 3-dimensional and, therefore, more pleasing to the eye.

In both text and graphics modes, the DOS Shell will usually give you a choice of *resolution* settings—the number of lines that appear horizontally on your screen. The default setting is 25 lines, but you can usually select up to 50 lines in text mode and 60 in graphics mode. The higher the resolution you select, the more information you'll be able to see on your screen. However, both text and other objects will appear smaller if you increase the resolution.

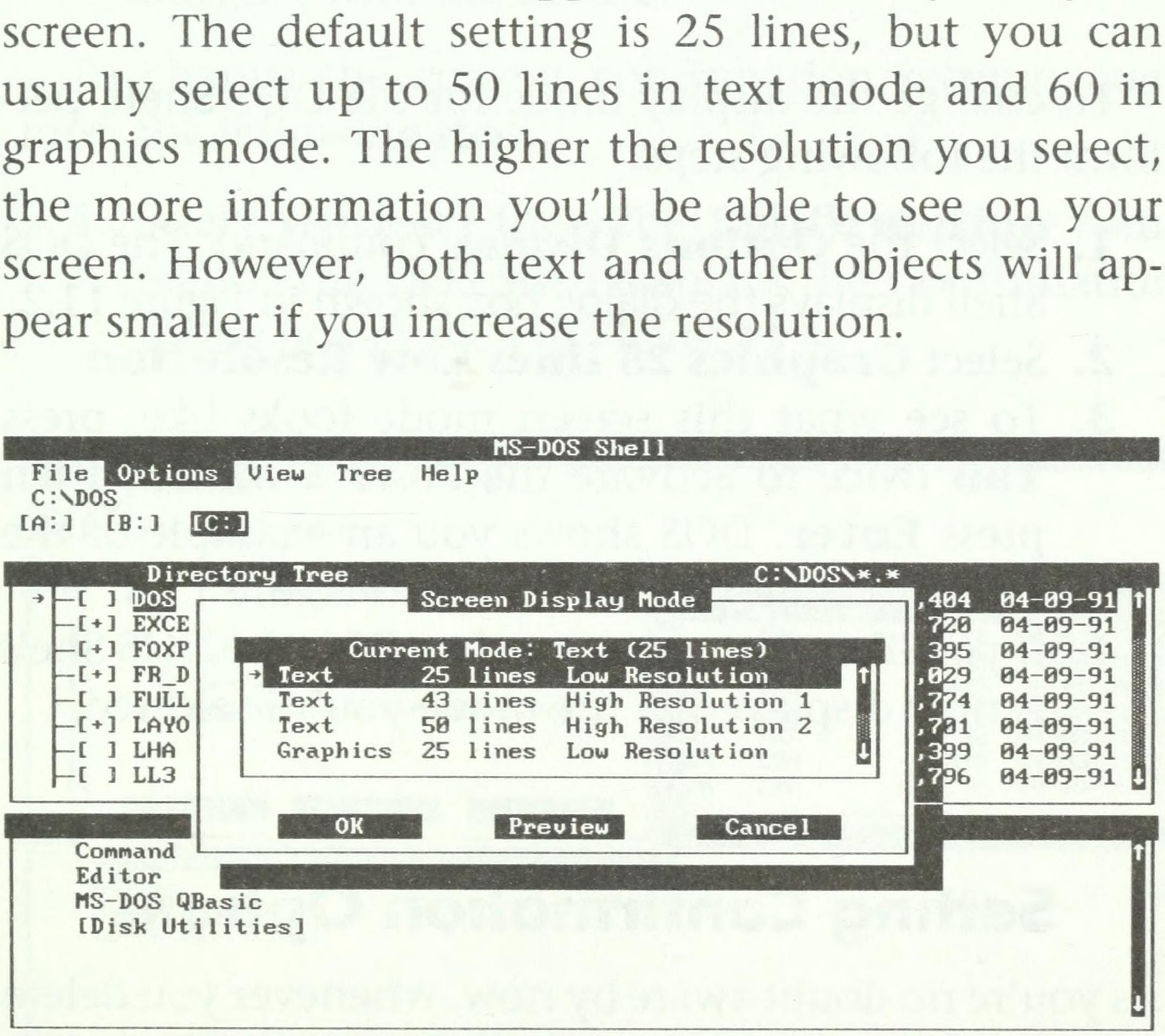



**Figure 11.2** Changing the display mode

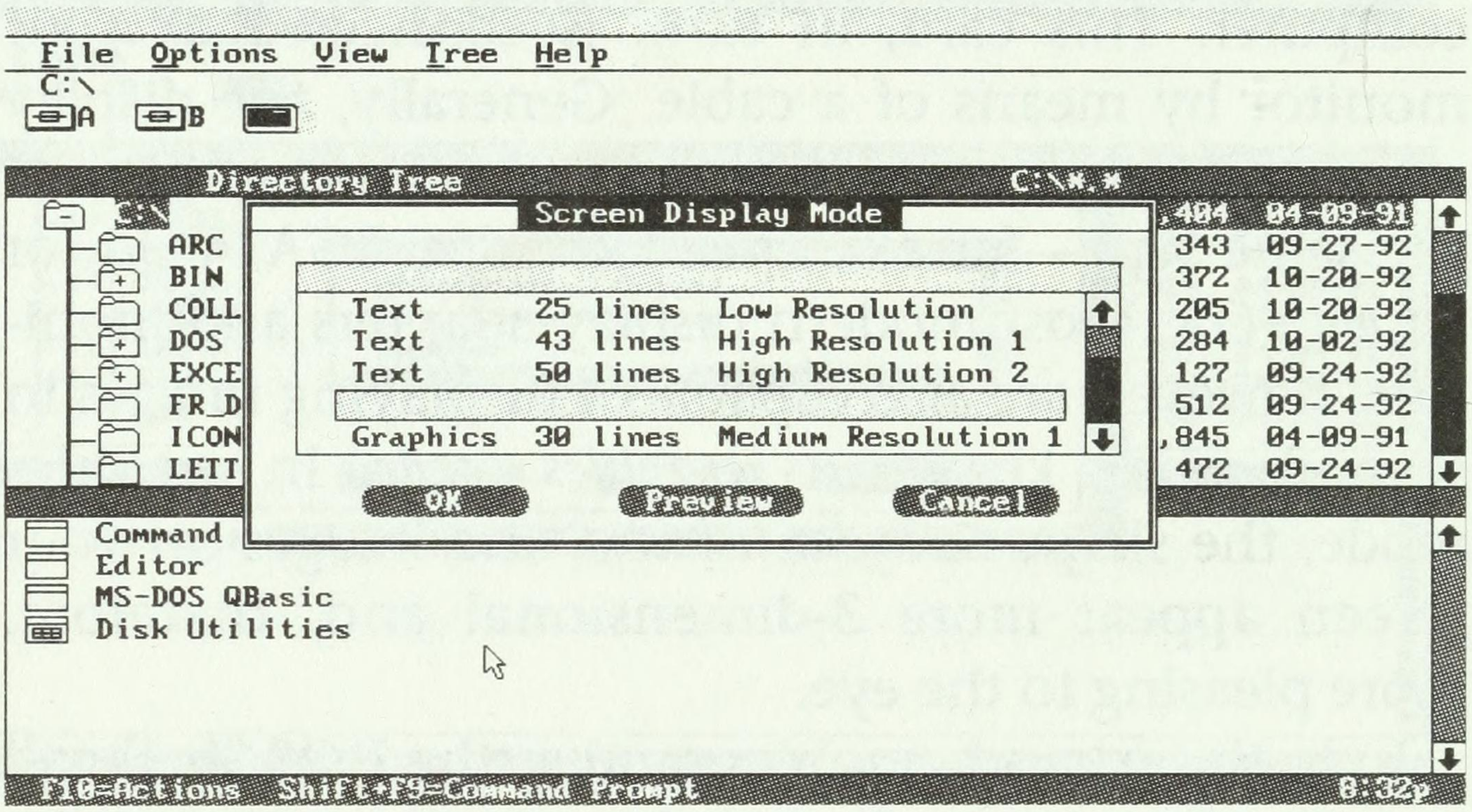



**Figure 11.3** Low resolution graphics mode

To change the display mode for the DOS Shell, perform the following steps:

1. Select the **Options Display** command. The DOS Shell displays the dialog box shown in Figure 11.2.
2. Select **Graphics 25 lines Low Resolution**.
3. To see what this screen mode looks like, press **Tab** twice to activate the Preview button, then press **Enter**. DOS shows you an example of the mode you selected, as shown in Figure 11.3.
4. If you like what you see, select **OK**. The DOS Shell is now displayed in the mode you've selected.

## Setting Confirmation Options

As you're no doubt aware by now, whenever you delete a file or directory, the DOS Shell shows a confirmation dialog box giving you a chance to reconsider. The same is true of many other types of operations in which

there's a chance that you've inadvertently chosen the wrong command or pressed the wrong key. If you'd rather not have the confirmations appear, you can use the Options Confirmation command to turn off the confirmation dialog boxes that are displayed when you do any of the following:

- Delete one or more files or directories.
- Replace one file with another—for example, when you copy a file from one directory to another, and the target directory already contains a file with the same name.
- Copy or move one or more files by clicking and dragging with the mouse.

To change the current confirmation settings, perform the following steps:

1. Select **Options** then the **Confirmation** command. The DOS Shell displays the Confirmation dialog box as seen in Figure 11.4.

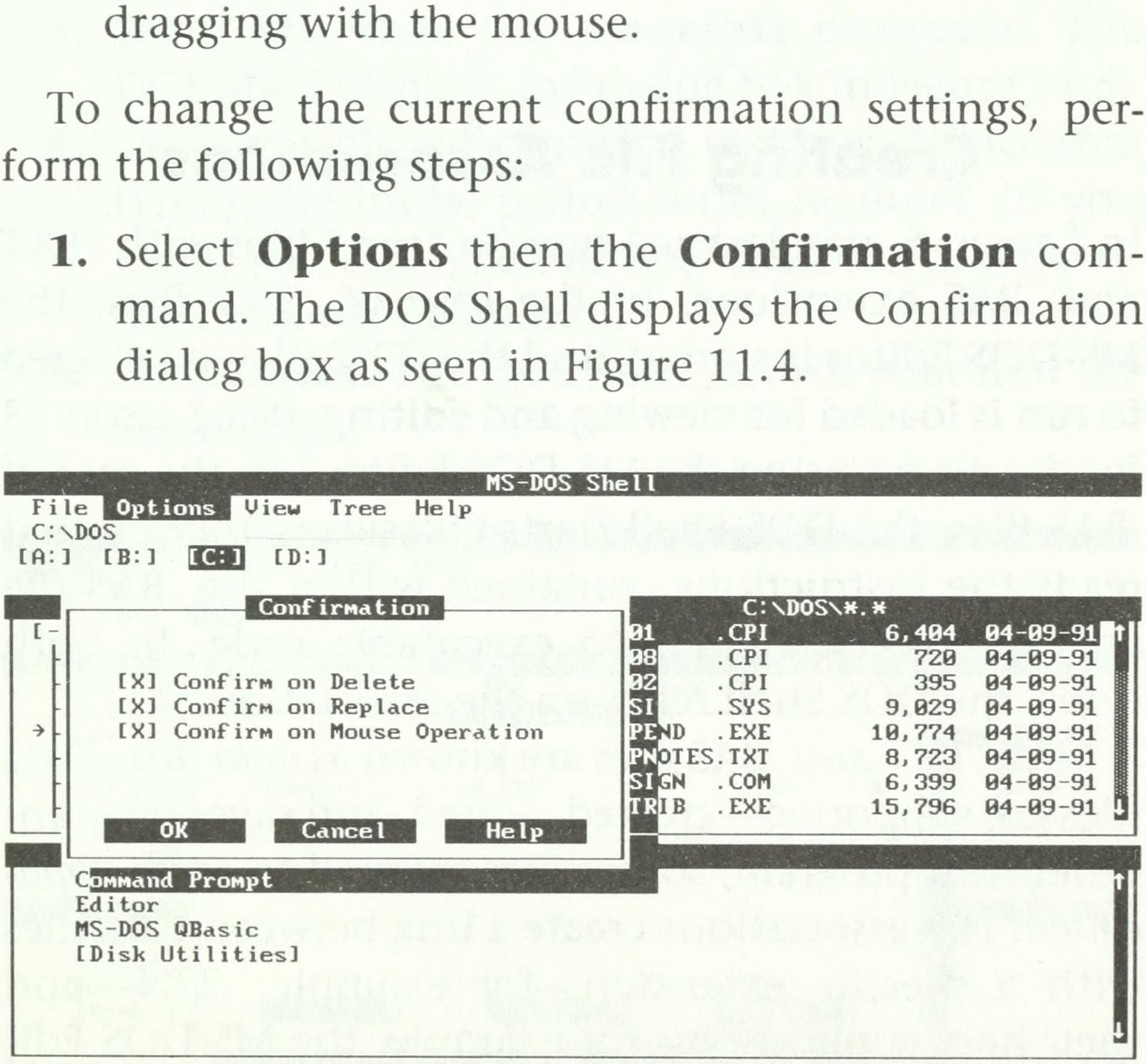



**Figure 11.4** The Confirmation dialog box

2. Clear the check box that corresponds to the type of confirmation you want to turn off. To do this, press **Tab** as necessary to activate the check box and press **Spacebar** to clear it. (Alternatively, you can clear a check box by clicking on it with the mouse.)
3. Select **OK** to implement the setting.

Don't change the current confirmation settings until you're experienced with the DOS Shell. The default settings help to ensure that you won't inadvertantly lose any of your files.

## Creating File Associations

In Lesson 5, you learned how to "run" files with .TXT and .BAS extensions. In the case of .TXT files, the MS-DOS Editor is started, and the .TXT file you elected to run is loaded for viewing and editing. (See Lesson 13 for details on using the MS-DOS Editor.) In the case of .BAS files, the DOS Shell starts QBasic, a program that reads the instructions contained within the .BAS file and interprets them into executable code. In both cases, the DOS Shell relies on file associations.

Both .TXT and .BAS files are known as *data files*. Data files are supported—created, edited, and saved—by another host program, sometimes referred to as an *application*. File associations create a link between data files with a specific extension—for example, .TXT—and their host application—for example, the MS-DOS Editor (C:\DOS\EDIT.COM). When you "run" the data file, the DOS Shell starts the associated application and automatically loads the data file into that application.

To create a file association, you use the File Associate command. For example, let's assume you use WordPerfect for your word processing. When saving files from this application, let's also assume you give those files a .WP extension; therefore you want to create an association between WordPerfect and files with a .WP extension. To do so, perform the following steps:

1. From the directory tree, select the directory that contains the application you want to associate with .WP files—for example, **C:\WP51**.
2. From the file list, select the program file that runs the application—for example, **WP.EXE**.
3. Select **File** then the **Associate** command. The DOS Shell displays the dialog box in Figure 11.5.
4. Type **wp** in the Extensions text box. Notice that typing the initial period is not required. (If you want, you can type multiple extensions, separating them with commas.)
5. Press **Enter**. The DOS Shell records your new file association.

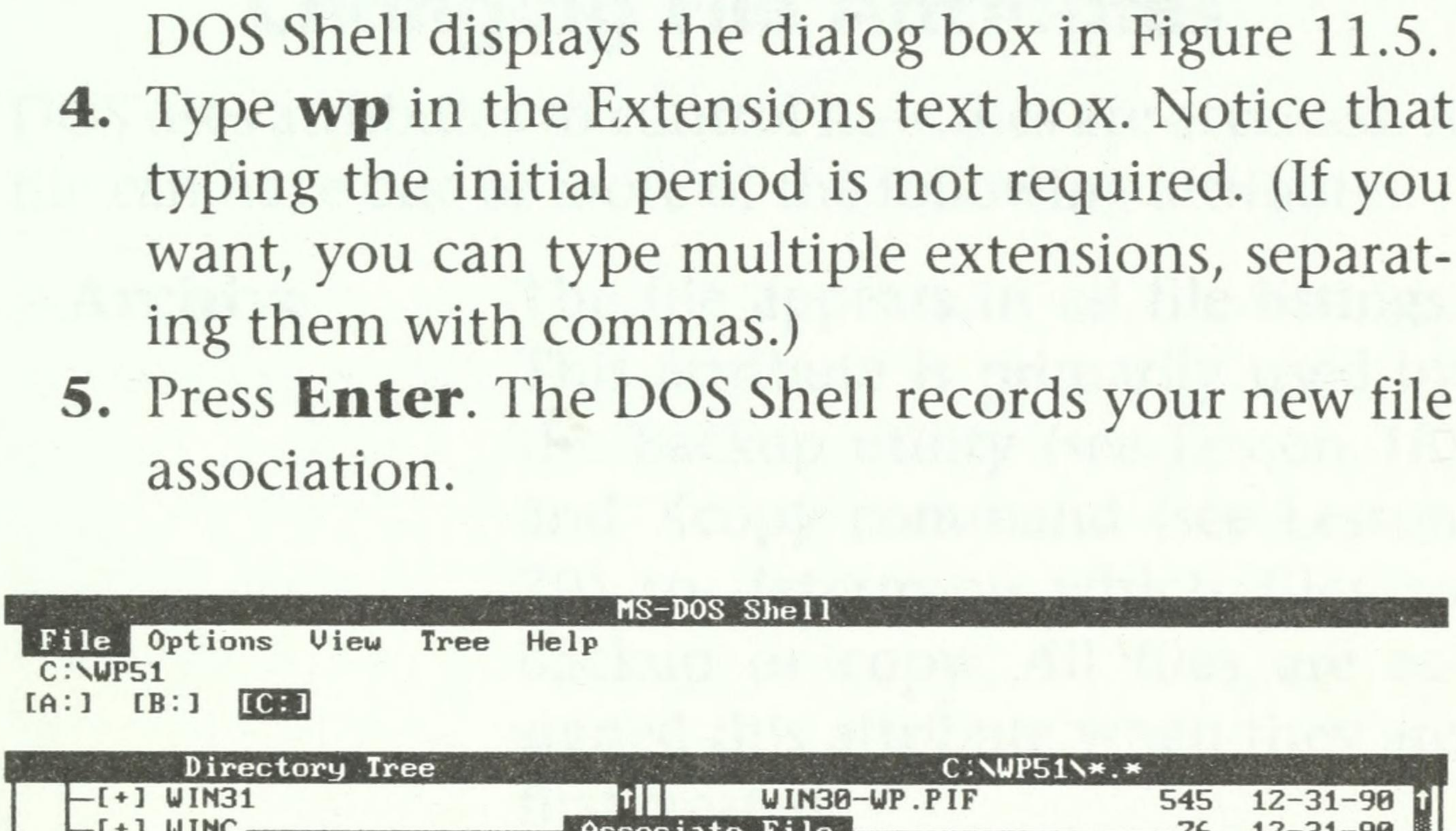



**Figure 11.5** Creating a file association

Once a file association has been created, you can begin using it immediately. For instance, you can select a file with a .WP extension in the file-list area and press Enter to run it. The DOS Shell will start WordPerfect and load the file into its work area.

◆ *Lesson* ◆

12

# Special File Operations

## Changing File Attributes

DOS uses attributes to control how files are accessed. A file can have one or more of the following attributes:

**Archive** The file appears in all file listings. This attribute is primarily used by the Backup utility (see Lesson 18) and Xcopy command (see Lesson 29) to determine which files to backup or copy. All files are assigned this attribute when they are first created.

**Read only** The file appears in all file listings but cannot be changed or deleted.

**Hidden** The file is concealed. It continues to exist but will not appear in any of the file listings.

**System** The file is a DOS system file, and like files with the Hidden attribute, files with this attribute do not appear in file listings.

To change file attributes, you use the File Change Attributes command. You may, for example, want to prevent another user who has access to your computer from changing or deleting the README.TXT file on your DOS directory. To do this, you'd assign a read-only attribute to the file in the following way:

1. Select your **DOS directory** from the directory tree.
2. Select **README.TXT** from the file-list area.
3. Select **File** then the **Change Attributes** command. The DOS Shell displays the dialog box in Figure 12.1.
4. Select the **Read only** option by pressing **Tab** and ↓ to move to the option and then pressing the **Spacebar**. (If you are using a mouse, simply click on this option.)
5. Select **OK** to confirm. The DOS Shell assigns the read-only attribute to the file.

Once the read-only attribute has been assigned to the file, you can't change its contents or delete it. To turn off the read-only attribute for README.TXT, follow steps 1 through 3 in the preceding example. When you reach step 4, turn off the Read only option by following the same procedure in that step.

To change the attributes of a group of files, select the files from the file-list area, and choose File then the Change Attributes command. When the DOS Shell displays a message asking whether you want to change

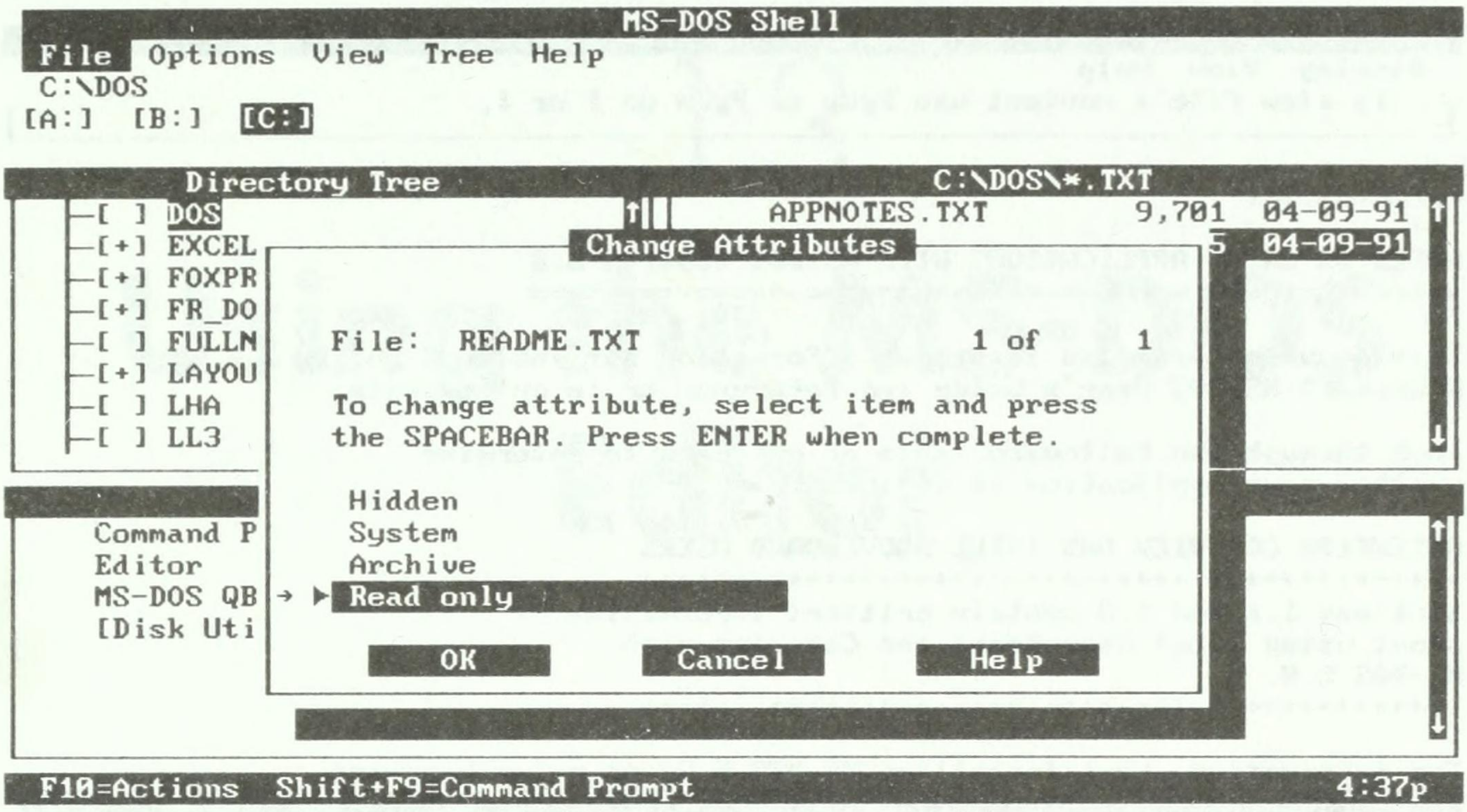



**Figure 12.1** Changing file attributes

the attributes for each file individually or as a group, choose the option you want.

