

TERMS AND CONDITIONS OF SALE AND LICENSE OF TANDY COMPUTER EQUIPMENT AND
SOFTWARE PURCHASED FROM RADIO SHACK COMPANY-OWNED COMPUTER CENTERS, RETAIL
STORES AND RADIO SHACK FRANCHISEES OR DEALERS AT THEIR AUTHORIZED LOCATIONS

USA LIMITED WARRANTY

I. CUSTOMER OBLIGATIONS

- A. CUSTOMER assumes full responsibility that this computer hardware purchased (the "Equipment"), and any copies of software included with the Equipment or licensed separately (the "Software") meets the specifications, capacity, capabilities, versatility, and other requirements of CUSTOMER.
- B. CUSTOMER assumes full responsibility for the condition and effectiveness of the operating environment in which the Equipment and Software are to function, and for its installation.

II. LIMITED WARRANTIES AND CONDITIONS OF SALE

- A. For a period of ninety (90) calendar days from the date of the Radio Shack sales document received upon purchase of the Equipment, RADIO SHACK warrants to the original CUSTOMER that the Equipment and the medium upon which the Software is stored is free from manufacturing defects. **This warranty is only applicable to purchases of Tandy Equipment by the original customer from Radio Shack company-owned computer centers, retail stores, and Radio Shack franchisees and dealers at their authorized locations.** The warranty is void if the Equipment or Software has been subjected to improper or abnormal use. If a manufacturing defect is discovered during the stated warranty period, the defective Equipment must be returned to a Radio Shack Computer Center, a Radio Shack retail store, a participating Radio Shack franchisee or a participating Radio Shack dealer for repair, along with a copy of the sales document or lease agreement. The original CUSTOMER'S sole and exclusive remedy in the event of a defect is limited to the correction of the defect by repair, replacement, or refund of the purchase price, at RADIO SHACK'S election and sole expense. RADIO SHACK has no obligation to replace or repair expendable items.
- B. RADIO SHACK makes no warranty as to the design, capability, capacity, or suitability for use of the Software, except as provided in this paragraph. Software is licensed on an "AS IS" basis, without warranty. The original CUSTOMER'S exclusive remedy, in the event of a Software manufacturing defect, is its repair or replacement within thirty (30) calendar days of the date of the Radio Shack sales document received upon license of the Software. The defective Software shall be returned to a Radio Shack Computer Center, a Radio Shack retail store, a participating Radio Shack franchisee or Radio Shack dealer along with the sales document.
- C. Except as provided herein no employee, agent, franchisee, dealer or other person is authorized to give any warranties of any nature on behalf of RADIO SHACK.
- D. **EXCEPT AS PROVIDED HEREIN, RADIO SHACK MAKES NO EXPRESS WARRANTIES, AND ANY IMPLIED WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE IS LIMITED IN ITS DURATION TO THE DURATION OF THE WRITTEN LIMITED WARRANTIES SET FORTH HEREIN.**
- E. Some states do not allow limitations on how long an implied warranty lasts, so the above limitation(s) may not apply to CUSTOMER.

III. LIMITATION OF LIABILITY

- A. **EXCEPT AS PROVIDED HEREIN, RADIO SHACK SHALL HAVE NO LIABILITY OR RESPONSIBILITY TO CUSTOMER OR ANY OTHER PERSON OR ENTITY WITH RESPECT TO ANY LIABILITY, LOSS OR DAMAGE CAUSED OR ALLEGED TO BE CAUSED DIRECTLY OR INDIRECTLY BY "EQUIPMENT" OR "SOFTWARE" SOLD, LEASED, LICENSED OR FURNISHED BY RADIO SHACK, INCLUDING, BUT NOT LIMITED TO, ANY INTERRUPTION OF SERVICE, LOSS OF BUSINESS OR ANTICIPATORY PROFITS OR CONSEQUENTIAL DAMAGES RESULTING FROM THE USE OR OPERATION OF THE "EQUIPMENT" OR "SOFTWARE." IN NO EVENT SHALL RADIO SHACK BE LIABLE FOR LOSS OF PROFITS, OR ANY INDIRECT, SPECIAL, OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY BREACH OF THIS WARRANTY OR IN ANY MANNER ARISING OUT OF OR CONNECTED WITH THE SALE, LEASE, LICENSE, USE OR ANTICIPATED USE OF THE "EQUIPMENT" OR "SOFTWARE." NOTWITHSTANDING THE ABOVE LIMITATIONS AND WARRANTIES, RADIO SHACK'S LIABILITY HEREUNDER FOR DAMAGES INCURRED BY CUSTOMER OR OTHERS SHALL NOT EXCEED THE AMOUNT PAID BY CUSTOMER FOR THE PARTICULAR "EQUIPMENT" OR "SOFTWARE" INVOLVED.**
- B. RADIO SHACK shall not be liable for any damages caused by delay in delivering or furnishing Equipment and/or Software.
- C. No action arising out of any claimed breach of this Warranty or transactions under this Warranty may be brought more than two (2) years after the cause of action has accrued or more than four (4) years after the date of the Radio Shack sales document for the Equipment or Software, whichever first occurs.
- D. Some states do not allow the limitation or exclusion of incidental or consequential damages, so the above limitation(s) or exclusion(s) may not apply to CUSTOMER.

IV. SOFTWARE LICENSE

RADIO SHACK grants to CUSTOMER a non-exclusive, paid-up license to use the TANDY Software on **one** computer, subject to the following provisions:

- A. Except as otherwise provided in this Software License, applicable copyright laws shall apply to the Software.
- B. Title to the medium on which the Software is recorded (cassette and/or diskette) or stored (ROM) is transferred to CUSTOMER, but not title to the Software.
- C. CUSTOMER may use Software on a multiuser or network system only if either, the Software is expressly labeled to be for use on a multiuser or network system, or one copy of this software is purchased for each node or terminal on which Software is to be used simultaneously.
- D. CUSTOMER shall not use, make, manufacture, or reproduce copies of Software except for use on **one** computer and as is specifically provided in this Software License. Customer is expressly prohibited from disassembling the Software.
- E. CUSTOMER is permitted to make additional copies of the Software **only** for backup or archival purposes or if additional copies are required in the operation of **one** computer with the Software, but only to the extent the Software allows a backup copy to be made. However, for TRSDOS Software, CUSTOMER is permitted to make a limited number of additional copies for CUSTOMER'S own use.
- F. CUSTOMER may resell or distribute unmodified copies of the Software provided CUSTOMER has purchased one copy of the Software for each one sold or distributed. The provisions of this Software License shall also be applicable to third parties receiving copies of the Software from CUSTOMER.
- G. All copyright notices shall be retained on all copies of the Software.

V. APPLICABILITY OF WARRANTY

- A. The terms and conditions of this Warranty are applicable as between RADIO SHACK and CUSTOMER to either a sale of the Equipment and/or Software License to CUSTOMER or to a transaction whereby Radio Shack sells or conveys such Equipment to a third party for lease to CUSTOMER.
- B. The limitations of liability and Warranty provisions herein shall inure to the benefit of RADIO SHACK, the author, owner and/or licensor of the Software and any manufacturer of the Equipment sold by Radio Shack.

VI. STATE LAW RIGHTS

The warranties granted herein give the **original** CUSTOMER specific legal rights, and the **original** CUSTOMER may have other rights which vary from state to state.

Tandy 3000/4000

**MS-DOS
Reference
Manual**

MS-DOS® Software:
©1981, 1985, 1987 Microsoft® Corporation.
Licensed to Tandy Corporation.
All Rights Reserved.

MS-DOS Supplemental Programs:
MS-DOS ©1981, 1985, 1987 Microsoft Corporation.
Licensed to Tandy Corporation.
All Rights Reserved.

Utility Files: ©1985, 1987 Tandy Corporation
All Rights Reserved.

Setup — Compatibility Software:
© Phoenix Software Associates Ltd.
Licensed to Tandy Corporation.
All Rights Reserved.

All portions of this software are copyrighted and are the proprietary and trade secret information of Tandy Corporation and/or its licensor. Use, reproduction, or publication of any portion of this material without prior written authorization by Tandy Corporation is strictly prohibited.

Tandy 3000/4000 MS-DOS Reference Manual:
©1986, 1987 Tandy Corporation.
All Rights Reserved.

Reproduction or use of any portion of this manual without express written permission from Tandy Corporation and/or its licensor is prohibited. While reasonable efforts have been made in the preparation of this manual to assure its accuracy, Tandy Corporation assumes no liability resulting from any errors in or omissions from this manual, or from the information contained herein.

Tandy is a registered trademark of Tandy Corporation.
Microsoft and XENIX are registered trademarks of Microsoft Corporation.

Lotus is a registered trademark of Lotus Development Corp.

Intel is a registered trademark of Intel Corporation.

10 9 8 7 6 5 4 3 2

Contents

1	Files and Directories	1
	Before You Start	1
	File Protection	1
	How MS-DOS Keeps Track of Your Files	2
	Multilevel Directories	2
	Paths and Pathnames	5
	Wildcards	7
	The ? Wildcard	7
	The * Wildcard	7
	Wildcard Examples	8
	Using Directories	8
	How to Create a Directory	9
	How to Change Your Working Directory	9
	How to Display Your Working Directory	9
	How to Delete a Directory	10
	How to Rename a Directory	11
	What Comes Next?	12
2	About Commands	13
	Types of MS-DOS Commands	13
	Redirecting Command Input and Output	16
	How to Redirect Your Output	16
	How to Redirect Input	17
	Filters and Pipes	17
	Command Pipes	17
	What Comes Next?	18
3	MS-DOS Commands	19
	Sample Command	21
	Command Options	22
	More About Options	23
	MS-DOS Commands	24

4 Batch Processing	157
Why Use Batch Files?	157
How to Create Batch Files	157
About Batch Processing	158
The Autoexec.bat File	159
How to Create an Autoexec.bat File	160
How to Create a Batch File with Replaceable Parameters	162
How to Use Named Parameters in a Batch File	163
How to Run a Batch File	165
How to Use Temporary Files	166
Batch Processing Commands	166
5 MS-DOS Editing and Function Keys	179
Special MS-DOS Editing Keys	179
How MS-DOS Uses the Template	179
The MS-DOS Editing Keys	180
How to Use the MS-DOS Template	181
The MS-DOS Control Characters	183
What Comes Next?	184
6 The Line Editor (Edlin)	185
How Edlin Works	185
How to Start Edlin	185
How to End Edlin and Save Your Changes	187
Special Editing Keys	188
Using Edlin Commands	194
Some Tips for Using Edlin Commands	194
Edlin Command Options	196
The Line Option	196
The Question Mark Option	197
The Text Option	197
Edlin's Commands Reference	197

8	Debug	.221
	Introduction	.221
	How to Start Debug	.222
	Method 1: Debug	.222
	Method 2: Command Line	.222
	Debug Command Information	.223
	Debug Command Parameters	.224
	Debug Error Messages	.258

Apendicies

A	Diskette Support	.259
	Types of Drives	.259
	Using Only One Diskette Drive	.260
B	How to Configure Your System	.263
	What is a Configuration File?	.263
	Config.sys Commands	.264
	Sample Config.sys File	.264
	Tandy 4000 Expanded Memory Manager	.279
C	Installable Device Drivers	.281
	Introduction	.281
	Ansi.sys	.283
	Display.sys	.288
	Driver.sys	.289
	Hdrive.sys	.291
	Lpdvr.sys	.293
	Printer.sys	.298
	Modevm.sys	.299
	Spooler.sys	.300
	Vdisk.sys	.301

D Using Your Hard Disk (Fdisk)	303
Hard Disk Types	304
Checking for a Configured and Formatted Disk	305
More About Fdisk	306
How to Create a Dos Partition	308
Selection 1: Create Primary DOS Partition	308
Creating More than One MS-DOS Partition	309
Selection 2: Create Extended DOS Partition	310
Selection 3: Create Logically Drive in the Extended DOS Partition	311
How to Change the Active Partition	312
How to Delete a DOS Partition	313
Selection 1: Delete Primary DOS Partition	314
Selection 2: Delete Extended DOS Partition	314
Selection 3: Delete Logical Drive in the Extended DOS Partition	315
How to Display Partition Data	316
How to Select the Next Fixed Disk Drive	317
E How to Use Code Pages	319
Introduction	319
National Language Support Codes	320
Commands that Support National Languages	321
New MS-DOS Commands	321
Enhanced MS-DOS Commands	321
New and Enhanced Configuration Commands	322
Date and Time Formats	322
How to Use Code Pages	323
How to Set Device Code Pages	325
How to Switch Between Code Pages	326
How to Switch Between Code Pages	326
How to List Current Code Pages	327
How to Refresh Lost Code Pages	327
How to Format a Disk with Country-Specific Information	328
F MS-DOS Message Directory	331

Files and Directories

This chapter explains:

- Protecting and keeping track of your files
- Working with multilevel directories
- Using wildcards

Before You Start

Before you read this chapter, you should already know how to start MS-DOS, format and make backup copies of disks, copy and delete files, and run programs. If you are unfamiliar with any of these procedures, refer to the *Tandy 3000/4000 MS-DOS Handbook* for more information.

File Protection

The MS-DOS operating system is a powerful and useful tool for processing personal files and business information. As with any computer, this information must be protected, since errors can occur and information can be misused. So, if you are doing work that cannot be replaced or that requires a lot of security, you should protect your programs.

You can take simple but effective measures like putting your disks away when you're not using them, or covering the write-protect notch on your program disks. Another way to protect your programs is by installing your equipment in a secure office or work area. If your disks contain valuable information, you should make backup copies of them on a regular basis. For more information on backing up disks, see the **backup** and **restore** commands in Chapter 3, "MS-DOS Commands."

How MS-DOS Keeps Track of Your Files

As you learned in the *Tandy 3000/4000 MS-DOS Handbook*, MS-DOS stores files in directories. In addition to directories, it uses an area on a disk called the *file allocation table*. When you format a disk with the **format** command, MS-DOS copies this table onto the disk and creates an empty directory, called the *root* directory. On each of your disks, the directories store the files, and the file allocation table keeps track of their locations. The table also allocates the free space on your disks so that you have enough room to create new files.

These two system areas, the directories and the file allocation table, enable MS-DOS to recognize and organize the files on your disks. To check these areas for consistency and errors, use the MS-DOS **chkdsk** command. For example, to check the disk in Drive A, type the **chkdsk** command, followed by a:

In response, MS-DOS displays a status report and any errors it has found, such as files that show a nonzero size in the directory but that really have no data in them.

For an example of such a display and for more information on **chkdsk**, see the description of the **chkdsk** command in Chapter 3, "MS-DOS Commands."

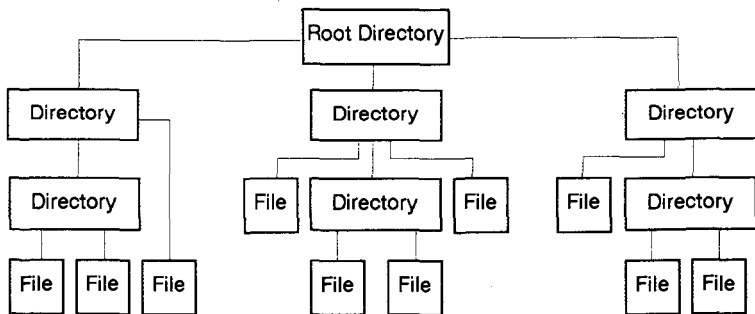
Multilevel Directories

When there is more than one user on your computer, or when you are working on several different projects, the number of files in the directory can become large and unwieldy. To deal with this large number of files, you might want to keep your files separate from a coworker's or organize your programs into convenient categories.

In an office, you can separate and organize files that belong to different people or that relate to specific projects by putting them in different file cabinets. For example, you might put your accounting programs in one file cabinet and your letters in another. You can do the same thing with MS-DOS by putting your files into different directories.

Directories let you group your files in convenient categories. These directories, in turn, can contain other directories (referred to as *subdirectories*). This organized file structure is called a *multilevel* or *hierarchical* directory system.

Note: The maximum number of files or directories that the root directory can contain varies, depending on the type of disk and disk drive you are using. Usually, the maximum number is 112 for a double-sided, double-density, 5 1/4-inch diskette. The maximum number of entries in the root directory of a 1.44 megabyte, 3 1/2-inch diskette is 224. This maximum capacity for a root directory can vary, depending upon how the disk is formatted. The number of subdirectories on a disk is not restricted.



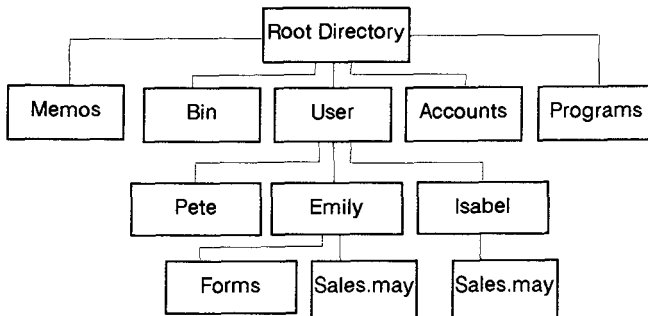
The first level in a multilevel directory is the root directory, which is created automatically when you format a disk and start putting files on it. Within the root directory, you can create additional directories and subdirectories.

As you create new directories for groups of files, or for other people using the computer, the directory system grows. And within each new directory you can add new files or create new subdirectories.

You can move around in the multilevel system by starting at the root and *traveling* through intermediate subdirectories to find a specific file. Conversely, you can start anywhere within the file system and travel toward the root. Or you can go directly to any directory without traveling through intermediate levels.

The directory that you are in is called the *working directory*. When you start your computer, you start out in the working directory. Similarly, when you create a file, you create it in the working directory. The filenames and commands discussed in this chapter relate to your working directory and do not apply to any other directories in the structure.

Because you can put files in different directories, you and your coworkers can have files with the same names, but with unrelated content. The following figure illustrates a typical multilevel directory structure:



In this example, five subdirectories of the root directory have been created. These subdirectories are:

- A directory of external commands, named Bin.
- A user directory containing separate subdirectories for all users of the system.
- A directory containing accounting information, named Accounts.
- A directory of programs, named Programs.
- A directory of text files, named Memos.

As you can see, Pete, Emily, and Isabel have their own directories, which are subdirectories of the User directory. Emily has a subdirectory named Forms, and both Emily and Isabel have Sales.may files in their directories, even though Isabel's Sales.may file is unrelated to Emily's.

This organization of files and directories is not important if you work only with files in your own directory, but if you work with someone else, or on several projects at once, the multilevel directory system becomes handy. For example, you could see a list of the files in Emily's forms directory by typing the following command:

```
dir \user\emily\forms
```

Note that a backslash (\) separates directories from other directories and files. In the previous example, the first backslash includes the root directory. The use of the backslash alone indicates the root directory. For example, the following command displays a list of the files in the root directory:

```
dir \
```

To see the files that Isabel has in her directory, you type the following command:

```
dir \user\isabel
```

This command tells MS-DOS to travel from the root directory to the user directory to the isabel directory, and to then display all filenames in the isabel directory.

Paths and Pathnames

When you use multilevel directories, you must tell MS-DOS where the files are located in the directory system. Both Isabel and Emily, for example, have files named Sales.may, so each must tell MS-DOS the directory in which her file resides when she wants to use it. This is done by giving MS-DOS a *pathname* to the file.

A pathname is a sequence of directory names followed by a filename. Each directory name is separated from the previous one by a backslash (\). A *path* differs from a pathname in that it does not include a filename. The general format of a pathname is as follows:

```
[\directoryname] [\directoryname...] \filename
```

A pathname can contain any number of directory names up to a total length of 63 characters. If a pathname begins with a backslash, MS-DOS searches for the file by starting at the root of the directory system. Otherwise, it begins at the working directory and searches along the path from there. Here are two examples.

The pathname of Emily's Sales.may file is:

`\user\emily\sales.may`

The pathname of Isabel's Sales.may file is:

`\user\isabel\sales.may`

When you are in your working directory, you can use a filename and its corresponding pathname interchangeably. Some sample names are:

<code>\</code>	The root directory.
<code>\programs</code>	A directory under the root directory that contains program files.
<code>\user\isabel\forms\1040</code>	A typical full pathname. This one is for a file named 1040 in the Forms directory, which belongs to Isabel.
<code>Sales.may</code>	A file in the working directory.

A *parent directory* is any directory that contains subdirectories. MS-DOS provides special shorthand notations for the working directory and the parent of the working directory, and automatically creates these two entries whenever you create a directory:

- . MS-DOS uses the shorthand name “.” to indicate the name of the working directory in all multilevel directory listings.
- .. These two dots are the shorthand name “..” for the working directory's parent directory (one level up). If you type the **dir** command followed by two dots, MS-DOS lists the files in the parent directory of your working directory.

If you type the following command, MS-DOS lists the files in the parent's parent directory:

`dir ..\..`

Wildcards

If you are using multilevel directories, it is easier to search for files on your disks if you use two special characters, called *wildcards*. The wildcard characters are the asterisk (*) and the question mark (?). They are useful in MS-DOS command lines because they give you flexibility when you are specifying paths and files.

The ? Wildcard

A question mark (?) in a filename or filename extension means that any character can occupy that position. The following command, for example, lists all filenames on the default drive that begin with the characters “memo”, that have any character in the next position, that end with the characters “aug”, and that have an extension of “.txt”:

```
dir memo?aug.txt
```

Here are some examples of files that might be listed by the preceding command:

```
MEMO2AUG.TXT
```

```
MEMO9AUG.TXT
```

```
MEMOBAUG.TXT
```

The * Wildcard

An asterisk (*) in a filename or filename extension means that any character can occupy that position or any of the remaining positions in the filename or extension. For example, the following command lists all the directory entries on the default drive with filenames that begin with the characters “memo” and that have an extension of “.txt”:

```
dir memo*.txt
```

Here are some examples of files that might be listed by this **dir** command:

```
MEMO2AUG.TXT
```

```
MEMO9AUG.TXT
```

```
MEMOBAUG.TXT
```

```
MEMOJULY.TXT
```

```
MEMOJUNE.TXT
```

MS-DOS ignores any filename characters that follow the asterisk wildcard, up to the period that separates the filename from its extension. For example, the command `dir *1.mem` lists all the files in the directory with the extension “.mem”, not just those files whose names end with the number 1.

Note: The wildcard abbreviation *.* refers to all files in the directory. This feature can be both powerful and destructive when used with MS-DOS commands. For example, the `del` command followed by the wildcard abbreviation *.* deletes all files on the default drive, regardless of filename or extension.

Wildcard Examples:

Suppose you want to find a certain accounting file but can't remember its exact name. You could list the directory entries for all files named accounts in the default directory of Drive A (regardless of their filename extensions). To do this quickly, you could type the following command:

```
dir a:accounts.*
```

Similarly, to list the directory entries for all files with “.txt” extensions or in a directory called Reports (regardless of their filenames) on the disk in Drive B, you could type the following command:

```
dir b:\reports\*.txt
```

This command is useful if your text files have “.txt” extensions. For example, by using the `dir` command with wildcard characters, you could see a list of all your text files—even if you don't remember their filenames. For more information on the `dir` command, refer to Chapter 3, “MS-DOS Commands.”

Using Directories

The following sections describe how to display, change, and delete any directory. It also describes how to create directories and subdirectories.

How to Create a Directory

To create a subdirectory in your working directory, use the **mkdir** (make directory) command. For example, to create a new directory named **User** under your working directory, type the following command:

```
mkdir user
```

After MS-DOS runs this command, a new directory exists under your working directory. You can also make directories anywhere in the directory structure by specifying **mkdir** followed by a path. MS-DOS automatically creates the directory.

To create files in the new directory, you can use the MS-DOS line editor, **edlin**. Chapter 6, “The Line Editor (Edlin),” explains how to use **edlin** to create and save files. You can also create and save files if you have a word processing program, such as **Scipsit**.

How to Change Your Working Directory

With MS-DOS, it is easy to change from your working directory to a different directory: type the **chdir** (change directory) command followed by a path. For example, to change the working directory to **\user**, you type:

```
chdir \user
```

You can also specify any path after the command so that you can travel around the directory structure. The following command, for example, puts you in the parent directory of your working directory:

```
chdir ..
```

How to Display Your Working Directory

All commands are executed while you are in your working directory. You can find out the name of the directory you are in by typing the MS-DOS **chdir** command with no path. For example, if your working directory is **\user\pete**, when you type **chdir** and press **[ENTER]**, you see:

```
A:\USER\PETE
```

This message shows your working drive, **A**, plus your working directory, **\user\pete**.

Shortcut: You can also type the letters **cd** for **chdir** to save time. For example, the following commands are the same:

```
cd \user\pete
chdir \user\pete
```

To see the contents of the \user\pete directory, you can use the MS-DOS **dir** command. The subdirectory might look like this:

Volume in drive A has no ID

Directory of A:\USER\PETE

		<Dir>	08-09-87	10:09a
		<Dir>	08-09-87	10:09a
TEXT		<Dir>	08-09-87	10:09a
FILE1	TXT	5243	08-04-87	9:30a
4 File(s)		836320 byte free		

Note that MS-DOS lists both files and directories in this output. As you can see from the display, Pete has a subdirectory named text; the . refers to the working directory \user\pete; the .. is short for the parent directory \user, and File1.txt is a file in the \user\pete directory. All these directories and files are on the disk in Drive A.

Note: Because files and directories are listed together, you cannot give a subdirectory the same name as a file in that directory. For instance, if you already have a path \user\pete, where Pete is a subdirectory, you cannot create a file named Pete in the \user directory.

How to Delete a Directory

If you create a directory and later decide that you no longer need it, you can delete it with the MS-DOS **rmdir** (remove directory) command.

The **rmdir** command lets you delete any directory by specifying its path, but the directory must be empty except for the . and .. entries. This prevents you from accidentally deleting files and directories.

To remove all the files in a directory (except for the “.” and “..” entries), type `del` followed by the path of the directory.

For example, to delete all files in the `\user\emily` directory, type the following command:

```
del \user\emily
```

MS-DOS prompts you with this message:

```
Are you sure (Y/N)?
```

If you really want to delete all the files in the directory, type `Y` (for Yes). If not, type `N` (for No) to stop the command.

Now you can use the `rmdir` command to delete the `\user\emily` directory by typing the following command:

```
rmdir \user\emily
```

Shortcut: To save time, you can also use the letters `rd` for the `rmdir` command.

How to Rename a Directory

There is no command to rename directories. You can, however, rename a directory that has no subdirectories. Suppose, for example, you want to rename the `\user\pete` directory and call it `\user\emily`. To do this, follow these steps. (Remember to press `ENTER` after each step.)

To create the new directory, type:

```
mkdir \user\emily
```

Then, to copy the files from the old directory to the new directory, type:

```
copy \user\pete\*.* \user\emily
```

Now, to delete the contents of the old directory, type:

```
del \user\pete\*.*
```

(Type `Y` in response to the prompt, `Are you sure?`)

Finally, to remove the old directory, type:

```
rmdir \user\pete
```

What Comes Next?

In this chapter, you've learned more about files and directories, about wildcards and how to use them, and about some basic MS-DOS commands that help you work with files and directories. In the next chapter, you'll learn about the two kinds of commands and about using redirection symbols.

Important Notice for MS-DOS Version 3.3

Backup and **restore** utilities are not always compatible between different versions of MS-DOS. To assure trouble free operations, always use the same version of MS-DOS to restore files that you used to backup those files.

About Commands

This chapter explains:

- Internal and external MS-DOS commands
- Redirecting input and output
- Command grouping symbols

Types of MS-DOS Commands

There are two types of MS-DOS commands:

- Internal commands
- External commands

Internal commands are the simplest, most commonly used commands. When you list the directory on your MS-DOS disk, you cannot see these commands because they are part of a file named Command.com. When you type internal commands, MS-DOS performs them immediately. This is because they were loaded into your computer's memory when you started MS-DOS. Following is a list of the MS-DOS internal commands:

break	del	mkdir	set
chcp	dir	path	shift
chdir	echo	pause	time
cls	exit	prompt	type
copy	for	rem	ver
ctty	goto	ren	verify
date	if	rmdir	vol

Some internal commands can use paths and pathnames. Specifically, four commands—**copy**, **dir**, **del**, and **type**—have greater flexibility when you specify a pathname after the command.

The formats of these commands are as follows:

copy *pathname pathname*

If the second *pathname* is a directory (a *path*), MS-DOS copies all the files you specify in the first *pathname* into that directory, as in the following example:

```
copy \user\pete\*. * sales
```

del *pathname*

If the *pathname* is a directory (a *path*), all the files in that directory are deleted. If you try to delete a path, MS-DOS displays the prompt, Are you sure (Y/N)? Type Y (for Yes) to complete the command, or N (for No) to stop the command. Example:

```
del\user\pete
```

dir *path*

The following command displays the directory for a specific path:

```
dir \user\pete
```

type *pathname*

You must specify a *pathname* (or filename) for this command. MS-DOS then displays this file on your screen in response to the **type** command. Example:

```
type \user\emily\report.nov
```

Any filename with an extension of “.com”, “.exe”, or “.bat” is considered an *external* command. For example, files such as Format.exe and Diskcopy.exe are external commands. Because all external commands are also files, you can create new commands and add them to MS-DOS. Programs that you create with most languages (including assembly language) can be “.exe” (executable) files. Note, however, that when you use an external command, you do not need to type its filename extension.

Note: If you have more than one external command with the same name, MS-DOS can run only one of them, according to the following order: “.com”, “.exe”, “.bat”.

To illustrate this precedence, suppose your disk contains the files Format.exe and Format.bat. If you were to type the external command **format**, MS-DOS would always run the program Format.exe first, and not run the Format.bat file at all.

The following external commands are described in Chapter 3, “MS-DOS Commands”:

append	find	recover
assign	format	replace
attrib	grftab	restore
backup	graphic	select
chkdsk	join	share
command	keyb	sort
comp	label	subst
diskcomp	mode	sys
diskcopy	more	tree
fastopen	nlsfunc	xcopy
fdisk	print	

Before MS-DOS can run external commands, it must read them into memory from the disk. When you give an external command, MS-DOS immediately checks your working directory to find that command. If it isn't there, you must tell MS-DOS which directory the external command is in. Do this by using the **path** command. When you are working with more than one directory, you might find it more convenient to put all the MS-DOS external commands in one directory. Then, when it needs them, MS-DOS can quickly find the external commands at one location.

Suppose, for example, that you are in a working directory named `\user\prog` and that the MS-DOS external commands are in `\bin`. To find the **format** command, you must tell MS-DOS to choose the `\bin` path, as illustrated in the following command. The command tells MS-DOS to search in your working directory and in the `\bin` directory for all commands:

```
path \bin
```

You need only specify this path once. If you want to know the current path, type **path** and press **ENTER**. MS-DOS then displays the working path on the screen.

You can automatically set your path when you start MS-DOS by including the **path** command in a file called `Autoexec.bat`.

For more information on the Autoexec.bat file, refer to Chapter 4, “Batch Processing.”

Redirecting Command Input and Output

Usually, MS-DOS receives input from the keyboard and sends its output to the screen. You can, however, redirect this flow of command input and output. For instance, you might want input to come from a file instead of from the keyboard, and you might want output from a command to go to a file or lineprinter instead of to the screen. By using redirection symbols, you can accomplish this. Redirection symbols also enable you to create pipes that let the output from one command become the input for another command.

How to Redirect Your Output

By default, most commands send output to your screen. If you want to change this and send the output to a file, simply use a greater-than sign (>) in your command. For example, the following command displays on the screen a directory listing of the disk in the default drive:

```
dir
```

The **dir** command can send this output to a file named Contents if you type:

```
dir > contents
```

If the Contents file doesn't exist, MS-DOS creates it and stores your directory listing there. If the Contents file does exist, MS-DOS replaces anything in the file with the new data. To append your directory or add one file to another (instead of replacing the entire file), use two greater-than signs (>>) to tell MS-DOS to append the output of the command (such as a directory listing) to the end of a specified file.

For example, the following command appends your directory listing to an existing file named Contents:

```
dir >> contents
```

If Contents doesn't exist, MS-DOS creates it.

How to Redirect Input

Often, it's useful to have input for a command come from a file instead of from the keyboard. This is possible in MS-DOS by using a less-than sign (<) in your command. For example, the following command sorts the file Names and sends the sorted output to a file called Namelist:

```
sort < names > namelist
```

Filters and Pipes

A *filter* is a command that reads your input, transforms it in some way, and then outputs it to your screen. In this manner, the input is “filtered” by the program.

MS-DOS filters include **find**, **more**, and **sort**. Their functions are:

- find** Searches for text in a file.
- more** Displays the contents of a file one screenful at a time.
- sort** Alphabetically sorts the contents of a file.

You can redirect the output from a filter into a file, or you can use it as input for another filter by using *pipes*. The following section explains how filters are piped together.

Command Pipes

To use the output from one command as the input for another, you can *pipe* the commands to MS-DOS. Piping is done by separating commands with the pipe symbol, which is a vertical bar (|). The following command, for example, displays an alphabetically sorted listing of your directory on the screen:

```
dir | sort
```

The pipe sends all output generated by the **dir** command (on the left side of the bar) as input to the **sort** command (on the right side of the bar). You can also use piping with redirection symbols if you want to send the output to a file. For example, the following command creates a file named Direct.lst on your default drive:

```
dir | sort > direct.lst
```

The `Direct.lst` file now contains a sorted listing of the directory on the default drive.

You can also specify a drive other than the default drive. Suppose, for example, you want to send the sorted data to a file named `Direct.lst` on Drive B. To do this, you could type:

```
dir | sort b:direct.lst
```

You can use more than one pipe on a command line. The following command, for instance, sorts your directory, shows it to you one screen at a time, and puts `--More--` at the bottom of your screen when there is more output to be seen:

```
dir | sort | more
```

Because you can pipe commands and filters together in many different ways, you can find many uses for them.

What Comes Next?

In this chapter, you've learned about internal and external MS-DOS commands and how to redirect their input and output. The next chapter lists each command in alphabetical order along with its syntax and comments about its use.

MS-DOS Commands

This chapter provides details about all the MS-DOS commands. The commands are listed in alphabetical order. In this chapter, you will learn:

- The symbols used to show internal commands, external commands, and commands that won't work over a network
- The structure of the command page
- What command options are
- Notational conventions that are used in this chapter
- How to use each MS-DOS command

External and internal commands are shown by the following symbols:



This shows that the command is internal.



Some MS-DOS commands do not work over a computer network. If you try to use these commands over a network, MS-DOS displays the error message, **Cannot *command*** to a network device, where *command* is the name of the command you typed. If the command does not work over a network (on a shared or remote device), you will see this symbol in the command description.



This shows that the command is external.

For various reasons, the following commands do not work over a network. (See the descriptions of the individual commands for more detailed explanations.)

chkdsk	join
diskcomp	label
diskcopy	recover
fdisk	subst
format	sys

The following page is a sample command page. Each command described in this chapter follows this sample command format.

Sample Command

Purpose:

The “Purpose” section tells you what the command does.

Syntax:

commandname [*options*]

where:

commandname is the name of an MS-DOS command.

options can include *drive*:, *path*, *filename*, *pathname*, *switch*, and/or *argument*.

Comments:

The “Comments” section describes the command and how to use it. It also explains why the command is useful and explores each of the command’s options.

Notes:

The “Notes” section discusses important points related to the command, for instance:

You can specify a drive and/or path before any command, unless otherwise specified in the “Notes” section.

Examples:

The “Examples” section gives one or more examples that illustrate the use of the command.

Command Options

Command options give MS-DOS extra information about a command. If you omit options, MS-DOS either prompts you to supply them, or it uses a default value. Refer to the individual command descriptions in this chapter for default values. MS-DOS commands use the following syntax:

command [*options*]

command is an MS-DOS command, and [*options*] is one or more of the following:

Option	Description
<i>drive:</i>	Refers to a disk drive name. You need to specify a drive name only if you are using a file that is not on the default drive. Information transferred between two disks is sent from a source drive to a target drive.
<i>path</i>	Refers to a directory name with the following syntax: [<i>\directory</i>][<i>\directory...</i>] <i>\directory</i>
<i>filename</i>	Refers to a file, and includes any filename extension. The <i>filename</i> option does not refer to a device or drive name.
<i>pathname</i>	Refers to a path plus a filename. A <i>pathname</i> uses the following syntax: [<i>\directory</i>][<i>\directory...</i>] <i>\filename</i>
<i>switches</i>	Control MS-DOS commands. Switches begin with a slash, for example, /p.
<i>arguments</i>	Provide more information to MS-DOS commands. You usually choose between arguments, for example, on or off.
<i>string</i>	Many commands work with <i>strings</i> of text. A <i>string</i> is a group of characters that can include letters, numbers, spaces, and any other characters. Searching for a particular word in a file is a common use of a <i>string</i> .

More About Options

The Tandy 3000/4000 MS-DOS Reference Manual uses the following conventions for command options:

Convention	Usage
italics	You must supply the text for any variable item shown in italics. For example, when <i>filename</i> appears, type the name of your file.
[brackets]	Items in brackets are optional. To include optional information, type only the information within the brackets. Do not type the brackets themselves.
...	An ellipsis (...) means that you can repeat an item as many times as necessary.
separators	Unless otherwise specified, you must use spaces to separate commands from their options, for example:

rename dull.doc sharpe.doc

With some commands, you can use a semicolon (;), an equal sign (=), or a tab to separate MS-DOS commands from their options. These characters are also known as separators. In this manual, spaces separate commands from their options.

The following pages briefly describe each MS-DOS command.

MS-DOS Commands

This chapter describes the following MS-DOS commands. Note that synonyms for commands are in parentheses.

Note: If you have only one diskette drive, refer to Appendix A, "Instructions for Users with Single Diskette Drive Systems," before running any of the following commands.

append	Sets a search path for data files.
assign	Assigns a drive letter to a different drive.
attrib	Sets or displays file attributes.
autofmt†	Prepares a hard disk for use with MS-DOS
backup	Backs up one or more files from one disk to another.
break	Sets CTRL-C check.
cache†	Establishes a RAM buffer for speeding disk access.
chcp	Displays or changes the current code page for the command processor, Command.com.
chdir	Changes directories or prints the working directory (cd).
chkdsk	Scans a disk drive and checks for consistency.
cls	Clears the screen.
command	Processes internal MS-DOS commands.
comp	Compares the contents of two sets of files
copy	Copies the specified file or files.
ctty	Changes the device from which you issue commands.
date	Displays and sets the date.
dc†	Compresses or decompresses files.
del	Deletes the specified file or files (erase).
dir	Lists the requested directory entries.
diskcomp	Compares disks.
diskcopy	Copies disks.

diskopt†	Optimizes disk file storage.
disktype	Determines a disk's type. Provides example format commands for diskettes.
exit	Exits the command processor, returns to the previous level.
fastopen	Decreases the time required to open frequently-used files and directories.
fdisk	Configures hard disks for MS-DOS.
find	Searches for a constant string of text.
fmat2000†	Formats a diskette for 720 kilobytes of storage.
format	Formats a disk to receive MS-DOS files.
graftabl	Loads a table of graphics characters.
graphics	Prepares MS-DOS for printing graphics.
hsect	Initializes a hard disk for formatting.
join	Joins a disk drive to a pathname.
keyb\$	Loads a keyboard program.
label	Labels disks.
lf	Turns off or on printer line feeds following carriage returns.
mkdir	Makes a directory (md).
mode	Sets operation modes for devices.
mon386†‡	Lets you load up to nine programs at once.
more	Displays output one screen at a time.
nlsfunc\$	Loads country-specific information.
patch†	Modifies disk files.
path	Sets a command search path.
print	Prints files.
prompt	Changes the prompt to one you specify.
rcrypt†	Encrypts or decrypts files.
recover	Recovers a bad disk or file.

ren	Renames the first file as the second file (rename).
replace	Replaces previous versions of files.
restore	Restores backed up files.
rmdir	Removes a directory (rd).
select§	Installs MS-DOS on a new diskette with desired country-specific information and keyboard layout.
set	Sets one string value to another in the environment, or displays the environment.
setup	Stores your computer's configuration in CMOS
share	Installs file sharing and locking.
shiptrakt†	Prepares a hard disk for moving.
sort	Sorts data forward or backward.
spooler	Sets the print spooler parameters.
subst	Substitutes a string for a pathname.
sys	Transfers MS-DOS system files from one drive to the drive specified.
time	Displays and sets the time.
tree	Displays directory and file names.
type	Displays the contents of a file.
ver	Prints the MS-DOS version number.
verify	Verifies all writes to a disk.
vol	Displays the volume label.
xcopy	Copies files and subdirectories.

† Tandy Utilities. These programs are not normally a part of MS-DOS 3.3

‡ These commands are only available for the Tandy 4000.

§ These commands are on a separate International Drivers Diskette (Cat. No. 700-4109). This diskette contains all the files you need to use International and Code Page commands.

All MS-DOS commands and the Tandy Utilities are described in detail on the following pages.

Append

External

Purpose:

Sets a search path for data files.

Syntax:

First use only:

```
append [/x] [/e]
```

To specify directories to be searched:

```
append [drive:]path [:[drive:]path ]...
```

To delete appended paths:

```
append;
```

where:

path is the directory that MS-DOS searches for a data file.

Comments:

Append lets you set a search path for data files. It accepts switches only the first time you use it. **Append** accepts these switches:

Switch	Purpose
/x	Extends the search path for data files. When looking for data files, MS-DOS first searches the current directory. If it doesn't find the needed data files there, it searches the first directory in the append search path. If MS-DOS still can't find the files, it continues to the next appended directory, and so on. MS-DOS will not search subsequent directories once it locates the data files.
/c	Stores appended directories in the MS-DOS environment. You can specify more than one path to search by separating each with a semicolon (;). If you type the append command with the path option a second time, MS-DOS discards the old search path and uses the new one.

If you don't use options with the **append** command, MS-DOS displays the current data path.

If you use the following command, MS-DOS sets the NUL data path:

```
append ;
```

This means that MS-DOS searches only the working directory for data files.

Notes:

- You can use the **append** command across a network to locate remote data files.
- If you are using the MS-DOS **assign** command, you must use the **append** command before **assign**.
- To set a search path for external commands, see the **path** command in this chapter.
- **Append** searches the data path for all files, regardless of their file extensions, only with the following MS-DOS system calls:

Code	Function
0FH	Open File (FCB)
23H	Get (FCB) File Size
3DH	Open Handle
11H	FCB search first (with /x switch only)
4EH	Handle find first (with /x switch only)
4BH	Exec (with /x switch only)

Examples:

Suppose you want to access data files in a directory called Letters on Drive B and in a directory called Reports on Drive A. To do this, use the following command:

```
append b:\letters;a:\reports
```

Suppose you wanted to use the /x extension switch so that **append** first searched the current directory for data files before using the appended search paths. To do this, you would type this command before you typed any other **append** command:

```
append /x
```

If you then typed the following command, MS-DOS would first search your current directory for data files. If MS-DOS didn't find the data files in your current directory, it would search the directory called \neworder on Drive C. If the files were not there, MS-DOS would search \bakorder on Drive C.

append c:\neworder;c:\bakorder

Assign

External

Purpose:

Assigns a drive letter to a different drive.

Syntax:

```
assign [drive1[ = ] drive2[ ... ] ]
```

where:

drive1 is the drive to which MS-DOS currently reads and writes.

drive2 is the drive to which you want MS-DOS to read and write.

Comments:

The **assign** command lets you read and write files on drives other than A and B for applications that use only those two drives. You cannot assign a drive being used by another program, and you cannot assign an undefined drive.

Do not type a colon after the drive letters *drive1* and *drive2*.

Notes:

To ensure compatibility with future versions of MS-DOS, use the **subst** command instead of **assign**. The following commands, therefore, are equivalent:

```
assign a = c
```

```
subst a: c:\
```

Because the **assign** command disguises the true device type, you should not use **assign**:

- with commands that require drive information (**backup**, **restore**, **label**, **join**, **subst**, **print**)
- during normal use of MS-DOS

Two other commands, **format** and **diskcopy**, ignore drive reassignments.

Examples:

To reset all drives to their original assignments, type the following command and press **ENTER**:

assign

To run an application on a hard disk drive, C, when the application requires you to put your program disk in Drive A and your data disk in Drive B, you would type:

assign a = c b = c

All references to Drives A and B would then go to Drive C.

Attrib

External

Purpose:

Displays or changes the attributes of selected files in a directory.

Syntax:

```
attrib [+/-r] [+/-a] [drive:]pathname [/s]
```

where:

- +r sets the read-only attribute of a file.
- r disables read-only mode.
- +a sets the archive attribute of a file.
- a clears the archive attribute of a file.

Comments:

The **attrib** command sets read-only and/or archive attributes for files. You can use wildcards to specify a group of files. The attributes of those files matching *filename* are displayed or modified based on the switch selection. **Attrib** doesn't accept a directory name as a valid filename.

The *drive:* and *pathname* specify the location of the file or files you want to reference. The /s switch processes all subdirectories as well as the specified path.

The **backup**, **restore**, and **xcopy** commands use the archive attribute as a control mechanism. Use the +a and -a options to select files that you want to back up with the **backup /m** command or copy with the **xcopy /m** or **xcopy /a** commands.

Notes:

If an application creates a file that has read and write permission, **attrib** forces read-only mode to allow file sharing over a network.

Examples:

To display the attribute of a file called News86 on the default drive, you would type:

```
attrib news86
```

The following command gives the file Report.txt read-only permission:

```
attrib +r report.txt
```

Setting a file as read-only prevents you from accidentally deleting or modifying it.

To remove read-only permission from the files in the \usr\pete directory on Drive B and from the files in any of its subdirectories, you would type:

```
attrib -r b:\usr\pete /s
```

As a final example, suppose you want to give a coworker a disk that contains all files in the default directory of the disk in Drive A, except for files with the extension .bak that contain old copies of edited files. To copy these files to a disk in Drive B, you would type:

```
attrib +a a:*. *  
attrib -a a:*.bak
```

and:

```
xcopy a: b: /m
```

or:

```
copy a: b: /a
```

If you use the **xcopy** /m switch, **xcopy** automatically turns off the archive bits of the files in Drive A as it copies them.

Autofmt

External

Purpose:

Initializes a hard disk for use and, if the hard disk is the primary hard disk, installs the MS-DOS system files required to make the disk bootable.

Syntax:

autofmt [/B]

Comments:

The **autofmt** command accepts one switch:

/b Tells **autofmt** to prompt you for bad sector information before formatting.

Autofmt makes hard disk initialization and formatting easy. In fact, for disks that have already received a low-level, or *hard*, format (for instance, with **hsect** or previous use of **autofmt**), the process can be completely automatic.

When you run **autofmt**, it:

- Performs a hard format of the hard disk (if the disk has not already received hard formatting).
- Partitions the hard disk into one or more MS-DOS partitions and initializes all partitions for use by MS-DOS.
- Copies the system files to the disk to make it bootable, if you tell **autofmt** to format a *primary* drive (Drive C).
- Creates a Config.sys file in the root directory of the disk (or in the first partition) if such a file is needed to set the system for specific drives and multiple partitions.
- Writes the hard drive's parameters to the system's CMOS RAM.
- Writes a hard disk report in the root directory of the disk (or in the first partition), containing information about the type of hard drive and its bad sectors (if any). To display this report file Autofmt.rep, use the **type** command.

Notes:

- If the disk is already hard formatted with **hsect** (or by previous use of **autofmt**) and you entered the bad sector information, the disk contains a table defining the disk type and bad sector locations. You do not need to use the **/b** switch. If you do include the **/b** switch, **autofmt** appends to the existing table any additional bad sector information you enter .

If your hard disk has not been previously formatted, use the **/b** switch with **autofmt**. Use the information that came with your hard disk to type bad sector information as requested by **autofmt**.

- If a hard disk has a capacity of more than 32M, it must be divided into more than one *partition*. MS-DOS treats each partition as a separate disk drive. **Autofmt** creates as many partitions as your drive requires and assigns drive names to each partition (d:, e:, and so on).
- If your disk requires more than one partition, **autofmt** installs in the Config.sys file the configuration information MS-DOS requires to manage the additional partition(s).
- The **autofmt** process can require up to 1 hour to complete. The time required depends on the capacity of your disk and whether your disk received hard formatting from Tandy.

Examples:

To use **autofmt** with a disk that has received hard formatting, boot the system and then, with your backup copy of the Startup Diskette in Drive A, type:

autofmt

A screen prompt asks you to:

Enter hard drive number to format (1 or 2) :

Press **[1]** to format your *primary* drive (Drive C).

Press **[2]** to format your *secondary* drive (Drive D).

The rest of the process is automatic.

To use **autofmt** with a disk that has not received hard formatting, place your Startup diskette in Drive A. Then, type the following command, and press **ENTER** :

autofmt /b

A screen prompt asks you to:

Enter hard drive number to format (1 or 2) :

Press **1** to format your *primary* drive (Drive C).

Press **2** to format your *secondary* drive (Drive D).

Next, **autofmt** asks you to enter any bad sector information. Type the data in the format indicated on the screen.

After you enter the data, the rest of the process is automatic.

As **autofmt** performs its operations, screen displays appear to indicate its progress.

When **autofmt** is complete, a prompt asks you to:

Remove diskette from drive A:, and

hit < Ctrl > < Alt > < Del >

Pressing **CTRL ALT DEL** causes your system to reboot using Drive C. Copy to Drive C any system files, commands, utilities, and any other programs you want on your hard disk.

Backup

External

Purpose:

Backs up one or more files from one disk to another.

Syntax:

```
backup [drive1:][ path][ filename] [drive2:] [/s]/[m]/[a] [/f] [/d:date]
[/t:time] [/L:][drive:][ path]filename]]
```

where:

drive1 is the disk drive that you want to back up.

drive2 is the target drive to which the files are backed up.