## Viewing a File's Contents

You can quickly view the contents of any file by using the File View File Contents command. To look at the contents of the APPNOTES.TXT file in your DOS directory, for instance, perform the following steps:

1. Select your **DOS directory** from the directory tree.
2. Select the **APPNOTES.TXT** file from the file list.
3. Select **File** then the **View File Contents** command. The DOS Shell displays the file in a viewer window, as shown in Figure 12.2.
4. To scroll the file up or down, press **PgUp**, **PgDn**, ↑, or ↓.
5. To return to the DOS Shell, press **Esc**.

```
                    MS-DOS Shell - APPNOTES.TXT
  Display  View  Help
    To view file's content use PgUp or PgDn or ↑ or ↓.

APPNOTES.TXT

NOTES ON USING APPLICATIONS WITH MS-DOS VERSION 5.0
===================================================

This document provides important information not included in the
Microsoft MS-DOS User's Guide and Reference or in online Help.

Look through the following table of contents to determine
whether your application is included.

ATTENTION CODEVIEW AND INTEL ABOVEBOARD USERS
+++++++++++++++++++++++++++++++++++++++++++++
Sections 1.1 and 1.3 contain critical information
about using Intel Aboveboard and CodeView with
MS-DOS 5.0.
+++++++++++++++++++++++++++++++++++++++++++++

For information about installing MS-DOS 5.0 and using hardware
 <┘=PageDown  Esc=Cancel  F9=Hex/ASCII                        9:24p
```

**Figure 12.2** Viewing the contents of a file

The File View File Contents command offers multiple advantages over other ways of viewing a file. To begin with, it's quick; if you can't remember what a file contains, just use this command to have a look.

Another advantage is that you can view just about any file, regardless of its format. As you begin to run applications under DOS, you'll discover that different applications save their data files in different formats. This means that two word processing programs, for example, cannot directly use each other's data files without conversion. With the File View File Contents command, however, you can look at the contents of a file, regardless of which application was used to produce it. Although the file may not be fully legible, you can usually get an idea of what it contains.

◆ *Lesson* ◆

13

# Using the MS-DOS Editor

## Editing ASCII Files

As noted earlier in the book, ASCII is an acronym for American Standard Code for Information Interchange, which is a commonly accepted file format for text files. This format is supported by numerous DOS applications; in fact, most popular DOS word processing, spreadsheet, and database packages are capable of reading and writing files in ASCII format. For this reason, ASCII files are often used as a means by which to share information between DOS applications.

DOS 5.0 comes with its own ASCII file editor, MS-DOS Editor. You can use this program to create, edit, and print simple letters and memos, and to create special files that automate DOS operations. In Lesson 30, for example, you'll learn how to use the MS-DOS Editor to create batch files (files with a .BAT extension and containing DOS commands). You can run a .BAT file just as you would any other program to execute the commands it contains.

## Creating a Memo

The MS-DOS Editor is a snap to use. As an exercise, let's assume you have to dash off a quick memo to one of your associates down the hall. To do this, perform the following steps:

1. Start the MS-DOS Editor by highlighting **Editor** in the program-list area and pressing **Enter**. The dialog box in Figure 13.1 is displayed, prompting you for an ASCII file to load into the Editor.
2. Press **Enter** to bypass the File to Edit dialog box in Figure 13.1. The opening screen of the MS-DOS Editor is then displayed.
3. Press **Esc** to clear the opening prompt. The main Editor screen then appears.
4. Type your memo. (Figure 13.2 shows an example.) To do this, just start typing. When you get to the end of a line, press **Enter**. DOS will move the cursor to the next line.

### *Saving the Memo*

Once you've created your memo, you're ready to save it to disk. To do this, perform the following steps:

1. Type **Alt+F** to open the File menu.
2. Type **a**, which selects the **Save As** option on the File menu. The Save As dialog box in Figure 13.3 is displayed.
3. Type **c:\mymemo.txt** in the File Name text box and press **Enter**. DOS saves the file under the name MYMEMO.TXT in the root directory of drive C (C:\), which now appears at the top of the MS-DOS Editor screen.

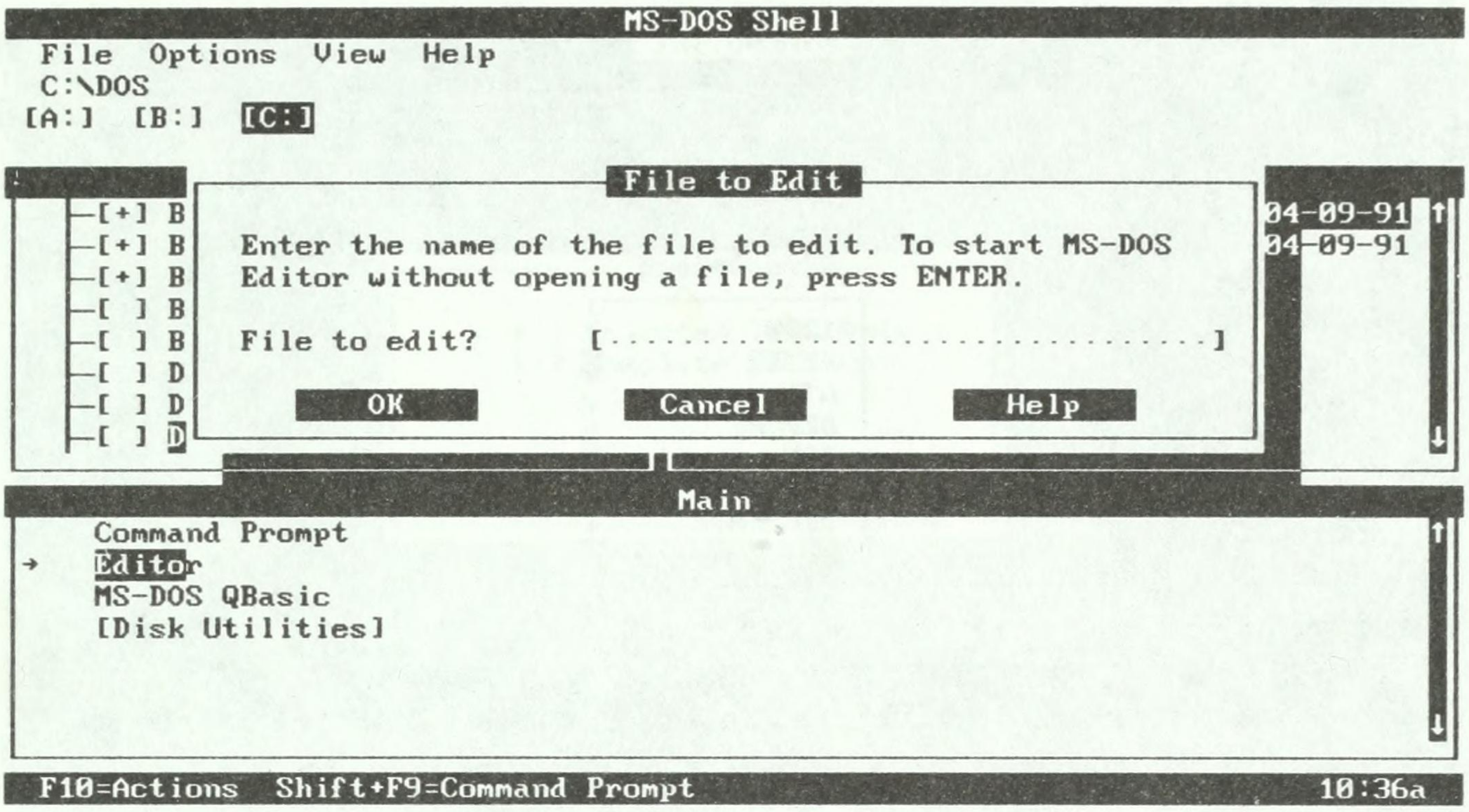



**Figure 13.1** Loading a text file into the Editor

If you want to save the file in a different directory in Step 3, type the path to the directory *in front* of the filename when saving the file. Alternatively, you can select a directory from the Dirs/Drives list in the Save As dialog box and then type a filename in the File Name text box.

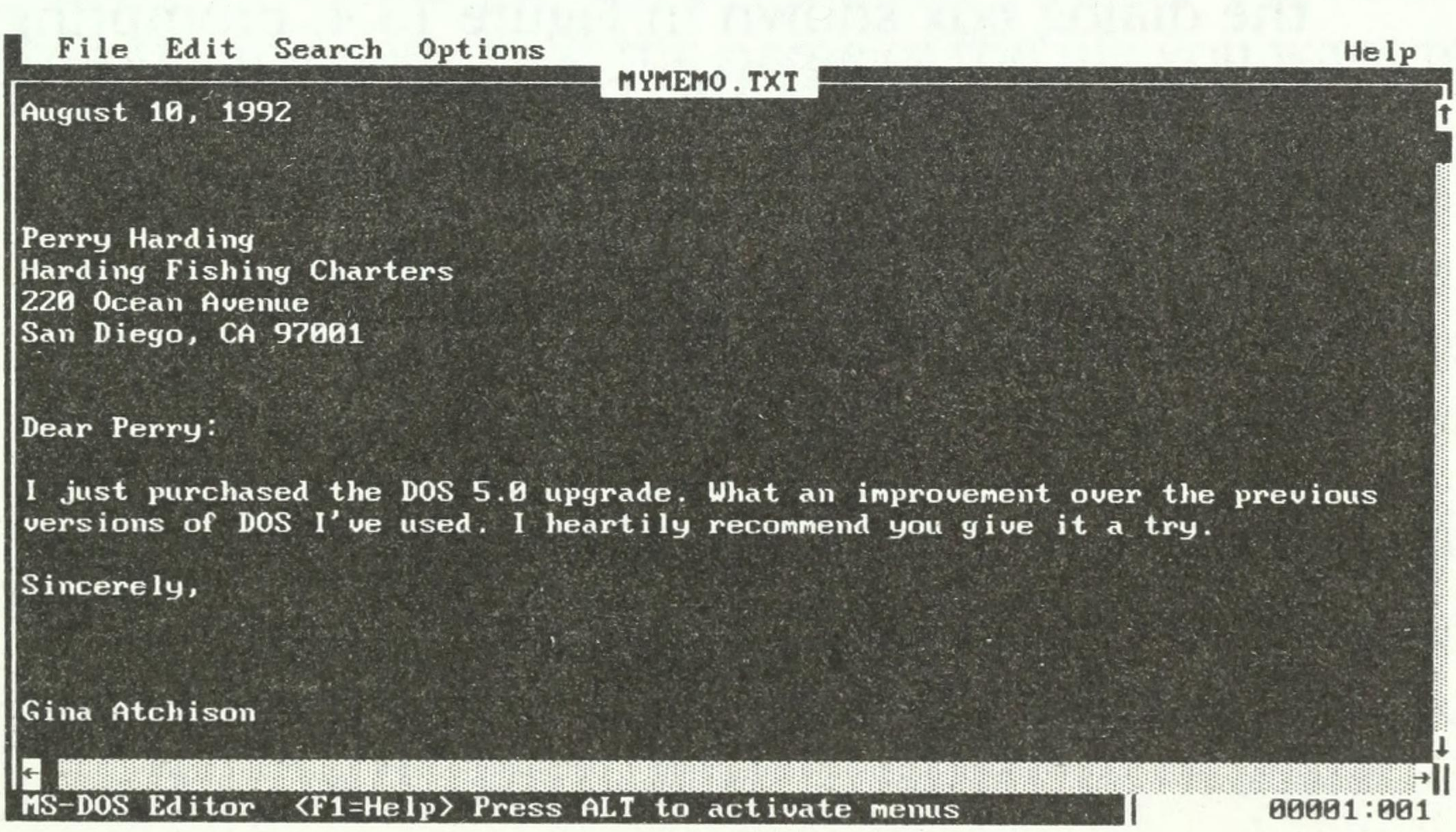



**Figure 13.2** A sample memo in the Editor

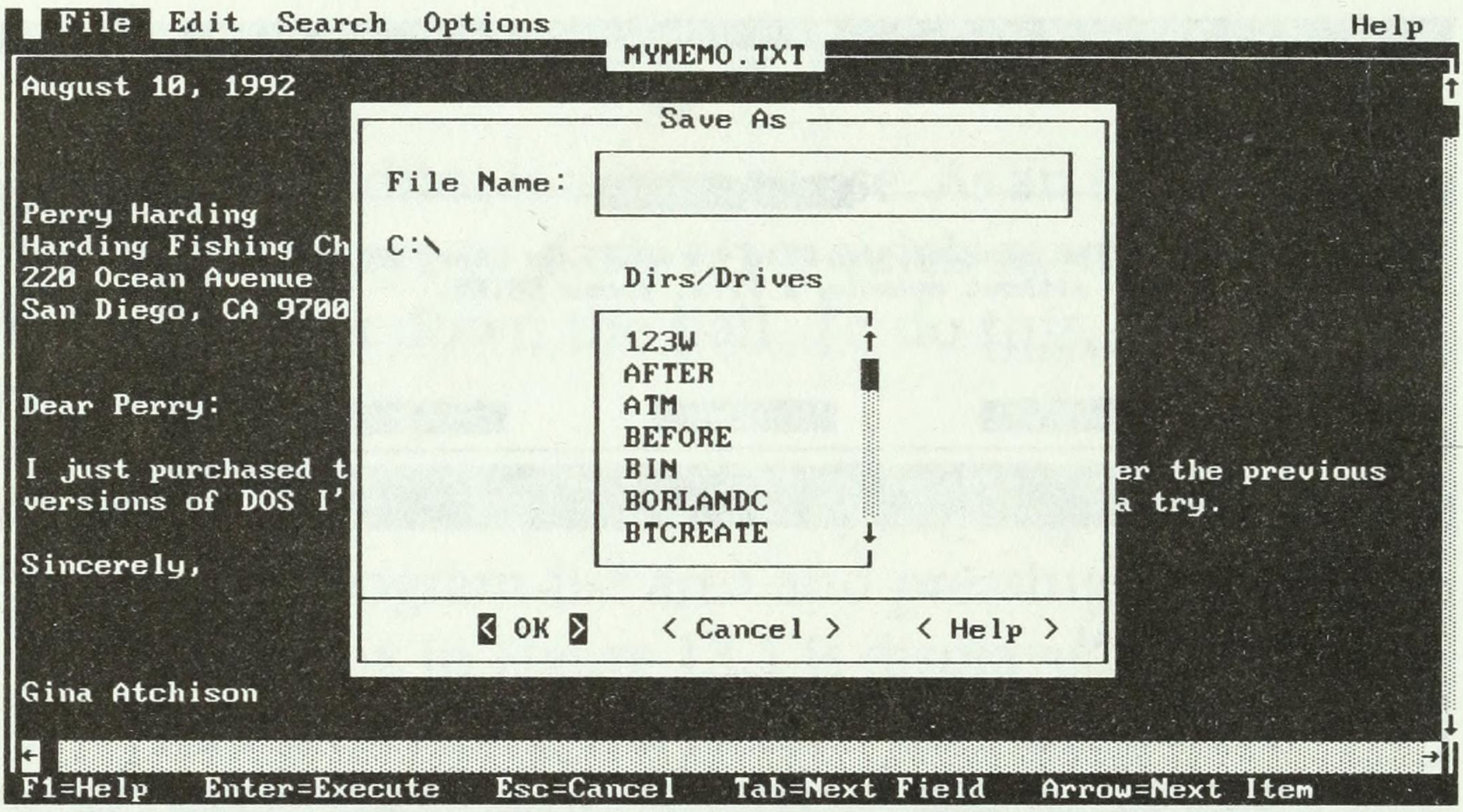



**Figure 13.3** Saving a file

## *Printing the Memo*

Here are the steps to print your simple memo:

1. Type **Alt+F** to open the File menu.
2. Type **p** to select the Print option. Editor displays the dialog box shown in Figure 13.4, prompting you to print either the complete document or just the currently selected text (complete document is the default).
3. Press **Enter** to print the complete document.

⇨ ***Note:*** The MS-DOS Editor's File Print command works only if your printer is turned on and connected to the first parallel port (LPT1).

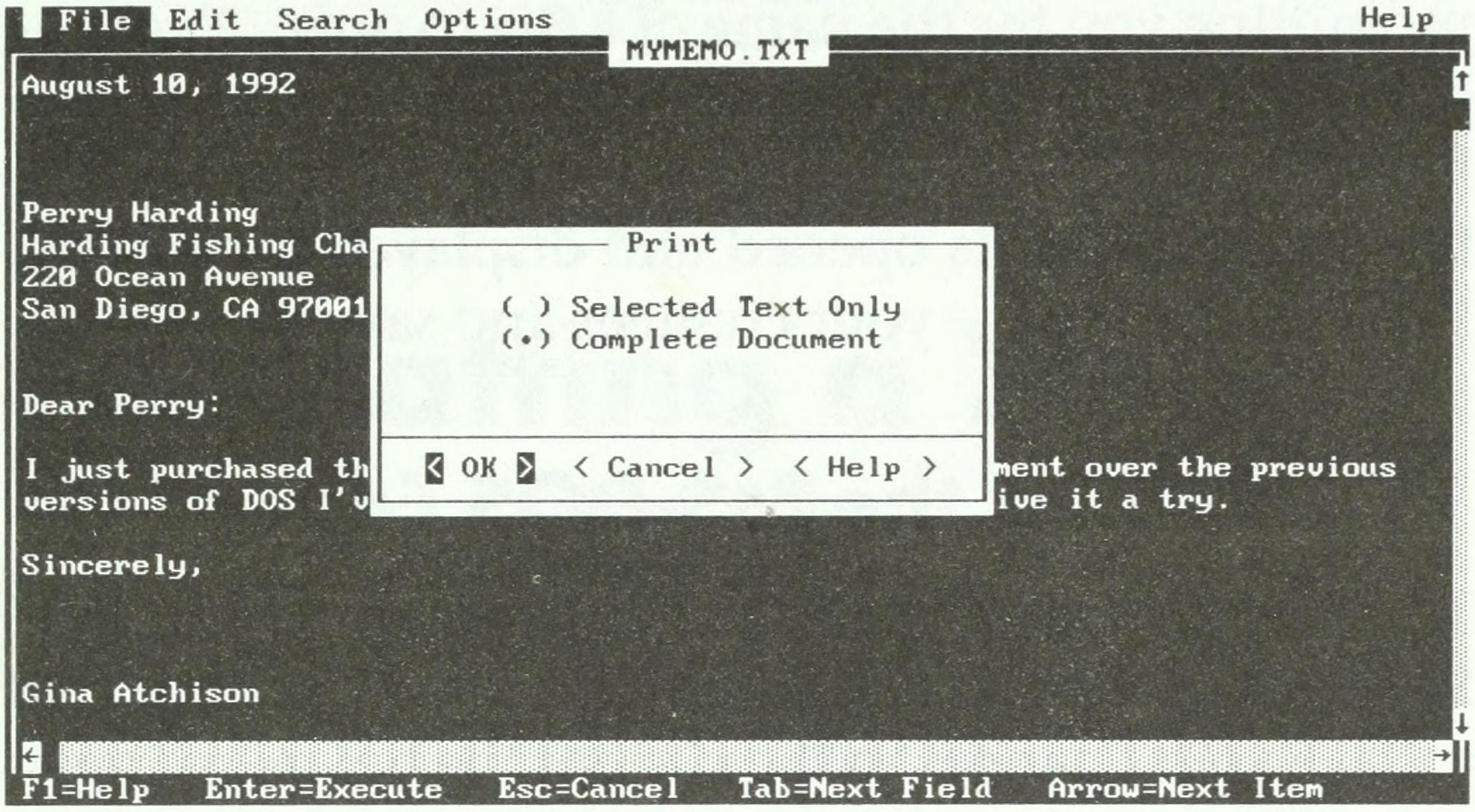



**Figure 13.4** Printing a file

## *Editing the Memo*

Once a text file has been saved to disk, you can start the MS-DOS Editor at any time and edit the file. If you start the MS-DOS Editor by selecting Editor in the DOS Shell program-list area, the dialog box shown earlier in Figure 13.1 is displayed, prompting you for the name of a file to open. Type the name of the file you want to edit in the text box, including its path—for example, c:\mymemo.txt—and press Enter. The MS-DOS Editor is started and the file you specified is loaded automatically. You can then make your changes. When you're done, select the File Save command to save your changes to disk.

If you're already working in the DOS Editor, you can open a file by using the File Open command. When you select this command, Editor displays a dialog box,

prompting you for the name of a file to open. Type the name of the file you want to edit including its path—for example, c:\dos\readme.txt—and press Enter. The file you specified is opened and displayed for editing. You can then make your changes and save the file.

## Exiting the MS-DOS Editor

To leave the MS-DOS Editor and return to the DOS Shell, select the File Exit command. If you've neglected to save your changes to the current file before trying to exit, Editor will notify you of this and give you a chance to save your changes before exiting.

◆ *Lesson* ◆

# 14

# Formatting a Floppy Disk

## Why Format?

Before you can use a floppy disk (that is, create directories on it or copy files to it), the disk must be formatted. To do this, use the DOS Format utility (FORMAT.EXE), which creates a root directory on the disk and a file allocation table (a special type of file that tells DOS where to look on the disk for files and directories).

To test whether the disk in a floppy drive has been formatted:

1. Press **Tab** to move to the drive-icons area of the DOS Shell screen.
2. Highlight the appropriate drive letter—for example, **A:**—and press **Enter**.
3. If the disk has been formatted, the directories and files on the disk will be displayed; otherwise you'll see the **General Failure** message shown in Figure 14.1, indicating the disk needs formatting (or that the disk has not been properly placed in the drive). Press **Esc** to clear the message.

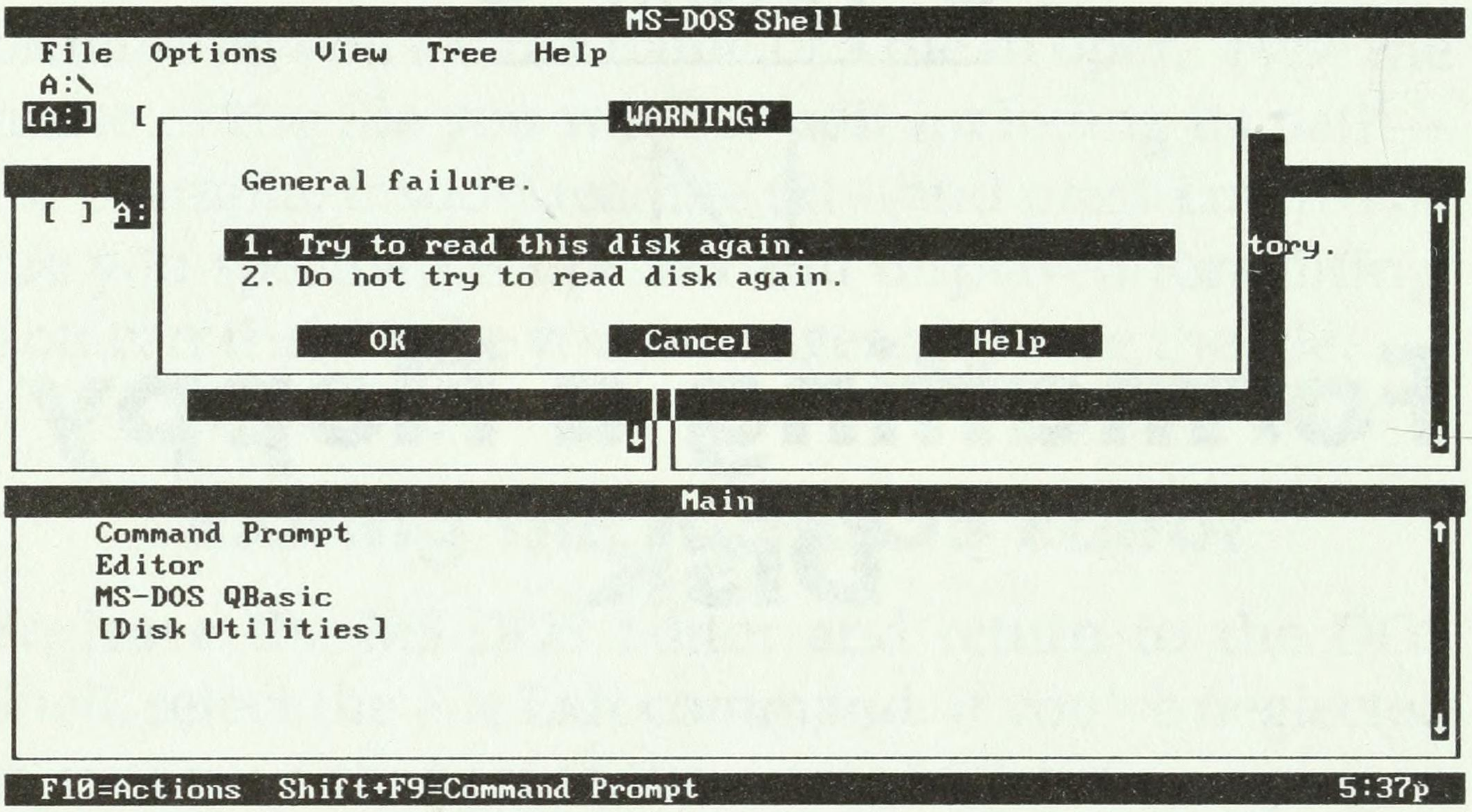



**Figure 14.1** A General Failure message

Formatting erases all the files and directories that are already on a floppy disk. Therefore, make sure the disk you are about to format is either completely blank or does not contain files you want to save.

## Formatting a New Disk

In preparation for formatting a new disk, insert the disk into a floppy drive (usually drive A for 5.25 inch disks or drive B for 3.5 inch disks). Then:

1. Activate the **program-list area**.
2. Highlight the **[Disk Utilities]** group option and press **Enter**.
3. Highlight the **Format** option and press **Enter**. The DOS Shell displays the dialog box shown in Figure 14.2, suggesting drive A (a:) as the drive to format.

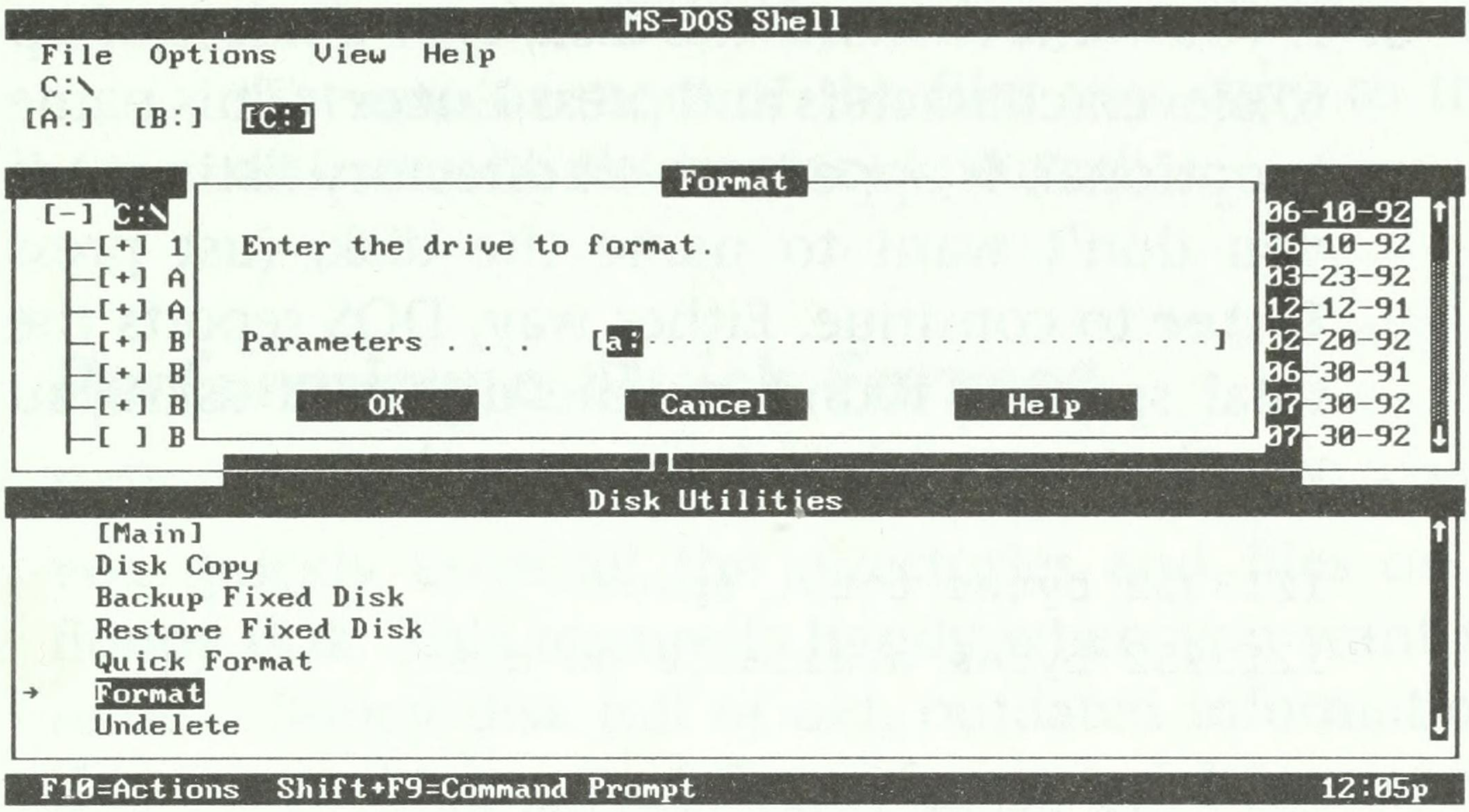



**Figure 14.2** Formatting a floppy disk

**4.** Type the letter of the drive containing the disk you want to format followed by a colon—for example **a:** or **b:**—and press **Enter**. Your screen will go blank, and you'll see the following instructions displayed:

```
Insert new diskette for drive A
and press ENTER when ready...
```

**5.** Press **Enter** to proceed with formatting. DOS reports its percentage of completion as it formats the disk (this takes a while). When formatting is complete, DOS prompts you for a volume label (a master name) for the disk as follows:

```
Checking existing format.
Formatting 1.2M
Format complete.

Volume label (11 characters, ENTER for None)?_
```

**6.** If you want to name the disk, type a name of up to eleven characters and press **Enter**. (This name is optional. It appears in all directory listings.) If you don't want to name the disk, just press **Enter** to continue. Either way, DOS reports the total space it found on the disk and asks you whether you wish to format another:

```
1213952 bytes total space
1213952 bytes available on disk

        512 bytes in allocation unit.
        2371 allocation units available on disk.

Volume Serial Number is 357E-13D0

Format another (Y/N)?
```

**7.** If you have another disk you want to format, type **Y** and press **Enter**; DOS will take you through another formatting cycle; otherwise, type **N** and press **Enter**. DOS displays the following:

```
Press any key to return to MS-DOS Shell....
```

**8.** Press any key. The DOS Shell is once again displayed on your screen.

You may occasionally run across a bad disk. When this happens DOS will warn you by reporting fewer available bytes than the total disk space. In addition, you might see a message like

```
7168 bytes in bad sectors
```

This means a small portion of the disk will not store data reliably. We recommend that you

*don't use* this disk. If you do, you take the risk that one or more of the files you copy to the disk may be not be stored properly.

## Performing a Quick Format

DOS 5.0's convenient new Quick Format feature lets you quickly erase all the directories and files on a floppy disk. This feature is handy when you want to reuse a floppy disk full of old, outdated information that you no longer need. To perform a quick format on a floppy disk, take these steps:

1. Insert the disk in a floppy drive.
2. Activate the **program-list** area.
3. Highlight the **[Disk Utilities]** group option and press **Enter**.
4. Highlight **Quick Format** and press **Enter**. The DOS Shell displays the dialog box shown in Figure 14.3, suggesting drive A (a:) as the drive to quick format.

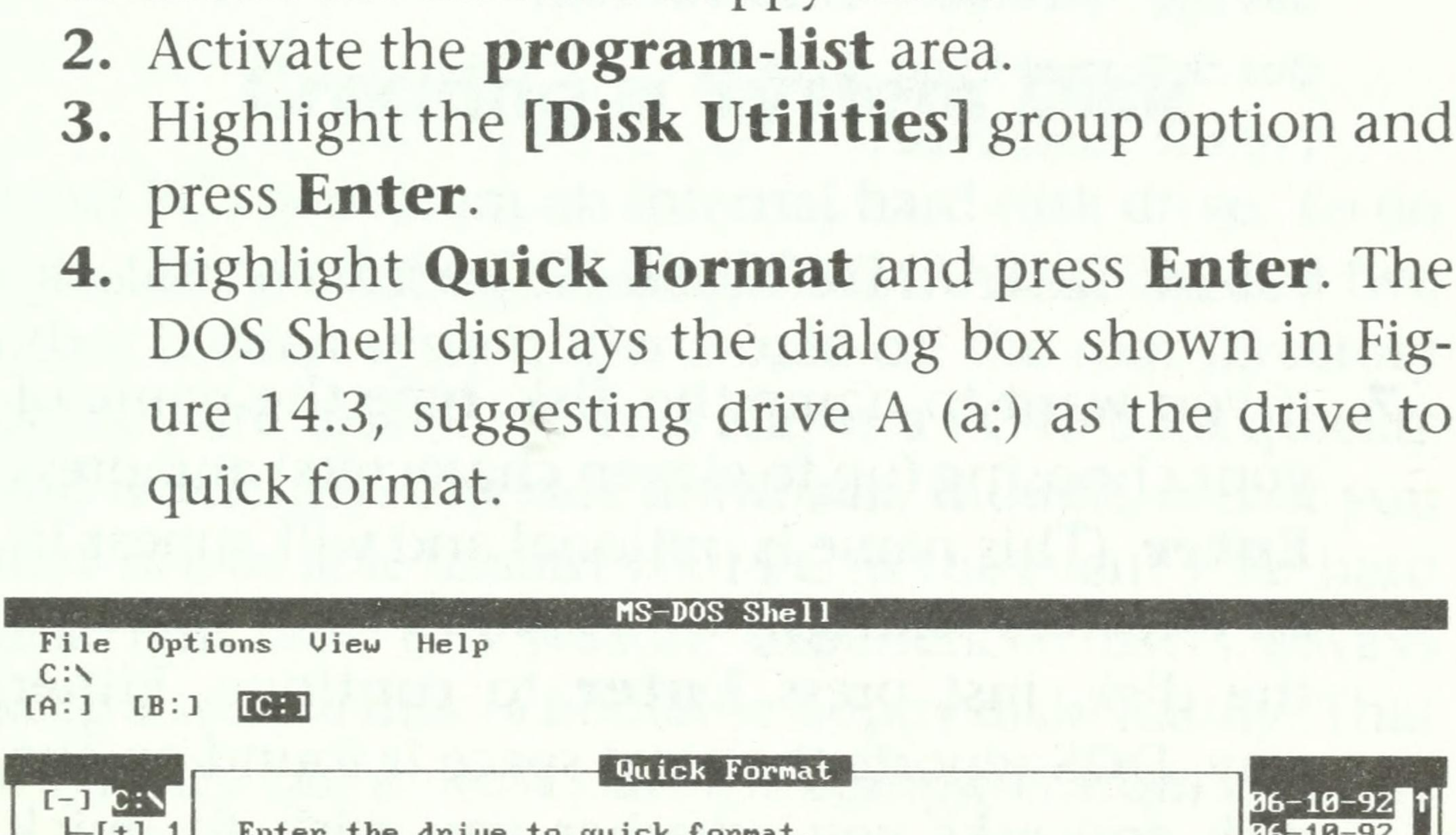



**Figure 14.3** Quick formatting a floppy disk

**5.** Type the letter of the drive containing the disk you want to quick format followed by a colon—for example **a:** or **b:**—and press **Enter**. Your screen will go blank then display the following instructions:

```
Insert new diskette for drive A
and press ENTER when ready...
```

**6.** Press **Enter** to proceed with quick formatting. DOS reports its progress and then prompts you for a volume label for the disk as follows:

```
Checking existing disk format.
Saving UNFORMAT information.
QuickFormatting 1.2M
Format complete.

Volume label (11 characters, ENTER for None)?_
```

**7.** If you want to name the disk, type the name of your choosing (up to eleven characters) and press **Enter**. (This name is optional and will appear in all directory listings.) If you don't want to name the disk, just press **Enter** to continue. Either way, DOS reports the total space it found on the disk and asks you whether you wish to quick format another.

```
1213952 bytes total space
1213952 bytes available on disk

        512 bytes in allocation unit.
        2371 allocation units available on disk.
```

```
Volume Serial Number is 357E-13D0

QuickFormat another (Y/N)?
```

**8.** If you have another disk you want to quick format, type **Y** and press **Enter**; DOS will take you through another quick format cycle; otherwise, type **N** and press **Enter**. DOS displays the following:

```
Press any key to return to MS-DOS Shell....
```

**9.** Press any key. The DOS Shell is once again displayed on your screen.

## Creating a System Disk

Most PCs boot from an internal hard-disk drive. To do this, they read the DOS file COMMAND.COM and two other hidden system files stored on the root directory of the hard disk (C:\). This allows a PC to boot quickly and efficiently. The one drawback, though, is that you may not be able to start your PC in the event your hard disk fails. For this reason, experienced users always keep a system disk (a bootable floppy disk) handy. That way, they can at least start the computer from a floppy disk drive and possibly recover some of the data on the failed hard disk.

To create a system disk, you must once again use the DOS Format utility. To do this, follow the steps outlined above under "Formatting a New Disk," except that in step 4, when DOS prompts you for a drive to format, type the letter of the target drive followed by a

colon, a space, a slash (/), and the letter s—for example, **a: /s**. When you press Enter to confirm this entry, DOS will take you though a normal formatting session. During the session, you'll see the message **System Transferred**. Once the system disk has been created, put it in a safe place. Then, if your hard disk ever fails, you can place the disk in drive A and restart your PC. Your PC will then boot from drive A rather than the failed hard disk, drive C.

◆ *Lesson* ◆

15

# Copying Floppy Disks

## Understanding Disk Copy

The Disk Copy utility (DISKCOPY.EXE) lets you create an exact duplicate of a floppy disk, and is often used to make backup copies of important floppy disks. Most software vendors recommend, for example, that you make backup copies of the disks on which their program is shipped in case the originals are ever damaged. The Disk Copy utility is ideally suited for this purpose.

Because the Disk Copy utility makes an *exact*—track for track—copy of a floppy disk, it can only be used to copy information between two disks of the same type (same size and amount of space available). This means you can't use Disk Copy to copy information between a 5.25 inch, 1.2 Mb disk and a 3.5 inch, 1.44 Mb disk, for instance.

Although the Disk Copy utility does not affect the source disk (the disk that is being copied), it does erase all data on the destination disk (the disk that is receiving information). Therefore, make absolutely sure the

destination disk is empty or does not contain any information you want to keep. If the destination disk is unformatted, DOS will format it for you during the Disk Copy procedure.

For this exercise, let's assume you want to make backup copies of your DOS 5.0 distribution disks. Let's also assume you received the program on three 3.5 inch, low-density (720K) disks. Therefore, to create backups, you'll need to have three blank, 3.5 inch, low-density disks at hand. When you're ready, perform the following steps:

1. Insert the first DOS disk (labeled **Disk 1-Setup**) into Drive B (your 3.5 inch drive.)
2. Activate the **program-list area**.
3. Highlight the **Disk Utilities** group option and press **Enter**. The program items in that group are now displayed.
4. Highlight the **Disk Copy** option and press **Enter**. The DOS Shell displays the dialog box shown in Figure 15.1, suggesting drive A (a:) as the drive to copy from (source) and drive B as the drive to copy to (destination).
5. Using the keyboard, change the source and destination parameters that read **a: b:** to read **b: b:**. When you're done, press **Enter**. Your screen will go blank and then display the following message:

```
Insert SOURCE diskette into drive B:
Press any key to continue. . .
```

6. Press **Enter** in response to this prompt. DOS displays a status message and begins copying the information on the source disk into your

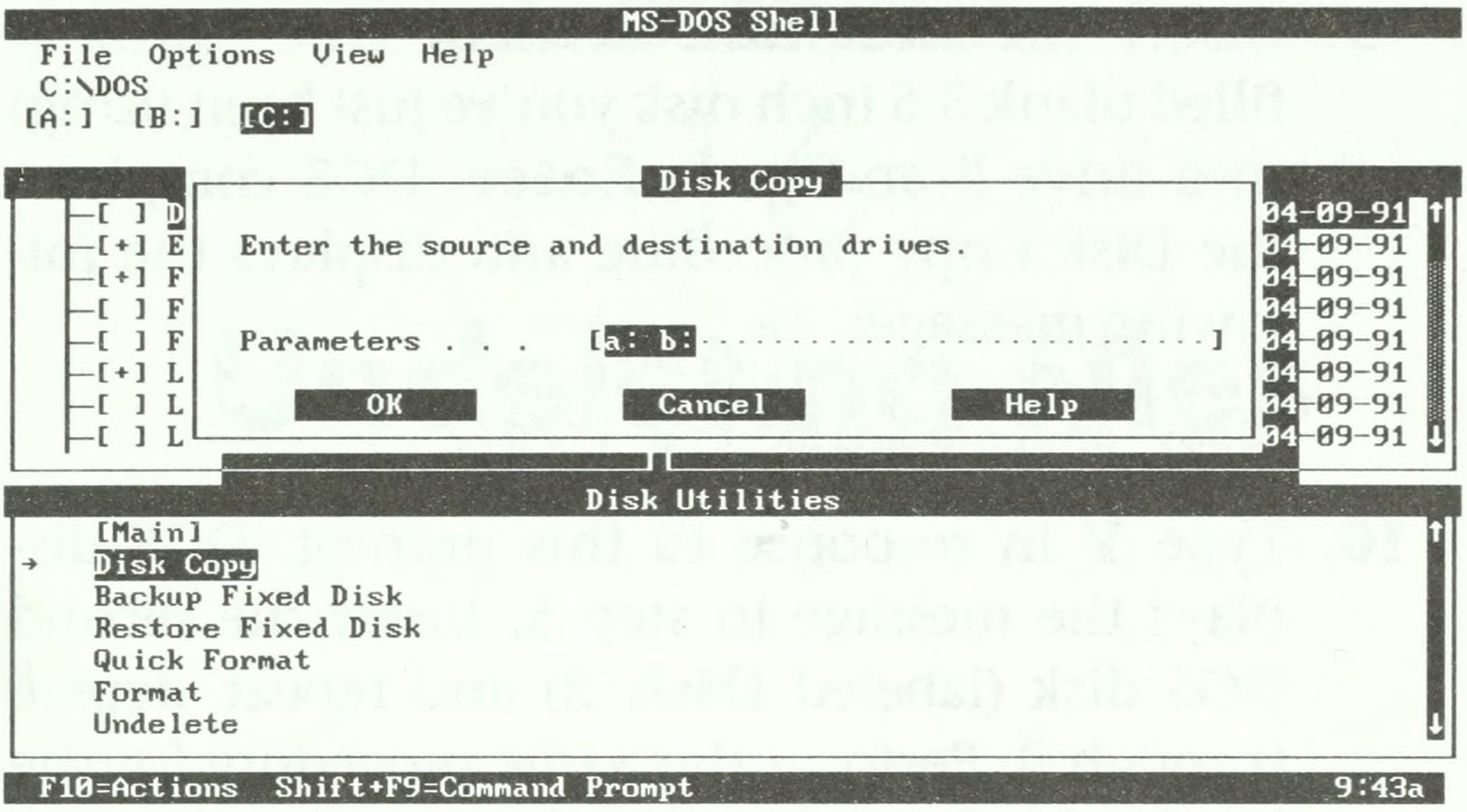



**Figure 15.1** Using the Disk Copy utility

computer's RAM. When RAM is full, DOS displays the message:

```
Insert TARGET diskette into drive B:
Press any key to continue. . .
```

**7.** Insert the **destination disk** (one of your blank 3.5 inch disks) into drive B and press **Enter**. DOS copies the information in RAM onto the destination disk. When this is done, DOS displays the message:

```
Insert SOURCE diskette into drive B:
Press any key to continue. . .
```

**8.** Insert the **source disk** (labeled **Disk 1**) into drive B and press **Enter**. DOS copies the rest of the information on the disk into RAM. When this is done, DOS displays the message:

```
Insert TARGET diskette into drive B:
Press any key to continue. . .
```

**9.** Insert the **destination disk** (the partially filled blank 3.5 inch disk you've just been using) into drive B and press **Enter**. DOS completes the Disk Copy procedure and displays the following message:

```
Copy another diskette (Y/N)?
```

**10.** Type **Y** in response to this prompt. DOS displays the message in step 5. Insert the second DOS disk (labeled **Disk 2**) and repeat steps 6 through 9. Perform this same procedure for the third DOS disk (labeled **Disk 3**). When the Disk Copy procedure for the third source disk is complete, Type **N** in response to the message in step 9. DOS displays the message:

```
Press any key to return to MS-DOS Shell....
```

**11.** Press any key. The DOS Shell screen is once again displayed.

◆ *Lesson* ◆

16

# Undeleting Files

## Understanding Undelete

Eventually you'll come to know that sinking feeling that comes from deleting a file unintentionally. When this happens, though, you generally don't have to worry about it; DOS 5.0 comes with an Undelete utility that lets you quickly recover a deleted file.