Comments:

The **backup** command can back up files on disks of different media (hard disks and diskettes). **Backup** also backs up files from one diskette to another, even if the disks have a different number of sides or sectors.

The **backup** command accepts the following switches:

Switch	Purpose
/s	Backs up subdirectories.
/m	Backs up only those files that have changed since the last backup.
/a	Adds the files to be backed up to those already on the backup disk. It does not erase old files on the backup disk. You cannot use this switch if files exist that were backed up using backup from MS-DOS version 3.2 or earlier.
/f	Formats the target disk if it is not already formatted. For this switch to function, the current path must be able to access the MS-DOS format command.
/d:date	Backs up only those files that you last modified on or after <i>date</i> .
/t:time	Backs up only those files that you last modified at or after <i>time</i> .
/L:drive:\path\filename	Makes a backup log entry in the specified file. If you do not specify a filename, backup places a file called Backup.log in the root directory of the disk that contains the files being backed up.

A backup log file uses the following format:

- The first line lists the date and time of the backup.
- A line for each backed-up file lists the filename and number of the backup disk on which the file resides.

If the backup log file already exists, **backup** appends the current entry to the file.

You can also use the backup log file when you need to restore a particular file from a diskette, but you must specify which disk to restore so that the **restore** command does not have to search for files. The **restore** command always puts a file back in the same place from which it was backed up. **Backup** displays the name of each file as it is backed up.

You should label and number each backup disk consecutively to help you restore the files properly using the **restore** command. If you are sharing files, MS-DOS lets you back up **only** those files to which you have access.

Notes:

- You cannot use an old version of the **restore** command (MS-DOS 3.2 or earlier) for files backed up with the MS-DOS 3.3 **backup** command.
- Unless you use the /a switch, **backup** erases the old files on a backup disk before adding new files to it.
- Do not use the **backup** command if the drive you are backing up has been assigned, joined, or substituted with the **assign**, **join**, or **subst** commands. If you do, you might not be able to restore the files with the **restore** command.

The **backup** program returns the following exit codes:

Code	Function
0	Normal completion
1	No files were found to back up
2	Some files not backed up due to sharing conflicts
3	Terminated by user
4	Terminated due to error

You can use the batch processing **if** command for error processing that is based on the error level returned by **backup**.

Examples:

Suppose Emily wants to back up all the files in the \user\emily directory on Drive C to a blank, formatted disk in Drive A. To do this, she would type:

```
backup c:\user\emily a:
```


Break

Internal

Purpose:

Sets CTRL-C check.

Syntax:

break [on]

or

break [off]

Comments:

Depending on the current process, you can use **CTRL** **C** to stop an activity (for example, to stop sorting a file). Normally, MS-DOS checks to see whether you press **CTRL** **C** while it is reading from the keyboard or writing to the screen or printer. If you set **break** to on, you extend CTRL-C checking to other functions, such as disk reads and writes.

Notes:

Some programs respond to **CTRL** **C** at any time. Setting **break** does not affect these programs.

Examples:

To check for CTRL-C only during screen, keyboard, and printer reads and writes, type:

break off

To see how **break** is set, type the **break** command and press **ENTER** .

Cache

External

Purpose:

Speeds up disk read or read/write operations by establishing a RAM buffer for storing data coming from or going to a disk.

Syntax:

To load cache:

```
cache [buffer size][/a][/e]
```

To configure cache:

```
cache [drive][/8][/9][/c][/g][/h][/o][/r][/s][/t][/w]
```

where:

buffer size is the size of the **cache** buffer, in kilobytes. If you do not specify *buffer size*, **cache** sets the buffer at 30K.

Comments:

The **cache** buffer reduces the number of times the system has to turn on a drive and locate sectors. Depending on the type of program you are running, and the size you establish for the **cache** buffer, using this utility can greatly reduce disk access time.

After initialization, you can use **cache** again to examine the **cache** operations and to set further options.

Cache accepts the following switches:

Switch	Purpose
/a	Causes the cache buffer to reside in <i>expanded</i> RAM. This option is valid for computers with an expanded memory board compatible with the Lotus/Intel/Microsoft expanded memory specifications and that have previously loaded an expanded memory management driver (Tmm.sys).
/e	Causes the cache buffer to reside in <i>extended</i> RAM. This option is valid only for Tandy 3000 and 4000 computers with extended RAM installed.

Other Options:

After you install **cache**, the following configuration parameters are available. (Type **cache**, followed by one or more parameters.)

<i>drive:</i>	Is the drive letter of the drive for which all the following parameters are effective. If you do not specify <i>drive:</i> , cache assumes the current drive.
/8	Sets cache for a diskette formatted at 8 sectors-per-track.
/9	Sets cache for a diskette formatted at 9 sectors-per-track.
/c	Clears the accumulated cache statistics. (See the /s switch.)
/g	Displays the current cache status in the following format: Tandy cache version 01.00.00 memory cache Current cache status Buffer Size 30 Kb Normal Ram Version 4 seconds floppy valid time Current Cache Distribution A: 70% 9 Sectors/Track B: 30% 9 Sectors/Track
/h	Sets cache for a high-capacity (5 1/4-inch, 1.2-megabyte or a 3 1/2-inch, 1.44-megabyte) diskette.

- /o** Toggles **cache** on and off. When you first run **cache**, the program turns caching on for all hard disk drives and off for all diskette drives.
- /r** Resets **cache** and clears all data in the **cache** buffer. Use **/r** before switching diskettes to ensure that all cached data is cleared from the **cache** buffer.
- /s** Displays **cache** statistics in the following format:

Tandy CACHE version 01.00 memory cache

Cache Statistics All Drives

Hit Ratio	=	56%
# Hits	=	8422
# Reads	=	14821
# Sectors Read	=	9881
# Interrupts	=	4786
# Ints Made	=	3100
Disk Read Time:		0:28:33:40
Disk Write Time:		0:10:10:49

The Hit Ratio indicates the percent of successful *hits* (the percentage of times the system finds the requested data in the **cache** buffer).

Hits is the total number of **cache** hits.

Reads is the number of times **cache** looked in the buffer for requested data.

Sectors Read is the total number of sectors that **cache** has loaded into the buffer since beginning operation.

Interrupts is the total number of times the system has called the **cache** routine to read one or more disk sectors.

Ints Made is the number of times **cache** has not intercepted calls to read disk sectors.

Disk Read Time and Disk Write Time show the amount of time the system has spent reading from and writing to the disk.

- /T:seconds** Sets the time in seconds that data stored in the **cache** buffer from diskette drives remains valid. After setting *seconds* for a drive, do not remove the diskette until you are sure the specified time has elapsed since the system last accessed the diskette. Setting *seconds* to 0 tells **cache** not to check the time and to assume that the diskette is not to be removed. Use the **/r** parameter before removing a diskette after setting the time to 0. If you do not specify *seconds*, **cache** uses 4 seconds.
- /W** Toggles **cache** writing on and off. When you initialize **cache**, writing is enabled (on) and the system saves disk writes to the buffer. Specifying **/w** once disables writing to the buffer. Specifying **/w** again causes **cache** to save disk writes to the buffer. This switch is only valid for buffers of less than 100K.

Notes:

- The Tandy Expanded Memory Management driver (Temm.sys) uses the computer's *shadow RAM* to simulate expanded memory and does not require the installation of an expanded RAM board.

Examples:

The following command turns on **cache**, reserving a default buffer size of 30K.

```
cache
```

The next command sets the valid time for Disk Drive A to 1 second and causes **cache** to stop sending disk writes to the buffer.

```
cache a: /t:1 /w
```

To turn off the **cache** functions, type this command:

```
cache /o
```

Chcp

Internal

Purpose:

Displays or changes the current code page for the command processor, Command.com.

Syntax:

`chcp [nnn]`

where:

nnn is the code page to start.

Comments:

The **chcp** command accepts one of the two prepared system code pages as a valid code page. An error message appears if you select a code page that has not been prepared for the system. If you type the **chcp** command without a code page, **chcp** displays the active code page and the prepared code pages for the system.

You can select any one of the prepared system code pages defined by the country command in Config.sys. The following are valid code pages:

Value	Code Page
437	United States
850	Multilingual
860	Portuguese
863	French-Canadian
865	Nordic

Any program that you run after starting a new code page will use the new code page. Programs that started before the new code page will continue to use the original code page.

To see the current code page setting, type:

`chcp`

MS-DOS responds with a message similar to:

Active code page: 850

Prepared system code pages: 850 437

If you select a code page that is not prepared for the system, MS-DOS displays a message similar to:

Code page 850 not prepared for system

Active code page: 437

Prepared system code pages: 437 865

If a device (screen, keyboard, printer) is not prepared for a code page, MS-DOS displays this error message:

Code page 850 not prepared for device xxx

Notes:

In the preceding example, the `chcp` command still changes the active code page, even if the selected code page is not prepared for a device. To change the active code page to the original code page, you must reissue the `chcp` command with the original code page selected.

Examples:

To set the code page for the current screen group to 863 (French-Canadian), you would type:

`chcp 863`

Chdir

Internal

Synonym:

`cd`

Purpose:

Changes a directory to a different path. Displays the working directory.

Syntax:

`chdir [path]`

Comments:

The **chdir** command changes your working directory to the directory you specify. A shorthand notation for the **chdir** command is **cd**. Thus, typing either of the following commands changes your current directory to the directory called Primetim:

`chdir \primetim`

`cd \primetim`

To display the name of your working directory, simply type:

`cd`

You can use two shortcuts to change your directory to a parent directory or subdirectory of your working directory. To illustrate, suppose you have a directory called Specials that has a subdirectory called Sponsors.

To change your working directory to `\specials\sponsors`, you would type:

`cd \specials\sponsors`

If your working directory is Specials, you can type the following command to change to the `\specials\sponsors` directory:

`cd sponsors`

Then, when you wanted to change your working directory back to the parent directory, `\specials`, you could type:

`cd ..`

Type `cd \` to return to the root directory. The root directory is the highest-level directory on your computer and is usually the directory that you see when you start MS-DOS.

Examples:

To display the name of the working directory, type `chdir` without a path. For example, if the working directory is `\user\pete` on Drive B, and you type, `chdir b:`, followed by ENTER, MS-DOS displays:

b:\user\pete

Chkdsk



Purpose:

Scans the disk in the specified drive and checks it for errors.

Syntax:

```
chkdsk [drive:][pathname] [/f] [/v]
```

Comments:

The **chkdsk** command shows the status of your disk. You should run **chkdsk** occasionally on each disk to check for errors. If you run **chkdsk** on a disk and **chkdsk** finds errors, it displays the error messages, followed by a status report.

A typical status report might look like this:

```
21309440 bytes total disk space
  53248 bytes in 2 hidden files
  30720 bytes in 14 directories
 5576704 bytes in 342 user files
  10240 bytes in bad sectors
15638528 bytes available on the disk

 655360 bytes total memory
 448704 bytes free
```

If you type a filename after **chkdsk**, MS-DOS displays a status report for the disk and for the individual file.

The **chkdsk** command accepts the following switches:

- /f Fixes errors on the disk. If you do not specify this switch, **chkdsk** does not correct errors that it finds in your directory. However, it does display messages about files that need to be fixed.
- /v Displays the name of each file in each directory as it checks the disk.

If you specify the /f switch, **chkdsk** displays an error message if it finds any open files on the disk. If you do not specify the /f switch and there are open files, **chkdsk** might give results that make it seem like there are lost clusters on the disk. This happens when the file allocation table has yet to be updated regarding open files. If **chkdsk** reports a large number of clusters as lost, you should consider repairing the disk.

Notes:

- **Chkdsk** does not correct errors on a disk unless you specify the /f switch. For more information on **chkdsk** errors, refer to the specific error message in Appendix F, "MS-DOS Message Directory."
- **Chkdsk** doesn't work on drives used in the **subst** or **join** commands.

Examples:

To save a **chkdsk** status report for future use, you can redirect the output from **chkdsk** to a file, status, by typing:

```
chkdsk a:status
```

Chkdsk then sends the errors it finds to the specified file. Remember, though, not to use the /f switch when you redirect **chkdsk** output.

If **chkdsk** finds errors on the disk in Drive A and you want to try to correct them, type:

```
chkdsk a: /f
```

Chkdsk now tries to correct any errors it finds on the disk in Drive A, prompting you for further information when necessary.

Cls

Internal

Purpose:

Clears the screen.

Syntax:

`cls`

Comments:

The `cls` command clears your screen, leaving only the MS-DOS prompt and a cursor.

Examples:

You might find it more comfortable to work with a “clean slate.” To start a new process with a clear screen, type:

`cls`

Command

External

Purpose:

Starts the command processor.

Syntax:

command [*drive:*][*path*][*ctty-dev*] [/e:*nnnnn*][/p][/c *string*]

where:

ctty-dev lets you specify a different device (such as AUX) for input and output.

Comments:

This command starts a new command processor (the MS-DOS program that contains all internal commands).

When you start a new command processor, you also create a new command environment. This new environment is a copy of the old, parent environment. However, you can change the new environment without affecting the old one.

The command processor is loaded into memory in two parts: transient and resident. Some application programs write over the transient memory part of Command.com when they run. When this happens, the resident part of the command processor looks for the Command.com file on disk so that it can reload the transient part.

The *drive:* and *path* options tell the command processor where to look for the Command.com file if it needs to reload the transient part into memory.

Command accepts these switches:

Switch	Purpose
<i>/e:nnnnn</i>	Specifies the environment size, where <i>nnnnn</i> is the size in bytes, ranging from 160 through 32,768. MS-DOS rounds this number up to the next logical paragraph boundary. The default value is 160 bytes.
<i>/p</i>	Keeps the secondary command processor in memory and does not automatically return to the primary command processor.
<i>/c string</i>	Tells the command processor to perform the command or commands specified by <i>string</i> and then to return automatically to the primary command processor.

If *nnnnn* is less than 160 bytes, MS-DOS defaults to 160 bytes and displays this message:

Invalid environment size specified

If *nnnnn* is greater than 32,768 bytes, MS-DOS displays the same message, but defaults to 32,768 bytes.

Notes:

For more information about the `ctty-dev` option, see the explanation of the `ctty` command in this chapter.

Examples:

The following command tells the MS-DOS command processor to do three things:

- Start a new command processor under the current program
 - Run the command `chkdsk b:`
 - Return to the first command processor
- command `/c chkdsk b:`

Comp

External

Purpose:

Compares the contents of two sets of files.

Syntax:

```
comp [drive:][pathname1] [drive:][pathname2]
```

Comments:

The **comp** command compares one file or set of files (*pathname1*) with a second file or set of files (*pathname2*). These files can be on the same drive or on different drives. They can also be in the same directory or in different directories.

The two sets of files you want to compare can have the same path and filenames – provided they are on different drives. If you type only a drive letter for the second option, **comp** assumes that *pathname2* is the same as *pathname1*. You can use wildcards (* and ?) to specify the pathnames.

If you do not type the *pathname* options or if you omit the *pathname2* option, **comp** prompts you for them. If either option contains only a drive or a path with no filename, **comp** assumes the *.* filename.

If the files you want to compare are on a different disk than **comp**, type the command with no options. When **comp** prompts you for the *pathname* options, you can insert the correct disk and type the filenames to be compared.

As **comp** proceeds, it displays the paths and names of the compared files. A message appears if **comp** cannot find a file matching the *pathname2* option, or if a directory path is invalid. If no file matches the *pathname1* option, **comp** prompts you for both the *pathname* options again.

During the comparison, a message appears for any location in the two files that contains mismatching information. The message indicates the offset into the files of the mismatching bytes and the contents of the bytes themselves (all in hexadecimal notation). The message has the following format:

Compare error at OFFSET XXXXXXXX

file1 = XX

file2 = XX

In this format, *file1* is the first filename typed, and *file2* is the second filename typed. After it finds ten mismatches, **comp** stops comparing and displays:

10 Mismatches - ending compare

If the file sizes are different, **comp** displays:

Files are different sizes, do you wish to
continue (Y/N)?

You can either continue the comparison or end it. If you choose to continue, **comp** compares the files until it reaches the end of the shorter file.

After a successful comparison (one containing no mismatches), **comp** displays:

Files compare OK

After the comparison of the two files ends, **comp** proceeds with the next pair of files that match the two pathname options, until it can find no more files that match the pathname1 option. **Comp** then displays:

Compare more files (Y/N)?

You now can compare two more files or end the comparison. To compare two more files, type Y (for Yes). **Comp** prompts you for two new path options.

For all file comparisons, **comp** first ensures that both files include end-of-file (CTRL-Z) marks. If they do not, **comp** displays this message, and the files are not compared:

EOF mark not found

Examples:

In the following example, **comp** compares each file with the extension .asm in the current directory on Drive C with each file of the same name (but with an extension of .bak) in the current directory on Drive B:

```
comp c:*.asm b:*.bak
```

Copy

Internal

Purpose:

Copies one or more files to another location. This command also appends files.

Syntax:

To copy files:

```
copy [drive:]pathname1 [drive:][pathname2 ][/v]/a[/b]
```

or

```
copy [drive:]pathname1 [/v]/a[/b][drive:][pathname2]
```

To append files:

```
copy pathname1 + pathname2 [...] pathnamexx
```

Comments:

If you do not specify *pathname2*, **copy** creates the duplicate in the working directory on the disk in the default drive. This copy has the same name, creation date, and creation time as the original file (*pathname1*). If the original file is on the default drive and you do not specify *pathname2*, the **copy** command stops. (You cannot copy a file to itself.) MS-DOS displays this error message:

File cannot be copied onto itself

0 File(s) copied

The **copy** command accepts the following switches:

Switch	Purpose
/v	Causes MS-DOS to verify that the sectors written on the target disk are recorded properly. If MS-DOS cannot verify a write, it displays an error message. Although recording errors rarely occur with the copy command, the /v switch lets you verify that critical data has been correctly recorded. It also causes the copy command to run more slowly, because MS-DOS must check each entry recorded on the disk.
/a	Lets you copy ASCII files. This switch applies to the filename preceding it and to all remaining filenames in the command, until copy encounters another /a or /b switch. This switch tells the command processor to read until it encounters the end-of-file mark.
/b	Lets you copy binary files. This switch applies to the filename preceding it and to all remaining filenames in the command, until copy encounters another /a or /b switch. This switch tells the command processor to read the number of bytes specified by the file size in the directory.

Notes:

- The **copy** command switches /a and /b perform differently, depending on whether you place them after the source filename or the target filename.
- When used with a source filename:

/a	Treats the file as an ASCII (text) file. Data in the file is copied up to but not including the first end-of-file mark. (In edlin , this is CTRL-Z.) The remainder of the file is not copied.
/b	Copies the entire file, including any end-of-file marks.

- When used with a target filename:

/a adds an end-of-file character as the last character of the file.
For example:

`copy memo.doc /a letter.doc`

/b Does not add an end-of-file character. For example:

`copy billing.asm /b billing2.asm`

- When you are combining files, the default switch is always **/a**.
- Do not try to append files if one of the source filenames has the same name or extension as the target. To copy all a directory's files and sub-directories, use the **xcopy** command instead of **copy**. Refer to the explanation of the **xcopy** command in this chapter for more information.

Examples:

To copy a file called **Animal.typ** from your working drive and directory to a directory on Drive C called **Bigcats**, type:

`copy animal.typ c:\bigcats`

The **copy** command also lets you append files. To do this, simply list any number of files as options to copy, each separated by a plus sign (+), and then specify a target file to send the combined files to. For example:

`copy intro.rpt + body.rpt + b:sum.rpt report`

This command combines files named **Intro.rpt**, **Body.rpt**, and **Sum.rpt** (from Drive B), and places them in a file called **Report** on the default drive. When you are appending files, the target file is created with the current date and time. If you omit the target file, MS-DOS combines the files and stores them under the name of the first specified file.

You can also combine several files into one by using wildcards. For example:

`copy *.txt combin.doc`

This command takes all files with an extension of **.txt** and combines them into one file named **Combin.doc**.

In the following example, each file that matches *.txt is combined with its corresponding .ref file. The result is a file with the same filename but with the extension .doc. Thus, File1.txt is combined with File1.ref to form File1.doc, Xyz.txt with Xyz.ref to form Xyz.doc, and so on:

```
copy *.txt + *.ref *.doc
```

The following **copy** command combines all files matching *.txt and all files matching *.ref into one file named Combin.doc:

```
copy *.txt + *.ref Combin.doc
```

Copy compares the filename of the source file with the filename of the target. If they are the same, that one input file is skipped, and MS-DOS displays the error message:

Content of destination lost before copy

Further joining proceeds normally. For example, the following command appends all *.txt files (except All.txt) to All.txt:

```
copy all.txt + *.txt
```

Using this command does not produce an error message.

Ctty

Internal

Purpose:

Lets you change the device from which you issue commands.

Syntax:

ctty device

where:

device specifies the device from which you are giving commands to MS-DOS.

Comments:

Ctty is useful if you want to change the device on which you are working. In this command, the letters **tty** represent your terminal, that is, your computer's screen and keyboard.

Notes:

Many programs do not use MS-DOS for input, output, or either. Instead, they send input directly to the hardware on your computer. The **ctty** command has no effect on these programs — it affects only programs that use MS-DOS.

Examples:

The following command moves all command I/O (input/output) from the current device (the console) to an AUX port, such as another terminal:

ctty aux

The next command moves I/O back to the console screen and keyboard:

ctty con

Date

Internal

Purpose:

Enters or changes the date known to the system.

Syntax:

date [*mm-dd-yy*]

Comments:

You can change the date either from your terminal or from a batch file. (MS-DOS does not automatically display a prompt for the date if you use an Autoexec.bat file, so you might want to include a **date** command in that file.) MS-DOS records this date in the directory when you create or change a file.

Remember to use only numbers when you type the date. Valid numbers are:

mm = 1-12

dd = 1-31

yy = 80-99 or 1980-1999

Separate the date, month, and year entries with hyphens (-) or slashes (/). MS-DOS is programmed to change months and years correctly, whether the month has 28, 29, 30, or 31 days.

You can change the *mm-dd-yy* format in which the date is displayed and entered. The **country** command in the Config.sys file enables you to change the date format to the European standard, *dd-mm-yy*. For more information on the Config.sys file, see Appendix B, "How to Configure Your System."

Notes:

- This command sets your computer's internal clock, if one exists.
- The *mm-dd-yy* format might vary if you are using a code page other than the one for the United States. For more information about international date formats, see Appendix E, "How to Use Code Pages."

Examples:

If you simply type the **date** command, MS-DOS displays the following message:

```
Current date is weekday mm-dd-yy  
Enter new date (mm-dd-yy): _
```

where:

weekday is the day of the week (for example, Tuesday).

If you do not want to change the date, press **ENTER**. Or, you can type a particular date after the **date** command, as in the following example:

```
date 3-9-88
```

In this case, the Enter new date: prompt does not appear after you press **ENTER**.

DC

External

Purpose:

Compresses files so they use less disk space and require less transfer time. Also returns compressed files to their normal state.

Syntax:

`dc [/c]/d]/q][pathname]`

where:

pathname is the file or group of files (specified by wild cards) to compress or decompress. Compressing or decompressing a file does not change its name.

Comments:

Dc accepts the following switches:

Switch	Purpose
/c	causes dc to compress a file. The dc utility displays an error message if you specify a file that is already compressed.
/d	causes dc to decompress a file. The dc utility displays an error message if you specify a file for decompression that is not compressed.
/q	displays a message indicating whether or not the specified file is compressed. No compressing or decompressing takes place.

Notes:

- Use **dc** to compress a file to save disk storage space, to reduce the time it takes to transfer or copy the file, or for archival purposes.
- **dc** uses a modified LEMPEL-ZIV algorithm that replaces certain byte patterns with codes. As it reads a file, **dc** creates a table of byte patterns and the codes representing each pattern. For decompression, the program reads the table and converts the code to its original state, and then removes the table.
- The degree of compression you can achieve depends on the length of the file and on the type of data in the file. **dc** can compress text and data files by 40 to 60 percent. It compresses binary data files, such as .exe and .com files, less than 20 percent.
- If you do not specify a switch, **dc** compresses standard files and decompresses files that are compressed.
- If the contents of the file you specify are such that **dc** cannot compress them, it does not change the file.

Examples:

Suppose you had a file called Myfile, located in the current directory. The following command compresses it:

```
dc /c myfile
```

The following command decompresses a file named Myfile, located in the Relate directory of Drive A.

```
dc /d a:\relate\myfile
```

This command compresses all files having an extension of .txt that reside in the current directory.

```
dc /c *.txt
```

Disktype

External

Purpose:

Displays information on the size and capacity of the indicated disk.

Syntax:

`disktype [drive]`

where:

drive is the disk drive containing the disk for which you want to determine the type.

Comments:

The disk you specify can be either a diskette or a hard disk. If you omit *drive*, **disktype** displays information about the current disk.

When you use **disktype** with diskettes, the program displays the number of sides, tracks, and sectors-per-track of the diskette. The program also displays the syntax of the **format** command you would use to format the diskette.

When you use **disktype** with a hard disk, the program displays the number of heads, cylinders, and sectors-per-track.

Examples:

The following command determines the type of diskette in Drive A:

`disktype a:`

The screen then shows a display similar to this:

```
Diskette is formatted with 2 sides, 40 tracks, 9 sectors/track
--- FORMAT A:/4
--- FORMAT B:
```

The following command checks the type of a Drive C hard disk:

`disktype c:`

The screen then shows a display similar to this:

```
Fixed disk contains 4 heads, 200 cylinders, 17 sectors/track
```

Del (Erase)

Internal

Purpose:

Deletes the specified files.

Syntax:

`del [drive:] pathname`

or

`erase [drive:] pathname`

Comments:

The **del** command lets you use the * and ? wildcards to delete more than one file at a time. While convenient, this method of deleting files can be dangerous, so use wildcards cautiously.

If you type `del *.*`, this tells MS-DOS that you want to delete all the files in the working directory. MS-DOS displays the prompt, Are you sure? If you type Y (for Yes) in response, MS-DOS deletes all files in the working directory.

To delete all the files in another directory, type the **del** command followed by the directory name.

Warning: Once you delete a file from your disk, you cannot recover it.

Examples:

The following command deletes a file named Vacation:

`del vacation`

If you have two files named Vacation.feb and Vacation.apr, you can delete them both with the following command:

`del vacation.*`

Dir

Internal

Purpose:

Lists the files in a directory.

Syntax:

`dir [drive:][pathname][p][w]`

Comments:

The **dir** command, typed by itself, lists all directory entries in the working directory on the default drive. If you include a drive name, such as `b:`, with the **dir** command, all entries in the default directory of the disk in the specified drive are listed.

The **dir** command accepts the following switches:

Switch	Purpose
<code>/p</code>	Selects page mode, causing the directory display to pause once the screen is full. To resume scrolling the display, press any key.
<code>/w</code>	Selects wide display and causes MS-DOS to display only filenames and not other file information. The wide display lists up to five files per line.

Dir lists each file with its size in bytes and the time and date of last modification.

Note that the following **dir** commands are equivalent, since you can use the wildcards `?` and `*` in the *pathname* option:

This command	Is equivalent to
<code>dir</code>	<code>dir *.*</code>
<code>dir filename</code>	<code>dir filename.*</code>
<code>dir .ext</code>	<code>dir *.ext</code>

Notes:

If you set the **country** command in the Config.sys file to a country other than the United States, the directory date and time formats might differ. For more information on the Config.sys file, see Appendix B, "How to Configure Your System."

Examples:

If your directory contains more files than you can see on the screen at one time, type the following:

```
dir /p
```

This command displays the directory one screen at a time. As one screen fills up, you can press any key to see the next screen of the directory listing.

Diskcomp



Purpose:

Compares the contents of the disk in the source drive to the contents of the disk in the target drive.

Syntax:

```
diskcomp [drive1:] [drive2:] [/1] [/8]
```

where:

drive1: is the source drive.

drive2: is the target drive.

Comments:

Diskcomp performs a track-by-track comparison of the disks. It automatically determines the number of sides and sectors per track based on the format of the source disk.

The **diskcomp** command accepts the following switches:

Switch	Purpose
/1	Compares only the first side of the disk, even if the disks and drives you are using are double-sided.
/8	Compares only the first eight sectors per track, even if the disks contain nine or 15 sectors per track.

If you specify only one drive, **diskcomp** uses the default drive as the target drive. If you specify the same drive as the source and target, **diskcomp** does a comparison using one drive, prompting you to insert the disks as appropriate. If all the tracks are the same, **diskcomp** displays the message:

Compare OK

If the tracks are not the same, **diskcomp** displays a compare error message that includes the track number and side number (0 or 1) where it found the mismatch.

If the target disk is not the same type as the source disk, **diskcomp** displays:

Drive types or diskette types not compatible

When **diskcomp** completes the comparison, it prompts you with:

Compare another diskette (Y/N)?_

If you type Y (for Yes), **diskcomp** asks you to insert the proper disks. After you do so, it performs the next comparison. If you type N (for No), **diskcomp** ends. If the disk in the default drive doesn't contain MS-DOS and you end **diskcomp**, you see this message:

Insert disk with COMMAND.COM in drive A
and strike any key when ready

Diskcomp does not work on network drives, and you cannot use it with assigned, joined, or substituted drives. If you attempt to use the **diskcomp** command with these types of drives, the screen displays an error message.

Notes:

- When comparing a disk with a backup disk that you made using the **copy** command, you might see the Compare error message, even if the files on the disks are identical. This is because the **copy** command duplicates the information but doesn't necessarily place it in the same location on the target disk. In this case, use the **fc** command to compare individual files on the disk. For more information, see the explanation of the **fc** command later in this chapter.
- **Diskcomp** does not work on network drives, and you cannot use it with assigned, joined, or substituted drives. If you try to use the **diskcomp** command with these types of drives, you see an error message. The **diskcomp** command returns the following exit codes:

Code	Function
0	Compared OK. The disks compared exactly.
1	Did not compare. The disks were not the same.
2	CTRL-C error. The user terminated with CTRL-C.
3	Hard error. An unrecoverable read or write error occurred-did not compare.
4	Initialization error. There is not enough memory—invalid drives or command line syntax.

- You can use the batch processing **if** command to perform error processing based on the error level returned by **diskcomp**.

Examples:

If your computer has only one diskette drive, Drive A, and you want to compare two disks, you can simply type:

diskcomp a:

MS-DOS then prompts you to insert each disk as required.

Diskcopy



Purpose:

Copies the contents of the diskette in the source drive to a formatted or unformatted diskette in the target drive.

Syntax:

`diskcopy [drive1:] [drive2:] [/1]`

where:

drive1 : is the source drive.

drive2 : is the target drive.

Comments:

drive1 : and *drive2* : can be the same. If you omit the drive options, MS-DOS prompts you for the drives. If the target disk is not formatted, **diskcopy** formats it with the same number of sides and sectors per track as the source disk.

The /1 switch allows you to copy only one side of a disk.

Warning: The **diskcopy** command works only with diskettes. You cannot use **diskcopy** with a hard disk.

If you omit both options, MS-DOS performs a single-drive copy operation on the default drive. If you omit only the second option, MS-DOS uses the default drive as the target drive. In either case, though, **diskcopy** destroys the contents of the target disk.

Diskcopy prompts you to insert the source and target disks at appropriate times and waits for you to press any key before continuing.

After copying, **diskcopy** then prompts you with the following message:

Copy another diskette (Y/N)? _

If you type Y (for Yes), MS-DOS prompts you to insert source and target disks, and performs the next copy on the drives that you originally specified. To end the **diskcopy** process, type N (for No).

Because disk space is not allocated sequentially, disks that have had several files created and deleted on them become fragmented. As a result, the first free sector found by **diskcopy** becomes the next sector allocated, regardless of its location on the disk.

A fragmented disk can delay finding, reading, or writing a file. To prevent further fragmentation, use either the **copy** command or the **xcopy** command to copy your disk, instead of **diskcopy**. Because the **copy** and **xcopy** commands copy files sequentially to a disk, the new disk will not be fragmented.

The following command, for example, copies all files from the disk in Drive A to the disk in Drive B:

```
xcopy a:.* b:
```

Diskcopy determines the number of sides to copy, based on the source drive and disk.

Notes:

- The **diskcopy** command returns the following exit codes:
 - 0 Copied successfully
 - 1 Non-fatal read/write error. An unrecoverable but non-fatal read or write error occurred.
 - 2 CTRL-C error. The user entered CTRL-C to terminate **diskcopy**.
 - 3 Fatal hard error. **Diskcopy** was unable to read the source disk or format the target disk.
 - 4 Initialization error. There is not enough memory—invalid drives or command line syntax.
- You can use the batch processing **if** command to perform error processing based on the error level returned by **diskcopy**.

Examples:

To copy the contents of the disk in Drive A to the disk in Drive B, use the following command:

diskcopy a: b:

Diskcopy prompts you to insert both disks and press any key to begin copying.

Diskopt

External

Purpose:

Optimizes file storage on the specified disk.

Syntax:

diskopt

Comments:

Diskopt rearranges a disk's file sectors into a contiguous (consecutive) format.

Always make a backup of a disk before optimizing it. If **diskopt** is interrupted during its operation, the entire contents of the disk might be lost.

Caution: Do not use diskopt on any disk that contains copy protected programs.

- To use **diskopt**, type the following command, then press **ENTER**:
diskopt

A prompt asks you:

Enter drive to optimize?

Respond by pressing the letter of the drive that contains the disk to optimize.

Diskopt analyzes each file on the indicated diskette or hard disk to determine whether the file's sectors are consecutive. If they are not, it rearranges the sectors to make them consecutive.

When files on a disk are fragmented, reading or writing to that file requires extra drive head movement and reduces the speed at which the drive can access data. Diskopt reduces disk access time to a minimum.

Notes:

- Do not interrupt **diskopt** by resetting or turning off your computer. Doing so might make the entire contents of the disk useless.
- The disk must contain at least one sector of free space before you can use **diskopt** successfully.

Examples:

This command executes **diskopt**. A prompt asks you for the name of the drive that contains the disk to optimize.

diskopt

Exit

Purpose:

Exits the Command.com program (the command processor) and returns to a previous level if one exists.

Syntax:

`exit`

Comments:

If you use the MS-DOS command program to start a new command processor, you can use the **exit** command to return to the old command processor. While running an application program, you can also exit to the MS-DOS command processor and then return to your program. For more information about command processors, see the explanation of **command**, earlier in this chapter.

Examples:

If you start a new command processor by typing the following command:

`command c:\`

you can then return to the previous command processor by typing:

`exit`

Fastopen

External

Purpose:

Decreases the amount of time needed to open frequently-used files and directories.

Syntax:

```
fastopen [drive:[ = nnn]][...]
```

where:

nnn is the number of files per disk.

Comments:

Access to files in a complex directory structure is time consuming. If you run application programs that use several files (such as a data base application), the time to open and close files slows your computer's performance. **Fastopen** tracks the location of files and directories on a disk for fast access.

Each time you open a file or directory, **fastopen** records its name and location. Then, if you reopen a file or directory recorded by **fastopen**, the access time is greatly reduced.

Fastopen works only on hard disks and will not work over a network. You can use **fastopen** with as many as four hard disks at one time. For each hard disk, **fastopen** will track *nnn* files or directories, where *nnn* ranges from 10 to 999. The default is 10.

Notes:

- You can invoke the **fastopen** command only once. To change **fastopen** settings, restart MS-DOS.
- **Fastopen** needs approximately 40 bytes of memory for each file or directory location it tracks.

Examples:

If you wanted MS-DOS to track the location of up to 100 files on Drive C, you could type:

```
fastopen c: = 100
```


Fdisk



Purpose:

Configures a hard disk for use with MS-DOS.

Syntax:

`fdisk`

Comments:

The **fdisk** command displays a series of menus to help you partition your hard disk for MS-DOS. With the **fdisk** command, you can:

- Create a primary MS-DOS partition
- Create an extended MS-DOS partition
- Change the active partition
- Delete an MS-DOS partition
- Display partition data
- Select the next fixed disk drive for partitioning on a system with multiple fixed disks

Notes:

- **Fdisk** doesn't work on drives used in the **subst** or **join** commands.
- For more information on using **fdisk**, see Appendix D, "Using Your Hard Disk (Fdisk)."

Find

External

Purpose:

Searches for a specific string of text in a file or files.

Syntax:

`find [/v] [/c] [/n] "string"[[drive:][pathname] ...]`

where:

string is a group of characters you want to find.

Comments:

The **find** command looks for *string* in one or more files. After searching the specified files, **find** displays any lines it found that contain the specified string.

String must be enclosed in quotation marks. Uppercase characters in *string* will not match lowercase characters.

If *string* contains quotation marks, you must enclose it in an additional pair ("*string*").

If you omit *pathname*, **find** acts as a filter. It takes input from the MS-DOS standard input (usually from the keyboard, a pipe, or a redirected file) and displays any lines that contain *string*. Wildcards (* and ?) are not allowed in filenames or extensions.

The **find** command accepts the following switches:

Switch	Purpose
/v	Displays all lines not containing the specified string.
/c	Displays only the number of lines that contain a match in each of the files.
/n	Precedes each line with its relative line number in the file.

If you specify the /c switch with the /v switch, **find** displays the number of lines that do not contain the string you typed. If you specify the /c switch with the /n switch, **find** ignores the /n switch.

Examples:

The following command displays all lines from the file `Pencil.ad` that contain the string `"Pencil Sharpener"`:

```
find "Pencil Sharpener" pencil.ad
```

The next command causes MS-DOS to display the names of all files on the disk in Drive B that do not contain the string `"date:"`:

```
dir b: | find /v "date"
```

To find the string, `"The dentist said, "Open wide!"`, in the file `Story.doc`, type:

```
find 'The dentist said, "'Open wide!'"' story.doc
```

Note that in this last example, the entire phrase you were searching for was set off by single quotation marks rather than double quotation marks. When you use unique marks to set off the phrase, **find** does not mistake the quotation marks for part of the phrase itself.

Fmat2000

External

Purpose:

Formats a standard 5 1/4-inch, double-sided diskette for 720 kilobytes of storage in a high-capacity diskette drive.

Syntax:

`fmat2000 drive [/s][/v]`

Comments:

For reliability with **fmat2000**, use 80-track, double-sided diskettes (Cat. No. 26-410).

Fmat2000 accepts the following switches:

Switch	Purpose
/s	Causes fmat2000 to copy the MS-DOS system files to the diskette as part of the format process. This makes the newly formatted diskette a system disk.
/v	Causes fmat2000 to prompt for a volume label after it formats the specified diskette. If you use this parameter, you can enter a label of a maximum of 11 characters, or you can press ENTER to bypass the prompt. Entering a volume label helps you to keep track of your diskettes.

Notes:

- If you have a high capacity drive, use **fmat2000** (instead of **format**) to double the amount of data you can store on your standard diskettes.
- You can use **fmat2000** to format either a new diskette for data storage, or to *clear* an old diskette for data storage. Formatting any disk erases all information on the disk. So, be sure any old diskettes that you want to format do not contain information you need.
- MS-DOS prompts you for the diskette you want to format. For example, if you specify Drive A, **fmat2000** asks you to:
Insert a new diskette for drive A
and strike ENTER when ready.

Remove the diskette currently in Drive A, and insert the diskette to be formatted. Press **ENTER**.

Diskettes formatted with **fmat2000** are compatible with diskettes used with the Tandy 2000 computer. You can read and write to such diskettes, using a Tandy 2000. You cannot, however, use them to boot the Tandy 2000.

Examples:

The following command formats a standard diskette in a high-capacity Drive B for 720 kilobytes of storage, and prompts you for a label to assign the diskette:

```
fmat2000 b: /s /v
```

Format



Purpose:

Formats the disk in the specified drive to accept MS-DOS files.

Syntax:

`format drive: [/1]/[4]/[8]/[n:xx]/[t:yy]/[v]/[s]`

or

`format drive: [/1]/[b]/[n:xx]/[t:yy]`

Comments:

The **format** command creates the directory and the file allocation tables on a disk. You must use this command to format all new disks before MS-DOS can use them.

Warning: Formatting destroys any previously existing data on a disk and ignores drive assignments created using the **assign** command.

You must specify the drive that you want to use to format a disk. **Format** then uses the drive type to determine the default format for a disk.

The **format** command accepts the following switches:

Switch	Purpose
/1	Formats a single-side of the diskette.
/4	Formats a 5 1/4-inch, double-sided disk in a high-capacity disk drive. If you are using a single- or double-sided drive, you might not be able to reliably read disks formatted with this switch.
/8	Formats eight sectors per track.
/b	Formats the disk, leaving ample space to copy an operating system, such as MS-DOS 3.3.
/s	Copies the operating system files listed in the file <code>Formats.tbl</code> from the disk in the default drive to the newly formatted disk. The newly formatted disk must be 1.2 megabytes or greater in size; otherwise, format rejects the command. If the operating system is not on the default drive, format prompts you to insert a system disk in the default drive (or in Drive A if the default drive is non-removable).
/t:tracks	Specifies the number of tracks on the disk. This switch formats 3 1/2-inch diskette to the number of tracks specified. For 720K-byte disks and 1.44-megabyte disks, this value is 80 (/t:80).
/n:sectors	Specifies the number of sectors per track. This switch formats a 3 1/2-inch disk to the number of sectors specified. For 720K-byte disks, this value is 9 (/n:9).
/v	Prompts you for a volume label for the disk you are formatting. A volume label identifies the disk and can be up to 11 characters in length (no tabs allowed). An example of a volume label is programs.

When you format a hard disk, **format** prompts you to verify the volume label by displaying:

Enter current Volume Label for drive x:

If your hard disk does not have a volume label, press **ENTER** .

Note that if your hard disk has never been formatted before, or if it has a bad boot sector, **format** does not prompt you for a volume label.

If the volume label that you enter does not match the label on the hard disk, **format** displays:

Invalid Volume ID Format failure

Otherwise, it continues:

WARNING, ALL DATA ON NON-REMOVABLE DISK
DRIVE X: WILL BE LOST!
Proceed with Format (Y/N)?_

To format your hard disk, type Y (for Yes) and press **ENTER** .

To cancel **format**, type N (for No) and press **ENTER** .

When formatting is complete, **format** displays a message showing the total disk space, any space marked as defective, the total space used by the operating system (when you use the /s switch), and the space available for your files.

Notes:

- Do not use the **format** command with drives used in the **assign**, **join**, or **subst** commands.

You cannot format drives over a network.

- For more information about formatting your hard disk, see Appendix D, “Configuring Your Hard Disk (Fdisk).”

For more information about disk volume labels, see the explanations of the **dir**, **label**, and **volume** commands in this chapter.

The following table shows the switches you can use for certain types of disks:

Disk type	Valid Switches
160/180K bytes	/1 /4 /8 /b /n /t /v /s
320/360K bytes	/1 /4 /8 /b /n /t /v /s
720K bytes	/n /t /v /s
1.2 megabytes	/n /t /v /s
1.44 megabytes	/n /t /v /s
hard disk	/v /s

The **format** command returns the following exit codes:

Code	Function
0	Successful completion
3	Terminated by user (<input type="text" value="CTRL"/> <input type="text" value="C"/>)
4	Fatal error (any error other than 0, 3, or 5)
5	N response to hard disk prompt, Proceed with format (Y/N)?

You can check these exit codes by using the error level condition with the **if** batch processing command. You can use the **select** command instead of **format** to format a disk with country-specific information. For more information, see the explanation of the **select** command later in this chapter.

Examples:

To format a high-density, 3 1/2-inch diskette in a high-capacity, 3 1/2-inch drive, type:

```
format a:
```

To format a standard-density, 3 1/2-inch diskette in a high-capacity, 3 1/2-inch drive, type:

```
format a:/n:9 /t:80
```

To format a standard-density, 3 1/2-inch diskette in a standard-capacity, 3 1/2-inch drive, type:

```
format a:
```

To format a high-density, 5 1/4-inch diskette in a high-capacity 5 1/4-inch drive, type:

format a:

To format a standard-density, 5 1/4-inch diskette in a high-capacity 5 1/4-inch drive, type:

format a: /4

To format a standard-density, 5 1/4-inch diskette in a standard-capacity, 5 1/4-inch drive, type:

format a:

Graftabl

External

Purpose:

Enables you to display an extended character set when using display adapters in graphics mode.

Syntax:

`graftabl [xxx]`

or

`graftabl /status`

where:

`xxx` is a code page identification number.

Comments:

Valid code pages (`xxx`) include:

Value	Code Page
437	United States (default)
850	Multilingual
860	Portuguese
863	French-Canadian
865	Nordic

If you type the **graftabl** command followed by the `/status` switch, MS-DOS displays the active character set.

After **graftabl** loads the character table, it displays:

Graphics characters loaded

You can load the graphics table only once each time you start MS-DOS, therefore you might want to put the **graftabl** command in your `Autoexec.bat` file to save time. If you try to load the same table a second time, **graftabl** displays:

Graphics characters already loaded

Notes:

- The **graftabl** command increases the size of MS-DOS resident in memory.
- For more information about using code pages, see the discussion of the **chcp** command in this chapter.
- The **graftabl** command returns the following exit codes:

Code	Function
0	Command successful
1	Table already loaded
2	File error occurred
3	Incorrect parameter, no action taken
4	Incorrect version of MS-DOS; version 3.3 required

You can check these exit codes using the error level condition with the **if** batch processing command.

Examples:

To load a table of graphics characters into memory, type:
 graftabl

Graphics

External

Purpose:

Lets you print a graphics display screen on a printer when you are using a color or graphics monitor adapter.

Syntax:

`graphics [ptype] [/b][/r][/cr][/lf]`

where:

*p*type is one of the following:

Type	Function
cgp220	Prints on a Tandy CGP-220 color ink jet printer.
standard	Prints on any Tandy printer (other than the CGP-220 or the DMP-110) without dip switch settings for Tandy and PC printer compatibility.
pcmode	Prints on a Tandy printer with a dip switch set for PC compatibility. Also, use pmode for other PC-compatible printers.
tmode	Prints on a Tandy printer with a dip switch set for Tandy compatibility.
dmp110	Prints on the Tandy DMP-110 printer.

Comments:

If you do not specify the *p*type option, **graphics** displays the following menu:

Tandy Graphics Screen Dump Utility
Version 3.20.00
Copyright 1985 Tandy Corp.,
All rights reserved.

Enter type of printer

- [A] For TANDY CGP-220 printer
- [B] For TANDY DMP Standard Resolution
printer
- [C] For PC compatible printer
- [D] For PC compatible printer in TANDY
MODE
- [E] for TANDY DMP-110 printer

Press the letter that corresponds to your type of printer.

Comments:

The **graphics** command accepts the following switches:

Switch	Purpose
/b	Prints the background in color when you specify a CGP-220 printer. If you omit /b, the background is not printed.
/r	Prints black and white (as seen on the screen) on the printer. The default is to print black as white and white as black.
/cr	Causes graphics to execute an end-of-line character as a carriage return. Otherwise, a carriage return causes both a carriage return and a line feed.
/lf	Causes graphics to send only a line feed as the end-of-line character /Lf and /cr are mutually exclusive.

Notes:

- To initialize screen printing, press **[SHIFT] [PRTSC]** .
- Graphics images are printed in a maximum of eight shades of gray on a black-and-white printer, or in 16 color on a CGP-220 printer. On a black-and-white printer, the image is printed sideways when the video mode is 640 x 200 or 640 x 350.
- The /cr and /lf switches only affect printers with dip switch settings for Tandy and PC compatibility. If you have such a printer, and it does not perform a line feed when required, or if it prints an unwanted blank line between text, use the appropriate /cr or /lf switch to correct the problem.
- Reset your computer to turn **graphics** off. If **graphics** is off, pressing the **[SHIFT] [PRTSC]** keys prints a text screen. To terminate printing, press **[SHIFT] [PRTSC]** again.
- **Graphics** increases the size of MS-DOS resident in memory.
- You can invoke **graphics** any number of times with different parameters. If the printer type remains the same, only one copy of the program remains in memory.

Examples:

To print a graphics screen on your printer, type:

graphics

Then, when the screen displays the information you want to print, hold down **[SHIFT]** and press **[PRTSC]** .

Hsect

External

Purpose:

Initializes hard disk tracks and sectors.

Syntax:

hsect

Comments:

The **hsect** command works the same as the **FORMAT HARD DISK** utility on your Utilities diskette. After using **hsect**, use **fdisk** and **format** to complete the hard disk format process.

After you enter the **hsect** command, the following prompt appears:

Which hard drive do you
want to format (C/D)
?

Press **C** **ENTER** to format an internal hard disk drive. To format a second internal or an external hard disk drive, press **D** **ENTER**.

After you select the disk to format, the screen displays the following warning:

All data on drive x will be
DESTROYED!!
Do you want to continue (Y/N)
?

If you continue, **hsect** erases all data from the hard disk. Therefore, use it only when you are preparing to install an operating system on your hard disk for the first time.

Press **N** **ENTER** to exit the formatting procedure and return to the main menu, or press **Y** **ENTER** to continue.

If you continue, **hsect** displays information about the disk you want to format, like this:

```
Hard drive C is type x
Number of heads = x
Number of cylinders = x
Is this correct (Y/N)
?
```

If the display matches your hard disk drive, press **Y** **ENTER**.

If you make an error in **setup**, or if you have a *non standard* hard disk, the type information will not match the type of your hard disk. To select other type configuration, type **N** **ENTER**. Then, answer the prompt that follow with the correct number of heads and cylinders and the interleave factor for your hard disk drive.

Also, press **N** **ENTER** at the Is this correct (Y/N) prompt if you want to change the interleave factor from the default of 3.