It is important to understand that the Undelete utility is intended for use with files that have been deleted, not with files that have been overwritten. So, if you highlight the name of a file in the DOS Shell file-list area and then press Del to delete it, chances are you can recover that file with Undelete. On the other hand, if you copy one file on top of another, thereby replacing the contents of the destination file with that of the source, you won't be able to recover the information that was in the destination file.

Another important point to be aware of with the Undelete utility is that time is of the essence; use it as soon as you discover you've deleted a file in error. If you wait, DOS may use the same disk area originally

occupied by the deleted file to store new files and directories. When this happens, your chances of recovering all the information in the deleted file are markedly reduced.

The Undelete utility cannot restore a directory that has been deleted. What's more, Undelete cannot restore a file if you've removed the directory that contained it.

When you invoke the Undelete utility within the DOS Shell, it scans the current directory for files that have been deleted. It then indicates how many deleted files were found and how many it can recover. Finally, it shows you each of the files it found in turn and gives you the option of either restoring the current file or moving on to the next one.

## Using Undelete in the Current Directory

The best way to explain how to use the Undelete utility is to delete and then recover a sample file. In this example, we'll delete the file APPNOTES.TXT, which is stored in your C:\DOS directory. (This file contains notes about running different applications under DOS 5.0 and has no impact on the operation of the software.) Once the file has been deleted, we'll recover it with Undelete. Make sure you follow the steps in this example carefully; otherwise, you may lose the file. To work this exercise, first make sure the DOS Shell is running and then follow these steps:

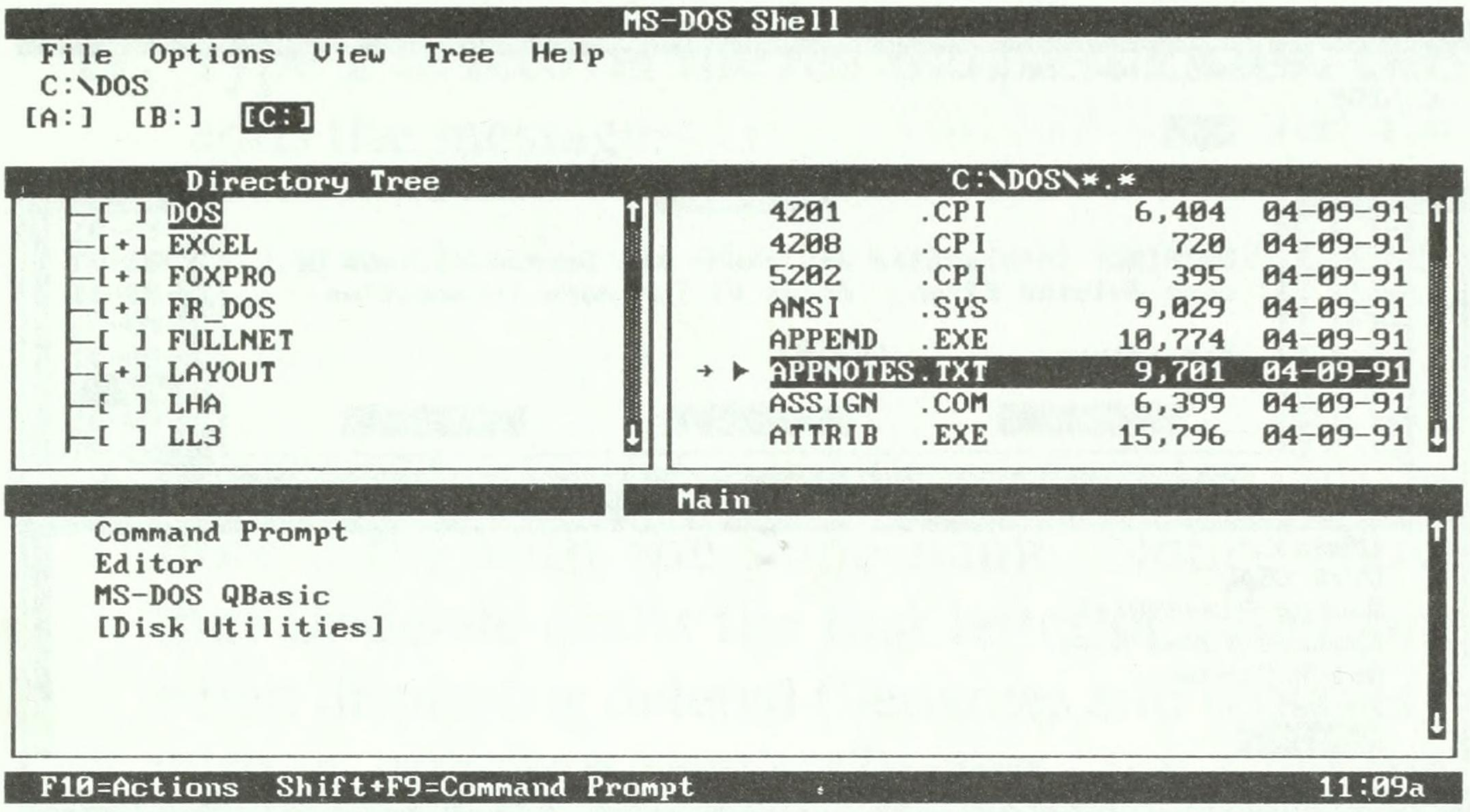



**Figure 16.1** Preparing to delete APPNOTES.TXT

1. Activate the **directory tree**.
2. Make your **C:\DOS** directory current by highlighting it.
3. Press **Tab** to move to the **file-list area** and highlight the file **APPNOTES.TXT**. Your screen should look like Figure 16.1.
4. Press **Del**. The DOS Shell displays the Delete File Confirmation dialog box showing the path and name of the APPNOTES.TXT file—for example, **C:\DOS\APPNOTES.TXT**. (If this filename is not displayed, you've probably selected the wrong filename; in that case, select **No** or **Cancel** to abort the operation and begin again.)
5. Select **Yes** to delete the APPNOTES.TXT file.
6. Without doing anything else, press **Tab** to activate the **program-list area**.
7. Highlight the **Disk Utilities** group and press **Enter** to open it.

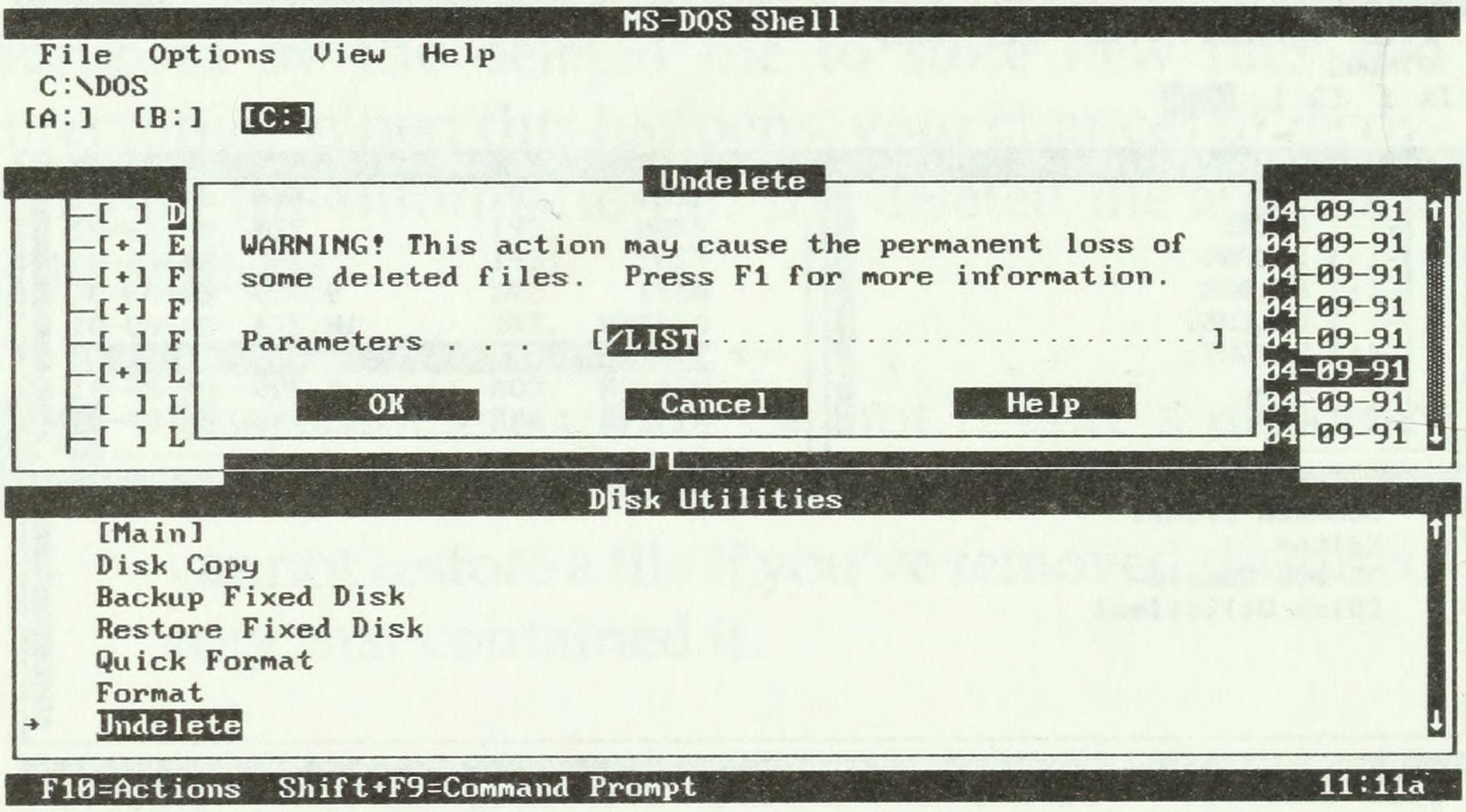



**Figure 16.2** The Undelete dialog box

8. Highlight **Undelete** (the last option) and press **Enter**. The DOS Shell displays the Undelete dialog box in Figure 16.2.
9. Press **Backspace** to delete the **/List** parameter in the Parameters text box. (This is one of four parameters supported by the Undelete utility and will be discussed in Lesson 17.)
10. Press **Enter** to proceed. Your screen will go blank, and then DOS displays the following:

```
Directory C:\DOS
File Specification *.*

     Deletion tracking file not found.

     MS-DOS directory contains   1 deleted files
     Of these,      1 may be recovered.

Using MS-DOS directory

     ?PPNOTES.TXT 9701 4-09-91 5:00a ...A Undelete (Y/N)?
```

**11.** Type **Y** to recover the file and press **Enter**. DOS adds the message:

```
Please type the first character for ?PPNOTES.TXT
```

***Note:*** The Undelete utility won't let you restore a file with the same name. Notice above that Undelete omits the first letter of a filename when displaying deleted filenames and replaces it with a question mark (?) sign—for example, ?PPNOTES.TXT. You are then asked to provide the missing letter so that DOS can recover the file. If during recovery, however, DOS detects another file in the same directory with the same name, it displays a warning message and gives you another chance to provide an alternate first letter.

**12.** Type **a**. DOS adds the letter you typed to the beginning of the filename and displays the messages:

```
File successfully undeleted.

Press any key to return to MS-DOS Shell....
```

**13.** Press any key. The MS-DOS Shell screen reappears, but the APPNOTES.TXT file does not appear in the file list.

**14.** Press **Tab** as necessary to activate the **directory tree**, change to another directory, and then change back to the DOS directory. (This will refresh the file list.) The APPNOTES.TXT file once again appears in the list.

## USING UNDELETE OUTSIDE THE CURRENT DIRECTORY

Note in the previous exercise, that you were asked to make your C:\DOS directory current before using the Undelete utility. Although this is a good idea, it's not entirely necessary; you can use the Undelete utility to recover files located outside the current directory. All you need to do is supply the appropriate path to the directory whose files you want to restore. For example, imagine the root directory (C:\) is current, but that you want to recover the deleted files in C:\DOS. To do this, invoke the Undelete utility as described in step 8 above. When the Undelete dialog box (Figure 16.2) is displayed, type **c:\dos** and press **Enter**, then follow the balance of the steps as given.

## *Selective Use of Undelete*

You can also be more selective with the Undelete command through the use of wildcard characters. For example, suppose you want to recover only those files in your C:\DOS directory that have a .TXT extension. To do this, invoke the Undelete utility as described in step 8 above. When the Undelete dialog box (Figure 16.2) is displayed, type **c:\dos*.txt** and press **Enter**, then follow the balance of the steps as given.

◆ *Lesson* ◆

17

# The Undelete-Mirror Utilities Team

## How Undelete and Mirror Work Together

The Undelete utility works in conjunction with another utility, Mirror (MIRROR.COM). When active, Mirror detects each file deletion and records the information needed by Undelete to reliably recover the file. This recovery information is stored in a separate file located on the root directory of the drive for which Mirror has been activated. In this way, Mirror maintains the equivalent of a deletion tracking system that is far more sophisticated than DOS alone can provide.

Undelete also supports four parameters or switches that let you control its operation. For example, you can get a list of all the files in a directory that have been deleted without recovering them; or, you can have all the files in a directory recovered automatically without prompting for confirmation.

# Setting Up a Deletion Tracking System

To set up a deletion tracking system for the Undelete utility, you'll need to run Mirror for a specific disk drive. Once activated, Mirror captures a copy of the disk root directory and File Allocation Table (FAT, a special file that tells DOS where directories and files are stored on the disk) each time you delete a file. This information is stored in a separate file named PCTRACKR.DEL (a system file) located on the root directory of the disk for which you have activated Mirror. Undelete will automatically use the information in PCTRACKR.DEL when recovering your files.

To run Mirror for a specific disk drive, you must run it with the **/t** parameter for each disk drive you want to track. To designate a specific disk drive, follow the /t parameter with an appropriate drive letter. For example, to run Mirror from within the DOS Shell for drive C, perform the following steps:

1. Select **File** then the **Run** command. The DOS Shell displays the Run dialog box as shown in Figure 17.1.
2. Type **mirror /tc** and press **Enter**. Your screen will go blank and then display the following information:

```
Creates an image of the system area.

Drive C being processed.

The MIRROR process was successful.

Deletion-tracking software installed.
```

```
The following drives are supported:
Drive C - Default files saved.

Installation complete.

Press any key to return to MS-DOS Shell....
```

**3.** Press any key. The DOS Shell screen appears.

In the preceding example, Mirror affects only drive C. However, you can set up a deletion tracking system for other disk drives as well by including additional /t parameters followed by the appropriate drive letters. To track deletions for disks drives A, B, and C then, you would use the following command line in step 2 of the preceding example:

```
mirror /ta /tb /tc
```

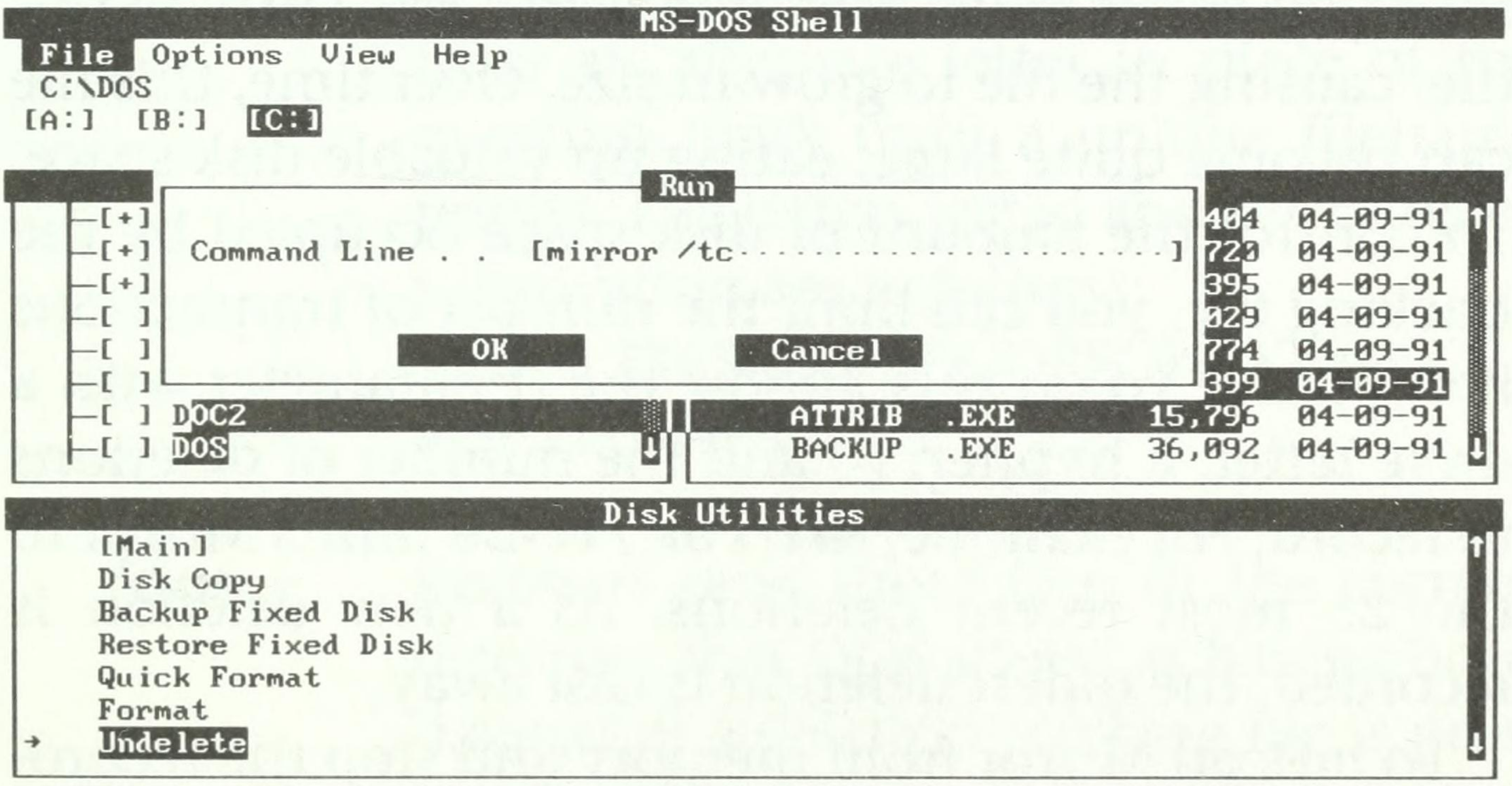



**Figure 17.1** Running Mirror

### Tip

The Undelete utility works even if you have not run Mirror recently. However, Undelete is more effective when the Mirror information file has been kept up to date. For this reason, many users prefer to have Mirror run automatically when their computer is started. You can do this by including the command needed to run Mirror in your AUTOEXEC.BAT file. (DOS reads this file automatically when you start your computer.) Refer to Lesson 32 for more details on entering commands in AUTOEXEC.BAT.

## Tracking File Control

As mentioned, Mirror saves its deletion tracking information in a file (PCTRACKR.DEL) stored on the root directory for which Mirror has been activated. Each time you delete a file, Mirror makes a new entry to this file, causing the file to grow in size. Over time, this file can become quite large, eating up valuable disk space. To control the amount of disk space occupied by the tracking file, you can limit the number of transactions it contains. To do this, follow the /t parameter with a drive letter, a hyphen (-) and the number of deletions to record. For example, **mirror /tc-25** limits Mirror to the 25 most recent deletions. As a new deletion is recorded, the oldest deletion is cast away.

To unload Mirror from memory and stop the recording of deletions, run Mirror with the /u parameter—for example, **mirror /u**.

# Setting Parameters for Undelete

As mentioned, the Undelete utility supports four parameters that control its operation. Here is a list of these parameters and what they do:

**/list** Shows a list of deleted files in the current directory but does not restore them. This is the default parameter when you run Undelete from the DOS Shell.

**/all** Recovers all deleted files in the specified directory without prompting for confirmation. If Mirror's deletion tracking file is present, Undelete will use it; otherwise, all deleted files will be recovered with a question mark as the first letter in the filename. If, upon recovering a file, Undelete detects another file with the same name in the current directory, it substitutes an alternate letter in place of the question mark until a unique filename results. The letters tested and the order of substitution are as follows:
%&-0123456789ABCDEFGHIJKLMNO
PQRSTUVWXYZ.

**/dos** Recovers only those files in the current directory that DOS shows as having been deleted. If a deletion tracking file is present, it will be ignored. You will be asked to confirm the recovery of each file.

**/dt** Only those files listed in the deletion tracking file produced by Mirror will be recovered. You will be asked to confirm the recovery of each file.

◆ *Lesson* ◆

18

# Backing Up Files

## Why Back Up?

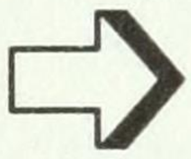
It's always a good idea to back up your directories and files periodically. The term "back up" simply means copying directories and files from one disk to another. Then, if a floppy or hard disk fails, you'll have a backup copy from which you can restore the lost data. DOS comes with two utilities designed for this purpose: Backup (BACKUP.EXE) and Restore (RESTORE.EXE). (The Restore utility is discussed in the next lesson.) The Backup utility is used to back up:

- all the files in a directory,
- all the files in a directory and its subdirectories,
- specific files or groups of files.

## Using Backup

When you back up files with the Backup utility, they are copied to a new location in compressed form. In this form, you can't directly access them. To access

these files, you must use the Restore utility, which decompresses the backed up files and restores them to their former locations, as you'll learn in the next lesson.



**Note:** Use the Backup and Restore utilities only when you have a large number of files to back up and, potentially, restore. Using these utilities would be appropriate, for example, when backing up your entire hard disk. On the other hand, if you have only a few files to back up, it's better to use the File Copy command discussed in Lesson 3; or, if you wish to quickly back up a few files in a selected directory and its subdirectories, use the Xcopy command, which is discussed in Lesson 29.

The Backup utility is frequently used to back up files located on your hard disk to one or more floppy disks in drive A or B. As you'll discover, often the volume of files that are backed up exceeds the capacity of a single floppy disk. When this happens, DOS prompts you to insert an additional disk into the floppy drive as needed. Therefore, before using the Backup utility, make sure you have a sufficient quantity of blank floppy disks on hand. If the disks are unformatted, DOS will format them during the back up procedure.