After you verify or change the hard disk information, **hsect** prompts:

```
Do you want to flag defective
tracks (Y/N)
?
```

Press **N** **ENTER** if your hard disk's Media Error Map shows no defective tracks. The following prompt appears on the screen:

```
Enter next head, cylinder pair or
press < enter > to quit.
?
```

For example, if your Media Error Map list Head 4, Cylinder 100 and Head 5, Cylinder 100 as defective, type the following. After each command line, press **ENTER**:

```
4,100
5,100
```

Press **ENTER** to begin the formatting procedure when you are through typing the defective head and track information.

Notes:

- Do not interrupt the program while it is formatting the drive.
- When the format is complete, **hsect** displays: Format complete!.

Examples:

The following command loads **hsect** from the current drive. **Hsect** then prompts for the drive to format:

hsect

The next command loads **hsect** from the diskette in Drive B. After it loads, **hsect** prompts for the drive to format:

b:hsect

Join



Purpose:

Joins a disk drive to a specific path.

Syntax:

`join [drive: drive: path]`

or

`join drive: /d`

Comments:

With the **join** command, you don't need to name physical drives with separate drive letters. Instead, you can refer to all the directories on a specific drive with one path. If the path already existed before you gave the **join** command, you cannot use it while the "join" is in effect. Note that you cannot join a drive if another process is using it.

If the path does not exist, MS-DOS tries to make a directory with that path.

After you give the **join** command, the first drive name becomes invalid, and if you try to use it, MS-DOS displays the Invalid drive error message.

If you type the **join** command by itself, MS-DOS displays the current drives that are joined.

Notes:

- The following commands do not work on drives used in the **join** command (or the **subst** command):

chkdsk	label
diskcopy	recover
fdisk	sys
format	

Examples:

You can join a drive only with a root level directory. For example, this command will work:

join d: c:\sales

But this command will not:

join d: c:\sales\regional

To reverse **join**, use the following format:

join *drive*: /d

Here, *drive*: represents the source drive, and the /d switch turns off the **join** command.

Keyb

External

Purpose:

Loads a keyboard program.

Syntax:

```
keyb [xx[, [yyy],[[drive:][path]filename]]]
```

where:

xx is a two-letter country code.

yyy is the code page that defines the character set.

filename is the name of the keyboard definition file.

Comments:

xx is one of the following two-letter codes:

Code	Keyboard type	Command
us	United States	keyb us (default)
fr	France	keyb fr
gr	Germany	keyb gr
it	Italy	keyb it
sp	Spain	keyb sp
uk	United Kingdom	keyb uk
po	Portugal	keyb po
sg	Swiss-German	keyb sg
sf	Swiss-French	keyb sf
dk	Denmark	keyb dk
be	Belgium	keyb be
nl	Netherlands	keyb nl
no	Norway	keyb no
la	Latin America	keyb la
sv	Sweden	keyb sv
su	Finland	keyb su
cf	Canada (French)	keyb cf

If you type **keyb** without options, MS-DOS displays a message similar to the following to show the current keyboard code, its related code page, and the current code page used by your console screen device (CON):

Current keyboard code: FR Code page: 437

Current CON code page: 437

To switch from the **keyb** program to the default board format (United States), press **CTRL** **ALT** **F2** .

Press **CTRL** **ALT** **F2** to again return to the memory-resident keyboard program.

The **keyb** command lets you use characters that are not part of the normal (QWERTY) keyboard format. Using the **keyb** command with one of the previous two-letter codes, you can type commands or text to MS-DOS using either the standard keyboard or a special keyboard.

Note that the characters that appear on your screen when you type on a standard keyboard do not necessarily match the labels on the keys. You can produce some characters in the non-United States keyboard sets by pressing **CTRL** **ALT** along with an appropriate character key. To produce accented (and unlauded) characters, you press *dead keys*. Dead keys are keys that do not display a character when used alone, but when followed by a letter, display that letter with the appropriate accent.

Notes:

You can also include the appropriate **keyb** command in your Autoexec.bat file so that you won't have to type it each time you start MS-DOS.

Examples:

To use a German keyboard, type the following command:

keyb gr

Label



Purpose:

Creates, changes, or deletes the volume label on a disk.

Syntax:

`label [drive:][label]`

where:

label is the new volume label, up to 11 characters.

Comments:

A volume label is a name you can specify for a disk. MS-DOS displays the volume label of a disk as a part of its directory listing to show you which disk you are using.

If a volume serial number exists, **label** also displays this eight-character number:

Volume Serial Number in drive *x* is *nnnn-nnnn*

If you do not specify a label, **label** prompts you with:

Volume in drive *x* is *xxxxxxxx*

Type a volume label of up to 11 characters or press **ENTER** if you do not want to update the volume label. A volume label can be up to 11 characters in length and can include spaces but not tabs. Type the volume label that you want, and press **ENTER**

Press **ENTER** without typing a label to delete the volume label. **Label** prompts you with the message:

Delete current volume label (Y/N)?_

If you type Y (for Yes), **label** deletes the volume label on the disk. Otherwise, the volume label stays the same.

Notes:

- You can use the MS-DOS **dir** or **vol** command to determine whether the disk already has a volume label.

- Label doesn't work on drives involved with **subst** or **join** commands.
- Do not use any of the following characters in a volume label:

* ? / \ | . , ; : + = [] () & ^

Examples:

To label a disk in Drive A that contains sales information for 1987, you might type:

label a:sales1987

Lf

External

Purpose:

Suppresses line feeds after carriage returns in output to printer 1, 2, or 3.

Syntax:

lf off

or

lf on

Comments:

Some printers automatically generate a line feed after they receive a carriage return. With some programs, this causes an extra line to print between each line of characters. **Lf** suppresses the line feed sent to the printer by the software.

Notes:

- **Lf** suppress only those line feed codes that immediately follow a carriage return.
- **Lf** supports the **mode lloff**, **mode lfon**, **mode dmp**, and **mode dwp** commands. **Lpdrv.sys** also supports these **mode** commands.

Examples:

To turn off line feeds that follow the carriage returns sent to your printer, type:

lf off

Mkdir

Internal

Synonym:

md

Purpose:

Makes a new directory.

Syntax:

`mkdir [drive:]path`

Comments:

The **mkdir** command lets you create a multilevel directory structure. Remember, however, that directories created with **mkdir** are always sub-directories of your working directory unless you explicitly specify a different path with the **mkdir** command.

Notes:

You cannot specify a drive before this command. For example, you could not type `a:mkdir newdir` because MS-DOS always assumes that the **mkdir** command is for the current drive.

Examples:

To create a directory that holds all your tax information, you could type the following command from your root directory:

```
mkdir \taxes
```

Now, suppose you want to create a directory named **Rental** under the `\taxes` directory to keep track of information about a duplex that you rent out. To do this from the root directory, type:

```
mkdir \taxes\rental
```

To create the same subdirectory from the `\taxes` directory, you could type either the previous command, or:

```
mkdir rental
```

Mode

External

Purpose:

Sets operation modes for devices.

Syntax:

Parallel printer mode:

mode LPTn[:][*chars*][,*lines*][,*p*]

and

mode lloff

or

mode lfon

Asynchronous communications mode:

mode Comm[:]*baud*[,*parity*[,*databits* [,*stopbits*[,*p*]]]

or

mode Comm:1200/75

Redirecting parallel printer output:

mode LPTn[:] = Comm[:]

Display modes:

mode *display*

or

mode [*display*],*shift*[,*t*]

Device code page modes:

mode device codepage prepare = ((*cpages*)[*drive*:][*path*]*filename*)

and

mode device codepage select = *yyy*

mode device codepage refresh

mode device codepage [/status]

CPU speed modes:

mode *speed*

Comments:

The **mode** command prepares MS-DOS for communication with devices such as parallel and serial printers, modems, and console screens. It prepares parallel printers and console screen devices for code page switching, lets you redirect output, and changes the speed at which your computer operates.

Parallel Printer Modes

For parallel printer modes, you can use PRN and LPT1 interchangeably. You can use the following options with the **mode** command to set parameters for a parallel printer:

Option	Purpose
<i>n</i>	Specifies the printer number: 1, 2, or 3.
<i>chars</i>	Specifies characters per line: 80 or 132.
<i>lines</i>	Specifies vertical spacing (lines per inch): 6 or 8.
<i>p</i>	Specifies that mode tries continuously to send output to the printer if a time-out error occurs. This option causes part of the mode program to remain resident in memory.

Default settings are LPT1, 80 characters per line, and 6 lines per inch. You can break out of a time-out loop by pressing **CTRL** **BREAK**.

You use **lfoff** or **lfon** to either enable or disable line feeds (ASCII character code 10) following carriage returns (ASCII character code 13).

Option	Purpose
lfoff	Sends an escape code sequence to the Lf.com terminate-and-stay-resident program to suppress line feeds after carriage returns.
lfon	Sends an escape code sequence to the Lf.com terminate-and-stay-resident program to cancel suppression of line feeds after carriage returns.

If your printer inserts unwanted spaces between lines, use **lfoff**. If your printer prints lines on top of each other, use **lfon**.

Asynchronous Communication Modes

Use the following options with the **mode** command to set parameters for serial ports:

Option	Purpose
<i>m</i>	Specifies the asynchronous communications (COM) port number: 1, 2, 3, or 4.
<i>baud</i>	Specifies the first two digits of the transmission rate: 110, 150, 300, 600, 1200, 2400, 4800, 9600 or 19,200.
<i>parity</i>	Specifies the parity: N (none), O (odd), or E (even). The default value is E.
<i>databits</i>	Specifies the number of data bits: 7 or 8. The default value is 7.
<i>stopbits</i>	Specifies the number of stop bits: 1 or 2. If baud is 110, the default value is 2; otherwise, the default value is 1.
<i>p</i>	Specifies that mode is using the COM port for a serial printer and continuously retrying if time-out errors occur. This option causes part of the mode program to remain resident in memory.

The default settings are COM1, even parity, and 7 databits.

Some international operations require your computer's RS232 interface to operate at different baud rates for input and output. To set your computer for such international operations, use the **1200/75** option (input = 1200 baud, output = 75 baud) as follows:

```
mode comx:1200/75
```

where *x* is the number of the serial port.

Display Modes

Use the following options with the **mode** command to set parameters for a display:

Option	Purpose
<i>display</i>	Specifies one of the following values: 40, 80, BW40, BW80, CO40, CO80, or MONO. The values 40 and 80 indicate the number of characters per line. BW and CO refer to a color graphics monitor adapter with color disabled (BW) or enabled (CO). MONO specifies a monochrome display adapter with a constant display width of 80 characters per line.
<i>shift</i>	Specifies whether to shift the display to the left or to the right. Valid values are L (for left) or r (for right).
<i>t</i>	Tells MS-DOS to display a test pattern to aid in aligning the display on the screen.

Device Code Page Modes

You can use the **mode** command to set or display code pages for parallel printers or your console screen device. Use the following options with **mode** to set or display code pages:

Option	Purpose
<i>device</i>	Specifies the device to support code page switching. Valid device names are con, prn, lpt1, lpt2, and lpt3.
<i>yyy</i>	Specifies a code page. Valid code pages are 437, 850, 860, 863, and 865.
<i>filename</i>	Identifies the name of the code page information (.cpi) file MS-DOS should use to prepare a code page for the specified device.
<i>cpages</i>	Specifies a list of code pages. Valid code pages are 437, 850, 860, 863, and 865

You can use four keywords with the **mode device codepage** command. Each causes the **mode** command to perform a different function. The following table explains each keyword:

Keyword	Function
prepare	Tells MS-DOS to prepare code pages for a given device. You must prepare a code page for a device before you can use it with that device.
select	Specifies which code page you want to use with a device. You must prepare a code page before you can select it.
refresh	If the prepared code pages for a device are lost due to hardware or other error, this keyword reinstates the prepared code pages.
/status	Displays the current code pages prepared and/or selected for a device. Note that the following commands both produce the same results: mode con codepage mode con codepage /status

Typing **/status** is optional.

For more information about using the **mode** command to set or display code pages, see Appendix E, "How to Use Code Pages."

CPU Speed Modes

You can use three **mode** commands to control the speed at which your computer operates. This control is useful for some programs that do not operate properly at your computer's fast (default) speed. The speed selections are:

Option	Result
slow	Reduces your computer's operating speed.
fast	Sets your computer to run at its fastest (default) speed. This option is only useful if you have previously used the slow or auto options.
auto	Sets the Tandy 1000, 3000, and 4000 to automatically change speeds as required by the software you are running.

Notes:

- You can use the following abbreviations with the **mode** command for code page modes:

Type	In Place Of
cp	codepage
/sta	/status
prep	prepare
sel	select
ref	refresh

- If you are using the **mode** command over a network, do not use the p switch for continuous retry.

Examples:

Suppose you want your computer to send its printer output to a serial printer. To do this, you need to use the **mode** command twice. The first **mode** command specifies the asynchronous communication modes, and the second **mode** command redirects the computer's parallel printer output to the asynchronous communication port specified in the first **mode** command.

For example, if your serial printer operates at 4800 baud with even parity, and if it is connected to the COM1 port (the first serial connection on your computer), you would type:

```
mode com1:48,e,,,p
```

```
mode lpt1: = com1:
```

If you have redirected parallel printer output from LPT1 to COM1 and then decide that you want to print a file using LPT1, type:

```
mode lpt1:
```

This command disables any redirection of LPT1.

Suppose you want your computer to print on a parallel printer that is connected to your computer's second parallel printer port (LPT2). To print with 80 characters per line and 8 characters per inch, you would type:

```
mode lpt2: 80,8
```

or

```
mode lpt2:.,8
```

If you want your computer to keep trying to print a file until your printer is ready to print it, type:

```
mode lpt2:80,8,p
```

To stop retrying, press **CTRL** **BREAK** or type the **mode** command without the **p** option.

To turn off carriage returns after line feeds, type:

```
mode lffoff
```

Mon386

External

Purpose:

Lets you load up to nine programs at the same time on the Tandy 4000.

Syntax:

`mon386`

Comments:

The number of programs you can run depends on your computer's memory. For example, you can have BASIC, a word processor, and a spreadsheet loaded simultaneously and switch from one program to another.

Because **mon386** allocates *extended random-access memory (RAM)* for use by applications, standard MS-DOS programs can utilize memory beyond the traditional 640K boundary.

To install mon386:

1. Start your system as outlined in the *Tandy 3000/4000 MS-DOS Handbook*.
2. At the system prompt, type:

`mon386`

The **mon386** menu appears in the upper right corner of the display with the following options and a list of current tasks and their assigned numbers.

= Task to run

A = Add a task

R = Remove a task

To add a task:

1. Type **a** at the Monitor menu. A prompt appears asking for the task size in K.
2. Enter the amount of memory your program requires. Valid sizes are 128, 256, 384, and 512. Other sizes automatically adjust to one of these values.
3. Press **[ENTER]**. The system prompt appears.
4. Load the program you want to add in the usual manner.

To remove a task:

Note: Before you remove a task, be sure to save its data to disk. Once you remove a task, you cannot recover it, and you lose any data the task created that you did not save.

1. To display the Monitor menu, press **[ALT] [=]**.
2. Type **r** at the Monitor menu. A prompt appears asking for a task number.
3. Type the number of the task you want to remove. **Mon386** then displays the task and asks for verification with a Y/N prompt.

To run a task:

1. Press **[ALT] [=]** to display the Monitor menu.
2. Enter the number of the task you want to run. The task's screen appears. While a task is running, its task number is displayed in the upper right corner of the screen.

To switch tasks:

1. Press **[ALT] [=]** to display the Monitor menu.
2. Enter the number of the task you want to run. In some instances, the monitor program cannot immediately leave a task (for example, during a **type** instruction). **Mon386** switches tasks as soon as possible.

Notes:

- Do not use **mon386** while the **cache** utility is installed.
- **Mon386** cannot run other multi-processing programs.
- **Mon386** controls memory above 1 megabyte (M). It cannot work with programs that also use this memory (such as **Temmm.sys**, **Vdisk.sys** and **Spooler.sys** when configured to access expanded memory).
- Two tasks accessing the same disk file can corrupt or destroy the data in that file.

More

External

Purpose:

Sends output to the console one screen at a time.

Syntax:

`more source`

or

`source | more`

where:

source is a file or command.

Comments:

More is a filter that reads from standard input (from a pipe or redirected file) and displays one screen of information at a time. **More** is commonly used to view long files.

For example, you can use the **dir** command, the **sort** command, or a filename as a source. When output from the commands or a file fills the screen, **more** stops the scrolling and displays --More-- at the bottom of the screen.

Press **ENTER** to display another screen of information. Keep pressing it when appropriate until you have read all the data.

Notes:

- To hold input information until it is displayed, the **more** command creates a temporary file on the disk. If the disk is full or write-protected, **more** will not work.
- For more information about using redirection symbols with commands, see Chapter 2, "About Commands."

Examples:

Suppose you have a long file called `Clients.new` that you want to view on your screen. The following command redirects the file through the **more** command to show the file's contents one screen at a time:

```
more clients.new
```

If you have a long file of customers, you could use the **more** command to view it one screen at a time. Suppose this file is called `Clients.new`. To see it, you would type:

```
type clients.new | more
```

Nlsfunc

External

Purpose:

Loads country-specific information.

Syntax:

```
nlsfunc [[drive:][path]filename]
```

where:

filename specifies the file containing country-specific information.

Comments:

Nlsfunc supports the use of extended country-specific information and code page switching.

The default value of *filename* is defined by the country command in your Config.sys file. If no country command exists in your Config.sys file, MS-DOS uses the Country.sys file in your root directory for country-specific information.

Examples:

Suppose you have a file on your disk called Newcdpg.sys that contains country-specific information. To use the information from that file rather than from the Country.sys file, you would type:

```
nlsfunc newcdpg.sys
```

To use the default country-specific information found in the Country.sys file, simply type:

```
nlsfunc
```

Path

Internal

Purpose:

Sets a command search path.

Syntax:

`path [drive:][path][:[drive:][path]...]`

or

`path;`

Comments:

The **path** command lets you tell MS-DOS where to search for external commands after it searches the working directory. The default is no path.

For instance, you can tell MS-DOS to search the `\user\pete` directory for external commands, by typing the **path** command followed by the directory name `\user\pete`. Then, until you exit MS-DOS or set another path, MS-DOS searches the `\user\pete` directory for external commands.

You can tell MS-DOS to search more than one path by specifying several paths, separating them with semicolons. If you use the **path** command without options, it displays the current path. If you use the following command, MS-DOS searches only the working directory for external commands:

`path ;`

This path is also called the NUL path.

Examples:

The following command tells MS-DOS to search three directories to find external commands. (The three paths for these directories are `\user\pete`, `b:\user\emily`, and `\bin`.)

`path \user\pete;b:\user\emily;\bin`

MS-DOS searches the paths in the order specified in the **path** command.

Print

External

Purpose:

Prints a text file on a lineprinter while you are processing other MS-DOS commands (usually called background printing).

Syntax:

```
print [/d:device][/b:size][/u:value1][/m:value2] [/s:timeslice]
[/q:qsize] [/t][/c][/p] [drive:][pathname]
```

Comments:

You can use the **print** command only if you have an output device, such as a printer or a plotter, connected to one of your computer's serial or parallel ports.

The **print** command accepts the following switches:

Switch	Purpose
/d:device	Specifies the print device name. The default device is LPT1. Other possible print device names for parallel ports are PRN, LPT2, and LPT3. COMx, where x is a number 1-4, refers to a serial port. (LPT1 and PRN both refer to the first parallel port on your computer.)
/b:size	Sets the size in bytes of the internal buffer. To speed up the print command, increase the value of <i>size</i> . The minimum value of <i>size</i> is 512, and the maximum value is 16,386.
/u:value1	Specifies the number of clock ticks print will wait for a printer. If the printer is not available within the time specified, the job will not run. The default for <i>value1</i> is 1.
/m:value2	Specifies the number of clock ticks print can take to print a character on the printer. Valid values for <i>value2</i> range from 1 to 255. The default is 2.
/s:timeslice	Specifies the interval of time to be used by the MS-DOS scheduler for the print command.

- /q:qsize** Specifies the number of files allowed in the print queue — if you want more than 10. The minimum value for the /q switch is 4, the maximum, 32, and the default, 10. To change this default number of files, you must use the **print** command without any filenames, for example:
- print /q:32**
- /t** Deletes all files in the print queue (those files waiting to be printed).
- /c** Turns on cancel mode and removes the preceding filename and all following filenames from the print queue.
- /p** Turns on print mode and adds the preceding filename and all following filenames to the print queue.

The **print** command, when used with no options, displays the contents of the print queue on your screen without affecting the queue.

Notes:

- You can use the /d, /b, /u, /m, /s, and /q switches only the first time you run the **print** command after starting MS-DOS.
- Each print queue entry can contain a maximum of 64 characters, including the drive name. You might need to change directories first to avoid using extensive pathnames.
- Some applications have their own **print** commands. You should use an application's print facility to print files that you create with that application.

Examples:

The following command empties the print queue for the device named LPT1:

```
print /t /d:lpt1
```

The following command removes the Pencil.tst file from the default print queue:

```
print a:pencil.tst /c
```

The next two commands show how to remove the file Pencil.tst from the queue and then add the file Pen.tst to the queue:

```
print pencil.tst /c
```

```
print pen.tst /p
```

Prompt

Internal

Purpose:

Changes the MS-DOS command prompt.

Syntax:

prompt [[*text*][\$ *character*]...]

Comments:

This command lets you change the MS-DOS system prompt (for example, A>). If, when using the **prompt** command, you do not type a new value, the prompt is set to the default value, which includes the default drive name.

You can use the characters in the **prompt** command to create special prompts:

Type These Characters	To Get This Prompt
\$q	The = character
\$\$	The \$ character
\$t	The current time
\$d	The current date
\$p	The working directory of the default drive
\$v	The version number
\$n	The default drive
\$g	The character
\$l	The character
\$b	The character
\$	ENTER-LINEFEED
\$e	ASCII code X'1B' (escape)
\$h	Backspace (to erase a character that has been written to the prompt line)

Examples:

The following example sets the drive prompt to *drive:current directory*:

```
prompt $p
```

The following command sets a two-line prompt that displays the following:

```
Time = current time
```

```
Date = current date
```

```
prompt Time = %t%_Date = %d
```

If your terminal has an ANSI escape sequence driver, you can use escape sequences in your prompts. The following command, for example, sets your prompt in inverse video mode and returns to video mode for other text:

```
prompt $e[7m$n:$e[m
```

Rcrypt

External

Purpose:

Changes the format of a file to make the contents meaningless to anyone who does not know the *encryption key* (zero to eight characters that **rcrypt** uses to alter the file format).

Syntax:

rcrypt *pathname1* [*pathname2*]

where:

pathname1 is the pathname of the file to encrypt.

pathname2 is the pathname of the file to receive the encrypted file. If you omit *pathname2*, **rcrypt** uses the standard output device, the computer's display.

Comments:

When you execute **rcrypt**, a prompt asks you to:

Enter encryption key :

Type zero to eight characters that **rcrypt** can use as the key in the encryption algorithm. The characters do not appear on the screen as you type them.

rcrypt now prompts:

Re-enter encryption key for verification :

Retype the encryption key exactly as you did the first time. Retyping helps ensure that you type the encryption key correctly so you can later restore the file.

rcrypt is actually a filter. You can pipe data through it. For example, the following command encrypts the data in a directory listing and stores it in a file named *Dirfile.dat*.

`dir | rcrypt dirfile.dat`

The same prompts shown in the previous note appear when you use **rcrypt** in this manner:

`rcrypt con text`

Here, **rcrypt** encrypts keyboard input to a file named Text. To terminate input, press **F6** **ENTER**.

Notes:

- You decrypt a file in the same manner in which you encrypt it. You must supply the same key that you used for encryption.
- You can encrypt both ASCII and non-ASCII files.

Examples:

The following encrypts a file named Open (if Open is a standard file), and stores it in a file named Secret. If Open is an encrypted file, this command decrypts it and stores it in its original configuration in a file named Secret.

```
rcrypt open secret
```

Recover



Purpose:

Recovers a file or disk containing bad sectors.

Syntax:

```
recover [drive:][path]filename
```

or

```
recover [drive:]
```

Comments:

If the **chkdsk** command shows that a sector on your disk is bad, you can use the **recover** command to recover the entire disk or only the file containing the bad sector.

The **recover** command causes MS-DOS to read the file sector by sector and to skip the bad sectors. When MS-DOS finds a bad sector, it no longer allocates your data to that sector.

Notes:

- The **recover** command does not work on a network from a remote workstation.
- **Recover** doesn't work on drives used in the **subst** or **join** commands.

Examples:

To recover a disk in Drive A, you would type:

```
recover a:
```

Suppose you have a file named **Pencil.ad** that has a few bad sectors. To recover this file, you would use the following command:

```
recover pencil.ad
```


Ren (Rename)

Internal

Purpose:

Changes the name of a file.

Syntax:

```
rename [drive:][path]filename1 filename2
```

or

```
ren [drive:][path]filename1 filename2
```

where:

filename1 is the old name.

filename2 is the new name.

Comments:

Ren renames all files matching *filename1*. However, because you cannot rename files across disk drives, the **ren** command ignores any drive name that you specify with *filename2*.

You can use wildcards (* or ?) in either filename option, but if you use them in *filename2*, **ren** does not change the positions of the corresponding character.

Examples:

The following command changes the extension of all filenames ending in .txt to .doc:

```
ren *.txt *.doc
```

In the next example, **ren** renames a file named Chap10 (on Drive B) to Part10:

```
ren b:chap10 part10
```

The newly renamed file Part10 remains on Drive B.

Replace

External

Purpose:

Updates previous versions of files.

Syntax:

```
replace [drive:] pathname1 [drive:] [ pathname2 ] [/a]/p[/r]/s[/w]
```

where:

pathname1 is the source path and filename.

pathname2 is the target path and filename.

Comments:

The **replace** command performs two functions:

It replaces files in the target directory with files in the source directory that have the same name.

When you specify the /a switch, **replace** adds files that exist in the source directory (but not in the target directory) to the target directory.

You can use wildcards in source filenames.

The **replace** command accepts the following switches:

Switch	Purpose
/a	Adds new files to the target directory instead of replacing existing ones. You cannot use this switch with the /s switch.
/p	Prompts you with the following message before it replaces a target file or adds a source file: Replace filename? (Y/N)_
/r	Replaces read-only files as well as unprotected files. If you do not specify this switch, any attempt to replace a read-only file causes an error and stops the replace process.
/s	Searches all subdirectories of the target directory while it replaces matching files. This switch is incompatible with the /a switch. Replace never searches subdirectories in the source path.
/w	Waits for you to insert a disk before beginning to search for source files. If you do not specify the /w switch, replace begins replacing or adding files immediately. If you specify /w but not /a, replace displays: Press any key to begin replacing files If you specify both the /w and /a switches, replace displays: Press any key to begin adding file(s) As files are replaced or added, replace displays their filenames on the screen. At the conclusion of the replace operation, it displays a summary line: nnn file(s) added/replaced or No files added/replaced

Notes:

- You cannot use the **replace** command to update hidden files or system files.
- Upon completion, **replace** returns one of the following exit codes:

Code	Function
0	Command successful
1	Command line error
2	File not found
3	Path not found
5	Access denied
8	Insufficient memory
15	Invalid drive
Other	Standard MS-DOS error

You can test for these codes by using the error level condition of the batch processing if command.

Examples:

Suppose various directories on your hard disk, Drive C, contain files named **Phones.cli** that contain client names and phone numbers. To update these files and replace them with the latest version of the **Phones.cli** file from the disk in Drive A, you would type the following command:

```
place a:\phones.cli c:\ /s
```

This command replaces every file on Drive C that is named **Phones.cli** with the file **Phones.cli** from the root directory on Drive A.

Suppose you want to add some new printer device drivers to a directory called **c:\mstools**, which already contains several printer driver files for a word processor. To do this, you would type:

```
place a:*.prd c:\mstools /a
```

This command searches the default directory of Drive A for any files that have the extension **.prd** (that don't currently exist in the **\mstools** directory on Drive C) and then adds these files to **c:\mstools**.

Restore

External

Purpose:

Restores files that were backed up using the **backup** command.

Syntax:

```
restore drive1: [drive2:][pathname][/s][/p][/b:date][/a:date]  
[/e:time][/L:time][/m] [n]
```

where:

drive1 contains the backed up files.

drive2 is the target drive.

pathname identifies the file(s) you want to restore.

Comments:

The **restore** command can restore files to different media from similar or dissimilar disk types. The **restore** command accepts the following switches:

Switch	Purpose
/s	Restores subdirectories also.
/p	Prompts for permission to restore any files matching the file specification that are read-only or that have changed since the last backup.
/b:date	Restores only those files last modified on or before <i>date</i> .
/a:date	Restores only those files last modified on or after <i>date</i> .
/e:time	Restores only those files last modified at or earlier than <i>time</i> .
/L:time	Restores only those files last modified at or later than <i>time</i> .
/m	Restores only those files modified since the last backup.
/n	Restores only those files that no longer exist on the target disk.

Once MS-DOS has restored the file, use the **dir** or type command to be sure that the file was restored properly.

Notes:

- **Restore** cannot restore the system files. Use the **sys** command to restore these files. The MS-DOS 3.3 **restore** command will restore files backed up with either the MS-DOS 3.3 **backup** command or an earlier version of **backup**. Upon completion, **restore** returns one of the following exit codes:

Code	Function
0	Normal completion
1	No files were found to restore
3	Terminated by user
4	Terminated due to error

You can test for these codes by using the error level condition of the **if** batch processing command.

Examples:

To restore a file named **Invest.mnt** from the backup disk in Drive A to the **\irsharpe** directory on Drive C, type:

```
restore a: c:\irsharpe\invest.mnt
```

Press **ENTER** to let MS-DOS know that the backup disk is in Drive A. Then, once MS-DOS has restored the file, use the **dir** or **type** command to be sure that the file was restored properly.

Rmdir (rd)

Internal

Synonym:

rd

Purpose:

Removes a directory from a multilevel directory structure.

Syntax:

`rmdir [drive:] path`

or

`rd [drive:] path`

Comments:

Rmdir removes a directory that is empty except for the “.” and “..” symbols. These symbols refer to the directory and its parent directory. Before you can remove a directory, you must delete its files and subdirectories.

Notes:

You cannot remove a directory that has hidden files. MS-DOS has hidden files. Also, some application programs create their own hidden files.

Examples:

Suppose you want to remove a directory named \user\pete. You would follow these steps:

1. To assure that the directory is empty, type:
`dir \user\pete`
2. Then, from any directory except \user\pete, type:
`rmdir \user\pete`

Remember that if you are working in the same directory that you are trying to remove, you'll receive the following error message:

Invalid path, not directory, or directory not empty.

Select

External

Purpose:

Installs MS-DOS on a new diskette with desired country-specific information and keyboard layout.

Syntax:

```
select[[drive1:] [drive2:][path]][yyy][xx]
```

where:

drive1 is the source drive.

drive2 is the target drive.

Comments:

The **select** command lets you install MS-DOS on a new disk along with country-specific information (such as date and time formats and collating sequence) for a selected country.

The **select** command does the following:

- Formats the target disk.
- Creates both the Config.sys and Autoexec.bat files on the new disk.
- Copies the contents of the source disk, track by track, to the source disk.

The source drive can be either Drive A or Drive B. The default source drive is Drive A. The default target drive is Drive B.

If you choose a hard disk as the target, MS-DOS prompts you to type the correct internal label for that disk. If you type the wrong label, **select** ends.

If you type the correct label, **select** displays a second warning similar to:

```
WARNING, ALL DATA ON NON-REMOVABLE DISK  
DRIVE D: WILL BE LOST!
```

```
Proceed with Format (Y/N)?
```


If you type N (for No), **select** ends. If you type Y (for Yes), **select** formats the target disk. You can use the following options with the MS-DOS **select** command:

Option	Purpose
yyy	Specifies the country code. MS-DOS gathers country-specific information such as time and date formats from the Country.sys file for the country code specified.
xx	Specifies the keyboard code for the keyboard layout used. For a list of valid keyboard codes, see the discussion of the keyb command.

Examples:

Suppose you want to create a new MS-DOS disk that included the country-specific information and keyboard layout for Germany. With your source disk in Drive B and your target disk in Drive A, you would type:

```
select b: a: 049 gr
```

MS-DOS displays this message:

```
SELECT is used to install DOS the first time. SELECT erases
everything on the specified target and then installs DOS. Do you
want to continue (Y/N)?_
```

If the disk in Drive A contains any data files, they will be erased, unless you type N (for No). If the disk is blank, or reusable, type Y (for Yes) and press **ENTER**.

MS-DOS then prompts you to insert a new disk in Drive A. After it formats the disk, MS-DOS copies files from the source disk in Drive B to the target disk in Drive A.

Set

Internal

Purpose:

Sets one string of characters in the environment equal to another string for later use in programs.

Syntax:

`set [string = [string]]`

Comments:

Use the **set** command only if you want to set values for programs you have written.

When MS-DOS recognizes a **set** command, it inserts the given *string* and its equivalent into a part of memory reserved for the environment. If the *string* already exists in the environment, it is replaced with the new setting.

If you specify only the first string, **set** removes any previous setting of that string from the environment.

If you use the **set** command without options, MS-DOS displays the current environment settings.

When batch processing, you can also use the **set** command to define your replaceable parameters by name instead of by number. For example, if your batch file contains the statement, `type %file%`, you could use the **set** command to set the name that MS-DOS will use for that variable. In the following command, for example, **set** stores a value for MS-DOS to replace the `%file%` parameter with the filename `Taxes.86`:

`set file = taxes.86`

To change the replaceable parameter names, you don't need to edit each batch file. Note also that when you use text (instead of a number) as a replaceable parameter, you must end the name with a percent sign.

The **set** command is especially useful in the `Autoexec.bat` file because it lets you automatically set strings or parameters when you start MS-DOS. See Chapter 4, "Batch Processing," for more information about the `Autoexec.bat` file.

Examples:

The following command sets the string `include` to `c:\inc` until you change it with another `set` command:

```
set include = c:\inc
```

Setup

External

Purpose:

Initializes the system configuration parameters retained by your computer's non-volatile memory (CMOS).

Syntax:

setup

Comments:

Use **setup** to install information about your system so that your computer knows its configuration immediately when it boots. Run **setup** :

- When you change the hardware configuration of your computer (add or subtract memory, disk drives, video adapters, and so on).
- When you want to change your computer's retained date and time.
- When you replace the CMOS RAM battery.

When you execute **setup**, screen prompts show the current configuration of each setting, and ask if they are correct. If the setting is correct, answer **Y** **ENTER**. Answer **N** **ENTER** if a setting is not correct. **Setup** then asks for the new setting. Some settings only require that you press either **Y** or **N**, followed by **ENTER**. Other settings require specific information.

Configuration information includes:

- The current time and date (at the moment you answer the prompt).
- Diskette Drive A and B types – 360K, 720K, 1.2M, and so on.
- Fixed Disk Drives C and D types. If you do not know your hard disk type, check your hard disk manual or look inside your computer at the hard disk label placed on the top of your drive housing.
- Base memory – either 512 or 640 (512,000 bytes or 640,000 bytes).
- Expansion memory – can be up to 16000 (for 16,000,000 bytes).
- The prime video adapter type – color, monochrome, or EGA.

Notes:

- When **setup** is complete, you must reset and reboot your computer.

Share

External

Purpose:

Installs file sharing and locking.

Syntax:

```
share [/f: space][/L: locks]
```

Comments:

You can see the **share** command only when networking is active. If you want to install shared files, you can include the **share** command in your Autoexec.bat file. To learn more about shared files, see your network documentation. The **share** command accepts the following switches:

Switch	Purpose
/f: <i>space</i>	Allocates file space (in bytes) for the MS-DOS storage area used to record file sharing information. The default value for <i>space</i> is 2048. Note that each open file requires enough space for the length of the full filename plus 11 bytes, since an average pathname is 20 bytes in length.
/L: <i>locks</i>	Allocates the number of locks you want to allow. The default value for the <i>locks</i> is 20.

Once you have used the **share** command in an MS-DOS session, MS-DOS checks all read and write requests.

Examples:

The following example loads file sharing and uses the default values for the /f and /L switches:

```
share
```

Sort

External

Purpose:

Reads input, sorts the data, and then writes the sorted data to your screen, to a file, or to another device.

Syntax:

`[source] | sort [/r][/+n]`

or

`sort [/r][/+n] source`

where:

source is a filename or command.

Comments:

The **sort** command is a filter program that lets you alphabetize a file according to the character in a certain column. The **sort** program uses the collating sequence table, based on the country code and code page settings.

The `|` and redirection symbols direct data through the **sort** utility from source. For example, you can use the **dir** command or a filename as a source. You can use the **more** command or a filename as a destination.

The **sort** command accepts the following switches:

Switch	Purpose
<code>/r</code>	Reverses the sort, that is, sorts from Z to A, and then from 9 to 0.
<code>/+n</code>	Sorts the file according to the character in column <i>n</i> , where <i>n</i> is some number. If you do not specify this switch, the sort command sorts the file according to the character in the first column.

Unless you specify a source, **sort** acts as a filter and accepts input from the MS-DOS standard input (usually from the keyboard, from a pipe, or redirected from a file).

Notes:

- **Sort** does not distinguish between uppercase and lowercase letters.
- Characters above ASCII code 127 are sorted based on information found in the `Country.sys` file, or in an alternate file specified by the `country` command in your `Config.sys` file.
- For more information about using redirection symbols with commands, see Chapter 2, "About Commands."

Examples:

The following command reads the file `Expenses.txt`, sorts it in reverse order, and displays it on your screen:

```
sort /r expenses.txt
```

The following command pipes the output of the `dir` command to the `sort` filter. This filter sorts the directory listing starting with column 14 (the column in the directory listing that contains the file size) and sends the output to the screen. The result is a directory sorted by file size:

```
dir | sort /+14
```

The following command does the same thing as the previous one, except that the `more` filter gives you a chance to read the sorted directory one screen at a time:

```
dir | sort /+14 | more
```

Subst



Purpose:

Substitutes a path with a string.

Syntax:

`subst [drive: drive: path]`

or

`subst drive: /d`

Comments:

The **subst** command lets you associate a pathname alias path with a drive letter. This drive letter then represents a virtual drive because you can use the drive letter in commands as if it represented an actual physical drive.

When MS-DOS finds a command that uses a virtual drive, it replaces the drive letter with the path and treats that new drive letter as though it belonged to a physical drive.

If you type the **subst** command without options, MS-DOS displays the names of the virtual drives in effect.

Use the /d switch to delete a virtual drive.

Notes:

- The following commands do not work on drives used in the **subst** command (or the **join** command):

chkdsk	label
diskcopy	recover
fdisk	sys
format	

Examples:

The following command creates a virtual drive, Drive Z, for the pathname b:\user\betty\forms:

```
subst z: b:\user\betty\forms
```

Note: This example assumes that you have included the line, `lastdrive = z`, in your `Config.sys` file.

Now, instead of typing the full pathname, you can reach this directory by typing the name of the virtual drive:

```
z:
```

Sys



Purpose:

Transfers the MS-DOS system files from the disk in the default drive to the disk in the specified *drive*.

Syntax:

`sys drive:`

Comments:

The **sys** command updates your system files on a disk. You must type a drive letter following the **sys** command.

Notes:

- The two files that are transferred are both hidden files that do not appear when you type the **dir** command.
- **Sys** does not transfer the **Command.com** file (the command processor). To transfer **Command.com** to the target disk, you must use the **copy** command. MS-DOS system files are no longer required to be contiguous. This means that you do not have to reformat your disk when you want to copy a new version of MS-DOS on a disk containing system files for MS-DOS version 3.2 or earlier.
- **Sys** doesn't work on drives involved with **subst** or **join** commands. **Sys** does not work on a network.

Examples:

To copy the MS-DOS system files from your working directory to a disk in Drive A, you would type:

`sys a:`

Time

Internal

Purpose:

Allows you to enter or change the time known to the system.

Syntax:

time [*hours: minutes[: seconds [. hundredths]]*]

Comments:

MS-DOS keeps track of time in a 24-hour format and uses the time information to update the directory whenever you create or change a file.

Typing the **time** command without options displays the current time and gives you an opportunity to change it:

Current time is *hh:mm:ss.cc*

Enter new time: _

If you do not want to change the time shown, press **ENTER**.

If you want to change the time, type in a new value in the 24-hour clock format. The following are valid values:

hours = 0-23

minutes = 0-59

seconds = 0-59

hundredths of seconds = 0-99

Separate these elements (seconds and hundredths of seconds are optional) with the separator defined in the country-dependent information file. For the United States, use a colon (:). You can also type the new time directly on the command line.

If you do not type a valid time, MS-DOS displays the following message and then waits for you to type a valid time:

Invalid time

Enter new time: _

Notes:

- You can change the **time** command format by changing the country command in the Config.sys file. For more information, see Appendix B, “How to Configure Your System.”
- The **time** command sets your computer’s internal clock.
- You cannot specify a drive for this command.

Examples:

To reset the time of day on your computer’s clock, you can type the **time** command by itself, and MS-DOS will prompt you for the correct time. Or, you can include the correct time when you type the command. For example, to set your computer’s clock at 1:36 p.m., you could type:

time 13:36

Tree

External

Purpose:

Displays the path (and, optionally, lists the contents) of each directory and subdirectory on the given drive.

Syntax:

```
tree [drive:] [/f]
```

Comments:

The **tree** command lists the full paths of all directories, along with the names of their subdirectories.

The /f switch displays the names of the files in each directory.

Notes:

Another way to list all the subdirectories in your working directory is to type `dir *`. This also reports all files that have no file extensions. Directories are identified with the label

Examples:

To see the names of all directories and subdirectories on your computer, type:

```
tree
```

To see, one screen at a time, the files in all the directories on Drive C, you could type:

```
tree c: /f | more
```

To print that same list on a printer, you would use:

```
tree c: /f prn
```

Type

Internal

Purpose:

Displays the contents of a text file on the screen.

Syntax:

`type [drive:]filename`

Comments:

You can use the **type** command to view a text file without modifying it. (Use **dir** to find the name of a file and **edlin** to change the contents of a file.)

Note that when you use **type** to display a file that contains tabs, all the tabs are expanded to the current setting for tabs (generally eight spaces). If you try to display a binary file or a file created by an application program, you might see strange characters on the screen, including bells, formfeeds, and escape sequence symbols.

Examples:

To display the contents of a file called **Holiday.mar**, you would type:

`type holiday.mar`

If the file whose contents you wish to display is fairly long, you could use a command like this to display the file's contents one screen at a time:

`type holiday.88 | more`

Ver

Internal

Purpose:

Prints the MS-DOS version number.

Syntax:

ver

Comment:

To see the version number of MS-DOS that you are using, simply type the **ver** command. The version number then appears on your screen.

Examples:

When you type the **ver** command, the following message appears:

MS-DOS Version 3.30

Verify

Internal

Purpose:

Turns the **verify** switch on or off when writing to a disk.

Syntax:

verify on
or
verify off

Comments:

Use the **verify** command to check that your files are written correctly to the disk (no bad sectors, for example). MS-DOS verifies the data as it writes it to a disk. You receive an error message only if MS-DOS is unable to successfully write your data to a disk.

Notes:

This command has the same purpose as the /v switch in the **copy** command.

Examples:

To see the current setting of **verify**, use the **verify** command without an option:

verify

Verify on remains in effect until a program changes it or until you type:

verify off

Vol

Internal

Purpose:

Displays the disk volume label or volume identification if it exists.

Syntax:

vol [*drive*:]

Comments:

This command displays the volume label of the disk in a specific drive. If you do not type a drive letter, MS-DOS displays the volume label of the disk in the default drive.

Notes:

- You cannot specify a drive name before this command. For example, if you typed `b:vol c:`, MS-DOS would display an error message because of the reference to Drive B. MS-DOS assumes that this command resides on the drive from which you are working.
- For more information about how MS-DOS uses volume labels, see the discussions of the **label** and **format** commands in this chapter.

Examples:

To find out the volume label for the disk in Drive A, you would type:

vol a:

If the volume label is "DOS 3-3," MS-DOS responds by displaying:

Volume in drive A is DOS 3-3

Xcopy

External

Purpose:

Copies files and directories, including lower level directories if they exist.

Syntax:

```
xcopy [drive:] pathname [drive:][pathname][a][d:date]  
[e][m][p][s][v][w]
```

or

```
xcopy drive:[ pathname] [drive:][pathname][a][d:date][e]  
[m][p][s][v][w]
```

Comments:

The first set of *drive:* and *pathname* parameters specifies the source file or directory that you want to copy. The second set names the target. You must include at least one of the source parameters. If you omit the target parameters, **xcopy** assumes that you want to copy the files to the default directory. If you do not specify the *pathname* option, **xcopy** uses the default directory with the default filename, *.*.

The **xcopy** command accepts the following switches:

Switch	Purpose
/a	Copies source files that have their archive bit set. Does not modify the archive bit of the source file. For information on how to set the archive attribute, see the explanation of the attrib command.
/d:date	Copies source files modified on or after the specified date. Note that the date format might vary, depending on the country code that you are using. For more information, see the discussion of the date command.
/e	Copies any subdirectories, even if they are empty. You must use this switch with the /s switch.
/m	Same as the /a switch, but after copying a file, it turns off the archive bit in the source file. For information on how to set the archive attribute, see the discussion of the attrib command.
/p	Prompts you with (Y/N?), enabling you to confirm whether you want to create each target file.
/s	Copies directories and lower level subdirectories, unless they are empty. If you omit this switch, xcopy works within a single directory.
/v	Verifies each file as it is written to the target to assure that the target files are identical to the source files.
/w	Waits before xcopy starts copying files. Xcopy displays: Press any key when ready to start copying files Press a key to continue. Press CTRL C to terminate the xcopy command.

Notes:

- If you have a disk that contains files in subdirectories and you want to copy it to a target disk that has a different format, you should use the **xcopy** command rather than **diskcopy**. The **diskcopy** command copies disks track by track and requires your source and target disks to have the same format.

- If **xcopy** encounters an error, it returns one of the following exit codes:

Code	Function
0	Copy without error
1	No files found to copy
2	CTRL-C entered by user to terminate xcopy
4	Initialization error. There is not enough memory—invalid drive or command line syntax, file not found, or path not found.
5	Int 24 error occurred. The user aborted from INT 24 error reading or writing disk.

You can test for these codes by using the error level condition of the batch processing **if** command.

Examples:

The following example copies all the files and subdirectories (including any empty subdirectories) on the disk in Drive A to the disk in Drive B:

```
xcopy a: b: /s /e
```

The **xcopy** command might prompt you to specify whether the target is a file or a directory. If you don't want to receive this prompt, type:

```
copy /b xcopy.exe mcopy.exe
```

This example creates a new command called **mcopy**. Now you can use the **mcopy** command the same way as you use the **xcopy** command, but **mcopy** automatically determines whether the target is a file or a directory.

Mcopy uses the following rules for copying files:

- If the source is a directory, the target is a directory.
- If the source includes multiple files, the target is a directory.

If you append a backslash (\) to the end of the target name, the target is a directory. For example, the following command creates the directory **a:\workers** if it doesn't already exist, and copies the file **Payroll** to it:

```
xcopy payroll a:\workers\
```

Batch Processing

This chapter explains:

- How to create a batch file
- How an Autoexec.bat file works
- How to use replaceable parameters in a batch file
- How to run a batch file

Note: If you are not writing batch programs, you do not need to read this chapter.

Why Use Batch Files?

You might often find yourself repeatedly typing the same sequence of commands to perform some common task. With MS-DOS, you can put this command sequence into a special file called a *batch file*, and then run the whole sequence of commands by simply typing the name of the batch file.

Note that you don't need to type the batch file's extension, even though all your batch files must include the .bat extension in their filenames.

MS-DOS performs these "batches" of your commands as if you had typed them from the keyboard. This is called *batch processing*. By using a batch file, you only have to remember to type one command, instead of several. In effect, you use batch files to create personalized commands.

How to Create Batch Files

You can create a batch file by using **edlin**, the MS-DOS line editor, by using the **copy** command, or by using a word processor that saves files as ASCII text. To create files with **edlin**, refer to Chapter 6, "The Line Editor (Edlin)" for more information. The examples in this chapter show you how to use the **copy** command to create batch files.

Suppose, for example, that you want to create a batch file to format and check a new disk. To do this, follow these steps:

1. First, type:

```
copy con checknew.bat
```

Press **ENTER**. This command tells MS-DOS to copy the information from the console (keyboard) to the file Checknew.bat.

2. Next, type the following lines, pressing **ENTER** after each:

```
rem This is a file to format and
rem check new disks.
rem It is named CHECKNEW.BAT.
pause Insert new disk in Drive B:
format b: /v
chkdsk b:
```

3. After the last line, press **CTRL** **Z** and then press **ENTER** to save the batch file. MS-DOS displays the message 1 File(s) copied to show that it created the file.

4. Now, to execute the file, type:

```
checknew
```

The result is the same as if you typed the lines in the .bat file from the keyboard as individual commands.

About Batch Processing

Here are a few things you should know before you run a batch process with MS-DOS:

- You must name each batch file with an extension of .bat.
- To execute a batch file, you type only its filename and not the extension.
- If you press **CTRL** **C** while the batch file is running, MS-DOS asks you to confirm that you want to terminate the batch process.
- If you remove the disk that contains a batch file that is running, MS-DOS prompts you to reinsert the disk so that it can continue processing the file.
- You can specify the name of another batch file as the last command in a batch file. This feature allows you to call one batch file from another when the first finishes.

- You can use any of the redirection symbols (<, >, >>) in a batch file. For more information on using these symbols, see Chapter 2, "About Commands."
- You can use an @ character at the front of a command line in a batch file to prevent that line from echoing.
- You can use the pipe symbol (|) in a batch file.
- Setting the directory or drive affects every subsequent command in the batch file.
- Setting environment strings also affects every subsequent command in the batch file.