## Backing Up a Directory

In this section we'll take you through an exercise that shows you how to back up the files in your C:\DOS

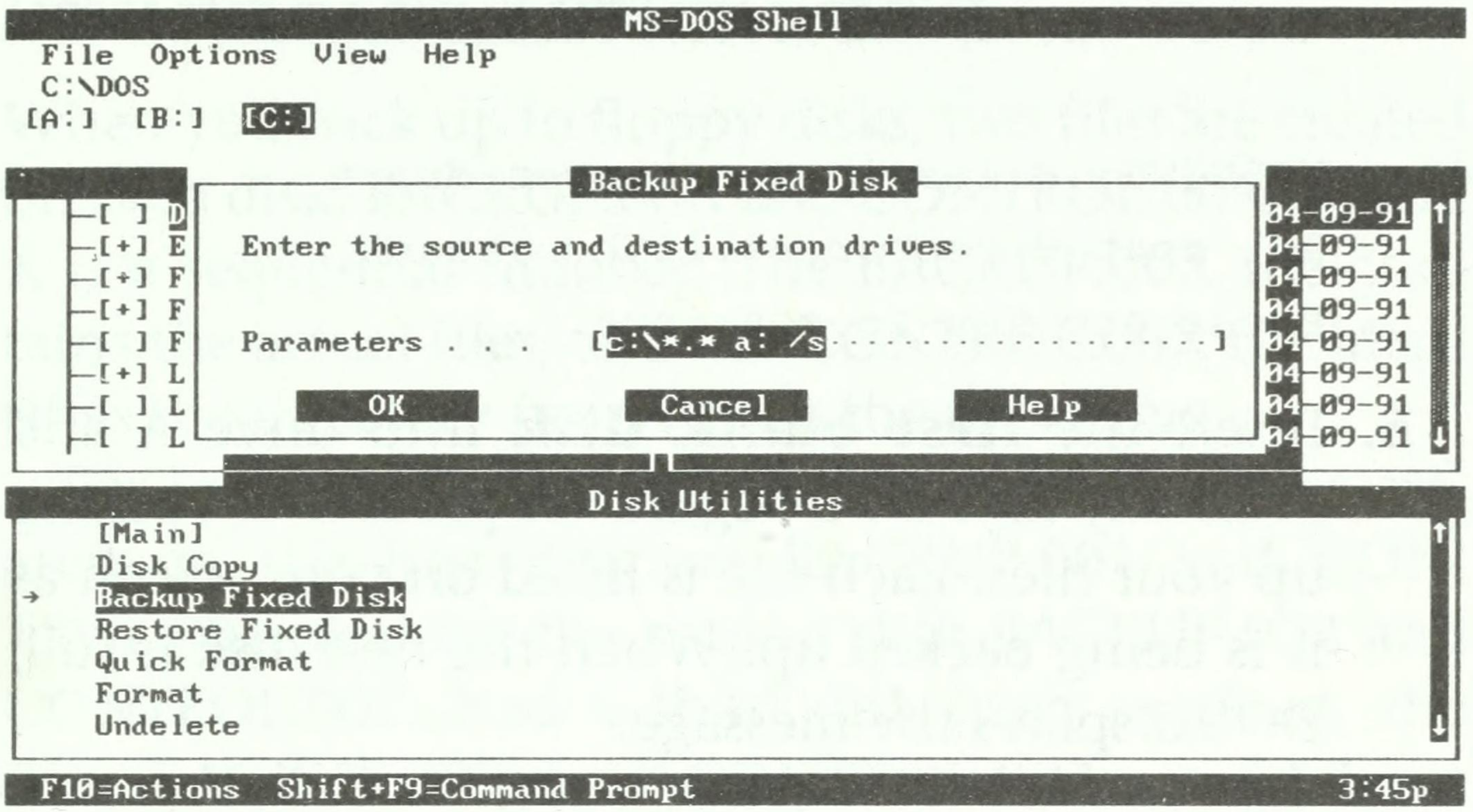



**Figure 18.1** Backing up to a floppy disk in drive A

directory to floppy disks in drive A. To follow along, you'll need two blank 5.25-inch, high density floppy disks. (If your computer uses 3.5-inch disks, use those instead and specify the appropriate drive.) When you're ready, start the DOS Shell if you haven't already and perform the following steps:

1. Activate the **program-list area**.
2. Highlight the **Disk Utilities** option and press **Enter**. The program items in the group are now displayed.
3. Highlight the **Backup Fixed Disk** option and press **Enter**. The DOS Shell displays the dialog box in Figure 18.1.
4. Change the text string **c:*.* a: /s** to read **c:\dos*.* a:**. (We'll discuss what the **/s** parameter means in the next section.) When you're done, press **Enter**. Your screen will go blank and then display the following message:

```
Insert backup diskette 01 into drive A:

WARNING: Files in the target drive
A:\ root directory will be erased
Press any key to continue. . .
```

5. Insert the **first blank disk** into drive A and press any key. DOS begins the process of backing up your files. Each file is listed on your screen as it is being backed up. When the first disk is full, DOS displays the message:

```
Insert backup diskette 02 into drive A:

WARNING: Files in the target drive
A:\ root directory will be erased
Press any key to continue. . .
```

6. Insert the **second blank disk** into drive A and press any key. DOS finishes backing up the rest of the files in the directory. When this is done, DOS displays the message:

```
Press any key to return to MD-DOS Shell....
```

7. Press any key. The DOS Shell screen is displayed.

The Backup utility erases all files and directories on the target backup disk. Therefore, make sure the disks you intend to use are either blank or contain data you no longer need. If you've already started the Backup utility and the message in step 4 of the preceding example is displayed, you can stop the Backup utility by pressing Ctrl+C or Ctrl+Break.

## HOW THE BACKUP UTILITY WORKS

When you back up to floppy disks, two files are created on each disk, BACKUP.00X and CONTROL.00X, where X is a sequential number. The BACKUP.00X file contains the actual files, and the CONTROL.00X file identifies the directory from which the files came.

For instance, in the previous example, the files created on the first disk are BACKUP.001 and CONTROL.001, and on the second disk BACKUP.002 and CONTROL.002. Had a third disk been required, the files BACKUP.003 and CONTROL.003 would have been created. In this way, DOS sequentially numbers each backup disk. This numbering is important and you must remember to provide these disks in the same order when using the Restore utility to recover the backed up files. For this reason, its always a good idea to label each disk as to the order in which it was used.

## BACKUP UTILITY PARAMETERS

The Backup utility accepts the following three parameters that control its operation:

- **Source**—The location of the files you want to back up. This can be a drive letter, a directory path, a wildcard pattern, a filename, or a combination of these.
- **Destination**—A drive letter that identifies where the backup files will be created.
- **Switches**—One or more of the switches that control how files are backed up. Table 18.1 shows a list of the more commonly used switches.

**Table 18.1 Commonly Used Switches for the Backup Utility**

| Switch | Description |
|---|---|
| /s | Backs up the files in the current directory and its subdirectories. |
| /m | Backs up only those files that have been changed since the last backup. This switch is often used in conjunction with the /a switch—for example: /a /m. |
| /a | Adds files to an existing backup disk without deleting the files that are already there. |
| /d:*date* | Backs up only those files created on or after a specific date. For example, to back up all files created on or after 10/12/93, use the parameter /d:10-12-93. |
| /l:*file* | Creates an ASCII text file showing a list of the files that were backed up. |

For instance, in step 4 of the preceding example, the parameters supplied for the Backup utility are:

```
c:\dos\*.* a:
```

Here the Source parameter is **c:\dos*.***, which uses a wildcard pattern (*.*) to identify all files in the DOS directory on drive C. (To back up just the .EXE files in this directory, you could use the wildcard pattern *.exe.) The Destination argument is **a:**, which identi-

fies drive A as the drive containing the disk on which the backup files will be created. No Switch parameter has been supplied here. We'll show you an example of how a switch is used in the next section.

***Tip***

You can also use the Backup utility to back up to a hard disk. For example, if you back up to drive C, DOS creates a directory off your root named BACKUP and stores its backup files there.

## *Backing Up a Directory and Its Subdirectories*

To back up a directory and all its subdirectories, use the **/s** switch with the Backup utility. If, for example, you want to back up your entire hard disk to floppy disks in drive A, follow the steps in the exercise in the preceding section until you get to step 4; then, press Enter to confirm the parameters supplied by the DOS Shell. As you may recall, these parameters are **c:*.* a: /s**.

Here, the Source parameter is **c:*.***, which identifies all files in the root directory of drive C. The Destination parameter is **a:**, identifying drive A as the drive containing the floppy disks on which the backup files will be created. Finally, the **/s** switch tells DOS to back up all subdirectories of the directory specified by the Source parameter. Because C:\ identifies the root directory, all directories on your hard disk will be backed up.

◆ *Lesson* ◆

19

# Restoring Files

## Understanding Restore

As explained in Lesson 18, the Restore utility lets you recover files that have been backed up with the Backup utility. Restore always restores files to their original location. It knows where to put each file by reading the CONTROL.00*X* file on the backup disk. (Review the previous lesson if necessary for details on this file.) If the directory from which the files were originally backed up no longer exists, Restore will create it when restoring the files.

During the restoration process, Restore overwrites the files you've specified, including any files that may have changed since your last backup. To prevent this, use the /p switch parameter (see Table 19.1), which prevents Restore from overwriting the more recent versions of your files with older versions. (See the next section for more details on the /p switch and other switches that control the Restore utility.)

# Restoring Files in a Specific Directory

In the previous lesson, you learned how to use the Backup utility to back up the files in your C:\DOS directory. In this lesson, you'll learn how to restore them. (Obviously, these files don't really need to be restored, but it will serve as a good example.)

Recall that in the lesson on the Backup utility, it was suggested that you carefully label each backup disk as to the order in which it was used. Now, as you prepare to restore files with the Restore utility, DOS will prompt you to insert these disks in the same order. Therefore, start the DOS Shell and have the backup disks you created in the previous example handy. When you're ready, perform the following steps:

1. Activate the **program-list area**.
2. Highlight the **[Disk Utilities]** program item and press **Enter**. The program items in the group are now displayed.
3. Highlight the **Restore Fixed Disk** option and press **Enter**. The DOS Shell displays the dialog box shown in Figure 19.1.
4. Type **a: c:** in the Parameters box and press **Enter**. Your screen will go blank and then DOS displays the following message:

```
Insert backup diskette 01 in drive A:
Press any key to continue . . .
```

5. Insert the **first backup disk** into drive A and press **Enter**. DOS displays a message indicating the date when the files were last backed up and

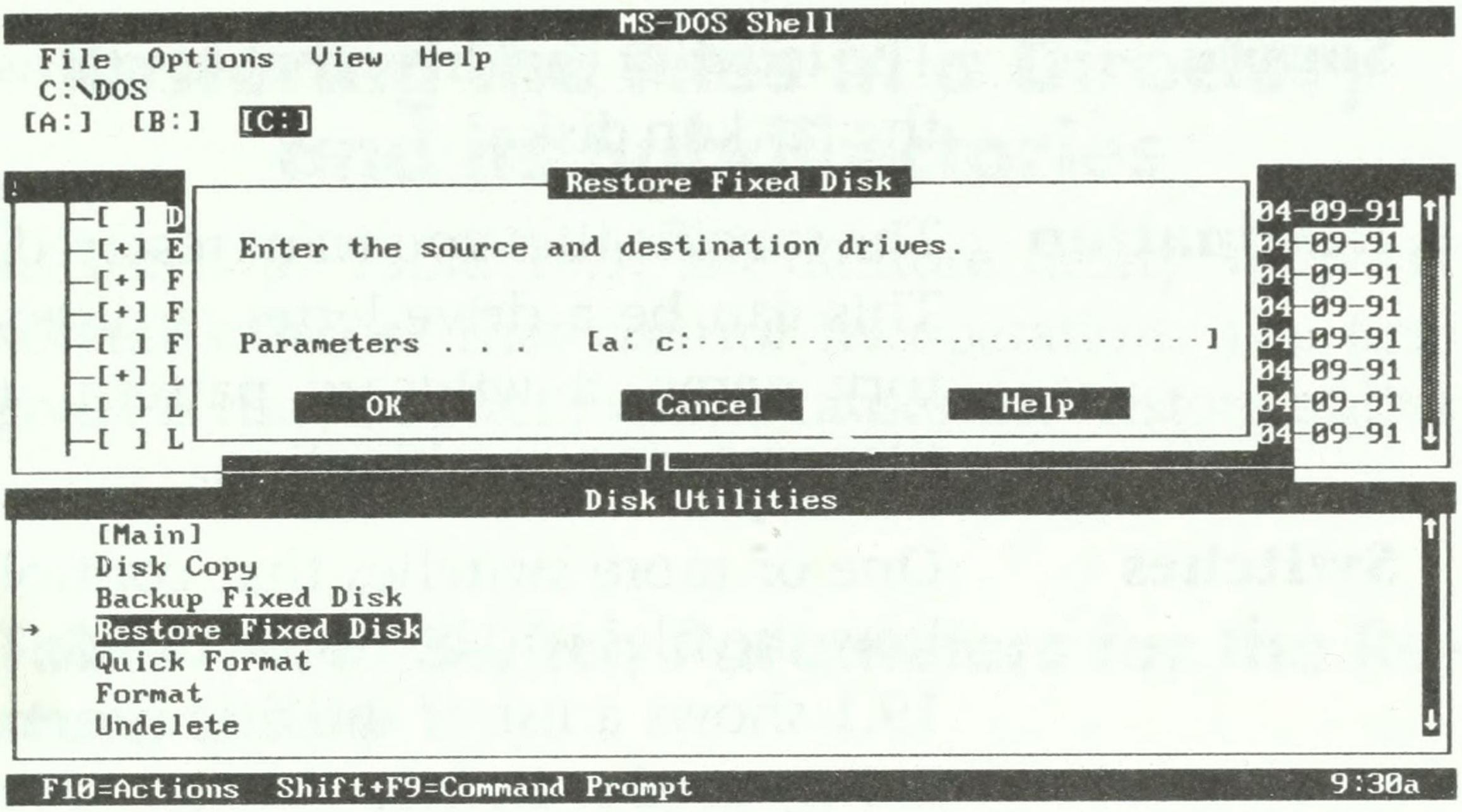



**Figure 19.1** Restore Fixed Disk dialog box

then starts restoring the files from the first disk. When this is done, DOS displays the following message:

```
Insert backup diskette 02 in drive A:
Press any key to continue . . .
```

6. Insert the **second backup disk** into drive A and press **Enter**. DOS restores the balance of the files to your C:\DOS directory. When this is done, DOS displays the message:

```
Press any key to return to MS-DOS Shell....
```

7. Press any key. The DOS Shell screen is displayed.

## *Restore Utility Parameters*

Like the Backup utility, the Restore utility supports three parameters that control its operation:

| | |
|---|---|
| **Source** | The letter of the drive that contains the backup disk. |
| **Destination** | The specific files you want restored. This can be a drive letter, a directory name, a wildcard pattern, a filename, or a combination. |
| **Switches** | One or more switches that control how the files will be restored. Table 19.1 shows a list of the more commonly used switches. |

For instance, in step 4 of the preceding example, you typed the parameters **a: c:** in the Restore Fixed Disk dialog box (Figure 19.1). Here, the Source parameter is **a:**, which defines drive A as the drive containing the backup disk. The Destination parameter is simply **c:**, which defines drive C as the location to restore the files to. Notice that no directory path has been defined here—for example, C:\DOS; this is not necessary. Recall that the Restore utility always returns files to their original locations. Therefore, because the files from C:\DOS are the only files on the backup disk, defining a directory path serves no purpose. On the other hand, if you wanted to restore only those files that have an .EXE extension to the C:\DOS directory, you would specify the Source and Destination parameters **a: c:\dos*.exe** in step 4 of the preceding example.

Also notice that no switch parameter has been used in step 4, causing Restore to use its default settings when restoring files. In the next section, we'll show you an example of how to use a switch parameter with the Restore utility.

# Restoring the Files in a Directory and its Subdirectories

As listed in Table 19.1, the Restore utility supports several switches that control its operation. The first given is the **/s** switch, which causes the Restore utility

**Table 19.1 Switch Parameters for the Restore Utility**

| Switch | Description |
|---|---|
| /s | Restores the files in the current directory and its subdirectories. |
| /p | Prompts you to choose whether to replace files that have changed since your last backup. |
| /m | Restores only those files that have been changed since your last backup. |
| /n | Restores only those files that no longer exist in the destination directory. |
| /a:*date* | Restores only those files created on or *after* a specific date. For example, to restore up all files created on or after 10/12/93, use the parameter /a:10-12-93. |
| /b:*date* | Restores only those files created on or *before* a specific date. For example, to restore all files created on or before 10/12/93, use the parameter /b:10-12-93. |

to restore the files in the current directory and all its subdirectories. To clarify its use, suppose you've previously used the Backup utility to back up all the files on your hard disk to floppy disks in drive A. To restore these same files, use the following parameters with the Restore utility: **a: c:*.* /s**.

Another switch commonly used with the Restore utility is the **/p** switch. When this switch is used, DOS will ask you to confirm the replacement of any files that have been changed since your last backup. So, in order to restore all the files on your hard disk, but also confirm the replacement of files that have been changed, use the following parameters with the Restore utility: **a: c:*.* /s /p**.

◆ *Lesson* ◆

# 20

# Task Switching

## About the Task Swapper

With the aid of the DOS Shell Task Swapper, you can run multiple programs at the same time and switch back and forth between them. You might use this capability, for example, to compose a business letter in a word processing program while at the same time crunching the numbers for the letter in a spreadsheet program.

Before you can start multiple programs, you must turn on the Task Swapper; it is turned off by default. To do so, select the **Enable Task Swapper** option from the **Options** menu. When you select this command, a rectangular dot is placed next to the command name, indicating the Task Swapper is turned on; in addition, a new area is added to lower right-hand corner of the DOS Shell screen named Active Task List. (We'll show you how this area is used in a moment.) To turn off the Task Swapper, simply select the **Options Enable Task Swapper** command a second time.

# Starting More Than One Program

In this section we'll show you how to simultaneously run the programs mentioned above—a word processing program and a spreadsheet program. For the sake of simplicity, we'll use Edit (see Lesson 13) for the word processing program and Quattro Pro for DOS from Borland International for the spreadsheet program. (If you have other programs of your own you would like to run, please feel free to substitute them.)

⇨ ***Note:*** To run a program from the DOS Shell that is not included in the program-list area, you'll need to know the name of its main program (executable) file. For example, the main program file for Quattro Pro DOS is Q.EXE and is found in the C:\QPRO directory.

To simultaneously run the programs mentioned above, perform the following steps:

1. Activate the **program-list area** and move the highlight to the **Editor** program item.
2. Press **Enter** twice to start the **Edit** program without opening a data file. A short time later, Edit appears on your screen.
3. Press **Ctrl+Esc** to return to the DOS Shell. Editor is added to the **Active Task List** area, as shown in Figure 20.1.
5. Activate the **directory tree** and move the highlight to the **C:\QPRO** directory. (If you are running another program, highlight it's directory in this step.)

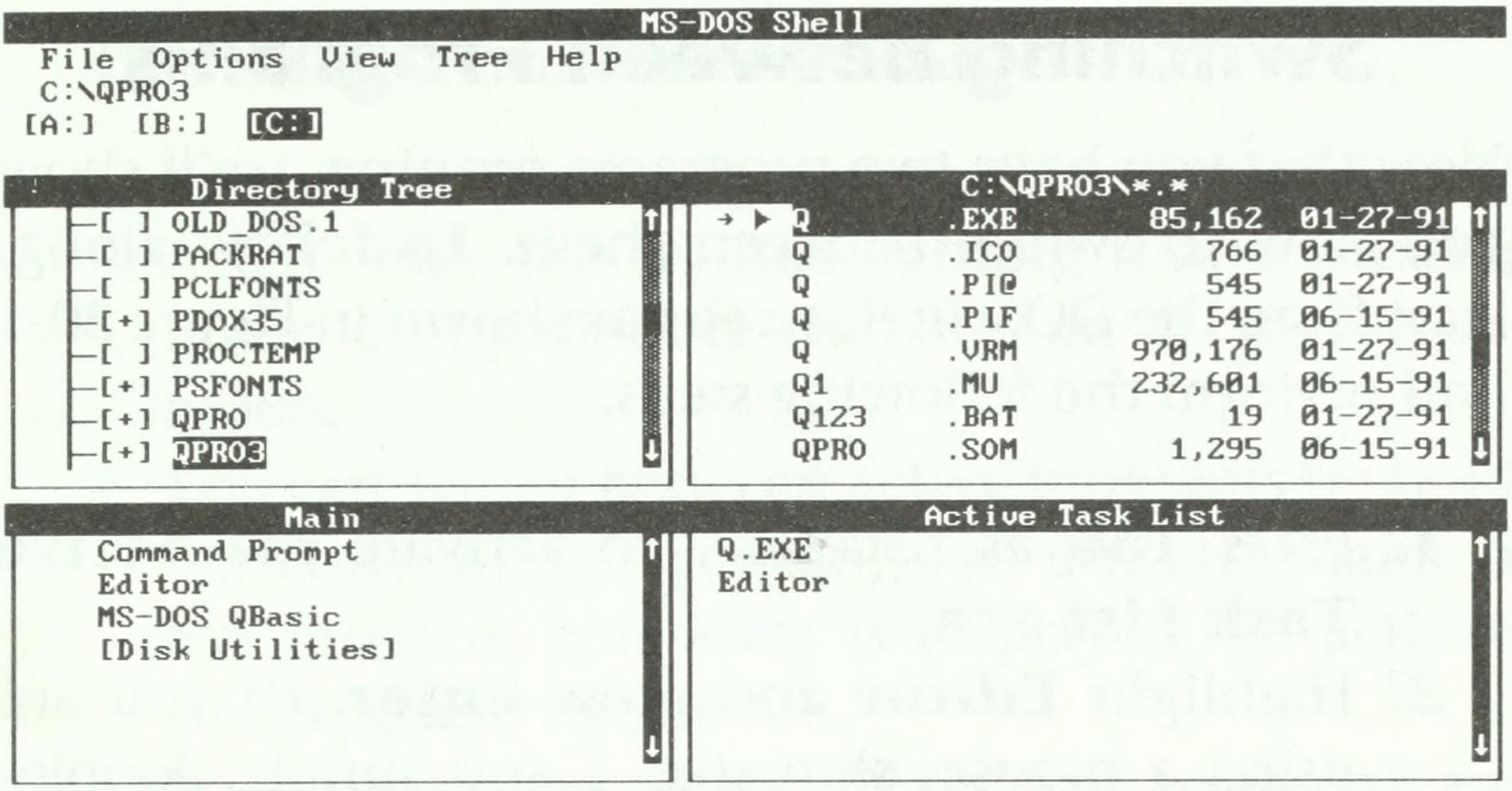



**Figure 20.1** The Active Task List area

6. Press **Tab** to move to the **file-list area** and move the highlight to **Q.EXE**.
7. Press **Enter** to run this program. A short time later Quattro Pro's main screen is displayed.
8. Press **Ctrl+Esc** to return to the DOS Shell. Quattro Pro's file name (Q.EXE) now appears in the Active Task List area along with Editor, as shown in Figure 20.1.

At this point, the two programs—MS-DOS Editor and Quattro Pro for DOS—are running. In the next section, we'll show you how to switch back and forth between them.

---

***Tip***

You can add a program to the Active Task List without running it. To do this, select the appropriate filename and type Shift+Enter. If you are using a mouse, hold down Shift and double-click on the filename.

---

# Switching between Programs

Now that you have two programs running, we'll show you how to switch between them. To follow along, start from the DOS Shell screen as shown in Figure 20.1 and perform the following steps:

1. Press **Tab** as necessary to activate the **Active Task List** area.
2. Highlight **Editor** and press **Enter**. (If you are using a mouse, skip step 1 and simply double-click on Editor.) The DOS Shell screen disappears and is replaced by the MS-DOS Editor screen. Now let's switch to Quattro Pro.
3. Hold down the **Alt** key and press **Tab**. (Continue holding down the Alt key.) DOS displays the name **MS-DOS Shell** in a colored bar at the top of your screen.
4. While still holding down **Alt**, press **Tab** again. DOS displays the filename **Q.EXE** in a colored bar at the at the top of your screen.
5. Release the **Alt** key. DOS activates Quattro Pro and its screen appears.
6. Press **Alt+Tab** again. DOS displays **Editor** in a colored bar at the top of your screen. When you release Alt, the MS-DOS Editor screen is displayed.
7. Press **Alt+Tab** again. DOS displays **Q.EXE** in a colored bar at the top of your screen. When you release Alt, Quattro Pro is activated and its screen is displayed.
8. Press **Ctrl+Esc**. The DOS Shell screen is once gain displayed.