Note: If you have more than one external command with the same name, MS-DOS runs only one of them, according to the following order of precedence:

```
.com  
.exe  
.bat
```

Suppose, for example, that your disk includes the files `Format.exe` and `Format.bat`. If you were to type the external command **format**, MS-DOS would always run the program `Format.exe` first.

In order to run the batch file `Format.bat`, you need to place it in a separate directory and give a path along with the external command.

For example, if your `Format.bat` file is in a directory named `\commands\batch` and you want to run it, type:

```
\commands\batch\format
```

The Autoexec.bat File

An `Autoexec.bat` file lets you run programs automatically when you start MS-DOS. This can be useful when you want to run a specific application under MS-DOS and when you want MS-DOS to execute a batch program each time you start your computer. By using an `Autoexec.bat` file you can avoid loading two separate disks to perform these tasks.

When you start your computer, MS-DOS searches the root directory of the default disk drive for a file named Autoexec.bat. If it finds the Autoexec.bat file, MS-DOS immediately processes it, bypassing the date and time prompts. If MS-DOS does not find an Autoexec.bat file, the date and time prompts appear automatically.

Note: MS-DOS does not prompt you for a current date and time unless you include the **date** and **time** commands in the Autoexec.bat file. See Chapter 3, "MS-DOS Commands," for more information on the **date** and **time** commands.

The figure on the following page shows what happens when you start MS-DOS:

How to Create an Autoexec.bat File

There are many things you can do with an Autoexec.bat file to help you use MS-DOS more efficiently. For instance, you might want to set the time and date, your path, and any other options that you plan to use on a regular basis.

When you create your Autoexec.bat file, you must put it in the root directory of your MS-DOS disk.

If, for example, you want to automatically load GW-BASIC and run a program called Menu each time you start MS-DOS, create an Autoexec.bat file as follows:

1. Type the following command and then press **ENTER**:

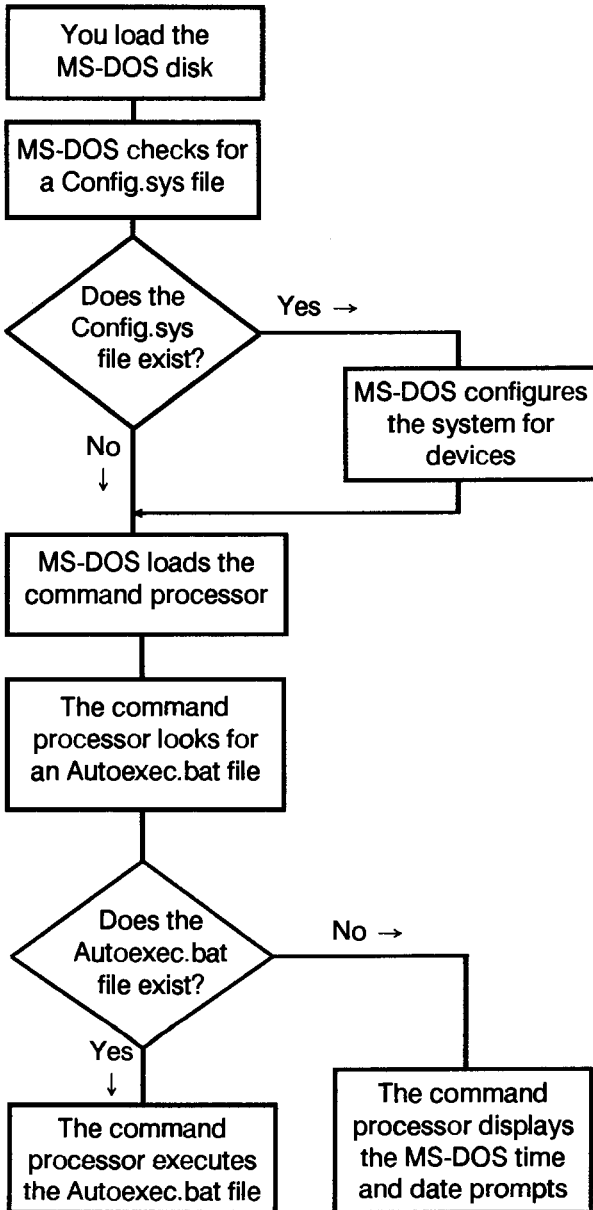
copy con autoexec.bat

This command tells MS-DOS to copy what you type from the keyboard into the Autoexec.bat file. Note that you must put the Autoexec.bat file in the root directory of your MS-DOS disk.

2. Now, type the following lines:

```
path = c:\;c:\bin;a:\
prompt [$p]
cls
gwbasic menu
```

3. After the last line, press **CTRL** **Z** and then press **ENTER**



3. After the last line, press **CTRL** **Z** and then press **ENTER** to copy these lines. The Menu program now runs automatically whenever you start MS-DOS.

Once your Autoexec.bat file is set up (as in this example), it sets your command search path, sets your prompt to display the default drive and directory, clears the screen, and tells MS-DOS to load GW-BASIC and run the Menu program. To run your own GW-BASIC program, type its name in place of Menu in the example.

In addition to GW-BASIC programs, you can also put any MS-DOS command or series of commands in the Autoexec.bat file.

How to Create a Batch File with Replaceable Parameters

There might be times when you want to create a program and run it with different sets of data. These data sets can be stored in various MS-DOS files.

With MS-DOS, you can create a batch (.bat) file with replaceable (*dummy*) parameters, where a parameter is a command option that you define. These parameters, named %0-%9, hold the places for the values that you supply when you give the batch command.

Replaceable parameters make batch files more flexible and easy to use. For example, you can create a batch file called Sorter.bat that sorts a file containing a specific sequence of characters or strings. Each time you run the Sorter batch file, you tell MS-DOS which string you want, which file to search to find that string, and which temporary file to use for sorting. Sorter then prints the resulting list on the printer.

To create the Sorter.bat file, type the following command and then press **ENTER**:

```
copy con sorter.bat
```

Now, type the following lines:

```
type %2 | find "%1" > %3
type %3 | sort > prn
del %3
```

To save the batch file, press **CTRL** **Z** and then **ENTER** . The batch file `Sorter.bat` now consists of three command lines and is on the disk in the default drive.

When you execute the file, MS-DOS sequentially replaces %1, %2, and %3 with the parameters you supply. If you use the dummy parameter %0, MS-DOS always replaces it with the drive name (if specified) and the filename of the batch file (for example, `Sorter`).

Notes:

You can specify up to ten replaceable parameters (%0-%9). If you want to specify more than ten, refer to the **shift** command later in this chapter.

If you use the percent sign as part of a filename within a batch file, you must type it twice. For example, to specify the file `Abc%.exe`, type `abc%%.exe` in the batch file.

How to Use Named Parameters in a Batch File

In addition to the ten numeric replaceable parameters, you can use named parameters in batch files. Named parameters allow you to define your replaceable parameters by name instead of by number.

When referred to in a file, the name of a replaceable parameter is placed between two percent signs.

Named parameters differ from replaceable parameters in that you do not specify their values on the command line.

Instead, MS-DOS retrieves the value of each named parameter from its environment.

You can use the MS-DOS **set** command to set the value of a named parameter before you run your batch file, or you can include the **set** command in your batch file.

For example, suppose that you want to create a batch file, `Mydel.bat`, that moves a file you want to delete into a separate directory. You might want to use this method to be sure that you don't delete files accidentally from an important directory.

The file Mydel.bat might contain the following lines:

```
echo off
echo Before you use this batch file, you must
echo specify the directory by typing the
echo following command at the MS-DOS prompt:
echo set deldir = directory
echo Press Control-C to exit if you haven't
echo set deldir or if deldir does not exist.
pause
copy %1 %deldir%
del %1
dir /w %deldir%
echo All done.
```

For this Mydel.bat file to work, you must create a directory named Deleted. The following command line sets the directory name to Deleted:

```
set deldir = \deleted
```

Now, to move the file Report23.jun to the Deleted directory, type:

```
mydel report23.jun
```

The batch file automatically replaces the %deldir% parameter with the directory name Deleted.

You might find named parameters easier to use than replaceable parameters because you won't have to include as much information on the command line. For example with Mydel.bat, you don't have to type the directory name on the command line, yet you can change the name of the directory without having to edit the batch file.

How to Run a Batch File

To run the batch file `Sorter.bat`, type the batch filename followed by the parameters that you want MS-DOS to substitute for %1, %2, and %3.

Suppose that on the disk in Drive A you have a file that lists your customers' names and regions. The file might look something like this:

Shores, Betty	north
Moynihan, Ann	south
Kraig, Heidy	north
Martin, Pete	east
Lennon, Patrick	south
Pai, Fernando	north
Evans, Rick	west
Moss, Melissa	north

If you want to print an alphabetical list of the customers in the north, you can run the `Sorter` batch file, with the appropriate parameters, by typing the following command and then pressing **ENTER**:

```
sorter north a:customer temp.fil
```

The output on the printer looks like this:

Kraig, Heidy	north
Moss, Melissa	north
Pai, Fernando	north
Shores, Betty	north

The next table shows how MS-DOS replaces each of the parameters:

Batch filename	(%0)	sorter
Parameter1	(%1)	north
Parameter2	(%2)	a:customer
Parameter3	(%3)	temp.fil

The result is the same as if you had typed each of the commands in `Sorter` with its parameters:

```
type a:customer | find "north" temp.fil
type temp.fil | sort prn
del temp.fil
```

Using the batch file, however, saves typing time and is much easier to remember.

How to Use Temporary Files

When using batch files, you can use a temporary file to hold your work. Use the same name each time you want to use a temporary file.

However, if you are using more than one batch file that uses the same temporary file, you might lose the contents of this temporary file. To avoid this problem, use a replaceable parameter to specify the name of the temporary file. Then, each time you run the batch file, you can substitute a unique filename, and you won't have to worry about information from one batch file getting into another. It's also a good idea to delete temporary files when you finish using them. This saves memory space on your disk.

Batch Processing Commands

Now that you have seen some of the capabilities of batch files, you'll find out, in this section, how to add power and flexibility to your batch programs by using batch processing commands. The following table lists these batch commands and describes what they do:

Command	What It Does
call	Calls one batch file from another without ending itself.
echo	Turns echo on or off, or displays the current setting.
for	Performs a command for a set of files.
goto	Processes commands starting with the line after the specified label.
if	Performs a command if a condition is met.
pause	Pauses during the processing of a batch file.
rem	Displays a comment in a batch file.
shift	Increases the number of replaceable parameters.

Batch commands are also internal commands, so each one is marked with the internal command icon (the boxed letter I) used in Chapter 3, "MS-DOS Commands."

Call

Internal

Purpose:

Calls one batch file from another without ending itself.

Syntax:

`call [drive:][path] batchfile [argument]`

where:

batchfile is the batchfile you want to call.

argument is the command in this batch file that is run following batchfile.

Comments:

The **call** command is used within one batch file to call another one. *Batchfile* must have a filename extension of *.bat*. When *batchfile* terminates, the calling batch file resumes running at *argument*. If *argument* is omitted, the calling batch file resumes running at the command immediately following the **call** command.

Notes:

- Do not use pipes and redirection symbols with the **call** command.
- A batch file can make a recursive call to itself, but there should be a termination condition that is eventually met.

Examples:

To run the Checknew.bat file from another batch file, use the following command within the first batch file:

`call checknew`

Echo

Internal

Purpose:

Turns the batch echo feature on and off.

Syntax:

```
echo [on]
echo [off]
echo [message]
```

Comments:

Normally, commands in a batch file appear on the screen when MS-DOS executes them. You can turn off this feature by using the off option with **echo**. Similarly, you can turn **echo** back on by using the on option.

If you do not specify on or off, **echo** displays the current setting. The command, **echo message** (where *message* is a line of text), is useful only if **echo** is off and if you are using a batch file. If, in your batch file, you type **echo** followed by a message, you can print messages on your screen.

You can also put several **echo message** commands in your batch file to display a message that is several lines in length.

An @ character placed in front of a command line in a batch file prevents that line from echoing.

Examples:

The following is an example of a batch file message of more than one line:

```
echo off
echo This batch file
echo formats and checks
echo new disks.
```

If you want to turn echo off and do not want the command itself to be echoed, include the @ character before the command line:

```
@echo off
```


For

Internal

Purpose:

Performs a command for a set of files.

Syntax:

for %%c in set do command

(for batch processing)

for %c in set do command

(for interactive processing)

Comments:

To avoid confusion with the %0-%9 batch parameters, the variable *c* can be any character except 0,1,2,3,...9.

set is (item)*

This command sequentially sets the *%%c* variable to each member of *set* and uses the variable to evaluate *command*. If a member of *set* is an expression involving a wildcard (* or ?), then the variable is set to each matching *item* from the disk. In this case, only one such item is in *set*, so the command ignores any item other than the first.

Examples:

The following example assigns the variable *%f* to any files ending with *.asm in the working directory:

*for %%f in (*.asm) do masm %%f*

The example then executes a command of the following form:

masm filename

Filename can be any one of the following:

invoice.asm
receipts.asm
taxes.asm

The following example assigns the variable `%%f` to the files named Report, Memo, and Address; it then deletes each of these files:

```
for %%f in (report memo address) do del %%f
```

You must use two percent signs (`%%`) so that one of the signs remains after the batch parameter (`%0-%9`) processing is complete.

If you have `%f` instead of `%%f`, the batch parameter processor sees the `%`, looks at `f`, decides that `%f` is an error (a bad parameter reference), and throws out the `%f` so that the `for` command never sees it.

Note that if you are using the `for` command outside of a batch file, you use only one percent sign.

Goto

Internal

Purpose:

Processes commands starting with the line after the specified label.

Syntax:

`goto [:] label`

Comments:

Goto lets you take commands from the batch file beginning with the line after the *label*, where a *label* is defined as the characters following **goto**. This label might include spaces but not other separators, such as semi-colons or equal signs. If your batch file does not contain the specified label, the batch process terminates.

Note: During batch processing, a batch file ignores any line that starts with a colon (:).

Examples:

The following example sends the program processor to the label named **Endonly** if no errors occur when you format the disk in Drive A:

```
:begin
echo off
format a: /s
if errorlevel 0 goto end
echo An error occurred during formatting.
:end
echo End of batch file.
```

If

Internal

Purpose:

Performs a command based on the result of a condition.

Syntax:

if [not] *errorlevel number command*

or

if [not] *string1 = = string2 command*

or

if [not] exist *filename command*

Comments:

The **if** statement allows conditional execution of commands. When the condition is true, MS-DOS executes the command; otherwise, it ignores the command. The conditions are described as follows:

Condition	Description
<i>errorlevel number</i>	True if, and only if, the previous program executed by <i>command.com</i> has an exit code equal to, or greater than, <i>number</i> . (When a program finishes, it returns an exit code via MS-DOS.) You can use this condition to perform other tasks that are based on the previous program's exit code.
<i>string1 = = string2</i>	True if, and only if, <i>string1</i> and <i>string2</i> are identical after parameter substitution. Strings cannot contain separators, such as commas, semicolons, equal signs, or spaces.
exist <i>filename</i>	True if, and only if, <i>filename</i> exists.

If you specify the **not** parameter, MS-DOS executes the command when the condition is false.

Notes:

For more information about exit codes returned by an MS-DOS command, see the specific command in Chapter 3, "MS-DOS Commands."

Examples:

The following example prints the message can't find datafile if the file Product.dat does not exist on the disk:

```
if not exist product.dat echo can't find datafile
```

The following example sends the program processor to the label named Endonly if no errors occur when you format the disk in Drive A.

```
:begin  
echo off  
format a: /s  
if errorlevel 0 goto end  
echo An error occurred during formatting.  
:end  
echo End of batch file.
```

Pause

Internal

Purpose:

Suspends execution of a batch file.

Syntax:

`pause [comment]`

Comments:

When a batch file is running, you might want to change diskettes or perform some other action. The `pause` command suspends execution of the batch file until you press any key (unless you press `CTRL C`).

When the command processor encounters **pause**, it prints this message:

Strike a key when ready . . .

If you press `CTRL C`, MS-DOS displays this message:

Terminate batch job (Y/N)?

If you type Y (for Yes) in response to this prompt, the batch file ends, and control returns to the operating system. Therefore, you can use **pause** to divide a batch file into pieces that allow you to end the batch command file at any intermediate point.

The *comment* parameter is useful when you want to display a special message. Unless `echo` is off, **pause** displays this comment before the **Strike a key** message.

Note: The **pause** and comment line of your batch file does not appear if `echo` is off.

Examples:

Suppose you want a program to display a message that asks the user to change disks in one of the drives. To do this, you might use the following:

Pause Please put a new disk into Drive A

If `echo` is on, this line precedes the **Strike a key** message when you run the batch file.

Rem

Internal

Purpose:

During execution of a batch file, **rem** displays remarks that are on the same line as the **rem** command in that batch file.

Syntax:

```
rem [comment]
```

Comments:

The *comment* parameter is a line of text that helps you identify and remember what your batch file does.

The only separators allowed in the comment are spaces, tabs, and commas.

You can use **rem** in your batch file without a comment to add spacing for readability.

Note: If **echo** is off, the **rem** comment is not displayed.

Examples:

The following example shows a batch file that uses remarks for both explanation and spacing:

```
rem This file formats and checks new disks
rem It is named checknew.bat
rem
pause Insert new disk in Drive B
format B: /v
chkdsk B:
```

Shift

Internal

Purpose:

Lets you change the position of replaceable parameters.

Syntax:

shift

Comments:

You can use the **shift** command to change the positions of (replaceable) command line parameters.

Usually, command files are limited to handling ten parameters, %0 through %9. But by using the **shift** command, you can access more than ten parameters. This means that if there are more than ten parameters given on a command line, those that appear after the tenth (%9) are shifted one at a time into %9.

You can use the **shift** command even if you have less than ten parameters.

Warning: There is no backward **shift** command. Once you execute **shift**, you cannot recover the first parameter (%0) that existed before the shift.

Examples:

The following file, called `Mycopy.bat`, shows how to use the **shift** command with any number of parameters. It copies a list of files to a specific directory.

```
rem mycopy.bat copies
rem any number of files
rem to a directory.
rem The command is
rem mycopy dir files
set todir = %1
:one
shift
if "%1" = "" goto two
copy %1 %todir%
goto one
:two
set todir =
echo All done
```

MS-DOS Editing and Function Keys

This chapter explains:

- The MS-DOS editing and function keys
- The editing template
- The MS-DOS control characters

Special MS-DOS Editing Keys

Many operating systems handle command input differently than MS-DOS does. One difference that sets MS-DOS apart is its set of special editing keys.

For instance, with MS-DOS you don't have to type the same sequences of keys repeatedly, because the most recently typed command line is automatically placed in a special storage area called a *template*.

By using the template and special editing keys, you can take advantage of the following MS-DOS features:

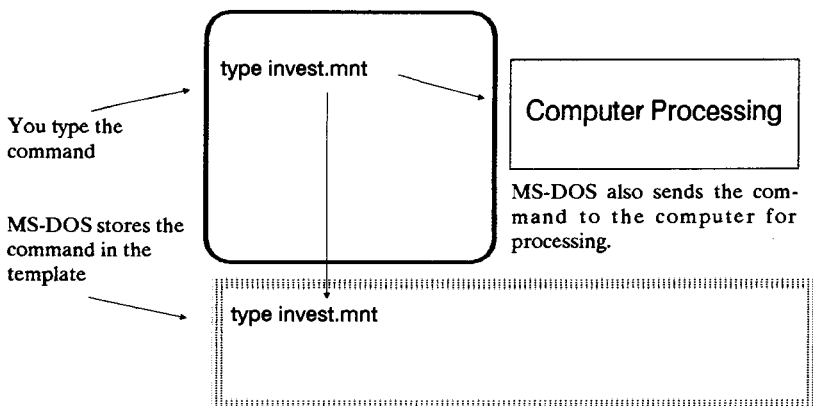
- Repeat a command instantly by pressing two keys.
- Edit a mistake in a command line without having to retype the entire line.
- Edit and execute a command line that is similar to a previous one with a minimum of typing.

How MS-DOS Uses the Template

Typing a command and pressing **ENTER** causes MS-DOS to send the command to the command processor (Command.com) for execution. At the same time, MS-DOS also saves a copy of this command in the template. You can then recall or modify the command by using the MS-DOS special editing keys.

The following figure shows how the template relates to the command line:

The MS-DOS Editing Keys



The MS-DOS editing keys are a set of editing tools that can save you time. You can use them to correct typing mistakes, repeat frequently used commands, or create similar command lines.

These keys are described briefly in the following table and in greater detail in Chapter 6, "The Line Editor (Edlin)."

Key	Editing Function
F1	Copies one character from the template to the command line.
F2	Copies characters up to the specified character in the template, and puts these characters on the command line.
F3	Copies all remaining characters in the template to the command line.
DEL	Skips over (does not copy) a character in the template.
F4	Skips over (does not copy) the characters in the template up to the specified character.
ESC	Voids the current input and leaves the template unchanged.

Key	Editing Function
INS	Enters/exits insert mode.
F5	Makes the new line the new template.
F6	Puts a CTRL Z (1AH) end-of-file character in the new template.

You'll use these special editing keys in conjunction with the template, which you'll learn to use in the next section.

How to Use the MS-DOS Template

Suppose you want to view directory information for the file `Invest.mnt`. To get this information, type:

```
dir invest.mnt
```

MS-DOS saves this command in the template. To repeat the command, press **F3** and **ENTER**.

When you press **F3**, MS-DOS copies the contents of the template to the command line; pressing **ENTER** then sends the command line to the command processor for execution.

To display information about a file named `Invest.rpt`, use the contents of the template. Pressing **F2**, followed by "m," copies all characters from the template to the command line, up to but not including the "m." MS-DOS displays:

```
dir invest._
```

Note that the underline is your cursor.

Now, type the letters "rpt" as shown below:

```
dir invest.rpt_
```

The command is now in the template and ready to go to the command processor for execution. To run the command, press **ENTER**.

Now, assume that you want to run the following command:

```
type invest.rpt
```

Type word "type", press **INS**, the space bar, **F3**, then **ENTER**.

As you type, the characters appear directly on the command line, overwriting their corresponding characters in the template. Before you press **INS**, “type” replaces “dir” (and the space following it) in the template. After you press **INS**, this automatic replacement feature is turned off.

To insert a space between the word “type” and the filename `Invest.rpt`, press **INS** and then the space bar. Finally, to copy the rest of the template to the command line, press **F3** and **ENTER**.

The command line `type invest.rpt` is processed by MS-DOS, and the template now looks like this:

`type invest.rpt.`

If you misspell a word, for instance “pyte” instead of “type”), an error occurs. Instead of losing the command, save the misspelled line before pressing **ENTER**. Press **F5**, then **ENTER** to create a new template:

`pyte invest.rpt`

Use **DEL** and **INS** to correct the line:

DEL DEL F1 INS yp F3

To illustrate how these keys affect the command line, compare each key with its result and a description of the effect:

Key Press	The Result	Description
DEL	—	Skips over first template character
DEL	—	Skips over second template character
F1	t_	Copies third template character
INS yp	typ_	Inserts two characters, y and p
F3	type invest.rpt_	Copies rest of template

Notice that **DEL** does not affect the command line. Instead, it affects the template by deleting the first character.

Similarly, pressing **F4** deletes characters in the template up to, but not including, a given character.

The MS-DOS Control Characters

A control character affects command line in a special way. For example, use **CTRL** **C** to stop the current command, and **CTRL** **S** to suspend the screen output from a command.

To type a key sequence, such as **CTRL** **C**, press **CTRL**, then while holding **CTRL**, press **C**.

The following table lists and describes the MS-DOS control characters:

Control Character	What It Does
CTRL C	Aborts the current command.
CTRL H	Removes the last character from a command line, and erases that character from the terminal screen.
CTRL J	Inserts a physical end-of-line but does not empty the command line. Use the LINEFEED key to extend the current logical line beyond the physical limits of the terminal screen.
CTRL N	Causes echoing of output to a line printer.
CTRL P	Causes terminal output to a line printer.
CTRL S	Halts screen scrolling. To resume screen scrolling, press CTRL S again.
CTRL X	Cancels the current line, empties the command line, then displays a backslash (\), a carriage return, and a linefeed. This key sequence does not affect the template used by the special editing commands.

What Comes Next?

This chapter has introduced you to MS-DOS's special editing and function keys. In the next chapter, you'll learn how to use them with Edlin, the MS-DOS line editor.

The Line Editor (Edlin)

Edlin is the MS-DOS line editor that you can use to create text files and save them on your disks. **Edlin** also helps update existing files by deleting, changing, and inserting lines. And even though it isn't a word processor, **edlin** does make it easy to create and revise files such as memos, letters, reports, or GW-BASIC programs.

This chapter explains:

- How to start **edlin**
- How to end **edlin** and save edits
- How to use the MS-DOS special editing keys with **edlin**
- How to use **edlin** commands

How Edlin Works

Edlin divides the text from a file into lines, each line containing up to 253 characters. It gives each line a number and always numbers the lines consecutively. But, even though you see these line numbers on the screen when you use **edlin**, they are not part of the file.

When you insert lines of text in a file, the line numbers after the inserted text are automatically adjusted. Similarly, when you delete lines in a file, the line numbers following the deleted text are automatically renumbered.

How to Start Edlin

To start **edlin**, type:

edlin filename

where *filename* is the file you want to edit. If you are creating a new file, *filename* should be the name or pathname of the file you want to create. If **edlin** does not find this file on the default disk drive, it creates a new file

with the name or pathname that you specify. For example, if you want to create a file called Budget.jun, type the following command and then press **ENTER** :

```
edlin budget.jun
```

Once you type the command, **edlin** displays the following:

```
New file
```

```
*  
_
```

Note that the **edlin** prompt is an asterisk (*).

To begin entering text, type an **i** (insert command). (The insert command is discussed later in this chapter.) You can now begin typing lines of text into your file. You can also use the **edlin** commands as they are introduced.

Note: Be sure to press **ENTER** at the end of each line.

Suppose you want to edit an existing file called Budget.may. To do this, type:

```
edlin budget.may
```

When **edlin** finds the Budget.may file, it loads the file into memory. If your computer has enough memory to load the entire file, **edlin** displays the following message:

```
End of input file
```

```
*
```

You can then edit the file using **edlin** commands.

If the file is too large to be loaded into memory, **edlin** loads lines from the file until memory is 3/4 full and then displays the asterisk (*) prompt. You can then edit the portion of the file that is in memory.

To edit the rest of the file, save some of the edited lines onto a disk so that you can free some memory space. **Edlin** can then load the remaining unedited lines from disk into memory. For more information on editing large files, see the **w** (write) and **a** (append) commands later in this chapter.

How to End Edlin and Save Your Changes

When you finish your editing session and the cursor is at the asterisk (*) prompt, you can save your original file and the updated (new) file by using the **e** (end) command (discussed later in this chapter). **Edlin** renames your original file with the extension **.bak** and saves the updated file with the filename and extension you gave when you started **edlin**.

Warnings:

You cannot update a file with an extension of **.bak**, because when you try to save your file, **edlin** always saves the original file as **.bak**, overwriting any existing **.bak** file. Thus, any changes made to the existing **.bak** file using **edlin** are lost. If you need to update a **.bak** file, rename it (using the MS-DOS **ren** command discussed in Chapter 3, "MS-DOS Commands") and provide another extension. You can then start **edlin** by using the new filename.

If a **.bak** file already exists and has read-only permission, you can save the updated file using the **e** (end) command, but the original file is not saved in a **.bak** file. **edlin** detects this condition when you first start to edit a file and displays the following message:

WARNING! Backup is Read-only -- backup will not be made

Special Editing Keys

To edit your text files, you can also use the template and special editing keys introduced in Chapter 5, "MS-DOS Editing and Function Keys."

The following table summarizes these commands, codes, and functions. Descriptions of these special editing keys follow the table.

Key	Purpose
F1	Copies one character from the template to the new line.
F2	Copies all characters from the template to the new line, up to the specified character.
F3	Copies all remaining characters in the template to the screen.
DEL	Does not copy (skips over) a character.
F4	Does not copy (skips over) the characters in the template, up to the specified character.
ESC	Clears the current input and leaves the template unchanged.
INS	Enters/exits insert mode.
F5	Makes the new line the new template.
←	Deletes a character from the command line, and places the cursor back one character in the template.

The following pages describe how to use each of the MS-DOS editing keys with **edlin**.

The **F1** Key

When you press **F1**, **Edlin** copies one character from the template to the current line and turns off the insert mode. To learn to use **F1** with **edlin**, type:

1:*Sharpe Office Supplies.

2:*
_

At the beginning of the editing session, the cursor (shown by the underline) is at the beginning of the line. When you press **[F1]**, **edlin** copies the first character, “S”, to Line 2:

2:*S_

Each time you press **[F1]**, one more character appears:

[F1]

2:*Sh_

[F1]

2:*Sha_

[F1]

2:*Shar_

The **[F2]** Key

When you press **[F2]**, **edlin** copies all the characters, up to the specified character, from the template to the current line. The given character is the one that you type immediately after pressing **[F2]**. **Edlin** does not copy or display the given character on the screen, but it does copy and display the characters up to the specified character. If the template does not contain the character, **edlin** does not copy anything.

When you use **[F2]**, you automatically turn off insert mode. As an example of how to use **[F2]** with **edlin**, type Sharpe Office Supplies. on Line 1. Then, press **[F2]**, and type c. **Edlin** copies the characters up to the “c” in the word “Office”.

[F2] c

2:*Sharpe Offi_

The **[F3]** Key

Pressing **[F3]** causes **edlin** to copy the characters in the template to the current line. No matter where the cursor is, **edlin** displays the rest of the line and leaves the cursor at the end of the line. To see how this works, type "Sharpe Office Supplies." on Line 1 and press **[F3]**. **Edlin** copies the characters in the template (Line 1) to the line with the cursor (Line 2):

[F3]

2:*Sharpe Office Supplies._

This command automatically turns off insert mode.

The **[DEL]** (delete) Key

When you press **[DEL]**, **edlin** skips the next template character. This action is similar to **[F1]**, except **[DEL]** skips a character in the template instead of copying it to the line. Thus, if the template already contains "Sharpe Office Supplies.", and you press **[DEL]**, **edlin** skips over the first character, "S". The cursor doesn't move as **edlin** changes the template. To see how much of the line is skipped, press **[F3]**. This action moves the cursor past the last character of the line:

[DEL] **[F3]**

2:*harpe Office Supplies._

The **[F4]** Key

When you press **[F4]**, **edlin** skips over all characters up to a given character in the template. **Edlin** does not copy characters or display any of the characters up to and including the given character. If the template does not contain that character, **edlin** does not skip any characters.

Note that **[F4]** acts similarly to **[F2]**, except **[F4]** skips over characters in the template instead of copying them to the current line.

Using the same template example, "Sharpe Office Supplies.", in Line 1, press **[F4]** and then type **c**. This causes **edlin** to skip over all the characters in the template up to the "c" in the word "Office". The cursor does not move as **edlin** changes the template.

Press **[F3]** to see how much of the line is skipped. This action displays the rest of the line and moves the cursor to the end of the line:

[F4] c

[F3]

2:*ce Supplies._

The **[ESC]** (Escape) Key

When you press **[ESC]**, **edlin** clears the current line but does not change the template. Pressing **[ESC]** also prints a backslash (\), carriage return, and linefeed, and turns off insert mode. The cursor is at the beginning of the line. If you then press **[F3]**, **edlin** copies the template to the current line, leaving the line as it was before you pressed **[ESC]**.

For example, type:

1: Sharpe Office Supplies.

2:*The World Leader_

Press **[ESC]** to cancel the current line, Line 2. Notice that a backslash appears on Line 2 to tell you it has been canceled:

[ESC]

2:*The World Leader\

Press **[ENTER]** to keep Line 1 or to perform any other editing functions. Now, if you press **[F3]**, **edlin** copies the original template to the line:

[F3]

2:Sharpe Office Supplies.

The **[INS]** (Insert) Key

Pressing **[INS]** toggles between insert mode and replace mode. When you start **edlin**, you are automatically in replace mode.

The first time you press **[INS]**, **edlin** enters insert mode. In insert mode the cursor in the template does not move, but it does move in the current line as you insert each character.

When you finish inserting and press **[INS]**, **edlin** re-enters replace mode, and the cursor is located on the same character in the template as when you entered insert mode.

The **[F5]** Key

When you press **[F5]**, **edlin** copies the current line to the template, deleting the previous contents.

Pressing **[F5]** also displays "@" and outputs a carriage return and a linefeed. Pressing **[F5]** causes **edlin** to empty the current line and turns off insert mode.

Pressing either **[F5]** or **[ESC]** perform almost the same function. However, **[F5]** changes the template and prints an @ instead of a backslash.

To see how to use **[F5]** with **edlin**, type:

```
1:*Sharpe Office Supplies.  
2:*_
```

Remember, the cursor (shown by the underline) is at the beginning of the line when you begin an editing session. Now, type the key sequences shown below. (The results are shown below each key sequence.)

```
[F2] c  
2:*Sharpe Offi_  
  
[INS] cial  
2:*Sharpe Official_  
  
[INS] Sharpeware.  
2:*Sharpe Official Sharpeware._
```

If you want to add a word at the beginning of this line, but you don't want to backspace and retype the whole line, press **[F5]** to put the current line into the template.

```
1: Sharpe Office Supplies.  
2:*Sharpe Official Sharpeware.@
```

The @ character indicates that the new line is now in the new template. To add the word “Introducing:” (followed by a space) at the beginning of the line, press **INS** and type the key sequences shown below. (The results are shown below each key sequence.)

INS Introducing:

2:*Introducing: _

Then, press **F3** to insert the contents of the template.

F3

2:*Introducing: Sharpe Official Sharpeware._

The **←** (Backspace) Key

This key deletes a character from the command line and moves the cursor one character to the left in the template.

You can also press **CTRL** **H** to backspace. For example, type:

1:*Sharpe Office Supplier_

You realize that you meant to type “Supplies” instead of “Supplier”. To correct this error, press **←** and type s.

←

1:*Sharpe Office Supplies_

Using Edlin Commands

Edlin includes several commands that help you edit lines in a file. The following table summarizes these commands:

Letter	Purpose
a	Appends lines from disk into memory
c	Copies lines
d	Deletes lines
<i>line</i>	Edits a line or lines
e	Ends editing session and saves edits
i	Inserts lines of text
L	Lists a range of lines
m	Moves a range of text to a specified line
p	Pages through a file 23 lines at a time
q	Quits the editing session without saving the file
r	Replaces text
s	Searches for text
t	Transfers the contents of another file into the file being edited
w	Writes specified lines from memory to disk

Some Tips for Using Edlin Commands

Once you start editing a file with **edlin**, you can use the **edlin** commands to edit lines of text in the file. Here are a few things to remember when using **edlin** commands:

You can use pathnames in commands. For example, by typing the following command, you can edit a file named *Report.may* in a subdirectory named *\sharpe\budget*:

```
edlin \sharpe\budget\report.may
```

You can refer to lines with numbers relative to the current line. (The current line is identified with an asterisk, *.) To indicate lines before the current line, use a minus sign with a number; to indicate lines after the current line, use a plus sign with a number. For example, to list ten lines before the current line, the current line, and ten lines after the current line, type:

-10, +10L

Note: The letter L indicates the L (list) command. This example uses an uppercase L to avoid confusion with the number 1, but you can also use a lowercase L.

Edlin ignores spaces between the line number and the command. For the examples in this chapter, spaces are omitted.

Generally, **edlin** allows you to type one command after another on the same command line. However, if you want to use the **edlin** line (edit) command to edit a specific line, you must separate the line number from the other commands with a semicolon. For example, the following command edits Line 15 and then displays Lines 10 through 20 on the screen:

15;-5, +5L

When using control key sequences, press and hold **CTRL** then press the control character (such as **Z** or **C** or **V**). For example, to search for a phrase that includes the **CTRL Z** character, use this command:

smonthly budget **CTRL Z** -5, +5L

Do not type **CTRL**. Instead, press **CTRL** as you press **Z**.

Use **CTRL V** (the quotation mark character) to insert control characters into text. **CTRL V** tells MS-DOS to recognize the next uppercase letter you type as a control character. For example, the following command finds the first occurrence of the end-of-file character, **CTRL Z**, in a file:

s **CTRL V Z**

Note: To insert **CTRL V** into the text, press **CTRL V** and then V.

The **CTRL Z** character is usually an end-of-file identifier for **edlin**. If you have **CTRL Z** characters elsewhere in your file, tell **edlin** that these other control characters are not end-of-file markers.

To tell **edlin** to ignore the **CTRL Z** characters, use the **/b** switch when you start **edlin**. For example, the following command opens the **Macro.asm** file for editing and ignores any **CTRL Z** characters:

```
edlin macro.asm /b
```

Edlin Command Options

Many **edlin** commands accept one or more options. The effect of a command option varies, depending on which command you use it with. The options are described in this section.

The Line Option

The line option is a line number that you type. Use a comma or space to separate the numbers from other line numbers, other options, and from the command. You can specify line in one of three ways:

Type	How It Works
<i>linenumber</i>	You can use any number less than 65,534. If you specify a number larger than the largest existing line number, then line refers to the line after the last line number.
period (.)	If you specify a period for line, it refers to the current line number. The current line is the last line you edited, not necessarily the line you last displayed. Edlin marks the current line with an asterisk (*) between the line number and the first character.
pound sign (#)	The pound sign indicates the line after the last line number in the file. Typing # for line is the same as typing the last line number plus one.
ENTER	If you type a command and press ENTER without any of the line markers in this list, edlin uses a default value for each command. (Default values can be different for each command.)

The Question Mark Option

The question mark (?) option tells **edlin** to ask if the correct string has been found. You use the question mark only with the **r** (replace) and **s** (search) commands. Before continuing, **edlin** waits for you to press **Y** or press **ENTER** for a Yes response or press any other key for a No response.

The Text Option

The text option specifies text to be found, to be replaced, or to replace other text. Use the text option only with the **s** (search) and **r** (replace) commands.

The remaining pages in this chapter describe the **edlin** commands. Each description includes the correct usage (syntax) of the command. Also, each command description includes several comments and examples that offer advice, help, and even some shortcuts for using **edlin**.

Edlin's Command Reference

The remaining portion of this chapter provide references to all of Edlin's commands.

Edlin's Append Command: a

Syntax:

[*n*]a

Comments:

If you are editing a file that is too large to read into memory all at once, you can use the **a** (append) command. This command lets you read portions of your file into memory as you need to work on them. The *n* parameter is the number of lines that you want to read into memory.

When you start **edlin**, it reads as many lines as possible into memory. If the size of your file exceeds available memory, you must edit your file in stages. That is, after you edit the first part of a large file, write the lines you have edited onto your disk. Then, you can load unedited lines from your disk into memory using the **a** (append) command.

Notes:

If you do not specify the number of lines to append, **edlin** reads lines into memory until memory is 3/4 full. If memory is already 3/4 full, this command does nothing. If available memory is full, you can free memory by quitting other applications that are running or by restarting MS-DOS. Restarting MS-DOS clears memory being used by programs that remain resident in memory, even after they finish running.

After the **a** (append) command reads the last line of the file into memory, **edlin** displays the message **End of input file**.

Example:

Suppose you have a file so large that the last 100 lines do not fit into memory. After editing the first part of the file and writing a portion back to the disk, use the **a** command to read the remaining 100 lines into memory:

100a

For information on writing edited lines to your disk, see the **w** (write) command later in this chapter.

Edlin's Copy Command: **c**

Syntax:

[*line*],[*line*], *line*[, *count*]c

Comments:

The **c** (copy) command copies a range of lines to a specified line number, and when used with the *count* option, the command copies this range as many times as specified.

The first and second *line* options specify the range of lines that you want to copy. If you omit the first or second *line* option, **edlin** defaults to the current line. The third *line* option specifies the line before which **edlin** is to place the copied lines.

If you overlap the line numbers, you receive an Entry error message. For example, this incorrect command produces an error message:

3,20,15c

If you do not specify a number for the *count* option, **edlin** copies the lines one time and automatically renumbers the file after the copy.

Example:

If you type:

1,5,6c

Edlin copies Lines 1 through 5 and duplicates them one time, beginning on Line 6. Thus, Lines 1 through 5 and 6 through 10 are identical. If Lines 6-10 exist before you use the copy command, these lines are moved down and renumbered.

Edlin's Delete Command: d

Syntax:

[*line*][, *line*]d

Comments:

The **d** (delete) command deletes a specified range of lines in a file. If you omit the first *line* option, **edlin** defaults to the current line (indicated by an asterisk next to the line number). If you omit the second *line* option, **edlin** deletes only the first line specified. Remember, too, that when you delete lines, **edlin** automatically rennumbers the file.

Example:

The following command deletes Line 7 and then rennumbers all the lines that follow:

```
7d
```

To delete a block of text from Lines 22 through 32, type:

```
22,32d
```

The **d** command removes Lines 22 through 32, inclusively, from your file.

Finally, suppose you want to delete a range of lines beginning with the current line, Line 7, through Line 11. Type:

```
,11d
```

Edlin's Line Edit Command

Syntax:

[*line*]

Comments:

The *line* option allows you to specify the line number of text you want to edit. When you type a line number, **edlin** displays the line number (and the text on that line) and reprints the line number on the line below. You can then retype the line or use **edlin**'s editing keys to edit it. The existing text on the line serves as the template until you press **ENTER**. If you press **ENTER** without typing a line number, **edlin** edits the line after the current line.

After you edit the line, press **ENTER** to accept the line.

Warning: If you press **ENTER** while the cursor is in the middle of a line, **edlin** deletes the remainder of the line.

Example:

Suppose that the following file exists and is ready to edit:

```
1: Dear Mr. Dimm,  
2:  
3: I was sorry to hear of your recent  
4: hospitalization due to electrical  
5: shock from our Automatic  
6: Pencil Sharpener.
```

You need to insert the product's name, X-1000, into Line 5. To edit that line, type 5. **Edlin** displays the contents of the line with the cursor below the line:

```
5:*shock from our Automatic  
5:*_
```


Press **F2** to skip to "A" in the word "Automatic." To enter the product name, press **INS**.

F2 A **INS**

X-1000

5:*shock from our X-1000

F3

5:*shock from our X-1000 Automatic

At the **edlin** prompt, type L to see the file:

1: Dear Mr. Dimm,

2:

3: I was sorry to hear of your recent

4: hospitalization due to electrical

5:*shock from our X-1000 Automatic

6: Pencil Sharpener.

Edlin's End/Save Command: **e**

Syntax:

e

Comments:

The **e** (end) command saves the edited file on your disk, renames the original input file *filename.bak*, and then exits **edlin**. If you created the file during the editing session, **edlin** does not create a backup (.bak) file.

The **e** (end) command takes no options. This means that you must select the directory and drive that you want to save the file on when you start **edlin**. If you don't select a drive when you start **edlin**, it saves the file on the disk in the default drive. However, you can still copy the file to a different drive using the MS-DOS **copy** command.

Before using the **e** command to save your file, be sure that the disk contains enough free space for the entire file. If it doesn't, **edlin** might not be able to write the entire file to the disk. When this happens, the edited file is lost, although **edlin** might have saved part of the file on the disk.

Notes:

If a .bak file already exists and is a read-only file, the updated file is saved by using the **e** (end) command. However, the original file does not detect this condition when you first start to edit a file, so MS-DOS displays this message:

Warning: Backup file is read only--
backup will not be made.

Example:

To end an **edlin** session and save the edits you made, type:

e

Edlin saves your edited file and returns you to the MS-DOS prompt (for example, **A>**).

Edlin's Insert Command: i

Syntax:

[*line*]i

Comments:

The **i (insert)** command allows you to insert text immediately before the specified line. If you are creating a new file, type the **i (insert)** command before you insert a new line of text. Text begins on Line 1, and the next line number appears automatically each time you press **ENTER**.

Edlin remains in insert mode until you press **CTRL** **C**. When you finish the insertion and exit insert mode, the line immediately following the inserted lines becomes the current line. **Edlin** automatically increments the line numbers that follow the inserted section by the number of lines that you inserted.

If you do not specify *line*, the default is the current line number, and **edlin** inserts the lines before the current line. If *line* is a number larger than the last line number or if you specify a pound sign (#) as *line*, **edlin** appends the inserted lines to the end of the file. In this case, the last line that you inserted becomes the current line.

Example:

Suppose the following file exists and is ready to edit:

```
1: Dear Mr. Dimm,  
2:  
3: I was sorry to hear of your recent  
4: hospitalization due to electrical  
5: shock from our X-1000 Automatic  
6: Pencil Sharpener.  
7:  
8: Sincerely,  
9:  
10: I.R. Sharpe, President
```

This letter doesn't really offer any compensation for the accident, so you might want to add a comforting thought to the letter to console Mr. Dimm. To insert text before Line 8, type:

8i

The result is:

8:*_

Now, type the following lines. The insertion begins on Line 8. Be sure to press **ENTER** at the end of each line.

8:*As a result of your accident, we

9:*are redesigning our manual to

10:*warn our customers against trying

11:*to sharpen metal objects.

To end the insertion, press **CTRL C** on the next line. To insert a blank line immediately before the current line (Line 12), type i. The result is:

12:*_

Insert a blank line by pressing **ENTER**, and end the insertion by pressing **CTRL C** on the next line. Then, to list the file and see the result, type L. The result is:

1: Dear Mr. Dimm,

2:

3: I was sorry to hear of your recent

4: hospitalization due to electrical

5: shock from our X-1000 Automatic

6: Pencil Sharpener.

7:

8: As a result of your accident, we

9: are redesigning our manual to

10: warn our customers against trying

11: to sharpen metal objects.

12:

13:*Sincerely,

14:

15: I.R. Sharpe, President

Edlin's List Command: L

Syntax:

[*line*][, *line*]L

Comments:

The **L** (list) command displays a range of lines, including the two lines specified. If you only specify one of the *line* options, **edlin** uses default values. For example, if you omit the first *line* option, as in the following example, **edlin** displays 23 lines, beginning 11 lines before the current line and ending with the specified line. The beginning comma shows that you omitted the first *line* option.

, *line* L

If you omit the second *line* option, **edlin** displays 23 lines, starting with the specified line. If you type **L** with no *line* option at all, **edlin** displays 23 lines, beginning 11 lines before the current line.

Notes:

If the specified line is more than 11 lines before the current line, the display is the same as if you omitted both *line* options.

An uppercase **L** is used here to avoid confusion with the number 1. You can use either an uppercase or a lowercase **L**.

Example:

To list Lines 5 through 10, inclusive, type:

5,10L

Edlin's Move Command: **m**

Syntax:

[line,][+] line, linem

Comments:

The **m** (move) command lets you transfer a block of text to another location in a file. The first and second *line* options specify the range of lines that you want to move. The third *line* option specifies the line to which you want to move the first line in the range.

Edlin automatically renumbers the lines after it moves them. For example, the following command moves 25 lines of text, beginning with the current line, to Line 100:

`, +25,100m`

If the line numbers that you specify overlap, **edlin** displays an Entry error message.

Example:

Suppose the following file exists and is ready to edit.

```
1: Dear Mr. Dimm,  
2:  
3: I was sorry to hear of your recent  
4: hospitalization due to electrical  
5: shock from our X-1000 Automatic  
6: Pencil Sharpener.  
7:  
8: As a result of your accident, we  
9: are redesigning our manual to  
10: warn our customers against trying  
11: to sharpen metal objects.  
12:  
13: Sincerely,  
14:  
15: I.R. Sharpe, President  
16: Sharpe Office Supplies  
17: The World Leader in Office Sharpeware  
18: Our motto: "You oughta be Sharpe too"
```

What if you prefer to have the motto at the start of the letter? If so, you can move Lines 16-18 to Line 1 by typing the following command:

```
16,18,1m
```

The result is:

1: Sharpe Office Supplies
2: The World Leader in Office Sharpeware
3: Our motto: "You oughta be Sharpe too"
4: Dear Mr. Dimm,
5:
6: I was sorry to hear of your recent
7: hospitalization due to electrical
8: shock from our X-1000 Automatic
9: Pencil Sharpener.
10:
11: As a result of your accident, we
12: are redesigning our manual to
13: warn our customers against trying
14: to sharpen metal objects.
15:
16: Sincerely,
17:
18: I.R. Sharpe, President

Edlin's Paging Command: p

Syntax:

[*line*][, *line*]p

Comments:

The **p** (page) command displays a file one screen (23 lines) at a time. The first *line* option specifies the first line of the display. The second *line* option specifies how many lines appear on each page. If you do not type the first *line* option, **edlin** starts the page with the line after the current line. If you do not type the second *line* option, **edlin** lists 23 lines on each page.

Example:

To view Lines 100 through 200 and to see the text one screen at a time, type:

100,200p

Edlin's Quit/No Save Command: q

Syntax:

q

Comments:

The **q** (quit) command is useful if you don't want to make any changes to a file. This command exits to the MS-DOS operating system and does not save any editing changes. If you use the **q** (quit) command, **edlin** prompts you to be sure you don't want to save the changes. If you want to save changes as you exit **edlin**, use the **e** (end) command.

Notes:

When you exit **edlin**, it erases any previous copy of the file that has a .bak extension. But, if you quit **edlin** (**q**) and reply **Y** (for Yes) to the **Abort edit (Y/N)?** message, **edlin** does not delete your previous backup copy.

Example:

The following example shows how to quit **edlin** without saving your changes:

Press **CTRL** **C** to leave insert mode.

At the asterisk (*) prompt, type **q**.

The message **Abort edit (Y/N)?** appears.

Type **Y** (for Yes) to terminate the edit, and then press **ENTER**.

Edlin's Replace Command: **r**

Syntax:

[*line*][, *line*][?]**r***text1* [CTRL][Z] *text2*

Comments:

The **r** (replace) command replaces all occurrences of a string of text in a specific range with a different string of text. The *line* options show the range that **r** (replace) uses. Each time **edlin** finds *text1*, it replaces it with *text2*. Then **edlin** displays each line that changes. For example, the following command changes the word “mine” each time it occurs in a 20 line file to the word “ours”:

1,20rmine [CTRL][Z] ours

If a line contains two or more replacements, it is displayed once for each change. If you include a question mark (?) in your command, **edlin** asks O.K.?

If you press [Y] (for Yes) or [ENTER], *text2* replaces *text1*, and **edlin** looks for the next occurrence of *text1*. If you press any other key in response, **edlin** does not make the change for that occurrence of *text1*. When **edlin** has made all the changes, the asterisk prompt reappears.

When you do not specify *text1*, the **r** command assumes the old (any previous) value. If this is the first replacement that you have made during this editing session and if you do not specify *text1*, the command ends. If you do not specify *text2*, you must end *text1* by pressing [ENTER].

If you omit the first *line* option, **edlin** uses the line after the current line, by default. The default for the second *line* option is the line following the last line of the file (represented by the symbol #). If you end *text1* with a [CTRL][Z] and do not specify *text2*, **edlin** assumes you want blank spaces for *text2*. For example, suppose you want to delete all occurrences of the word “clients” from your file. To do this, type the command shown below, and then press [CTRL][Z] and [ENTER]:

clients

The next command replaces clients with the previous *text2*:

clients

The following command makes the previous *text1* become the previous *text2*:

r

Note that “previous” refers to an earlier string of text specified in an s or r command.

Example:

Suppose the following file exists and is ready for editing:

```
1: Dear Mr. Dimm,  
2:  
3: I was sorry to hear of your recent  
4: hospitalization due to electrical  
5: shock from our X-1000 Automtic  
6: Pencil Sharpener.  
7:  
8: As a result of your accident, we  
9: are redesigning our manual to  
10: warn our customers against trying  
11: to sharpen metal objects.  
12:  
13: Sincerely,  
14:  
15: I.R. Sharpe, President
```

Now, suppose that you want to replace all occurrences of the word “our” with the word “the” in Lines 5 through 10. To do this, type 5,10 rour; press **CTRL Z**, type the, and press **ENTER**. The result is:

```
5: shock from the X-1000 Automatic  
8: As a result of ythe accident,  
9: are redesigning the manual to  
10: warn the customers against
```

In the previous example, two unwanted changes occurred in Lines 8 and 10. To avoid the changes and to confirm each replacement, use the same file with a slightly different command. The following example shows you how to replace only certain occurrences of “our” with “the”. At the **edlin** prompt, type the following and then press **ENTER** :

```
1,15? rour CTRL-Z the
```

The result is

```
5: shock from the X-1000 Automatic
O.K.? y
8: As a result of ythe accident, we
O.K.? n
9: are redesigning the manual to
O.K.? y
10: warn the customers against trying
O.K.? n
*
_
```

Type the list command, **L**, to see the result of all these changes:

```
5: shock from the X-1000 Automatic
.
.
8: As a result of your accident, we
9: are redesigning the manual to
10: warn our customers against trying
.
.
```

Edlin's Search Command: s

Syntax:

`[line][, line][?]s text`

Comments:

The **s** (search) command searches a range of lines for a string of text. The first and second *line* options specify the range of lines for **edlin** to search. You end the *text* option by pressing **ENTER**.

Edlin displays the first line that matches the string; that line then becomes the current line. Unless you type the question mark (?) option, the **s** (search) command ends when it finds the first match. If **edlin** cannot find a line with a match, it displays the message Not found.

If you include the question mark option (?), **edlin** displays the first line with matching text and prompts you with the message O.K.? If you press either **Y** (for Yes) or **ENTER**, this line becomes the current line, and the search ends. If you press any other key, the search continues until another match is found or until all lines have been searched. (The search ends when **edlin** displays the Not found message.)

If you do not type the first *line* number, **edlin** defaults to the line after the current line, and if you do not type the second *line* number, **edlin** defaults to # (the line after the last line of the file).

If you omit the *text* option, **edlin** uses the text from any previous **s** or **r** (replace) command. If this is the first **s** or **r** command you have used this session and you have not specified a search string, the **s** command ends immediately.

Example:

Suppose the following file exists and is ready for editing:

```
1: Dear Mr. Dimm,  
2:  
3: I was sorry to hear of your recent  
4: hospitalization due to electrical  
5: shock from our X-1000 Automatic  
6: Pencil Sharpener.  
7:  
8: As a result of your accident, we  
9: are redesigning our manual to  
10: warn our customers against trying  
11: to sharpen metal objects.  
12:  
13: Sincerely,  
14:  
15: I.R. Sharpe, President
```

To search for the first occurrence of the word “to”, type 2,12 sto and press **ENTER**. Edlin displays the following lines:

```
3: I was sorry to hear of your recent
```

To search through several occurrences of a string until the correct string is found, type the command 1, ? sto. The result is:

```
3: I was sorry to hear of your recent  
O.K.?_
```

If you press any key (except ☐ Y ☐ or ☐ ENTER), the search continues, so press ☐ N (for No):

O.K.? n

Continue:

4: hospitalization due to electrical

O.K.? _

Now, press ☐ Y to terminate the search:

O.K.? y

*
_

Edlin's Transfer Command: **t**

Syntax:

*[line]*t *filename*

Comments:

The **t** (transfer) command puts the contents of one file into another file, or into the text you are typing. **Edlin** inserts the filename at the line number you give in the *line* option and then automatically renumbers the lines. If you omit the line number, **edlin** inserts the text on the current line.

Example:

To copy a file named *Irsharpe.mem* to Line 12 of the file you are editing, use the following command:

```
12 t irsharpe.mem
```

Edlin's Write Command: **w**

Syntax:

[*n*]**w**

Comments:

The **w** (write) command writes a specified number of lines to disk. The *n* option specifies the number of lines that you want to write to the disk. You need this command only if the file you are editing is too large to fit into memory. When you start **edlin**, it reads lines from your file until memory is 3/4 full.

To edit the remainder of your file, you must write the edited lines in memory onto your disk. Then you can load additional unedited lines from your disk into memory by using the **a** (append) command, which is described earlier in this chapter.

Example:

Suppose you have a file so big that the last 100 lines do not fit into memory. After you edit the first part of the file, free up enough space to edit the last part of the file with this command:

125**w**

Note: If you do not specify the number of lines for **edlin** to write, it writes lines until memory is 3/4 full. But, it does not write any lines to your disk until memory is more than 3/4 full. Also, **edlin** renumbers all of the lines so that the first remaining line becomes Line 1.

Debug

In this chapter, you will learn:

- How to start the **debug** utility
- How to use the **debug** commands and parameters

Note: You need to read this chapter only if you are writing or testing executable programs (.com or .exe files) or object (.obj) files.

Introduction

The **debug** utility is a debugging program that provides a controlled testing environment for binary and executable object files. Note that **edlin**, the MS-DOS line editor, is used to alter source files; **debug** is **edlin**'s counterpart for binary files.

Debug eliminates the need to reassemble a program to see if a problem has been fixed by a minor change. It allows you to alter the contents of a file or the contents of a CPU register, and then immediately re-execute a program to check the validity of the changes made.

Terminate **debug** commands by pressing **CTRL** **C**.

CTRL **S** stops screen scrolling. To restart scrolling, press a key other than **CTRL** **C** or **CTRL** **S**. These commands are consistent with the control character functions available at the MS-DOS command level.

How to Start Debug

Debug can be started two ways. By the first method, you type all commands in response to the **debug** prompt (-). By the second method, you type all commands on the line used to start **debug**.

Method 1: Debug

To start **debug** using Method 1, type:

```
debug
```

Debug responds with the hyphen (-) prompt, signaling that it is ready to accept your commands. Since you didn't specify a filename, you can use other commands to work on current memory, disk sectors, or disk files.

Warning: When **debug** (version 2.0) is started, it sets up a program header at offset 0 in the program work area. In previous versions of **debug**, you can overwrite this header. You can still overwrite the default header if you don't give a filename to **debug**. If you are debugging a .com or .exe file, however, do not tamper with the program header below address 5CH, or **debug** will terminate.

Do not restart a program after the following message is displayed:

```
Program terminated normally
```

You must reload the program with the **n** (name) and **L** (load) commands for it to run properly.

Method 2: Command Line

To start **debug** using a command line, you must use the following syntax:

```
debug [filename [arglist]]
```

For example, if you specify a filename, the following is a typical command to start **debug**:

```
debug file.exe
```

Debug loads file.exe into memory starting at 100 (hexadecimal) in the lowest available segment. The BX:CX registers are loaded with the number of bytes placed into memory.

If you do include a *filename*, you might also specify an *arglist*. An *arglist* is a list of filename parameters and switches that are to be passed to the program *filename*. So when *filename* is loaded into memory, it is loaded as if it had been started with a command of the form, **debug***filename arglist*.

Here, *filename* is the file to be debugged, and *arglist* is the rest of the command line used when **debug** calls and loads *filename* into memory.

Debug Command Information

Each **debug** command consists of a single letter followed by one or more parameters. Additionally, the control characters and special editing functions described in Chapter 5, “MS-DOS Editing and Function Keys,” apply to **debug** as well.

If a syntax error occurs in a **debug** command, **debug** reprints the command line and indicates the error with a caret (^) and the word “Error” as in the following example:

```
dcs:100 cs:110
  ^ Error
```

Note that when typing commands and parameters you can use any combination of uppercase and lowercase letters. The **debug** commands are listed below. Following this list, the commands and their parameters are described in greater detail.

Debug Command	Function
a [<i>address</i>]	Assemble
c <i>range address</i>	Compare
d [<i>range</i>]	Dump
e <i>address [list]</i>	Enter
f <i>range list</i>	Fill
g [= <i>address [address...]</i>]	Go
h <i>value value</i>	Hex
i <i>value</i>	Input
L [<i>address [drive:record record]</i>]	Load
m <i>range address</i>	Move
n <i>filename [filename]</i>	Name
o <i>value byte</i>	Output
q	Quit
r [<i>register-name</i>]	Register
s <i>range list</i>	Search
t [= <i>address</i>] [<i>value</i>]	Trace
u [<i>range</i>]	Unassemble
w [<i>address [drive:record record]</i>]	Write

Debug Command Parameters

All **debug** commands accept parameters, except the **q** (quit) command. Parameters can be separated by delimiters (spaces or commas), but a delimiter is required only between two consecutive hexadecimal values. Thus, the following commands are equivalent:

```
dcs:100 110
d cs:100 110
d,cs:100,110
```

Parameter	Definition
<i>drive:</i>	A one-digit hexadecimal value that indicates which drive a file will be loaded from or written to. The valid values are 0-3, where 0 = A:, 1 = B:, 2 = C:, 3 = D:.
<i>byte</i>	A two-digit hexadecimal value placed in or read from an address or register.
<i>record</i>	One-digit to three-digit hexadecimal value that indicates the logical record number on the disk and the number of disk sectors to be written or loaded. Logical records correspond to sectors; however, since they represent the entire disk space, their numbering differs.
<i>value</i>	A hexadecimal value of up to four digits that specifies a port number or the number of times a command should repeat s functions.
<i>address</i>	<p>A two-part designation containing either an alphabetic segment register or a four-digit segment address plus an offset value. You can omit the segment name or segment address, in which case the default segment DS is used for all commands except g, l, t, u, and w, for which the default segment is CS. All numeric values are hexadecimal.</p> <p>Following is an example address:</p> <p style="margin-left: 40px;">CS:0100 04BA:0100</p> <p>Note that the colon is required between the segment name (whether numeric or alphabetic) and the offset value.</p>
<i>range</i>	Contains two addresses, for example address address or one address, an L, and a value (<i>address L value</i> where <i>value</i> is the number of lines on which the command should operate (L80 is assumed)). The second type of range cannot be used if another hexadecimal value follows, since the hexadecimal value is interpreted as the second address of the range.

Here are some example ranges:

```
CS:100 110
CS:100 L 10
CS:100
```

The following example, however, is illegal:

```
CS:100 CS:110
    ^ Error
```

The limit for *range* is 10000 (hexadecimal). To specify a value of 10000 with only four digits, type 0000 (or 0).

list

A series of byte values or strings. List must be the last parameter on the command line.

Following is an example list:

```
fcs:100 42 45 52 54 41
```

string

Any number of characters enclosed in quotation marks. The quotation marks can be either single (') or double ("). If the delimiter marks must appear within a string, you must use the double quotation marks. For example, the following strings are legal:

```
"This 'string' is okay."
"This ""string"" is okay."
```

However, this string is illegal:

```
""This "string" is not okay.""
```

Note that the double quotation marks are not necessary in the following strings:

```
"This "string" is not necessary."
'This ""string"" is not necessary.'
```

The ASCII values of the characters in the string are used as a list of byte values.

Assemble

Purpose:

Assembles 8086/8087/8088 mnemonics directly into memory.

Syntax:

A[*address*]

Comments:

If it finds a syntax error, **debug** responds with the message below and then redisplay the current assembly address:

^ Error

All numeric values are hexadecimal, and you must type them as 14 characters. Also, you must specify prefix mnemonics in front of the opcode to which they refer. You can, however, type them on a separate line.

The segment override mnemonics are CS:, DS:, ES:, and SS:. The mnemonic for the far return is RETF. String manipulation mnemonics must explicitly state the string size. For example, use MOVSW to move word strings, and use MOVSB to move byte strings.

The assembler automatically assembles short, near, or far jumps and calls, depending on byte displacement, to the destination address. You can override these jumps and calls by using a NEAR or FAR prefix, as in the following example:

```
0100:0500 JMP 502           ; a 2-byte short jump
0100:0502 JMP NEAR 505      ; a 3-byte near jump
0100:505 JMP FAR 50A        ; a 5-byte far jump
```

You can abbreviate the NEAR prefix to NE, but the FAR prefix cannot be abbreviated.

Debug cannot tell whether some operands refer to a word memory location or to a byte memory location. In this case, the data type must be explicitly stated with the prefix, WORD PTR or BYTE PTR. Acceptable abbreviations are WO and BY. For example:

```
NEG    BYTE PTR [128]
DEC    WO [SI]
```

Debug also cannot tell whether an operand refers to a memory location or to an immediate operand. So it uses the common convention that operands enclosed in square brackets refer to memory. For example:

```
MOV AX,21          ; Load AX with 21H
MOV AX,[21]         ; Load AX with the
                    ; contents
                    ; of memory location 21H
```

Two popular pseudo-instructions are available with the **a** (assemble) command: the DB opcode, which assembles byte values directly into memory; and the DW opcode, which assembles word values directly into memory. Following are examples of both:

```
DB      1,2,3,4,"THIS IS AN EXAMPLE"
DB      "THIS IS A QUOTATION MARK: '"
DB      "THIS IS A QUOTATION MARK: '"
DW      1000,2000,3000,"BACH"
```

The **a** command supports all forms of register indirect commands. For example:

```
ADD     BX,34[BP + 2].[SI-1]
POP     [BP + DI]
PUSH    [SI]
```

All opcode synonyms are also supported, as in the next example:

```
LOOPZ   100
LOOPE   100

JA       200
JNBE    200
```

For 8087 opcodes, the **WAIT** or **FWAIT** prefixes must be explicitly specified, as in these last examples:

```
FWAIT FADD ST,ST(3)      ; This line assembles  
                          ; an FWAIT prefix  
LD TBYTE PTR [BX]       ; This line does not
```

Compare

Purpose:

Compares the portion of memory specified by *range* to a portion of the same size beginning at the specified address.

Syntax:

C range address

Comments:

If the two areas of memory are identical, there is no display, and **debug** returns with the MS-DOS prompt. If there are differences, they are displayed in this format:

address1 byte1 byte2 address2

Example:

The following commands have the same effect:

C100,1FF 300

or:

C100L100 300

Each command compares the block of memory from 100 to 1FFH with the block of memory from 300 to 3FFH.

Dump

Purpose:

Displays the contents of the specified range of memory.

Syntax:

D[*range*]

Comments:

If you specify a range of addresses with the **d** (dump) command, the contents of the range are displayed. If you don't use parameters with the **d** command, 128 bytes are displayed at the first address (DS:100) after the address displayed by the previous **d** command. The dump is displayed in two portions: a hexadecimal dump (each byte shown in hexadecimal value) and an ASCII dump (bytes shown in ASCII characters).

Nonprinting characters are denoted by a period (.) in the ASCII portion of the display. Each display line shows 16 bytes, with a hyphen between the eighth and ninth bytes. Each displayed line begins on a 16-byte boundary. At times in this chapter, displays are split to fit them on the page.

Example:

Type the command:

```
dcS:100 110
```

Debug displays the dump in the following format:

```
04BA:0100 42 45 52 54 41 ... 4E 44 TOM SAWYER
```

If you type the **d** command, the display is formatted as described. Each line of the display begins with an address incremented by 16 from the address on the previous line.

Each subsequent **d** (typed without parameters) displays the bytes immediately following those last displayed. If you type the following command, the display is formatted as described above, but 20H bytes are displayed:

```
DCS:100 L 20
```

If you then type the following command, the display is formatted as described above, but all the bytes in the range of lines from 100H to 115H in the CS segment are displayed:

DCS:100 115

Enter

Purpose:

Enters byte values into memory at the specified address.

Syntax:

E *address*[*list*]

Comments:

If, when using the **e** (enter) command, you type the optional list of values, the byte values are replaced automatically. (If an error occurs, no byte values are changed.)

If you type the address without the optional list, **debug** displays the address and its contents, repeats the address on the next line, and then waits for your input. At this point, the **e** (enter) command waits for you to perform one of the following actions:

Replace a byte value with a value you type. Type the value after the current value. If the one you type is not a legal hexadecimal value or if it contains more than two digits, the illegal or extra character is not echoed.

Press the space bar to advance to the next byte. To change the value, type the new value as described in the previous action. If when you press the space bar you move beyond an 8-byte boundary, **debug** starts a new display line with the address displayed at the beginning.

Type a hyphen (-) to return to the preceding byte. If you decide to change a byte behind the current position, typing the hyphen returns the current position to the previous byte. When you type the hyphen, a new line is started with its address and byte value displayed.

Press **ENTER** to terminate the **e** command. You can do so at any byte position.

Example:

Suppose you type the following command:

```
ECS:100
```

Now suppose that **debug** displays the following:

```
04BA:0100 EB._
```

To change this value to 41, type the number 41 at the cursor:

```
04BA:0100 EB.41_
```

To step through the subsequent bytes, press the space bar until you see the following:

```
04BA:0100 EB.41 10. 00. BC._
```

To change BC to the number 42, for instance, type the number at the cursor, as follows:

```
04BA:0100 EB.41 10. 00. BC.42_
```

Notice that the value 10 should be 6F. To correct this value, type the hyphen as many times as needed to return to Byte 0101 (value 10), and then replace 10 with 6F:

```
04BA:0100 EB.41 10. 00. BC.42-
```

```
04BA:0102 00.-_
```

```
04BA:0101 10.6F_
```

Pressing **ENTER** ends the **e** command and returns you to the **debug** command level.

Fill

Purpose:

Fills the addresses in the specified range with the values in the specified list.

Syntax:

F range list

Comments:

If the range contains more bytes than the number of values in the list, the list is used repeatedly until all bytes in the range are filled. If the list contains more values than the number of bytes in the range, the extra values in the list are ignored. If any of the memory in the range is not valid (bad or nonexistent), the error occurs in all succeeding locations.

Example:

Suppose you type the following command:

```
F04BA:100 L 100 42 45 52 54 41
```

In response, **debug** fills memory locations 04BA:100 through 04BA:1FF with the bytes specified. The five values are then repeated until all the 100H bytes are filled.

Go

Purpose:

Executes the program currently in memory.

Syntax:

G[= *address* [*addresses*]]

Comments:

If you type the **g** (go) command by itself, the program currently in memory executes as if it had run outside **debug**. If you set = *address*, execution of the **g** command begins at the address specified.

The equal sign (=) is required so that **debug** can distinguish the start = *address* from the breakpoint *addresses*.

With the other optional addresses set, execution stops at the first address encountered, regardless of that address's position in the list of addresses that halt execution or program branching. When program execution reaches a breakpoint, the registers, flags, and decoded instruction are displayed for the last instruction executed. The result is the same as if you had typed the **r** (register) command for the breakpoint address.

You can set up to ten breakpoints, but only at addresses containing the first byte of an 8086 opcode. If you set more than ten breakpoints, **debug** returns the BP error message.

The user stack pointer must be valid and must have 6 bytes available for this command. The **g** command uses an IRET instruction to cause a jump to the program under test. The user stack pointer is set, and the user flags, Code Segment register, and Instruction Pointer are pushed on the user stack. (If the user stack is not valid or is too small, the operating system can crash.) An interrupt code (0CCH) is placed at the specified breakpoint address(es).

When **debug** encounters an instruction with the breakpoint code, it restores all breakpoint addresses to their original instructions. If you don't halt execution at one of the breakpoints, the interrupt codes are not replaced with the original instructions.

Example:

Suppose you type the following command:

GCS:7550

The program currently in memory executes up to the address 7550 in the CS segment. **Debug** then displays registers and flags, after which the **g** command terminates.

When **debug** encounters a breakpoint and you type the **g** command again, the program runs as if you had typed the filename at the MS-DOS command level. The only difference is that program execution begins at the instruction after the breakpoint, rather than at the usual start address.

Hex

Purpose:

Performs hexadecimal arithmetic on the two specified parameters.

Syntax:

Hvalue value

Comments:

Debug adds the two parameters and then subtracts the second parameter from the first. The results are displayed on one line — first the sum, then the difference.

Example:

Suppose you type the following command:

H19F 10A

In response, **debug** performs the calculations and then displays the following result:

02A9 0095

Input

Purpose:

Inputs and displays one byte from the port specified by value.

Syntax:

Ivalue

Comment:

A 16-bit port address is allowed.

Example:

Suppose you type the following command:

`I2F8`

Suppose also that the byte at the port is 42H. **Debug** inputs the byte and then displays the following:

`42`

Load

Purpose:

Loads a file into memory.

Syntax:

L[*address* [*drive: record record*]]

Comments:

Set BX:CX to the number of bytes read. The file must be named either when you start **debug** or with the **n** (name) command. Both the **debug** invocation and the **n** command format a filename properly in the normal format of a File Control Block at CS:5C.

If you use the **L** (load) command without any parameters, **debug** loads the file into memory beginning at address CS:100 and sets BX:CX to the number of bytes loaded.

If you type the **L** command with an address parameter, loading begins at the memory location specified by the address. If you use the **L** command with all parameters included, absolute disk sectors are loaded instead of a file.

Each record is taken from the specified drive. (The drive name is numeric: 0 = A:, 1 = B:, 2 = C:, etc.) **Debug** begins loading with the first specified record and continues until the number of sectors in the second record have been loaded.

Example:

Suppose you type the following commands after starting **debug**:

```
-NFILE.COM
```

Now, to load File.com, type the **L** command.

Debug then loads the file and displays the **debug** prompt. Suppose now that you want to load only portions of a file or certain records from a disk. To do this, type:

```
L04BA:100 2 0F 6D
```

Debug loads 109 (6D hex) records, beginning with logical record number 15, into memory beginning at address 04BA:0100. Then once it loads the records, **debug** returns the hyphen (-) prompt.

If the file has an .exe extension, it is relocated to the load address specified in the header of the .exe file. The address parameter is always ignored for .exe files. The header itself is stripped off the .exe file before it is loaded into memory. So, the size of an .exe file on disk differs from its size in memory.

If you name the file as a .hex file either using the **n** (name) command or when you start **debug**, typing the **L** command with no parameters causes **debug** to load the file beginning at the address specified in the .hex file. If the **L** command includes the option *address*, **debug** adds the address specified in the **L** command to the address found in the .hex file to determine the start address at which to load the file.

Move

Purpose:

Moves the block of memory specified by range to the location beginning at the specified address.

Syntax:

M range address

Comments:

Overlapping moves (where part of the block overlaps some of the current addresses) are always performed without loss of data. Addresses that can be overwritten are moved first. For moves from higher to lower addresses, the sequence of events is to first move the data beginning at the block's lowest address and then work toward the highest.

For moves from lower to higher addresses, the sequence is to first move the data beginning at the block's highest address and then work toward the lowest.

Note that if the addresses in the block being moved do not have new data written to them, the data in the block before the move remains. The **m** (move) command copies the data from one area into another, in the sequence described, and writes over the new addresses. This action is why the sequence of the move is important.

Example:

Suppose you type the following command:

```
MCS:100 110 CS:500
```

In response, **debug** first moves address CS:110 to CS:510, then CS:10F to CS:50F, and so on until CS:100 is moved to CS:500. To review the results of the move, type the **d** (dump) command using the same address you used with the **m** command.

Name

Purpose:

Sets filenames.

Syntax:

Nfilename [filename...]

Comments:

The **n** (name) command performs two functions. First, it assigns a filename for a later **L** (load) or **w** (write) command. So if you start **debug** without naming a file, you must type the command *n filename* before a file can be loaded. Second, the **n** command assigns filename parameters to the file being debugged. In this case, **n** accepts a list of parameters used by the file being debugged.

Note that these two functions overlap. Consider, for example, the following set of **debug** commands:

```
-NFILE1.EXE  
-L  
-G
```

The **n** command uses these commands to perform the following steps:

1. It first assigns the filename `File1.exe` to the file to be used in any later **L** or **w** commands.
2. It also assigns the `File1.exe` filename to the first filename parameter used by any program that is later debugged.
3. The **L** command then loads `File1.exe`.
4. The **g** (go) command causes `File1.exe` to run with `File1.exe` as the single filename parameter. (`File1.exe` runs as if you had typed `File1.exe` at the command level.)

A more useful chain of commands might look like this:

```
-NFILE1.EXE  
-L  
-NFILE2.DAT FILE3.DAT  
-G
```

In this example, the **n** command sets File1.exe as the filename for the subsequent **L** command, which loads File1.exe into memory. The **n** command is then used again, this time to specify the parameters to be used by File1.exe. Finally, when the **g** command is run, File1.exe is executed as if file1 File2.dat File3.dat had been typed at the MS-DOS command level.

Note that if you were to execute a **w** command now, File1.exe, the file being debugged, would be saved with the name File2.dat. To avoid this kind of result, always execute an **n** command before either an **L** or **w** command. There are four regions of memory that can be affected by the **n** command:

Memory Location	Contents
CS:5C	FCB for File 1
CS:6C	FCB for File 2
CS:80	Count of characters
CS:81	All characters typed

The first filename parameter that you specify for the **n** command has a file control block (FCB) set up at CS:5C. If you name a second filename parameter, an FCB is set up for this parameter beginning at CS:6C. The number of characters typed in the **n** command (exclusive of the first character, N) is given at location CS:80.

The actual stream of characters given by the **n** command (again, exclusive of the letter N) begins at CS:81. Note that this stream of characters can contain switches and delimiters that are legal in any command typed at the MS-DOS command level.

Example:

A typical use of the **n** command is as follows:

```
DEBUG PROG.COM  
-NPARAM1 PARAM2/C  
-G  
-
```

In this case, the **g** command executes the file in memory as if you had typed the following command line:

```
PROG PARAM1 PARAM2/C
```

Testing and debugging therefore reflect a normal run-time environment for Prog.com.

Output

Purpose:

Sends the specified byte to the output port specified by *value*.

Syntax:

Ovalue byte

Comment:

A 16-bit port address is allowed.

Example:

Suppose you want **debug** to output the byte value 4F to output port 2F8. To do this, type:

O2F8 4F

Quit

Purpose:

Terminates the **debug** utility.

Syntax:

Q

Comments:

The **q** (quit) command takes no parameters and exits **debug** without saving the file you're currently working with. You return to the MS-DOS command level.

Example:

To end the debugging session, type the following and press **ENTER**.

q

Debug terminates, and control returns to the MS-DOS command level.

Register

Purpose:

Displays the contents of one or more CPU registers.

Syntax:

R[*register-name*]

Comments:

If you do not type a register-name, the **r** (register) command dumps the register storage area and displays the contents of all registers and flags.

If you do type a register-name, Debug displays in hexadecimal the 16-bit value of that register. A colon appears as a prompt. Now, either type a value to change the register or press **ENTER** if you don't want a change.

Following is a list of the valid register-names:

AX	BP	SS	
BX	SI	CS	
CX	DI	IP	(IP and PC both refer
DX	DS	PC	to the Instruction Pointer.)
SP	ES	F	

Any other entry for register-name results in a **BR** error message.

When you type **f** as the register-name, **debug** displays each flag with a two-character alphabetic code. Change any flag by typing the opposite two-letter code. The flags are then either set or cleared.

The flags are listed below with their codes for SET and CLEAR:

Flag Name	Set	Clear
Overflow	OV	NV
Direction	DN (Decrement)	UP (Increment)
Interrupt	EI (Enabled)	DI (Disabled)
Sign	NG (Negative)	PL (Plus)
Zero	ZR	NZ
Auxiliary Carry	AC	NA
Parity	PE (Even)	PO (Odd)
Carry	CY	NC

When you type the **rf** command, the flags are displayed (in the order shown in the previous table) in a row at the beginning of a line. At the end of the list of flags, **debug** displays a hyphen (-).

You can enter new flag values in any order as alphabetic pairs. You do not have to leave spaces between these values. To exit the **r** command, press **ENTER**. Any flags for which you do not specify new values remain unchanged.

If you type more than one value for a flag, **debug** returns a DF error message. If you enter a flag code other than one of those shown in the table above, **debug** returns a BF error message. In both cases, the flags up to the error in the list are changed; those flags at and after the error are not.

When you start **debug**, the segment registers are set to the bottom of free memory, the instruction pointer is set to 0100H, all flags are cleared, and the remaining registers are set to zero.

Example:

If you type the following command, **debug** displays all registers, flags, and the Register decoded instruction for the current location:

R

If the location is CS:11A, for example, the display will look similar to this:

```
AX=0E00 BX=00FF CX=0007 DX=01FF SP=039D BP=0000
SI=005C DI=0000 DS=04BA ES=04BA SS=04BA CS=04BA
IP=011A NV UP DI NG NZ AC PE NC
04BA:011A          CD21          INT    21
```

If you then type the following command, **debug** displays these flags:

```
RF
NV UP DI NG NZ AC PE NC - _
```

Now, you can type any valid flag designation, in any order, with or without spaces. For example:

```
NV UP DI NG NZ AC PE NC - PLEICY
```

In response, **debug** displays the **debug** prompt. To see the changes, type either the **r** or **rf** command:

```
RF
NV UP EI PL NZ AC PE CY - _
```

Press **ENTER** to leave the flags this way or to specify different flag values.

Search

Purpose:

Searches the specified range for the specified list of bytes.

Syntax:

S range list

Comments:

The list can contain one or more bytes, each separated by a space or comma. If the list contains more than one byte, only the first address of the byte string is returned. If the list contains only one byte, all addresses of the byte in the range are displayed.

Example:

Suppose you type the following command:

SCS:100 110 41

Debug displays:

04BA:0104

04BA:010D

-type:

Trace

Purpose:

Executes one instruction and displays the contents of all registers, flags, and the decoded instruction.

Syntax:

T[=*address*] [*value*]

Comments:

If you include the =*address* option in the **t** (trace) command, tracing occurs at the specified =*address*. The *value* option causes **debug** to execute and trace the number of steps specified by *value*.

The **t** command uses the hardware trace mode of the 8086 or 8088 microprocessor. Consequently, you can also trace instructions stored in ROM (Read Only Memory).

Example:

Suppose you type the following command:

T

In response, **debug** returns a display of the registers, flags, and decoded instruction for that one instruction. Assuming, for this example, that the current position is 04BA:011A, **debug** returns the following display:

```
AX=0E00 BX=00FF CX=0007 DX=01FF SP=039D
BP=0000
SI=005C DI=0000 DS=04BA ES=04BA SS=04BA CS=04BA
IP=011A NV UP DI NG NZ AC PE NC
04BA:011A      CD21      INT      21
```

If you type the following command, **debug** executes sixteen (10 hex) instructions beginning at 011A in the current segment, and then displays all registers and flags for each instruction as it is executed.

The display scrolls away until the last instruction is executed, and then stops. Now you can see the register and flag values for the last few instructions performed:

T=011A 10

Remember that you can press **CTRL S** at any time to stop the display from scrolling. This lets you study the registers and flags for any instruction.

Unassemble

Purpose:

Disassembles bytes and displays the source statements that correspond to them, with addresses and byte values.

Syntax:

U[range]

Comments:

The display of disassembled code looks like a listing for an assembled file. If you type the **u** (unassemble) command without parameters, 20 hexadecimal bytes are disassembled at the first address after that displayed by the previous **u** command. If you type the **u** command and include the range parameter, **debug** disassembles all bytes in range. But if you specify range only as an address, 20H bytes are disassembled.

Example:

Suppose you type the following command:

U04BA:100 L10

In response, **debug** disassembles 16 bytes, beginning at address 04BA:0100:

04BA:0100	206472	AND	[SI + 72],AH
04BA:0103	69	DB	69
04BA:0104	7665	JBE	016B
04BA:0106	207370	AND	[BP + DI + 70],DH
04BA:0109	65	DB	65
04BA:010A	63	DB	63
04BA:010B	69	DB	69
04BA:010C	66	DB	6
04BA:010D	69	DB	69
04BA:010E	63	DB	63
04BA:010F	61	DB	61

Now, suppose you type:

U04ba:0100 0108

The display shows:

```
04BA:0100  206472  AND  [SI + 72],AH
04BA:0103  69 DB    69
04BA:0104  7665    JBE  016B
04BA:0106  207370  AND  [BP + DI + 70],DH
```

If the bytes in some addresses are altered, the disassembler alters the instruction statements. You can then type the **u** command for the changed locations, for the new instructions viewed, and for the disassembled code used to edit the source file.

Write

Purpose:

Writes the file being debugged to a disk file.

Syntax:

W[*address* [*drive: record record*]]

Comments:

If you do not use parameters with the **w** (write) command, **BX:CX** must already be set to the number of bytes to be written; the file is written beginning from **CS:100**. If you type the **w** command with just an address, the file is written beginning at that address. If you have used a **g** (go) or **t** (trace) command, you must reset **BX:CX** before using the **w** command without parameters.

Note that if a file is loaded and modified, the name, length, and starting address are all set correctly to save the modified file (as long as the length has not changed).

You must name the file either with the initial **debug** startup command or with the **n** (name) command. Both the **debug** startup command and the **n** command properly format a filename in the normal format of a File Control Block at **CS:5C**.

If you include parameters when you use the **w** command, the write begins from the memory address specified, and the file is written to the specified drive. (The drive name is numeric: 0 = A:, 1 = B:, 2 = C:, etc.) **Debug** writes the file beginning at the logical record number specified by the first record. **Debug** then continues to write the file until the number of sectors specified in the second record have been written.

Warning: Writing to absolute sectors is extremely risky because the process bypasses the file handler.

Example:

If you type the following command, **debug** writes out the contents of memory to the disk in Drive B, beginning with the address CS:100. The data written out starts in the disk logical record number 37H and consists of 2BH records.

WCS:100 1 37 2B

When the write is complete, **debug** displays its prompt again.

Debug Error Messages

During a **debug** session, you might receive any of the following error messages. Each error ends the current **debug** command but does not end **debug** itself.

Error Code	Definition
BF	Bad flag. You attempted to change a flag, but the characters you typed were not one of the acceptable pairs of flag values. See the r (register) command for the list of acceptable flag entries.
BP	Too many breakpoints. You specified more than ten breakpoints as parameters to the g (go) command. Retype the g command using ten or fewer breakpoints.
BR	Bad register. While using the r command, you typed an invalid register name. See the r command for the list of valid register names.
DF	Double flag. You typed two values for one flag. You can specify a flag value only once per rf command.

Diskette Support

Types of Drives

MS-DOS 3.3 supports both 1.44-megabyte (high-capacity) and 720-kilobyte (standard) 3 1/2-inch diskette drives. You can use 720K diskettes in both drives. However, you can use 1.44M diskettes only in a 1.44M drive.

In most cases, you access diskettes in these drives in the same manner you access 5 1/4-inch drives. However, when using the **format** command to format a 720K diskette in a high-capacity drive, you must use the **/n:nn** and **/t:nn** switches.

/N:nn specifies the number of sectors-per-track that **format** uses to format a standard diskette in a high-capacity drive. Always use the value 9 in place of the variable *nn*.

/T:nn specifies the number of tracks that **format** uses to format a standard diskette in a high-capacity drive. Always use the value 80 in place of the variable *nn*.

For instance, to format a 720K diskette in a high-capacity Drive A, type:

format a: /n:9 /t:80

The following shows the **format** switches you can use with the various types of diskette drives and diskettes:

Drive Type	Diskette Type	Valid Switches
5 1/4-inch double-sided	5 1/4-inch double-sided	/1 /8 /b /v /s
5 1/4-inch high-capacity	5 1/4-inch double-sided	/4 /1 /8 /b /v /s
5 1/4-inch high-capacity	5 1/4-inch high-density	/v /s
3 1/2-inch standard	3 1/2-inch standard	/v /s
3 1/2-inch high-capacity	3 1/2-inch high-density	/v /s
3 1/2-inch high-capacity	3 1/2-inch standard	/v /s /n /t
hard disk		/v /s

Using Only One Diskette Drive

If you have only one floppy disk drive, you can still type MS-DOS commands as if you had two disk drives on your system.

To do so, think of your one-drive system as having two drives (Drive A and Drive B). But instead of A and B representing two physical drives, they represent disks. Remember that when you specify Drive B when the Drive A disk was used last, MS-DOS prompts you to insert the disk for Drive B. For example:

```
A> copy command.com b:
Insert diskette for drive B:
and strike any key when ready
1 File(s) copied
A> _
```

If you specify Drive A when the Drive B disk was used last, MS-DOS asks you to change disks again, this time prompting you to insert the Drive A disk.

When using a batch file to execute commands, you follow the same procedure. MS-DOS waits for you to insert the appropriate disk and press any key before it continues.

Important: The letter displayed in the system prompt represents the default drive where MS-DOS looks to find a file when you do not specify a drive name; this letter does not represent the last disk used.

Assume, for example, that A is the default drive. If the last command performed was `dir b:`, MS-DOS would act as if the Drive B disk is still in the drive. The system prompt, however, is still `A>`, because A is still the default drive. If you type the `dir` command, MS-DOS prompts you for the Drive A disk, because it is the default drive and you did not specify another drive in the `dir` command.

How to Configure Your System

What is a Configuration File?

The configuration file `Config.sys` is a file that contains certain commands that MS-DOS checks when you start up your computer. Each time you start MS-DOS, it searches the root directory of the drive in which it was started for a file named `Config.sys`.

The `Config.sys` file lets you change your system's default configuration settings. For example, you can add installable device drivers to your system by including special commands in your `Config.sys` file.

You can use the `dir` command to see if the `Config.sys` file is already on your disk. If it is on the disk, you can use the `type` command to display it. For more information about the `dir` and `type` commands, see Chapter 3, "MS-DOS Commands."

If `Config.sys` already exists, you may want to include other commands in the file. For example, you may want to configure for a new device, such as a mouse or an external drive.

If you do not have a `Config.sys` file on your MS-DOS disk, you can use the MS-DOS line editor, `edlin`, to create one and then save it on the MS-DOS disk in your root directory. If `Config.sys` already exists and you want to change it, you can use `edlin` to edit it.

Config.sys Commands

The following table briefly describes the purpose of each configuration command:

Command	Purpose
break	Sets CTRL-C check.
buffers	Sets the number of sector buffers.
country	Allows for international time, date, and currency.
device	Installs the device driver in the system.
drivparm	Defines parameters for block devices.
fcbs	Specifies the number of FCBs that can be open concurrently.
files	Sets the number of open files that can access certain MS-DOS system calls.
lastdrive	Sets the maximum number of drives you may access.
shell	Begins execution of the shell from a specific file (usually Command.com).
stacks	Supports the dynamic use of data stacks.

Sample Config.sys File

A typical configuration file might look like this:

```
buffers = 30
device = c:\dev\network.sys
break = on
shell = c:\bin\command.com c:\bin /p
lastdrive = z
```

The following explains how this sample Config.sys file configures the system:

Command	Purpose
buffers	Sets the number of buffers in memory to 30.
device	Sets a searchpath of c:\dev\network.sys to find the device driver being added to the system. In this case, it is network software. Generally, you will receive a disk with some installation software when you purchase a new device. Make sure that the device driver is in the directory that you specify in the device command.
break	Makes it possible to use CTRL C to stop a program, depending on the program that you are running.
shell	<p>Sets the MS-DOS command processor to the Command.com file located in the \bin directory on the disk in Drive C.</p> <p>a:\bin tells the command processor where to look for Command.com if it needs to reread the disk.</p> <p>The /p switch is used by the Command.com shell. For more information about Command.com and its switches, see Chapter 3, "MS-DOS Commands."</p>
lastdrive	Sets to z the last available label for a logical or physical drive. In other words, on this computer, letters from a to z are available as labels for logical or physical disk drives.

The following pages describe each configuration command in detail.

Break

Purpose:

Sets CTRL-C check.

Syntax:

`break = on`

or

`break = off`

Default:

`break = on`

Comments:

Some programs let you use `CTRL C` to stop an activity (for example, to stop sorting a file). Normally, MS-DOS checks to see whether you have pressed `CTRL C` only while it is reading from the keyboard or writing to the screen or printer. Therefore, setting **break** to on extends CTRL-C checking to other functions, such as disk reading and writing.

Example:

To turn off CTRL-C checking, put the following line in your Config.sys file:

`break = off`

Buffers

Purpose:

Allows you to set the number of disk buffers that MS-DOS allocates in memory at the time you start the system.

Syntax:

`buffers = x`

where:

x is the number of disk buffers, from 2 to 255.

Default:

Memory size	Buffers
For a base system	2
Any disk over 360K bytes	3
128K to 255K bytes	5
256K to 511K bytes	10
512K bytes or more	15

Comments:

A *disk buffer* is a block of memory that MS-DOS uses to hold data when reading or writing.

For applications such as word processors, a number between 10 and 20 provides the best performance. If you plan to create a lot of subdirectories, you may even want to increase the buffers value to between 20 and 30. Remember, though, that buffers take up 512 bytes of space, so the more buffers you have, the less memory you will have available for applications.

Note: Feel free to experiment with different buffer settings to see how different values affect the way your personal computer operates.

Examples:

To create 20 disk buffers, put the following line in your Config.sys file:

`buffers = 20`

Country

Purpose:

Country allows MS-DOS to use international time, date, currency, and case conversions.

Syntax:

```
country = xxx[, [yyy]][, [drive:]filename]]
```

where:

xxx is the phone country code

yyy is the code page for the country

filename is a file containing country information

Default:

Unless otherwise specified, United States settings are assumed.

If you do not specify *filename*, MS-DOS uses the *Country.sys* file for country-specific information.

Comments:

This configuration command identifies to MS-DOS which country's character set you intend to use.

For a list of valid country codes, see Appendix E, "How to Use Code Pages."

Examples:

The following example sets country to France (033) and converts international currency, time, date, and case to French conventions:

```
country = 033
```

Device

Purpose:

Installs the specified device driver on the system list.

Syntax:

`device = [drive:][path]filename[argument]`

where:

argument includes any switches accepted by *filename*.

Default:

None

Comments:

The standard installable device drivers provided with MS-DOS are `Ansi.sys`, `Display.sys`, `Driver.sys`, `Printer.sys`, and `Vdisk.sys`. For more information on these installable device drivers, see Appendix C, "Installable Device Drivers."

If you purchase a new device, like a mouse or a scanner, you generally will receive device driver software with that device. These installable device drivers can be installed using the **device** command. Once you have installed a device driver, make sure that the device driver is in the directory that you specify in any **device** commands.

Note: The device drivers `Country.sys` and `Keyboard.sys` are loaded automatically by MS-DOS. Do not try to load either of these with the **device** command. If you do, it will "hang" your system (that is, MS-DOS will not start).

Examples:

If you plan to use the ANSI escape sequences described in Appendix C, "Installable Device Drivers," you should create a `Config.sys` file that contains the following command:

```
device = ansi.sys
```

This command causes MS-DOS to replace all keyboard input and screen output support with the ANSI escape sequences.

Drivparm

Purpose:

This command allows you to define parameters for block devices when you start MS-DOS, overriding the original MS-DOS device driver settings.

Syntax:

`drivparm = /d: number [/c] [/f: factor] [/h: heads] [/n] [/s: sectors]
[/t: tracks]`

Default:

None

Comments:

Setting **drivparm** overrides any previous block device driver definitions.

The following list describes how each switch is used:

Switch	Function
/d: number	Physical drive number, ranging from 0 to 255. This means that drive number 0 = A, 1 = B, 2 = C, etc.
/c	Shows that change-line (doorlock) support is required. This means that the device driver is able to tell whether the door of a floppy disk drive is open or closed. If the door is open, the device driver assumes that the drive does not have a disk in it yet.

/f:factor Specifies the device type (form factor). The default value for *factor* is 2. *factor* = form factor index, where:

- 0 = 160/180K bytes, or
- 0 = 320/360K bytes
- 1 = 1.2 megabytes
- 2 = 720K bytes (3 1/2-inch disk)
- 3 = 8-inch single density
- 4 = 8-inch double density
- 5 = Hard disk
- 6 = Tape drive
- 7 = 1.44 megabytes (3 1/2-inch disk)

Default values for the following switches depend upon the form factor specified with the /f: switch. If you do not specify the /f: switch, **drivparm** uses a default of 720K bytes (3 1/2-inch disk).

Switch	Function
/h: heads	Maximum head number, ranging from 1 to 99. The default value is 2.
/n	Specifies a nonremovable block device.
/s: sectors	Number of sectors per track, ranging from 1 to 99. The default value is 9.
/t: tracks	Number of tracks per side on the block device, ranging from 1 to 999.

Examples:

Suppose your computer has an internal tape drive on Drive D that is configured at startup to write 20 tracks of 40 sectors per track. If you want to reconfigure this tape drive to write 10 tracks of 99 sectors each, you can put the following line in your Config.sys file:

```
drivparm = /d:3 /f:6 /h:1 /s:99 /t:10
```

This command line overrides the default device driver settings, and supports a tape drive as Drive D (in this case the logical and physical drive numbers are identical). This tape drive has one head and supports a tape format of 10 tracks and 99 sectors per track. (This assumes that the device driver for the tape device supports this configuration of tracks and sectors.) So to create a tape that you can read on another computer, one which can read only this alternate format, you might want to use this method.

FCBS

Purpose:

Allows you to determine the number of file control blocks (FCBS) that can be open concurrently.

Syntax:

`fcbs=x,y`

where:

x is the number of files that file control blocks can open at one time

y is the number of files opened by `fcbs` that MS-DOS cannot close automatically.

Default:

`fcbs = 4,0`

Comments:

A *file control block* is a data structure in real mode used to control open files.

The allowed values for *x* range from 1 to 255. The allowed values for *y* also range from 1 to 255. If an application tries to open more than *x* files by `fcbs`, then all but the first *y* files may be closed by MS-DOS.

Note: To access files, it is best to use file handles instead of file control blocks. However, some older applications that you want to use might require you to use the `fcbs` command in your `Config.sys` file. Thus, you should only use the `fcbs` command if an application requires you to do so.

Example:

To open up to four files by FCBS and to protect the first two files from being closed, put the following line in your `Config.sys` file.

`fcbs = 4,2`

Files

Purpose:

Sets the number of open files that the MS-DOS system calls can access.

Syntax:

`files = x`

where:

x is the number of open files that the system calls can access

Default:

`files = 8`

Comments:

The valid values for *x* range from 8 to 255. The maximum number of files that one program can have open at a time is 65,534.

Note: A process must issue MS-DOS system call 67H to activate the extended handle.

Examples:

To let MS-DOS open 20 files at one time, put the following line in your Config.sys file:

`files = 20`

Lastdrive

Purpose

Sets the maximum number of drives you may access.

Syntax:

`lastdrive = x`

where:

x can be any letter from A to Z

Default:

`lastdrive = e`

Comments:

The *x* value represents the last valid drive that MS-DOS will accept. The minimum number is equal to the number of drives you have installed on your computer.

This command is useful only in a network environment. At startup, MS-DOS recognizes five drives you have on your system. To make any extra drives defined by `lastdrive` valid, a network redirection must occur.

Note that MS-DOS allocates a data structure for each drive that you specify, so you shouldn't specify more drives than are necessary.

Examples:

The following command sets the last drive to M, unless you have added an external logical device with `Driver.sys`. For information about `Driver.sys`, see Appendix C, "Installable Device Drivers."

`lastdrive = m`

Shell

Purpose:

Begins execution of the shell (top-level command processor) from a file defined by the specified *pathname*.

Syntax:

`shell = [drive:][path] filename`

where:

filename is a command processor program

Default:

The default command processor for MS-DOS is Command.com.

Comments:

Instead of reading the standard Command.com, MS-DOS starts the processor specified in filename.

System programmers who write their own command processors (instead of using the MS-DOS file, Command.com) should use the **shell** command to specify the name of their **shell** program.

MS-DOS sets the COMSPEC environment variable equal to the drive:, path, and filename specified on the **shell** command line. This setting overrides the default value for COMSPEC (the drive and pathname of the command processor initially used to start MS-DOS). The operating system uses the COMSPEC environment setting to determine which file to use when reloading any transient part of the command processor.

Note: The **shell** command does not accept switches. However, if the new command processor does accept switches, you can include those switches in this syntax. For example, suppose **shell = Newcmdp.com**. And suppose Newcmdp.com accepts the /c, /p, /e switches. You can include any of these switches in the **shell** command line.

Thus, the following would be a valid command:

`shell = newcmdp.com /p`

Examples:

The following command uses the file `\bin\newshell` as the command processor:

```
shell = \bin\newshell
```

Stacks

Purpose:

Supports the dynamic use of data stacks.