This example demonstrates that:

- You can activate any program in the DOS Shell Active Task List by highlighting it and pressing Enter or by double-clicking on it with your mouse.
- You can switch between active programs by holding down Alt and pressing Tab until the name of the program you want is displayed. To activate the program, release Alt.
- You can return to the DOS Shell screen at any time by typing Ctrl+Esc. Alternatively, you can hold down Alt and press Tab until the name MS-DOS Shell appears, and then release Alt.

## Quitting Programs

To quit a program when you have multiple programs running, switch to the program and use its Exit command to end it. When you do this, your screen will go blank and DOS will display the message:

```
Press any key to return to MS-DOS Shell....
```

When you press a key, the DOS Shell is once again displayed.

***Tip***

If a program locks up when you are running multiple programs, use Alt+Tab or Ctrl+Esc to return to the DOS Shell. Select the program name in the Active Task List and press Del. This ends the program.

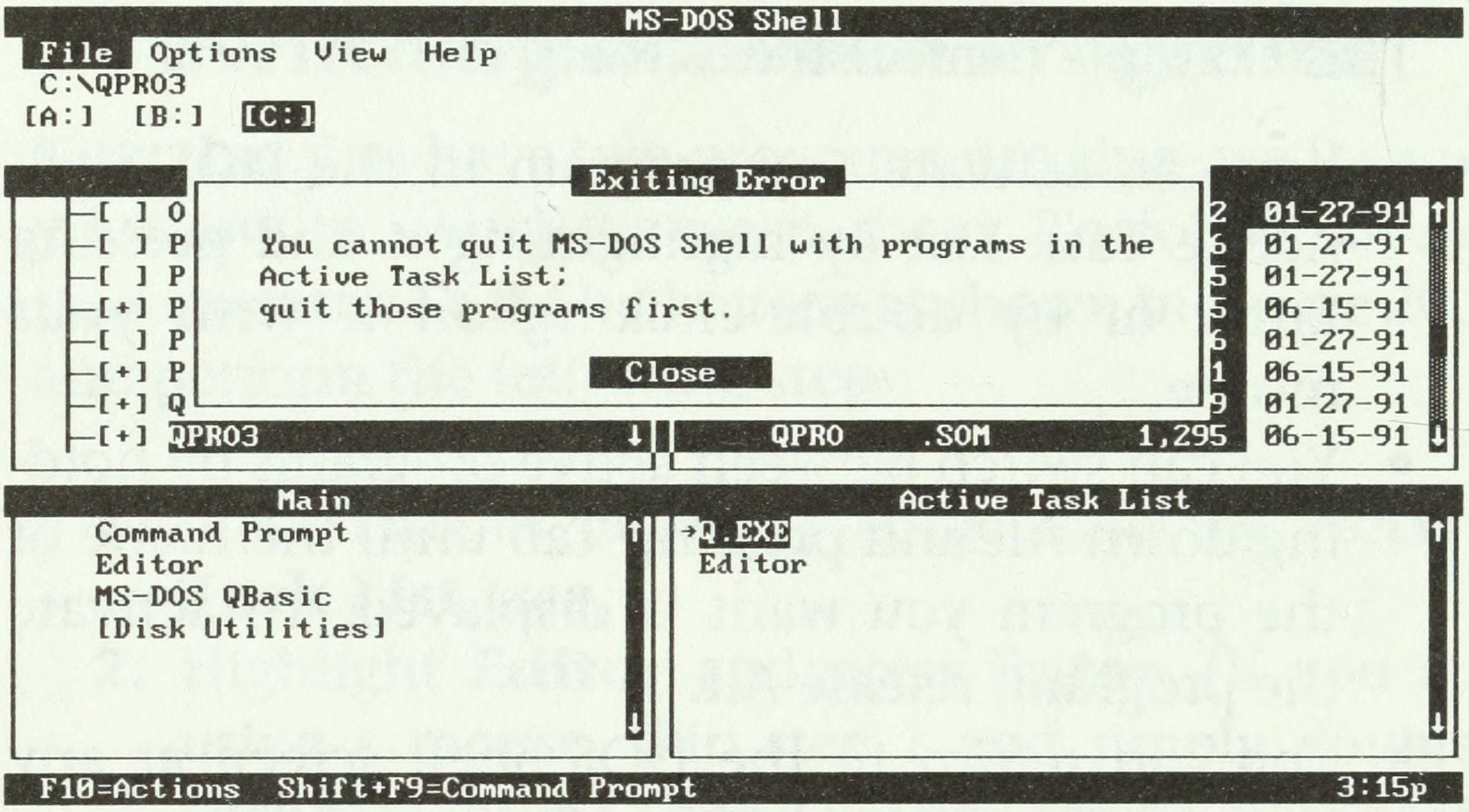



**Figure 20.2** The DOS Shell won't exit when a program is running.

You must quit all programs before leaving the DOS Shell. If you attempt to quit the DOS Shell without first quitting all programs, the error message shown in Figure 20.2 is displayed. Select Close to clear this error message, quit all programs, and then select File Exit (or Alt+F4) to leave the DOS Shell.

◆ *Lesson* ◆

# 21

# Working at the DOS Command Line

## The Purpose of the DOS Shell

Until now, you've learned how to manage the power of DOS by using the DOS Shell. In truth, though, the DOS Shell is something of a facade. When you look behind the scenes, you'll discover that the primary purpose of the DOS Shell is to run the programs located in your C:\DOS directory. Each command you enter or operation you perform in the DOS Shell corresponds to one of these programs.

In fact, you don't need the DOS Shell to run the programs located in your C:\DOS directory. You can run them directly from the DOS command line. Indeed, you'll probably find that the more you use DOS, the more tempted you'll be to circumvent the DOS Shell and work directory from the command line. Most veteran users find this method of interacting with DOS to be easier and more efficient; they also find that their commands tend to execute more quickly.

# Understanding COMMAND.COM

The commands you'll use most often at the DOS command line are contained within (internal to) COMMAND.COM. This file is loaded into memory automatically each time you boot your computer. COMMAND.COM includes most of the file and directory management commands that are the "nuts and bolts" of the DOS command language. They include commands such as Dir, Copy, Rename, and Del, which are discussed in more detail in the sections that follow.

The remaining commands in the DOS command language—including Format, Edit, Diskcopy, and others—are external to COMMAND.COM, and they are stored in the .EXE and .COM files in your C:\DOS directory. We'll also touch upon many of the more commonly used of these in the sections that follow.

# Using Commands

## *Entering a Command*

Remember from Lesson 1 that the first thing you see when your computer has finished booting is the DOS prompt. On most computers, it looks like this:

```
C:\>
```

To enter a command, you simply type the command name and its associated parameters on the line following the prompt. (We'll show you an example of how to do this in a moment.) Pressing Enter executes the command you've typed. When the command has fin-

ished executing, the DOS prompt is again displayed awaiting your next command.

**Tip**

You can access the DOS prompt from within the DOS Shell by choosing "Command Prompt" from the program-list area or by pressing Shift+F9. To return to the DOS Shell at any time, type exit and press Enter.

Let's try entering the Dir command, which displays a list of the subdirectories and files in the current directory. To execute the command, type **dir** at the command line and press **Enter**. When you do this, DOS displays a listing similar to the following:

```
 Volume in drive C is MS-DOS_5
 Volume Serial Number is 1738-8B53
 Directory of C:\

DOS             <DIR>      09-24-91   5:26p
COMMAND  COM        47845 04-09-91   5:00a
AUTOEXEC BAT          297 07-30-92   5:48p
123             <DIR>      09-24-91   6:00p
BIN             <DIR>      09-24-91   6:04p
DBASE           <DIR>      09-24-91   6:04p
OLD_DOS  1      <DIR>      09-24-91   6:11p
WP51            <DIR>      09-24-91   6:31p
HIMEM    SYS        13616 06-07-91   3:10a
MOUSE    SYS        31701 05-01-90   3:00a
CONFIG   SYS          181 07-30-92   1:59p
       11 file(s)      93640 bytes
                    16633856 bytes free
```

Many of the DOS commands, like Dir (above), display either a listing or various messages during their execution. Once the command has finished executing, much of this information will remain displayed on your screen. You can clear your screen of this information by using the Cls command; simply type **cls** and press **Enter**. DOS will clear your screen leaving only the DOS prompt.

## COMMAND SYNTAX

Most DOS commands have three parts: a command name, parameters, and switches. In some cases, the parameters and switches are optional. For example, building on the Dir command example in the previous section, let's suppose you want to get a list of .EXE files in the current directory. But, instead of the single-column list shown in the previous section, you want to see a multiple-column list so that more filenames can appear on your screen. To do this, use the following command:

```
dir *.exe /w
```

The command name is again **dir**, and the parameter for the command is ***.exe**, which is an optional parameter that uses a wildcard pattern to describe only the .EXE files in the current directory. Finally, the last argument for the command is the optional **/w** switch. Notice that a switch begins with a forward slash (/) followed by a letter. This particular switch causes a multiple-column listing to be displayed for the Dir command. Here is the output from this command:

```
 Volume in drive C is MS-DOS_5
 Volume Serial Number is 1738-8B53
 Directory of C:\DOS

NLSFUNC.EXE   SETVER.EXE   DEBUG.EXE   EDLIN.EXE   EMM386.EXE
FASTOPEN.EXE  FDISK.EXE    MEM.EXE     SHARE.EXE   UNDELETE.EXE
 XCOPY.EXE     DOSSHELL.EXE DOSSWAP.EXE PRINT.EXE   RECOVER.EXE
 HELP.EXE      QBASIC.EXE   APPEND.EXE  ATTRIB.EXE  BACKUP.EXE
 CHKDSK.EXE    COMP.EXE     FC.EXE      FIND.EXE    LABEL.EXE
 RESTORE.EXE   SORT.EXE     EXE2BIN.EXE EXPAND.EXE  JOIN.EXE
 REPLACE.EXE   SUBST.EXE    DELOLDOS.EXE
        33 file(s)    1109514 bytes
                      16642048 bytes free
```

## Pausing a Command

You can pause the execution of a command by pressing Ctrl+S or the Pause key. The command will stop executing until you press any key (except Pause). Let's say you just used the Dir command to list a large directory. But when you pressed Enter to activate this command, the names of files and directories scrolled by so quickly you couldn't read them. To pause the command enabling you to read the contents of the list, press the Pause key. DOS will pause the listing until you press any key other than Pause.

## Cancelling a Command

If you change your mind about a command you've just invoked, you can cancel the action at any time by pressing Ctrl+c or Ctrl+Break. Even if the command has already started executing, pressing one of these key sequences suspends further execution. Again, let's as-

sume you just used Dir to get that large directory listing, but then realized you didn't need it. Rather than waiting for the entire listing to be displayed, press **Ctrl+c** in midstream. Further execution of the command will cease and you will be returned to the DOS prompt.

## Repeating the Last Command

DOS supports two special keys that let you easily repeat the last command you've entered at the DOS prompt. These are:

- F1, which enters your last command one letter at a time. (You can achieve the same effect by pressing →).
- F3, which enters all of the characters in the previous command.

You can use these special keys to display the last command, modify it a bit, and then rerun it. For example, suppose your last command was Dir, used to get a list of files in the current directory. Now, however, you would like to see a list of files on your C:\DOS directory. To do this:

1. Press **F3**. DOS displays the last command as: **dir**.
2. Press the **Spacebar** and type **c:\dos*.***. At this point the command line looks like **dir c:\dos*.***.
3. Press **Enter** to execute the command. DOS shows you list of files in your C:\DOS directory.

Now let's suppose you want to see a list of the files in your root directory (C:\). To do this press F1 (or →)

seven times. The command line now reads: **dir c:**. When you press Enter to complete the command, DOS shows you a list of the files in your root directory.

## Using Doskey

Doskey is an innovation in DOS 5.0 that makes the command line much easier to use. Doskey maintains a list of your most recent commands, which you can access quickly to retrieve and reuse an old command. To start Doskey, simply type **doskey** at the command line and press **Enter**.

Once Doskey is running, it starts writing a history of your recent commands in the order in which you've entered them. For example, suppose your last two commands were:

```
cd \dos
dir
```

The first command (cd \dos) makes your DOS directory current. The second command (dir) shows a list of files in that directory. Now, to repeat the last command, press ↑. DOS displays **dir** on the command line, after which you can press Enter to run the command.

Building on this example, let's say you want to access the previous command, cd \dos, modify it to cd \ (which will make your root directory current), and reuse it. To do this:

1. Press ↑ three times. DOS will display **cd \dos** on the command line.
2. Press **Backspace** three times. The command line now reads **cd **.

3. Press **Enter** to execute this command. DOS makes the root directory (C:\) the current directory.

If you can't remember the commands you've entered most recently, press F7, and Doskey will show you a numbered list of your most recent commands.

***Tip***

If you add the word doskey to your AUTOEXEC.BAT file, Doskey will run automatically when you start your computer. (See Lesson 32 for details on how to modify AUTOEXEC.BAT.)

## Getting Help at the Command Line

If you are having difficulty with a command, or you simply want to learn a little more about it, type **help** followed by the name of the command and press **Enter**. DOS will display information that includes a brief description of what the command does, its syntax, and any switches it may support. For example, to get help with the Dir command, type **help dir** and press **Enter**.

◆ *Lesson* ◆

# 22

# Working with Drives and Directories

## Commonly Used Commands

The four most commonly used commands for managing directories at the DOS command line are:

**Chdir (cd)** Changes to a directory, making it current.

**Dir** Enables you to view the contents of a directory.

**Mkdir (md)** Creates a directory.

**Rmdir (rd)** Removes a directory.

We'll discuss each of these commands in more detail in the sections that follow.

## Changing to a Different Drive

You can refer to a specific drive at the DOS command line by using its drive letter followed by a colon (:). To refer to drive A, for example, you would use **a:**.

When you enter a drive letter followed by a colon at the DOS command line and press Enter, DOS changes to that drive, thereby making it current. So, to make drive A current, simply type **a:** and press **Enter**. (Make sure there is formatted floppy disk in drive A before trying this or DOS will display an error message.) To switch back to drive C, type **c:** and press **Enter**.

When you change to a drive for the first time, DOS shows the root directory on that drive as the current directory. If, on the other hand, you have accessed other directories on that same drive during the current session, DOS shows the last directory you accessed on the drive to be the current directory when you change to the drive.

## Changing to a Directory

In Lesson 2, you learned how to change from one directory another by using the DOS Shell. You can do the same thing at the DOS command line by using the Chdir command (cd for short): Type **cd** followed by a space and the path to the directory.

As you may recall, the path to a directory is represented by a list of the directory names that form the connection between the root directory (\) and the directory in question. The directory names in the path are separated by backslashes (\). If your root directory is the current directory on your PC and you want to change to your DOS directory, making it the current directory, use the following command:

```
cd dos
```

Taking this example a step further, let's suppose that your DOS directory has a subdirectory named MYFILES. To change to this directory from your root directory, use this command:

```
cd dos\myfiles
```

To move one level up the directory tree to the parent directory, use the cd command with **..** (dot-dot). Building on the previous example, if your C:\DOS\MYFILES directory is current, you can use the following command to move up one level to C:\DOS:

```
cd ..
```

You can move to any directory on a disk, regardless of which directory is current, by following the cd command with a backslash (\), indicating the root directory, and the path of the directory. So, if the current directory is C:\DOS\MYFILES and you want to move to another directory named C:\WP51\DOCS, use:

```
cd \wp51\docs
```

## Viewing the Contents of a Directory

As you've already seen, if you've followed the exercises in Lesson 21, the Dir command generates a list of the files and subdirectories in a directory. To review: To use the command, change to the directory whose contents you want to view (use the cd command discussed in the previous section); then type **dir** and press **Enter**. DOS generates a listing similar to the one that follows:

```
 Volume in drive C is MS-DOS_5
 Volume Serial Number is 1738-8B53
 Directory of C:\

DOS            <DIR>      09-24-91   5:26p
COMMAND  COM       47845 04-09-91   5:00a
AUTOEXEC BAT         297 07-30-92   5:48p
123            <DIR>      09-24-91   6:00p
BIN            <DIR>      09-24-91   6:04p
DBASE          <DIR>      09-24-91   6:04p
OLD_DOS  1     <DIR>      09-24-91   6:11p
WP51           <DIR>      09-24-91   6:31p
HIMEM    SYS       13616 06-07-91   3:10a
MOUSE    SYS       31701 05-01-90   3:00a
CONFIG   SYS         181 07-30-92   1:59p
       11 file(s)      93640 bytes
                    16633856 bytes free
```

This is the same listing that we generated in Lesson 21. But now, its contents will be explaned further. The names of files are listed along with their size in bytes and their date and time of creation. The names of subdirectories are listed as well. To indicate a subdirectory, DOS places <DIR> next to its name.

The Dir command also supports an optional Path parameter that lets you specify a wildcard pattern to describe the list of files and/or directories you want to see. As you may recall, DOS supports two wildcard symbols: ? and *; the ? symbol can represent any single character in a wildcard pattern, and * can represent any group of characters. For example, the wildcard pattern *.* represents any filename with any extension; the wildcard pattern *. represents any filename with no extension. Because very few people add an extension

to a directory name when it is created (see the next section), this wildcard pattern is frequently used to generate a list of subdirectories of the current directory. To do this yourself, type **dir *.** at the DOS prompt and press **Enter**.

***Tip***

Use the Tree command to see a simple diagram of subdirectories beneath a directory. To use this command, make the directory you want to examine current (use the cd command), then type tree and press Enter. DOS will display a listing similar to the following:

```
C:\DOS>tree
Directory PATH listing for Volume MS-DOS_5
Volume Serial Number is 1738-8B53
C:.
|___MYFILES
    |___BUSINESS
    |___PERSONAL
```

## Creating Directories and Subdirectories

If you place all your files in a single directory, it will soon become so crowded that finding your files will become a nightmare. To prevent this, plan a workable structure of parent directories, create them, and then create any needed subdirectories beneath those parent directories. With a logical structure of directories in

place, you shouldn't have any trouble locating important files.

To create a directory at the command line, you use the Mkdir command (md for short). This command creates a subdirectory of the current directory using the name you specify. For example, let's suppose that as part of your directory plan, you've decided to create a directory off your root (\) named MYFILES with two subdirectories called PERSONAL and BUSINESS. To do this, perform the following steps:

1. Type **cd** \ and press **Enter** to make your root directory current.
2. Type **md myfiles** and press **Enter**. This creates the directory named MYFILES.
3. Type **cd myfiles** and press **Enter**. This makes the new MYFILES directory current.
4. Type **md personal** and press **Enter**. This creates a subdirectory of MYFILES named PERSONAL.
5. Type **md business** and press **Enter**. This creates another subdirectory of MYFILES named BUSINESS.

Once these directories and subdirectories have been created, you can begin using them immediately. You can start, for instance, using the DOS Copy command (see the next lesson) to copy your personal files to C:\MYFILES\PERSONAL and your business files to C:\MYFILES\BUSINESS.

## Removing Directories

If you make a mistake while creating a directory, or you decide a directory is no longer needed, you can remove

it. To do this from the command line, you use the Rmdir command (rd for short).

Be aware, however, that the rd command will not let you remove a directory that contains either files or subdirectories. Therefore, make sure a directory is empty before you attempt to remove it. (See the next lesson for details on how to copy those files you want to keep to another directory and how to delete those files you no longer need.) In addition, the rd command will not let you remove a directory that is current, i.e., the directory you're in. In that case, change to the parent directory of the directory you want to remove before removing it.

As an exercise of the above, let's now remove the directory you created in the previous section (MYFILES and its subdirectories PERSONAL and BUSINESS). To do this, perform the following steps:

1. Type **cd \myfiles** and press **Enter** to make MYFILES current.
2. Type **rd personal** and press **Enter**. This removes the subdirectory named PERSONAL.
3. Type **rd business** and press **Enter**. This removes the subdirectory named BUSINESS.
4. Type **cd ..** and press **Enter**. This makes the parent directory of MYFILES—your root directory (\)—current.
5. Type **rd myfiles** and press **Enter**. This removes the directory named MYFILES.

◆ *Lesson* ◆

# 23

# Getting Information about Files

## Using Dir with Files

So far, although we've mentioned using the Dir command to obtain information about files, we've really seen it in action only with directories and subdirectories. This lesson, therefore, explains using Dir specifically with files.

You can use the Dir command to get information about one file or a group of files. You can also use switches to control the organization and structure of any listing generated by Dir.

## Getting Information about a Specific File

To check for the existence of a specific file in a directory, type **dir**, leave a space, and then type the name of a file. To check for the existence of the file README.TXT in your C:\DOS directory, for example, begin by making that directory current. Then type **dir**

**readme.txt** at the DOS prompt and press **Enter**. DOS displays a listing similar to the following:

```
 Volume in drive C is MS-DOS_5
 Volume Serial Number is 1738-8B53
 Directory of C:\DOS

README   TXT      33655 04-09-91   5:00a
       1 file(s)      33655 bytes
                  16609280 bytes free
```

## Getting Information about a Group of Files

You can also use the Dir command to get information about a group of files that have something in common—for example, similar names or matching file extensions. To do this, follow the Dir command with a wildcard pattern that describes the files you want to list. For example, to access a list of the files in your C:\DOS directory that begin with the letter C and have an .EXE extension, type **dir c*.exe** at the DOS prompt and press **Enter**. DOS generates the following listing:

```
 Volume in drive C is MS-DOS_5
 Volume Serial Number is 1738-8B53
 Directory of C:\DOS

CHKDSK   EXE     16200 04-09-91   5:00a
COMP     EXE     14282 04-09-91   5:00a
       2 file(s)      30482 bytes
                  16605184 bytes free
```

## Controlling the Files List

The Dir command also accepts several switches that let you control the structure and scrolling method for the files list generated by the command. One of the most useful is the **/p** switch, which displays one full screen of file information, pauses so you can read the information, and displays the message:

```
Press any key to continue...
```

When you press a key, the next screenful of information is displayed, and so on, until the command runs it course. Here is an example:

```
dir *.txt /p
```

Another useful Dir command switch is **/w**, first introduced in Lesson 21. It generates a multicolumn listing of files. For example, if you typed **dir *.com /w**, the output when your C:\DOS directory is current would be:

```
 Volume in drive C is MS-DOS_5
 Volume Serial Number is 1738-8B53
 Directory of C:\DOS

FORMAT.COM   KEYB.COM     MODE.COM     MIRROR.COM   SYS.COM
UNFORMAT.COM DOSKEY.COM   DOSSHELL.COM EDIT.COM     MSHERC.CM
ASSIGN.COM   DISKCOMP.COM DISKCOPY.COM GRAFTABL.COM GRAPHICSCOM
MORE.COM     LOADFIX.COM  TREE.COM     COMMAND.COM
      19 file(s)     257710 bytes
                   16596992 bytes free
```

You will find the **/w** switch convenient when you want to glance at all the file or directory names at once.

◆ *Lesson* ◆

# 24

# Working with Files from the Command Line

## Frequently Used Commands

When working with files at the command line, you'll frequently use the three commands below:

| | |
|---|---|
| **Copy** | Copies files from one directory to another. |
| **Rename** | Lets you assign a new name to a file. |
| **Del** | Deletes the file(s) you specify. |

We'll show you how to use each of these commands in the sections that follow.

## Copying Files from the Command Line

To Copy files from one directory to another, you use the Copy command. This is undoubtedly one of the most frequently used commands in DOS.

Copy accepts two parameters: Source and Destination. The Source is the path and name of the file(s) you

want to copy; Destination is the path and name to which you want to copy the file(s).

## Copying a Single File

To demonstrate how to copy a single file using the Copy command, we'll copy README.TXT from your C:\DOS directory to the root directory (C:\) directory. It's a simple process: Type **copy c:\dos\readme.txt c:** at the DOS prompt and press Enter. DOS then displays the message: **1 file(s) copied**.

When you use the Copy command to copy a file from one directory to another, DOS overwrites any file in the Destination directory that has the same name as the file you are copying. If you know there is a duplicate file in the Destination directory, you can prevent it from being overwritten by changing the name of the source file as it is being copied. Read the next section for details on how to this.

## Changing the Name of a File While Copying

To change the name of a file while copying it, specify a new filename as part of the Destination parameter for the Copy command. For instance, building on the example in the previous section, let's try copying README.TXT from C:\DOS to the root directory under the name TEXT.TXT. To do this, simply type **copy c:\dos\readme.txt c:\text.txt** at the DOS prompt and press Enter. DOS displays the message **1 file(s) copied**.

**Tip**

If you are copying a file from one directory to another on the same drive, you can leave out the drive designator in the Source and Destination parameters for the Copy command. For example, copy \dos\read-me.txt \text.txt has the same effect as the command shown above.

## Copying a Group of Files

Copying a group of files is similar to using Dir with a group of files in that they too must have something in common, such as matching file extensions. Copying a group of files is accomplished by using a wildcard pattern as the Source argument for the Copy command. Let's try an example by copying the APPNOTES.TXT and README.TXT files in your C:\DOS directory to the root directory (C:\). (Notice that these are the only two files in your C:\DOS directory that have a .TXT extension.) To do this, type **copy c:\dos*.txt c:** at the DOS prompt and press **Enter**. DOS displays the message: **`2 file(s) copied`**.

## Copying Files to a Different Drive

The Copy command also can be used to copy files from one drive to another. For example, if you want to copy the APPNOTES.TXT and README.TXT files in your C:\DOS directory to a floppy disk in drive A, insert a formatted floppy disk in drive A and then type **copy c:\dos*.txt a:** at the DOS prompt. Press **Enter** and DOS will copy the files to drive A and display the message: **`2 file(s) copied`**.

## Renaming Files

To change the name of a file *without* copying it, you use the Rename command (Ren for short). To use this command type **ren** followed by a space, the path and name of the file whose name you want to change, another space, and the new name for the file. As an exercise, let's try changing the name of the README.TXT file in your C:\DOS directory to DOS5-READ.ME. To do this, type **ren c:\dos\readme.txt dos5read.me** at the DOS prompt and press **Enter**. To change the name back, type the following command line at the prompt: **ren c:\dos\dos5read.me readme.txt**.

The Ren command has a built-in safeguard; it won't let you overwrite a file in the current directory with the same name. So, if the new name you chose for a file matches that of an existing file in the same directory, DOS displays the message: **Duplicate filename or file not found**.

Like Copy, Ren can be used in conjunction with the DOS wildcard symbols to rename groups of files. In this way, then, a series of text files in the current directory that all have .PRN extensions can be renamed as a group to have .TXT extensions. To make this change, type **ren *.prn *.txt** at the DOS prompt.

## Deleting Files

Deleting files from the DOS command line is accomplished with the Del command. Simply type **del** followed by a space, and the path and name of the file you want to delete. For example, if you have been following along with the examples in this section, you

now have three files (README.TXT, TEXT.TXT, and APPNOTES.TXT) in your root directory that don't belong there. To delete these files, make your root directory current (type **cd** \ and press **Enter**), then type the following three commands at the DOS prompt, pressing **Enter** after each one:

```
del readme.txt
del text.txt
del appnotes.txt
```

This cleans up the .TXT files in your root directory.

The Del command, too, can be used with a wildcard pattern to delete groups of files. Let's say hypothetically that the README.TXT, TEXT.TXT, and APPNOTES.TXT files are the only .TXT files in your root directory and you want to delete them. (Check to make absolutely sure these are the only .TXT files in that directory before trying this next example.) To delete these files, type **del *.txt** at the DOS prompt and press **Enter**.

## *Undeleting Files*

If you accidentally delete the wrong file(s) with the Del command, often you can recover them with the Undelete utility. Refer back to Lesson 16 for details on how to use this important utility.

◆ *Lesson* ◆

25

# Changing File Attributes

## The Attrib Command

Lesson 12 explained how to change the attributes of a file from within the DOS Shell. This lesson demonstrates how to change the attributes for one or more files from the DOS command line with the Attrib command. Attrib supports the following parameters that determine the attributes of a file:

**a (Archive)** The archive attribute is used by the Backup and Xcopy commands to determine which files are affected by these commands. All files are assigned the archive attribute when first created.

**r (Read only)** The file appears in directory listings but cannot be changed or deleted.

**h (Hidden)** The file continues to exist but does not appear in directory listings.

**s (System)** The file is a DOS system file. Like files with the Hidden attribute, files with the System attribute do not appear in directory listings.

To find out which attributes are presently assigned to the files in the current directory, type **attrib** at the DOS prompt and press **Enter**. DOS will display a listing similar to the following:

```
     SHR      C:\IO.SYS
     SHR      C:\MSDOS.SYS
A             C:\COMMAND.COM
A             C:\AUTOEXEC.BAT
A    SHR      C:\MIRORSAV.FIL
A             C:\HIMEM.SYS
A             C:\MOUSE.SYS
A             C:\CONFIG.SYS
      S       C:\PCTRACKR.DEL
```

Notice that some files have more that one attribute assigned to them; for example, the first file in the list is a system, hidden, read-only file.

To change an attribute for a file, type **attrib** followed by a space, a plus or minus sign (**+** or **–**), an attribute symbol (**a**, **r**, **h**, or **s**), another space, and the name of the file. A plus before the attribute symbol turns the attribute on, and a minus turns it off.

As an exercise, let's say that to protect the README.TXT file in your C:\DOS directory from being deleted or changed, you've decided to assign it a read-only attribute. To do this, first make your C:\DOS directory current (type **cd \dos** and press **Enter**) and then type **attrib +r readme.txt**. Press Enter to execute this command and DOS immediately turns on the

read-only attribute for the file. To turn off the read-only attribute for this file, type **attrib -r myfile.txt**, and press **Enter**.

Attrib, too, can be used on a group of files by using a wildcard pattern in place of a filename. So, to assign the read-only attribute to all the .TXT files in your C:\DOS directory you would type **attrib +r *.txt**.

◆ *Lesson* ◆

# 26

# Running Programs from the Command Line

## About Executable Files

Perhaps the easiest and most frequent operation you'll perform at the DOS command line is running programs. To run a program, you must know the name of its executable file. As you may recall, executable files have an .EXE or .COM. extension. Once you know the filename, simply type its path and name at the command line (you don't need to include the extension) and press **Enter**. DOS will load the program and begin executing the instructions it contains.

For example, the executable file name for the MS-DOS Editor is EDIT.COM, and it is located in your C:\DOS directory. To run this program, type **c:\dos\edit** at the DOS prompt and press **Enter**. A short time later the MS-DOS Editor's opening screen will appear.

If the program you want to run resides in the current directory, you don't need to specify its path when running the program. For example, if your C:\DOS

directory is current, you can run the MS-DOS Editor simply by typing **edit** and then pressing **Enter**.

***Tip***

You can include a Path statement in your AUTOEXEC.BAT file that gives DOS a list of directories to search when attempting to run a program. If the path to a directory containing the program you want to run is included in the Path statement in AUTOEXEC.BAT, you can run the program from any directory simply by typing its executable filename and pressing Enter. And because the DOS directory is almost always included in the Path statement, you can execute DOS commands from any directory without specifying a path. (See Lesson 32 for more details on this topic.)

## Passing Parameters to Programs

You also can pass a parameter to a program when you run it, thus controlling its operation. To do this, type the path and name of the program's executable file followed by a space and the appropriate parameter. For example, to run the MS-DOS Editor and have the README.TXT file in your C:\DOS directory loaded automatically, make your C:\DOS directory current (type **cd \dos** and press **Enter**), then enter **edit readme.txt**. When you press Enter, the MS-DOS Editor starts and the README.TXT file is loaded.

# Making the Leap from the DOS Shell

In various lessons throughout this book, we've shown you how to run different DOS utilities using the program items in the program-list area of the DOS Shell. In truth, however, each of these program items runs an executable program file located in your C:\DOS directory. As you might have guessed, you can run these same programs directly from the DOS command line. Table 26.1 shows the relationship between each program item in the DOS Shell, the program it runs, the DOS command required to run the same program from the command line, and the lesson in which the program is discussed.

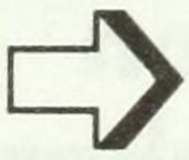
When you select any one of the program items listed in Table 26.1 from the DOS Shell, you are presented

## Table 26.1 Shell Program Group Options and Corresponding Commands

| Program Item | DOS Program | DOS Command | Lesson Number |
|---|---|---|---|
| Editor | EDIT.COM | Edit | 13 |
| Disk Copy | DISKCOPY.COM | Diskcopy | 15 |
| Backup Fixed Disk | BACKUP.EXE | Backup | 18 |
| Restore Fixed Disk | RESTORE.EXE | Restore | 19 |
| Quick Format | FORMAT.COM | Format (/q) | 14 |
| Format | FORMAT.COM | Format | 14 |
| Undelete | UNDELETE.EXE | Undelete | 16 |

with a dialog box requesting parameters for that program item. After you supply the appropriate parameters and confirm the dialog box, the DOS Shell runs the associated executable program file in your C:\DOS directory and passes the parameters you've supplied to that program. To accomplish the same thing from the DOS command line, type the appropriate command at the DOS prompt, followed by the same parameters you would supply in the DOS Shell dialog box. Once the program starts running, follow the procedures outlined in the appropriate lesson.

For example, if you select the Format program item from the DOS Shell Disk Utilities group, you are presented with a dialog box requesting a drive to format; Drive A (a:) is the suggested parameter. You can achieve the same result simply by typing **format a:** at the DOS command line and pressing **Enter**. DOS will run the FORMAT.COM utility in your C:\DOS directory and format the floppy disk in Drive A. Once the program starts running, follow the same procedures outlined in Lesson 14, "Formatting Floppy Disks."



***Note:*** There is one exception to the foregoing involving the DOS Shell Quick Format program item. When you select this program item, a dialog box is presented requesting a drive to quick format. The suggested parameter is a: (meaning drive A). When you confirm this dialog box, however, the shell actually runs the FORMAT.COM program and passes the a: parameter followed by the /s switch. To do the same thing at the DOS command line, you need to use the following command: format a: /s.

◆ *Lesson* ◆

# 27

# Command Line Shortcuts

## Cutting Down on Typing

Some people object to working from the DOS command line because of the amount of typing involved. Admittedly, typing the path to a directory that is three of four levels deep can be a time-consuming process. There are, however, a few shortcuts that can help to reduce the amount of time you spend typing at the DOS command line.

## Current Shortcuts

### *Taking Advantage of the Current Drive*

As you know by now, the current drive is the disk drive in which you are presently working. DOS always knows which drive that is and it assumes your commands apply to the directories and files on that drive. Therefore, provided the effect of a command is confined to the current drive, you don't need to include

the current drive letter in the command. For example, the command **copy c:\myfiles\notes.txt c:\work** can become **copy \myfiles\notes.txt \work**.

## Taking Advantage of the Current Directory

The current directory, like the current drive, is the one in which you happen to be working, and DOS will always look in this directory automatically when searching for files. Therefore, if the file you are referring to is located in the current directory, you don't have to include the path to the directory in front of the filename. This means that if the current directory is C:\MYFILES\PERSONAL and you want to copy its .TXT files to a disk in drive A, instead of using the command **copy \myfiles\personal*.txt a:**, you can use **copy *.txt a:**.

### Tip

DOS substitutes the current directory as the Destination argument for the Copy command, so if you want to copy all the .TXT files in the root directory on drive A to the current directory on drive C, use the command copy a:*.txt.

In addition, if the command you are using affects only files in the current directory, you can leave out the path entirely. For example, let's suppose the current directory is C:\MYFILES\PERSONAL and you want to copy a file named NOTES.TXT to a new file named JULY.TXT in that same directory. Rather than using the command **copy \myfiles\personal**

**notes.txt \myfiles\personal\july.txt**, you can simply substitute **copy notes.txt july.txt**.

## Subdirectory Shortcut

When referencing a subdirectory of the current directory, it's not necessary to include the full path to the subdirectory in a command. If, for example, the current directory is C:\MYFILES\PERSONAL, and you want to copy all the .TXT files in this directory to a subdirectory named C:\MYFILES\PERSONAL\JULY, use the simple command **copy *.txt july** instead of **copy *.txt \myfiles\personal\july**.

## Dot and Dot Dot

Every DOS directory contains a directory called . (dot). This is a special directory that stands for "the current directory." Therefore, when you want to refer to the current directory, you can use a . to do so. Similarly, another directory called .. (dot dot) stands for the parent directory of the current directory. When you refer to this directory, you can use .. for its name. For example, let's suppose the current directory is C:\MYFILES\PERSONAL and you want to copy a file named JULY.TXT in that directory to C:\MYFILES\WORK. You can use the command **copy july.txt ..\work**.

# Previous Location Shortcuts

DOS also remembers the last directory that was used on each of your drives. As you know, when you boot your computer, the current directory is the root direc-

tory of each of your disk drives. If you use the Chdir (cd) command to make another directory current, and then switch to another drive, DOS remembers the last directory you were in when you left the old drive. And you can refer to this directory simply by using the drive letter. For example, let's say that the current directory on drive C is C:\MYFILES\PERSONAL. Then you switch to drive A by typing **a:** and pressing **Enter**. Next suppose you want to copy all the .TXT files in the current directory on drive A to your C:\MYFILES\PERSONAL directory. You can do this with the following command: **copy *.txt c:**.

Notice that you don't have to spell out the \MYFILES\PERSONAL directory name on drive C. DOS remembers the last directory you were in on drive C and responds accordingly.

◆ *Lesson* ◆

28

# Working with ASCII Files

## Printing ASCII Files

To print an ASCII file on your printer while you are using other DOS commands, type **print** followed by a space and the path and name of the file you want to print. When you press Enter to execute the command, DOS will prompt you to specify the port (device) to which your printer is connected. Once you provide this information, DOS prints the file.

The AUTOEXEC.BAT file in your root directory is an ASCII file, so we'll use it as our sample file. (This file is discussed further in Lesson 32.) To print this file, perform the following steps:

1. Type **cd** \ and press **Enter** to make your root directory current.
2. Type **print autoexec.bat**. DOS displays the following message prompting you for the port to which your printer is connected:

```
Name of list device [PRN]:
```

3. Type the **name of the port** to which your printer is connected and press **Enter**. You can use any of the following names. lpt1, lpt2, lpt3, com1, com2, com3, or com4. If you don't specify a port name, DOS will print to PRN (the same as lpt1); in this case, just press **Enter**. DOS displays the following messages:

```
Resident part of Print loaded

C:\AUTOEXEC.BAT is currently being printed
```

Once you specify a port setting for the print command, DOS will begin printing the file. At this point, you can begin using other DOS commands without having to wait for the Print Command to complete its task.

## Listing ASCII Files

When you want to see the contents of an ASCII file displayed on your screen, use the Type command. Enter **type** followed by a space and the path and name of the ASCII file whose contents you want to view. Suppose, for example, that you want to look at the contents of the ASCII file AUTOEXEC.BAT stored on the root directory of your computer. To do this:

1. Type **cd** \ and press **Enter** to make your root directory current.
2. Enter **type autoexec.bat** and press **Enter**. DOS lists the contents of AUTOEXEC.BAT on your screen.

If you are using the Type command to display a particularly large ASCII file, DOS will scroll the listing until the end of the file is reached. Often this scrolling occurs so quickly that you can't read the contents of the file. To stop this scrolling at any time, press Ctrl+s or the Pause key. DOS will pause the scrolling until you press any key (except Pause).

◆ *Lesson* ◆

29

# Copying Entire Directories

## The Xcopy Command

You can copy an entire directory and its subdirectories by using the Xcopy command. This command automatically copies all the files in a directory and (optionally) its subdirectories to the target location you specify. Thus, with a single command, you can create a mirror image of a directory and its subdirectories at another location.

## Copying a Single Directory

Let's again call on the directory MYFILES for this example. We'll copy all its files on drive C to a floppy disk in drive A. When the copy is complete, a new directory named MYFILES will exist on the floppy disk in drive A and contain all the files from the MYFILES directory on drive C. To do this, make sure there is a formatted

floppy disk in drive A and then perform the following steps:

1. Type **xcopy c:\myfiles a:\myfiles** and press **Enter**. (The use of drive letters—c: and a:—in the source and target parameters for the Xcopy command are required here.) DOS displays the following message:

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

2. Type **d** and press **Enter**. DOS creates the MYFILES directory on the floppy disk in drive A and then copies the files from C:\MYFILES to that directory.

***Tip***

You can avoid the message displayed in step 1 of the preceding example by adding a backslash (\) to the end of the target directory name.

It is not necessary to use matching directory names when copying directories from one drive to another. In the preceding example, for instance, you could use a command like:

```
xcopy c:\myfiles a:\newfiles
```

and DOS would create the directory NEWFILES on drive A, and copy the contents of C:\MYFILES to it.

## Copying to an Existing Directory

The Xcopy command also can be used simply to copy all the files in one directory to another *existing* directory. This allows you, for example, to copy all the files in a directory named MYFILES on drive C to an existing directory named OLDFILES on drive A. To do this, use the following command:

```
xcopy c:\myfiles a:\oldfiles
```

## Copying Subdirectories

Using Xcopy to copy subdirectories to a new location requires only that you add the **/s** switch to the Xcopy command. To demonstrate, let's return to our directory named MYFILES with its two subdirectories, PERSONAL and BUSINESS. Now let's say you want to recreate this entire directory structure on a floppy disk in drive A. To do this, make sure you have a formatted floppy disk in drive A and then perform the following steps:

1. Type **xcopy c:\myfiles a:\myfiles /s** and press **Enter**. DOS displays this message:

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

2. Type **d** and press **Enter**.

DOS creates the MYFILES directory on the floppy disk in drive A and then copies the files from C:\MYFILES

to it. Beneath this directory, DOS creates two subdirectories of this directory named PERSONAL and BUSINESS. It then copies the contents C:\MYFILES\PERSONAL to A:\MYFILES\PERSONAL and the contents of C:\MYFILES\BUSINESS to A:\MYFILES\BUSINESS.

By default, the Xcopy command will not copy an empty subdirectory unless you use the /e switch along with the /s switch, as in /s /e.

◆ *Lesson* ◆

# 30

# Creating Batch Files

## Understanding Batch Files

Batch files let you automate operations at the DOS command line. These files are simply ASCII files with a .BAT extension that contain DOS commands. You can run a batch file simply by typing its name (with or without the .BAT extension) at the DOS prompt. DOS will read the commands in the file and execute them as quickly as your PC will allow. You can create a batch file using any text editor capable of opening and saving ASCII files—for example, the MS-DOS Editor.

## Creating a Simple Batch File

In this section, we'll create a simple batch file that makes your C:\DOS directory current and then copies all the .COM files in that directory to drive A. Although this batch file won't be terribly useful, you'll be able to apply the principles you learn when creating your own batch files.

In preparation for this example, insert a formatted floppy disk in drive A, then do the following:

1. Working from the DOS prompt, type **edit** and press **Enter**. The MS-DOS Editor opening screen appears.
2. Press **Esc** to clear the opening prompt.
3. Type **c:** and press **Enter**. (This line will make your C drive current no matter which drive you happen to be using when you run the batch file.)
3. Type **cd \dos** and press **Enter**. (This line will make your DOS directory current.
4. Type **copy *.com a:** and press **Enter**. Your batch file is finished. Your screen should now look like Figure 30.1.
5. Type **Alt+F**. This opens the MS-DOS Editor File menu.
6. Type **s**. This selects the Save option from the File menu and displays the Save dialog box.

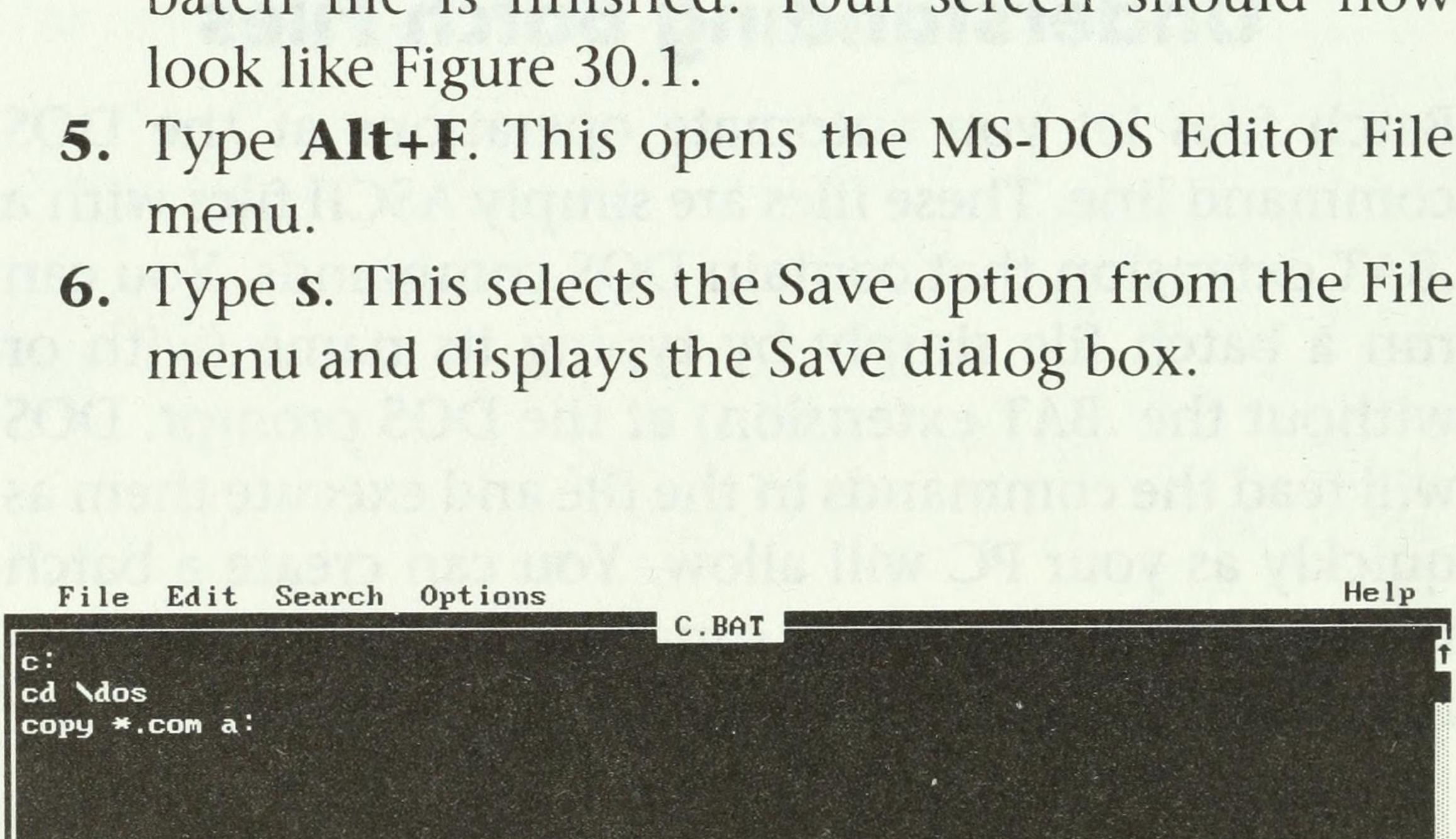



**Figure 30.1** A sample batch file

7. Type **c:\dos\c.bat** and press **Enter**. This saves the file under the name C.BAT in your C:\DOS directory.
8. Type **Alt+F** again to open the File menu, and type **x**. This selects the Exit option from the file menu. The MS-DOS Editor is removed from memory and you are returned to the DOS command line. Your new batch file is now ready.