Syntax:

`stacks = n, s`

where:

n is the number of stacks

s is the size of each stack

Default:

Stacks = 9,128

Comments:

The valid values for *n* range from 0 to 64. The valid values for *s* range from 0 to 512.

When a hardware interrupt occurs, MS-DOS allocates one stack from *n* stack specified. When `stacks = 0,0`, MS-DOS does not switch stacks at interrupt time.

Examples:

To allocate eight stacks of 512 bytes each for hardware interrupt handling, you would include the following command in your Config.sys file:

`stacks = 8,512`

Tandy 4000 Expanded Memory Manager

Temmm.sys is an Expanded Memory Manager that enables application programs to access memory beyond the traditional 640K memory boundary on the Tandy 4000. This program emulates *Expanded Memory* as defined in the Lotus®/Intel®/Microsoft® (LIM) expanded memory specification. With Temmm.sys, software can access RAM between the 640K and 1M address, as well as memory above 1M.

The standard Tandy 4000 is equipped with 1M of RAM on the main board. Of this RAM, the operating system and applications use 640 kilobytes. The system also reserves an additional 80K. This leaves 304K of memory available to specially written software such as Temmm.sys.

The RAM disk driver Vdisk.sys and the printer spooler program Spooler.sys, as well as other application software programs, can use an expanded memory manager such as Temmm.sys.

To install Temmm.sys, add a **device** command in your Config.sys file. The format for the **device** command is:

```
device = temmm.sys [xxxx] [ly] [Mz]
```

where:

xxxx represents the actual size of expanded memory (in kilobytes) to be emulated. Use a value in the range 304 to 8192. The system truncates the value you specify to be divisible by 16. The default value is 304K – all of the memory between 640K and 1M that is not used by the system.

y (ly) represents the number of the I/O port addresses emulated by Temmm.sys. If you have one or more hardware boards in your system that also use the port addresses accessed by Temmm.sys, use the following chart to select port address that do not conflict with these boards. If you do not specify y, the default is 5.

y (decimal)	Port Addresses Used (Hex)			
0	208	4208	8208	C208
1	218	4218	8218	C218
5	258	4258	8258	C258
6	268	8268	8268	C268
10	2A8	42A8	82A8	C2A8
11	2B8	42B8	82B8	C2B8
14	2E8	42E8	82E8	C2E8

z (Mz) represents the frame base address for Temm.sys to use for LIM page mapping. The default is 3.

z Value	Frame Address (Hex)
0	C400
1	C800
2	CC00
3	D000
4	D400
5	D800
6	DC00
7	E000

Examples:

Insert the following lines in your Config.sys file to initialize the Tandy Expanded Memory Manager and use your system's 304K bytes of free memory in which to run **vdisk**, **spooler**, and **cache**. These lines give **vdisk** 240K of memory. **Spooler** and **cache** each have 32K of memory.

```
device=temm.sys 304 l5 M3
device=vdisk 240 /A
device=spooler 32 /A
device=cache 32 /A
```

Note:

Do not use Temm.sys with **mon386**.

Installable Device Drivers

Introduction

Device drivers are programs that let the operating system recognize devices that are not part of the computer. Examples of devices are a modem, a printer, a mouse, and an external disk drive. Some device drivers are already installed with MS-DOS. Other device drivers, called *installable device drivers*, come with MS-DOS for you to install if you need them.

This appendix describes the Tandy and MS-DOS installable device drivers:

Device Driver	Purpose
Ansi.sys	Loads the ANSI character set to provide programmers with extended screen and keyboard features.
Display.sys†	Supports code page switching on the console device.
Driver.sys	Enables your system to support more than two diskette drives and more than two hard disk drives. Also lets you assign more than one logical drive letter to a single drive.
Hdrive.sys	Lets you extend the range of hard disks your computer can use.
Lpdvr.sys†	Lets you configure your system to take full advantage of your printer's capabilities.
Modevm.sys†	(For use only in countries outside the United States). Improves video display output on a VM-1 monitor (Cat. No. 26-5111) when used with the Tandy Deluxe Text Display Adapter (Cat. No. 25-3046) or the Deluxe Graphics Display Adapter (Cat. No. 25-3047).

Device Driver	Purpose
Printer.sys†	Provides code page support for PRN, LPT1, LPT2, and LPT3.
Spooler.sys	Lets you continue processing data with your computer while using your printer.
Vdisk.sys	Lets you establish a simulated (<i>virtual</i>) disk drive in RAM.

† These device drivers are on a special MS-DOS International Drivers Diskette (Cat. No. 700-4109).

For more information about the device configuration command, which is used to install these device drivers, see Appendix B, "How to Configure Your System."

Ansi.sys

The Ansi.sys installable device driver lets you use ANSI escape sequences. An ANSI escape sequence is a series of characters (beginning with an escape character or keystroke) developed by the American National Standards Institute (ANSI). These sequences are used to define functions for MS-DOS. Specifically, you can change graphics functions and affect the movement of the cursor.

To install Ansi.sys, include a command line of the following form in your Config.sys file:

```
device = [drive] [path]ansi.sys
```

The escape sequences used in the Ansi.sys file are listed in the following section.

ANSI Escape Sequences Used with MS-DOS

This section lists and explains valid ANSI escape sequences for MS-DOS.

The variables listed in the escape sequences themselves are as follows:

Code	Description
<i>Pn</i>	Numeric parameter—a decimal number that you specify with ASCII digits.
<i>Ps</i>	Selective parameter—a decimal number that you use to select a subfunction. You may specify more than one subfunction by separating the parameters with semicolons.
<i>Pl</i>	Line parameter—a decimal number that you specify with ASCII digits.
<i>Pc</i>	Column parameter—a decimal number that you specify with ASCII digits.

The escape sequences are:

Sequence	Function
ESC [<i>Pl</i> ; <i>Pc</i> H	Cursor Position (CUP)
ESC [<i>Pl</i> ; <i>Pc</i> F	Horizontal & Vertical Position (HVP) CUP and HVP move the cursor to the position specified by the parameters. When no parameters are provided, the cursor moves to the home position (the upper-left corner of the screen).
ESC [<i>Pn</i> A	Cursor Up (CUU) This sequence moves the cursor up <i>Pn</i> lines without changing columns. If the cursor is already on the top line, MS-DOS ignores the CUU sequence.
ESC [<i>Pn</i> B	Cursor Down (CUD) This sequence moves the cursor down <i>Pn</i> lines without changing columns. If the cursor is already on the bottom line, MS-DOS ignores the CUD sequence.
ESC [<i>Pn</i> C	Cursor Forward (CUF) The CUF sequence moves the cursor forward <i>Pn</i> columns without changing lines. If the cursor is already in the far right column, MS-DOS ignores the CUF sequence.
ESC [<i>Pn</i> D	Cursor Backward (CUB) This escape sequence moves the cursor back <i>Pn</i> columns without changing lines. If the cursor is already in the far left column, MS-DOS ignores the CUB sequence.
ESC [6 n	Device Status Report (DSR) The console driver outputs an RCP sequence when it receives the DSR escape sequence.

ESC [s	Save Cursor Position (SCP) The console driver saves the current cursor position. This position can be restored with the RCP sequence.
ESC [u	Restore Cursor Position (RCP) This sequence restores the cursor position to the value it had when the console driver received the SCP sequence.
ESC [2 J	Erase Display (ED) The ED sequence erases the screen. The cursor then goes to the home position.
ESC [K	Erase Line (EL) This sequence erases from the cursor to the end of the line (including the cursor position).

ESC [*Ps*;...; *Ps m* Set Graphics Rendition (SGR)

The SGR escape sequence calls the graphic functions specified by the following numeric parameters. These functions remain until the next occurrence of an SGR escape sequence.

Graphics Functions

0	All attributes off
1	Bold on
2	Faint on
3	Italic on
5	Blink on
6	Rapid blink on
7	Reverse video on
8	Concealed on
30	Black
31	Red
32	Green
33	Yellow
34	Blue
35	Magenta
36	Cyan
37	White
40	Black
41	Red
42	Green
43	Yellow
44	Blue
45	Magenta
46	Cyan
47	White
48	Subscript
49	Superscript

Parameters 30 through 47 meet the ISO 6429 standard.

ESC[=Ps h or
ESC[= h or
ESC[= 0 h or
ESC[?7h

Set Mode [SM]

The SM escape sequence changes the screen width or type to one of the following parameters.

Screen Width Parameters

0	40 x 25 B&W
1	40 x 25 color
2	80 x 25 B&W
3	80 x 25 color
4	320 x 200 color
5	320 x 200 B&W
6	640 x 200 B&W
7	Wraps at the end of each line

ESC[=Ps l or
ESC[= l or
ESC[= 0 l or
ESC[?7l

Reset Mode [RM]

Parameters for RM are the same as for SM (Set Mode), except Parameter 7 resets the mode that causes wrapping at the end of each line.

Display.sys

Display.sys is an installable device driver that supports code page switching for the console device.

To install Display.sys, insert a command line of the following form in your Config.sys file:

```
device = [drive:] [path]display.sys con[:] = ( type[, [hwcp][, n]] )
```

or

```
device = [drive:] [path]display.sys con[:] = ( type[, [hwcp][, (n,m)]] )
```

The following list describes how each variable is used:

Option	Function
<i>type</i>	The display adapter in use. Valid values include MONO, CGA, EGA, and LCD.
<i>hwcp</i>	The code page supported by the hardware. The following values are allowed: 437 (United States) 850 (Multilingual) 860 (Portugal) 863 (French-Canadian) 865 (Norway)
<i>n</i>	The number of additional code pages that can be supported. This number is dependent on the hardware. MONO and CGA do not support other fonts, so <i>n</i> must be 0. EGA can be 2. LCD can be 1.
<i>m</i>	The number of sub-fonts that are supported for each code page.

Driver.sys

Driver.sys is an installable device driver that supports external floppy disk drives.

To install Driver.sys, insert a command line of the following form in your Config.sys file:

```
device = driver.sys /d: number [/c][/f: factor][/h: heads][/n]
[/s: sectors][/t: tracks]
```

The Driver.sys device driver accepts the following switches:

Switch	Function
/d: <i>number</i>	Physical drive number, ranging from 0 to 255. The first physical floppy disk drive is Number 0, and is referenced from the MS-DOS command line as Drive A. Drive Number 1 is the second physical floppy disk drive. Drive 2 is the third, which must be external.
/c	Shows that change-line (door-lock) support is required. This means that the device driver will be able to tell whether the door of a floppy disk drive is open or closed. If the door is open, the device driver will assume that the drive does not have a disk in it yet.
/f: <i>factor</i>	Specifies the device type (form factor). The default is 2. <i>factor</i> = form factor index, where 0 = 160/180K bytes, or 0 = 320/360K bytes 1 = 1.2 megabytes 2 = 720K bytes (3 1/2-inch disk) 3 = 8-inch single density 4 = 8-inch double density 5 = Hard disk 6 = Tape drive 7 = 1.44 megabytes (3 1/2-inch disk)
/h: <i>heads</i>	Maximum head number, ranging from 1 to 99. The default value is 2.

- /n* Specifies a nonremovable block device. A fixed disk is an example of a nonremovable block device.
- /s: sectors* Number of sectors per track, ranging from 1 to 99. The default value is 9.
- /t: tracks* Number of tracks per side on the block device, ranging from 1 to 999. The default value is 80.

Example:

To add an external 720K-byte drive to your computer, you would include the following line in the Config.sys file:

device = driver.sys. d:02

Hdrive.sys

(For Non-Standard Hard Disks)

Hdrive.sys lets you use an extended range of hard disk drive types with your system. **Hdrive's** function is to read the drive parameters from your hard disk, and update the system parameters if they do not match.

Immediately after you add any hard disk drive to your system, your first step is to run the **setup** program to tell the system the drive type.

To determine your drive type, compare the number of cylinders and the number of heads for your drive with the numbers in the following table. All Tandy hard drives have the drive type written on the drive. If your information exactly matches a type in the table, you have a standard drive. If it does not, you have a non-standard drive.

Note: The following table might vary, depending on the version of the ROM BIOS in your computer. Use **setup** to list the standard hard drive types supported by your computer.

Type	Cylinders	Heads	Type	Cylinders	Heads
1	306	4	18	977	7
2	615	4	19	1024	7
3	615	6	20	733	5
4	940	8	21	733	7
5	940	6	22	733	5
6	615	4	23	306	4
7	462	8	25	615	4
8	733	5	26	1024	4
9	900	15	27	1024	5
10	820	3	28	1024	8
11	855	5	29	512	8
12	855	7	36	512	8
13	306	8	37	989	5
14	733	7	38	1024	5
15	Reserved		39	820	6
16	612	4	40	1024	8
17	977	5	41	306	6

If you have a standard hard disk, you can specify the type while running **setup**, and you do not need to use **hdrive**. This is because the standard drive types are recorded in permanent CMOS memory in your computer. When you finish running **setup**, you can use the **autofmt** utility or the **hsect**, **fdisk**, and **format** utilities to prepare your hard disk for use.

If you have a non-standard hard disk (the number of heads and cylinders does not exactly match any entry in your hard disk table) observe the following steps. For information about **setup**, **hsect**, **fdisk**, and **format**, see the *Tandy 3000/4000 MS-DOS Reference Manual*. For information about creating Config.sys files, see "Config.sys Commands" in Appendix B.

1. Run **setup** and select a hard drive type that has the same number of heads but fewer cylinders than the hard disk you want to format.
2. Run **hsect** to perform the low level format of the drive. **Hsect** displays the number of heads and cylinders for the drive type that you selected in Step 1. It then asks if they are correct. Answer **N** (for No). Then, enter the correct number of heads and cylinders when prompted. **Hsect** stores the information you enter onto the disk.
3. After **hsect** is complete, create a Config.sys file on your boot (Startup) diskette with the following statement on it (or add this statement to an already existing Config.sys file):

```
device = hdrive.sys
```
4. If the Hdrive.sys is not in the root directory of your boot diskette, copy it to that directory.
5. Boot the system, using the boot diskette you prepared in Step 3.
6. Run **fdisk** to partition the hard disk.
7. After the hard disk is partitioned, **fdisk** prompts you to insert the DOS diskette into Drive A and press any key to restart the system. Use the boot diskette you prepared in Step 3.
8. Run the **format** utility once for each partition created in Step 4. Be sure to use **format's** /s switch when formatting the active partition.
9. Create a Config.sys file in the root directory of your hard disk containing the same statement shown in Step 3.
10. Your hard disk is ready for use. The next time you boot, your computer will use the hard disk. You do not need to insert a boot diskette.

Lpdrr.sys

The loadable printer driver (provided on the Tandy MS-DOS International Drivers Diskette Cat. No. 700-4109) provides any Tandy printer with several capabilities. The functions of **lpdrr** are:

- Set the number of lines per page
- Set the vertical tabs
- Set the form feed function
- Set the horizontal tabs
- Set the skip perforation function
- Cancel the skip perforation function
- Ignore the next *n* codes
- Reset the printer driver
- Convert a single code into a series of printer codes
- Repeat *n* characters
- Suppress the line feed function

The **device** command that you need to place in the Config.sys file for the printer driver is:

```
device=lpdrr.sys
```

To use any of the printer functions, you must do the following:

1. Install the printer driver by adding this statement to your Config.sys file:

```
device=lpdrr.sys
```

Do this only once.

2. Set up your printer. If your printer has switches to control carriage returns, line feed, or both, then set CR = NL and LF = LF or NL. You can also use the **mode** command to configure **lpdrr** to your switch settings.

If printed text is doubled spaced or if one line is overprinting another, use **mode lloff** or **mode lfon** to turn line feeds after a carriage return off or on. See the **mode** command for more information.

3. Look up the function's control code sequence in the Printer Control Codes table in this section. Then, find the equivalent ASCII code(s) in the ASCII Character Code table in this appendix. For example, the control code needed to set lines per page is ESCAPE C; *n*. The ASCII equivalent of ESCAPE is 27. The ASCII equivalent of C is 67. *n* is the number of lines.
4. Send the control code sequence, in ASCII form, to the printer driver. You can do this in any of three ways.
 - By using BASIC's LPRINT statement with the CHR\$ function, as described in the *Tandy 3000/4000 BASIC Reference Manual*.
 - By making an MS-DOS function call.
 - By making a BIOS call, as described in the *MS-DOS Technical Reference Manual*.

The following is an example of using BASIC to set the skip perforation to 5.

```
LPRINT CHR$(27);CHR$(78);CHR$(05)
```

CHR\$(27) sends the ESCAPE code. CHR\$(78) sends an "N" character. CHAR\$(05) sends the number of lines to skip.

Notes:

- Do not load Lpdrv.sys when using Graphics.com.
- If you want to place both Spooler.sys and Lpdrv.sys in a Config.sys file, always place Spooler.sys before Lpdrv.sys.

Printer Control Codes

Set lines per page

ESCAPE C;*n*

Sets the page length of *n* lines. *n* is a number in the range of 1-127. Lines per page is initially set at 66. Issue this command before setting vertical tabs or form feeds.

Set horizontal tabs

ESCAPE D; *tab1*; *tab2*; *tab3*;...;NUL;

Sets the horizontal tab stops at *tab1*, *tab2*, *tab3*, and so on. The numbers can be in the range 1-131. Tab stops initially are set to every eight columns. Use ESCAPE D to change them. ESCAPE D;0 resets tabs to the initial state.

Set vertical tabs

ESCAPE B; *tab1*; *tab2*; *tab3*;...NK;NUL;

Sets the vertical tab stops to *tab1*, *tab2*, *tab3*, and so on. The values for the tab stops can be in the range of 1 to the page length minus 1. When the printer is turned on, no tab stops are set, and the printer advances according to line feeds. Use Escape B to set the tabs. ESCAPE B ; 0 resets the tabs to the initial state.

Horizontal tab

HT

Tabs to the next horizontal tab stop.

Vertical tab

VT

Tabs down to the next vertical tab stop.

Advance to top of page (form feed)

FF

Advances paper to the next top of page. The printer position is initially top of form. A form feed advances the printer to the top of the next page. To change the number of lines per page use ESCAPE C.

Set skip perforation

ESCAPE N; *n*;

Sets the number of lines to skip to *n*. Use ESCAPE N to put blank lines on the page perforation. Skip is initially set to 0 lines.

Cancel skip perforation

ESCAPE O

Cancels ESCAPE N.

Pass *n* codes directly to the printer

ESCAPE V; *n*;

Tells the printer to accept the next *n* codes as printer control codes.

Reset (cancel) driver

CAN or DEL

Resets the printer port.

Suppress line feed after carriage return

ESCAPE Y;*n*

An *n* value of 0 turns off line feed suppression. An *n* value greater than 0 turns the line feed suppression on. Initially line feeds are suppressed.

Repeat a character *n* times

FS; *n*; *char*

Prints a character or string translation *n* times.

Translate *char* to *string*

ESCAPE W; *n*; *char*; *string*

Defines a character to string conversion.

Disable **lpdrv**r

ESCAPE!

Disables all of **lpdrv**r's capabilities. (See the following "Notes:")

Notes:

- Use BIOS function AH = 1, DX = printer (02-), int 17H to effectively reset **lpdrv**r and your printer. This resets the **lpdrv**r mode and allows it to begin a new command or ESC sequence. It also resets the top of form counter. Some programs do this before printing, such as Print.com. This method does not change the form length, tabs, or skip perforation settings.
- When using the BASIC LPRINT command to define character-to-string translations, the following features can alter the output to **lpdrv**r and the printer:

If the string is longer than WIDTH, BASIC inserts a carriage return and a line feed.

BASIC automatically sends a line feed (CHR\$(10)) with a carriage return code (CHR\$(13)).

- You might need to disable **lpdrv**r when running a word processing program which is correctly configured for your printer. ESCAPE CHR\$(33) disables **lpdrv**r when it is enabled and enables **lpdrv**r when it is disabled. Resetting **lpdrv**r with interrupt 17H as previously mentioned also enables **lpdrv**r and resets the top of form counter.

On some printers, the ESCAPE CHR\$(33) sequence is interpreted as a *change to PC emulation mode* instruction. If your printer has this capability, you must turn the printer off, then on, after issuing the ESCAPE sequence.

- Graphics.com does not operate properly to print color graphics on the Tandy CGP-220 printer unless **lpdrv**r is disabled. **lpdrv**r must also be disabled to download character fonts or for using the high-resolution graphics mode on the Tandy 2100P printer.
- **lpdrv**r detects graphical data sequences and passes these codes directly to DMP printers. When graphics printing is completed, the internal parameters are set to top of form. The printer must be manually set to top of form.

Printer.sys

Printer.sys is an installable device driver that supports code page switching for parallel ports LPT1, LPT2, and LPT3. (The port name PRN may be substituted for LPT1 to refer to the first parallel port.)

To install Printer.sys, insert a command line of the following form in your Config.sys file:

```
device = [drive:] [path]printer.sys lpt x = (type[, [hwcp ]], [n])
```

or

```
device = [drive:] [path]printer.sys lpt x = (type[, [(hwcp1, hwcp2,  
hwcp3, ...)]], [n])
```

The Printer.sys device driver accepts the following options:

Option	Function
<i>type</i>	The printer in use.
<i>hwcp</i>	The code page supported by the hardware. The following values are allowed: 437 (United States) 850 (Multilingual) 860 (Portugal) 863 (French-Canadian) 865 (Norway)
<i>n</i>	The number of additional code pages that can be supported. This number is dependent on the hardware.

Modevm.sys

Modevm is an installable device driver that improves video display output on a VM-1 monitor (Cat. No. 26-5111) used outside the United States.

To install this driver, international users need to place the following command in the Config.sys file:

device = modevm.sys

Spooler.sys

Spooler.sys is a loadable driver that buffers data to the printer by temporarily placing the data in memory. **Spooler** *spools* the data to the printer whenever the printer is available. In this way, you can continue to use your computer to process data while printing other data.

The **device** command that you need to place in Config.sys to install the printer buffer driver is:

```
device = spooler.sys [ printer ][size][/e]
```

or

```
device = spooler.sys [ printer ][size][/a]
```

where:

printer is the number of the printer you want to use. It can be either 1 or 2. If you omit the number, the system assumes you want to use Printer 1. Notice that you must precede the number with a slash (/). You can install two spoolers if you have two printers connected.

size is the buffer size in kilobytes. If you omit *size*, the system uses 2K.

/e Lets you use extended memory (above one megabyte) as a virtual disk. If you use this switch, you cannot use the */a* switch.

/a Lets you use an expanded memory board that meets the Lotus/Intel/Microsoft Expanded Memory Specification if the Expanded Memory Manager software has been installed. If you use this switch, you cannot use the */e* switch.

Vdisk.sys

Vdisk.sys is an installable device driver that lets you use a portion of your computer's memory as if it were a hard disk. This memory area is called a *virtual* or RAM disk.

Virtual disks are much faster than hard disks because the information they contain is always loaded into memory. If your computer has extended memory installed (starting at the one megabyte boundary), or if you have an expanded memory board that meets the Lotus / Intel / Microsoft Expanded Memory Specification, you can use this expanded memory for one or more virtual disks. Otherwise, Vdisk.sys places virtual disks in low memory.

Note: Using Vdisk (including a `device = Vdisk.sys` statement in your `Config.sys` file) increases the size of MS-DOS resident in memory.

To install Vdisk.sys, include a command line of the following form in your `Config.sys` file:

```
device = vdisk.sys [disksize] [sectorsize] [entries] [/e]
```

or

```
device = vdisk.sys [disksize] [sectorsize] [entries] [/a]
```

The Vdisk.sys device driver accepts the following options:

Option	Function
<i>disksize</i>	Specifies the disk size in kilobytes. The default size is 64K bytes; the minimum size, 2K.
<i>sectorsize</i>	Specifies the sector size in bytes. The default size is 128 bytes. The following values are allowed: 128, 256, and 512 bytes.

- entries* Specifies the number of root directory entries. The default value is 64; the minimum, 2; the maximum, 512.
- Vdisk.sys adjusts the value of entries to the nearest sector boundary.
- /e* Lets you use extended memory (above one megabyte) as a virtual disk. If you use this switch, you cannot use the */a* switch.
- /a* Lets you use an expanded memory board that meets the Lotus/Intel/Microsoft Expanded Memory Specification if the Expanded Memory Manager software has been installed. If you use this switch, you cannot use the */e* switch.

Note: When you reset or turn off the power on your computer, the information stored in virtual disks is lost.

Along with the codes and characters in the previous table, please note the following special functions:

Preparing Your Hard Disk

Before MS-DOS can use a hard disk, the disk has to be *initialized*. This initialization involves three steps. First, you must perform *low level* or *hard formatting* of the disk. Then, you must create one or more *partitions*. Finally, you *soft format* each partition.

Your MS-DOS system diskette contains a Tandy program named **autofmt** that automatically performs all of these steps. It prepares a hard disk for use with MS-DOS, including *hard* and *soft* formatting, and partitions the disk. **Autofmt** also installs the MS-DOS system files. If you use **autofmt**, there is nothing else that you need to do to prepare your hard disk, and you do not need the information in this appendix.

However, if you want to take a direct hand in formatting and preparing your hard disk, you must follow these steps:

1. Run the **setup** program supplied on your MS-DOS Startup Diskette. When **setup** asks for the type of your hard disk, refer to the **MEDIA ERROR MAP** that came with the disk and enter the type shown there.

If you are not using a Tandy drive, select a type that has the same number of heads and cylinders as your drive. See the following section "Hard Disk Types" for a list of drive types. "Hdrive.sys" in Appendix C describes what you have to do if you have a non-standard drive (a drive that is not included in the table of pre-defined drive types).
2. Run **hssect** to perform the low-level formatting of the drive. For information on **hssect**, see the reference in Chapter 3.
3. Run **fdisk** to partition the hard disk. For more information, see the **fdisk** reference in Chapter 3. Other sections of this appendix provide additional information on **fdisk**.
4. Run **format** to soft format the MS-DOS partitions created by **fdisk**. For more information see the reference to **format** in Chapter 3.

Hard Disk Types

A table of hard disk types (similar to the following) is built into your computer's BIOS ROM. Your computer might have more or less entries, depending on the ROM version you have. To determine the hard disk types recognized by your system, use the **setup** utility and enter a question mark when asked for the drive type. **Setup** then lists to the screen the types it recognizes.

Type	Cylinders	Heads	Type	Cylinders	Heads
1	306	4	18	977	7
2	615	4	19	1024	7
3	615	6	20	733	5
4	940	8	21	733	7
5	940	6	22	733	5
6	615	4	23	306	4
7	462	8	25	615	4
8	733	5	26	1024	4
9	900	15	27	1024	5
10	820	3	28	1024	8
11	855	5	29	512	8
12	855	7	36	512	8
13	306	8	37	989	5
14	733	7	38	1024	5
15	Reserved		39	820	6 original
16	612	4	40	1024	8
17	977	5	41	306	6

To determine the type of hard disk our have, compare the number of cylinders and the number of heads for your drive with the numbers in the table. (Each hard disk drive comes with the information you need.) If your information exactly matches a type in the table, you have a standard drive. If it does not, you have a non-standard drive.

If you have a non-standard hard disk, specify a type that has the same number of heads and fewer cylinders than your drive has. Then, run **setup**, and use **hsect** to perform an initial or hard format of your disk. You then must

place an `Hdrive.sys` statement in your `Config.sys` file. For complete information on preparing and using a non-standard hard disk, see “`Hdrive.sys`” in Appendix C.

If you are using 5 1/4-inch diskettes, most of the MS-DOS hard disk setup programs are on the your MS-DOS Startup Diskette. The **format** program is on the MS-DOS Operating Diskette. If you are using 3 1/2-inch diskettes, all of the MS-DOS and supplemental programs are on one diskette.

Checking for a Configured and Formatted Disk

There might be times when you are installing a hard disk for your computer and do not know whether it has been formatted or whether MS-DOS is already installed. To find out whether this has been done, do the following:

1. First, try to to start MS-DOS from your hard disk.

If it starts, your hard disk is both configured and formatted, and the MS-DOS system files are on the disk. If MS-DOS does not start, your disk is not formatted to start MS-DOS, but it might have been configured.

2. If MS-DOS does not start, check to see if the disk has been configured with **fdisk**. Do this by placing the MS-DOS master diskette in Drive A, and press `CTRL` `ALT` `DEL` to start MS-DOS. Then, run **fdisk** and select the `Display Partition Data` option to see if any MS-DOS partitions exist.

If any do exist, your disk has been configured. If no partitions exist, follow the instructions in this appendix to configure your disk.

After your hard disk is configured, be sure to format your disk with the MS-DOS **format** /s command before you copy files onto the disk. Otherwise, your files will be unreadable.

More About Fdisk

Hard disks can be divided into one to four separate sections, called *partitions*. Partitions separate your hard disk into individual areas, and each partition can contain a different operating system.

To prepare your hard disk for the MS-DOS operating system, you must create a partition for MS-DOS, called a *DOS partition*. You can create a DOS partition on your hard disk by using a menu-driven utility called **fdisk**. You can use **fdisk** to do any of the following:

- Create a primary DOS partition
- Create an extended DOS partition
- Change the active partition
- Delete a DOS partition
- Display partition information
- Review or modify the configuration of another hard disk on your computer

Warning: Reconfiguring your disk with **fdisk** destroys all existing files in the partition you modify. Be sure to have a backup of all files on your disk before you change an MS-DOS partition with **fdisk**.

How to Start Fdisk

The **fdisk** utility is easy to use because it uses menus to lead you through each procedure. To start **fdisk**, follow these steps:

Place the MS-DOS Startup diskette in Drive A. Type the following command, and press **ENTER**:

fdisk

In response, **fdisk** displays its main menu on your screen. This menu lists five choices. If your computer has only one hard disk, Choice 5 does not appear.

Disk Options

Current Fixed Disk Drive: 1

Choose one of the following:

1. Create DOS Partition
2. Change Active Partition
3. Delete DOS Partition
4. Display Partition Data
5. Select Next Fixed Disk Drive

Enter choice: [1]

Press ESC to return to DOS

The following sections describe each option and show the menus and other information the options display. To return to MS-DOS from the main menu, press **ESC**. You can also use the **ESC** key to return to the main menu from any of the **fdisk** menus.

Most of the **fdisk** menus display a default value. To choose the default value, press **ENTER**. To choose another value, just type the value you want, and press **ENTER**.

How to Create a DOS Partition

If you choose the first option on the main menu and if your hard disk is not yet completely partitioned, **fdisk** displays a screen like the following. If no extended partitions exist, the third option is not displayed:

Create DOS Partition

Current Fixed Disk Drive: 1

1. Create Primary DOS Partition
2. Create Extended DOS Partition
3. Create Logical DOS Drive(s) in
the Extended DOS Partition

Enter choice: [1]

Press ESC to return to Fdisk Options

Selection 1: Create Primary DOS Partition

You must create a primary MS-DOS partition first before you can create any extended MS-DOS partitions on your disk. In most cases, you will need only one MS-DOS partition for your entire disk.

Press **[ENTER]** to accept the default selection (1) and create a primary MS-DOS partition,

The Create Primary DOS Partition menu appears next:

Create Primary DOS Partition

Current Fixed Disk Drive: 1

Do you wish to use the maximum size
for a DOS partition and make the DOS
partition active (Y/N).....? [Y]

Press **[ESC]** to return to Fdisk Options

To use your entire hard disk for MS-DOS, use **fdisk** only once to create the primary MS-DOS partition. If you want to use all the hard disk (up to 32 megabytes) for MS-DOS, press **[ENTER]** to accept the default selection (Y).

Fdisk then displays the following message:

System will now restart

Insert DOS diskette in drive A:

Press any key when ready . . .

Put your MS-DOS Startup Diskette in Drive A, and press any key to restart MS-DOS.

Now that you have created your MS-DOS partition, you must format your hard disk so that MS-DOS can use it. If you want to start MS-DOS from your hard disk, remember to use the /s switch with the **format** command. For example, if you are formatting the disk in Drive C and want to start MS-DOS from that disk, type the following command:

format c: /s

(For more information about **format**, see Chapter 3, "MS-DOS Commands.")

Creating More than One MS-DOS Partition

You can choose to create a primary MS-DOS partition smaller than the maximum size. To do this, type N (for No) in response to the question on the first Create Primary DOS Partition menu. **Fdisk** displays a second Create Primary DOS Partition menu like the following. From this menu, you can specify the size of the primary MS-DOS partition:

Create Primary DOS Partition

Current Fixed Disk Drive: 1

Partition	Status	Type	Start	End	Size
-----------	--------	------	-------	-----	------

Total disk space is 732 cylinders.

Maximum space available for partition
is 732 cylinders.

Enter partition size..... [732]

Press ESC to return to Fdisk Options

The space available on your hard disk is measured in *cylinders*, also called *tracks*. This menu shows the total number of cylinders available for a hard disk partition and prompts you to enter the size of your new partition. The default size for the partition is the maximum available space on the hard disk. Press **[ENTER]** if you want the default size; otherwise, type the size (in cylinders) that you want for the partition, and press **[ENTER]**.

Any part of the disk that you do not use for the primary MS-DOS partition can be used for an extended MS-DOS partition.

Selection 2: Create Extended DOS Partition

Use **fdisk** to create an extended partition if your disk is larger than 32-megabytes (the maximum partition size) or to designate one or more logical drives for the disk. To select Create Extended DOS Partition, type 2, and then press **[ENTER]**. In response, **fdisk** displays a menu like this one:

Create Extended DOS Partition

Current Fixed Disk Drive: 1

Partition	Status	Type	Start	End	Size
C: 1	A	PRI DOS	0	599	600

Total disk space is 1263 cylinders.

Maximum space available for partition
is 663 cylinders.

Enter partition size..... [663]

Press ESC to return to Fdisk Options

This menu shows the total number of cylinders available for an extended partition. The default partition size is the maximum available space on the hard disk. Press **[ENTER]** if you want the default; otherwise, type the size (in cylinders) that you want for the partition, and press **[ENTER]**.

Note: If **fdisk** finds any defective tracks at the start of the partition, it adjusts the partition boundaries to avoid those bad tracks.

Selection 3: Create Logical Drive in the Extended DOS Partition

When you have created an extended partition, you must specify one or more drive letters for that area of the disk. **Fdisk** automatically displays this menu after you create an extended partition. **Fdisk** displays a menu similar to the following if you choose Option 3 from the Create DOS Partition menu:

Create Logical DOS Drive(s)

Drv	Start	End	Size
D:	650	1049	400

Total partition space is 1000 cylinders. Maximum space available for logical drive is 600 cylinders.

Enter logical drive size..... [600]

Press ESC to return to Fdisk Options

You can designate the entire partition as one logical drive or divide it into two or more logical drives. For example, if you want to segregate a particular application and its data files in their own drive, create a second logical drive on the partition.

Because you cannot use an MS-DOS extended partition without a drive letter, **fdisk** continues to prompt you for logical disk drive information until the whole partition has been assigned to a logical drive.

When the entire partition is assigned to logical drives, **fdisk** displays this message:

All available space in the Extended DOS partition is assigned to logical drives.

Press **[ESC]** to return to the main **fdisk** menu.

From there, you can restart MS-DOS or select another option.

How to Change the Active Partition

If you choose the second option on the main menu, **fdisk** displays a screen showing the status of each partition on your hard disk. The active partition, indicated by a status of A, contains the operating system and files you access when you turn on or reset your computer. If you have created a partition on your disk with another operating system, this menu allows you to make that partition the active partition. Only one partition is active at a time; the others are not active.

For example, if you have both XENIX and active DOS partitions on your disk, the Change Active Partition menu might look like this:

Change Active Partition

Current Fixed Disk Drive: 1

Partition	Status	Type	Start	End	Size
C: 1		non DOS	0	1	1
2		non DOS	2	401	400
3	A	PRI DOS	402	731	330

Total disk space is 732 cylinders.

Enter the number of the partition you
want to make active.....: [1]

Press ESC to return to Fdisk Options

Select the partition you want to activate by typing the partition number and pressing **ENTER**. If you press only **ENTER**, Fdisk selects the active partition number as its default.

If your hard disk contains only MS-DOS partitions, **fdisk** displays the following message instead of prompting you for the partition that you want to activate:

The only startup partition on Drive 1
is already marked active.

Press ESC to return to FDISK Options.

How to Delete a DOS Partition

If you choose the third option on the main menu, **fdisk** displays the following menu, which asks you to identify whether the partition you want to delete is a primary or extended DOS partition:

Delete DOS Partition

Current Fixed Disk Drive: 1

1. Delete Primary DOS Partition
2. Delete Extended DOS Partition
3. Delete Logical DOS Drive(s) in
the Extended DOS Partition

Enter choice: []

Press ESC to return to Fdisk Options

Type the selection number you want and press the **ENTER** key. The next menu, whether for a primary or extended DOS partition, shows the status of that partition. When you delete a DOS partition, **fdisk** deletes the partition boundaries and any data that existed in that partition. Once you delete the partition, you cannot recover the data that was on it.

Note: You cannot use **fdisk** to delete a non-DOS partition. Instead, to continue using MS-DOS after you have deleted the DOS partition, you must put an MS-DOS program disk into Drive A. To start a different operating system in another partition of your hard disk, change the active partition to that number before you delete the DOS partition.

Selection 1: Delete Primary DOS Partition

The Delete Primary DOS Partition menu will look similar to this:

Delete Primary DOS Partition

Current Fixed Disk Drive: 1

Partition	Status	Type	Start	End	Size
C: 1	A	PRI DOS	0	399	400
2		EXT DOS	400	731	332

Total disk space is 732 cylinders.

Warning! Data in Primary DOS
partition will be lost. Do you wish
to continue.....? [N]

Press ESC to return to Fdisk Options

If you don't want to remove the primary DOS partition, press to accept the default value (N).

To delete the primary DOS partition:

1. Type Y.
2. Press .

Selection 2: Delete Extended DOS Partition

If you choose to delete an extended partition, you must first delete the logical drives associated with that partition.

Selection 3: Delete Logical Drive in the Extended DOS Partition

To delete a logical drive, type 3 to select the option Delete Logical DOS Drive(s) in the Extended DOS Partition from the Delete DOS Partition menu. Then, press **ENTER**. **Fdisk** displays a menu like the following:

Delete Logical DOS Drive(s)

Drv	Start	End	Size
D:	400	999	600
E:	1000	1399	400

Total partition space is 1000 cylinders.

Warning! Data in the logical DOS drive
will be lost. What drive do you wish
to delete.....? []

Press ESC to return to Fdisk Options

Type the letter of the drive you want to delete, and press the **ENTER** key. **Fdisk** displays this message:

Are you sure.....? [N]

If this logical drive contains valuable data you have not backed up, press **ENTER**. This stops **fdisk** from deleting the logical drive.

Note: Be sure to back up all files you will need from the logical drive before you delete the drive. When **fdisk** deletes a logical drive or partition, the data is destroyed.

If you are sure you want to delete the drive, type Y (for Yes).

How to Display Partition Data

If you choose the fourth option on the main menu, **fdisk** displays a menu that contains information about each of the partitions on your hard disk.

For example, the Display Partition Information menu might look like this:

Display Partition Information

Current Fixed Disk Drive: 1

Partition	Status	Type	Start	End	Size
C: 1	A	PRI DOS	0	399	400
2		EXT DOS	400	731	332

Total disk space is 732 cylinders.

The Extended DOS partition contains logical DOS drives. Do you want to display logical drive information? [Y]

Press ESC to return to Fdisk Options

This information screen identifies the partitions on your disk. It shows each partition's number, status, and type, its starting and ending cylinder numbers, and its size in cylinders.

If you have an extended partition, **fdisk** asks if you want to see information about that partition's logical drives. To display a screen similar to the following, type Y and press **ENTER** :

Display Logical DOS Drive(s)

Drv	Start	End	Size
D:	400	999	600
E:	1000	1399	400

Press ESC to return to Fdisk Options

Press the **ESC** key to return to the main menu.

How to Select the Next Fixed Disk Drive

This option appears on the **fdisk** main menu only if you have more than one hard disk attached to your computer. If you choose this option, **fdisk** changes the current disk drive to the next drive.

For example, if the current disk drive is Drive C and if you choose Option 5 on the main menu, **fdisk** changes the current disk drive to Drive D. You can then choose any of the **fdisk** options (1-4) to prepare the second fixed disk for MS-DOS. Or, you can select Option 5 once again to select the next drive. For example, if there is not a third fixed disk, **fdisk** changes the current disk drive from D back to C.

After you have selected the next drive, **fdisk** displays the main menu again. Note that near the top of the screen, there is a message that looks something like this:

Current fixed disk drive: 2

The activity you select will be performed on the disk shown in this line.

How to Use Code Pages

Introduction

MS-DOS 3.3 provides national language support through the use of language-specific code pages. If you live in, or work with, a country other than the United States, you might choose to use the MS-DOS commands that support code page switching.

A *code page* is a table that defines the character set you are using. A *character set* is a country-specific or language-specific group of characters that are translated from the code page table and displayed by your screen or printer. Each code page character set contains 256 characters. An example of a character set is the set of letters, numbers, and symbols (such as accent marks) used by French-Canadians.

MS-DOS 3.3 supports five different code pages:

- 437 - United States code page.
- 850 - Multilingual code page. This code page includes all characters for most languages of European, North American, and South American countries.
- 860 - Portuguese code page.
- 863 - French-Canadian code page
- 865 - Nordic code page. This code page includes all characters for the Norwegian and Danish languages.

MS-DOS also provides national language support through the use of two other keyboard codes:

- A *country code* defines the country in which you live or work. MS-DOS uses this code to prepare and assign default code pages for your system. MS-DOS recognizes 19 different country codes.
- A *keyboard code* defines the type of keyboard you are using. MS-DOS recognizes 17 different keyboard codes.

National Language Support Codes

The following table lists each country (or language) supported by MS-DOS 3.3. The table also lists the related country codes, default code page assignments, and related keyboard codes. The code pages shown are automatically prepared by MS-DOS when you load the corresponding country code through the Config.sys **country** command. If you do not specify a country code, MS-DOS loads the default United States code page 437.

Country or Language	Country Code	Code Pages	Keyboard Code
United States	001	437, 850	US
French-Canadian	002	863, 850	CF
Latin America	003	437, 850	LA
Netherlands	031	437, 850	NL
Belgium	032	437, 850	BE
France	033	437, 850	FR
Spain	034	437, 850	SP
Italy	039	437, 850	IT
Switzerland	041	437, 850	SF, SG
United Kingdom	044	437, 850	UK
Denmark	045	865, 850	DK
Sweden	046	437, 850	SV
Norway	047	865, 850	NO
Germany	049	437, 850	GR
English (International)	061	437, 850	-
Portugal	351	860, 850	PO
Finland	358	437, 850	SU
Arabic countries	785	437	-
Israel	972	437	-

Notes:

- Both Swiss-French and Swiss-German use country code 041.
- Code pages for Arabic and Hebrew languages are not available. Country codes 785 and 972 assume United States code page 437, but include country-specific date and time conventions.

Commands that Support National Languages

Several MS-DOS commands — new and old — support code page selection and national languages.

New MS-DOS Commands

MS-DOS 3.3 includes three new commands:

nlsfunc	Loads the file containing country-specific information.
chcp	Displays or changes the current code page for the system and all prepared devices.
select	Installs MS-DOS on a new floppy disk with selected country-specific information and keyboard code.

Enhanced MS-DOS Commands

In addition to the new commands, MS-DOS 3.3 includes several enhanced MS-DOS commands that support code page selection. The most significant enhancements include:

keyb	Allows you to select a country-specific keyboard code for the keyboard you are using, and a code page for the character set you prefer. You can also select an alternate keyboard definition file (other than the default Keyboard.sys file) with this command, if another exists.
mode	Includes several new options: <ul style="list-style-type: none">● Preparing a code page for a device● Selecting a code page for a device● Displaying the code pages prepared and selected for a device● Refreshing code pages that were lost due to hardware error

New and Enhanced Configuration Commands

Two Config.sys commands also support country-specific information:

- | | |
|----------------|---|
| country | Identifies the country in which you work or live. This command also defines country-specific conventions to be used, such as date and time formats and sorting sequence for the character set. |
| device | Installs device drivers in the system, including two MS-DOS installable device drivers that support code page switching. These device drivers are called <ul style="list-style-type: none">● Display.sys – used to install a standard console screen device with code-page support● Printer.sys – used to install a standard parallel printer with code-page support |

Date and Time Formats

Four other MS-DOS commands – **date**, **backup**, **restore**, and **time** – now use country-specific date and time conventions, based on the code pages you choose to use.

The following table lists the **date** and **time** formats related to each country (or language group). These formats are determined by the country code set in your Config.sys file.

For each country, the **date** format column shows how MS-DOS would display January 3, 1989, and the **time** format column shows how MS-DOS would display 5:35 p.m. (with zero seconds and zero hundredths of seconds).

Country or Language	Country Code	Date Format	Time Format
United States	001	1-03-1989	17:35:00.00
French-Canadian	002	1989-01-03	17:35:00.00
Latin America	003	03/01/1989	17:35:00.00
Netherlands	031	03-01-1989	17:35:00.00
Belgium	032	03/01/1989	17:35:00.00
France	033	03/01/1989	17:35:00.00
Spain	034	03/01/1989	17:35:00.00
Italy	039	03/01/1989	17:35:00.00
Switzerland	041	03.01.1989	17.35.00.00
United Kingdom	044	03-01-1989	17:35:00.00
Denmark	045	03/01/1989	17.35.00.00
Sweden	046	1989-01-03	17.35.00.00
Norway	047	03/01/1989	17.35.00.00
Germany	049	03.01.1989	17.35.00.00
English (International)	061	03-01-1989	17:35:00.00
Portugal	351	03/01/1989	17:35:00.00
Finland	358	03.01.1989	17.35.00.00
Arabic countries	785	03/01/1989	17:35:00.00
Israel	972	03-01 1989	17:35:00.00

How to Use Code Pages

Unless you specify otherwise, MS-DOS assumes that you want to use the United States character set. To set your system page to support another character set, you need to do four things:

- Set the country code in your Config.sys file. This code identifies the country in which you live or work.
- Load the Country.sys file or other file containing the country-specific information for your country.
- Set the system code page. For most country codes, MS-DOS automatically prepares two system code pages and selects the primary code page for your country automatically. If you want to use the other code page prepared for your country, you can use the **chcp** command.
- Set the keyboard code with the **keyb** command.

Note: Remember that when you change your `Config.sys` file, you must restart MS-DOS to enable the new settings.

Example:

Suppose you live in Quebec, Canada. You would follow these steps to use the French-Canadian character set with your system:

1. First, add the following line to your `Config.sys` file:
`country=002`
2. Then, restart MS-DOS so that MS-DOS can read your revised `Config.sys` file.
3. Next, type the `nlsfunc` command to load the country-specific information found in the `Country.sys` file on your system:

`nlsfunc`

Note: If you forget to type the `nlsfunc` command, MS-DOS does not allow you to specify code pages or keyboard codes.

MS-DOS automatically selects the French-Canadian code page for you to use. Because your country code is 002, MS-DOS has also prepared the Multilingual code page for your system.

4. To change the system code page, type:
`chcp 850`
5. Select the French-Canada keyboard code CF by typing the following command:

`keyb cf`

Note: In place of Steps 3, 4, and 5, you could add the following lines to your `Autoexec.bat` file. Then you would not have to type these commands each time you started MS-DOS:

`nlsfunc`

`chcp 850`

`keyb cf`

Now your computer is set up to use the French-Canadian character set. Since your console screen and printer are independent devices, you also need to set them up for national language support. The next section explains how to do this.

How to Set Device Code Pages

MS-DOS 3.3 lets you define code pages for screen and parallel printer devices that support code pages switching. Unless you want to use the United States code page 437, you need to set up your screen and printer to use the same code page as the rest of your system.

To set up your console screen device (CON) to use code pages, use the *code pages* Config.sys *device* command to load the Display.sys device driver.

Example:

For example, if you are using an EGA display, and want to use the Multilingual code page 850. You could include this command in your Config.sys file:

```
device = display.sys con = (ega,850,2)
```

The last option, 2, allows you to prepare up to two code pages for this device. This is useful if you want to switch between code pages.

Note: Remember to restart MS-DOS to initiate the changes you have made to the Config.sys file.

If you have a parallel printer connected to your personal computer, you need to prepare the same code pages for your printer as for the rest of your system. To do this, use the Config.sys *device* command again to load the installable device driver called Printer.sys.

Example:

If you have a printer that is compatible with the IBM Proprinter, model 4201, connected to LPT1, you would include the following line in your Config.sys file:

```
device = printer.sys lpt1 = (4201,850,2)
```

This command line assumes that the Printer.sys file is on the same disk as your Config.sys file. The last variable, 2, lets you prepare up to two code pages for this printer.

Note: There is no limit to the number of times you can use the *device* command in your Config.sys file.

How to Switch Between Code Pages

If you work in an environment that uses more than one language, you might need to switch between code pages. For example, suppose you work for an international company with offices in New York, London, Stockholm, and Oslo. You might need to use two or three different code pages to read or work from the correspondence you receive from your other offices.

To illustrate how to switch code pages for your system and your devices, suppose that you want to change to Nordic code page 865 to work with some information you receive from the Oslo office. You would follow these steps:

1. First, be sure you have typed the **nlsfunc** command. You only need to type this command once in order to load the country-specific information from the **Country.sys** file.

2. Prepare the code page for each device you intend to use. For example, you would type the following command to prepare code page 865 for the parallel printer connected to LPT2:

```
mode lpt2 codepage prepare = ((865) 4201.CPI)
```

MS-DOS then displays the following message to let you know the code page was prepared for your device:

```
MODE Prepare Codepage function completed
```

3. To prepare code page 865 for your console screen device (CON) you would type the following command:

```
mode con codepage prepare = ((865) 4201.CPI)
```

4. Next, change the code page for the system and all prepared devices by typing the following:

```
chcp 865
```

The display on your screen might flicker slightly as MS-DOS loads a new code for that device.

5. If for some reason you want to load a different code page for a single prepared device, you would use the **select** keyword with the **mode** command. For example, to load code page 850 for your printer, type this command:

```
mode lpt2 codepage select = 865
```

MS-DOS then displays the following message to let you know the code page was prepared for your device:

MODE Select Codepage function completed

Note: If you want to use these commands on a regular basis, you can include these command lines in your Autoexec.bat file.

How to List Current Code Pages

You can list the current prepared and selected code pages for your console screen or a parallel printer by using the **mode** command in the following form:

mode device codepage

For example, to display the current code pages for your console screen device, use the **mode** command in the following form:

mode con codepage

When you do, MS-DOS displays a message similar to this one:

Active codepage for device CON is 437

hardware codepages:

Codepage 850

prepared codepages:

Codepage 437

Codepage 850

Codepage not prepared

Codepage not prepared

MODE Status Codepage function completed

How to Refresh Lost Code Pages

It is possible for prepared code pages to be lost due to hardware errors or other reasons. For example, if you prepared code pages for your printer, and then turned off the printer, the current code page might be lost. You can use the *refresh* keyword with the **mode** command to restore the lost code page.

To illustrate, suppose you had selected code page 850 as the active code page for your console screen (CON), but because of a hardware error, the active code page was lost. You could type the following commands to reinstate the active code pages for your screen:

```
mode con codepage prepare = ((850) ega.cpi)
mode con refresh
```

How to Format a Disk with Country-Specific Information

MS-DOS 3.3 includes a special command, **select**, that:

- Formats a disk.
- Creates a Config.sys file and Autoexec.bat files with country-specific information.
- Copies the contents of the source disk to the target disk.

Warning: Do not use the **select** command with a disk that already contains data files, unless you have backed up the files. Any data on the disk is destroyed when the disk is formatted by either the **select** command or the **format** command.

Example:

To illustrate how **select** works, suppose after configuring your hard disk with **fdisk** you wanted to format your Drive C hard disk. You also want to include the Latin American code page and keyboard code on your hard disk. After placing your MS-DOS master disk in Drive A, you could type the following:

```
select a: c: 003 la
```

After formatting the disk in Drive C, **select** creates two files on the target disk – Autoexec.bat and Config.sys. The contents of the Autoexec.bat file looks something like this:

```
path c:
keyb la 437
echo off
cls
date
time
ver
```

The contents of the Config.sys file looks similar to the following:

```
country = 003, 437
```

Finally, **select** copies the MS-DOS files to the disk on Drive D. If the Autoexec.bat and Config.sys files exist on Drive A, **select** does not copy them to Drive C.

MS-DOS Message Directory

MS-DOS displays three types of messages:

- MS-DOS utility messages
- MS-DOS device error messages
- Application program messages

MS-DOS utility and device error messages are listed in this appendix. For instructions about error messages related to non-MS-DOS software, see your application's documentation.

If a disk or device error occurs at any time during a command or program, MS-DOS displays an error message and includes this prompt:

Abort, Ignore, Retry, Fail?_

MS-DOS waits for you to type one of the following responses:

- a** Abort. End the program requesting the disk read or write.
- i** Ignore. Ignore the bad sector and pretend the error did not occur. This can result in lost data.
- r** Retry. Repeat the operation. Use this response when you have corrected the error (for example, with Not ready or Write protect errors).
- f** Fail. This causes the current MS-DOS system operation to end (fail) and the application to continue.

Note: Some diskette device errors do not display the Ignore option. Instead, they display this prompt:

Abort, Retry, Fail?_

Usually, you will want to recover by first typing *r* (to try again). If the second attempt fails, type *a* to terminate the process.

This section describes MS-DOS messages, their causes, and, if they are the result of an error, how to correct the error. Following each message is the name of the program or programs that might generate the message.

Abort edit (Y/N)? [Edlin]

MS-DOS displays this message when you choose the **edlin q** (quit) command. The **q** command exits the editing session without saving any editing changes. Type **Y** (for Yes) or **N** (for No).

Access denied [Attrib][Find][Print][Replace][Xcopy]

You tried to replace a write-protected, read-only, or locked file.

Active Code Page: xxx [Chcp]

xxx is the code page currently being used by the system.

Active Code Page for device *ddd* is *xxx* [Mode]

xxx is the code page currently being used by the device *ddd*.

Active Code Page not available from con device [Keyb]

The code page that the system is currently using is not supported on the console (screen) you are using.

Add filename? (Y/N) [Replace]

Replace displays this prompt if you specify the **/w** switch. Type **Y** (for Yes) if you want to add the file to the disk or **N** (for No) if you do not want to add the file.

Adding filename [Replace]

Replace displays this prompt to let you know that it is adding this file to your disk.

All files canceled by operator [Print]

MS-DOS displays this message when you specify the **/t** switch with the **print** command.

All logical drives deleted in the Extended DOS Partition

[Fdisk]

Any logical drives previously associated with the extended DOS partition on your disk are now removed.

Allocation error, size adjusted [Chkdsk]

The size of the file indicated in the directory was not consistent with the amount of data actually allocated to the file. The file was truncated to match the amount of data allocated.

All specified file(s) are contiguous [Chkdsk]

All files are written sequentially on the disk. To correct this error automatically, specify the **chkdsk /f** switch.

APPEND already installed [Append]

You have already used the **append** command once since you turned on your computer. Now you are trying to use either the **/x** or **/e** switch with this command. These switches are only valid the first time you type the **append** command. If you want to change the **append** switch, reboot your computer. Then, type the **append** command with the switch you want to use. Otherwise, use the **append** command without these switches. For more information about the **append** command, see Chapter 3, "MS-DOS Commands."

APPEND/ASSIGN Conflict [Append]

You cannot use the **append** command on an assigned drive. Cancel the drive assignment before using the **append** command with this drive again.

Are you sure (Y,N)? [MS-DOS]

MS-DOS displays this message if you try to delete all files in the working directory by using the ***.*** wildcard. Type **Y** (for Yes) to delete all the files or **N** (for No).

Attempted write-protect violation [Format]

The diskette you are trying to format is write-protected.

***** Backing up files to drive x: *****

Diskette Number: *n* [Backup]

Backup displays this message while backing up files to the specified drive.

Be sure to label backup diskettes with the appropriate backup diskette number for use in restoring them later.

Bad call format reading (or writing) drive x:

[MS-DOS device error]

The length of the request header passed to the device header was incorrect.

Bad command error reading (or writing) drive x:

[MS-DOS device error]

A device driver issued an incorrect command to the device specified in the error message.

Bad command or file name [MS-DOS]

The command cannot find the program you asked it to run. Check to see that you typed the command line properly and that the file or command is on the disk or in the command path.

Bad or missing Command Interpreter [MS-DOS]

MS-DOS cannot find the Command.com file on the disk; either the file is missing from the root directory, or the file is invalid. You also receive this message if Command.com has been moved from the directory it was originally in when you started MS-DOS. Either restart the system with a disk that contains Command.com, or copy the Command.com file from your backup MS-DOS master disk onto the disk used to start MS-DOS.

Bad or missing filename [MS-DOS]

You specified a device incorrectly in the Config.sys file. Check the accuracy of the device command in the Config.sys file.

Bad or Missing Keyboard definition file [Keyb]

MS-DOS cannot find the **Keybxx** file that you specified with the **keyb** command. Check to see that the file you specified exists on the disk. Also check to see that your path includes the directory in which this file resides. Then, retype the command. If you get this message again, the **Keyboard.sys** or **Keyb.com** file might be corrupted.

Bad Partition Table [Format]

This message means that there is no DOS partition on the hard disk. You must run **fdisk** to create a DOS partition on your hard disk.

Bad unit error reading drive x: [MS-DOS device error]

Invalid subunit numbers were passed to the device driver.

BREAK is off (or on) [MS-DOS]

This message tells you the current setting of **break**.

Cannot CHDIR to path - tree past this point not processed [Chkdsk]

Chkdsk is checking the structure of the directory and is unable to go to the specified directory. All subdirectories underneath this directory will not be verified. To correct this error automatically, specify the **chkdsk /f** switch.

Cannot CHDIR to root [Chkdsk]

Chkdsk is checking the tree structure of the directory and is unable to return to the root directory. **Chkdsk** is not able to continue checking the remaining subdirectories. Try to restart MS-DOS. If this error persists, the disk is unusable.

Cannot CHKDSK a Network drive [Chkdsk]

You cannot check drives that are redirected over the network.

Cannot CHKDSK a SUBSTed or ASSIGNED drive [Chkdsk]

You cannot check drives that have been substituted or assigned.

Cannot COPY from (or to) a reserved device [Xcopy]

You cannot copy files from or to a device.

Cannot create extended DOS partition while logical drives exist [Fdisk]

Your disk has one or more logical drives assigned to it. These must be deleted before you can create an extended DOS partition. Delete all logical drives by using **fdisk**. Then, create the extended DOS partition.

Cannot create extended DOS partition without primary DOS partition on disk 1 [Fdisk]

You are trying to create an extended DOS partition, but your first hard disk does not contain a primary DOS partition. First, create the primary DOS partition on your first hard disk. Then, if you have more room on that disk, or if you have a second hard disk, you can create an extended DOS partition.

Cannot create a zero cylinder partition [Fdisk]

You are trying to create a partition with a size of 0 cylinders. You must allocate a minimum of 1 cylinder to any partition you create.

**Cannot create Subdirectory BACKUP on drive x:
[Backup]**

The disk might be write-protected, full, or the backup subdirectory might already exist and be read-only. Use another disk as a target disk.

**Cannot DISKCOMP to or from
an ASSIGNED or SUBSTed drive [Diskcomp]**

One of the drives that you specified is a drive that you created using the **assign** or **subst** command.

**Cannot DISKCOMP to or from
a network drive [Diskcomp]**

You cannot compare diskettes on drives that have been redirected over the network.

**Cannot DISKCOPY to or from
an ASSIGNED or SUBSTed drive [Diskcopy]**

One of the specified drives was created with the **assign** or **subst** command.

**Cannot DISKCOPY to or from
a network drive [Diskcopy]**

You cannot copy disks to or from drives that have been redirected over the network.

Cannot do binary reads from a device [Copy]

The **copy** cannot be done in binary mode when you are copying from a device. You should either not use the **/b** switch, or you should use the **/a** switch to specify an ASCII copy.

Cannot edit .BAK file--rename file [Edlin]

You attempted to edit a file that had a filename extension of **.bak** (a back-up copy created by **edlin**). If you must edit a file that has an extension of **.bak**, either rename or copy the file and give it a different extension.

Cannot exec BASICA.COM [MS-DOS]

BASICA cannot be executed by MS-DOS. Check to see that the **Basica.com** file is on the disk you are using. If **Basica.com** is not in your working directory, make sure that the **path** command points to the directory in which it is located. Try executing BASICA again. If you get the same message, the file itself might be bad. Try restoring **Basica.com** from backup.

Cannot format an ASSIGNED or SUBSTed drive [Format]

You attempted to format a drive currently mapped to another drive by the **assign** or **subst** command. Run **assign** or **subst** again, and clear all drive assignments.

Cannot FORMAT a Network drive [Format]

You cannot format drives that are redirected over the network.

Cannot FORMAT nonremovable drive *x* [Backup]

You are trying to back up files with the /f switch. MS-DOS will not allow you to format the target disk specified. Be sure you want to back up files to a hard disk. If you do, you must use a hard disk that is formatted already.

Cannot JOIN a Network drive [Join]

You cannot join drives that are redirected over the network.

Cannot LABEL a Network drive [Label]

You cannot label a drive that is shared on a network server station.

Cannot LABEL a SUBSTed or ASSIGNED drive [Label]

You cannot label a drive if it has been substituted with the **subst** command or assigned with the **assign** command. Check the command line to be sure you specified a valid filename.

Cannot perform a cyclic copy [Xcopy]

When you are using the /s switch, you cannot specify a target that is a sub-directory of the source.

Cannot recover . entry, processing continued [Chkdsk]

The "." entry (working directory) is defective and cannot be recovered.

**Cannot recover .. entry,
Entry has a bad attribute (or link or size) [Chkdsk]**

The ".." entry (parent directory) is defective and cannot be recovered. If you have specified the /f switch, **chkdsk** tries to correct the error automatically.

Cannot RECOVER a Network drive [Recover]

You cannot recover files on drives that are redirected over the network.

Cannot SUBST a Network drive [Subst]

You cannot substitute drives that are redirected over the network.

Cannot SYS to a Network drive [Sys]

You cannot transfer the system files to drives that are redirected over the network. For more information about the net **print** command, see the *Microsoft Networks User's Guide*.

Cannot use FASTOPEN for drive x: [Fastopen]

Fastopen works only with local, fixed disks and can work with a maximum of four disks at a time. You might be trying to use **fastopen** over a network, with a diskette, or with more than four disks at one time, none of which is possible with **fastopen**.

Cannot use PRINT - Use NET PRINT [Print]

You must use the Net **print** command to print files.

CHDIR .. failed, trying alternate method [Chkdsk]

When checking the tree structure, **chkdsk** was not able to return to a parent directory. It will try to return to that directory by starting over at the root and searching again.

xxxxxxx code page drive cannot be initialized [MS-DOS]

MS-DOS cannot start either the Printer.sys or Display.sys program. Check the device command line in your Config.sys file. You probably included an illegal parameter. For more information, see Appendix C, "Installable Device Drivers."

Code page not prepared [Mode]

You have selected a code page that has not yet been prepared for the system, or one that does not have the correct font to support the current video mode. To prepare a code page for the system, use the **mode prepare** command. If you have installed the Display.sys installable device driver, be sure the device command line in your Config.sys file allows for additional subfonts. For more information, see Appendix C, "Installable Device Drivers," and Appendix B, "How to Configure Your System."

Code page xxx not prepared for all devices [Chcp]

You have selected a code page that is not currently supported by a device. First, be sure your device supports code page switching and that it is currently on-line. If the device supports code page switching, use the **mode** prepare command to prepare the device for the code page. Then, retry the **chcp** command.

Code page xxx not prepared for system [Chcp]

Chcp is unable to select a code page for the system. First, be sure that **nlfunc** is installed. If you have not used the device command in your **Config.sys** file to install device drivers, you can retry the **chcp** command. If you are using installable device drivers with your system, you must use the **mode** prepare command to prepare the specific code page for each device on your system. Then, retype the **chcp** command.

Code page operation not supported on this device [Mode]

You have specified a device and code page combination that MS-DOS does not recognize as valid. Check to see that the device you specified exists and that you have listed a valid code page. Also, check to see that that code page is supported on that device.

Code page requested xxx is not valid for given keyboard code [Keyb]

The keyboard code and code page specified are not compatible. Retype the **keyb** command with compatible keyboard code and code page.

Code page specified has not been designated [Keyb]

You have typed the **keyb** command with an option the system doesn't recognize. You must first prepare the associated code page for your console screen device. Use the **mode** prepare command to prepare the associated code page for CON. Then, retype the **keyb** command.

Code page specified has not been prepared [Keyb]

You have typed the **keyb** command with an option the system doesn't recognize. You must first prepare the associated code page for your console screen device. Use the **mode prepare** command to prepare the associated code page for CON. Then, retype the **keyb** command.

Code page specified is inconsistent with invoked code page [Keyb]

This warning message lets you know that the **keyb** option you've selected does not coincide with the code page for your console screen device (CON). Use the **mode select** command if you also want to change the code page for CON.

Code page specified is inconsistent with selected code page [Keyb]

This warning message lets you know that the **keyb** option you've selected does not coincide with the code page for your console screen device (CON). Use the **mode select** command if you also want to change the code page for CON.

Code page xxx [Mode]

This is the code page currently being used by the device specified.

Code pages cannot be prepared [Mode]

You have either specified a duplicate code page for this device or tried to prepare more than the total number of code pages supported for this device. Check the device command line in your Config.sys file to see how many prepared code pages are allowed for this device. Use the **/status** option of the **mode** command to find out which code pages are already prepared for this device. For more information, see Appendix B, "How to Configure Your System" and Chapter 3, "MS-DOS Commands."

Compare another diskette (Y/N)? [Diskcomp]

Diskcomp displays this message when it has completed its comparison of the diskettes. Type Y (for Yes) if you want to compare more diskettes or type N (for No) if you don't.

**Compare error on disk
side *s*, track *t* [Diskcomp]**

Diskcomp found a difference on the diskette in the specified drive, side *s*, track *t*.

Compare OK [Diskcomp]

Diskcomp displays this message if the diskettes are identical.

Compare process ended [Diskcomp]

Diskcomp displays this message if a fatal error occurred during the comparison.

**Comparing *t* tracks
n sectors per track, *s* side(s) [Diskcomp]**

This message confirms the format of the diskettes that you are comparing.

COM port does not exist [Mode]

You have specified an invalid COM port.

Contains *n* non-contiguous blocks. [Chkdsk]

The disk contains fragmented files. If you want to copy this disk, you should use the **copy** or **xcopy** command instead of the **diskcopy** command. The new copy will then store the new files sequentially.

Content of destination lost before copy [Copy]

The source file that you specified in the **copy** command was overwritten before the **copy** process completed. Refer to the **copy** command for the proper syntax.

Convert lost chains to files (Y/N)? [Chkdsk]

Chkdsk displays this message if it finds information on the disk that isn't allocated properly in the disk's File Allocation Table. If you type Y (for Yes) in response to this prompt, **chkdsk** recovers the lost blocks it found when checking the disk. **Chkdsk** then creates a proper directory entry and a file for each lost chain with a filename in the form: File~~nnnn~~.chk. If you type N (for No), **chkdsk** frees the lost blocks so that they can be reallocated and does not recover any data that was in those lost blocks.

Copy another diskette (Y/N)? [Diskcopy]

The Diskcopy command has completed processing. Type Y (for Yes) if you want to copy another diskette or type N (for No) if you don't.

Copying t tracks n Sectors/Track, s Sides [Diskcopy]

Diskcopy displays this message during copying.

Copy process ended [Diskcopy]

Diskcopy could not copy the entire diskette. Use the **copy** or **xcopy** command to copy specific files onto the diskette.

Copyright 1981,82,83,84,85,86,87 Microsoft Corp. [MS-DOS]

This message appears on most MS-DOS utility and command banners.

Corrections will not be written to disk [Chkdsk]

There are errors on the disk, but **chkdsk** will not correct them because you did not specify the /f switch. You must specify the **chkdsk** switch to correct disk errors.

Current code page settings [Mode]

This informational message shows current and prepared code pages for the device specified and for the system.

Current date is *mm-dd-yy* [Date]

The **date** command displays this message. Enter the correct date and press **ENTER**.

Current keyboard code: *xx* code page: *yyy*

Current CON code page: *zzz* [Keyb]

This message displays the current keyboard code (and its associated code page) and current code page used by your console screen device (CON).

Current keyboard does not support this code page [Keyb]

The code page selected is not compatible with the current keyboard code. Check the code page you have selected. If it is correct, change the keyboard code with the **keyb** command.

Current time is *hh:mm:ss*. *hh* [Time]

The **time** command displays this message. Enter the correct time and press **ENTER**.

Data error reading drive *x*: [MS-DOS device error]

MS-DOS could not read the data from the disk properly. This is often due to a defective disk. Try typing **r** (for Retry) several times, or type **a** (for Abort) to end the program. (It's a good idea to make a new copy of the disk, because if it's defective, you can lose information.)

Delete current volume label (Y/N)? [Label]

If a current volume label exists, **label** displays this message in response to the prompt to enter the new volume label for this disk. If you want to delete the volume label, type **Y** (for Yes); otherwise, type **N** (for No).

xxxxxxx device driver cannot be initialized [MS-DOS]

You are trying to install a device driver by using a device command line in your **Config.sys** file. The syntax on that command line is wrong. See Appendix C, "Installable Device Drivers," for the correct syntax of MS-DOS installable device drivers.

Device Error during Status [Mode]

MS-DOS found an error with the specified device when it was checking the status of that device. The problem might be due to a device that does not support code pages, a device not properly prepared for code page switching, a device that cannot support more code pages than those already prepared, or a device with a bad or irregular font file. Check the device command line in your Config.sys file. Be sure that the command syntax and limits for subfonts and additional code pages are all correct. Also, check to see if your device supports code page switching. Consult the hardware vendor if you are unsure.

Device Error during Prepare [Mode]

MS-DOS found an error with the specified device when preparing that device for code page switching. The problem might be due to a device that does not support code pages, a device not properly prepared for code page switching, a device that cannot support more code pages than those already prepared, or a device with a bad or irregular font file. Check the device command line in your Config.sys file. Be sure that the command syntax and limits for subfonts and additional code pages are all correct. Also, check to see if your device supports code page switching. Consult the hardware vendor if you are unsure.

Device Error during Select [Mode]

MS-DOS found an error with the specified device. The problem might be due to a device that does not support code pages, a device not properly prepared for code page switching, a device that cannot support more code pages than those already prepared, or a device with a bad or irregular font file. Check the device command line in your Config.sys file. Be sure that the command syntax and limits for subfonts and additional code pages are all correct. Also, check to see if your device supports code page switching. Consult the hardware vendor if you are unsure.

Device Error during write of font file to device [Mode]

MS-DOS found an error when it tried to write the font file to the specified device. The problem might be due to a device that does not support code pages, a device not properly prepared for code page switching, a device that cannot support more code pages than those already prepared, or a

device with a bad or irregular font file. Check the device command line in your `Config.sys` file. Be sure that the command syntax and limits for sub-fonts and additional code pages are all correct. Also, check to see if your device supports code page switching. Consult the hardware vendor if you are unsure.

Device or code page missing from font file [Mode]

MS-DOS did not find a definition of the indicated code page for this device in the font file. Use the **mode** command to specify another code page for this device. Also, check to see that the font file supports the code page you want to use. This error also can cause specified code pages to be undefined. Use the **mode** command to prepare and refresh lost code pages.

Device *ddd* not prepared [Mode]

No code page has been prepared for this device.

DEVICE Support Not Present [Diskcomp][Diskcopy]

The disk drive does not support MS-DOS 3.3 device control.

Directory is joined [Chkdsk]

Chkdsk does not process directories that are joined. Use the **join /d** command to unjoin the directories, and then run **chkdsk** again.

**Directory is totally empty,
no . or .. [Chkdsk]**

The specified directory does not contain references to working and parent directories. Delete the specified directory and recreate it.

Directory not empty [Join]

You can only join onto an empty directory.

Disk error reading (or writing) drive x:

[MS-DOS device error]

MS-DOS could not read the data from the disk properly. This is often due to a defective disk. Try typing **r** (for Retry) several times, or type **a** (for Abort) to end the program. (It's a good idea to make a new copy of the disk, because if it's defective, you can lose information.)

Disk error reading (or writing) FAT [Chkdsk]

One of your File Allocation Tables has a defective sector in it. MS-DOS automatically uses the other FAT. You should copy all your files onto another disk. To correct this error automatically, simply specify the **chkdsk /f** switch.

Diskette bad or incompatible [Diskcopy]

The source diskette is not formatted or is formatted incorrectly. You cannot copy it.

Disk full. Edits lost [Edlin]

Edlin was not able to save your file due to lack of disk space.

You should always be sure that there is enough room on the default disk to save your file before you use the **edlin E** (end) command. You should also be sure that the default disk is not write-protected.

Disk unsuitable for system disk [Format]

The **format** program detected a bad track on the disk where system files should reside. You should use this disk to store data only.

Do not specify filename(s)

Command format: DISKCOMP d: d:[/1] [/8] [Diskcomp]

You specified an incorrect switch or gave a filename in addition to a drive name.

Do not specify filename(s)

Command format: DISKCOPY d: d:[/1] [Diskcopy]

You specified an incorrect switch or gave a filename in addition to a drive name.

Do you see the leftmost 0? (Y/N) [Mode]

Mode displays this message to help you align the test pattern on your screen. Type Y (for Yes) if you can see the leftmost 0 in the test pattern, or type N (for No) if you want to shift the display to the right.

Do you see the rightmost 9? (Y/N) [Mode]

Mode displays this message to help you align the test pattern on your screen. Type Y (for Yes) if you can see the rightmost 9 in the test pattern, or type N (for No) if you want to shift the display to the left.

Do you wish to use the maximum size for a DOS partition and make the DOS partition active (Y/N).....[] [Fdisk]

You are formatting your hard disk.

Type Y (for Yes) and press **ENTER** to format your entire hard disk as the primary DOS partition, or type N (for No) and press **ENTER**

Does name specify a file name or directory name on the target (F = file D = directory)? [Xcopy]

Xcopy displays this prompt if the target directory does not exist. Type f if the name specifies a file or d if the target specifies a directory that does not currently exist.

(.)(..) Does not exist [Chkdsk]

This is an informational message from **chkdsk** indicating that either the . or .. directory entry is invalid.

DOS 2.0 or later required [Attrib][Backup][Fc][Graphics][Join][Mode][Restore][Subst]

You cannot use these utilities with 1.xx versions of MS-DOS.

Drive D already deleted [Fdisk]

You tried to delete Drive D, but it had already been deleted.

Drive deleted [Fdisk]

You deleted a hard drive from the system.

Drive has been changed or deleted [Fdisk]

You changed or deleted a hard drive on the system.

Drive letter must be specified [Format]

You did not specify the drive letter for the drive that you want to format. You must specify the name of the drive that you want to format.

Drive x: not ready

Make sure a diskette is inserted into the drive and the door is closed [Diskcomp][Diskcopy]

The drive is empty, or you did not close the door of the diskette drive.

Drive types or diskette types not compatible [Diskcomp][Diskcopy]

You must have the same size and type of diskettes to run these commands. For example, you cannot copy from a single-sided diskette to a double-sided diskette or compare a high-density diskette with a low-density diskette. You should use **Fc** if you want to compare the files on the diskettes. If you want to copy the diskette, you can use **copy** (or **xcopy**) or **reformat** the target diskette so that it's the same type as the source diskette, or use a diskette of the same type.

Duplicate file name [Rename]

You tried to rename a file to a filename that already exists, or the name you specified could not be found.

ECHO is off (or on) [MS-DOS]

This message tells you the current status of Echo.

End of input file [Edlin]

The entire file was read into memory. If the file was read in sections, this message indicates that the last section of the file is in memory.

Enter current Volume Label for drive x: [Format]

Format asks you to enter the current volume label for verification before it formats the hard disk in the specified drive. If you do not know what the volume label is, press **CTRL** **C** to terminate this command, and give the **vol** command for the specified drive. Then, give the **format** command again.

Enter new date: [Date]

You must respond to this prompt when you start MS-DOS or use the **date** command. Type the date in the format indicated by the prompt, or press **ENTER** to accept the current date.

Enter new time: [Time]

You must respond to this prompt when starting MS-DOS. Type the time as indicated by the prompt, or press **ENTER** to accept the current time.

Entry error [Edlin]

The last command you typed contained a syntax error. Retype the command with the correct syntax, and press **ENTER**.

Entry has a bad attribute (or link or size) [Chkdsk]

This message can be preceded by one or two periods to show which subdirectory is invalid. If you have specified the **/f** switch, **chkdsk** tries to correct the error automatically.

Error during read of Font file [Mode]

MS-DOS found an error when it tried to read the font file for the code page specified.

Error in country command [MS-DOS]

You used the incorrect syntax for the **country** command in your **Config.sys** file. For the correct syntax of this configuration command, see Appendix B, "How to Configure Your System."

Error in .EXE file [MS-DOS]

The .exe file you have asked MS-DOS to load has an invalid internal format. You cannot run this program. Check to be sure that you are using the correct version of MS-DOS.

Error opening log file [Backup]

MS-DOS cannot open the backup log file. Check the drive and path specified with this command. Remember that the log file cannot be located on the target drive. If you did not specify a filename for the log, the error occurred when MS-DOS tried to open and create Backup.log on the source diskette.

Error reading/writing partition table [Format]

Format could not read or write the partition table. You should run **fdisk** on the disk and then try formatting it again.

Errors found, F parameter not specified Corrections will not be written to disk [Chkdsk]

Chkdsk found errors on the disk. If you have not specified the /f switch, **chkdsk** continues printing messages but will not correct the errors. You should run **chkdsk** with the /f switch if you want to correct the problems encountered by the **chkdsk** command.

Errors on list device indicate that it may be off-line. Please check it. [Print]

Your printer is not turned on.

Error trying to open backup log file Continuing without making log entries. [Backup]

You specified the **backup** /L switch, but **backup** could not create the backup log file.

Error writing to device [MS-DOS]

You tried to send too much data to a device, so MS-DOS was unable to write the data to that device.

EXEC failure [MS-DOS]

MS-DOS either found an error when reading a command or the **files** command in the Config.sys file is set too low. Increase the value of the **files** command in the Config.sys file, and restart MS-DOS.

Extended DOS partition already exists [Fdisk]

You cannot create another extended DOS partition.

Extended DOS partition created [Fdisk]

You have created an extended DOS partition on your hard disk.

Extended DOS partition deleted [Fdisk]

You have deleted an extended DOS partition from your hard disk.

Failure to access code page font file [Mode]

MS-DOS cannot open the font file for the specified code page. Check to see that you typed font file name and its pathname correctly. Also, check the Config.sys file to see that the device driver for this device has been properly installed. If the Config.sys file is incorrect, correct it and restart MS-DOS before retyping the **mode** command.

Failure to access country.sys [Select]

MS-DOS cannot open the Country.sys file. Check to see that your path points to the directory in which Country.sys resides. Then, retype the command.

Failure to access device: xxx [Mode]

You are trying to specify a code page for a particular device, but MS-DOS cannot access the device listed. Retype the command using an existing device. Be sure you are typing the device name correctly.

Failure to access keyboard.sys [Select]

MS-DOS cannot open the Keyboard.sys file. Be sure Keyboard.sys exists on your source disk.

Fastopen already installed [Fastopen]

Fastopen is already installed on the system.

Fastopen installed [Fastopen]

This informational message acknowledges that you have just installed fastopen.

FCB unavailable reading (or writing) drive x:
[MS-DOS device error]

An unusual error has occurred. This error usually requires an experienced programmer to fix it. Type r (for Retry) or a (for Abort).

fc: cannot open filename - No such file or directory [Fc]

One of the files that you specified doesn't exist. Check the directory for the correct filename.

fc: filename longer than filename [Fc]

After reaching the end of one of the files in a file comparison, the other file still has data left that was not yet compared.

fc: incompatible switches [Fc]

You have specified switches that are not compatible (for example, /b and /L). You should not combine binary and ASCII comparison switches.

fc: no differences encountered [Fc]

The files are the same.

fc: out of memory [Fc]

You do not have enough memory to perform the comparison.

File allocation table bad [MS-DOS]

The disk might be defective. Run **chkdsk /f** to check the disk.

File allocation table bad drive x: [Chkdsk]

This message means that the disk was not formatted or was formatted improperly. It could also mean that an operating system other than MS-DOS is on the disk. Run **chkdsk /f** to check the disk. If this message is displayed again, you must reformat the disk.

File filename canceled by operator [Print]

MS-DOS displays this message when you specify the **/t** switch with the **print** command.

File cannot be converted [Exe2bin]

The input file is not in the correct format.

File cannot be copied onto itself [Copy][Replace][Xcopy]

The source filename you specified is the same as the target filename.

File creation error [MS-DOS][Edlin][Restore][Xcopy]

You tried to add a new filename or replace a file that already exists in the directory, or there was not enough space for the file. If the file already exists, it is a read-only file and cannot be replaced. This error message can also occur if the root directory is full, out of files, or if the filename is the same as a volume or directory or a hidden (or system) file.

File is READ-ONLY [Edlin]

The file is designated read-only, so you cannot change it.

File name must be specified [Edlin]

You did not specify a filename when you started **edlin**. You should type the **edlin** command followed by a filename.

File not found

[Chkdsk][Edlin][Fc][Find][Print][Recover][Rename][Xcopy]

MS-DOS could not find the file that you specified, or you tried to rename a file with a name already in the directory. Check to see that you entered the filename correctly.

File not in PRINT queue [Print]

The file that you specified was not in the **print** queue, so you cannot remove it from the queue. Check to see that you entered the filename correctly.

**Files cannot be added to this diskette
Unless the PACK (/P) switch is used
Set the switch (Y/N)? [Backup]**

The target disk does not have enough room for any of the files on the source disk without dividing them across disks. If you do not want to divide a file across disks, type N (for No). If your files are larger than will fit on one diskette, type Y (for Yes).

***** Files were backed up
at time on date *** [Restore]**

This is an informational message only.

FIND: Access denied [Find]

You cannot access the file. Be sure that the disk is not write-protected, read-only, or locked.

FIND: File not found [Find]

MS-DOS could not find the file that you specified. Be sure you have typed the filename correctly.

FIND: Invalid number of parameters [Find]

You specified either too many or too few options in the command line.

FIND: Invalid Parameter [Find]

One of the switches you specified is wrong.

FIND: Read error in filename [Find]

The **find** command could not read the specified file.

FIND: Syntax error [Find]

Check to be sure that you have typed the command correctly.

**First cluster number is invalid,
entry truncated [Chkdsk]**

The file directory entry contains an invalid pointer to the data area. If you specified the /f switch, the file is truncated to a zero-length file.

FIRST diskette bad or incompatible [Diskcomp]

Diskcomp cannot recognize the format on the source disk. You should run **chkdsk** to help you identify the problem.

Fixups needed - base segment hex: [Exe2bin]

The source (.exe) file contained information indicating that a load segment is required for the file. You must specify the absolute segment address where the finished module is to be located.

Font File contents invalid [Mode]

MS-DOS cannot use the contents of the font file specified. Be sure you are typing the name of the font file correctly. Retype the command. If this message is displayed again, your font file might have been altered or corrupted. Recopy this file from the master MS-DOS disk. Type the command again. This error can also cause existing selected code pages to be undefined. Use the **mode** command to prepare these code pages again and to refresh them.

For cannot be nested [MS-DOS]

You cannot nest **for** commands in a batch file.

Format another (Y/N)? [Format]

Format displays this message when it has finished formatting a disk. Type Y (for Yes) if you want to format another disk, or type N (for No) if you don't. If you accidentally type Y, you can cancel the format process by typing **CTRL C** in response to the message Strike any key.

Format complete [Format]

Format displays this message when it has finished formatting the disk in the specified drive.

Format failure [Format]

MS-DOS could not format the disk. This message is usually displayed with an explanation as to why the command failed.

Format not supported on drive x: [Format]

You cannot use **format** to format this drive. You might have specified device parameters that your computer cannot support.

Formatting while copying [Diskcopy]

Diskcopy displays this message if the target diskette has never been formatted.

General failure reading (or writing)

drive x: [MS-DOS device error]

An unusual error has occurred. This error usually requires an experienced programmer to fix it. Type r (for Retry) or a (for Abort).

Graftabl needs DOS version 2.0 or later [Graftabl]

You cannot use **graftabl** with 1.xx versions of MS-DOS.

Graphics characters already loaded [Graftabl]

The **Graftabl** command displays this message if you have already loaded the table of graphics characters into memory.

Graphics characters loaded [Graftabl]

The **graftabl** command displays this message after it loads the table of graphics characters into memory.

Hardware code pages:

Prepared code pages: [Mode]

This message lists the current code pages prepared for the device specified.

Has invalid cluster, file truncated [Chkdsk]

The file directory entry contains an invalid pointer to the data area. If you specified the /f switch, the file is truncated to a zero-length file.

Head: hhh Cylinder: ccc [Format]

Format displays the head and cylinder number of the track currently being formatted.

Illegal device name [Mode]

Your computer does not recognize this device name.

Incompatible system size [Sys]

The system files occupy more space on the source diskette than is available on the target diskette. You cannot use the **sys** command to transfer the system files to this diskette.

Incorrect APPEND Version [MS-DOS]

You are not using the MS-DOS 3.3 **append** command. You are using another incompatible version.

Incorrect DOS Version

[Append][Attrib][Backup][Chkdsk][Diskcomp][Diskcopy][Edlin][Fc]
[Find][Format][Graphics][Join][Keybxx][Label][Mode][More][Print]
[Recover][Replace][Restore][Share][Sort][Subst][Sys][Tree]
[Xcopy]

Some MS-DOS utilities will not run on older versions of the operating system, and many are written to run only on the exact version of MS-DOS that they were created for. You must use the correct version of MS-DOS to run this command.

Incorrect DOS Version, use DOS 2.00 or later [Link]

Some MS-DOS utilities will run only on MS-DOS version 2.00 or later versions.

Incorrect number of parameters [Join][Subst]

You specified either too many or too few options in the command line.

Incorrect parameter [Assign][Share]

One of the options you specified is wrong.

Infinite retry on parallel printer timeout [Mode]

Your printer is off-line or not ready. If it appears to be ready, try pressing **CTRL** **ALT** **DELETE** to reset the computer.

Insert backup diskette *n* into drive *x*: [Backup][Restore]

This message prompts you for the *n*th backup diskette. Put the next diskette into the specified drive. Be sure to label each backup diskette in the appropriate order for use when restoring the files.

**Insert destination disk in drive *x*:
and strike any key when ready** [Sys]

This message appears when you are using **sys** to transfer the operating system with a single diskette drive. You should insert a diskette in the appropriate drive, and press any character or number key to begin processing.

**Insert diskette for drive *x*:
and strike any key when ready** [MS-DOS]

This message appears when MS-DOS is copying and formatting. You should insert a diskette in the appropriate drive, and press any character or number key to begin processing.

**Insert diskette with batch file
and press any key when ready [MS-DOS]**

The diskette containing your batch file is not in the drive you originally specified. Reinsert the diskette that contains the batch file in the appropriate drive.

**Insert DOS diskette in drive x:
and strike ENTER when ready [Format]**

You typed the **format /s** command, but the diskette in the default drive does not contain MS-DOS system files. Insert a diskette with the files **Io.sys** and **Msdos.sys** in the drive specified, and press any key.

Insert FIRST diskette into drive x: [Diskcomp]

This message prompts you for the first diskette that you want to compare.

**Insert last backup diskette in drive x:
Strike any key when ready [Backup]**

This message prompts you for the final backup diskette. After you put the final backup diskette into the drive specified, press any alphanumeric key to continue the **backup** process.

Insert restore target diskette into drive x: [Restore]

Restore displays this prompt if you are restoring files to a diskette. Put the target diskette into the specified drive.

Insert SECOND diskette into drive x: [Diskcomp]

This message prompts you for the diskette that you want to compare with the first diskette.

Insert source disk [Backup]

This message prompts you to put the source diskette into the drive.

Insert SOURCE diskette into drive x: [Diskcopy]

This message prompts you to put the diskette to be copied into the specified drive.

**Insert system diskette in drive x:
and strike any key when ready [Sys]**

Sys needs a diskette from which to read the Io.sys and Msdos.sys files. Insert a system diskette into the specified drive, and press any character or number key to start the system copy process.

Insert TARGET diskette into drive x: [Diskcopy]

Diskcopy displays this message to prompt you to place the target diskette into the specified drive. If your computer has one diskette drive, this message prompts you to put the proper diskette into the drive.

Insufficient disk space [MS-DOS][Replace][Sort][Xcopy]

The disk is full and does not contain enough room to perform the specified operation.

Insufficient memory

[Backup][Chkdsk][Diskcomp][Diskcopy][Edlin][Replace][Restore]
[Sort][Xcopy]

There is not enough memory in your computer to perform the specified operation. Before retrying this operation, you must free memory by deleting files. In **edlin**, you are able to free memory by typing a **w** (write) command followed by an **a** (append) command.

Insufficient memory for system transfer [Format]

Your memory configuration is insufficient to transfer the MS-DOS system files Io.sys and Msdos.sys with the **format** /s switch.

Insufficient room in root directory.

Erase files in root and repeat CHKDSK [Chkdsk]

Chkdsk always recovers lost files into the root directory. In this case, your root directory is full. Delete some files in your root directory, or move them to another directory to make room for the lost files.

Intermediate file error during pipe [MS-DOS]

The pipe operation uses temporary files on the disk that are deleted automatically once the piping process is complete. An error has occurred in one of these files. Be sure that there is enough room on the disk for the temporary file and that the diskette is not write-protected, and try the command again.

Internal error [Fc][Mode][Share]

This message indicates an error in the utility.

**Internal stack overflow
System Halted [MS-DOS]**

The system tried to use more stacks than were available. This caused a series of hardware interrupts and halted the system. Restart MS-DOS. Then, edit your Config.sys file and allocate more stack resources. For more information, see Appendix B, "How to Configure Your System."

Invalid argument [Backup][Fc][Restore]

You have specified an invalid argument. Refer to Chapter 3, "MS-DOS Commands," for the correct syntax of the command, and try again.

Invalid baud rate specified [Mode]

You have specified an incorrect baud rate. Valid choices are 110, 150, 300, 600, 1200, 2400, 4800, and 9600. You must specify at least the first two digits of the baud rate.

Invalid characters in volume label [Format][Label]

The volume label should only contain up to 11 alphanumeric characters.

Invalid code page specified [Chcp]

You selected an invalid code page number. Retype the command with the correct code page.

Invalid COMMAND.COM
Insert COMMAND.COM disk in default drive
and strike any key when ready [MS-DOS]

The program you have just run used almost all the memory. MS-DOS must now reload the Command.com file from diskette. However, either MS-DOS cannot find Command.com on the diskette, or the copy it has found is the incorrect version. Insert a diskette that contains a copy of Command.com into the default drive. (It must be the same version with which you started MS-DOS.)

Invalid country code [MS-DOS]

In your Config.sys file you have specified a country number that is not in the table of files configured in this version of MS-DOS. Country codes must be in the range 1-99 and are set by Tandy.

Invalid country code or code page [MS-DOS]

MS-DOS found an invalid country code or code page number in your Config.sys file. Correct the country command line in your Config.sys file.

Invalid current directory [Chkdsk]

Your diskette has an invalid directory on it. You might be able to recover some of the files on this diskette by copying them with the **copy** command. Otherwise, you must replace the diskette.

Invalid date [Date][Xcopy]

You specified an invalid date in response to the **date** prompt. Enter a valid date. Refer to Chapter 3, "MS-DOS Commands," for the proper syntax of the **date** command.

Invalid Date/Time [Backup]

You specified an invalid date with one of the **backup** command switches. Refer to Chapter 3, "MS-DOS Commands," for the proper syntax of the **backup** command, and then try again.

Invalid device [MS-DOS]

The device specified was not AUX, CON, NUL, or PRN.

Invalid device parameters from device driver [Format]

Format displays this message when the number of hidden sectors is not evenly divisible by the number of sectors per track. (That is, the partition does not start on a track boundary.) This might happen if you tried to format a hard disk that previously had been formatted with MS-DOS 2.x without first running **fdisk**, or if you have set the device driver parameters incorrectly. Check the **Config.sys** file for incorrect **device** or **drivparm** commands.

Invalid directory [MS-DOS]

The directory you specified either does not exist or is invalid. Check to see that you entered the directory name correctly.

Invalid disk change reading (or writing) drive x: [MS-DOS device error]

You changed the diskette in a drive when you weren't supposed to. Put the diskette back in the drive, and type **r** (for **Retry**).

Invalid drive in search path [MS-DOS]

The drive does not exist.

Invalid drive or filename [Edlin][Recover]

You did not type a valid drive name or filename. Enter a valid drive name or filename.

Invalid drive specification

**[Backup][Chkdsk][Diskcomp][Diskcopy][Format][Label][Print]
[Replace][Restore][Sys][Tree][Xcopy]**

The drive is incorrect or does not exist. Enter a valid drive name.

Invalid environment size specified [Command]

You gave an invalid number of bytes with the **/e** switch. You must specify a number between 160 and 32,768 (bytes).

Invalid keyboard code specified [Keyb]

You selected an invalid keyboard code with the **keyb** command. Retype the command with the correct keyboard code.

Invalid language specified [Keyb]

You typed an invalid keyboard code with the **keyb** command. See the **keyb** command in Chapter 3, "MS-DOS Commands," for a list of valid keyboard codes. Retype the command using a valid keyboard code.

Invalid number of parameters

[Attrib][Backup][Fc][Find][Recover][Restore][Xcopy]

Either you did not specify an option or string, or you specified the wrong number of options in the command line.

Invalid parameter(s)

[Backup][Chkdsk][Diskcomp][Diskcopy][Edlin][Find][Format][Join][Mode][Print][Replace][Restore][Sort][Subst][Sys][Tree][Xcopy]

One of the switches you specified is wrong or does not exist. Refer to Chapter 3, "MS-DOS Commands," to be sure you are using the correct switches.

Invalid path, not directory, or directory not empty

[MS-DOS]

You are unable to remove the directory requested for one of the specified reasons.

Invalid path (or file not found)

[Attrib][Backup][Copy][Restore][Tree][Xcopy]

You have entered a pathname or filename that does not exist. Enter a valid pathname or filename with the command.

Invalid path or parameter [Append]

You specified a file or directory that does not exist. Enter a valid pathname or filename with the **append** command.

Invalid signature in COUNTRY.SYS file [Select]

Select reads the Country.sys file to verify the country code. **Select** quits if it cannot find a proper file header or a specific country code.

Invalid signature in KEYBOARD.SYS file [Select]

Select reads the Keyboard.sys file to verify the keyboard code. **Select** quits if it cannot find the proper file header or a specific keyboard code.

Invalid STACK parameter [MS-DOS]

The syntax of the **stack** command in your Config.sys file includes an invalid parameter. See Appendix B, "How to Configure Your System," for the correct syntax of that configuration command.

Invalid sub-directory entry [Chkdsk]

The subdirectory that you specified either does not exist or is invalid. Check whether you typed the subdirectory name correctly.

Invalid syntax [MS-DOS]

You used the wrong syntax when typing a command. See Chapter 3, "MS-DOS Commands," for the correct syntax of an MS-DOS command.

Invalid syntax on DISPLAY.SYS code page driver [MS-DOS]

You used the wrong syntax when you typed the **device** command in your Config.sys file to load Display.sys. See Appendix B, "How to Configure Your System," for the correct syntax of **device**. Also see Appendix C, "Installable Device Drivers," for information about the Display.sys installable device driver.

Invalid syntax on PRINTER.SYS code page driver [MS-DOS]

You used the wrong syntax when you typed the **device** command in your Config.sys file to load Printer.sys. See Appendix B, "How to Configure Your System," for the correct syntax of **device**. Also see Appendix C, "Installable Device Drivers," for information about the Printer.sys installable device driver.

Invalid syntax on PRINTER.SYS code page switching device drivers [Mode]

You used the wrong syntax when you typed the **device** command in your Config.sys file to load Printer.sys. See Appendix B, "How to Configure Your System," for the correct syntax of **device**. Also see Appendix C, "Installable Device Drivers," for information about the Printer.sys installable device driver.

Invalid time [Time]

You specified an invalid time. Refer to Chapter 3, "MS-DOS Commands," for the correct syntax, and try the command again.

Invalid Volume ID [Format]

Format displays this message if you enter a volume label that doesn't match the label on the hard disk you want to format. It then quits the **format** process. Use the **vol** command to find out what the volume label for the hard disk is, and then try the command again.

Invalid working directory process cannot continue [Chkdsk]

The current directory of the disk being checked is damaged and unusable.

x is not a choice, Please enter y-z [Fdisk]

You tried to select an invalid option **x**. Select a valid option from the range shown (**y-z**).

KEYB has not been installed [MS-DOS]

No alternate keyboard code has been installed for your system. If you want to use keyboard code other than the default U.S. (QWERTY) keyboard, use the **keyb** command to install it.

Label not found [MS-DOS]

Your batch file contains a **goto** command to a nonexistent label.

Last backup diskette not inserted
Insert last backup diskette in drive x:
Strike any key when ready [Backup]

This message prompts you for the final backup diskette. After you put the final backup diskette into the drive specified, press any alphanumeric key to continue the **backup** process.

***** Last file not backed up *** [Backup]**

Backup could not back up the last file on the diskette. This message can occur if there is no more room on the target disk. It can also occur if there was an error in the source file or on the target disk. You might have to back up this file separately to another disk.

Line too long [Edlin]

During an **edlin r** (replace) command, the string given as the replacement caused the line to expand beyond the limit of 253 characters. You should divide the long line into two lines and retry the **r** command.

List output is not assigned to a device [Print]

When you first type the **print** command, MS-DOS asks you what device you want to specify as a printer. This message appears if **print** is set up for a device that does not exist.

Lock violation reading (or writing) drive x:
[MS-DOS device error]

A program tried to access part of a file that another program was using. Type **a** (for Abort), or wait awhile and type **r** (for Retry).

x lost cluster(s) found in y chains
Convert lost chains to files (Y/N)? [Chkdsk]

Chkdsk displays this message if it finds information on the disk that isn't allocated properly in the disk's File Allocation Table. If you type Y (for Yes) in response to this prompt, **chkdsk** recovers the lost blocks it found when checking the disk. **Chkdsk** then creates a proper directory entry and a file for each lost chain with the filename of the form: Filennnn.chk. If you did not specify the /f switch, **chkdsk** displays: x bytes would be freed. If you type N (for No), **chkdsk** frees the lost blocks so that they can be reallocated and does not recover any data that was in those lost blocks. If you did not specify the /f switch, **chkdsk** does nothing.

Logging to file x [Backup]

The **backup** command is writing a backup log to the file specified.

Logical DOS drive created, drive letters changed or added
[Fdisk]

You have created or revised one or more logical drives.

LPTx: not redirected [Mode]

Mode could not redirect the parallel printer port. Check to see whether you have specified the proper options.

LPTx: redirected to COMx: [Mode]

Output on the parallel printer port will now be sent to this asynchronous communications port.

LPTx: set for 80 [Mode]

The parallel printer port has been set for 80 columns.