## Running the Batch File

To run your new batch file, type **c** at the DOS prompt and press **Enter**. DOS makes your C drive current, changes to your C:\DOS directory, and begins copying all the .COM files in that directory to the disk in drive A. When the batch file is finished running, you are returned to the DOS prompt.

If you don't like what a batch file is doing, you can stop it at any time by typing **Ctrl+c** or **Ctrl+Break**. DOS will display the message:

```
Terminate batch job (Y/N)?
```

Type **Y** to stop the batch file and return to the DOS prompt.

◆ *Lesson* ◆

# 31

# Rebooting Your PC

## When Your Computer Hangs

Although it is rare, you may come across a program that "hangs up" your computer. You can tell this has happened when your PC stops responding to the keyboard and mouse input. If you continue pressing keys, your PC will eventually start to beep. At this point, your only option is to reboot (restart) your PC. Do this by pressing **Ctrl+Alt+Del**. (This is sometimes referred to as a *warm boot*.)

When you restart your PC with Ctrl+Alt+Del, your screen initially will go blank and then your PC will attempt to reboot itself. During this process, it checks its hardware and internal configuration, as though you had flipped its power switch off and then on again.

### Tip

Sometimes when a program goes haywire, it blows away the part of its memory that interprets the Ctrl+Alt+Del sequence. When this happens, pressing Ctrl+Alt+Del will have no effect; your only option at this point is to turn your PC's power switch off and on, or use the Reset switch if your PC has one.

◆ *Lesson* ◆

# 32

# Modifying AUTOEXEC.BAT

## The PC's Boot Sequence

When you start your PC, it performs a series of internal hardware and system checks. Next it checks drive A for a bootable disk—a disk with COMMAND.COM and two other hidden DOS system files, IO.SYS and MSDOS.SYS. If a bootable disk is there, your computer boots from that disk (loads the DOS system files into memory from that disk); otherwise, your computer looks to the root directory of drive C for the DOS system files. If these files are found on the root directory of drive C, your PC will boot from that drive instead.

During the boot procedure, DOS loads the file COMMAND.COM, which gives your PC the ability to respond to the routine DOS commands required to manage directories and files. It also automatically looks for two other files—CONFIG.SYS and AUTOEXEC.BAT. If these files are present, DOS loads them and runs the instructions they contain.

Both CONFIG.SYS and AUTOEXEC.BAT are ASCII text files that can be viewed and edited using the MS-DOS Editor. They also contain command-line instructions, similar to the ones you might enter at the DOS prompt. The purpose of these command lines is to run different programs that customize the internal configuration and operation of your PC.

The CONFIG.SYS file is always run first. In general, this file contains instructions for loading device drivers (special programs that tell DOS how to interact with your specific equipment). A version of CONFIG.SYS is created automatically by DOS 5.0 when you install the software on your PC.

The AUTOEXEC.BAT file is run next. While this file can contain a variety of instructions, its primary function is to run the programs and DOS commands you invoke each time you start your computer. Therefore, this is the configuration file most frequently edited by DOS users. You can use AUTOEXEC.BAT do such things as change the appearance of the DOS prompt, or define a sequence of directories in which DOS will automatically search for files. Like CONFIG.SYS, DOS 5.0 creates a version of AUTOEXEC.BAT when you install the software.

## Before You Proceed

In the sections that follow, we'll ask you to load AUTOEXEC.BAT into the MS-DOS Editor to have a look at the file contents. We'll also tell you about different entries you can make in the file to customize the operation of your PC. Before you proceed, however, we

*strongly* recommend that you back up AUTOEXEC.BAT and create a system disk (a bootable floppy disk). (Both procedures are defined below.) That way, if anything goes wrong, you'll be able to reboot your PC and start fresh.

## Backing Up AUTOEXEC.BAT

To back up AUTOEXEC.BAT, follow these steps:

1. Working from the DOS command line, type **c:** and press **Enter** to make drive C current.
2. Type **cd ** and press **Enter** to make the root directory of your PC current.
3. Type **copy autoexec.bat autoexec.bak** and press **Enter**. This creates a second copy of AUTOEXEC.BAT under the name AUTOEXEC.BAK.

## Creating a System Disk

To create a system disk, you need a blank, formatted floppy disk. When you're ready, follow these steps:

1. Insert the formatted floppy disk in drive A.
2. Working from the DOS command line, type **cd \dos** and press **Enter** to make the C:\DOS directory current.
3. Type **sys a:** and press **Enter**. A short time later, DOS displays the message:

```
System Transferred
```

## IF SOMETHING GOES WRONG

If something goes wrong and your PC doesn't work the way it used to, or you cannot reboot your PC, perform the following steps:

1. Insert the system disk you created in the previous section into drive A.
2. Press **Ctrl+Alt+Del**. Your computer will restart and DOS will prompt you for the date and time. Press **Enter** in response to both of these prompts. The DOS prompt **A:>** will then be displayed.
3. Type **c:** and press **Enter** to make drive C current.
4. Type **copy autoexec.bak autoexec.bat** and press **Enter**.
5. Remove the system disk from drive A and press **Ctrl+Alt+Del** to reboot your computer.

## Loading AUTOEXEC.BAT into MS-DOS Editor

To take a look at AUTOEXEC.BAT, you'll need to load the file into the MS-DOS Editor. To do this, perform the following steps:

1. Working from the DOS command line, type **edit** and press **Enter**. A short time later the opening screen of the MS-DOS Editor will appear.
2. Press **Esc** to clear the opening prompt.
3. Type **Alt+F** to open the File menu.
4. Type **o**. This selects the Open command from the File menu and displays the Open dialog box.

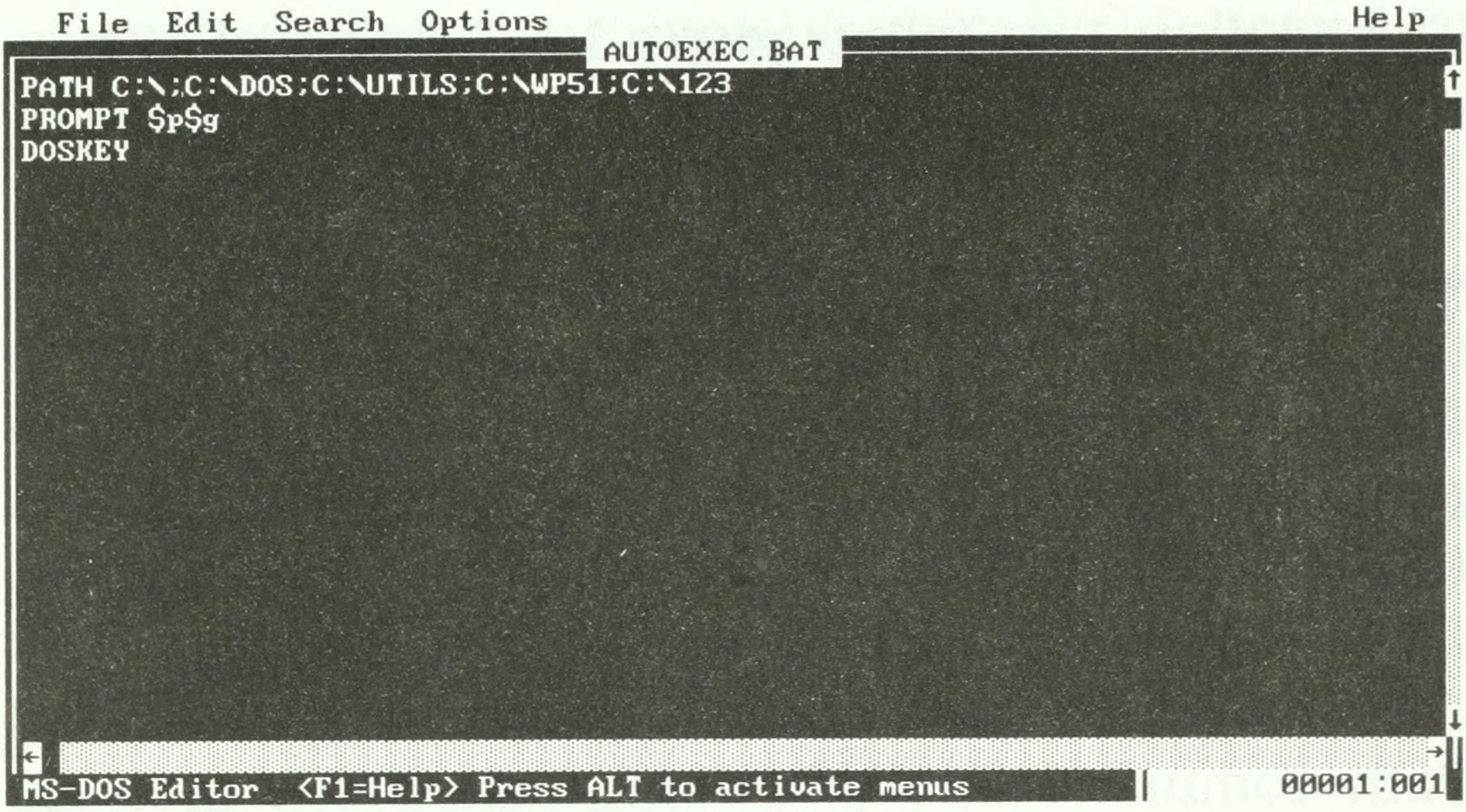



**Figure 32.1** A typical AUTOEXEC.BAT file

**5.** Type **c:\autoexec.bat** and press **Enter**. The AUTOEXEC.BAT file is loaded and now appears on your screen, similar to the one shown in Figure 32.1, although your particular version of AUTOEXEC.BAT may appear different.

## AUTOEXEC.BAT Commands

There are three commands in AUTOEXEC.BAT that any serious DOS user should know about. They are:

**Path** Defines a series of directories in which DOS will search for program files.

**Prompt** Configures the appearance of the DOS prompt.

**Doskey** Loads the Doskey program that allows you to easily repeat commands recently entered at the DOS command line.

The sections that follow give a brief overview of how these commands are used.

## The Path Command

The Path command is used to define a series of directories through which DOS will search for program files (.EXE, .COM, or .BAT files). When you run a program, DOS will always look in the current directory first. After that, DOS looks in the directories defined by the Path command in AUTOEXEC.BAT. If the program you want to run is included in one of these directories, DOS will run it, regardless of which directory happens to be current. This means you can run a program from any directory just by typing its file name at the command line and pressing Enter; you don't have to include the full path to the program file.

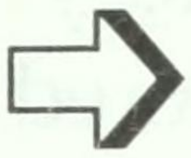
To create a Path statement, type **PATH** followed by a space and the drive and directory path of each of the directories that you want DOS to search. Separate your entries with semicolons(;). DOS will search the directories in the order in which they are listed. For example, the AUTOEXEC.BAT file in Figure 32.1 includes the following Path command statement:

```
PATH C:\;C:\DOS;C:\UTILS;C:\WP51;C:\123
```

With this Path statement, DOS will first search the root directory (\) on drive C, followed by C:\DOS, C:\UTILS, C:\WP51, and C:\123, in that order. If it can't find the program you are trying to run in any of these directories, DOS displays an error message.

If a directory containing a program that you run frequently is not included in your Path statement, you can easily add it. To do this, move the cursor to the end

of the Path statement, type **;** (semicolon) followed by the drive and path to the directory.



**Note:** When you install a new program, its Installation program will, in many cases, attempt to update the Path statement in AUTOEXEC.BAT to include the directory in which the program has been installed.

## The Prompt Command

Unless you specify otherwise, the DOS prompt consists of the current drive letter followed by a greater than (>) symbol, for example, **C>**. The Prompt command, however, lets you control the appearance of the DOS prompt.

Probably you already have a Prompt statement in your AUTOEXEC.BAT file, but in case you don't, we recommend you include the following statement on a separate line:

```
prompt $p$g
```

This particular version of the Prompt command has two parameters: **$p** and **$g**; **$p** causes the path to the current directory to be displayed as part of the DOS prompt. Without this, it's very difficult to tell which directory you're in when working at the DOS command line. The **$g** parameter causes a greater than (>) symbol to appear at the end of the DOS prompt. With this in mind, if C:\MYFILES is the current directory, your DOS prompt would look like this:

```
C:\MYFILES>
```

## The Doskey Command

The Doskey command runs the Doskey utility (DOSKEY.COM) located in your C:\DOS directory. This program records the commands you've entered recently and allows you to quickly recall and rerun them. (Refer to Lesson 21 for details on how to work with this utility.) If the Doskey command is not included in your AUTOEXEC.BAT file, you can easily include it by typing **doskey** on a separate line.

## Saving AUTOEXEC.BAT in the MS-DOS Editor

If you made any changes to AUTOEXEC.BAT in the MS-DOS Editor, you must save them to disk. To do this, perform the following steps:

1. Type **Alt+F** to open the File menu.
2. Type **s**. This selects the Save command from the File menu and saves the file to disk.
3. Type **Alt+F** to open the File menu again.
4. Type **x**. This selects the Exit command from the File menu and shuts down the MS-DOS Editor.

Finally, in order for any changes you've made to AUTOEXEC.BAT to take effect, you must reboot your PC by pressing **Ctrl+Alt+Del**.

# Epilogue

# Where To Go from Here

Obviously, this book doesn't tell you everything there is to know about DOS. In fact, it only scratches the surface. The following are suggested areas in which you may want to do additional research.

**Doskey** — Editing command lines and creating macros.

**Commands** — Important DOS commands that have not been discussed in this book: Append, Chkdsk, Debug, Fastopen, Mem, Recover, and Unformat.

**Batch files** — Sophisticated batch file techniques, including branching, looping, pausing, and having one batch file call another.

**Optimizing** — Optimizing memory, setting up a disk cache, or setting up a RAM disk.

| | |
|---|---|
| **Configuration** | Customizing AUTOEXEC.BAT and CONFIG.SYS, configuring your ports, and modifying your screen and keyboard. |
| **Disk Management** | Recovering disk space with Chkdsk /f, unformatting, and partitioning. |
| **Command Techniques** | Redirection, piping, and filters. |
| **Variables** | Environment variables—for example, Path, Comspec, Shell, and Temp. |
| **QBasic** | Writing your own custom BASIC programs. |

# Index


ASCII files
  editing, 93
  listing, 182–183
  printing, 181–182
  working with, 181–183
AUTOEXEC.BAT file, 181, 182
  backing up, 197
  including a Path statement in, 174
  loading into MS-DOS Editor, 198–199
  modifying, 195–202
  running Mirror in, 120
  saving in the MS-DOS Editor, 202

Backup utility, 89, 123–129
  operation of, 127
  parameters for, 127
  switches used for, 128
Batch files, 36
  creating, 189–191
  running, 191
  understanding, 189

Chdir (cd) command, 151, 152, 180
Chkdsk utility, 79
COMMAND.COM file, 105, 195
  understanding, 144
Command line
  getting help at, 150
  running programs from, 173–176
  working with files from, 163–167
Commands
  cancelling, 147–148
  entering, 144–147

for managing
directories, 151
pausing, 147
repeating last, 148–149
selecting multiple files
for, 45–49
using, 144–149
CONFIG.SYS file, 195–196
Copying
changing filenames
while, 164–165
of entire directories,
185–188
of file groups, 165
of files, 22–27
of files between drives,
27
of files to different
drives, 165
of files to other
directories, 24–27
of floppy disks, 107–110
of single files, 164
of subdirectories,
187–188

Del command, 34, 144,
163–167
Del key, 9, 19, 20, 34, 78
Destination disk, 107–108,
109, 110
Directories
backing up, 124–126,
129
changing, 13–16,
152–153
commands for
managing, 151
copying, 185–188
copying files to, 24–27
creating, 18–19, 155–156
defined, 2
removing, 19–20,
156–157
rereading all, 65
restoring files in,
132–136
selecting all files in, 49
viewing the contents of,
153–155
ways of viewing, 51–57
working with, 11–20,
151–157
Disk Copy utility, 107–110
Disk information settings,
54–57
Disks
bad, 102–103
copying, 107–110
formatting, 99–106
Disk Utilities group, 42,
71, 113
program items in, 73–74
programs in, 43
Display mode, 81–84
DOS command line
running programs from,
175–176
working at, 143–150
DOS hierarchical structure,
11–12
DOS prompt, 2, 3, 144, 145

DOS Shell
changing display colors, 81–82
changing screen arrangement, 51–57
customizing, 81–88
exiting, 10
moving between areas of, 4–5
purpose of, 143
working with, 3–10
Drives
changing, 12–13, 151–152
copying files between, 27
copying files to different, 165
working with, 151–157

Editing
of a memo, 97–98
of ASCII files, 93
keys for, 9
Editor, 73
creating a memo using, 94–98
exiting, 98
loading a text file into, 95
using, 93–98
End key, 9, 15
Executable files (.EXE), 35, 39, 146, 173–174
displaying, 59–61
Exit command, 41, 202

File Allocation Table (FAT), 118
File groups
copying, 165
getting information about, 160
File list
controlling, 59–65
running a program from, 35–37
selecting a file in, 24
Filenames
changing while copying, 164–165
duplicate, 31–32
rules for, 21
Files. *See also* Executable files
backing up, 123–129
changing the sort order of, 63–65
copying, 22–27, 163–165
deleting, 34, 166–167
getting information about, 159–161
moving, 29–32
naming, 21–22
overwriting, 31–32, 34
printing, 96–97
renaming, 32–34, 166
rereading all, 65
restoring, 131–136
searching for, 67–70
selecting all, 49
selecting for a command, 45–49

undeleting, 111–116
ways of viewing, 51–57
working with, 21–27
working with from the command line, 163–167
Floppy disks
copying, 107–110
formatting, 99–106
Format, 43, 74, 100
Formatting a disk, 99–106

General Failure message, 99, 100
Graphics mode, 83
Groups of files, searching for, 68–70

Hard disk
Backup utility for backing up, 129
failure of, 105
Help, 37, 76, 150
Hidden files, displaying, 62–63
Home key, 9, 15

Main program group, 41–43, 71
items in, 72–73
programs in, 42
Mkdir (md) command, 151, 156
Mouse
moving a file using, 30–31
moving around the DOS Shell using, 5
running programs using, 41
selecting files using, 47, 49
use in file copying, 25–27
using to view directories, 16
MS-DOS Editor, 40–41. *See also* Editor
loading AUTOEXEC.BAT into, 198–199
saving AUTOEXEC.BAT in, 202
using, 93–98
Multiple programs, 137, 141–142

Path command, 199, 200–201
Path parameter, 154
Paths
defined, 18
File Run command and, 40
Path statement, 174
creating, 200
Pause key, 183
Printing
of a memo, 96
of ASCII files, 181–182
Print option, 96
Program/File Lists view, 55

Program groups
adding program items to, 76–78
building, 71–80
creating new, 74–75
rearranging, 78–79
reviewing contents of existing, 72–74
understanding, 71–74
Program items
changing the properties of, 79–80
deleting, 78
in the Disk Utilities group, 73–74
in the Main program group, 72–73
Programs. *See also* Multiple programs
passing parameters to, 174
running, 35–43
running from the command line, 173–176
simultaneous running of, 138–139
switching between, 140–141
Prompt command, 199, 201

QBasic program files, 36
Quick Format feature, 103–105
Quick Format program, 43, 73

RAM (Random Access Memory), 2
Read only attribute, 89, 90
Rebooting the PC, 193–194
Rename (Ren) command, 33–34, 144, 163, 166
Renaming files, 32–34, 166
Reorder command, 78
Repeating last commands, 148–149
Restore utility, 123, 124, 131–136
parameters for, 133–134
switch parameters for, 134, 135
Restoring files, 131–136
Rmdir (rd) command, 151, 157
Root directory, 11

Searching
for a group of files, 68–70
for a single file, 67–68
Search path, 40
Shell program group options, 175
Single files
copying, 164
searching for, 67–68
Source disk, 107, 109
Source parameter, 127, 128, 129

for the Copy command, 163–164
in the Restore utility, 134
Subdirectories, 11
backing up, 129
copying, 187–188
creating, 155–156
revealing and hiding, 16–17
Switch parameters, 127–128, 134, 135
System disk, creating, 105–106, 197–198
System files, displaying, 62–63
System Transferred message, 106

Tab key, 10
Task switching, 137–142
Text box, 8
Text files, loading into the Editor, 95
Text mode, 82
Time, of file creation, 53
Type command, 182, 183
Typing shortcuts, 177–179

Undelete utility, 34, 111–116, 167
Mirror utility and, 117–122
selective use of, 116
setting parameters for, 121
using in the current directory, 112–115
using outside the current directory, 116
Undeleting files, 111–116

Viewing
of directory contents, 153–155
of file and disk information, 54–55
of file contents, 91–92
of files and directories, 51–57
of hidden and system files, 62–63
Volume label, prompt for, 101

Warm boot, 193
Wildcard characters, 18, 22, 61–62
for copying file groups, 165
Undelete utility and, 116
use in searching for file groups, 69–70
for use with the Del command, 167
Wildcard patterns, specifying, 61–62, 154–155

Xcopy command, 89, 124, 169, 185, 186, 187

BANTAM COMPUTER BOOKS

# FRIENDLY COMPUTER BOOKS

## A Quick, No-Nonsense Approach to Learning the Basics of DOS

As a first-time user, you need to learn to use DOS to do your work more productively. You're not interested in lengthy explanations. Instead, you want an introduction that teaches you the basic features of the software, gets you up and running quickly, and lets you do your work.

That's why we've created Friendly Computer Books. The topic-oriented organization, spacious design, step-by-step approach, and liberal use of illustrations ensure that the essential details needed to use **DOS** literally jump out at you. The special lay-flat binding guarantees that you won't have to fight with your book as you read it. And the very low price means that learning DOS does not require a major investment.

**Since the publication of Bantam's *Windows 3 Power Tools*, The LeBlond Group has developed a reputation as the premier authors of books on operating systems.**

NOVICE

PC

OPERATING SYSTEMS

ISBN 0-553-56214-2

0 76783 00599 0 56214