LPTx: set for 132 [Mode]

The parallel printer port has been set for 132 columns.

Maximum available space for partitions is xxx cylinders
[Fdisk]

This is an informational message.

Maximum number of logical DOS drives installed [Fdisk]

You have installed the maximum number of logical DOS drives allowed by MS-DOS. You cannot create any more logical DOS drives.

Memory allocation error.

Cannot load MS-DOS, system halted [MS-DOS]

Restart MS-DOS. If this error persists, make a new copy of the MS-DOS diskette from your backup copy of the system diskette.

Missing from the file is either the device ID or the code page [Mode]

The code page specified is not supported in the code page information (.cpi) file, or the .cpi file does not support the printer specified. For a list of valid cplist values in the **mode** command, see Chapter 3, "MS-DOS Commands."

MODE *fff* code page function complete [Mode]

This message is informational only.

--More-- [More]

Press the space bar to view more of the file or directory.

MORE: Incorrect DOS version [More]

The **more** command does not run on MS-DOS versions before 2.0.

Must specify COM1, COM2, COM3 or COM4 [Mode]

You must specify a serial port.

Must specify destination line number [Edlin]

You did not specify the destination line number for an **edlin c** (copy) or **m** (move) command. Retype the command with a destination line number.

Must specify ON or OFF [MS-DOS]

The command requires either an ON or an OFF argument.

Name of list device [PRN]: [Print]

This prompt appears the first time that **print** is run and the /d switch is not specified. You can specify the name of any valid device, which then becomes the **print** output device. If you press **ENTER**, MS-DOS uses the default list device PRN.

New file [Edlin]

Edlin prints this message if it does not find a file with the name you specified. If you are creating a new file, ignore this message. If you do not intend to create a new file, check to see whether you have correctly typed the name of the file that you wish to edit.

NLSFUNC already installed [Nlsfunc]

Nlsfunc stays resident in memory once it is initialized. You have already loaded it into memory.

No Append [Append]

No paths have been appended. If you would like to append a path for data files, use the **append** command.

No appended directories [Append]

You did not specify a path with the **Append** command.

No code page has been selected [Chcp]

No code pages have been selected for the system. If you would like to select a code page, use the **chcp** command.

No COM: ports [Mode]

Your computer does not have an asynchronous communications (serial) port.

No files added (or replaced) [Replace]

The **replace** command did not add or replace any files.

No files found filename [Replace]

Replace could not find matching source or target files.

No free file handles.

Cannot start COMMAND.COM, exiting [MS-DOS]

Restart MS-DOS. If this message recurs, increase the files command value in the Config.sys file.

No logical drives defined [Fdisk]

There are no logical drives defined for your system.

No paper error writing device dev [MS-DOS device error]

The printer is either out of paper or not turned on.

No path [Path]

You typed **Path** and pressed **[ENTER]** to find out your search path, but you didn't set a command search path.

No primary DOS partition to delete [Fdisk]

You have selected the **fdisk** option to delete your primary DOS partition, but that partition does not exist.

No room for system on destination disk [Sys]

There is not enough room for the system files on the target disk. Delete some files to make room for the system files, or use another disk. You might need to reformat the disk to put the system on it.

No room in directory for file [Edlin]

You tried to create or save a file to the root directory, but it is either full, or you specified an invalid disk drive or filename. Check the command line that you used to start **edlin** for an invalid filename or disk drive entry. If your command contains no invalid entries, you should run the **chkdsk** program for the specified disk drive. If the status report shows that the disk directory is full and if there is still enough memory left on the disk, you might be able to create the file in a subdirectory. (This is because subdirectories are not limited in size as is the root directory.) Otherwise, remove the disk and replace it with another formatted disk.

No room in root directory [Label]

There is not enough room in the root directory for a volume label. Delete or move some of the files from the root directory to make room for the volume label.

No source drive specified [Backup]

You must specify a source drive.

No space left on device [Backup][Fc][Restore]

You cannot back up or restore any more files, and you cannot send any more output from a file comparison to your disk because the target disk is now full. You should probably delete some of the files on the disk to make more room.

No space to create logical drive [Fdisk]

You are trying to create a logical drive, but there is no space available to do so.

No sub-directories exist [Tree]

You have specified the /s switch, but the directory does not contain sub-directories.

No such file or directory [Backup][Fc][Restore]

One or more of the files or directories that you specified does not exist.

No target drive specified [Backup]

You must specify a target drive for this command.

No version of Graphic Character Set Table is loaded [Graftabl]

For information only.

Non-DOS disk error reading (or writing) drive x:

[MS-DOS device error]

MS-DOS does not recognize the disk format because the disk is missing information or contains another operating system. Try running the **chkdsk** command to correct the problem. (See Chapter 3, "MS-DOS Commands," for information about **chkdsk**.) If running **chkdsk** does not solve the problem, reformat the disk by using the **format** command even though this will destroy all the files on the disk.

Non-standard version of Graphic Character Set Table is already loaded [Graftabl]

MS-DOS cannot recognize the current table of graphics characters because it has been modified since it was loaded.

Non-system disk or disk error

Replace and strike any key when ready [Format][Sys]

Replace the disk with the proper disk, and press any alphanumeric key to continue.

***** Not able to back up (or restore) file**

******* [Backup]

This message can occur if there was an error in the source file or on the target disk. Use the **chkdsk** command on the source disk to see if you can determine the problem.

Not a graphics printer file [Graphics]

The file you are printing does not contain graphics.

Not enough memory [Join][Share][Subst]

There is not enough memory for MS-DOS to run the command.

Not enough room to merge the entire file [Edlin]

There was not enough room in memory to hold the file during an **edlin t** (transfer) command. You must free some memory by writing some files to a disk or by deleting some files before transferring this file.

Not found [Edlin]

You specified an **edlin s** (search) or **r** (replace) command that was unable to find a further occurrence of the specified search or replace string.

Not ready error reading (or writing) drive x:

[MS-DOS device error]

The device (usually a drive or printer) specified in the error message is not ready to accept or transmit data. This often happens when the diskette drive door is open. If this is the problem, close the door and type **r** (for Retry), or check to see if the printer is on and ready to print.

O.K.? [Edlin]

This is an **edlin** prompt that occurs during either the **s** (search) or **r** (replace) command processing. If you press any key except **Y** (for Yes) or **ENTER**, the search or replace process continues.

One or more CON code pages invalid for given language

[Keyb]

Keyb examined all prepared code pages and has found that at least one code page is incompatible for your screen console device (CON). This is only a warning to let you know that your keyboard and screen console device are working from different code pages.

Only non-bootable partitions exist [Fdisk]

None of the partitions left can boot MS-DOS.

Only partitions on drive 1 can be made active [Fdisk]

You are trying to create an active partition on a hard disk other than that found on the first hard disk drive. This is not allowed.

Out of environment space [Command][MS-DOS]

There is not enough room in the program environment to accept more data. To increase the size of the existing environment, use the **/e** switch with the **command** command, or remove some of the existing environment variables by using the **set** command.

Parameters not compatible [Format][Replace]

You have specified switches that cannot be used together.

Parameters not compatible with fixed disk [Format]

You have used a switch that is not compatible with the specified drive.

Parameters not supported [MS-DOS][Format]

You have specified parameters that MS-DOS does not support.

Parameters not supported by Drive [Format]

Format displays this message when the device driver for this drive does not support Generic IOCTL function requests.

Partition selected (x) is not bootable, active partition not changed [Fdisk]

You are trying to change active partitions, but MS-DOS cannot be booted from the partition selected.

Path(name) too long [Print][Replace][Xcopy]

The pathname you specified was too long. You might need to change directories to use this command with files in deep subdirectories.

Path not found [Chkdsk][Replace][Subst][Xcopy]

You specified an invalid pathname.

**Press any key to begin adding (replacing) file(s)
[Replace]**

When you specify the /w switch, **replace** displays this message to prompt you to start replacing files.

Press any key to begin formatting x: [Format]

This prompt is issued before you format a disk. Press any key to begin the format process. Or, if you wish to end this command, press **CTRL C**.

**Press any key to begin recovery of the
n file(s) on drive x: [Recover]**

This prompt is issued before you recover a disk or file. Press any key to begin the recovery. Recovered files are named Filennnn.rec. If you wish to end this command, press **CTRL C**.

Press any key when ready . . . [Diskcomp][Diskcopy]

This prompt gives you time to insert the appropriate diskettes before copying them. When you have inserted the diskettes into the appropriate drives, press any key to begin the **diskcopy** process. Or, if you wish to end this command, press **CTRL C**.

Previously prepared code page replaced [Mode]

This command changed the selected code page for a specific device by using another prepared code page.

Primary DOS partition already exists [Fdisk]

You are trying to create a primary DOS partition, but one already exists. If there is space available on your hard disk, try to create an extended DOS partition instead.

Primary DOS partition created [Fdisk]

You have successfully created a primary DOS partition on your disk.

Primary DOS partition deleted [Fdisk]

You have deleted the primary DOS partition from your disk.

Printer error [Mode]

The printer is off or is not ready to print.

Printer lines per inch set [Mode]

Mode has set the number of lines per inch for the printer.

PRINT queue is empty [Print]

There are no files waiting to be printed.

PRINT queue is full [Print]

There is only room for 10 files in the list of files waiting to be printed. You can make room for more by using the **print /q** switch. The limit is 32 files.

**Probable non-DOS disk
Continue (Y/N)? [Chkdsk]**

The disk you are using is not recognized by this version of MS-DOS. The disk was either created by another system with a format that is not supported on this version of MS-DOS or is not an MS-DOS disk.

Do not continue processing if **chkdsk** returns this message for a diskette. If this message returns for a hard disk, the information describing the characteristics of the disk to MS-DOS has been destroyed. In this case, you can continue **chkdsk** processing by typing Y (for Yes). This error can mean that the File Allocation Table (FAT) is bad and that the disk is unusable.

Processing cannot continue [Chkdsk]

There is not enough memory in your machine to run **chkdsk** for this disk. You must obtain more memory to run **chkdsk**.

Program too big to fit in memory [MS-DOS]

You need more memory to run your application. It is possible that some programs you have run are still using some memory. You can try to restart MS-DOS; however, if you still receive this message, you still need more memory.

Read error, COUNTRY.SYS [MS-DOS]

MS-DOS cannot read the Country.sys file. Retry the command. If you get the same message, the Country.sys file is probably corrupted. Restore the file from backup.

Read error in filename [Edlin][Find]

MS-DOS could not read the entire file.

Read error, KEYBOARD.SYS [MS-DOS]

MS-DOS cannot read the Keyboard.sys file. Retry the command. If you get the same message, the Keyboard.sys file is probably corrupted. Restore the file from backup.

Read fault error reading drive x: [MS-DOS]

MS-DOS is unable to read data from the device (usually a disk drive). Check to see that the diskette is properly inserted in the drive, and then type r (for Retry).

Reading source file(s)... [Xcopy]

Xcopy is now reading the source files that you specified.

Reinsert diskette for drive x: [Format]

Reinsert the diskette being formatted in the indicated drive.

Replace filename? (Y/N) [Replace]

Replace displays this prompt if you specify the /w switch. Type Y (for Yes) if you want to replace the existing file, or type N (for No) if you do not want to replace the file.

Replace the file (Y/N)? [Restore]

The file that you want to restore from backup already exists on your target disk. Type Y (for Yes) and press **ENTER** to overwrite the file. Type N (for No) and press **ENTER** if you don't want to replace the file on your target disk with the file from the backup diskette.

Replacing filename [Replace]

Replace displays this prompt to let you know that it is replacing this file that exists on your disk.

Requested logical drive size exceeds the maximum available space [Fdisk]

You are trying to create a logical drive that is larger than the space available.

Requested partition size exceeds the maximum available space [Fdisk]

You are trying to create a partition on your hard drive that is larger than the space available.

Requested Screen Shift out of range [Mode]

You cannot shift the display any farther.

Resident part of PRINT installed [Print]

This is the first message that MS-DOS displays when you issue the **print** command. It means that to process the **print** command with other processes, available memory has been reduced by several thousand bytes.

Resident portion of MODE loaded [Mode]

Part of the **Mode** program is now resident in memory, and available memory has been reduced by several thousand bytes.

Resident portion of NLSFUNC loaded [Nlsfunc]

Nlsfunc stays resident in memory once it is initialized. This informational message lets you know that you have already loaded **Nlsfunc** into memory.

Restore file sequence error [Restore]

You have restored files in the wrong order. You must insert the backup diskettes in the same order that they were backed up.

***** Restoring files from drive x: *****

Diskette: n [Restore]

This message is displayed during the restore process.

Resynch failed. Files are too different [Fc]

Fc compares what can be loaded into memory. If no lines match in the portion of the files in the buffer space, **Fc** displays this message.

Same drive specified more than once [Fastopen]

You tried to activate **fastopen** for the same drive more than once. There is no need to reactivate it for the same drive.

SECOND diskette bad or incompatible [Diskcomp]

The second diskette does not contain the same format as the first diskette, or **Diskcomp** does not recognize the format of the second diskette. You should run **chkdsk** to help you identify the problem.

Sector not found error reading (or writing) drive x:
[MS-DOS device error]

This error usually means the disk has a defective spot so that MS-DOS cannot find the requested information on it. You should copy all files from the disk onto a good disk and then try to reformat the defective disk.

Sector size too large in file filename [MS-DOS]

The specified device driver loaded by Config.sys uses a sector size larger than that of any other device driver on the system. You cannot run this device driver.

Seek error reading (or writing) drive x:
[MS-DOS device error]

MS-DOS is unable to locate the information on the disk. Be sure that the disk is properly inserted in the drive, or try a different drive.

SHARE already installed [Share]

Share can only be installed once.

Sharing violation reading drive x: [MS-DOS device error]

A program tried to access a file that another program was currently using. Type a (for Abort), or wait awhile and type r (for Retry).

SORT: Incorrect DOS version [Sort]

Sort does not run on MS-DOS versions before 2.0.

SORT: Insufficient disk space [Sort]

The disk is full.

SORT: Insufficient memory [Sort]

There is not enough memory to run the **sort** program.

Source and target drives are the same [Backup][Restore]

You specified the same drive for the source and target disks.

Source disk is Non-removable [Backup]

This is an informational message indicating that the source disk is a hard disk.

Source does not contain backup files [Restore]

You are attempting to restore files from a diskette that does not contain backup files.

Source is a floppy (or hard) disk [Restore]

This is an informational message only.

Source path required [Replace]

You did not specify a source path for the **replace** command.

**Specified drive does not exist,
or is non-removable [Diskcomp][Diskcopy]**

You cannot compare or copy hard disks with this command. You must specify the name of a valid diskette drive.

Specified MS-DOS search directory bad [MS-DOS]

The **shell** command in the Config.sys file is incorrect. Be sure that the Command.com file exists and that MS-DOS can find it.

Strike a key when ready... [MS-DOS]

This prompt occurs during command processing and is always accompanied by another message. This message is also displayed if you have inserted a **pause** command in a batch file. Usually, MS-DOS asks you to insert diskettes into appropriate drives before this prompt. To begin command processing, press any character, any number key, the space bar, or **ENTER**.

Syntax error [Attrib][Find][MS-DOS]

You have entered a command incorrectly. Check to be sure you have typed the command correctly. Remember to enclose the **find** command string in double quotation marks.

System transferred [Format][Sys]

The system files were transferred during **format** or **sys** command processing.

Target cannot be used for backup [Backup]

Either the target disk has an unrecognizable format, or it is bad. Do not use the disk, try to format the disk with the **format** command, or run **chkdsk** on it to determine the problem.

Target disk is Non-removable [Backup]

This is an informational message that the target disk is a hard disk.

Target diskette is write protected [Diskcopy]

The target diskette either has a write-protect tab on it, or it does not have a write-protect notch. If you want to destroy any existing information on the diskette, remove the write-protect tab and give the command again. If the diskette does not have a write-protect notch, you cannot use it as a target diskette.

Target diskette may be unusable [Diskcopy]

Either the target diskette has an unrecognizable format, or it is bad. Try to format the diskette with the **format** command, or run **chkdsk** on it to determine the problem.

Target is a floppy (or hard) disk [Backup]

This is an informational message only.

Target is full [Restore]

There is no more room on the target disk for restored files. You must delete some of the files on the disk to make room for these files, or use another disk.

Target is Non-Removable [Restore]

This is an informational message only.

Terminate batch job (Y/N)? [MS-DOS]

If you press **CTRL** **C** while in batch mode, MS-DOS asks you whether or not you wish to end batch processing. Type Y (for Yes) to end processing, or type N (for No) to continue.

**The current active keyboard table is xx with code page: yyy
The current active CON code page is zzz [Keyb]**

This is an informational message that shows the current keyboard code, code page for the system, and code page for the console (screen).

The last file was not restored [Restore]

There was not enough room on the target disk for the file, or the last file was bad. Use the **chkdsk** command to determine the problem.

The only bootable partition on drive 1 is already marked active [Fdisk]

You are trying to change the active partition. The active partition must reside on the first hard disk drive on your system and must be bootable. The only bootable partition on the first hard disk drive is already the active partition.

Too many drive entries [Fastopen]

You can use **fastopen** with up to four hard drives. You have tried to specify a fifth drive.

Too many files open [Edlin][Label]

MS-DOS could not open the .bak file or write the volume label due to the lack of available system file handles. Increase the value of the files command in the Config.sys file.

Too many name entries [Fastopen]

The total number of entries specified for a drive exceeded the maximum of 999.

Too many open files [Backup][Fc][Restore][Xcopy]

MS-DOS could not open the files that you want to compare due to the lack of available system file handles. Increase the value of the files command in the Config.sys file.

Track 0 bad - disk unusable [Format]

The **format** command can accommodate defective sectors on the disk, except for those near the beginning. You must use another disk.

Unable to create directory [Mkdir][Xcopy]

MS-DOS could not create the directory you specified. Check to see that there is not a name conflict. You might have a file with the same name, or the disk might be full.

Unable to create KEYB table in resident memory [Keyb]

MS-DOS tried to create a country-specific table for the keyboard code specified, but failed. Check the amount of available memory. There might not be enough memory available to create this table.

Unable to erase [Backup]

Backup could not erase the files on the target disk. Check to see that the files on the backup diskette are not read only and that the disk is not write-protected.

Unable to shift Screen [Mode]

Mode is unable to shift the test pattern on the screen any farther.

Unexpected DOS Error *n* [Replace]

An unexpected error *n* occurred, where *n* is the MS-DOS error number.

Unrecognized command in CONFIG.SYS [MS-DOS]

There is an invalid command in your Config.sys file. Refer to Appendix B, "How to Configure Your System," for a list of valid statements.

Unrecognized printer [Graphics]

You are using an invalid printer. Check to see whether you entered the command properly, or refer to Chapter 3, "MS-DOS Commands," to be sure that you have specified a valid printer name.

Unrecognized printer port [Graphics]

The printer device name that you specified was invalid. You might need to set the printer port by using the **mode** command.

Unrecoverable error in directory

Convert directory to file (Y/N)? [Chkdsk]

This message is displayed if **chkdsk** is unable to correct an error in a directory. If you respond Y (for Yes) to this prompt, **chkdsk** converts the bad directory into a file. You can then fix the directory or delete it. If you respond N (for No) to this prompt, you might not be able to write to or read from the bad directory.

Unrecoverable read (or write) error on drive *x*:

[MS-DOS device error]

MS-DOS is unable to read or write data to the specified device. Be sure that the diskette is properly inserted in the diskette drive. Then, type *r* (for Retry). If the error occurs again, type *a* (for Abort).

**usage: fc [/a] [/b] [/c] [/l] [/lb *n*] [/w]
[t] [/n] [/NNNN] file1 file2 [Fc]**

One of the switches that you have specified is invalid.

VERIFY is off (or on) [MS-DOS]

This message tells you the current setting of the **verify** command.

nnn version of Graphic Character Set Table is already loaded [MS-DOS]

You tried to load the Graphic Character Set Table with the **graftabl** command. This table already exists in memory.

nnn version of Graphic Character Set Table is now loaded [Graftabl]

You have loaded the Graphic Character Set Table into memory.

Volume in drive x: has no label [Dir][Label][Vol]

This is an informational message displayed in response to the **dir**, **label**, or **vol** command.

Volume in drive x: is filename [Dir][Label][Vol]

This is an informational message displayed in response to the **dir**, **label**, or **vol** command.

Volume label (11 characters, ENTER for none)? [Format][Label]

MS-DOS displays this message when you specify the **label** command or when you use the **/v** switch in the **format** command. Type a volume label, or press **ENTER** to indicate that you do not want a volume label for this disk. It's a good idea, though, to specify a volume label to help you identify your disks.

WARNING, ALL DATA ON NON-REMOVABLE DISK DRIVE x: WILL BE LOST!

Proceed with Format (Y/N)? [Format]

This message appears when you try to format a hard disk that already contains data. If you type **Y** (for Yes) the data on the disk is erased. If you do not want the files on your hard disk erased, type **N** (for No). Copy the files to a diskette, and repeat the **format** command.

Warning! Date in the extended DOS partition could be destroyed.

Do you wish to continue.....? [n] [Fdisk]

You are trying to delete an extended DOS partition. To continue with the operation and delete the extended DOS partition specified, type Y (for Yes) and press **[ENTER]**. Or, if you do not want to delete the partition, press **[ENTER]** to select the default response N (for No).

Warning! Date in the primary DOS partition could be destroyed.

Do you wish to continue.....? [n] [Fdisk]

Fdisk is about to delete the primary DOS partition from your hard disk. To continue the deletion process and remove the partition, type Y and press **[ENTER]**. If you do not want to remove the partition, select the default N (for No) response by only pressing **[ENTER]**.

Warning - directory full [Recover]

The root directory is too full for **recover** processing. Delete some files in the root directory to free space for the recovered files, and try the command again.

Warning! Diskette is out of sequence

Replace diskette or continue if okay

Strike any key when ready [Restore]

You should restore the diskettes in the order that you backed them up.

Warning! File filename

is a hidden (or read-only) file

Replace the file (Y/N)? [Restore]

This message prompts you as to whether you want to replace a hidden or read-only file. Type Y (for Yes) if you want to restore the hidden or read-only file from the backup diskette. Type N (for No) if you do not want to restore this file.

**Warning! File filename
was changed after it was backed up
Replace the file (Y/N)? [Restore]**

This message prompts you as to whether you want to replace a backup file that has been changed. Type Y (for Yes) if you want to restore this file, or type N (for No) if you do not.

**Warning! Files in the target drive
BACKUP (or root) directory will be erased [Backup]**

Backup found files in the target drive, and you did not specify the /a switch to append files.

Warning! No files were found to back up [Backup]

Backup did not find any files to back up on the diskette you specified.

Warning! No files were found to restore [Restore]

Restore did not find the file that you wanted to restore from the backup diskette.

Warning: Read error in EXE file [Exe2bin]

The amount read was less than the size of the header. This is a warning message only.

**Warning! The partition marked active is not bootable
[Fdisk]**

The active DOS partition must be bootable.

Write fault error writing drive x: [MS-DOS device error]

MS-DOS is unable to write data to the specified device. Be sure that the diskette is properly inserted in the diskette drive. Then, type r (for Retry). If the error occurs again, type a (for Abort).

Write protect error writing drive x: [MS-DOS device error]

You tried to write data on a write-protected diskette. If the diskette has a write-protect tab on it, you must remove the tab before you can write on the diskette. (You should consider first why the diskette was write-protected.) If the diskette doesn't have a write-protect notch, you cannot write on that diskette.

Index

Numbers/Symbols

", in strings, 81, 226, 258
#, in Edlin, 196
%%c variable, 169, 170
%0-%9, in batch files, 162-164,
169-170
' , in strings, 226, 258
* wildcard, 7, 8
*, current line marker, Edlin, 196
*, Edlin prompt, 186
+ a attribute, 32
+ r attribute, 32
- prompt, Debug, 222, 254
-a attribute, 32
-r attribute, 32
. and .., shorthand for parent
directories, 6
., in Edlin, 196
24-hour clock, 146
:, in batch files, 171
:, in Debug commands, 224-225
:, in commands, 23
:, in Debug, 225
<, for redirecting input, 17
=, in commands, 23
> and > >, appending and
redirecting output with, 16
? option, Edlin, 197
? wildcard, 7
\\, in directories, 4

Numbers/Symbols (*Continued*)

\\, in pathnames, 5
, for piping commands, 17

A

aborting activities and commands
 See CTRL-C
address, Debug parameter, 225
allocation of files, 2
alphabetizing file contents, 141-
142
ALT- = keys, 114
Ansi.sys
 cursor functions, 284-285
 erase functions, 285
 escape sequences, 283-287
 installing, 283
 screen graphics affected by,
 286-287
 screen width parameters, 287
Append (a), Edlin command, 198
Append command, 27-29
appending
 files, 59-60
 output, 16
archive attribute, setting, 32
arglist, definition, 223
arguments command option, 22
Assemble, Debug command,
 227- 229
Assign command, 30-31

- asterisk (*)
 - * wildcard, 7, 8
 - current line marker in Edlin, 196
 - Edlin prompt, 186
- asynchronous communication
 - modes, 108
- Attrib command, 32-33
- Autoexec.bat file
 - creating, 160, 162
 - Date and Time commands in, 160
 - definition, 159
 - functions of, 159-160
- Autofmt command, 34-36
 - See also Format command

B

- background printing, 120-122
- backslash (\)
 - in directories, 4
 - in pathnames, 5
- Backspace key, 193
- Backup command, 37-39
 - exit codes, 39
 - switches for, 38
 - See also COPY command;
DISKCOPY command;
RESTORE command;
XCOPY command
- Backup.log file, 38
- bad sector information, 34-35
- bad sectors, recovering, 127
- .BAK files, editing, 187
- .BAT extension, 157
- .BAT files, 14
 - See also batch files

batch files

- Autoexec.bat file, 159-160
- command execution in, 159
- creating, 157-158
- named parameters in, 163-164
- replaceable parameters in, 162-163
- rules for, 158-159
- running, 158, 165
- temporary files, 166
- terminating, with CTRL-C, 158, 174
- uses for, 157
- See also batch processing commands

batch processing, definition, 157

batch processing commands

- Call, 167
- Echo, 168
- For, 169-170
- Goto, 171
- If, 172-173
- Pause, 174
- Rem, 175
- Shift, 176-177
- summary chart, 166
- See also batch files

baud rate, 108

BF (bad flag), error code, 258

block device drivers, Drivparm command and, 270

BP (too many breakpoints), error code, 258

BR (bad register), error code, 258

Break command, 40

- in Config.sys, 266

breakpoints, with Go command,
236-237

Buffers command, in Config.sys,
267

byte, Debug parameter, 225

C

Cache command, 41-44

options, 42-44

switches for, 42

Call, batch processing command,
167

CD command

See Chdir command

change directory command

See Chdir command

Chcp command, 45-46

Chdir command, 47-48

CD as abbreviation for, 9

changing directories, 9

displaying directories, 9

check disk command

See Chkdsk command

Chkdsk command, 2, 49-50

switches for, 50

clearing the screen, 51

Cls command, 51

code pages

changing or displaying with

Chcp, 45-46

country codes, 319-320

date and time formats, 322-
323

definition, 319

device code pages, setting,
325

keyboard codes, 319-320

code pages (*Continued*)

listing current code pages,
327

MS-DOS support for, 319

Printer.sys variables, 298

refreshing lost code pages,
327-328

switching between, 288, 326

using, 323-324

See also national language
support

colon (:)

in batch files, 171

in Debug commands, 224-225

column parameter, Ansi.sys, 283

.COM files, 14

combining files, with Copy, 59-60

Command command, 52-53

switches for, 53

command options

arguments, 22

conventions for, 23

definition, 22

drive:, 22

filename, 22

path, 22

pathname, 22

string, 22

switches, 22

command processor

See Command command

commands

Append, 27-29

Assign, 30-31

Attrib, 32-33

Autofmt, 34-36

Backup, 37-39

Break, 40

commands (*Continued*)

- Cache, 41-44
- Chcp, 45-46
- Chdir, 47-48
- Chkdsk, 49-50
- Cls, 51
- Command, 52-53
- Comp, 54-56
- Copy, 14, 57-60
- Ctty, 61
- Date, 63
- Date command, 62
- Dc, 64-65
- Del, 14, 67
- Dir, 8, 14, 68-69
- Disktype, 66
- external commands, 14-15
- internal commands, 13-14
- Mcopy, 155
- Mkdir, 8-9
- networks and, 20
- Path, 15
- Restore, 38-39
- Rmdir, 10-11
- sample description, 21
- separators (punctuation) for, 23
- summary list of, 24-26
- symbols for, 19
- Type, 14
- types of, 13
- Xcopy, 153-155
- See also batch processing
 - commands; Config.sys
 - Debug commands;
 - Edlin commands
- Comp command, 54-56
- Compare, Debug command, 230

- comparing
 - disks, with Diskcomp, 70-72
 - files, with Comp, 54-56
- compressing files, with Dc, 64-65
- concatenating files, with Copy, 59-60
- conditions, with If command, 172
- Config.sys
 - Break, 266
 - Buffers, 267
 - Country, 268
 - definition, 263
 - Device, 269
 - Drivparm, 270-272
 - FCBS, 273
 - Files, 274
 - Lastdrive, 275
 - list of commands, 264
 - sample file, 264, 265
 - Shell, 276
 - Stacks, 278
 - Temmm.sys and, 279-280
- configuring hard disks
 - See hard diskpreparing
- configuring your system
 - See Config.sys
- control characters
 - inserting, in Edlin, 195
 - list of, 183
 - using, 183
 - See also specific control characters, such as CTRL-C
- conventions, for command options, 23
- Copy (c), Edlin command, 199

Copy command, 57, 60
 pathnames with, 14
 switches for, 58-59
 See also Backup command;
 Diskcopy command;
 Xcopy command
country codes, 319-320
Country command, in Config.sys,
 268
country-specific information
 formatting a disk with, 328-
 329
CTRL-ALT-DEL, restarting the
 system with, 36
CTRL-C
 aborting Debug commands
 with, 221
 Break command and, 40
 stopping activities with, 40,
 183
 terminating batch files with,
 158, 174
CTRL-H, 183
CTRL-J, 183
CTRL-N, 183
CTRL-P, 183
CTRL-S
 starting and stopping display
 with, 183, 221
CTRL-V, 195
CTRL-X, 183
CTRL-Z, 55, 195-196
Ctty command, 61
cursor functions, Ansi.sys, 284

D

databits, 108
date and time formats, 322-323
Date command, 62-63
Dc command, 64-65
 switches for, 64
dead keys, 101
Debug
 aborting commands, 221
 error codes and messages,
 258
 introduction, 221
 starting, 222-223
 suspending the display, 221
 See also Debug command
 parameters; Debug
 commands
Debug command parameters
 address, 225
 byte, 225
 drive, 225
 list, 226
 range, 225-226
 record, 225
 string, 226
 value, 225
Debug commands
 : in, 224-225
 Assemble, 227- 229
 Compare, 230
 delimiters for, 224
 Dump, 231
 editing and function keys in,
 223
 Enter, 233-234,
 Fill, 235
 Go, 236-237

Debug commands (*Continued*)

- Hex, 238
- Input, 239
- Load, 240-241
- Move, 242
- Name, 243-244
- Output, 246
- Quit, 247
- Register, 248-250
- Search, 251
- summary chart of, 224
- syntax errors, 223
- Trace, 252-253
- Unassemble, 254-255
- uppercase and lowercase letters in, 223
- Write, 256-257
- See also Debug; Debug command parameters

decoded instructions, Trace command and, 252

Del command, 10, 67

- pathnames with, 14

DEL key, 180, 190

Delete (d) command, Edlin, 200

delete command

See Del command

delimiters

in Debug commands, 224

in strings, 226

device code page modes, 109

abbreviations for, 111

device code pages, setting, 325

Device command, in Config.sys, 269

device drivers

definition, 281

Device command and, 269

device drives (*Continued*)

Display.sys, 288

Driver.sys, 289-290

Drivparm command and,
270-271

Hdrive.sys, 291-292

Lpdvr.sys, 293-297

Modevm.sys, 299

Printer.sys, 298

Spooler.sys, 300

summary chart of, 281-282

Vdisk.sys, 301-302

See also names of specific
drivers, such as Ansi.sys

devices, Ctty command and, 61

DF (double flag), error code, 258

Dir command, 68-69

pathnames with, 14

switches for, 68

wildcards and, 8

directories

Append command and, 27-29

backslash (\) in, 4

changing with Chdir, 9-10

Copy command and, 57

creating with Mkdir, 8-9

creating, with Mkdir, 105

deleting all files in, 10-11

deleting with Rmdir, 10-11

displaying with Chdir, 9-10

Fastopen command and, 79

maximum number of files and
directories in, 3

parent directories, 6

Path command and, 119

pathnames and, 5-6

paths, 6

Ren command and, 128

- directories (*Continued*)
 - renaming, 11
 - root directory, 2-3
 - subdirectory names and filenames, 10
 - traveling through, 3
 - uses for, 2-3
 - variation in display of, 69
 - wildcards and, 6-8
 - working directory, 3, 9-10
 - Xcopy command and, 153, 154-155
- disassembling code, with
 - Unassemble command, 254-255
- disk buffer, definition, 267
- disk cacheing
 - See Cache command
- disk drives
 - Assign command and, 30-31
 - Join command and, 98-99
 - Lastdrive command and, 275
 - single disk drive systems, 259-261
 - Subst command and, 143-144
 - types and valid switches, 260
- Diskcomp command, 70-72
 - exit codes, 72
 - switches for, 70
- Diskcopy command, 73-75
 - exit codes, 74
 - See also Backup command;
Copy command; Xcopy
command
- diskettes
 - See disks
- Diskopt command, 76-77
- disks
 - checking with Chkdsk, 2, 49-50
 - comparing with Diskcomp, 70-72
 - country-specific information, formatting with, 328-329
 - formatting, 85-89
 - high capacity, formatting, 83-84
 - optimizing, with Diskopt, 76-77
 - protecting, 1
 - recovering, 127
 - size and capacity, displaying, 66
 - types and format switches, 88, 260
 - See also hard disks
- Disktype command, 66
- display modes, 109
- Display.sys
 - installing, 288
 - variables, 288
- displaying file contents, 149
- drive: command option, 22
- drive, Debug parameter, 225
- Driver.sys, 289-290
 - installing, 289
 - switches for, 289-290
- drives
 - See disk drives
- Drivparm command, in Config.sys, 270-272
 - switches for, 270-271
- dummy parameters, in batch files, 162, 163
- Dump, Debug command, 231

E

Echo, batch processing
 command, 168
editing and function keys, 179-180
 Backspace, 193
 Debug commands and, 223
 DEL, 180, 190
 Edlin editing keys, 188
 ENTER key, 186, 196
 ESC key, 180, 191
 F1, 180, 188
 F2, 180, 189
 F3, 180, 190
 F4, 180, 190
 F5, 180, 192
 F6, 180
 INS, 180, 191
 See also control characters;
 template, MS-DOS
Edlin
 * prompt, 186
 .BAK files and, 187
 functions and uses, 185
 quitting, 187
 saving files, 187
 starting, 185, 186
 summary chart of editing
 keys, 188
 See also Edlin commands
Edlin commands
 Append (a), 198
 command options, 196-197
 Copy (c), 199
 Delete, 200
 End/Save (e), 203
 Insert (i), 204-205
 Line Edit, 201-202
 List (L), 206

Edlin commands (*Continued*)

 Move (m), 207-209
 multiple commands, 195
 Paging (p), 210
 pathnames and, 194
 Quit/Nosave (q), 211
 referencing line numbers, 195
 Replace (r), 212-214
 Search (s), 215-217
 summary chart of, 194
 tips for using, 194-196
 Transfer (t), 218
 Write (w), 219
 See also Edlin
encryption of files, 125-126
end-of-file mark (CTRL-Z), 55
End/Save (e) command, Edlin,
 203
Enter, Debug command, 233-235
ENTER key, in Edlin, 186, 196
environment
 Command.com and, 52-53
 Set command and, 137-138
equal sign (=), in commands, 23
Erase command
 See Del command
erase functions, Ansi.sys, 285
error messages
 Debug, 258
 MS-DOS message directory,
 331-391
ESC key, 180, 191
escape sequences, Ansi.sys, 283-
 287
.EXE files, 14
exit codes
 Backup command, 39
 Diskcomp, 72

exit codes (Continued)

Diskcopy command, 74

Format command, 88

Graftabl command, 91

Replace command, 131

Restore command, 133

Xcopy command, 155

Exit command, 78

expanded memory

Cache command and, 42

Tmm.sys and, 279-280

extended character set

See characters

extended memory

Cache command and, 42

Mon386 command and, 113

external commands

definition, 14

list of, 15

order of precedence, 14

paths with, 15

symbol for, 19

F

F1 key, 180, 188

F2 key, 180, 189

F3 key, 180, 190

F4 key, 180, 190

F5 key, 180, 192

F6 key, 180

Fastopen command, 79

FCBS command, in Config.sys,
273

Fdisk command, 80

menu for, 307

selecting next fixed disk
drive, 317

starting, 306-307

Fdisk command (Continued)

uses for, 306

See also hard disks –

partitioning

file allocation table, 2

File Control Blocks (FCBS)

command, in Config.sys, 273

filename command option, 22

filename option, device code

page modes, 109

filenames

difference from directory

names, 10

interchangeability with

pathnames, 6

See also wildcards

files

allocation, 2

appending, with Copy, 59-60

attributes, setting and

displaying, 32-33

comparing, with Comp, 54-56

compressing, with Dc, 64-65

displaying contents of, 149

encrypting, 125-126

Fastopen command and, 79

maximum number of, in

directories, 3

number allowed in print

queue, 121

number of open files, 274

protecting, 1

recovering, 127

restoring backup files, 132-133

sharing and locking, 140

updating with Replace, 129-

131

- files (*Continued*)
 - verifying while copying, 151
 - See also batch files
 - Files command, in Config.sys, 274
 - Fill, Debug command, 234
 - filters and pipes
 - command pipes, 17-18
 - definition, 17
 - list of filters, 17
 - See also redirecting input and output
 - Find command, 81-82
 - switches for, 81
 - fixed disks
 - See hard disks
 - flags
 - Register command and, 248-250
 - Trace command and, 252-253
 - Fmat2000 command, 83-84
 - switches for, 83
 - For, batch processing command, 169-170
 - foreign keyboard programs, 100-101
 - foreign languages
 - See national language support
 - Format command, 85-89
 - disk types and switches, 88
 - exit codes, 88
 - switches for, 86
 - See also Autofmt command;
Fmat2000 command;
Hsect command
 - formatting
 - Autofmt command, for hard disks, 34-36
 - Format command for, 85-89
 - high-capacity diskettes, 83-84
 - Hsect command, for hard disks, 95-97
 - with country-specific information, 328-329
 - function keys
 - See editing and function keys
- G**
- Go, Debug command, 236-237
 - Goto, batch processing command, 171
 - Graftabl command, 90-91
 - exit codes, 91
 - graphics characters
 - See Graftabl command
 - Graphics command, 92-94
 - switches for, 93
 - graphics, screen, Ansi.sys, 286-287
 - greater-than sign (>), 16
- H**
- halting activities and commands
 - See CTRL-C
 - hard disks – formatting
 - with Autofmt, 34, 35, 36
 - with country-specific information, 328-329
 - with Format command, 86-87
 - with Hsect, 95-97
 - hard disks – partitioning
 - changing the active partition, 312
 - deleting, 313-315

hard disks — partitioning
 (Continued)
 displaying partition data, 316
 extended DOS partition, 310
 logical drive in extended DOS
 partition, 311
 menu for creating, 308
 more than one partition,
 309-310
 primary partition, 308-309
 with Autofmt, 34-36
 hard disks — preparing
 checking for configured and
 formatted disk, 305
 procedure for, 303
 types of hard disks, 304-305
 See also Fdisk command
 hard disks, non-standard
 See Hdrive.sys
 Hdrive.sys, 291-292
 chart of drive types, 291
 procedure for using, 292
 uses, 291
 Hex, Debug command, 238
 hierarchical directories
 See directories
 Hsect command, 95-97
 hyphen (-) prompt, Debug, 222

I

If, batch processing command,
 172-173
 initializing hard disks
 See hard disks — preparing
 Input, Debug command, 239
 INS key, 180, 191
 Insert (i), Edlin command, 204-
 205

internal commands
 definition, 13
 list of, 13
 pathnames with, 13-14
 symbol for, 19
 international time, date, currency,
 and case conversions, 268

J

Join command, 98-99

K

Keyb command, 100-101
 keyboard codes, 319-320
 keyboard programs, foreign
 switching between, 101
 See also national language
 support
 keys and key combinations
 See control characters; editing
 and function keys; names of
 specific keys and key
 combinations
 keywords
 abbreviations for, 111
 for device codepage
 command, 110

L

Label command, 102-103
 See also Vol command
 languages, foreign
 See national language
 support
 Lastdrive command, in
 Config.sys, 275

- less-than sign (<), for redirecting input, 17
- Lf command, 104
- Line Edit, Edlin command, 201-202
- line editor
 - See Edlin
- line feeds
 - enabling and disabling, 107
 - suppressing, with Lf, 104
- line numbers, referencing in Edlin, 195
- line option, Edlin, 196
- line parameter, Ansi.sys, 283
- List (L), Edlin command, 206
- list, Debug parameter, 225-226
- Load, Debug command, 240-241
- lowercase and uppercase letters
 - Debug commands and, 223
 - Sort command and, 142
- Lpdrv.sys
 - functions of, 293
 - installing, 293-294
 - printer control codes, 294-297
- LPT1, 2, and 3, device name, 107, 120
- M**
- make directory command
 - See Mkdir command
- Mcopy command, 155
- Md command
 - See Mkdir command
- memory
 - See CMOS memory;
 - expanded memory; extended memory
- messages
 - in batch files, 168
 - MS-DOS message directory, 331-391
- Mkdir command, 8-9, 105
- mnemonics, with Assemble command, 227
- Mode command, 106-112
 - asynchronous communication modes, 108
 - CPU speed modes, 111
 - device code page modes, 109-110
 - display modes, 109
 - parallel printer mode, 107
- Modevm.sys, 299
- Mon386 command, 113-115
- Monitor menu, 114
- monitors
 - setting display modes, 109
 - VM-1, installing, 299
- More command, 116-117
- Move
 - Debug command, 242
 - (m), Edlin command, 207-209
- MS-DOS
 - See operating system
- multilevel directories
 - See directories
- Multitasking
 - See Mon386 command
- N**
- Name, Debug command, 243-245
- named parameters, in batch files, 163-164

- national language support
 - commands supporting, 321-322
 - country codes, 319-320
 - formatting disks for, 328-329
 - keyboard codes, 319-320
 - See also code pages;
keyboard programs,
foreign
- networks
 - Append command and, 28
 - nonfunctional commands
with, 20
 - sharing and locking files, 140
- Nlsfunc command, 118
- non-standard hard disks, setting
up. See Hdrive.sys
- NUL path, 28, 119
- numeric parameter, Ansi.sys, 283
- O**
- opcodes, with Assemble
command, 228-229
- operating system
 - displaying version of, 150
 - illustration of function, 161
 - installing, with Select, 135-
136
 - transferring while formatting,
86
 - transferring with Sys, 145
- optimizing disks, 76-77
- Output, Debug command, 246
- P**
- Paging (p), Edlin command, 210
- parallel printer mode, 107
- parameters, Ansi.sys, 283-287
- parameters, Debug
 - address, 225
 - byte, 225
 - drive, 225
 - list, 226
 - range, 225-226
 - record, 225
 - string, 226
 - value, 225
- parameters, named, in batch files,
163-164
- parameters, replaceable
 - creating, in batch files, 162-
163
 - defining, with Set, 137
 - difference from %%c
variable, 169-170
 - shifting, 176-177
- parent directories, 6
- parity, 108
- partitioning hard disks
 - See hard disks – partitioning
- Path command, 15, 119
- path command option, 22
- pathname command option, 22
- paths and pathnames, 6
 - \ in, 5
 - Edlin commands and, 194
 - examples, 6
 - external commands and, 15
 - interchangeability with
filenames, 6
 - internal commands and,
13-14
 - number of characters
allowed in, 5
 - Subst command and, 143-144

- pause of screen display
 - with CTRL-S, 183, 221, 253
 - with More, 116-117
- Pause, batch processing command, 174
- percent signs (%), in batch files, 162-164
- period (.), current line marker, in Edlin, 196
- pipes
 - See filters and pipes
- ports, asynchronous communication, 108
- pound sign (#), in Edlin, 196
- prefix mnemonics, with Assemble command, 227
- prepare keyword, 110
- Print command, 120-122
 - switches for, 120
- print spooler
 - See Spooler.sys
- Printer.sys, 298
 - installing, 298
 - options, 298
- printers
 - control codes, with
 - Lpdvr.sys, 294-297
 - graphics printer options, 92
 - parallel printer mode, 107
 - redirecting printer output, 112
 - time-out error, 107
- printing the screen, 94
- PRN device name, 107, 120
- Prompt command, 123-124
- pseudo-instructions, with
 - Assemble command, 228

- punctuation, in commands, 23
 - See also numbers/symbols category at top of index

Q

- question mark (?) option, Edlin, 197
- question mark (?) wildcard character, 7
- Quit, Debug command, 247
- Quit/Nosave (q), Edlin command, 211
- quotation marks ("), in strings, 81, 226

R

- RAM disk
 - See Vdisk.sys
- range, Debug parameter, 225-226
- Rd command
 - See Rmdir command
- read-only attribute, setting, 32-33
- record, Debug parameter, 225
- Recover command, 127
- Recrypt command, 125-126
- redirecting input and output
 - appending output, 16
 - pipes and, 17
 - redirecting input, 17
 - redirecting output, 16
 - redirecting printer output, 112
- Sort command and, 141-142
 - See also filters and pipes
- refresh keyword, 110
- Register, Debug command, 248-250

registers
 Register command and, 248-250
 Trace command and, 252-253
Rem, batch processing command, 175
Remove directory command
 See Rmdir command
Ren command, 128
Rename command
 See Ren command
renaming directories, 11
Replace (r), Edlin command, 212-214
Replace command, 129-131
 exit codes, 131
 switches, 130
replaceable parameters
 creating, in batch files, 163
 defining, with Set, 137
 difference from %%c variable, 169-170
 in batch files, 162
 shifting, 176, 177
restarting the system, with CTRL-ALT-DEL, 36
Restore command, 38-39, 132-133
 exit codes, 133
 switches for, 132
 See also Backup command
Rmdir command, 10-11, 134
root directory, 2-3
 returning to, 48
RS-232 port
 See asynchronous communication modes

S

saving files, Edlin, 187
screen display
 controlling with More, 116-117
 printing, 94
 starting and stopping with CTRL-S, 183, 221, 253
screen graphics, Ansi.sys, 286-287
screen width parameters, Ansi.sys, 287
Search
 Debug command, 250
 (s), Edlin command, 215-217
search path
 See paths and pathnames
Select command, 135-136
select keyword, 110
selective parameter, Ansi.sys, 283
semicolons (;) in commands, 23
separators, in commands, 23
 See also numbers/symbols category at top of index
serial communications, 108
Set command, 137-138
Setup command, 139
shadow RAM, 44
Share command, 140
 switches for, 140
Shell command, in Config.sys, 276
shift display mode, 109
SHIFT PRTSC keys, 94
Shift, batch processing command, 176-177
shorthand notation for parent directories, 6
single disk drive systems, 259-261

Sort command, 141-142
 switches for, 141
spaces, in commands, 23
Spooler.sys, 300
 installing, 300
 switches for, 300
 uses for, 300
Stacks command, in Config.sys,
 278
status keyword, 110
stopbits, 108
stopping activities and commands
 See CTRL-C
string command option, 22
string, Debug parameter, 226
strings
 quotation marks (") in, 81,
 226
 setting, 137-138
subdirectories
 See directories
Subst command, 143-144
switches
 See specific commands and
 device drivers for
 switches used with
switches command option, 22
Sys command, 145
system
 See operating system
system command
 See Sys command

T

t display mode, 109
Tandy Expanded Memory
 Management. See Temm.sys

Temm.sys
 definition, 279
 installation, 279-280
 use of shadow RAM, 44
template, MS-DOS, 181
 correcting errors, 182
 definition, 179
 using, 181, 182
 See also editing and function
 keys
temporary batch files, 166
test pattern (t) display mode, 109
text option, Edlin, 197
time and date formats, 322-323
Time command, 146-147
time-out error, 107
Trace, Debug command, 252-253
Transfer (t), Edlin command, 218
Tree command, 148
 switches for, 148
24-hour clock, 146
Type command, 149
 pathnames and, 14

U

Unassemble, Debug command,
 254-255
updating files with Replace, 129-
 131
uppercase letters
 See lowercase and uppercase
 letters

V

value, Debug parameter, 225
variables
 %%c, 169-170
 Display.sys, 288

Vdisk.sys
 function, 301
 installing, 301
 options, 301-302
 switches, 302
Ver command, 150
Verify command, 151
version of MS-DOS, displaying,
 150
vertical var (|), piping commands
 with, 17
virtual disk, 301
virtual drives, 143
VM-1 monitors, installing, 299
Vol command, 152
 See also Label command
volume label, 86-87
 invalid characters in, 103
 See also Label command;
 Vol command

W

wildcards
 * wildcard, 7, 8
 ? wildcard, 7
 definition, 6
 Del command and, 67
 Dir command and, 68
 Ren command and, 128
working directory
 changing with Chdir, 9
 definition, 3
 displaying with Chdir, 9-10
 illustration of, 4
Write
 Debug command, 256-257
 (w), Edlin command, 219

X

Xcopy command, 153-155
 exit codes, 155
 switches for, 154
 See also Backup command;
 Copy command;

